



**INSTITUTO POLITÉCNICO NACIONAL**  
**CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN**



**UN ALGORITMO GENERAL PARA LA DESCOMPOSICIÓN  
SEMÁNTICA DE GEO-IMÁGENES**

**TESIS QUE PARA OBTENER EL GRADO DE  
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN**

**PRESENTA**

**M. EN C. JOSÉ GIOVANNI GUZMÁN LUGO**

**DIRECTOR DE TESIS**

**DR. SERGUEI LEVACHKINE**

**MÉXICO, D.F., NOVIEMBRE 2007**

# Contenido

Listado de acrónimos y siglas .....	vii
Índice de figuras .....	ix
Índice de tablas .....	xxi
Capítulo 1 Introducción.....	1
1.1    Introducción .....	3
1.2    El Problema a resolver.....	3
1.3    Motivación .....	8
1.4    Objetivo general.....	10
1.5    Objetivos particulares .....	10
1.6    Hipótesis.....	11
1.7    Organización del documento .....	12
Capítulo 2 Estado del arte .....	15
2.1    Introducción .....	17
2.2    Metodologías orientadas a píxeles .....	17
2.2.1    Un esquema de clasificación de imágenes jerárquico automático.....	17
2.2.2    Segmentación de imágenes usando segmentación de color basada en las ondeletas rápidas y el crecimiento de regiones direccional.....	21
2.2.3    Recuperación de imágenes basado en semántica: Un enfoque probabilístico .....	24
2.2.4    Etiquetado semántico de imágenes que combina color, textura y palabras clave...	27
2.2.5    Diseño e implementación de un sistema de recuperación semántica de imágenes	30
2.2.6    Aprendizaje de la semántica de primitivas y de escenas de imágenes para su clasificación y recuperación .....	33

2.2.7	Técnicas basadas en la percepción para la clasificación y recuperación semántica de imágenes	36
2.2.8	Segmentación, clasificación y marcas de agua para autenticación semántica de imágenes y video	40
2.2.9	Morfología matemática en espacios de color aplicada al análisis de imágenes cartográficas	43
2.2.10	Segmentación rápida de imágenes basada en análisis de multi-resolución y ondeletas	47
2.3	Metodologías orientadas a objetos	52
2.3.1	Técnica de segmentación basada en borde y región para la extracción de objetos artificiales grandes en una colección de imágenes satelitales de alta resolución	52
2.3.2	Análisis mental-semántico e integración de datos orientados a objetos de información visual. Una primera versión	57
2.3.3	Herramienta comercial eCognition	59
2.4	Enfoque conceptual	62
2.4.1	En busca del concepto perdido	62
2.5	Comentarios y observaciones	65
Capítulo 3 Marco teórico		67
3.1	Introducción	69
3.2	Percepción remota	69
3.2.1	Tipos de imágenes satelitales	71
3.2.2	Visualización de imágenes satelitales	72
3.2.3	Corrección de imágenes satelitales	75
3.3	Procesamiento digital de imágenes	77
3.3.1	Preprocesamiento	78
3.3.2	Preprocesamiento en el dominio espacial	79
3.3.3	Segmentación	86

3.3.4	Agrupamiento ( <i>clustering</i> ) .....	91
3.3.5	Etiquetado de componentes conexas .....	100
3.3.6	Esqueleto de una imagen .....	102
3.4	Ontologías .....	105
3.5	Árboles dinámicos .....	112
Capítulo 4 Metodología propuesta .....		115
4.1	Introducción .....	117
4.2	Dominio del problema y restricciones .....	117
4.3	Compresión semántica.....	121
4.3.1	Jerarquía compacta de los segmentos de una imagen .....	123
4.3.2	Propiedades y atributos de la compresión semántica .....	124
4.3.3	Construcción de la jerarquía de figuras isotrópicas .....	130
4.3.4	Algoritmo de compresión semántica .....	131
4.3.5	Modo iterativo del algoritmo de compresión semántica .....	136
4.4	Descomposición semántica.....	141
4.5	Etapas de análisis.....	144
4.6	Etapas de síntesis.....	144
4.6.1	Extracción de regiones .....	145
4.6.2	Reconocimiento de objetos geográficos.....	147
4.6.3	Especialización de los objetos geográficos.....	150
4.6.4	Representación alterna .....	155
4.7	Ontología para la especialización de los objetos geográficos.....	160
4.7.1	Preguntas de competencia.....	160
4.7.2	Reutilización de ontologías existentes .....	161
4.7.3	Términos importantes para la ontología.....	163
4.7.4	Definición de las clases y jerarquías de clases .....	163

4.7.5	Definición de las propiedades de las clases: <i>slots</i> .....	169
4.8	Descomposición semántica iterativa .....	170
4.9	Comentarios y observaciones .....	171
Capítulo 5 Resultados experimentales.....		173
5.1	Introducción .....	175
5.2	Sistema desarrollado.....	176
5.3	Algoritmo de reconocimiento .....	178
5.4	Ontología de aplicación.....	187
5.5	Casos de estudio para la descomposición semántica .....	190
5.5.1	Caso de estudio 1: Geoimagen de una isla, ejemplo 1 .....	190
5.5.2	Caso de estudio 2: Geoimagen de una isla, ejemplo 2 .....	209
5.5.3	Caso de estudio 3: Geoimagen de los Grandes Lagos.....	219
5.5.4	Caso de estudio 4: Geoimagen de la República Mexicana.....	234
5.6	Comparación con eCognition .....	245
5.7	Enfoque propuesto <i>vs</i> enfoque clásico .....	250
5.7.1	Agrupamiento.....	250
5.7.2	Eliminación de regiones parásitas.....	251
5.7.3	Comentarios de los resultados.....	252
5.7.4	Caso de estudio 1: Geoimagen de una isla, ejemplo 1. ....	252
5.7.5	Caso de estudio 2: Geoimagen de una isla, ejemplo 2. ....	256
5.7.6	Caso de estudio 3: Geoimagen de los Grandes Lagos.....	257
5.7.7	Caso de estudio 4: Geoimagen de la República Mexicana.....	258
5.7.8	Uso de preprocesamiento .....	259
5.8	Generación de geoimágenes invariantes .....	262
5.9	Comentarios y observaciones .....	264
Capítulo 6 Conclusiones .....		267

6.1	Introducción .....	269
6.2	Conclusiones.....	269
6.3	Trabajo futuro .....	272
Referencias.....		275
Anexo A Resultados del enfoque clásico.....		287
A.1	Resultados de la etapa de análisis con enfoque clásico.....	289
A.1.1	Caso de estudio 1: Geoimagen de una isla, ejemplo 1 .....	292
A.1.2	Caso de estudio 2: Geoimagen de una isla, ejemplo 2 .....	298
A.1.3	Caso de estudio 3: Geoimagen de los Grandes Lagos.....	304
A.1.4	Caso de estudio 4: Geoimagen de la República Mexicana.....	310
Anexo B Código fuente.....		317
B.1	Código fuente del algoritmo de compresión semántica .....	319



## Listado de acrónimos y siglas

- A2R2V.** De análogo a raster y de raster a vector (*Analog To Raster To Vector*).
- CDT.** Triangulación de Delaunay restringida (*Constrained Delaunay Triangulation*).
- DLS.** Simetría local discreta (*Discrete Local Symmetry*).
- EDS.** Especificación de codificación de datos ambientales (*Environmental Data Coding Specification*).
- FDP.** Función de densidad de probabilidad.
- FFT.** Transformada rápida de fourier (*Fast Fourier Transform*).
- GIS.** Sistema de Información Geográfica (*Geographic Information System*).
- GMM.** Modelo mixto Gaussiano (*Gaussian Mixture Model*).
- HSL.** Tonalidad, saturación, luminosidad (*Hue, Saturation, Lightness*).
- INEGI.** Instituto Nacional de Estadística, Geografía e Informática.
- ISODATA.** Técnicas de análisis de datos iterativas auto organizadas (*Iterative Self-Organizing Data Analysis Techniques*).
- MRA.** Análisis multi-resolución (*Multi-resolution Analysis*).
- PSLG.** Grafo plano de líneas rectas (*Planar Straight – Line Graph*).
- OCR.** Reconocimiento óptico de caracteres (*Optical Character Recognition*).
- PDI.** Procesamiento digital de imágenes.
- RGB.** Rojo, verde, azul (*Red, Green, Blue*).
- SMV.** Máquina de soporte vectorial (*Support Vector Machine*).
- SLS.** Simetría local suavizada (*Smoothed Local Symmetry*).
- SPOT.** Satélite para la observación de la Tierra (*Satellite Pour l’Observation de la Terre*).
- SVD.** Descomposición de valores singulares (*Singular Value Decomposition*).



## Índice de figuras

Figura 1–1. Ejemplo de geoimágenes (a) Versión raster (b) Tipo vectorial. ....	4
Figura 1–2. Ejemplo de un mapa.....	5
Figura 1–3. Conversión de una geoimagen raster a formato vectorial; (a) Geoimagen original; (b) Versión digital vectorial. Fuente: GoogleMaps.....	6
Figura 1–4. Ejemplos de dos geoimágenes (a) República Mexicana (b) Bahía de Cancún, Quintana Roo. Fuente: Microsoft Maps.....	8
Figura 2–1. Jerarquía obtenida con el conjunto de imágenes de entrenamiento.....	20
Figura 2–2. Resultados al aplicar el algoritmo propuesto.....	23
Figura 2–3. Transformación del espacio de color (a) Imagen en el espacio RGB (b) Componente Y del modelo YUV (c) Componente U (d) Componente V.....	23
Figura 2–4. Comparación entre wavelet y cambio de muestreo (a) Imagen original; (b) Componente LL con nivel k=3; (c) Resultado al cambiar el muestreo de la imagen original.....	24
Figura 2–5. Etiquetado por regiones (a) Imagen original (b) Regiones de objetos artificiales. ....	25
Figura 2–6. Resultados del algoritmo, (a) Imágenes a procesar (b) Palabras clave asociadas. ....	29
Figura 2–7. Ejemplo de una red semántica.....	30
Figura 2–8. Ejemplo de relación <i>es un</i> . ....	31
Figura 2–9. Arquitectura propuesta.....	32
Figura 2–10. Metodología para el cálculo de la semántica de una imagen.....	34
Figura 2–11. Técnica de Chen (a) Imagen original (b) Colores dominantes adaptativos; (c) Clases de textura; (d) Segmentación final.....	37
Figura 2–12. Cuantificación de características para cada segmento (a) Imagen segmentada; (b) Extracción del segmento de interés; (c) Composición de colores dominantes; (d) Orientación de la textura.....	38
Figura 2–13. Imágenes con semántica diferente (a) Imagen original (b) Imagen con fondo alterado. ....	40
Figura 2–14. Esquema general de la aplicación propuesta.....	41

Figura 2–15. Ejemplos de diferentes tipos de gradiente de color (a) luminancia (b) tonalidad (c) saturación ponderada (d) supremo (e) cromático (f) perceptual. ....	44
Figura 2–16. Ejemplos de dos versiones del operador <i>top-hat</i> aplicados a las imágenes de la Figura 2–15(a) y (b) respectivamente; (a) Aplicando el operador <i>top-hat</i> negro acromático; (b) Resultado al aplicar el operador <i>top-hat</i> cromático. ....	45
Figura 2–17. Arquitectura del algoritmo desarrollado.....	45
Figura 2–18. Ejemplos de posprocesamiento (a)Imagen de color original; (b) Capa obtenida de texto/gráfico; (c) Adelgazamiento morfológico; (d) Capa de texto/gráfico combinada con la imagen de color original.....	47
Figura 2–19. Resultado de la metodología propuesta. (a)(d)(g) Imágenes originales; (b)(e)(h) Resultado aplicando el método de Otsu con 3 umbralados en las imágenes originales; (c)(f)(i) Resultado aplicando el algoritmo propuesto y expansión directa con una transformada de nivel 3. ....	51
Figura 2–20. Resultados de diferentes algoritmos de segmentación (a) Imagen original; (b) Segmentación manual; (c) ISO-DATA; (d) Algoritmo <i>watershed</i> ; (e) Crecimiento de regiones.....	54
Figura 2–21. Etapa de extracción de los modelos de bordes. ....	55
Figura 2–22. Resultados y comparaciones (a) Imagen original; (b) Resultado aplicando el algoritmo propuesto; (c) Resultado aplicando el algoritmo de corrimiento de media a la imagen de la Figura 2.20a; (d) Resultado usando el sistema eCognition.....	56
Figura 2–23. Resultados de la etapa de análisis con diferentes niveles semánticos; (a) Imagen Axe; (b) Imagen Plait. ....	58
Figura 2–24. Etapa de Análisis en el sistema A2R2V; (a) Imagen original; (b) Inspección de la imagen.....	59
Figura 2–25. Pantalla principal de eCognition. ....	60
Figura 2–26. Definición de un nuevo algoritmo en eCognition. ....	61
Figura 2–27. Elementos de la metodología de LINNEO <sup>+</sup> .....	63
Figura 3–1. Línea del tiempo de los satélites Landsat. Fuente: NASA (NASA, 2007). ....	71
Figura 3–2. Ejemplo de la técnica de pseudocolor; (a) Imagen original; (b) Imagen en pseudocolor. ....	74
Figura 3–3. Imagen en color verdadero. ....	74
Figura 3–4. Comparación entre geoimágenes (a) Satélite TIROS-1 (b) Satélite IKONOS. ....	75
Figura 3–5. Fenómenos que afectan la radiación electromagnética (a) Dispersión (b) Absorción. .	76

Figura 3–6. Manipulación del histograma.....	80
Figura 3–7. (a) Imagen en niveles de gris; (b) Histograma correspondiente. ....	87
Figura 3–8. Uso de la Transformada de Hough; (a) Imagen original; (b) Detección de los vértices formados por las líneas. ....	90
Figura 3–9. Agrupamiento de patrones de puntos en dos dimensiones. ....	92
Figura 3–10. Algoritmo de etiquetado de componentes conexas.....	101
Figura 3–11. Una simetría local en el dominio del espacio continuo. ....	102
Figura 3–12. Línea recta discretizada donde no es posible encontrar para un punto cualquiera su simetría local. ....	103
Figura 3–13. Dos elementos de contorno CE-A y CE-B de una simetría local discreta. ....	104
Figura 3–14. Un estado particular de un conjunto de bloques sobre una mesa. ....	107
Figura 3–15. Tipos de ontologías, acorde a su nivel de generalización. ....	108
Figura 3–16. Ejemplo de un árbol cuaternario.....	112
Figura 3–17. Ejemplo de árbol dinámico.....	113
Figura 4–1. Ejemplo de una geoimagen.....	118
Figura 4–2. Ejemplo de una geoimagen donde las características de bajo nivel de los objetos geográficos están alteradas por un fenómeno natural.....	119
Figura 4–3. Ejemplo de una geoimagen (b) Resultado de la clasificación usando el algoritmo k-medias con la herramienta comercial PC-Geomática.....	122
Figura 4–4. Ortogonalización y homogeneización de la longitud de los vectores $u, v$ aplicando la transformación para $n=4$ ; (a) Datos originales; (b) Imagen isotrópica.....	130
Figura 4–5. Esquema general del algoritmo de compresión semántica. ....	132
Figura 4–6. Árbol dinámico para iniciar el proceso de compresión semántica. ....	133
Figura 4–7. Ejemplo de fusión de segmentos en el árbol dinámico. ....	134
Figura 4–8. Cuantificación de una característica o parámetro. ....	134
Figura 4–9. Modo iterativo del algoritmo de compresión semántica.....	136
Figura 4–10. Representación de la primera iteración del algoritmo de compresión semántica...	137
Figura 4–11. Ejecución del algoritmo de compresión semántica con tres iteraciones tomando como base la geoimagen de intensidad.....	138
Figura 4–12. Uso de las geoimágenes de intensidad promedio (ABS) y de distancia ( $d$ ) en el algoritmo de compresión semántica.....	139
Figura 4–13. Algoritmo general de descomposición semántica. ....	142

Figura 4–14. Diagrama general de la etapa de síntesis.....	145
Figura 4–15. Ejemplo de objetos geográficos. ....	147
Figura 4–16. Ejemplo de capas temáticas.....	149
Figura 4–17. Ejemplificación de información geográfica incompleta. Fuente: Galería gratuita de LANDSAT (LANDAST Satellite Image Gallery, 2007). ....	154
Figura 4–18. Relación entre dos objetos geográficos. ....	154
Figura 4–19. Configuraciones diferentes donde se cumple la relación <i>rodeado por</i> . ....	155
Figura 4–20. Máscaras para la detección de píxeles redundantes. ....	158
Figura 4–21. Resultados de la eliminación de píxeles redundantes (a) Segmento de contorno original; (b) Resultado después de eliminar los píxeles redundantes.....	159
Figura 4–22. Ontología para la especialización de objetos geográficos. ....	165
Figura 4–23. Adición de los objetos geográficos a una temática.....	165
Figura 4–24. Ontología con diferentes clases de especialización, caso 1. ....	166
Figura 4–25. Ontología con diferentes clases de especialización, caso 2. ....	167
Figura 4–26. Ejemplo de una ontología con la relación <i>rodeado</i> .....	168
Figura 4–27. Geoimagen de los Acantilados de Mohar, Irlanda. ....	169
Figura 4–28. Geoimagen de la República Mexicana. ....	170
Figura 5–1. Formulario principal de la aplicación de descomposición semántica.....	176
Figura 5–2. Conjunto de imágenes de salida del algoritmo de compresión semántica. ....	177
Figura 5–3. Regiones para la determinación de vectores prototipo para la temática de cuerpo de agua.....	179
Figura 5–4. Regiones para la determinación de vectores prototipo para la temática de suelo, variante 1. ....	180
Figura 5–5. Regiones para la determinación de vectores prototipo para la temática de suelo, variante 2. ....	181
Figura 5–6. Código en MATLAB para la determinación de la media y la mediana. ....	182
Figura 5–7. Ontología de aplicación para la especialización de objetos geográficos. ....	189
Figura 5–8. Geoimagen original que representa una isla. ....	191
Figura 5–9. Histograma de las componentes roja, verde y azul de la geoimagen de la isla 1.....	191
Figura 5–10. Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S;	

(d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.....	193
Figura 5–11. Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad promedio de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	194
Figura 5–12. Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen del parámetro S2f de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	196
Figura 5–13. Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de la característica de largo de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	198
Figura 5–14. Conjunto de geoimágenes que se obtienen después de la quinta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen del parámetro Ss de la iteración 4; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	199
Figura 5–15. Simplificación de la geoimagen del caso de estudio 1; (a) Geoimagen original; (b) Geoimagen de intensidad promedio (iteración 1); (c) Geoimagen del parámetro S2f (iteración 2); (d) Geoimagen de la característica L (iteración 3); (e) Geoimagen del parámetro Ss (iteración 4); (f) Geoimagen del parámetro Ss (iteración 5). ....	201
Figura 5–16. Simplificación de la geoimagen del caso de estudio 1 con otra cadena de compresión semántica; (a) Geoimagen original; (b) Geoimagen de intensidad promedio (iteración 1); (c) Geoimagen de la característica de área (iteración 2); (d) Geoimagen del parámetro Ss (iteración 3). ....	202
Figura 5–17. Extracción de las regiones de los objetos geográficos (a) Geoimagen de la clase 1; (b) Geoimagen de la clase 2.....	203

Figura 5–18. Resultado de la compresión semántica sobre la región extraída que representa un cuerpo de agua.....	207
Figura 5–19. Resultado de la compresión semántica sobre la región extraída que representa suelo. .....	208
Figura 5–20. Geoimagen de otro accidente geográfico tipo isla. ....	209
Figura 5–21. Histogramas de las componentes roja, verde y azul del accidente geográfico tipo isla. .....	209
Figura 5–22. Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro S <sub>s</sub> ; (e) Característica S <sub>z</sub> ; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.....	210
Figura 5–23. Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro S <sub>s</sub> ; (e) Característica S <sub>z</sub> ; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	212
Figura 5–24. Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de área de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro S <sub>s</sub> ; (e) Característica S <sub>z</sub> ; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. .....	213
Figura 5–25. Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de área de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro S <sub>s</sub> ; (e) Característica S <sub>z</sub> ; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. .....	215
Figura 5–26. Geoimagen de los pixeles que fueron agrupados con la primera intensidad de color en la geoimagen del parámetro S <sub>s</sub> .....	217
Figura 5–27. Geoimagen de los pixeles que fueron agrupados con la segunda intensidad de color en la geoimagen del parámetro S <sub>s</sub> .....	217
Figura 5–28. Geoimagen de los Grandes Lagos. ....	219

Figura 5–29. Histogramas de las componentes de color roja, verde y azul de la geoimagen de la República Mexicana. ....	219
Figura 5–30. Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.....	220
Figura 5–31. Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	222
Figura 5–32. Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	223
Figura 5–33. Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	225
Figura 5–34. Conjunto de geoimágenes que se obtienen después de la quinta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 4; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	226
Figura 5–35. Geoimagen simplificada con la característica de área, obtenida en la quinta iteración. ....	228
Figura 5–36. Extracción de regiones de la intensidad de color 1.....	229
Figura 5–37. Extracción de regiones de la intensidad de color 2.....	229
Figura 5–38. Objetos geográficos especializados como tipo <i>INLAND</i> .....	231
Figura 5–39. Objetos geográficos especializados como tipo islote.....	232

Figura 5–40. Objetos geográficos <i>INLAND</i> especializados como lagos.....	232
Figura 5–41. Objetos geográficos <i>INLAND</i> especializados como lagunas.....	233
Figura 5–42. Geoimagen de la República Mexicana. ....	234
Figura 5–43. Histogramas de las componentes de color roja, verde y azul de la geoimagen de la República Mexicana. ....	234
Figura 5–44. Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.....	235
Figura 5–45. Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	237
Figura 5–46. Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	239
Figura 5–47. Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de distancia de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f. ....	240
Figura 5–48. Geoimagen simplificada con la característica de largo.....	242
Figura 5–49. Resultado de la extracción de regiones de la geoimagen de la característica de largo. ....	243
Figura 5–50. Parámetros de para el algoritmo de segmentación multi-resolución. ....	246
Figura 5–51. Resultado de la segmentación multiresolución. ....	246
Figura 5–52. Criterios para la segmentación basada en conocimiento. ....	247
Figura 5–53. Geoimagen final segmentada. ....	247
Figura 5–54. Clasificación de objetos con radio mayor o igual que 0.3 en la componente roja. ...	248

Figura 5–55. Clasificación de objetos con radio mayor o igual que 0.4 en la componente roja. ...	249
Figura 5–56. Clasificación de objetos con radio mayor o igual que 0.25 en la componente roja. .	249
Figura 5–57. Algoritmo de clasificación. ....	251
Figura 5–58. Análisis tradicional basado en procesamiento del color.....	252
Figura 5–59. Resultados de cada iteración para la simplificación de la geoimagen de intensidad promedio con un umbral de similitud igual a 30; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3; (d) Iteración 4; (e) Iteración 5; (f) Iteración 6.....	253
Figura 5–60. Resultados del agrupamiento para la simplificación de la geoimagen de intensidad promedio con un umbral de similitud igual a 30; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3; (d) Iteración 4; (e) Iteración 5; (f) Iteración 6.....	254
Figura 5–61. Resultados de la simplificación de la geoimagen de intensidad promedio; (a) Geoimagen de salida del algoritmo de compresión semántica; (b) Resultado del algoritmo de agrupamiento.....	255
Figura 5–62. Eliminación de regiones parásitas de la geoimagen agrupada; (a) Imagen original; (b) Resultado del algoritmo de poda.....	255
Figura 5–63. Resultados de la compresión semántica de la geoimagen de intensidad promedio, usando un umbral de similitud igual a 15; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.....	256
Figura 5–64. Resultados del agrupamiento c-medias difuso de las geoimágenes obtenidas en la compresión semántica de la Figura 5–63; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.....	257
Figura 5–65. Resultados de la compresión semántica de la geoimagen de intensidad, usando un umbral de similitud igual a 10; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3. ....	257
Figura 5–66. Resultados del agrupamiento $k$ -medias de las geoimágenes obtenidas en la compresión semántica de la Figura 5–65; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.....	258
Figura 5–67. Resultados de la compresión semántica de la geoimagen de intensidad, usando un umbral de similitud igual a 20; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3. ....	258
Figura 5–68. Resultados del agrupamiento ISODATA de las geoimágenes obtenidas en la compresión semántica de la Figura 5–67; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.....	259
Figura 5–69. Descomposición en componentes de color de la geoimagen del caso de estudio 1.	260
Figura 5–70. Resultados del algoritmo de compresión semántica; (a) Geoimagen de intensidad; (b) Geoimagen de intensidad promedio. ....	261
Figura 5–71. Resultados de la clasificación usando $k$ -medias; (a) Geoimagen de intensidad clasificada; (b) Geoimagen de intensidad promedio clasificada.....	261

Figura 5–72. Diferencia entre las dos geoimágenes clasificadas. ....	262
Figura 5–73. Número de segmentos de cada iteración de una cadena de compresión semántica. ....	263
Figura A–1. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.....	292
Figura A–2. Resultados del algoritmo de agrupamiento $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40. ....	293
Figura A–3. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.....	295
Figura A–4. Resultados del algoritmo de agrupamiento $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40. ....	296
Figura A–5. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.....	298
Figura A–6. Resultados del algoritmo de agrupamiento $c$ -medias difuso aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35. ....	299
Figura A–7. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.....	301
Figura A–8. Resultados del algoritmo de agrupamiento $c$ -medias difuso aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35. ....	302
Figura A–9. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.....	304
Figura A–10. Resultados del algoritmo de agrupamiento $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30. ....	305

Figura A–11. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.....	307
Figura A–12. Resultados del algoritmo de agrupamiento $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30. ....	308
Figura A–13. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 20; (e) 22.5; (f) 25. ....	310
Figura A–14. Resultados del algoritmo de agrupamiento ISODATA aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 30; (e) 32.5; (f) 35. ....	311
Figura A–15. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 20; (e) 22.5; (f) 25. ....	313
Figura A–16. Resultados del algoritmo de agrupamiento ISODATA aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 30; (e) 32.5; (f) 35. ....	314



## Índice de tablas

Tabla 2–1. Definición de los tamaños de ventana para cada nivel de muestreo. ....	26
Tabla 2–2. Etiquetas para segmentos. ....	39
Tabla 3–1. Funciones de transferencia para la modificación del histograma.....	81
Tabla 3–2. Ecuaciones de filtros de ordenamiento.....	86
Tabla 3–3. Ecuaciones de filtros de promediado .....	86
Tabla 3–4. Máscaras más comunes para la segmentación basada en bordes.....	88
Tabla 4–1. Definiciones de algunos cuerpos de agua. ....	152
Tabla 4–2. Resumen de la etapa de síntesis. ....	159
Tabla 5–1. Herramientas usadas para los algoritmos de la metodología de descomposición semántica. ....	176
Tabla 5–2. Cuantificación de la media, mediana y desviación estándar para las regiones muestra de la temática de cuerpo de agua.....	184
Tabla 5–3. Cuantificación de la media, mediana y desviación estándar para las regiones muestra de la temática de suelo, variante 1. ....	185
Tabla 5–4. Cuantificación de la media, mediana y desviación estándar para las regiones muestra de la temática de suelo, variante 2. ....	186
Tabla 5–5. Vectores prototipo para las temáticas de cuerpo de agua y suelo. ....	187
Tabla 5–6. Definiciones de los objetos geográficos para la ontología de aplicación .....	187
Tabla 5–7. Parámetros del algoritmo de compresión semántica para el caso de estudio 1. ....	192
Tabla 5–8. Resultados para cada una de las regiones del caso de estudio 1.....	204
Tabla 5–9. Distancias entre los vectores a reconocer y los vectores prototipos de las temáticas. ....	204
Tabla 5–10. Especialización de los objetos geográficos del caso de estudio 1. ....	205
Tabla 5–11. Parámetros para el algoritmo de compresión semántica para el caso de estudio 2. .	210
Tabla 5–12. Resultados para cada una de las regiones del caso de estudio 3.....	218
Tabla 5–13. Distancias entre los vectores a reconocer y los vectores prototipos de las temáticas. ....	218

Tabla 5–14. Parámetros del algoritmo de compresión semántica para el caso de estudio 3. .... 220

Tabla 5–15. Resultados para cada una de las geoimágenes extraídas del caso de estudio 2. .... 230

Tabla 5–16. Distancias entre los vectores a reconocer y los vectores prototipos de las temáticas.  
..... 230

Tabla 5–17. Parámetros del algoritmo de compresión semántica para el caso de estudio 4. .... 235

Tabla 5–18. Resultados para cada una de las regiones del caso de estudio 4..... 244

Tabla 5–19. Distancias entre los vectores a reconocer y los vectores prototipos de las temáticas.  
..... 245

Tabla A–1. Cadenas de compresión semántica usadas en el análisis con enfoque clásico. .... 289

## Resumen

En este trabajo se presenta una metodología orientada a objetos para extraer la semántica de una geoimagen descrita por medio de un conjunto de etiquetas en lenguaje natural así como un mecanismo para generar una representación alterna de dichos objetos geográficos. Esta metodología está compuesta por dos grandes etapas: análisis y síntesis.

La etapa de análisis está integrada por el algoritmo de compresión semántica. Mediante este algoritmo, es posible hacer una detección de los principales objetos geográficos que componen la geoimagen, aunque existan variaciones significativas en las características de bajo nivel de la misma, o incluso, ruido añadido durante su adquisición. La simplificación de una geoimagen consiste en fusionar los segmentos adyacentes que componen la imagen cuando existe una similitud en una determinada característica o parámetro de compresión semántica. Este conjunto de características y parámetros permite obtener las regiones que describen los objetos geográficos con base en la cuantificación de color, geometría y topología de los objetos geoespaciales. El resultado final de esta etapa es una geoimagen con intensidades aproximadamente uniformes para cada objeto geográfico.

La etapa de síntesis está enfocada a realizar una extracción de los objetos geográficos que se deben reconocer y un etiquetado en dos niveles (general y especializado), lo que equivale a considerar la información tanto local como global de una geoimagen para su correcta interpretación. Al realizar un etiquetado general se pretende asociar una etiqueta que corresponde a la temática a la que pertenecen las regiones de los objetos geoespaciales de una misma clase, basándose en las características de intensidad RGB. Esto permite hacer uso de un algoritmo tradicional de reconocimiento debido a que no es necesaria una identificación detallada de los mismos. Para hacer una especialización de cada objeto geográfico, se propone un algoritmo de especialización que toma en cuenta las relaciones geométricas y topológicas entre los objetos, representadas en una ontología de aplicación del dominio geográfico. El conjunto de etiquetas generadas representa la semántica de la geoimagen. Finalmente, se define el mecanismo que permite generar una representación alterna de los objetos geoespaciales.



## Abstract

This thesis presents an object oriented methodology for the the extraction of the semantic of a geoimage which is described by a set of natural language labels. The methodology also provides a mechanism to generate an alternative representation of such objects. This methodology is comprised of two main stages: analysis and synthesis.

The analysis stage is comprised of the semantic compression algorithm. Using this algorithm it is possible to detect the main geographic components of a geoimage. The algorithm is capable to perform well, under the presence of important variations in the low level characteristics of the image or even acquisition noise. The image simplification process consists of fusing adjacent segments of an image when there is a certain amount of similarity in a given set of characteristics and parameters. This set of characteristics and parameters allow us to identify the regions that describe the geographic objects. This latter process is based on the quantification of color, geometry and topology of the geospatial objects. The result of this stage is a geoimage composed of geomages with intensities that are approximately uniform

The synthesis stage has the objective of extracting the main geographic objects that have to be identified, and a labeling in two levels (general and specialized), which is equivalent to consider both local and global information of a geoimage. The aim of the general labeling process is to associate a label of the adequate thematic to each region, taking into account the RGB characteristics of the image. This labeling and the fact that detailed identification is not necessary, allow us to make use of a traditional recognition algorithm. In order to specialize each geographic object we propose a specialization algorithm that takes into account geometric and topologic relations among objects, and represents them in an ontology of the geographic application domain. The obtained set of labels represents the geoimage semantic. Lastly, we define a mechanism to generate an alternative representation of the geospatial objects.



# Objetivos

## Objetivo general

Definir una metodología general de descomposición semántica integrada por una etapa de análisis que permita extraer los objetos geográficos de una geoimagen dada basándose en un algoritmo de compresión semántica, así como de una etapa de síntesis que reconozca y asigne etiquetas en lenguaje natural a los objetos geográficos que integran la geoimagen, empleando una ontología de aplicación.

## Objetivos particulares

1. Definir el concepto de geoimagen como una superclase de imágenes digitales que describe objetos geográficos.
2. Expresar el concepto de semántica como un conjunto de etiquetas en lenguaje natural de los objetos geográficos que contiene una geoimagen.
3. Definir un conjunto de características integrado por atributos y propiedades de los objetos geográficos para el algoritmo de compresión semántica.
4. Conceptualizar e implantar un algoritmo de compresión semántica que permita simplificar las características de bajo nivel de los objetos geográficos y describirlos mediante intensidades de color uniformes.
5. Formular una etapa de síntesis para el reconocimiento de los objetos geográficos, la cual proporcionará un conjunto de etiquetas de los objetos identificados.

6. Definir una ontología de aplicación que permita especializar los objetos geográficos, integrada por clases y relaciones que permitan especializar el tipo de un objeto geográfico dado.

# Capítulo 1

## Introducción

### ***Resumen***

En este capítulo se describen detalladamente el problema a resolver, el objetivo general, los objetivos particulares y las hipótesis del presente trabajo de investigación, así como la organización del documento.



## 1.1 Introducción

El uso de los Sistemas de Información Geográfica (*Geographic Information System - GIS*) ha crecido sustancialmente en los últimos años. Lo anterior se deriva principalmente de la mayor accesibilidad a equipos de cómputo de alto desempeño, así como a imágenes satelitales o de fuentes remotas a las que anteriormente no se tenía fácil acceso, con las que es posible obtener información vectorial para realizar tareas de planificación, construcción, simulación, evaluación de daños, entre otros. Sin embargo, con la creciente demanda y aplicación de este tipo de imágenes, también han surgido diversos problemas a los que los científicos están tratando de dar solución, como son el análisis automático, reconocimiento, clasificación, descomposición de objetos, entre otros. En este capítulo se describe el problema a resolver, el objetivo general, los objetivos particulares y las hipótesis de la investigación.

## 1.2 El Problema a resolver

En la actualidad, los datos espaciales (datos asociados con la representación de un territorio en la Tierra) son de gran utilidad para diversos grupos, ya que por medio de ellos es posible obtener información del contenido de una determinada área geográfica como es: vías de comunicación, hidrología, vegetación, áreas urbanas, por citar algunos. Con dicha información es posible planificar y diseñar adecuadamente infraestructura de transporte y comunicación, viviendas, complejos industriales, etc., con la seguridad de que no se afectará alguna zona geográfica, como puede ser un asentamiento urbano.

Desafortunadamente, muchas organizaciones en México obtienen manualmente la información espacial empleando mapas cartográficos en formato analógico, lo cual ocasiona entre otras cosas:

a. Demora en los resultados. La mayoría de los mapas contienen demasiada información (curvas de nivel, vías de transporte, hidrología, poblados, etc.) por lo cual, el encontrar los datos requeridos se vuelve una tarea laboriosa y posiblemente la información se requiera rápidamente.

b. Datos no actualizados. El mapa que se esté consultando puede tener 3, 5, 10 o más años de antigüedad, con lo cual, los datos obtenidos pueden ser erróneos o simplemente no encontrarse.

c. Error en los cálculos. Se puede requerir de medir la distancia que hay entre dos objetos de interés, dado que los mapas generalmente están a escala, la medición manual seguramente dará un resultado demasiado impreciso.

Estos problemas pueden ser minimizados o eliminados, haciendo uso de imágenes cartográficas en formato digital, ver Figura 1–1. En la Figura 1–1(a) se muestra una imagen raster, formada por un conjunto de píxeles en diferentes intensidades rojo, verde y azul, mientras que en el segundo caso, Figura 1–1(b), se tiene una imagen vectorial formada por un conjunto de datos o vectores compuestos por tres primitivas básicas: puntos, líneas y polígonos.

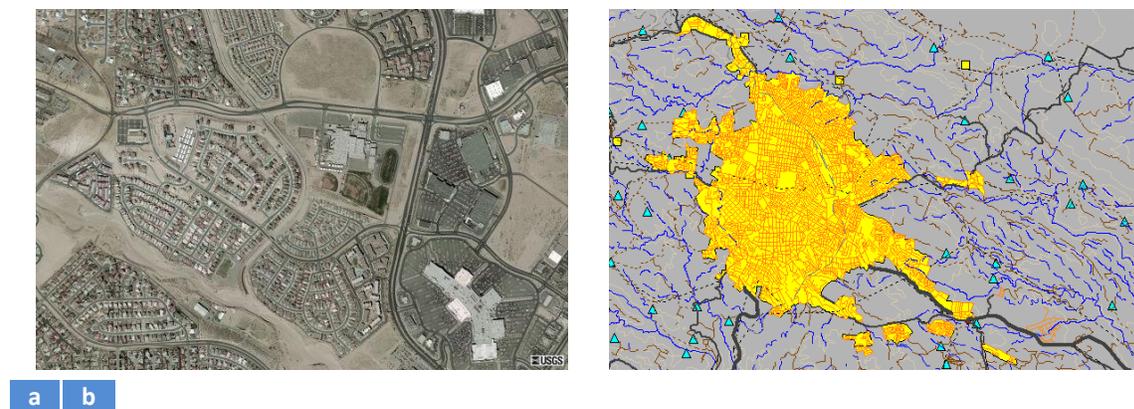


Figura 1–1. Ejemplo de geoimágenes (a) Versión raster (b) Tipo vectorial.

Algunas personas podrían usar el término mapa para referirse a la Figura 1–1(b). Sin embargo, es importante dejar claro cuándo una imagen puede ser llamada mapa. Un mapa es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que puede ser también esférica como ocurre en los globos terráneos. El que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancias, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto (Wikipedia, 2007).

Otra definición de mapa es: “Un mapa es una representación reducida, generalizada y matemáticamente determinada, de la superficie terrestre sobre un plano, en las cuales se

interpreta la distribución, el estado y los vínculos de los distintos fenómenos naturales y sociales, seleccionados y caracterizados de acuerdo con la asignación concreta del mapa” (Salitchev, 1979). Dentro de nuestro contexto, entenderemos que **representación reducida** se refiere a que el mapa tiene una escala; el término **generalizada** se refiere a que el mapa contiene lo más relevante con base al propósito del mapa, y **matemáticamente determinada** expresa que el mapa utiliza un sistema de coordenadas y un tipo de proyección para ubicar cada elemento del mismo. Cuando se cumplan las condiciones anteriores, podremos calificar a la imagen como un mapa, ver Figura 1–2.



Figura 1–2. Ejemplo de un mapa.

En ocasiones la información raster permite obtener los datos requeridos para una aplicación en particular. Podemos por ejemplo, conocer la trayectoria de una vía de comunicación, pero en otras situaciones, es necesario poseer información única descriptiva acerca de los objetos geográficos que relacionen su semántica; por ejemplo, la longitud de un río, la profundidad de una cuenca, la cantidad de habitantes de una población, aplicar modelos matemáticos de simulación para determinar la probabilidad de que un río se desborde, que un bosque quede desforestado a corto o mediano plazo etc. Esto implica contar con la semántica espacial de los objetos geográficos, es decir, reconocer claramente la información espacial que describe un concepto o temática<sup>1</sup>.

El término semántica, se refiere a los aspectos del significado o interpretación de un determinado código simbólico, lenguaje o representación formal. En principio, cualquier medio de expresión

<sup>1</sup> Esto es conocido como análisis espacial sobre datos.

(código, lenguas,...) admite una correspondencia entre expresiones de símbolos o palabras y situaciones o conjuntos de cosas encontrables o inferibles en el mundo físico o abstracto que puede ser descrito por dicho medio de expresión (Wikipedia, 2007). Otra definición es, “la semántica se refiere al estudio de las relaciones entre símbolos y el significado de dichas relaciones” (Farshad & Sabine, 2002).

Si bien es cierto que al contar con la información en formato vectorial su manipulación es más sencilla, la mayoría de las veces se parte del hecho de que esta información ya fue primeramente digitalizada y clasificada en las temáticas que corresponden, todo a partir de una imagen *raster* o una carta impresa. Para ello, se puede emplear un proceso manual, automático o guiado por un experto. Sin embargo, un alto porcentaje de estas técnicas tienden a generar errores en mayor o menor grado, como puede ser, una mala digitalización, clasificación errónea, etc. Por otra parte, el convertir u obtener datos en formato vectorial a partir de un conjunto de datos raster es un proceso demasiado complejo. El objetivo sería que, a partir de una geoimagen, se pueda obtener una versión digital vectorial compuesta, como ya fue citado, por puntos, líneas y polígonos (ver Figura 1–3).



a b

Figura 1–3. Conversión de una geoimagen raster a formato vectorial; (a) Geoimagen original; (b) Versión digital vectorial. Fuente: GoogleMaps.

En consecuencia, **los problemas actuales dentro del área de la geocomputación, están enfocados a la definición generalizada de algoritmos semánticos para el análisis y procesamiento de imágenes de objetos geográficos.**

Comúnmente, dentro de la percepción remota se hace uso de técnicas y algoritmos del procesamiento digital de imágenes, por citar algunos ejemplos tenemos: los filtros en el dominio

espacial tales como los de promediado, de ordenamiento, entre otros; de igual manera, el uso de la FFT (*Fast Fourier Transform*) para aplicar filtros en el dominio de la frecuencia, en la solución de problemas de umbralado, segmentación como son los algoritmos ISO-Data,  $k$ -medias, crecimiento de regiones, de reconocimiento, de morfología matemática, etc.

Aunque los trabajos de investigación dentro de esta área están enfocados a definir metodologías que permitan lograr una conversión de datos en formato raster a vector, el contar con una versión simplificada de una geoimagen original no es el único punto de relevancia, ya que puede ser de interés el definir un algoritmo que permita determinar el significado de la geoimagen. *Considerando la imagen satelital de la Figura 1–4a, puede surgir la siguiente pregunta ¿Qué representa la geoimagen?* Probablemente, la respuesta que darían algunas personas que contestaran a esta interrogante sería *República Mexicana* o *México*. Sin embargo, al observar la Figura 1–4b, el problema ya no es trivial debido a que las personas pueden responder diferente y sobre todo, la cantidad de información puede no ser suficiente para hacer un reconocimiento; tal vez la respuesta solamente se limitaría a decir bahía. Esto se debe a que cada persona tiene una forma diferente de reconocer e interpretar el contenido de una imagen; adicionalmente, el conocimiento *a priori* juega un papel fundamental, ya que no es posible reconocer un objeto si no se tiene conocimiento previo de él.

Algo importante de resaltar es que cuando se observa una imagen, cada individuo realiza un proceso de selección que le permite extraer el contenido importante o relevante de la imagen (su semántica) y con ello determinar su significado. *Precisamente, dentro de nuestro trabajo de investigación, lo que nos interesa es determinar el contenido semántico de una geoimagen.*

La mayoría de los métodos actuales de procesamiento de imágenes, tales como los *wavelets* (ondeletas) y la descomposición fractal o el análisis de textura, pierden los componentes vitales de la imagen, es decir, la estructura semántica. La semántica es el estudio del significado de los *conceptos*. En la presente propuesta se pretende definir un conjunto de algoritmos basados en análisis y síntesis, que permitan obtener los elementos semánticos de una geoimagen para hacer una descomposición, clasificación y recuperación de los objetos presentes en la misma.



Figura 1–4. Ejemplos de dos geoimágenes (a) República Mexicana (b) Bahía de Cancún, Quintana Roo. Fuente: Microsoft Maps.

A grandes rasgos, se pretende desarrollar una metodología que permita, dada una geoimagen de entrada (imagen original), hacer una descomposición de la misma para obtener un conjunto de temáticas, seguido de una especialización del tipo o nombre de los objetos que geográficos que componen cada temática, para describir la semántica de la imagen. Por ejemplo, para un mapa de la República Mexicana, se podría generar una capa con la información hidrológica, en otra capa, información de las áreas verdes, por citar algunos (Guzmán et al., 2006b).

### 1.3 Motivación

La motivación de este trabajo se fundamenta en varias razones que explicaremos a continuación. Primeramente, como ya fue citado, el uso de la información geoespacial tiene un amplio espectro de aplicaciones, ya que con ella se puede hacer planeación de construcciones, simulaciones de fenómenos naturales como son incendios, inundaciones, deslizamientos, erupciones volcánicas, etc. Desafortunadamente, hacer una interpretación de estos datos resulta complejo y las necesidades de las personas se concentran en utilizar un subconjunto de toda la información que se puede obtener en una geoimagen. Es por ello, que el contar con la información aislada y clasificada en capas temáticas brinda mayor utilidad que el tener un conjunto de datos *crudos*.

Adicionalmente, se puede hacer una representación alterna utilizando primitivas vectoriales como pueden ser puntos, líneas y polígonos, ya que una versión vectorial de un conjunto de datos geoespaciales ocupa una menor cantidad de espacio que su correspondiente versión raster, esto

ahorraría espacio porque se realizaría una compresión de la información geoespacial (Guzmán et al., 2006a), (Guzmán & Quintero, 2006).

El objetivo que se persigue con esta línea de investigación, a mediano o largo plazo, sería contar con un sistema autónomo que pueda realizar tareas que actualmente son llevadas a cabo por los humanos, con la ventaja de que el sistema pueda procesar automáticamente un conjunto cualquiera de imágenes satelitales, sin sufrir ciertos padecimientos como son el error asociado a fatiga, cansancio, estrés, etc. Esto brindaría un alto grado de confiabilidad a los resultados que se obtuvieran y mayor precisión, ya que la interpretación de la geoinformación se haría con base en métodos cuantitativos y no a juicios personales.

Sin embargo, para llegar a este punto, se requiere primeramente establecer las bases que permitan solucionar problemas básicos, entre ellos, el poder aislar la información total que se pueda obtener de una geoimagen. Aunque este problema ha sido atacado por diversos investigadores, no se ha llegado a una solución que pueda ser aplicada en la mayoría de los casos.

Con este trabajo se pretende crear una metodología general y confiable. General porque pretendemos que su uso no quede únicamente demostrado con imágenes muy simples o con condiciones controladas, como pueden ser la iluminación o la textura. Confiable porque los resultados que se obtengan serían tan verdaderos como los que se obtendrían haciendo el mismo procedimiento de forma manual.

El ser humano se caracteriza por observar y procesar cotidianamente una gran cantidad de información visual. Este proceso inicia con la adquisición de una imagen o secuencia de imágenes de una escena en particular mediante el sistema visual. Posteriormente esta información es enviada al cerebro, donde se lleva a cabo un análisis complejo que permite segmentar la información original, y finalmente asociar nombres o etiquetas a los objetos en cuestión, con lo que una persona puede concluir que está observando un edificio, una persona, un vehículo, etc. Uno de los principales objetivos de la comunidad científica, es el tratar de imitar este proceso mediante medios artificiales, por ejemplo, mediante las computadoras.

Hacemos énfasis en el término imitar, por el hecho de que para poder decir que un sistema artificial “observa” como el ser humano, primeramente sería necesario describir concretamente y comprobar, los procesos que realiza el cerebro humano, lo cual, no ha sido posible de deducir. Pero lo que sí es posible afirmar, es que los seres humanos asocian etiquetas a la información visual que procesan.

Este trabajo pretende imitar el proceso humano en el análisis de geoimágenes. Por ello, lo que se necesita es definir una metodología que a partir de una geoimagen origen, genere como salida un conjunto de etiquetas de los objetos que integran dicha geoimagen (su semántica).

## **1.4 Objetivo general**

Definir una metodología general de descomposición semántica integrada por una etapa de análisis que permita extraer los objetos geográficos de una geoimagen dada basándose en un algoritmo de compresión semántica, así como de una etapa de síntesis que reconozca y asigne etiquetas en lenguaje natural a los objetos geográficos que integran la geoimagen, empleando una ontología de aplicación.

## **1.5 Objetivos particulares**

1. Definir el concepto de geoimagen como una superclase de imágenes digitales que describe objetos geográficos.
2. Expresar el concepto de semántica como un conjunto de etiquetas en lenguaje natural de los objetos geográficos que contiene una geoimagen.
3. Definir un conjunto de características integrado por atributos y propiedades de los objetos geográficos para el algoritmo de compresión semántica.

4. Conceptualizar e implantar un algoritmo de compresión semántica que permita simplificar las características de bajo nivel de los objetos geográficos y describirlos mediante intensidades de color uniformes.
5. Formular una etapa de síntesis para el reconocimiento de los objetos geográficos, la cual proporcionará un conjunto de etiquetas de los objetos identificados.
6. Definir una ontología de aplicación que permita especializar los objetos geográficos, integrada por clases y relaciones que permitan especializar el tipo de un objeto geográfico dado.

## 1.6 Hipótesis

1. Aunque la información de bajo nivel de los objetos geográficos es heterogénea y diversa, es posible extraer los objetos más significativos de una geoimagen.
2. Con base en un conjunto de características (propiedades y atributos) es posible definir un algoritmo de compresión semántica que simplifique las regiones de los objetos geográficos presentes en una geoimagen.
3. En este trabajo, se entenderá por semántica de una geoimagen al conjunto de etiquetas en lenguaje natural de las regiones de los objetos geográficos generadas por el algoritmo de compresión semántica.
4. Dicho conjunto de etiquetas es una aproximación de los objetos geográficos representados en una imagen geográfica, debido a que para poder detectar los objetos, es necesario realizar cierto agrupamiento entre objetos más pequeños.
5. Existe una ontología de dominio geográfico y sus subconjuntos (ontologías de aplicación) que involucran las relaciones necesarias y suficientes entre los objetos geográficos, para poder hacer un reconocimiento particular (especialización) de estos objetos.

6. Es posible definir un algoritmo de reconocimiento general que asigne una etiqueta a cada una de las temáticas obtenidas mediante la compresión semántica.
7. Con base en un conjunto de características que se obtiene a partir de la información de bajo nivel de una geoimagen, es posible hacer una identificación de las temáticas presentes en la geoimagen, cada una de ellas descrita como un conjunto de regiones.
8. Con una ontología de aplicación se puede hacer una especialización de los objetos geográficos de cada temática, basándose en las relaciones de los objetos geoespaciales.
9. Es posible delimitar la interpretación de una relación geográfica, tomando en cuenta la topología y la geometría de los objetos geográficos.
10. Es posible hacer una representación alterna de los objetos geográficos mediante primitivas tipo punto, línea o área, donde las dos últimas preserven las propiedades topológicas de cada objeto geoespacial.

## 1.7 Organización del documento

El resto del documento se encuentra dividido de la siguiente manera:

En el Capítulo 2 se describen los trabajos publicados hasta el momento relacionados con el presente trabajo de investigación. No existe una metodología que involucre todos los algoritmos que se proponen en este trabajo; las publicaciones descritas cubren parte de las etapas de nuestra metodología. En particular, se estudian algoritmos de segmentación, clasificación y etiquetado de objetos con base en la semántica de una geoimagen.

El Capítulo 3 cita los conceptos teóricos y las definiciones matemáticas que servirían como base para el entendimiento y comprensión de la metodología desarrollada. Se tratan temas de la percepción remota, procesamiento digital de imágenes y ontologías. En la sección de procesamiento digital de imágenes, se describen tópicos fundamentales para el análisis y el

procesamiento de las imágenes digitales, como son: preprocesamiento en el dominio espacial, segmentación, agrupamiento (*clustering*), etc. Posteriormente se estudia el concepto de las ontologías y su utilidad en el área de la computación.

En el Capítulo 4 se describe detalladamente el algoritmo de descomposición semántica desarrollado en esta tesis, el cual persigue describir la semántica de una geoimagen como un conjunto de etiquetas de los objetos que se extraen de una geoimagen comprimida semánticamente. Se divide en dos etapas: análisis y síntesis, donde la primera tiene el propósito de determinar la información (objetos geográficos) semánticamente importante de una geoimagen; la etapa de síntesis procesa esta información clasificándola primeramente en temáticas y posteriormente especializándola en etiquetas de lenguaje natural, haciendo uso de una ontología de aplicación del dominio geográfico.

El Capítulo 5 muestra los resultados obtenidos en el desempeño de la metodología propuesta. Se realizaron experimentos con diferentes tipos de geoimágenes para comprobar la eficiencia y las áreas de oportunidad de la metodología. De igual manera, se muestra una comparación con los resultados que se obtienen con una herramienta comercial reconocida.

Finalmente, en el Capítulo 6 se citan las conclusiones de esta investigación, aportes y las posibles líneas para el trabajo futuro.



## **Capítulo 2**

### **Estado del arte**

#### ***Resumen***

En este capítulo se describen y analizan las diferentes publicaciones que se relacionan con el trabajo de investigación. Al final de cada publicación se enlistan diversos comentarios referentes a las desventajas o limitaciones de cada técnica, los cuales ayudarán a guiar mejor la presente investigación.



## 2.1 Introducción

En este capítulo se describirán los principales trabajos relacionados con nuestra línea de investigación. Es importante hacer énfasis en el hecho de que no se encontró publicación o sistema alguno que trate de cubrir todas las etapas, así como el enfoque seguido, como se propone en este trabajo. La mayoría de las publicaciones se centran en resolver el problema de la segmentación y el reconocimiento del contenido de una geoimagen. Algunos están orientados a funcionar con imágenes digitales de escenarios naturales, mientras que otros a trabajar con imágenes provenientes de sensores remotos. Sin embargo, el tipo de imágenes con el que es posible trabajar es muy reducido (dominio del problema), lo cual limita ampliamente la viabilidad de la metodología correspondiente. Para su estudio, los trabajos se clasificaron en metodologías orientadas a píxeles y orientadas a objetos.

## 2.2 Metodologías orientadas a píxeles

Estas metodologías se caracterizan por usar directamente la información contenida en la imagen a procesar. Como ya fue citado en el capítulo anterior, tienen como desventaja el eliminar las componentes semánticas relevantes de las imágenes. Como se podrá observar, usan algoritmos conocidos de procesamiento de imágenes, sobre los que se realizan modificaciones para ajustarlos a los problemas que intentan resolver.

### 2.2.1 Un esquema de clasificación de imágenes jerárquico automático

**An Automatic Hierarchical Image Classification Scheme**

**Jim Huang, S. Ravi Kumar, Ramón Zabih (1998)**

En este artículo, se propone un método para realizar una clasificación jerárquica de imágenes empleando aprendizaje supervisado. La metodología recae en la extracción de características de bajo nivel y realizar subsecuentemente una reconfiguración del espacio de características usando descomposición de valores singulares (*Singular Value Decomposition – SVD*), lo que permite eliminar tanto ruido como dimensionalidad. Los datos de entrenamiento se usan para obtener un

árbol de jerarquía, el cual puede ser empleado para clasificar nuevas imágenes (Huang et al., 1998).

Como antecedentes a esta publicación, los autores mencionan el desarrollo del sistema en Internet WebSeek, el cual básicamente es un sitio donde un usuario puede realizar una búsqueda mediante palabras claves en un banco de imágenes digitales (WebSeek, 2007). Para poder obtener las palabras de una imagen, se tiene que proporcionar un archivo adicional a la imagen a clasificar<sup>2</sup>, el cual contiene las palabras claves de la imagen; sin embargo, no se aplica ningún algoritmo que permita comprobar que efectivamente las palabras claves proporcionadas corresponden con el contenido semántico de la imagen, es decir, no se realiza ningún tipo de análisis sobre la imagen.

Por lo anterior, cuando un usuario ingresa una palabra para obtener todas las imágenes que satisfacen el criterio de búsqueda, se pueden obtener imágenes que no corresponden con lo esperado por el usuario.

Un sistema de clasificación de imágenes parte de la definición de tres problemas fundamentales: 1) definición de las características que representarán a la imagen 2) organización de los datos característicos extraídos y 3) la forma en que se clasificará la imagen. A partir de un conjunto de imágenes de entrenamiento, para las cuales se conoce en cada una de ellas la clase o categoría a la que pertenece, por ejemplo, ríos, rocas, cascadas, avión, motor, etc., se usa una técnica conocida como correlograma de bandas de color para extraer las características de bajo nivel de la imagen. Básicamente, el correlograma es la probabilidad normalizada de que para cada punto de la imagen  $p_1$ , a una distancia  $d$ , se encuentre un píxel  $p_2$  que tenga el mismo color de  $p_1$ . Para medir la distancia entre pixeles, se usa la norma  $L_1$ <sup>3</sup>. Suponiendo que una imagen  $I$  tiene un total de  $m$  colores, se definirá un vector beta con  $m$  componentes.

Posteriormente, se usa la descomposición SVD para reducir el número de componentes del vector beta. En el artículo se propone una metodología para encontrar el valor de las  $k$ -componentes que

---

<sup>2</sup> Cuando se desea dar de alta una nueva imagen en este sistema, en realidad se proporciona un archivo HTML que contiene tanto las palabras claves como la imagen digital.

<sup>3</sup> La norma  $L_1$  a la que se refieren estos autores es la de la distancia  $D_4$ .

se deben usar de dicha descomposición; el criterio se basa en obtener el porcentaje del último de los valores propios de la descomposición SVD. Terminado este proceso, se tiene un total de  $n$  vectores beta con  $k$  componentes, donde  $n$  es el número de imágenes de entrenamiento.

Para generar el árbol de jerarquía, se define una matriz de confusión en base a cada una de las imágenes conocidas a priori. Dicha matriz de confusión se obtiene determinando a qué clase semántica pertenece cada imagen a priori, cuando dicha imagen es quitada del conjunto original de entrenamiento. Esto es, el conjunto de entrenamiento puede ser el conjunto  $I = \{I_1, I_2, \dots, I_n\}$ , primeramente se elimina la imagen  $I_1$  del conjunto, lo cual resulta en  $I = \{I_2, \dots, I_n\}$ . Con la imagen  $I_1$ , se prueba el sistema para extraer su vector beta reducido a  $k$ -componentes y determinar la imagen  $I_j$  con la que tiene más similitud y con ello saber a qué clase semántica pertenece  $I_j$ . El resultado es la clase de confusión a la que pertenece la imagen  $I_1$ .

Con la matriz de confusión calculada, se aplica la técnica de cortes normalizados para separar dos grupos, en los que la asociación entre clases sea mínima y la asociación entre grupos sea máxima, para finalmente generar el árbol de jerarquía entre las clases de entrenamiento. Cuando una nueva imagen necesita ser clasificada, se obtiene su vector beta reducido y se busca en los del conjunto de entrenamiento el que minimice la diferencia entre los vectores beta. En la Figura 2–1 se muestra la jerarquía obtenida por los autores para un conjunto de imágenes de entrenamiento. Se puede observar que existen nodos en la jerarquía donde no hay nombres; solamente se asignan nombres en los nodos terminales, es decir, cuando se ha reconocido un objeto.

### Comentarios

En este caso, los autores proponen una técnica que se basa en el correlograma normalizado para poder extraer características de la imagen original. Desafortunadamente para imágenes reales, la complejidad del algoritmo es demasiado elevada. En las pruebas realizadas por los autores, se usan imágenes con 8 niveles de intensidad para cada componente RGB, es decir, un total de 512 colores diferentes. En la realidad, las imágenes comúnmente presentan un rango de intensidades de 256 niveles para cada componente, lo cual de entrada limita ampliamente el uso de esta metodología en imágenes reales.

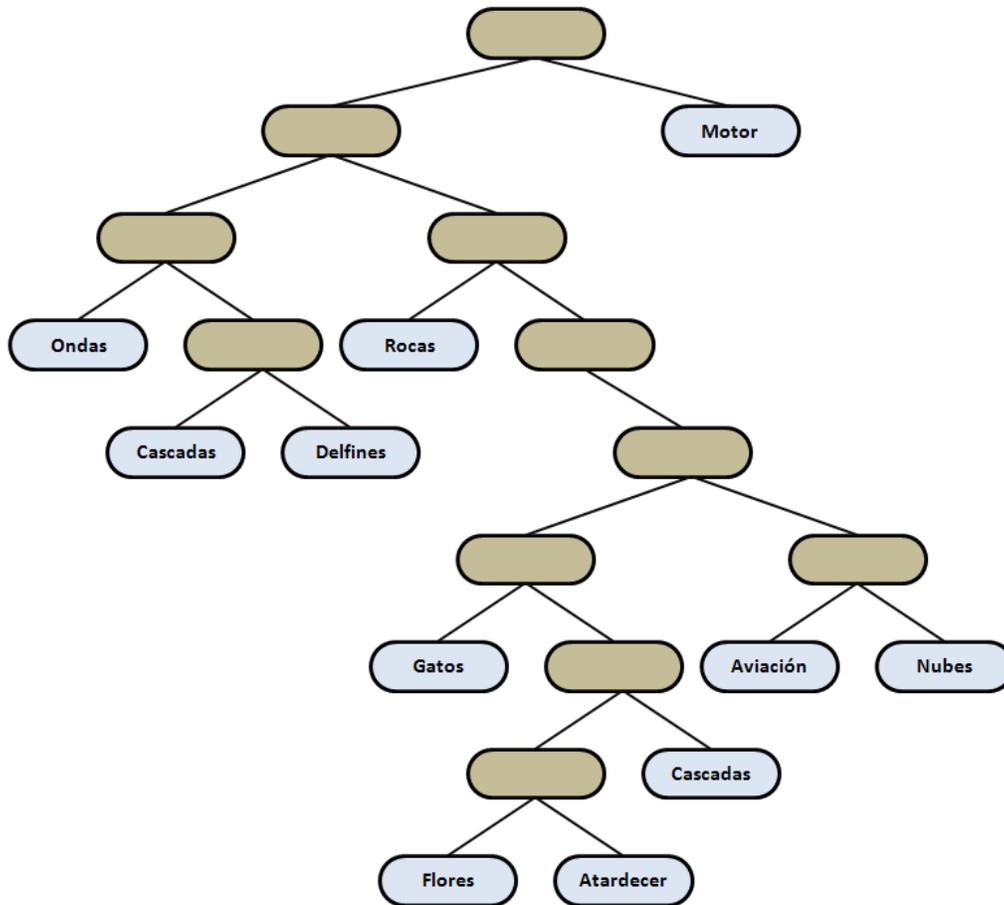


Figura 2–1. Jerarquía obtenida con el conjunto de imágenes de entrenamiento.

A lo anterior, también podemos agregar que el correlograma se basa ampliamente en calcular distancias para cada píxel de la imagen a procesar. Nuevamente, en casos prácticos, donde se pueden tener muestreos de 1024 x 768 píxeles o más, común para las geoimágenes, el tiempo de cómputo del correlograma puede no ser adecuado.

El uso de la técnica SVD, según los autores, permite eliminar tanto ruido como el número de componentes de la matriz original. SVD es usado en otras áreas como en procesamiento de texto y sobre todo, en compresión de imágenes. Sin embargo, no se justifica claramente la forma de elegir las  $k$ -componentes, debido a que no se muestra una tabla comparativa donde indiquen los resultados obtenidos empleando diversos umbrales al propuesto en el artículo.

Posiblemente, lo que resulta más ambiguo es la jerarquía de clasificación que obtienen para el conjunto de imágenes de entrenamiento. Debido a que como se mostró en la Figura 2–1 no existe

una relación entre los niveles de la jerarquía, por ejemplo, si observamos los nodos que contienen clases relacionadas con seres vivos, en un nivel se encuentran los gatos, mientras que en otros están los delfines. En consecuencia, si bien la metodología puede clasificar imágenes muy restringidas, por su muestreo tanto espacial como en color, el diseño de la misma no muestra una estructura semántica correcta. Asimismo, no se explica qué pasa si una imagen no tiene un contenido semántico similar al de las imágenes de entrenamiento, esto es, no se justifica claramente la forma de hacer crecer la jerarquía.

### 2.2.2 Segmentación de imágenes usando segmentación de color basada en las ondeletas rápidas y el crecimiento de regiones direccional

**Image Segmentation with Fast Wavelet-Based Color Segmentation and Directional Region Growing**

**Chan Din-Yuen, Lin Chih-Hsueh & Hsieh Wen-Shyong (2005)**

Estos autores comentan que dentro de las principales dificultades que se tienen en los algoritmos de segmentación existentes se encuentran: complejidad de los cálculos o requerir un elevado conocimiento a priori. Su trabajo de investigación está basado en la definición de una nueva metodología que permite segmentar imágenes sin requerir una gran cantidad de conocimientos a priori (únicamente un umbral denotado como  $T$ ), así como de algoritmos que según los autores, procesan la imagen de manera rápida.

Su propuesta está compuesta de tres grandes etapas, las cuales son:

1. Uso de la transformada discreta de wavelets.
2. Crecimiento de regiones dirigido.
3. Inspección de la textura entre objetos.

En la primera parte se hace uso de la transformada discreta de wavelets para procesar una imagen previamente transformada del espacio RGB, al espacio YUV. Los autores no justifican el motivo por el que cambian el espacio de color, pero en base a su publicación, se concluye que permite trabajar una imagen tanto en color, como en niveles de gris; con esto se define un algoritmo más general. Cuando la imagen original contiene diversas tonalidades de color, se usarán las componentes U y V, mientras que si el espacio de color solamente contiene variaciones de intensidad (niveles de gris) se usará la componente Y (Din-Yuen et al., 2005).

Primeramente, se aplica iterativamente un proceso de descomposición empleando la transformada discreta ondeleta sobre las imágenes de las componentes U y V. Al aplicar una transformación de este tipo, se generan 4 nuevas imágenes, 3 de las cuales describen componentes en alta frecuencia (bordes horizontales, verticales y diagonales principalmente), la cuarta imagen generada contiene las componentes de baja frecuencia (LL) de la imagen original. Al terminar cada transformación se calcula un parámetro para determinar si se debe aplicar otro nivel de la transformación ondeleta o se detiene el proceso. A grandes rasgos, este parámetro se obtiene combinando los valores U y V de cada imagen LL para cada par de punto de la imagen. Si algunos puntos no satisfacen un criterio, su parámetro respectivo es igualado a cero.

Terminado el proceso de transformación mediante ondeletas, se procede primeramente a normalizar las intensidades de color para generar una paleta de los principales colores presentes en las imágenes LL. Con esta paleta de colores se hace una segmentación de la imagen original.

Por último, mediante el crecimiento dirigido de regiones y la inspección de la textura entre objetos, se analizan aquellas regiones que sean vecinas para determinar si son independientes o deben ser fusionadas como región única mediante la inspección de la similitud entre la textura de cada región. En la Figura 2–2 se muestran las imágenes originales y los resultados de la segmentación obtenidos empleando esta nueva metodología. Se puede observar que se hace una segmentación asociando en la misma región, los píxeles de las regiones que describen la misma información. Sin embargo, este tipo de resultado puede no ser el adecuado para la mayoría de las aplicaciones.

### **Comentarios**

Los autores hacen énfasis en el hecho de que se pretende desarrollar una metodología que permita obtener los objetos semánticos de una imagen real. El principal punto a destacar es que no proporcionan una definición de objeto semántico, únicamente se limitan a comentar que pretenden segmentar de manera similar a como lo realiza el sistema visual humano.

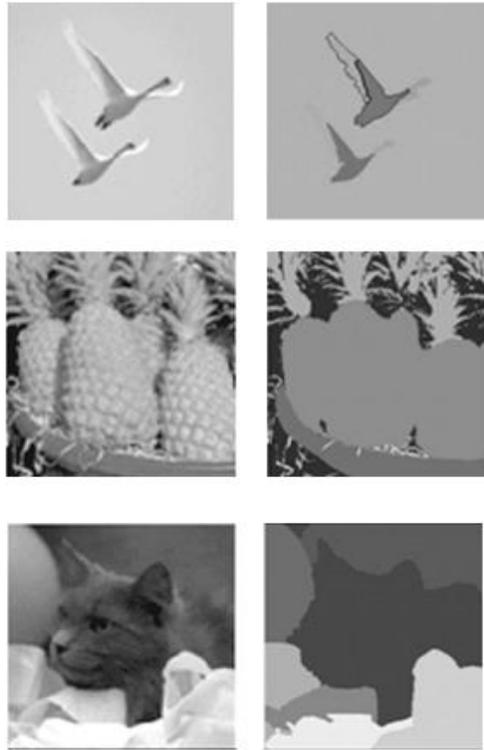


Figura 2-2. Resultados al aplicar el algoritmo propuesto.

Desafortunadamente, no existe una explicación justificada que detalle el mecanismo como segmenta el sistema visual humano, por lo que no es posible determinar si los resultados que obtienen estos investigadores coinciden con la forma en que segmenta una persona. Asimismo, no existe una justificación clara del por qué realizar un cambio en el espacio de color, transformando del espacio RGB al espacio YUV. En la Figura 2-3 se muestra los resultados de realizar este cambio del espacio de color.

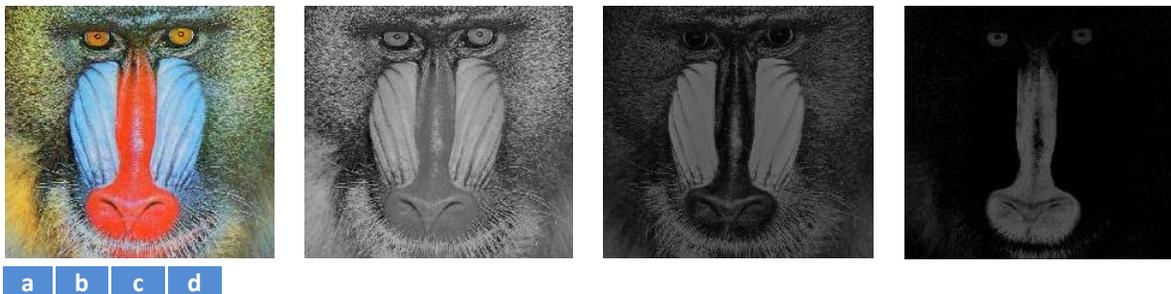


Figura 2-3. Transformación del espacio de color (a) Imagen en el espacio RGB (b) Componente Y del modelo YUV (c) Componente U (d) Componente V.

Adicionalmente, el uso de los wavelets permite determinar el nivel  $k$  hasta el cual se debe detener la transformación para obtener la paleta de colores o de niveles de gris. Sin embargo, al realizar una serie de pruebas con un script propio escrito en Matlab, se obtuvieron resultados interesantes. En la Figura 2–4(a) se muestra la componente U de la imagen original a procesar, en la Figura 2–4(b) la imagen de baja frecuencia que se obtiene al aplicar 4 niveles de la transformada Haar, y por último en la Figura 2–4(c), la imagen que se obtiene al hacer un cambio de muestreo en la imagen original. Se puede apreciar que las dos últimas imágenes son muy similares. La Figura 2–4(b) es la que se usaría en la metodología propuesta por los autores para obtener la paleta de colores. Entonces, posiblemente el uso de la transformada wavelet no es obligatorio, ya que con otro algoritmo más sencillo se podrían obtener los mismos resultados.

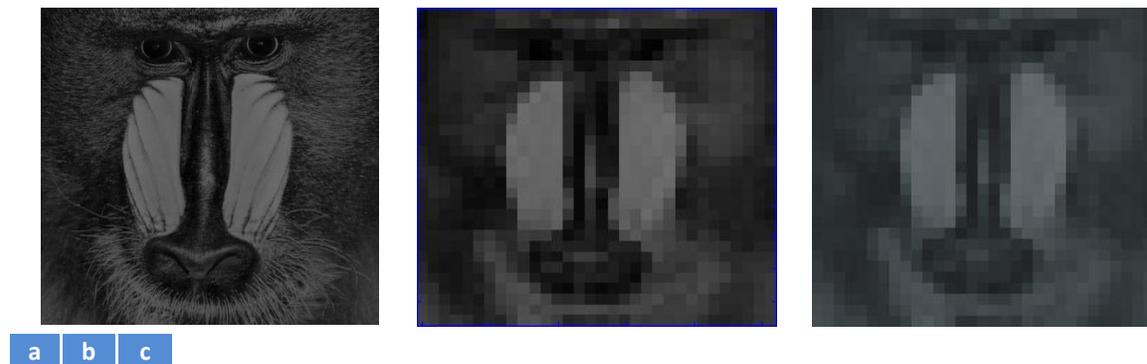


Figura 2–4. Comparación entre wavelet y cambio de muestreo (a) Imagen original; (b) Componente LL con nivel  $k=3$ ; (c) Resultado al cambiar el muestreo de la imagen original.

Por último, se menciona que los algoritmos se ejecutan en corto tiempo, con lo cual se podría aplicar en sistemas de tiempo real. Desafortunadamente, no hay ninguna sección dedicada al análisis de complejidad que permita justificar claramente dicho argumento.

### 2.2.3 Recuperación de imágenes basado en semántica: Un enfoque probabilístico

#### Semantic Based Image Retrieval: A Probabilistic Approach

Ben Bradshaw (2000)

Este artículo describe una técnica para obtener imágenes basándose en la semántica subyacente de las imágenes. Para obtener dichas semánticas el autor propone una metodología jerárquica basada en probabilidad. El conjunto de etiquetas obtenidas permiten clasificar una imagen compuesta de objetos naturales, artificiales, y si es una escena de interior o de exterior. El marco

de aplicación de la jerarquía combina la técnica de máxima probabilidad por clases, la cual se estima en diferentes niveles, para posteriormente estimar la probabilidad de pertenencia a una clase. A diferencia de los trabajos que se han hecho en este campo, el algoritmo propuesto puede determinar las probabilidades en cualquier punto de la imagen y solamente se requiere de un pequeño conjunto de imágenes para realizar el entrenamiento del sistema (Bradshaw, 2000).

Las principales contribuciones de esta publicación son: la ubicación de las etiquetas semánticas en una porción de la imagen (ver Figura 2–5), la poca cantidad de imágenes de entrenamiento que se requiere<sup>4</sup>, la reducida complejidad en la ejecución de la clasificación así como también, la identificación de las etiquetas por medio de probabilidades. Lo anterior es de vital importancia cuando se quiere hacer consultas sobre un conjunto amplio de imágenes.



Figura 2–5. Etiquetado por regiones (a) Imagen original (b) Regiones de objetos artificiales.

La propuesta define dos tipos de etiquetas semánticas. El primer conjunto trata de etiquetar pequeñas regiones de la imagen como naturales o artificiales (objetos construidos por el hombre), mientras que el segundo conjunto trata de etiquetar la imagen completa para determinar si fue tomada dentro de un edificio o desde el exterior.

Para hacer la clasificación, el algoritmo inicia haciendo un muestreo recursivo sobre la imagen, empleando un tamaño de ventana o bloque como se especifica en la Tabla 2–1. Al terminar el muestreo anterior, se tiene un total de  $n$  bloques en cada nivel de muestreo. Para cada bloque se calcula un vector característico de 26 componentes para cada bloque. Los vectores característicos cuantifican propiedades de color y textura, empleando las transformadas Ohta y de ondeletas compleja, respectivamente. Para simplificar cálculos y tiempo de procesamiento, se aplica el discriminante de Fisher a cada vector.

<sup>4</sup> El autor cita que se requieren solamente 100 imágenes.

Tabla 2–1. Definición de los tamaños de ventana para cada nivel de muestreo.

Nivel	Tamaño de la ventana o bloque
0	Imagen completa
1	128 x 128
2	64 x 64
3	32 x 32
4	16 x 16

Posteriormente, se determina una densidad de probabilidad en cada nivel, usando los valores escalares que se obtienen del discriminante de Fisher. Por último, empleando técnicas de probabilidad y la técnica de máxima probabilidad por clase, se combinan las probabilidades de los niveles, para obtener la probabilidad de que una imagen pertenezca a cierta clase.

Debido a que la técnica de máxima probabilidad por clase requiere de probabilidades a priori, se hace un entrenamiento utilizando un conjunto de imágenes homogéneas de objetos naturales y otro de objetos artificiales, para el primer tipo de clasificación; para el segundo tipo, se usan imágenes tomadas desde el interior de un edificio o al aire libre.

### Comentarios

Primeramente, no se ha demostrado que mediante un enfoque probabilístico, se resuelvan completamente los problemas de clasificación, y para nuestro caso, la clasificación semántica. Esto se puede observar claramente en los resultados obtenidos de este trabajo de investigación que son cercanos al 85% y en los casos óptimos, del 91% de eficiencia.

Un enfoque probabilístico determina la probabilidad de que un objeto pertenezca a cada clase del conjunto de clases disponibles, y asigna el objeto a la clase con la probabilidad más alta; sin embargo, el argumento anterior no garantiza que el objeto realmente deba ser clasificado en dicha clase; simplemente, el clasificador concluye que el objeto es muy similar a las características de la clase a la que se está asociando. En nuestro problema de estudio, se debe tener la certeza de que al clasificar un objeto en una clase, es porque pertenece a esa clase.

Aunque se requiere de 100 imágenes de cada clase para realizar un entrenamiento del algoritmo, si es necesario hacer una clasificación para otras categorías, se requiere realizar otro entrenamiento, con lo cual, esta metodología representa muchas limitantes.

Asimismo, dentro de la publicación se cita el hecho de que las imágenes no exceden de un tamaño de 400 x 400 píxeles, lo cual es un tamaño muy reducido y pudo influir directamente en los tiempos de procesamiento. Debido que no se justifica el tipo de complejidad de la metodología propuesta, al aplicarlo con imágenes reales comunes (800 x 600, 1024 x 768, etc.) se podría obtener un aumento cuadrático, exponencial, o de otro tipo, en el tiempo de respuesta para la clasificación.

Otro punto que no queda claro del algoritmo, es la forma en que se realiza el muestreo recursivo sobre las imágenes. Únicamente se indica que un punto en particular de la imagen (píxel) se toma como centro para todas las ventanas que haya que aplicar, pero no se indica claramente la forma de tomar el siguiente punto de referencia. Ni tampoco, se indica por qué se debe llegar hasta un bloque mínimo de 16 x 16 píxeles; si el aplicar una ventana de 8 x 8 da mejores resultados, es una interrogante abierta.

Por último, las imágenes usadas con fines de pruebas, fueron tomadas de la galería de imágenes de la herramienta comercial Corel. El principal problema, es que la mayoría de las imágenes incluidas en dicha galería, están bajo condiciones muy optimistas de iluminación y posición; no se hicieron pruebas con imágenes que presenten ruido, distorsión, etc.

#### **2.2.4 Etiquetado semántico de imágenes que combina color, textura y palabras clave**

##### **Semantic Labeling of Images Combining Color, Texture and Keywords**

**Andres Dorado & Ebroul Izquierdo (2003)**

Estos autores publican una técnica para el etiquetado de imágenes que combina color y textura con palabras clave. Empleando un método basado en la similitud de color junto con una técnica de minería de datos, se propagan las palabras claves extraídas de un subconjunto de imágenes dentro de una base de datos de larga escala. Asimismo, un método basado en el análisis de las

propiedades de textura se aplica para relacionar las palabras reservadas con las regiones de la imagen a procesar (Dorado & Izquierdo, 2003).

El proceso de asociar etiquetas a una imagen se divide en dos etapas: etiquetado global y local. En el etiquetado global se utilizan las palabras clave para clasificar la imagen completa. Para ello se parte de información *a priori*, que consiste de una serie de tuplas de una imagen con una o varias palabras reservadas. Los autores no explican cómo asociar a una imagen de entrada, una etiqueta de esta base de conocimientos.

Para el etiquetado local primeramente se hace una partición de la imagen original usando un tamaño predeterminado. Los investigadores utilizaron el tamaño de 8 x 8 para particionar las imágenes empleadas en la sección de pruebas. Realizado lo anterior, se emplea el algoritmo c-medias para clasificar las regiones con textura similar. Para cuantificar la textura de cada región emplean las ondeletas de Gabor, obteniendo como resultado un conjunto de vectores característicos.

Como parte adicional de información *a priori*, se deben tener texturas sintéticas asociando a cada una de ellas, un vector característico que las describa, así como una palabra clave. Mediante la distancia Euclidiana entre cada vector de la imagen particionada y los que integran la base de conocimientos, se asocia la palabra clave de la textura sintética a la región en estudio, que minimiza la distancia entre vectores. En la Figura 2–6 se muestran dos imágenes de prueba<sup>5</sup> y las palabras clave que fueron asignadas a dichas figuras.

### Comentarios

El etiquetado local es el algoritmo que tiene la mayor parte de la complejidad en este trabajo, debido a que es ahí donde hace la partición de la imagen original para después clasificar en regiones similares empleando el algoritmo c-medias. El principal problema de este algoritmo es que consume tiempo de procesamiento, como los autores lo mencionan en sus conclusiones y trabajo futuro. Por otra parte, no justifican el por qué usar este algoritmo, así como las ondeletas

---

<sup>5</sup> Se buscaron imágenes similares a las de la publicación original, ya que la calidad de las mismas era muy deficiente.

de Gabor para describir las texturas. En lo que respecta al etiquetado global, los autores no especifican la forma en que se asocian las etiquetas candidatas a la imagen completa.



Montañas, puesta de sol, cielo, nubes.



Montañas, árboles, cielo.

a

b

Figura 2–6. Resultados del algoritmo, (a) Imágenes a procesar (b) Palabras clave asociadas.

A lo anterior también podemos agregar que no se describe cómo generar toda la información a priori, es decir, lo siguiente no puede ser concluido con el artículo: ¿Qué es una textura sintética?, ¿Tiene el mismo tamaño que las regiones de partición?, ¿Para describir un mismo objeto, se proporcionan texturas iguales o diferentes?, ¿Cuánto es la cantidad mínima de texturas sintéticas para que funcione correctamente el algoritmo? De igual forma no se da la explicación requerida para hablar sobre las palabras clave, si describen objetos como entes atómicos, o bien las palabras clave describen partes de objetos divisibles, como la palabra clave nieve que puede ser parte de una montaña.

Por otra parte, como en el artículo anterior, los autores toman imágenes de la galería de Corel para hacer sus pruebas. Por ello, hubiera sido más conveniente emplear imágenes obtenidas de

otras fuentes. Las imágenes con las que realizaron pruebas son solo de montañas con presencia de nubes, es decir, no probaron la eficiencia del algoritmo con un espectro más amplio de objetos.

### 2.2.5 Diseño e implementación de un sistema de recuperación semántica de imágenes

#### An Image Semantic Retrieval System Design and Realization

Tong – Zhen Zhang, Yong – Gang Fu (2005)

En este artículo, se presenta una nueva arquitectura para el etiquetado semántico de imágenes y consulta. Emplean una red semántica como modelo de representación de la semántica y para hacer una consulta sobre el conjunto de imágenes, usan la combinación de palabras clave semánticas, una ontología lingüística y características de bajo nivel (Zhang & Fu, 2005).

Este trabajo se apoya en dos trabajos relacionados previamente: una red semántica (Lu et al., 2000) y la asociación semántica (Zhuang et al., 1999). Una red semántica se representa por medio de un conjunto de palabras clave, las cuales están relacionadas con una colección de imágenes donde a cada relación se le asigna un determinado peso. Como se puede observar en la Figura 2–7, cada imagen se puede asociar con múltiples palabras clave, con lo cual una imagen puede representar diferente información semántica. Asimismo, cada relación  $w_{ij}$ , denota el grado de asociación entre la palabra clave  $i$  con la imagen  $j$ . Con esta red, la similitud entre imágenes puede obtenerse cuantificando la similitud entre los conjuntos de palabras clave con las que están relacionadas las imágenes.

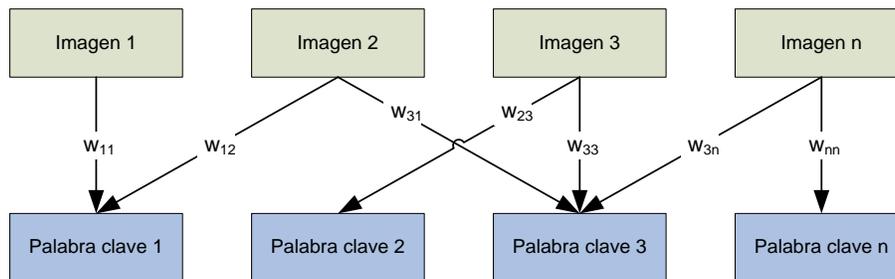


Figura 2–7. Ejemplo de una red semántica.

El principal inconveniente de una red semántica es la cantidad de trabajo que se requiere para poder construirla. En diversas publicaciones (Zhong et al., 2002), se ha abordado este problema y se presentan tres soluciones para ello. El primero es que a partir del conjunto de imágenes y

palabras clave, se establezcan manualmente todas las relaciones. El segundo es obtener las palabras clave por diversos métodos como: del nombre del archivo, buscar etiquetas específicas para imágenes incrustadas en una página HTML o del título de la página web, etc. El tercer método es usando una técnica de retroalimentación con el usuario para incluir las palabras clave en una consulta.

El otro trabajo en el que se basan, es el desarrollado por Zhuang, el cual emplea WordNet como ontología lingüística (Zhuang et al., 1999). WordNet fue desarrollado por George Miller de la Universidad de Princeton, donde cada sustantivo se organiza en base a sinónimos. En dicha ontología se emplean tres tipos de relación: *es un*, *miembro de* y *parte de*. En la Figura 2–8 se puede observar un ejemplo de la relación *es un*.

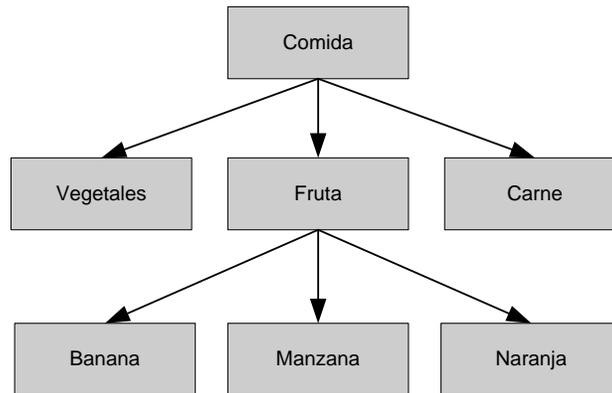


Figura 2–8. Ejemplo de relación *es un*.

La arquitectura propuesta por estos investigadores se puede observar en la Figura 2–9. Existe una colección que contiene una gran cantidad de imágenes. En cada imagen, existe un vector de características predeterminadas de bajo nivel. Asimismo, se tiene una red semántica y una ontología del tipo WordNet. El usuario ingresa una consulta y el sistema procede a calcular la similitud entre las imágenes en la colección y la consulta, para lo cual se emplean tanto la red semántica como la ontología. De acuerdo al grado de similitud, se genera una lista de resultados ponderados u ordenados, para que posteriormente la interfaz de usuario despliegue la lista de imágenes al usuario. Es en este punto donde el usuario puede retroalimentar al sistema, para modificar las palabras clave asociadas a la lista de imágenes de resultado, cambiar pesos en la red semántica, entre otras opciones.

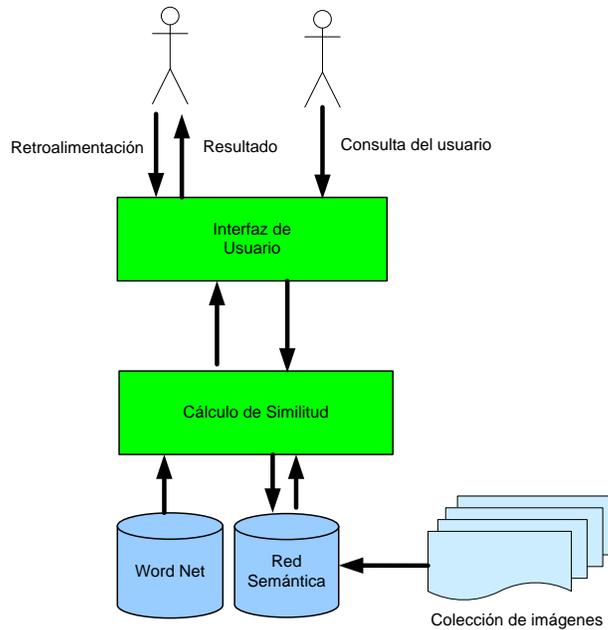


Figura 2–9. Arquitectura propuesta.

### Comentarios

El uso tanto de la red semántica con relaciones ponderadas y una ontología para la búsqueda de sinónimos, proporciona gran robustez a la propuesta de estos investigadores, debido a que es posible manejar las palabras clave de una forma más eficiente a diferencia de otras propuestas que se han estudiado en este documento. Otro punto que es de resaltar es la flexibilidad de incrementar las imágenes de la colección, así como la retroalimentación del usuario que permite corregir errores en el sistema.

A pesar de que se explica detalladamente la mayoría de los aspectos de la metodología, no se justifica claramente la forma de obtener el vector con las características de bajo nivel para cada imagen de la colección. Este punto puede influir notoriamente en la eficiencia de la metodología, debido a que la mayoría de los trabajos relacionados se enfocan en emplear diversas técnicas para describir las características de bajo nivel de una imagen. Con esto, se reduce la aportación científica de este artículo, ya que tanto la red semántica como la ontología, son trabajos previamente desarrollados.

Desafortunadamente, la publicación no cuenta con una sección de pruebas y resultados donde se demuestre numéricamente la eficiencia del algoritmo propuesto. A pesar de que los autores dicen en el artículo que el sistema ya fue implementado, solamente se describe mediante palabras el funcionamiento del mismo. Se omiten datos fundamentales como el número de imágenes en la colección, resolución y tipo de las imágenes empleadas.

### 2.2.6 Aprendizaje de la semántica de primitivas y de escenas de imágenes para su clasificación y recuperación

#### Learning Primitive and Scene Semantics of Images for Classification and Retrieval

Cheong Yiu Fung, Kia Fock Loe (1999)

Los autores publican una técnica de aprendizaje basada en semántica para poder clasificar y consultar imágenes digitales. Consiste en dividir primeramente la imagen en una matriz de regiones continuas con un tamaño de 32 x 32 píxeles cada una. Posteriormente se define la semántica en dos niveles: (1) Semántica primitiva a nivel de región, donde los autores se enfocan en el reconocimiento del significado de las características de bajo nivel, para cada una de las regiones. (2) Semántica de la escena, relacionada con el reconocimiento de las clases de escenas de alto nivel presentes en la imagen, a través de la asociación de las semánticas primitivas obtenidas en el punto anterior (Fung & Loe, 1999).

Debido a que cada región describe una zona más homogénea que la imagen completa, es posible encontrar un mapeo entre las características de bajo nivel de la región y el correspondiente significado semántico. El sistema aprende y representa este mapeo como un conjunto de palabras de imágenes. Estas palabras convierten las características visuales de las regiones, en palabras que indican qué es lo que hay en esa región. Los autores afirman que una vez que se ha reconocido la semántica primitiva de una imagen, la semántica de la escena se puede identificar correctamente a partir de patrones de asociación de las etiquetas de las imágenes (*picture words*) usando técnicas estadísticas.

En la Figura 2–10 se muestran los componentes principales de la propuesta de este trabajo de investigación. En el filtrado de cada región se emplea un filtro basado en la teoría de medición de información de una imagen, desarrollada por Dumains (Dumains, 1999). El objetivo es eliminar

aquellas regiones de la matriz que no cubran una cantidad mínima de información, que no sean espacialmente homogéneas y que pudieran afectar el cálculo de la semántica primitiva. Para modelar cada región de la imagen, se usa el espacio de color LUV, donde se generan vectores de 6 componentes para cada región.

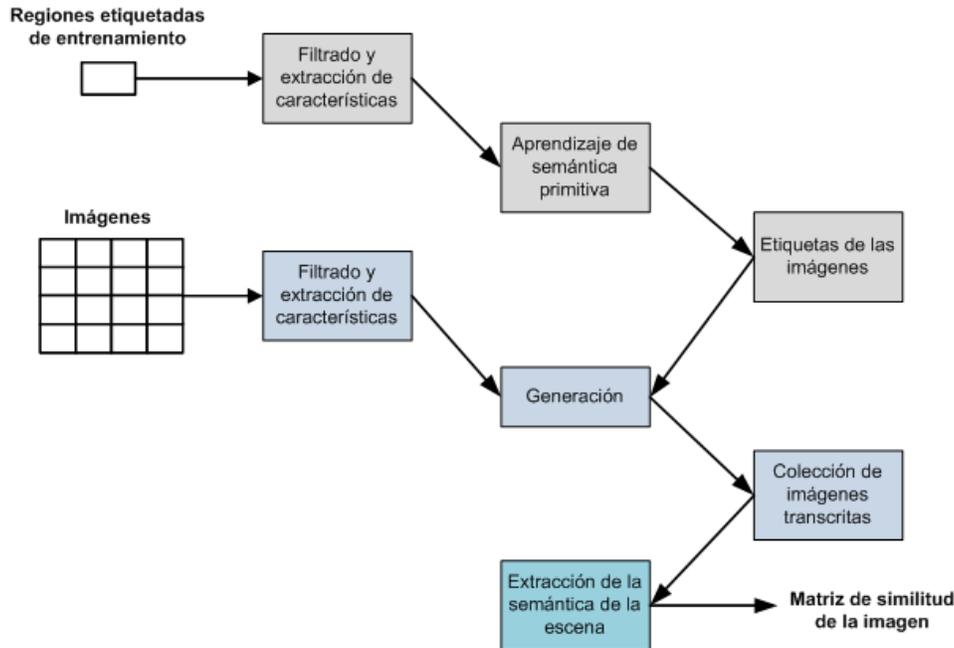


Figura 2–10. Metodología para el cálculo de la semántica de una imagen.

Este algoritmo requiere de una etapa de aprendizaje de las etiquetas de las imágenes, que se forman a partir de un conjunto de regiones de entrenamiento con sus respectivas clases semánticas conocidas. Debido a que las regiones para una misma clase pueden tener diferentes condiciones de iluminación, o bien, que haya un error del usuario en el que una región de una clase se asigne a otro grupo de regiones, los vectores característicos que se obtendrían en el espacio LUV para las regiones de dicha clase presentarían variaciones elevadas. Por lo anterior, los autores proponen un algoritmo basado en el  $k$ -medias, para hacer una segmentación de todos los vectores característicos de todas las clases en el espacio de color LUV. Primeramente se hace una división de regiones hasta llegar a un umbral preestablecido para posteriormente realizar una fusión de las regiones que pertenezcan a la misma clase semántica.

Definido el vocabulario o conjunto de etiquetas de imágenes, se define un conjunto de imágenes para que sean clasificadas usando el vocabulario previamente definido. Como ya fue citado, cada

imagen se divide en regiones de 32 x 32 píxeles, las cuales pasan por un proceso de filtrado, se extraen sus vectores característicos para obtener la semántica primitiva y se comparan dichos vectores con los existentes en la base de conocimientos empleando una versión modificada del clasificador de vecino más cercano. La semántica de la clase se fundamenta en el uso de la técnica de descomposición de valores singulares (*Singular Value Decomposition - SVD*), para obtener las clases de mayor relevancia del conjunto de regiones en que fue descompuesta la imagen.

### Comentarios

Como principal limitante de esta metodología se tiene que las consultas que se pueden realizar a la aplicación tienen que contener palabras ya existentes en el vocabulario del sistema. Es aquí donde el uso de un diccionario de palabras, como en el artículo anterior, sería de gran utilidad para la identificación de sinónimos o relaciones entre las palabras.

Adicionalmente, como datos numéricos referentes a las pruebas del sistema, se indica únicamente que para la etapa de aprendizaje se usaron 6574 regiones que representaban un total de 20 clases semánticas. A partir de este conjunto se creó un vocabulario de 65 palabras. En la publicación, no se explica una etapa de aprendizaje *a posteriori*, para incrementar el contenido del vocabulario. Por lo anterior, hay que hacer un nuevo aprendizaje para nuevas palabras, lo cual limita ampliamente esta teoría.

La etapa de semántica primitiva se describe ampliamente en la publicación; sin embargo, la obtención de la semántica de la escena, es expuesta muy escuetamente. De hecho, solamente se menciona que se usa la técnica SVD para reducir el tamaño de los vectores de la primera etapa. Aunado a lo anterior, las pruebas se realizan también con imágenes de una galería, como ya se ha citado ampliamente en este documento, lo cual no permite corroborar la eficiencia del sistema con imágenes que presenten ruido u otros fenómenos adversos.

### 2.2.7 Técnicas basadas en la percepción para la clasificación y recuperación semántica de imágenes

#### Perceptually Based Techniques for Semantic Image Classification and Retrieval

Dejan Depalov, Thrasyvoulos Pappas, Dongge Li, Bhavan Gandhi (2006)

La acumulación de largas colecciones de imágenes plantea la necesidad de diseñar e implementar esquemas eficientes e inteligentes que permitan obtener imágenes basándose en el contenido de las mismas. Lo anterior requiere el entendimiento de las categorías semánticas importantes que usan las personas para hacer una clasificación de las imágenes, así como la extracción de características importantes de la imagen que permitan discriminar entre dichas categorías (Depalov et al., 2006).

En la actualidad, los algoritmos para la extracción de características de bajo nivel, como son el color, la textura y la forma, están bastante estudiados y desarrollados. Sin embargo, la extracción de características de bajo nivel de la imagen que estén correlacionadas con la semántica de alto nivel es un problema abierto. Por lo anterior, estos autores publican en este trabajo un algoritmo para la clasificación semántica que utiliza el algoritmo de segmentación basado en color y textura desarrollado por Chen (Chen et al., 2005). El artículo se enfoca en la asignación de etiquetas semánticas a los segmentos resultantes de la segmentación. Estos descriptores son la clave para poder relacionar las características de bajo nivel con la semántica de alto nivel de una imagen.

La selección apropiada de características de color y textura es crucial tanto para la segmentación como para la clasificación de imágenes digitales. El método de Chen se basa en dos tipos de características espacialmente adaptativas. La primera proporciona una descripción localizada de la composición de color de la textura y el segundo modela la orientación de la textura en el dominio de la frecuencia. Los autores remarcan que ambas incorporan modelos de la percepción humana y características de señal. La característica de composición de color explota el hecho de que el sistema visual humano no es capaz de percibir simultáneamente un número grande de colores.

Para obtener los colores dominantes en una imagen se hace uso del algoritmo de segmentación adaptativo definido por Pappas (Pappas, 1992), ver Figura 2–11(b). La extracción de las

características espaciales de la textura está basada en una descomposición de la frecuencia en multiescala con cuatro orientaciones de sub-banda (horizontal, vertical,  $+45^\circ$  y  $-45^\circ$ ).



Figura 2–11. Técnica de Chen (a) Imagen original (b) Colores dominantes adaptativos; (c) Clases de textura; (d) Segmentación final.

La energía local de los coeficientes de sub-banda se emplea como una caracterización sencilla, pero efectiva, de la textura espacial. En cada posición de píxel, el máximo de los cuatro coeficientes determina la orientación de la textura. Se aplica un filtro mediana para hacer un suavizado en las regiones homogéneas, así como para eliminar los cambios bruscos entre cambios significativos de texturas. Finalmente los píxeles se clasifican en dos clases: de tipo liso o no liso; adicionalmente, los píxeles se clasifican de una clase de textura no lisa en un tipo de orientación dominante, esto es horizontal, vertical,  $+45^\circ$ ,  $-45^\circ$  o complejas, estas últimas, donde no hay una orientación dominante.

En la Figura 2–11(c) se puede apreciar el resultado de la extracción de características de textura, en este caso, las regiones sin cambios o lisas están de color negro, las horizontales en gris y las complejas en blanco. El algoritmo de segmentación combina los resultados de la composición de color y las características espaciales de la textura para obtener los segmentos de textura uniforme; esto se ilustra en la Figura 2–11(d).

Para cada uno de los segmentos obtenidos después de aplicar el algoritmo de Chen, es necesario obtener una serie de características. Los autores deciden utilizar un análisis local por segmento. En cada segmento aislado, se cuantifican las propiedades de combinación de color y las propiedades de textura. En lo que respecta a la combinación de color, se determinan el primer color dominante de la región, así como la distancia entre el segundo y el primer color dominante. Para describir una textura se usa un vector de 6 componentes que describe la orientación de la textura, que son: suavidad, horizontal, vertical, compleja,  $+45^\circ$  y  $-45^\circ$ . En la Figura 2–12 se ilustra este procedimiento.

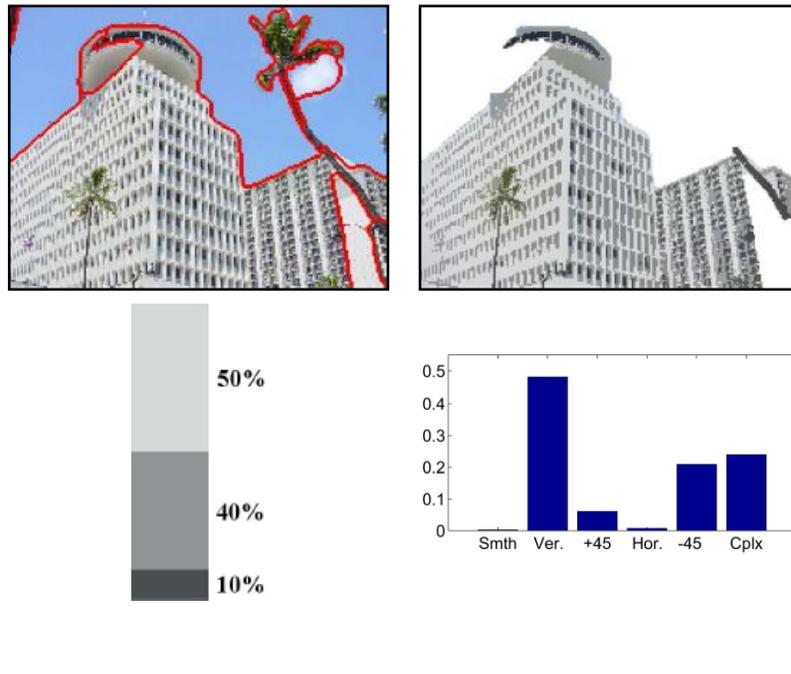


Figura 2–12. Cuantificación de características para cada segmento (a) Imagen segmentada; (b) Extracción del segmento de interés; (c) Composición de colores dominantes; (d) Orientación de la textura.

Para hacer el etiquetado semántico, se eligió un conjunto de palabras de otra publicación (Lin et al., 2003), se usan etiquetas para describir segmentos y otras para escenas en conjunto (Tabla 2–2). Las pruebas se realizaron sobre un conjunto de 2500 fotografías, la mayoría de las cuales fueron obtenidas de la biblioteca de fotografías de Corel, así como de un repositorio de imágenes de los investigadores. Al igual que como en otros trabajos relacionados, se parte de una etapa de entrenamiento donde se asigna una de las etiquetas de la base de conocimientos a cada segmento.

Tabla 2–2. Etiquetas para segmentos.

Natural			
Vegetación	Cielo	Forma del Terreno	Agua
Hierba Árboles/arbustos Bosque Flores	Cielo de día Cielo de noche Soleado Nubes Amanecer/Atardecer	Nieve Montaña/Loma Tierra Pavimentado/Camino	
Artificial			
Edificio/Casa Puente Carro Bote Avión Otro			
Humano			
Cara Persona Gente			

**Comentarios**

Este artículo se enfoca en tomar en consideración aspectos acerca de cómo los usuarios hacen una clasificación de una imagen dada. En vez de generar un listado enorme de etiquetas, se procede a tomar las más importantes o que con mayor frecuencia usan las personas; para ello reutiliza el resultado de otro trabajo de investigación, por lo que no es un resultado propio.

El trabajo se divide en tres grandes etapas: segmentación, extracción de características y asignación de etiquetas, pero la aportación de este trabajo no es de novedad debido a que el trabajo de Chen como algoritmo de segmentación, genera un conjunto de segmentos que a juzgar por las imágenes publicadas por estos autores, elimina grandes problemas que se tienen al trabajar con otros algoritmos de segmentación conocidos. Como principal inconveniente del algoritmo de Chen se puede observar, tomando como referencia la Figura 2–12, que en la parte inferior del edificio existe una palmera, pero el resultado final es un segmento que incluye tanto al edificio como a la palmera. Por lo anterior, podrían existir ciertos objetos que no fueran identificados empleando esta técnica de segmentación (sub-segmentación). Esto es relevante al tratar de determinar la semántica de una imagen, porque se podría omitir contenido semántico

importante, aunque fuera un objeto muy simple como una palmera, que para un usuario dado puede ser de gran significado.

En la sección de resultados se omiten datos como son la resolución espacial de las imágenes de prueba y tiempos estimados de procesamiento; como los autores indican que el algoritmo de segmentación es complejo, esto puede afectar seriamente los tiempos de consulta para un sistema práctico.

### 2.2.8 Segmentación, clasificación y marcas de agua para autenticación semántica de imágenes y video

**Segmentation, Classification and Watermarking for Image / Video Semantic Authentication**

**Ching – Yung Lin & Belle L. Tseng (2002)**

Estos autores abarcan el problema de la autenticación de información semántica, debido a que una alteración del contenido semántico de una imagen puede pasar inadvertida por un sistema de reconocimiento. Para comprender mejor lo anterior, consideremos las imágenes de la Figura 2–13. Se puede observar que la versión de la Figura 2–13b tiene el fondo modificado, pero desde el punto de vista semántico, si se buscara el objeto “Lena”, un sistema de validación nos diría que ambas imágenes son correctas, aunque la segunda versión no es completamente la misma.



a b

Figura 2–13. Imágenes con semántica diferente (a) Imagen original (b) Imagen con fondo alterado.

Por lo anterior, estos investigadores proponen una nueva metodología para la autenticación de una imagen o video a nivel semántico. Este método usa aprendizaje estadístico, segmentación

visual de objetos y esquemas de clasificación para comprender la semántica del contenido visual (Lin & Tseng, 2002). En la Figura 2–14 se muestra un panorama general de la arquitectura propuesta. El sistema consiste principalmente de dos etapas: una en que se agrega una marca de agua y otra en que se agrega una marca de autenticación. En el proceso de agregar una marca de agua, el objetivo es embeber una marca de agua que incluya información de los modelos, es decir, de los objetos que están presentes en un clip del video o en la imagen, en donde para decidir los objetos que están presentes, se puede usar el resultado de la segmentación y la clasificación automática, o usar una anotación semiautomática. Desarrollaron un sistema de nombre VideoAnnEx que usa tres tipos de etiquetas: fondo del escenario, objetos y eventos.

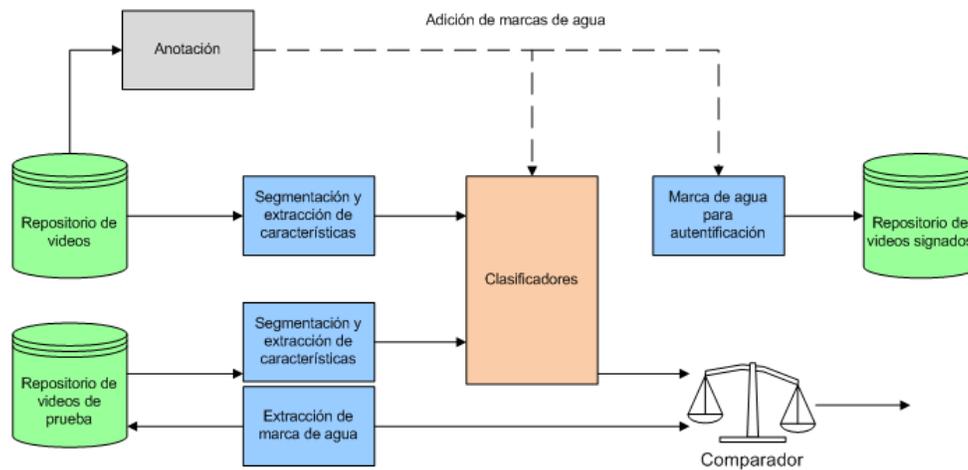


Figura 2–14. Esquema general de la aplicación propuesta.

También es necesaria una etapa de entrenamiento para modelar conceptos. Para modelar numéricamente la semántica de los objetos, se emplea el Modelo Mixto Gaussiano (*Gaussian Mixture Model – GMM*) o una Máquina de Soporte Vectorial (*Support Vector Machine*). Debido a que la capacidad del clasificador se limita a los modelos que son especificados a partir del entrenamiento del sistema; en algunos casos es necesario etiquetar manualmente a los objetos que no fueron reconocidos, es decir, hacer un reentrenamiento del sistema. Con lo anterior, el clasificador puede aprender nuevos modelos o modificar los ya existentes en su base de conocimientos.

Empleando un programa de IBM (CueVideo), cuando se trabaja con archivos de video se determinan cuadros clave. Para cada cuadro clave, se determinan características de color, textura,

estructura y forma. El color se representa usando histogramas lineales HSV y momentos de color. La estructura se determina calculando histogramas de la dirección de bordes y para la textura se usan matrices de ocurrencia en niveles de gris. Finalmente para la forma se usan los momentos invariantes de Dudani (**Dudani et al., 1977**).

Para segmentar el fondo de la escena, se usa un método de crecimiento de regiones basado en bloques tomando como referencia dos cuadros clave, denotados como I y P. El criterio del algoritmo se basa en los histogramas de color y de esquinas, así como en el índice de dirección de las texturas de Tamura (**Tamura & Yamawaki, 1978**). Por otra parte, para encontrar los objetos de primer plano se calculan los vectores de movimiento sobre los cuadros I y P, los cuales se usan para determinar los objetos con crecimiento de regiones en el dominio espacial junto con restricciones de seguimiento en el dominio del tiempo.

Con los resultados anteriores, se agrega una marca de agua de los modelos en la imagen original. Se hace uso de un algoritmo de marcas de agua (*watermarking*) propuesto por Lin (**Ling & Chang, 2001**), el cual es invariante a la rotación, escalamiento y traslación para añadir información. El proceso de autenticación se ejecuta comparando el resultado de la clasificación con la información proveniente de la marca de agua.

### **Comentarios**

En esta propuesta los autores abordan un problema significativo, que consiste en mantener detalles de bajo nivel al trabajar con la semántica de una imagen. Dichos detalles permiten que al comparar o autenticar imágenes, la semántica no sea el único criterio tomado en cuenta. Sin embargo, los investigadores recurren a técnicas desarrolladas por otros científicos, por lo que su aportación no es muy original. Al igual que en otras publicaciones, los autores no ilustran claramente el tipo de videos o imágenes de prueba empleados con el sistema desarrollado, los tiempos de procesamiento, etc. Adicionalmente, para agregar nuevos modelos a la base de conocimientos, se necesita realizar un proceso manual.

Existen ciertos detalles significativos de esta propuesta que no son explicados, por ejemplo, al trabajar con videos digitales, se indica que se determinan cuadros clave para hacer la

segmentación y extracción de características, pero desafortunadamente no se indica si el flujo de video debe tener una longitud mínima, su resolución espacial, un número de cuadros clave como mínimo, entre otros aspectos.

El esquema de segmentación propuesto se simplifica al trabajar con pares de cuadros clave, ya que a partir de estos cuadros se separa el fondo de la escena de los objetos del primer plano; para ello clasifican como fondo de la escena toda aquella región de información que se mantiene “constante” en los dos cuadros clave, pero como se toman cuadros separados por un cierto intervalo de tiempo, se puede dar la situación de que haya un movimiento inicial de un objeto de la escena, pero que regrese a su posición inicial en el segundo cuadro clave, por lo que para el sistema este objeto es estático, y en realidad no lo es. Este caso patológico no es abordado en esta investigación.

Otro punto relevante es que los autores indican que la metodología sirve para trabajar tanto con flujo de video como con imágenes estáticas; sin embargo, en la sección de pruebas se limitan a trabajar con una serie de videos únicamente.

### **2.2.9 Morfología matemática en espacios de color aplicada al análisis de imágenes cartográficas**

#### **Mathematical Morphology in Color Spaces Applied to the Analysis of Cartographic Images**

**Jesús Angulo & Jean Serra (2003)**

Desde el punto de vista del procesamiento de imágenes, el contenido de un mapa cartográfico de color, está compuesto típicamente de regiones de color así como de pequeñas estructuras, tales como textos, símbolos, líneas, etc. En consecuencia, tanto las características geométricas como el color son información relevante para extraer los objetos semánticos espaciales, donde dichas características pueden extraerse manualmente o mediante un sistema de cómputo (Angulo & Serra, 2003).

Estos autores identifican dos etapas principales: la segmentación de la imagen, para definir las regiones de color y para extraer los detalles de gráficos y texto, y por otra parte, el reconocimiento de los caracteres empleando Reconocimiento Óptico de Caracteres (*Optical Character*

*Recognition – OCR*), identificación de símbolos, indexación de colores, etc. Como aportación de esta publicación, se presenta un método para realizar un análisis completo de los mapas cartográficos por medio de operadores morfológicos. La propuesta estudia el uso de elementos de color de un mapa cartográfico para realizar una partición de las jerarquías de la imagen en regiones homogéneas de color para extraer una capa binaria de los detalles geométricos y de color.

Citan que el trabajar con el espacio de color RGB tiene varias desventajas, entre las que se encuentran: las componentes se encuentran correlacionadas, pérdida de interpretación visual por parte del usuario, información no uniforme, etc. Trabajo como el de Hanbury (Hanbury 2001), demuestra que resulta ventajoso trabajar con otros espacios como el HLS, HSV, por mencionar algunos. Por lo anterior, los autores deciden trabajar con una variante del espacio HLS formulada por ellos mismos.

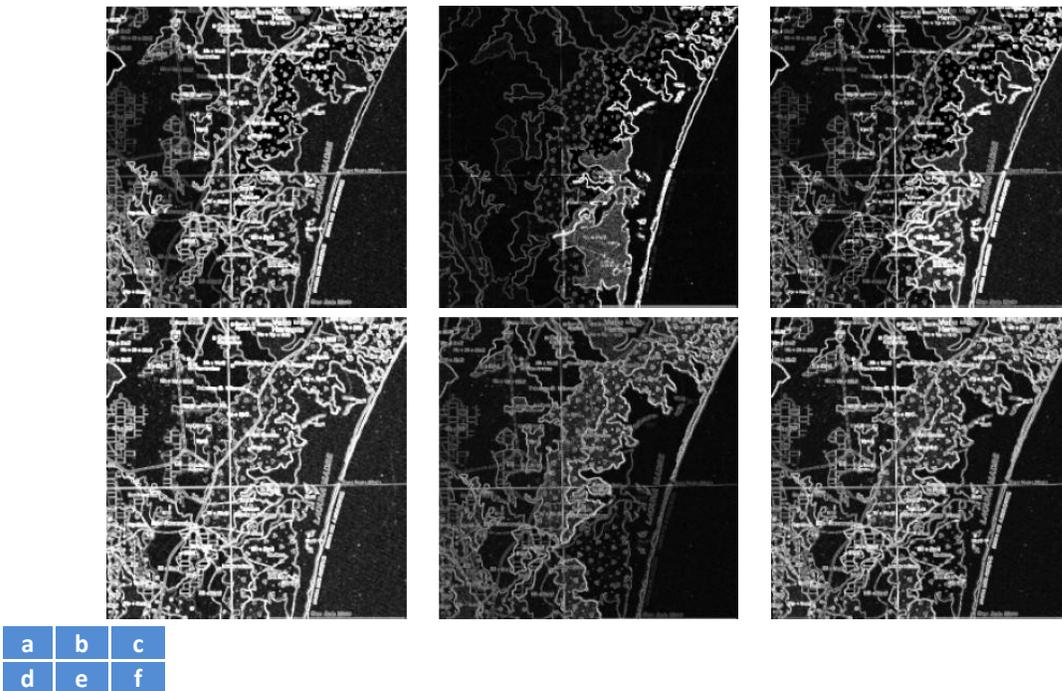


Figura 2–15. Ejemplos de diferentes tipos de gradiente de color (a) luminancia (b) tonalidad (c) saturación ponderada (d) supremo (e) cromático (f) perceptual.

Definen formalmente los operadores morfológicos de gradiente y *top-hat* para el caso de imágenes de color. Para el caso de gradiente consideran seis variantes: de luminancia, tonalidad, de saturación ponderada, supremo, cromático y perceptual. En la Figura 2–15 se ilustran ejemplos

de estos gradientes. La transformación *top-hat* es un operador de gran utilidad que permite detectar objetos contrastantes claros sobre fondos no uniformes. Desde la perspectiva de Meyer (Meyer, 1977) existen dos versiones de este operador; el residuo entre una función numérica y la apertura o clausura.

Al igual que para el caso de gradientes, definen tres variantes de dicho operador: *top-hat* blanco acromático, *top-hat* negro acromático y *top-hat* cromático. Las diferencias entre estas definiciones se puede observar visualmente en la Figura 2–16.

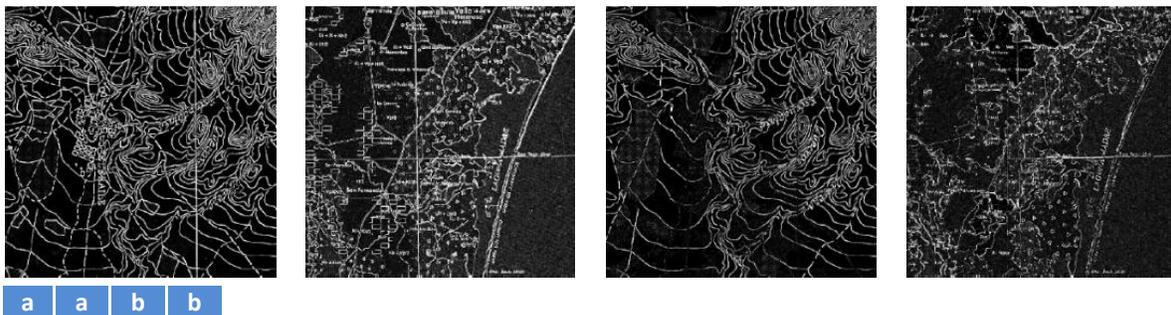


Figura 2–16. Ejemplos de dos versiones del operador *top-hat* aplicados a las imágenes de la Figura 2–15(a) y (b) respectivamente; (a) Aplicando el operador *top-hat* negro acromático; (b) Resultado al aplicar el operador *top-hat* cromático.

La arquitectura del sistema propuesto por estos autores se muestra en la Figura 2–17. A continuación se hará una breve descripción de cada etapa.

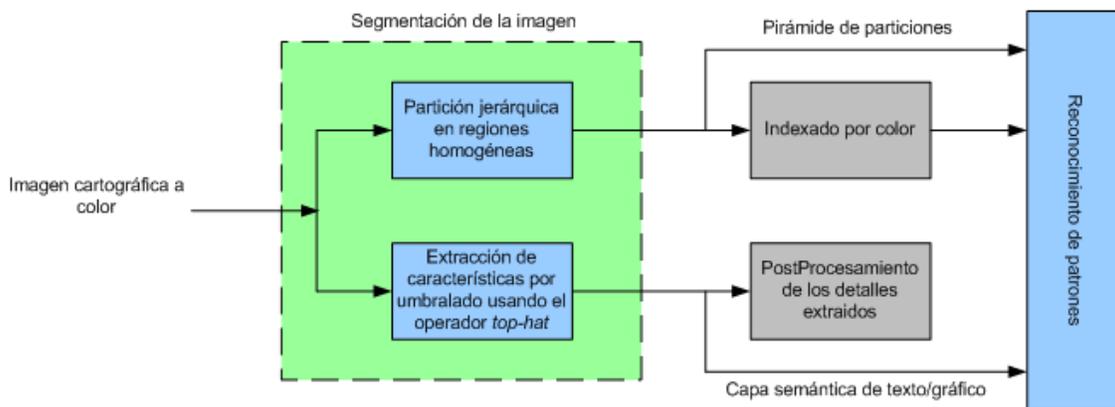


Figura 2–17. Arquitectura del algoritmo desarrollado.

La partición jerárquica en regiones homogéneas tiene el propósito de segmentar la imagen en regiones disjuntas, donde el contenido de cada una de ellas es homogéneo respecto al color,

textura, etc. Con el propósito de tener resultados más flexibles, proponen usar segmentación multiescala, es decir, aquella en que la partición está compuesta por una pirámide jerárquica donde los niveles sucesivos están más simplificados que los anteriores. Estos mismos autores desarrollaron dos algoritmos para hacer este tipo de segmentación, el primero está basado en la técnica de vertientes de agua (*watersheds*), mientras que el segundo se basa en la fusión de particiones cromáticas-acromáticas ordenadas por la componente de saturación.

Básicamente la extracción de características involucra calcular los *top-hats* de color para obtener los detalles de texto y gráficos. El principal problema se presenta en el umbralado de los *top-hats*, ya que si se usa directamente un algoritmo para determinar el umbral, se pueden obtener resultados con alta presencia de ruido. Por ello definen un operador de apertura de área, el cual es un filtro conectado que elimina las estructuras de brillo cuya área es menor que un umbral de tamaño preestablecido.

En la etapa de indexado de color y posprocesamiento se determinan dos histogramas bivariantes, uno de ellos combinando las componentes de matiz y saturación, y el otro combinando la luminancia y la saturación. Estos histogramas pueden ser considerados como índices para un sistema de recuperación basado en el contenido de color. Dentro del posprocesamiento se pueden aplicar operadores morfológicos como el de adelgazamiento para poder separar los datos de texto y gráficos, como se ilustra en la Figura 2-18.

### Comentarios

Su aportación consiste en el uso de varios operadores que funcionan con el modelo de tonalidad, saturación, luminancia (*Hue, Saturation, Lightness – HSL*): el gradiente de color, operador *top-hat* y apertura de área. Con las variantes de cada operador, se pueden resaltar o minimizar características específicas de las imágenes raster en donde se aplica. Desafortunadamente, no se plantea detalladamente un algoritmo que permita decidir de manera automática o semi-automática los operadores que se deben usar para una imagen dada.

Aunque se usan un espacio no lineal así como operadores diferentes a los tradicionales, el esquema de la aplicación es muy similar a las ya conocidas dentro del área de reconocimiento de

patrones: realizar una segmentación e identificación de características relevantes de texto y objetos geográficos que permitan hacer una identificación del contenido de la imagen.

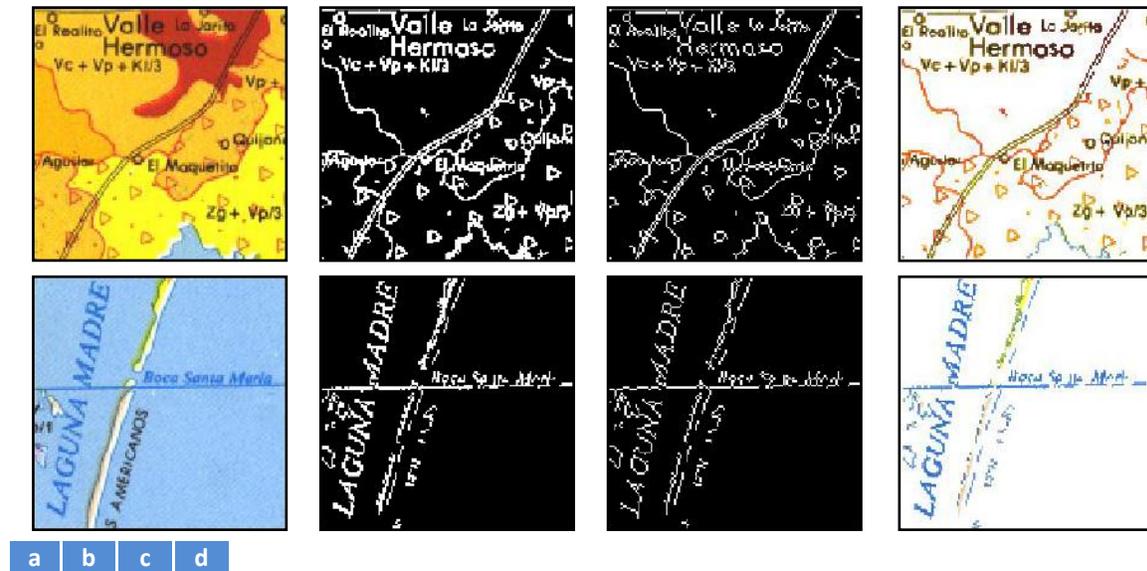


Figura 2–18. Ejemplos de posprocesamiento (a)Imagen de color original; (b) Capa obtenida de texto/gráfico; (c) Adelgazamiento morfológico; (d) Capa de texto/gráfico combinada con la imagen de color original.

La principal diferencia de este trabajo de investigación con otros mencionados, es el uso de un espacio de color no lineal reemplazando el espacio lineal RGB. Los autores citan otras publicaciones en las cuales se justifica que el usar un espacio no lineal proporciona mejores resultados a los que se han logrado con los métodos tradicionales que operan sobre el espacio RGB. Sin embargo, para poder justificar claramente dicho argumento, hubiera sido conveniente mostrar una comparación entre imágenes finales en el espacio RGB y el HSL.

### 2.2.10 Segmentación rápida de imágenes basada en análisis de multi-resolución y ondeletas

Fast image segmentation based on multi-resolution analysis and wavelets

Kim Byung-Gyu, Shim Jae-Ick, Park Dong-Jo (2003)

La segmentación de una imagen juega un papel importante en la visión por computadora y el procesamiento de imágenes para interpretar y analizar una imagen capturada. El umbralado (*thresholding*) basado en la distribución de una característica, se usa ampliamente en la

segmentación en las múltiples regiones de una imagen de entrada, por su simplicidad y capacidad de poder ser implementado en procesos de tiempo real. Se han desarrollado diversas técnicas de umbralado, la mayoría de las cuales requieren de la optimización de una determinada función criterio.

Por ello, los autores proponen un algoritmo para determinar el fondo del valle del histograma que separe de la manera más óptima los dos grupos de segmentos del histograma (modas) **(Byung-Gyu et al., 2003)**. Las técnicas de umbralado se clasifican en paramétricas y no paramétricas; su diferencia consiste en que en las técnicas paramétricas se asume que existe un modelo con la misma función de densidad de probabilidad (FDP) que la de la distribución de niveles de gris de cada grupo, la que deberá tener un comportamiento de tipo Gaussiano. La desventaja es que los métodos paramétricos no son útiles cuando la diferencia entre el modelo supuesto y la distribución real es significativa. El algoritmo propuesto por Kittler e Illingworth es uno de los más empleados **(Kittler & Illingworth, 1986)**.

Por otra parte, las técnicas no paramétricas están enfocadas en separar las distribuciones de niveles de gris en una forma óptima, acorde a un criterio que puede ser: la varianza de cada clase, la varianza entre clases, varianza total, entropía, etc. **(Kittler & Illingworth, 1986)**. Como ejemplos de estas técnicas están los métodos de Otsu **(Otsu, 1979)** y Calvard **(Calvard, 1978)**. La mayoría de estos métodos buscan los valores óptimos de umbralado en base a un criterio en el espacio de características<sup>6</sup>. Generalmente el tiempo de búsqueda del valor óptimo se incrementa si la dimensión del espacio de características también se incrementa y viceversa. En consecuencia, los autores desarrollaron una técnica que permite reducir el espacio original de búsqueda.

La metodología propuesta por estos autores se basa en el análisis multi-resolución (*MRA-Multi-resolution Analysis*), el cual es ampliamente usado para la representación y procesamiento de señales, debido a que permite describir señales a menores resoluciones y espacios escalados. En el análisis multi-resolución una señal puede ser vista en varios niveles de aproximaciones o resoluciones. Aplicando MRA, se puede dividir una señal compleja en múltiples señales más simples, y cada señal puede ser tratada independientemente. La transformada *wavelets* es una

---

<sup>6</sup> Dentro de los criterios se pueden tomar parámetros de intensidad, color y textura, entre otros.

herramienta especialmente útil para la aproximación, compresión de datos, así como para la eliminación de ruido.

Esta metodología define el nivel de gris como una característica y sea  $h(k)$  la distribución original<sup>7</sup> de una imagen dada. Los pasos del algoritmo son los siguientes:

1. Con la distribución original  $h(k)$ , se aplica la transformada de ondeletas con un nivel  $m$  por medio de una operación de decimación por  $2^m$ , después de calcular la convolución de la distribución original  $h(k)$  con  $\phi(t)$ .

$$h^m(l) = WT^m[h(k)], m \in Z = h_A^m(l) + h_W^m(l) \quad (2.1)$$

Donde  $h_A^m(l)$  es la señal de tendencia (*trend signal*) de la distribución original y  $h_W^m(l)$  representa los detalles de la distribución, todo esto en el  $m$ -ésimo nivel. Para cualquier señal de tendencia en el nivel  $m$ , ésta se descompone sucesivamente en dos señales de menor dimensión en el nivel  $m + 1$ .

$$h_A^m(l) = h_A^{m+1}(l) + h_W^{m+1}(l) \quad (2.2)$$

2. Usando la señal  $h_A^m(l)$  de la resolución  $m$ , los valores adecuados de umbralado,  $\theta_i^m$  ( $i = 1, \dots, L$ ) se determinan por medio de una función discriminante, por ejemplo, la separabilidad de Otsu, la discrepancia basada en entropía, u otro criterio. Por otra parte,  $L$  denota el número de umbrales deseados.
3. Se expanden los valores  $\theta_i^m$  seleccionados en el nivel  $m$  al espacio original  $h(k)$ . En este algoritmo desarrollado, se proponen dos modos de expansión, la directa y la de refinamiento después de una expansión directa.
  - a. En el modo de expansión directa, el umbral final  $\theta_i$  se obtiene aplicando un factor de  $2^m$ .

$$\theta_i = \hat{\theta}_i = \theta_i^m \times 2^m, \quad \text{para } i = 1, \dots, L \quad (2.3)$$

<sup>7</sup> La distribución es el histograma de la imagen original.

- b. El procedimiento de refinamiento del umbral, consiste en obtener primeramente el umbral empleando la Ecuación (2.3) y después aplicando para el  $i$ -ésimo umbral expandido  $\theta_i$  la Ecuación (2.4).

$$\theta_i = \max_{\theta \in [\hat{\theta}_i - \theta_{var}, \hat{\theta}_i + \theta_{var}]} F(\theta) \quad (2.4)$$

Donde  $F(\theta)$  es la función discriminante empleada para la segmentación y  $\theta_{var}$  es el rango dinámico para el procedimiento de refinamiento. Para los propósitos de este artículo se propuso el valor de 4 para  $\theta_{var}$ . Adicionalmente, se usa la separabilidad de Otsu como función discriminante  $F(\theta)$ .

4. La imagen original es particionada en múltiples segmentos usando los valores óptimos  $\theta_i$  encontrados en el paso anterior.

En la Figura 2–19 se muestra la comparación aplicando el método de Otsu directamente en la imagen original, así como aplicando la transformada *wavelet* de nivel 3 usando el método de expansión directa; los valores entre paréntesis denotan los umbrales empleados. Para realizar comparaciones con otros algoritmos, los autores proponen usar la función de bondad (*goodness function*) ( $G$ ) (Liu & Yang, 1994). Esta función está definida como:

$$G(I) = \sqrt{M} \times \sum_{i=1}^M \frac{e_i^2}{\sqrt{A_i}} \quad (2.5)$$

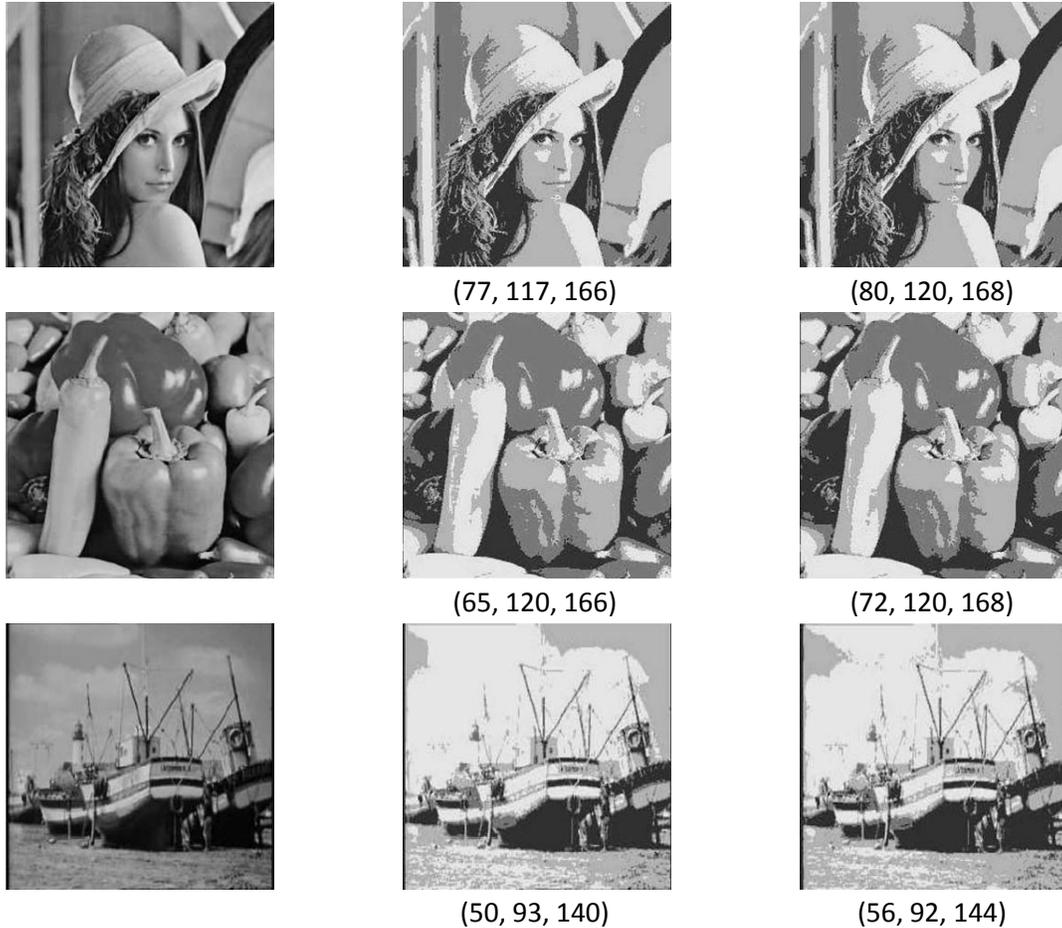
Donde:

- $M$  Es el número de regiones en la imagen segmentada.
- $A_i$  Es el número de píxeles en la  $i$ -ésima región.
- $e_i$  Es la suma de la distancia Euclidiana de los vectores de color o intensidades de la imagen original y la imagen segmentada en la  $i$ -ésima región.

### Comentarios

Algunos algoritmos de segmentación pueden producir muy buenos resultados pero al costo de una gran cantidad de cálculos, es decir, un elevado tiempo de cómputo o procesamiento. Con la

metodología propuesta por estos autores, se hace una reducción de la cantidad de muestras de la imagen digital al aplicar la transformada de ondeletas. Usando el método de Otsu se determina un conjunto de umbrales para poder hacer un umbralado de la imagen original.



a	b	c
d	e	f
g	h	i

Figura 2-19. Resultado de la metodología propuesta. (a)(d)(g) Imágenes originales; (b)(e)(h) Resultado aplicando el método de Otsu con 3 umbralados en las imágenes originales; (c)(f)(i) Resultado aplicando el algoritmo propuesto y expansión directa con una transformada de nivel 3.

Esta técnica es de utilidad cuando las intensidades de las clases de los objetos no se traslapan, es decir, que para cada objeto o región de la escena los niveles de gris que lo describen se encuentren en un rango específico y que no sea utilizado por otros objetos. Sin embargo, en escenarios reales complejos esta situación difícilmente se cumple, principalmente en objetos geoespaciales.

Como aspectos positivos de este trabajo, los autores hacen uso de una ecuación matemática (función de bondad) para tratar de cuantificar numéricamente la calidad de las imágenes obtenidas; lo anterior no es tomado en cuenta por las publicaciones citadas previamente. De igual forma, se define una técnica que permite ajustar los umbrales determinados en el histograma de una imagen reducida con la transformada de ondeletas.

Para poder observar completamente el funcionamiento de esta técnica, hace falta agregar un algoritmo de segmentación que permita extraer los objetos de la imagen original.

## 2.3 Metodologías orientadas a objetos

A diferencia de las técnicas revisadas previamente, estas metodologías toman en cuenta información adicional para poder hacer la descomposición y clasificación de las imágenes a procesar, definiendo criterios que permiten aislar la información como **objetos** sin tratar la información como datos crudos.

### 2.3.1 Técnica de segmentación basada en borde y región para la extracción de objetos artificiales grandes en una colección de imágenes satelitales de alta resolución

**Edge and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery**

**Marina Mueller, Karl Segl & Hermann Kaufmann (2004)**

El análisis automático de imágenes de desastres naturales siempre ha sido una tarea desafiante. Esto es aplicable sobre todo para el análisis de datos de fuentes remotas, donde por ejemplo, las características de los sensores y las condiciones variantes atmosféricas pueden influir sobre la calidad de la imagen. Debido a que algunas aplicaciones en los Sistemas de Información Geográfica (SIG), como son el monitoreo de coberturas de área, o la evaluación de daños después de desastres naturales, dependen de la detección precisa de los límites de campos agrícolas. Solo recientemente, los sensores de satélites como IKONOS o Quick-Bird pueden ofrecer imágenes pancromáticas con una resolución de 1 ó 0.7 metros respectivamente (**Mueller et al., 2004**).

Por lo anterior, estos autores desarrollaron una metodología de segmentación orientada a objetos para la extracción de objetos largos, artificiales, especialmente campos agrícolas, de imágenes satelitales pancromáticas, basada en la combinación de técnicas de bordes y crecimiento de regiones. La justificación para este algoritmo, es que cuando se trata de detectar ciertas regiones extensas, como los campo de cultivo, en imágenes de alta resolución (ver Figura 2–20a), existen tres dificultades:

- Los pequeños detalles pueden causar grandes variaciones en las intensidades de los niveles de gris dentro de los objetos que están siendo procesados, con lo cual se originan muchas clases (problema conocido como sobre segmentación).
- Bajo contraste de los objetos en los límites de las regiones, lo cual podría impedir la detección de dichas regiones y en consecuencia tener una sub segmentación.
- Que las regiones tengan formas geométricas complejas con bordes rectilíneos extensos y que no sean necesariamente de forma rectangular, lo cual impide el uso de modelos geométricos fijos.

Lo anterior afecta seriamente el desempeño de las técnicas de segmentación clásicas como son los algoritmos ISO-DATA, el crecimiento de regiones y el vertiente de aguas, proporcionando resultados distintos a los que se obtendrían mediante una segmentación manual, como se muestra en la Figura 2–20. La técnica desarrollada en esta publicación combina los métodos de bordes y uno basado en regiones.

El método propuesto está integrado por dos etapas: (1) extracción de los modelos de bordes y (2) segmentación guiada por bordes, basada en regiones. La extracción de los modelos de bordes, consiste en tres pasos:

- Se extraen a múltiples escalas los bordes candidatos para realizar una detección de todos los modelos de bordes importantes.

- Los bordes en cada escala son evaluados acorde al modelo de borde dado y se seleccionan únicamente los bordes que satisfacen completamente el modelo empleado.

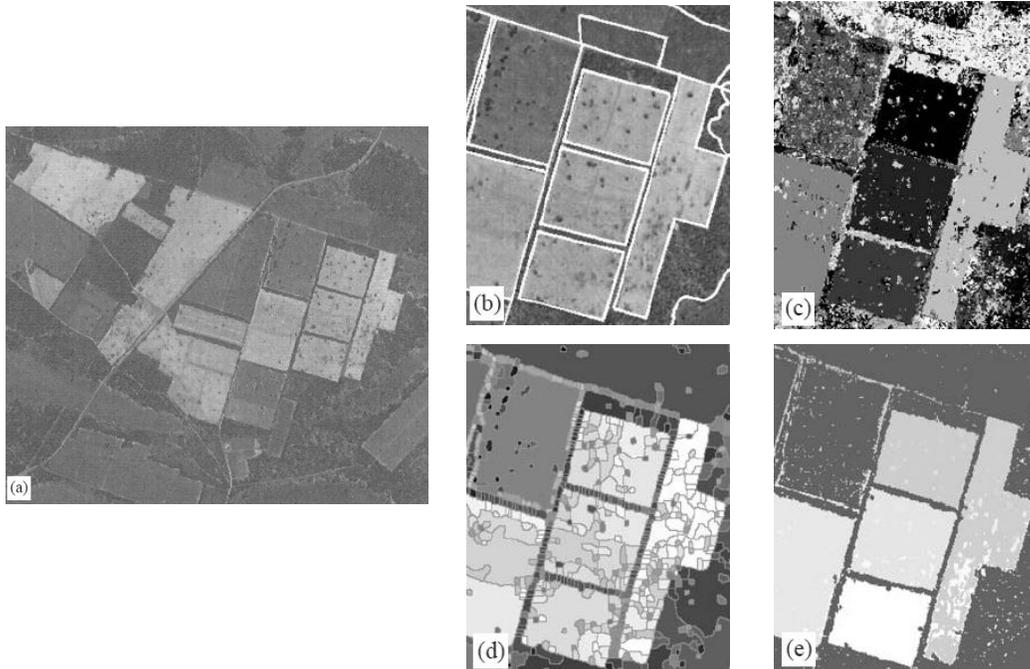


Figura 2–20. Resultados de diferentes algoritmos de segmentación (a) Imagen original; (b) Segmentación manual; (c) ISO-DATA; (d) Algoritmo *watershed*; (e) Crecimiento de regiones.

- El tercer y último paso consiste de lo siguiente:
  - Eliminar los bordes parásitos, determinando la longitud de cada uno de ellos y empleando un umbral de longitud mínima.
  - Empleando una técnica de seguimiento de líneas, se dividen los bordes en segmentos de líneas más pequeños para un análisis, hasta determinar las semillas de los bordes. Una semilla es un segmento de borde que ya no puede ser dividido en segmentos más pequeños. Las semillas pueden ser clasificadas en:
    - Fuertes. Son aquellas que poseen un contraste mínimo.
    - Rectas. Las cuales tienen un ángulo mayor al de umbral preestablecido.
    - Largas. Que tienen una gran longitud.
  - Se hace un proceso de alargamiento de los bordes, tomando como parámetros las semillas y su vecindad.

- Se realiza otro proceso de podado de los bordes cuya longitud es inferior a la de un umbral dado.

El resumen de estos pasos se puede observar en la Figura 2–21

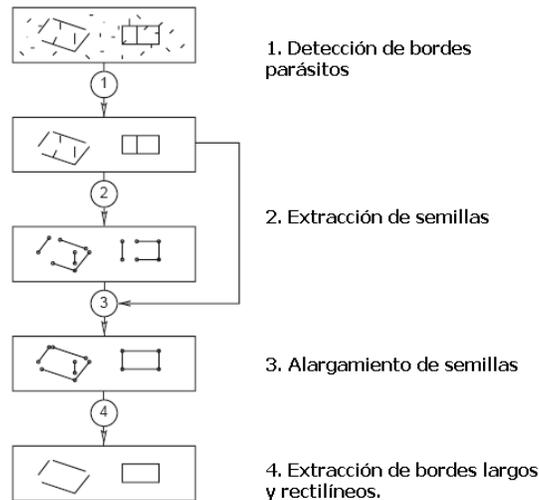


Figura 2–21. Etapa de extracción de los modelos de bordes.

La etapa de segmentación guiada por bordes basada en regiones, está dividida en tres pasos independientes:

- Se hace un preprocesamiento de suavizado (*smoothing*) de la imagen, empleando un filtro mediana con máscara de 3 x 3. La variante a considerar es que de los 8 vecinos de un píxel  $P$ , solamente se toman en cuenta los que se encuentran del mismo lado de  $P$ .
- Se realiza un crecimiento de regiones guiado por bordes. El algoritmo tradicional de crecimiento de regiones considera únicamente las condiciones de homogeneidad tales como la distancia entre las intensidades de nivel de gris y el valor medio.

En este artículo los autores plantean las siguientes dos condiciones de borde:

- Un píxel  $P$  no pertenece a la región  $R$ , si  $P$  se encuentra en el lado contrario a un borde, o
- $P$  se encuentra dentro de una región abierta reducida, formada por una región que está creciendo.

La primera condición previene que una región se extienda sobre los límites de los bordes, mientras que la segunda permite cerrar los boquetes que se podrían formar en los bordes. En la Figura 2–21 se muestran los resultados de esta técnica, comparando los resultados con el algoritmo de corrimiento de media, así como el sistema eCognition (**Ecognition, 2007**) en una versión de prueba (con el sistema eCognition tuvieron que hacer uso de más de 20 parámetros para obtener el resultado correspondiente).

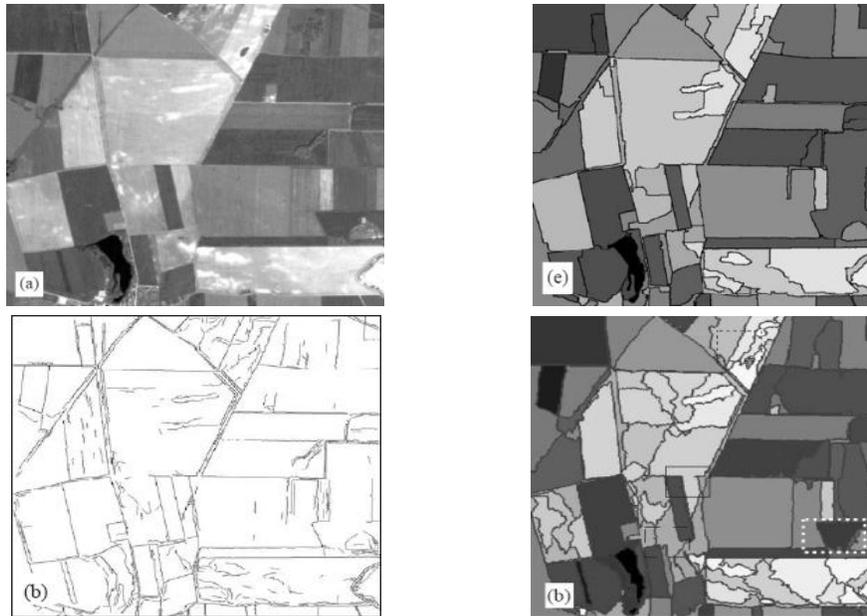


Figura 2–22. Resultados y comparaciones (a) Imagen original; (b) Resultado aplicando el algoritmo propuesto; (c) Resultado aplicando el algoritmo de corrimiento de media a la imagen de la Figura 2.20a; (d) Resultado usando el sistema eCognition.

### Comentarios

Esta metodología persigue una extracción de regiones orientada a objetos, enfocándose a objetos geospaciales grandes. Se basa en emplear algoritmos clásicos del PDI como es la técnica de crecimiento de regiones.

Desafortunadamente, la técnica para extracción de bordes hace un ajuste de los resultados obtenidos mediante técnicas tradicionales, considerando que los objetos son de dimensiones grandes, por lo que los pequeños bordes pueden ser parásitos, o bien, que haya un rompimiento

en la continuidad de un borde del contorno de un objeto geográfico. Este tipo de problema se puede resolver haciendo uso del algoritmo de Deriche-Cocquerez.

Aunque se obtienen mejores resultados a los obtenidos con el sistema eCognition, se puede observar en las imágenes de la Figura 2–22 que existen regiones cuya intensidad no es homogénea y el resultado del algoritmo genera regiones diferentes, cuando debería generarse una sola región para describir este objeto geográfico.

### **2.3.2 Análisis mental-semántico e integración de datos orientados a objetos de información visual. Una primera versión**

**Semantic-Mind Analysis and Objected-Oriented Data Integration of Visual Information: A Primer**

**Serguei Levachkine & Victor Alexandrov (2003)**

Este trabajo describe la forma de extraer componentes semánticos de un conjunto no ordenado de datos. Los autores presentan un método orientado a objetos, que como ya se explicó en la Introducción, permite preservar las características esenciales de una geoimagen, como son: información contextual y formas de los objetos, entre otros. Su metodología consiste de dos etapas fundamentales, denotadas como Análisis y Síntesis. La primera etapa tiene el objetivo de estructurar jerárquicamente el contenido de la información a procesar. Dos componentes básicos del Análisis son los siguientes:

- Un algoritmo de detección de segmentos con colores o brillos similares.
- Un mecanismo de indexado secundario para los segmentos detectados.

Se basa en el empleo de estructuras de datos dinámicas adaptativas, que usa un conjunto inicial de semillas (segmentos de imagen) que son las regiones de color o de niveles de gris presentes en la imagen original. Esta estructura contiene características relevantes de la región, como son: color, ubicación, geometría, topología (**Levachkine & Alexandrov, 2003**).

A partir de este conjunto de regiones, se realiza un proceso iterativo de fusión entre regiones poco diferentes. Es en esta parte donde se controla el nivel semántico (llamado así por los autores), ya que a mayor cantidad de iteraciones, se obtendrán menos regiones, y en consecuencia, menos

objetos para su reconocimiento en la etapa de síntesis. Al finalizar esta etapa se tiene una red semántica interrelacionada de segmentos de imagen. En la Figura 2–23 se pueden observar los resultados de la etapa de Análisis.



Figura 2–23. Resultados de la etapa de análisis con diferentes niveles semánticos; (a) Imagen Axe; (b) Imagen Plait.

En la etapa de síntesis, primeramente se tiene que analizar la estructura jerárquica de la etapa anterior, para en caso de ser necesario, hacer una reestructuración de las relaciones de la red semántica. En base a las características determinadas de cada segmento de imagen, se identifica si el segmento corresponde a una vía de comunicación, un elemento hidrológico, etc.

El modelo propuesto lo aplican a una herramienta de conversión de imágenes raster a información vectorial, conocida como A2R2V (*Analog to Raster to Vector*), el cual es un sistema completo que permite identificar los objetos geográficos y elementos de texto presentes en un mapa cartográfico. En la Figura 2–24 se muestran algunos ejemplos de la etapa de Análisis del sistema A2R2V.

### Comentarios

Este artículo presenta un método orientado a objetos que permite hacer una descomposición semántica de una imagen. Las dos etapas propuestas (Análisis y Síntesis) permiten la intervención

del usuario para poder controlar el nivel semántico de descomposición, a diferencia de otros autores donde el resultado no puede ser manipulado por el usuario.

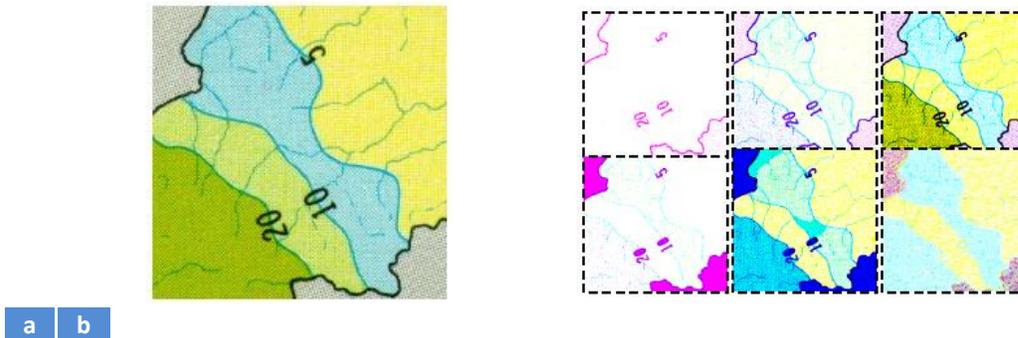


Figura 2–24. Etapa de Análisis en el sistema A2R2V; (a) Imagen original; (b) Inspección de la imagen.

Es importante el uso de una estructura de datos jerárquica la cual contiene atributos cuantitativos de los segmentos de imagen (ubicación, color, geometría, topología). Esta información puede ser proporcionada al usuario, y no limitarse únicamente a indicar el conjunto de objetos identificados.

Con base en lo expuesto por los autores, el sistema desarrollado presenta un alto grado de eficacia para el análisis de imágenes cartográficas. Sin embargo, no existe una comparación real con otros sistemas que realizan tareas similares.

Este artículo servirá como base para nuestro trabajo, ya que se adoptará un esquema general en cuanto a usar una etapa de análisis y otra de síntesis. Sin embargo, la diferencia radica en el dominio del problema, ya que A2R2V está enfocado a trabajar con mapas cartográficos en formato raster, mientras que en nuestro caso se pretende usar geoimágenes de fuentes remotas (p.e. en imágenes satelitales).

### 2.3.3 Herramienta comercial eCognition

Este programa de cómputo se basa en el concepto de que la información semántica necesaria para la interpretación de una imagen no reside en un solo píxel, sino en objetos (conjunto de píxeles) y las relaciones existentes entre ellos. A partir de este concepto, eCognition ofrece una herramienta cuya tecnología orientada a objetos y de análisis multi escala permite elaborar clasificaciones de

imágenes más precisas y eficientes que los métodos tradicionales de clasificación (técnicas orientadas a píxeles) (Definiens, 2007).

A continuación se describirán algunas características de esta herramienta. En la Figura 2–25 se puede observar la pantalla principal de la versión 5.0, que está disponible en el sitio Web de la empresa propietaria de esta herramienta.

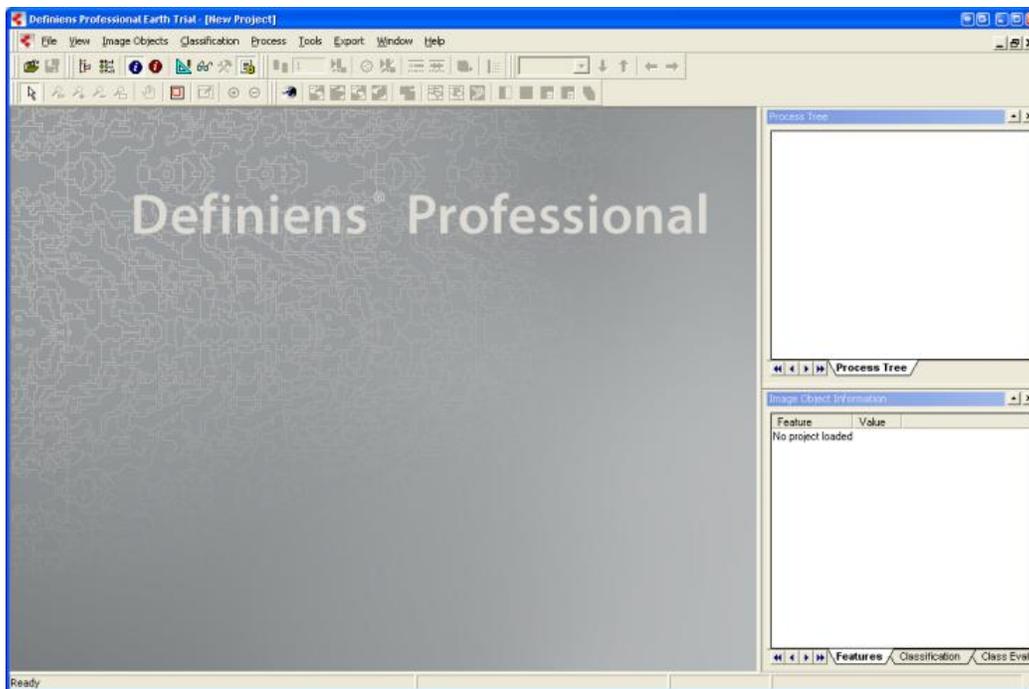


Figura 2–25. Pantalla principal de eCognition.

El programa eCognition proporciona un lenguaje artificial para desarrollar algoritmos avanzados de análisis de imágenes. Estos algoritmos usan los principios del análisis de imágenes orientado a objetos y el procesamiento local adaptativo. Lo anterior se logra, definiendo una serie de procesos conocidos como conjunto de reglas dentro de un árbol de procesos. Un proceso sencillo es la unidad elemental de un conjunto de reglas.

Un proceso sencillo permite aplicar un algoritmo específico a una región de interés en la imagen. Todas las condiciones para una clasificación, así como la selección de una región de interés, deben incorporar información semántica. Los procesos pueden tener un número arbitrario de procesos hijos (Definiens, 2007).

En la Figura 2–26 se puede apreciar el cuadro de diálogo que permite definir un nuevo proceso. En este ejemplo, se está seleccionando un algoritmo de segmentación de la biblioteca de algoritmos incorporados en esta herramienta.

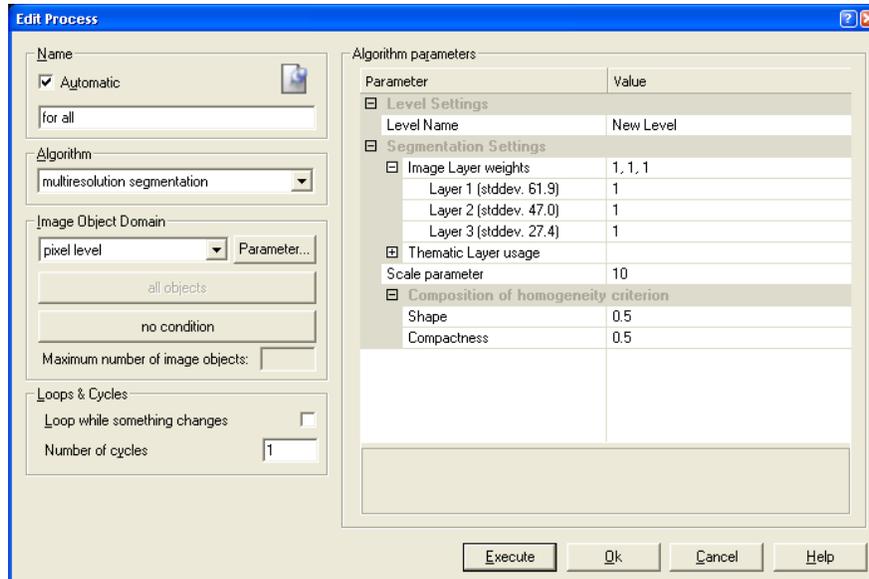


Figura 2–26. Definición de un nuevo algoritmo en eCognition.

## Comentarios

Un punto favorable de esta herramienta de cómputo es que no existe otro sistema comercial que logre hacer análisis orientado a objetos de imágenes geográficas. El control de los algoritmos de análisis de imágenes reside en proporcionar una gran cantidad de parámetros, lo que implica conocimiento muy especializado del usuario para poder obtener resultados adecuados.

En la sección de pruebas se podrá analizar una comparación de los resultados que se obtienen usando este programa con la metodología de descomposición semántica. Desafortunadamente, no fue posible hacer muchos experimentos con esta herramienta, dado que solamente se tuvo acceso a la versión de demostración de eCognition, la cual está limitada a usar una sola imagen de ejemplo adjunta al programa.

## 2.4 Enfoque conceptual

Acorde con Martínez Trinidad ([Martínez, 2001](#)) en muchas ciencias aplicadas está presente el problema de revelar la estructura subyacente en una colección de objetos. Esta información de los objetos se encuentra típicamente almacenada en archivos planos, bases de datos o algún otro medio electrónico en forma estructurada. El problema de clasificar o estructurar esta información ha sido intensamente estudiado en el reconocimiento de patrones no supervisado. Los métodos desarrollados en esta área, forman agrupamientos sobre la base de parejas de objetos muy parecidos (o próximos) e ignoran la utilidad del significado de los agrupamientos. Las técnicas tradicionales del reconocimiento de patrones no supervisado tienen la desventaja de formar agrupamientos los cuales no tienen una interpretación conceptual.

Para Michalski el agrupamiento conceptual consiste en agrupar objetos en clases conceptualmente simples basándose en los valores de sus atributos, tomando en consideración el conocimiento acerca de las relaciones semánticas entre los atributos de los objetos o conceptos globales que puedan ser usados para caracterizar las agrupaciones de objetos ([Michalski, 1983](#)) ([Michalski, 1986](#)).

En este enfoque, las clases (agrupaciones de entidades) se generan formulando primero descripciones conceptuales de estas clases y clasificando entonces las entidades de acuerdo con las descripciones.

### 2.4.1 En busca del concepto perdido

**Béjar Javier (1993)**

Este autor comenta que para una clasificación y agrupación de observaciones se puede hacer uso de los métodos estadísticos para el análisis de datos, algunos de los cuales se basan en la construcción de conjuntos de objetos mediante algoritmos de agrupación, a partir de sus descripciones y utilizando medidas de distancia entre ellos, definidas a partir de una métrica. El principal inconveniente es que tienden a construir agrupaciones de objetos puramente sintácticas, ajustadas a una serie de criterios independientes del contexto en el que se engloban los datos ([Béjar, 1993](#)).

Utilizando las ideas del agrupamiento conceptual este investigador implementó la herramienta LINNEO<sup>+</sup>. Esta técnica se centra en dominios en los que hay una débil estructuración, debida al poco consenso que existe entre los expertos. LINNEO<sup>+</sup> parte de descripciones de observaciones mediante pares atributo-valor, una medida de similitud y un criterio de agregación y construye un conjunto de clases, que son susceptibles de ser modificadas mediante la utilización de relaciones entre los atributos de las descripciones, para conseguir una mejora de los conceptos obtenidos, ver Figura 2–27.

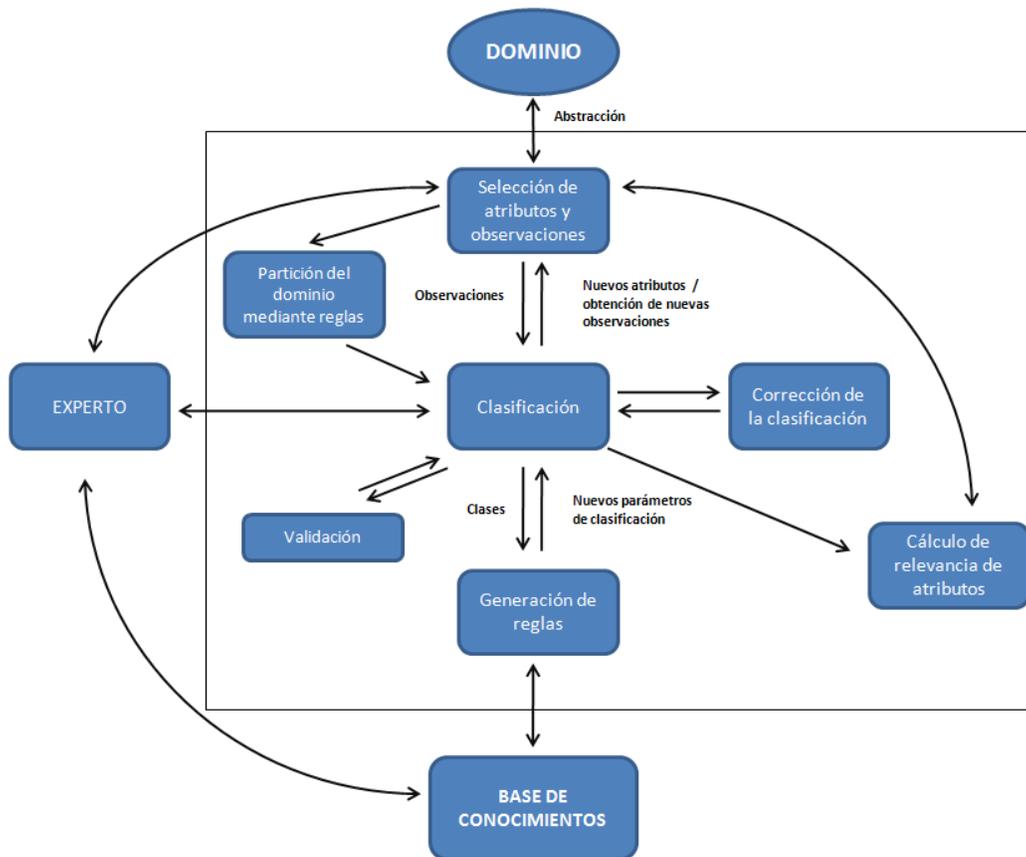


Figura 2–27. Elementos de la metodología de LINNEO<sup>+</sup>.

### Comentarios

Las técnicas de agrupamiento tienen la desventaja de que para aplicaciones donde las características cuantificadas no presentan una diferencia muy marcada, se pueden presentar traslapes en los centroides de los vectores característicos que modelan las clases; adicionalmente, se puede dar el caso de que no todas las características puedan ser medidas para todos los objetos

(dominios poco estructurados). Esto dificulta de manera significativa la definición de tales vectores prototipo. A lo largo de varias décadas, diversos investigadores han tratado de perfeccionar y extender los trabajos de Michalski acerca del agrupamiento conceptual, tal es el caso de la herramienta LINNEO<sup>+</sup>.

Uno de los puntos a destacar de esta propuesta es que trata de modelar el conocimiento de un grupo de expertos, lo que permite enriquecer una base de conocimientos a partir de diversas fuentes. Sin embargo, la situación se complica significativamente cuando al generar una descripción de una clase, dos expertos difieren en la descripción. Considerando el caso de un experto del dominio geográfico, para él la forma en que se interpreta una geoimagen que contiene yacimientos minerales, puede ser diferente a la de otro experto; aunque posiblemente uno de los dos sea el único que tiene la verdad en sus afirmaciones, esto implica una comprobación adicional para poder hacer una discriminación en las suposiciones que recibe de un grupo de expertos, es decir, tener la precaución en la generación del agrupamiento conceptual<sup>8</sup>. Lo anterior es similar a lo que ocurre en las técnicas tradicionales de agrupamiento, cuando un objeto no es tan representativo de una clase, lo que genera que el centroide de un vector prototipo tenga un considerable desplazamiento.

En consecuencia, el trabajo que implica modelar este tipo de conocimiento puede ser equivalente al que se requiere en refinar una técnica de agrupamiento, hasta el grado en que los vectores prototipo de las clases permitan obtener los resultados requeridos en la solución de un problema dado, aunque en el caso de dominios poco estructurados es más recomendable emplear el enfoque que se propone en el agrupamiento conceptual. Por otra parte, los resultados expuestos en esta investigación, no permiten apreciar correctamente el funcionamiento de la herramienta propuesta.

Sin embargo, existe similitud entre el objetivo de esta investigación con la idea propuesta por Michalski, en el aspecto de que se persigue modelar las clases considerando no únicamente las características cuantitativas de cada objeto geográfico, sino también se contemplan las propiedades geométricas y topológicas, es decir, el contexto de los objetos geográficos. Asimismo

---

<sup>8</sup> En la publicación, se comenta que la forma de agregar las descripciones puede influir en el resultado o agrupamiento final.

la formulación de las descripciones de las clases de los objetos geográficos se plasma en una ontología de aplicación del dominio geográfico, por medio de clases, relaciones y restricciones.

## 2.5 Comentarios y observaciones

De lo revisado en las publicaciones expuestas, podemos concluir que queda mucho trabajo por hacer en lo que se refiere a la descomposición semántica de una geoimagen. Como puntos desfavorables de estas propuestas tenemos:

1. En la mayoría de los casos se emplea un solo algoritmo de segmentación o particionamiento de la imagen original.
2. El entrenamiento es *a priori* y no se puede realizar *a posteriori*, o bien, el sistema parte de un conjunto muy reducido de información en su base de conocimientos.
3. Son propuestas en su mayor parte orientadas a píxeles, con lo que únicamente se obtienen etiquetas de lo reconocido en la imagen.
4. No se obtienen datos para su interpretación por el usuario, por ejemplo: topología, geometría, entre otros.
5. No definen claramente lo que es la semántica de una imagen, o específicamente, la interpretación que dan de la semántica.
6. Posiblemente por el espacio para la publicación, se omiten aspectos fundamentales como es la de una etapa de preprocesamiento o posprocesamiento.
7. No existe una forma de comprobar los resultados, ya que no hay una disponibilidad de los algoritmos desarrollados por los investigadores, lo cual nos limita a confiar en que los resultados obtenidos son los reportados en sus respectivas publicaciones. En consecuencia, no

se puede hacer una tabla comparativa tratando de usar el mismo conjunto de imágenes con todas las metodologías presentadas.

## **Capítulo 3**

### **Marco teórico**

#### ***Resumen***

En este capítulo se plasman los fundamentos teóricos de las técnicas y algoritmos que fueron tomados como base para la metodología que será presentada en el capítulo siguiente. El contenido de este capítulo está enfocado al procesamiento digital de imágenes y las ontologías.



## 3.1 Introducción

En este capítulo se estudia el marco teórico de los algoritmos usados en esta tesis. Nuestro marco teórico se enfoca en estudiar primeramente tópicos fundamentales de la percepción remota; posteriormente del procesamiento digital de imágenes y el reconocimiento de patrones, en donde se analizarán algoritmos de preprocesamiento, segmentación, reconocimiento y clasificación. Asimismo se realizará un estudio de las ontologías, tema base para nuestro trabajo de investigación. La mayor parte de los fundamentos matemáticos de esta sección fueron recopilados usando información de cuatro autores ([Pratt, 2001](#)), ([Gonzalez & Woods, 2002](#)), ([Phillips, 2000](#)) y ([Jähne, 2002](#)).

## 3.2 Percepción remota

La percepción remota (*Remote Sensing*) o Teledetección, puede definirse como la ciencia y arte de obtener información de un objeto mediante el análisis de los datos adquiridos con la ayuda de algún dispositivo que no está en contacto físico con dicho objeto. A pequeña escala, una persona al leer un libro, está ejecutando un acto de percepción remota. Su sistema sensorial asignado a la visión está en primera instancia identificando letras, luego agrupándolas en palabras y más tarde en oraciones, para finalmente, sumando estas oraciones, obtener un párrafo. Hasta este punto ha realizado una **interpretación física** de lo que hay escrito. Pero ¿cuál es el **significado** de lo que hay escrito? **interpretando** el texto ya **analizado**, el lector es capaz de detectar un **contexto**: ha transferido a su cerebro la información intrínseca de ese texto. En un segundo análisis de carácter más reflexivo, su cerebro procesa la información recién ingresada, cotejándola con la ya existente en su memoria, lo que le permite sacar sus propias conclusiones acerca de lo que ha leído. En resumen, de lo que en un principio eran un conjunto de caracteres, surge una idea. Obviamente, este proceso (además de ser sensiblemente más complejo) se lleva a cabo a una velocidad tal, que afortunadamente nos es imposible concientizar cada uno de estos pasos, pero básicamente esto es lo que sucede cuando leemos ([TeleDet, 2007](#)).

El gran salto en esta disciplina se produjo en la década de 1960 cuando las plataformas satelitales reemplazaron a las aéreas y surgieron los sensores electrónicos multispectrales. Los objetos

terrestres, iluminados por la radiación solar, reflejan ésta luego de introducir en ella modificaciones inducidas por la misma estructura y composición de dichos objetos. La radiación reflejada es capturada por los sensores del satélite, la que es parcialmente procesada a bordo de éste y retransmitida a estaciones receptoras terrestres para su posterior procesamiento y análisis (fuente emisora, sensor, cerebro).

En la percepción remota existen dos características de la radiación electro-magnética que son de interés, a saber: longitud de onda y frecuencia. El espectro electromagnético cubre desde longitudes de onda cortas (rayos X y gamma) a longitudes de onda largas (microondas y ondas de señales de radio). Algunos materiales de la superficie de la Tierra, principalmente rocas y minerales, emiten luz visible o fluorescente cuando son iluminados por medio de los rayos ultravioleta. La zona en que nuestros ojos pueden percibir en forma de luz las radiaciones dentro del espectro electromagnético, es conocida como espectro visible.

Debido a que los seres humanos solamente podemos percibir una porción muy pequeña del espectro, se han desarrollado dispositivos para “percibir” otras regiones. Las señales visibles cubren longitudes de onda aproximadamente entre los 0.4 y 0.7  $\mu\text{m}$ . La longitud de onda más corta del espectro visible corresponde al violeta. De los colores que se perciben con mayor frecuencia, se encuentran los siguientes: violeta: 0.4 – 0.446  $\mu\text{m}$ , azul: 0.446 – 0.500  $\mu\text{m}$ , verde: 0.500 – 0.578  $\mu\text{m}$ , amarillo: 0.578 – 0.592  $\mu\text{m}$ , naranja: 0.592 – 0.620  $\mu\text{m}$ , rojo: 0.620 – 0.7  $\mu\text{m}$ . Aunque se percibe la luz del sol con un color uniforme, ésta se encuentra compuesta de radiación de varias longitudes de onda, principalmente de regiones del espectro ultravioleta, infrarrojo y del espectro visible.

La siguiente región del espectro electromagnético que resulta de interés es la infrarroja (IR), la cual cubre las radiaciones de longitudes de onda entre los 0.7  $\mu\text{m}$  y 100  $\mu\text{m}$ ; puede ser dividida en dos regiones por sus propiedades de radiación: infrarrojo reflejado e infrarrojo emitido o térmico. El IR reflejado es usado en la percepción remota en formas muy similares a la radiación del espectro visible. Cubre longitudes de onda entre los 0.7  $\mu\text{m}$  a 3.0  $\mu\text{m}$ .

Por otra parte, la región del infrarrojo térmico es un poco diferente a las regiones del espectro visible y esencialmente es la radiación que la superficie de la Tierra emite en forma de calor. Cubre

longitudes de onda entre los 3.0  $\mu\text{m}$  a 100  $\mu\text{m}$ . Es la porción del espectro que recientemente se ha comenzado a usar en la percepción remota. Las microondas tienen longitudes de onda entre 1 nm y 1m. Es la región de longitudes de onda más largas usadas dentro de la percepción remota.

### 3.2.1 Tipos de imágenes satelitales

El origen de las imágenes satelitales es muy diverso, pero existen dos programas desarrollados por agencias espaciales internacionales que han colocado muchos satélites artificiales en órbita con la Tierra y cuyas imágenes son usadas por muchos grupos de investigación. Dichos grupos son los proyectos de los satélites LANDSAT y SPOT.

#### 3.2.1.1 Imágenes Landsat

Son todas aquellas que tienen como fuente de origen alguno de los satélites del programa que lleva el mismo nombre. El programa Landsat tiene la misión de desarrollar y colocar satélites en órbita y es llevado a cabo por la Administración Nacional de Aeronáutica y del Espacio (*National Aeronautics and Space Administration* - NASA). Este proyecto está integrado por siete satélites (Figura 3–1), el primero de ellos lanzado el 23 de Julio de 1972 y el más reciente, el Landsat 7, el día 15 de Abril de 1999. Solamente hubo un fallo en la puesta en órbita del Landsat 6, ya que este satélite no pudo alcanzar la velocidad requerida para mantenerse en órbita.

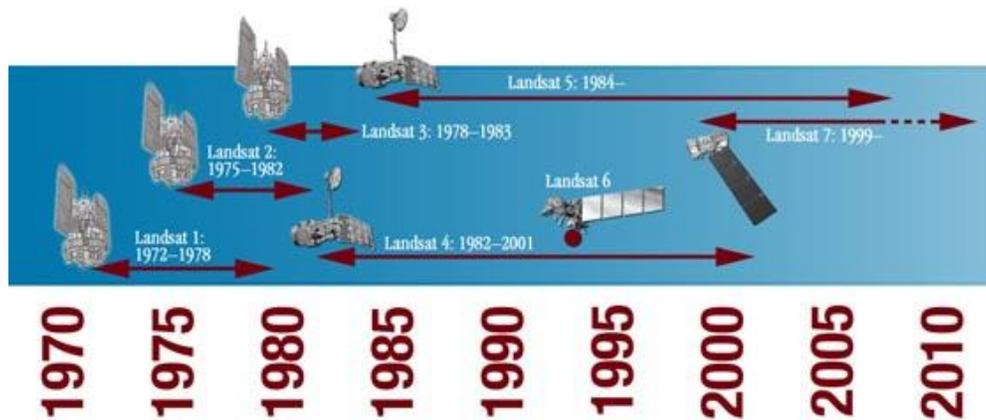


Figura 3–1. Línea del tiempo de los satélites Landsat. Fuente: NASA (NASA, 2007).

Durante 34 años, el programa LANDSAT ha recolectado información espectral de la superficie terrestre, creando un archivo histórico que ningún otro programa satelital ha podido igualar en calidad, detalle, cobertura y tamaño (NASA, 2007).

### 3.2.1.2 Imágenes SPOT

SPOT (Satélite para la observación de la Tierra o *Satellite Pour l'Observation de la Terre*) es un sistema óptico de imágenes de alta resolución que opera en el espacio. Este programa es parte de la estrategia de observación de la Tierra del Centro Nacional de Estudios Espaciales Francés (*Centre National D'Etudes Spatiales* - CNES). Los participantes en el programa SPOT son el CNES, los servicios científicos, técnicos y culturales de Bélgica (*Belgian scientific, technical and cultural services* - SSTC) así como la Agencia Espacial Nacional de Suecia (*Swedish National Space Board* – SNSB). Está integrado por cinco satélites, SPOT 1 – 1986 / 2003; SPOT 2 – 1990; SPOT 3 – 1993 / 1996; SPOT 4 – 1998 y SPOT 5 – 2002 de los cuales 3 siguen en operación.

Desde el lanzamiento del primer satélite SPOT en 1986, esta familia de satélites ha observado nuestro planeta y proporcionado imágenes de gran calidad. De estos satélites se tiene aproximadamente una colección de 10 millones de imágenes. De acuerdo con el CNES, los satélites SPOT son los satélites civiles de su generación con mejor desempeño, lo que ha permitido que se conviertan en el estándar de imágenes satelitales (CNES, 2007).

## 3.2.2 Visualización de imágenes satelitales

La imagen adquirida por un sensor remoto comúnmente está en formato digital. Dicha imagen es una matriz con filas y columnas, donde cada elemento es llamado píxel (*picture element*). Por cada una de las bandas existirá una de estas matrices. Al terminar una fila de escaneo usualmente se incorporan parámetros de localización geográfica y calibración del sensor. La cantidad y tipo de esta información dependerá del formato de imagen. Además, al comienzo de la imagen suelen ir los encabezados que le indicarán a la computadora el formato de la imagen, cantidad de líneas, columnas y la cantidad de bits que ocupa cada píxel.

La cantidad de bits usada por el píxel determinará finalmente cuanta información puede llevar este píxel. Un píxel de 1 bit, sólo puede tener 2 niveles de información "encendido" o "apagado", uno de 2 bits tendrá 4 niveles y uno de  $n$  bits tendrá  $2^n$  niveles de información. En percepción remota lo más usual es encontrar pixeles entre 8 y 10 bits. Una imagen de 8 bits es capaz de almacenar 256 niveles de información; por ejemplo si la imagen es en escala de grises, serán 256 niveles o tonos de gris (Chuvieco, 1990).

Las diferentes bandas espectrales captadas por los sensores de un satélite pueden combinarse para generar imágenes coloreadas que mejoran la visualización de determinadas escenas.

### 3.2.2.1 *Imágenes en falso color*

El ojo humano puede ver muchos más detalles en las imágenes en color que en blanco y negro. Incluso se pueden apreciar más detalles en una imagen de falso color que en otra equivalente en color natural o verdadero.

En el procesamiento de imágenes de satélite, a menudo se generan imágenes en falsos colores porque incrementan la percepción de determinados detalles de la superficie. Una imagen en falso color es una representación artificial de una imagen multispectral.

En algunas aplicaciones de teledetección, puede ser útil asociar las clases de cobertura del suelo con colores familiares, por ejemplo, el pasto con color verde. En otros casos, se prefieren los colores contrastantes para resaltar objetos de interés en el fondo.

La codificación que se emplea en las imágenes multibanda, se basa en la numeración de las bandas que integran dicha imagen, siguiendo el orden específico de rojo, verde y azul. De esta manera, una imagen que use la banda 7 para el rojo, la banda 4 para el verde, y la banda 2 para el azul se la designa como: 7, 4, 2.

### 3.2.2.2 Imágenes en pseudocolor

A diferencia de las imágenes en falso color que están compuestas por la información de tres bandas diferentes, las imágenes en pseudocolor consisten en imágenes monocromas en las que se ha substituido la escala de grises por una tabla o paleta de colores. En la Figura 3–2 se muestra una imagen original y otra versión usando esta técnica.

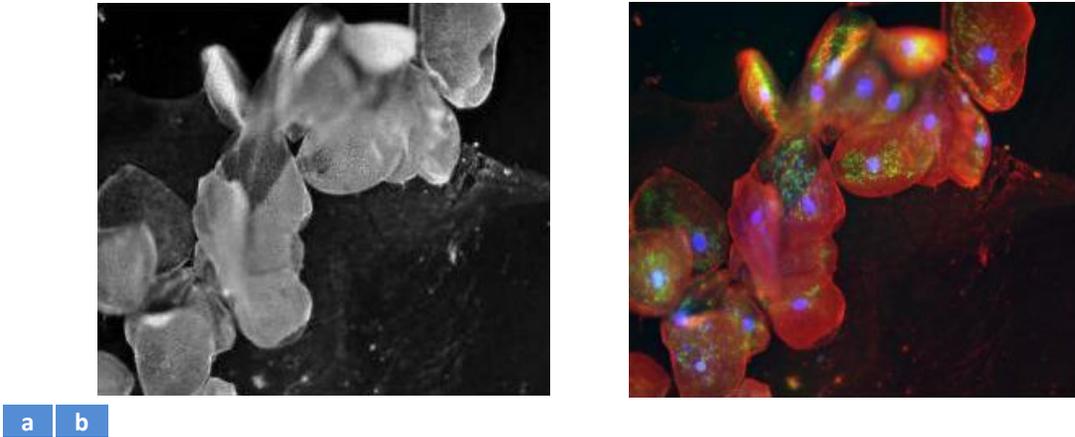


Figura 3–2. Ejemplo de la técnica de pseudocolor; (a) Imagen original; (b) Imagen en pseudocolor.

### 3.2.2.3 Imágenes en color verdadero

Reciben este nombre, todas aquellas imágenes donde el color de cada elemento discreto corresponde con el color real de los objetos capturados, como ejemplo de este tipo de imagen se puede observar la Figura 3–3.



Figura 3–3. Imagen en color verdadero.

### 3.2.3 Corrección de imágenes satelitales

Los sensores remotos actuales han permitido incrementar sustancialmente la resolución espacial de las geoimágenes. Por ejemplo, los primeros satélites lanzados en 1960 (el primero fue el TIROS-1) permitían capturar imágenes únicamente en escala de grises y con muy poca resolución (1.1 km), ver Figura 3–4(a) (Wikipedia, 2007), mientras los más recientes como IKONOS y Quick-Bird pueden ofrecer imágenes pancromáticas con una resolución de 1 a 0.7 metros respectivamente, Figura 3–4(b). La resolución original de esta geoimagen es de 4313 x 2553 píxeles (Space Imaging, 2002).

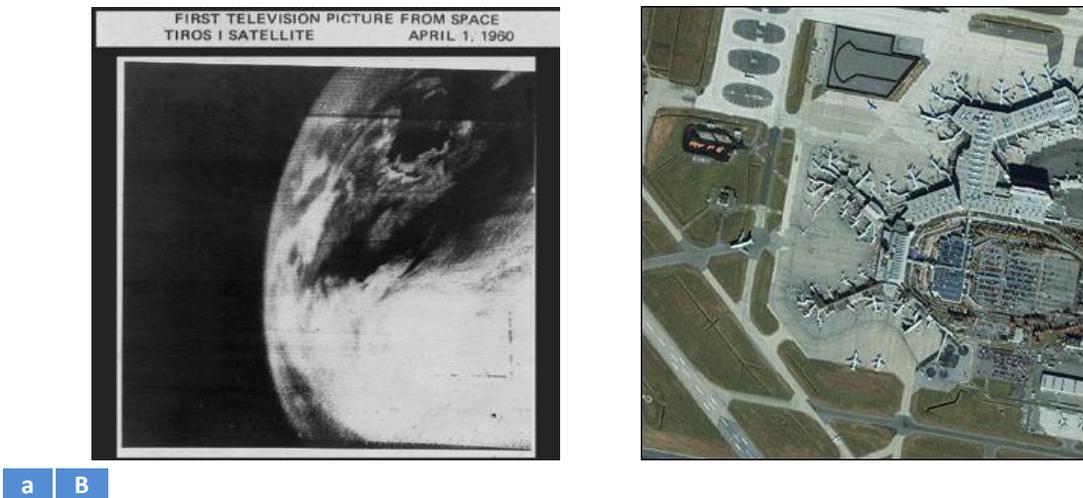


Figura 3–4. Comparación entre geoimágenes (a) Satélite TIROS-1 (b) Satélite IKONOS.

Sin embargo, antes de procesar una imagen satelital, puede ser necesario aplicar diferentes tipos de correcciones que son descritas a continuación.

#### 3.2.3.1 Correcciones radiométricas

La energía electromagnética puede sufrir alteraciones antes de llegar a la superficie de la Tierra, lo cual puede ocasionar que esta energía no incida completamente en el área de estudio y en consecuencia el proceso de captura por parte de los sensores remotos no sea el ideal, fenómeno conocido comúnmente como perturbación atmosférica. Las alteraciones involucradas son: dispersión (*scattering*) y/o absorción (CCRS, 2007). La dispersión ocurre cuando las partículas o moléculas grandes de gas que están presentes en la atmósfera, interactúan con la radiación y

ocasionan que sea desviada de su camino original, como se muestra esquemáticamente en la Figura 3–5(a). Por otra parte, la absorción se presenta cuando la radiación electromagnética interactúa con la atmósfera. En contraste con la dispersión, este fenómeno causa que las moléculas en la atmósfera absorban energía en diferentes longitudes de onda. Las tres causantes principales de este fenómeno son el monóxido de carbono, el ozono y el vapor de agua, ver Figura 3–5(b).

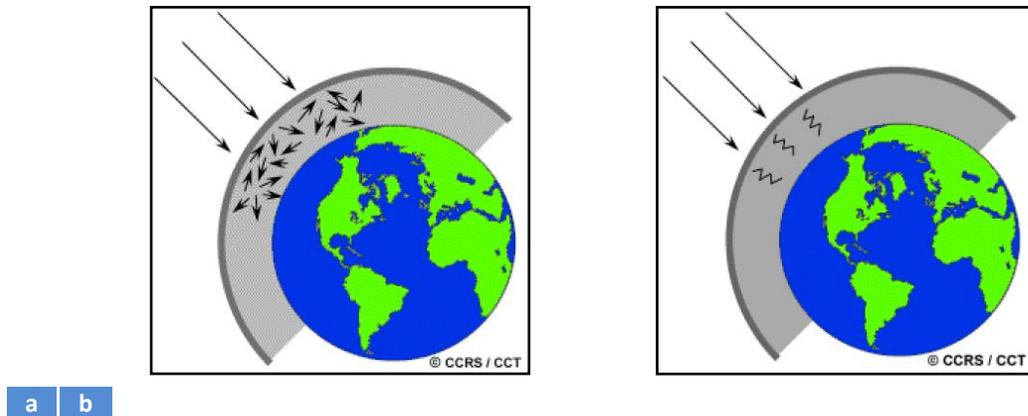


Figura 3–5. Fenómenos que afectan la radiación electromagnética (a) Dispersión (b) Absorción.

La corrección por la atenuación atmosférica es especialmente importante para objetos oscuros, tales como los cuerpos de agua. Existe una serie de técnicas para remover el efecto de la atmósfera. Las más complejas modelan matemáticamente las condiciones de la atmósfera en el momento y lugar en que la imagen fue capturada, lo que a menudo requiere de la toma de información en el terreno.

### 3.2.3.2 Correcciones geométricas

La curvatura de la Tierra, la no uniformidad del movimiento del sensor remoto y la no linealidad del espejo que captura una región geográfica producen distorsión. La corrección de dichas distorsiones geométricas puede ser lograda de dos formas:

- La posición de los píxeles se cambia, pero éstos mantienen sus valores de intensidad. Debido a que este método es de limitada precisión se usa sólo para correcciones geométricas simples.

- Realizar un nuevo muestro de la imagen. Una imagen digital geoméricamente distorsionada es transformada en una imagen sobre una rejilla igualmente espaciada, no distorsionada. La intensidad de los pixeles en la nueva grilla debe ser determinada por interpolación, usando los pixeles vecinos en la imagen distorsionada.

### 3.2.3.3 Mejoramiento de la imagen

Los métodos de mejoramiento de imagen son aplicados a una imagen que ya ha sido corregida radiométrica y geoméricamente. El mejoramiento está diseñado para ayudar al usuario de la información geoespacial para extraer e interpretar la información pictográfica. Esto se logra al enfatizar las características o patrones de interés; por ejemplo, el ojo humano puede identificar muchos más colores que tonos de gris, por lo que el uso de una paleta de colores puede representar mucha más información que una en tonos de gris.

Desafortunadamente, el preprocesamiento es un proceso subjetivo, es decir, no es posible aplicar modelos matemáticos o probabilísticos para determinar los algoritmos que se deben emplear sobre la imagen original. Generalmente, el usuario decide las técnicas más adecuadas a aplicar basadas en conocimientos *a priori*.

## 3.3 Procesamiento digital de imágenes

El procesamiento digital de imágenes (PDI) es un área de la computación, que durante los últimos 50 años ha realizado un gran aporte de algoritmos y técnicas que permiten mejorar, restaurar, descomponer, segmentar, clasificar, interpretar y describir imágenes digitales. En esta sección se describirán algunos tópicos fundamentales del PDI, con el objeto de definir los algoritmos que serán usados en la metodología propuesta y que se detallan en el capítulo siguiente.

Aunque el término formal adoptado en este documento es el de una geoimagen, cuya definición formal se describe más adelante, dicho término es una expresión que permite denotar con el mismo nombre a diversos tipos de imágenes geográficas digitales. Por lo anterior, los algoritmos

definidos en esta área pueden ser aplicados sin ningún inconveniente en las geoimágenes empleadas en nuestra metodología.

**Definición 3–1.** Una **imagen digital** es una versión discreta de una imagen continua, que se puede obtener en un instante de tiempo sobre un área rectangular  $-J \leq j \leq J, -K \leq k \leq K$  representada como una matriz de tamaño  $M \times N$ :

$$\mathbf{F} = [F(x, y)] = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,N} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M,1} & f_{M,2} & \cdots & f_{M,N} \end{bmatrix} \quad (3.1)$$

Para  $1 \leq x \leq M$  y  $1 \leq y \leq N$ . Comúnmente cada elemento de la matriz es conocido como píxel (que proviene de las palabras *picture element*).

**Definición 3–2.** Una **geoimagen** es una imagen digital que describe o representa objetos de la superficie terrestre (objetos geográficos), como puede ser una imagen remota, imagen cartográfica, una imagen en falso color de un modelo digital de elevación (*Digital Elevation Model – DEM*)<sup>9</sup>, imagen LANDSAT, SPOT, etc.

Con respecto a esta definición, es importante señalar que un DEM es una matriz de valores, donde cada valor describe la elevación de la región muestreada; mientras que la imagen de un DEM, es una imagen digital que se genera empleando alguna técnica, como la de falso color, que asigna una intensidad de color a cada muestra de elevación descrita por el DEM. En consecuencia, al generar una imagen de un DEM, se pierde la información de altura quedando únicamente información de color.

### 3.3.1 Preprocesamiento

Los métodos de preprocesamiento son clasificados en dos categorías: métodos en el dominio espacial y métodos en el dominio de la frecuencia. Cuando se trabaja en el dominio espacial, se

<sup>9</sup> La interpretación de las imágenes anteriores (remota, cartográfica y DEM) es la usada comúnmente en el área de la geocomputación.

manipulan directamente las intensidades o colores de la geoimagen original, mientras que el dominio de la frecuencia aplica para aquellos métodos que utilizan, entre otras, la Transformada de Fourier de la imagen (Gonzalez & Woods, 2002).

Anteriormente, la mayoría de las aplicaciones daban prioridad a los métodos en el dominio espacial, debido a que para aplicar un algoritmo en el dominio de la frecuencia se requería calcular la transformada de Fourier y su inversa, lo cual ocasionaba que el tiempo de procesamiento fuera alto. Sin embargo, los avances tanto en la optimización del cálculo de la transformada de Fourier usando la Transformada Rápida de Fourier (*Fast Fourier Transform – FFT*), como en los sistemas de cómputo, ha dado lugar a que el uso de técnicas en el dominio de la frecuencia sea cada vez más común. Adicionalmente, los métodos en este dominio se caracterizan por una simplificación significativa de los cálculos, ya que la convolución, que es la operación base para los filtros, en el dominio de la frecuencia se transforma en una operación sencilla de multiplicación.

### 3.3.2 Preprocesamiento en el dominio espacial

Una gran cantidad de operaciones del procesamiento de imágenes son lineales por naturaleza, en las que un elemento de la imagen de salida se obtiene a partir de combinaciones lineales de los píxeles de la imagen de entrada<sup>10</sup>. Entre las operaciones que están dentro de esta categoría tenemos la superposición, la convolución, la transformación unitaria y el filtrado lineal discreto.

Considerando un elemento de una imagen de entrada  $F(n_1, n_2)$  de tamaño  $N_1 \times N_2$ , una operación lineal que genera una imagen de salida  $P(m_1, m_2)$  de tamaño  $M_1 \times M_2$  puede ser generalizada mediante la expresión:

$$P(m_1, m_2) = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} F(n_1, n_2) O(n_1, n_2; m_1, m_2) \quad (3.2)$$

Donde el operador de núcleo (*kernel operator*)  $O(n_1, n_2; m_1, m_2)$  representa una constante ponderada, que en general es una función que depende de los valores actuales de las coordenadas

<sup>10</sup> Se describirán en mayor medida algoritmos en el dominio espacial. El lector puede encontrar más detalles del funcionamiento de metodologías y algoritmos del dominio de la frecuencia en las referencias de este trabajo.

de las imágenes de entrada y salida. Los métodos de mejoramiento de una imagen abarcan el uso de filtros de realce, suavizamiento y modificación del histograma.

### 3.3.2.1 Modificación del histograma

Consiste en hacer un ajuste al histograma de la imagen original, con el objetivo de que el histograma de la imagen resultante siga un patrón o forma predefinido; se utiliza en imágenes donde las regiones oscuras no siempre son perceptibles. Existen dos tipos de variantes en esta técnica, que son la modificación del histograma no adaptativa (ver Figura 3–6) y las adaptativas.

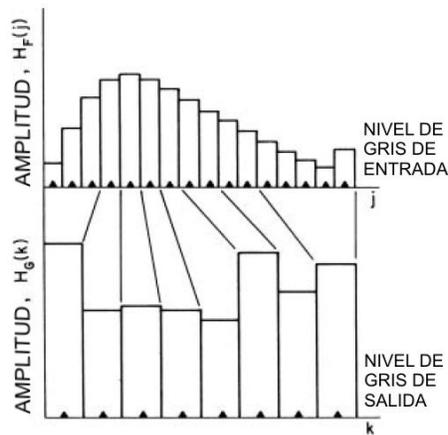


Figura 3–6. Manipulación del histograma.

El proceso de modificación del histograma puede considerarse como una transformación puntual monótonica  $g_k = T\{f_j\}$ , donde la amplitud de la variable de entrada  $f_1 \leq f_j \leq f_J$  es mapeada en una variable de salida  $g_1 \leq g_k \leq g_K$  tal que la densidad de probabilidad de salida  $P_R\{g_k = b_k\}$  sigue una forma deseada para una densidad de probabilidad de entrada  $P_R\{f_j = a_j\}$ , donde  $a_j$  y  $b_k$  son las variables de reconstrucción de los niveles  $j$ -ésimo y  $k$ -ésimo. Dado que las distribuciones de probabilidad de entrada y salida deben sumar 1, entonces:

$$\sum_{j=1}^J P_R\{f_j = a_j\} = 1 \quad (3.3)$$

$$\sum_{k=1}^K P_R\{g_k = b_k\} = 1 \quad (3.4)$$

Estas distribuciones acumulativas deben ser iguales para cualquier índice de entrada  $j$ . Esto es, la probabilidad de que los pixeles en la imagen de entrada tengan una amplitud menor o igual a  $a_j$ , debe ser igual a la probabilidad de que los pixeles de salida tengan una amplitud menor o igual a  $b_k$ , donde  $b_k = T\{a_j\}$ ; debido a que el argumento de la transformación es monótonico, tenemos:

$$\sum_{n=1}^k P_R\{g_n = b_n\} = \sum_{m=1}^j P_R\{f_m = a_m\} \quad (3.5)$$

Donde la expresión de lado derecho es una sumatoria que denota la distribución acumulativa de la imagen de entrada, mientras que la del lado izquierdo describe la distribución acumulativa para la imagen de salida. Para una imagen cualquiera, la distribución acumulativa se puede remplazar por el histograma acumulativo, lo que da la siguiente ecuación:

$$\sum_{n=1}^k P_R\{g_n = b_n\} = \sum_{m=1}^j H_F(m) \quad (3.6)$$

En la Tabla 3–1 se muestran algunas funciones de transferencia para ciertos modelos de densidad de probabilidad comunes.

Tabla 3–1. Funciones de transferencia para la modificación del histograma

Filtro	Ecuación
<b>Uniforme</b>	$p_g(g) = \frac{1}{g_{max} - g_{min}}$ $g_{min} \leq g \leq g_{max}$
Función de transferencia	$g = (g_{max} - g_{min})P_f(f) + g_{min}$
<b>Exponencial</b>	$p_g(g) = \alpha \exp\{-\alpha(g - g_{min})\}$ $g \geq g_{min}$
Función de transferencia	$g = g_{min} - \frac{1}{\alpha} \ln\{1 - P_f(f)\}$
<b>Rayleigh</b>	$p_g(g) = \frac{g - g_{min}}{\alpha^2} \exp\left\{-\frac{(g - g_{min})^2}{2\alpha^2}\right\}$ $g \geq g_{min}$
Función de transferencia	$g = g_{min} + \left[2\alpha^2 \ln\left\{\frac{1}{1 - P_f(f)}\right\}\right]^{1/2}$

Tabla 3–1. (Cont...) Funciones de transferencia para la modificación del histograma

Filtro	Ecuación
<b>Hiperbólico</b>	$p_g(g) = \frac{1}{3} \frac{g^{-2/3}}{g_{max}^{1/3} - g_{min}^{1/3}}$
Función de transferencia	$g = \left[ g_{max}^{1/3} - g_{min}^{1/3} \left[ P_f(f) + g_{max}^{1/3} \right] \right]^3$
<b>Hiperbólico logarítmico</b>	$p_g(g) = \frac{1}{g[\ln\{g_{max}\} - \ln\{g_{min}\}]}$
Función de transferencia	$g = g_{min} \left( \frac{g_{max}}{g_{min}} \right)^{P_f(f)}$

Donde la distribución de probabilidad acumulativa  $P_f(f)$  de la imagen de entrada es aproximadamente igual al histograma acumulativo, es decir:

$$P_f(f) \approx \sum_{m=0}^j H_F(m) \tag{3.7}$$

### 3.3.2.2 Filtrado de una imagen

Los algoritmos de filtrado permiten realzar o atenuar los detalles de bajo nivel de una imagen dada, o bien, reducir el ruido presente en la imagen; para ello se clasifican en dos categorías: de suavizamiento y realce. Los primeros tienden a minimizar las componentes de alta frecuencia de una imagen, que corresponden a los detalles o bordes; por otra parte, los algoritmos de realce tienen un objetivo opuesto, es decir, hacer énfasis o resaltar las componentes de alta frecuencia. Generalmente se aplican para eliminar los elementos parásitos o no deseados de una imagen (ruido).

No existe un algoritmo universal que funcione para todos los tipos de ruido, sino es la práctica la que le permite a un usuario contar con conocimiento *a priori* sobre el algoritmo en particular que proporciona los mejores resultados<sup>11</sup>, según el patrón de ruido presente en la imagen, pero sí existen sugerencias o casos de aplicación para los filtros de mejoramiento de una imagen.

<sup>11</sup> De igual forma, el término “mejores resultados” es subjetivo, ya que no existe una forma de medir la calidad de una imagen; generalmente el sistema visual humano es el juez determinante.

La operación base para estos filtros es la operación de convolución, la cual, matemáticamente, está dada por la siguiente expresión:

$$g(x, y) = \sum_{s=-1}^a \sum_{t=-b}^b h(s, t) f(x + s, y + t) \quad (3.8)$$

Donde  $a = (m - 1)/2$  y  $b = (n - 1)/2$ ,  $m$  es el número de renglones de la máscara del filtro,  $n$  es el número de columnas,  $h(s, t)$  se refiere a la intensidad en la posición  $(s, t)$  de la máscara y  $f(x, y)$  a la posición de la máscara en la imagen a procesar. Un filtro de mejoramiento basado en máscaras, debe cumplir con ciertas propiedades, las cuales son:

1. **Corrimiento nulo** (*Zero shift*). La característica más importante de un filtro es que no debe desplazar la posición de los objetos. Cualquier error introducido por un operador de preprocesamiento causaría errores en las estimaciones de la posición y posiblemente en otras características geométricas de los objetos. En orden, para no causar un desplazamiento, la función de transferencia de un filtro debe ser real. Un filtro que cumple esta propiedad es conocido como filtro de fase cero, debido a que no introduce ninguna componente de fase en ninguna de las componentes periódicas de una imagen. Una función de transferencia real implica que la máscara del filtro debe de ser simétrica, esto es:

$$h_{-m,n} = h_{m,n}, h_{m,-n} = h_{m,n} \quad (3.9)$$

2. **Preservación de la media.** Un operador de mejora debe preservar el valor promedio. Esta condición indica que la función de transferencia para una onda nula es 1, o equivalentemente, que la suma de los coeficientes de la máscara es uno.

$$\hat{h}(0) = 1, \text{ o bien, } \sum_m \sum_n h_{mn} = 1 \quad (3.10)$$

3. **Función de transferencia monotónicamente decreciente.** Intuitivamente, se podría esperar que cualquier operador de filtrado afecte en mayor grado conforme la escala es menor. En términos matemáticos, esto significa que la función de transferencia decrece monotónicamente con el número de ondas:

$$\hat{h}(\tilde{k}_1) \leq \hat{h}(\tilde{k}_2), \text{ si } \tilde{k}_2 > \tilde{k}_1 \quad (3.11)$$

4. **Isotropía.** En la mayoría de las aplicaciones, la atenuación o realce debe ser el mismo en cualquier dirección, con el objeto de no dar preferencia a ninguna de ellas. Entonces, tanto la máscara del filtro como la función de transferencia, deben ser isotrópicas. En consecuencia, la máscara depende únicamente de la magnitud de la distancia del píxel central, mientras que la función de transferencia depende de la magnitud del número de ondas:

$$h(x) = h(|x|) \text{ y } \hat{h}(\tilde{k}) = \hat{h}(|\tilde{k}|) \quad (3.12)$$

Los filtros suavizantes se clasifican en dos grandes grupos: de promediado y de ordenamiento. Los filtros de ordenamiento son filtros espaciales cuya respuesta se basa en el ordenamiento de los píxeles contenidos en un determinado entorno, acorde con el tamaño y forma de la máscara del filtro. La respuesta del filtro en cualquier punto de la imagen está determinada por el resultado de algún criterio de ordenamiento.

Acorde con Plataniotis, sean  $\mathbf{X}$  una variable aleatoria  $p$ -dimensional, es decir,  $\mathbf{X} = [X_1, X_2, \dots, X_p]^T$  y  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  un conjunto de  $n$  muestras de la multivariable  $\mathbf{X}$ . Cada una de las muestras  $\mathbf{x}_i$  es un vector  $p$ -dimensional, esto es  $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{ip}]^T$ . Estas muestras pueden ser ordenadas por medio de un ordenamiento marginal, condicional, parcial o reducido (**Plataniotis & Venetsanopoulos, 2000**).

En el esquema del ordenamiento marginal (M-ordenamiento), las muestras multivariadas son ordenadas independientemente a lo largo de cada una de las  $p$ -dimensiones:

$$\begin{aligned} \mathbf{x}_{1(1)} &\leq \mathbf{x}_{1(2)} \leq \dots \leq \mathbf{x}_{1(n)} \\ \mathbf{x}_{2(1)} &\leq \mathbf{x}_{2(2)} \leq \dots \leq \mathbf{x}_{2(n)} \\ &\dots \\ \mathbf{x}_{p(1)} &\leq \mathbf{x}_{p(2)} \leq \dots \leq \mathbf{x}_{p(n)} \end{aligned} \quad (3.13)$$

El ordenamiento condicional (C-ordenamiento) de las muestras multivariadas consiste en ordenar condicionalmente uno de los conjuntos marginales de observación. Entonces, una de las componentes marginales es calificada y las otras componentes de cada vector son enlistadas de

acuerdo con la posición de la componente calificada. Asumiendo que se califica la primera dimensión, el ordenamiento de las muestras puede ser representado como sigue:

$$\begin{aligned}
 \mathbf{x}_{1(1)} &\leq \mathbf{x}_{1(2)} \leq \dots \leq \mathbf{x}_{1(n)} \\
 \mathbf{x}_{2[1]} &\leq \mathbf{x}_{2[2]} \leq \dots \leq \mathbf{x}_{2[n]} \\
 &\dots \\
 \mathbf{x}_{p[1]} &\leq \mathbf{x}_{p[2]} \leq \dots \leq \mathbf{x}_{p[n]}
 \end{aligned}
 \tag{3.14}$$

Donde  $\mathbf{x}_{1(i)}$ ,  $i = 1, 2, \dots, n$  es el resultado de aplicar un ordenamiento marginal en la primera dimensión, mientras que  $\mathbf{x}_{j[i]}$ ,  $j = 2, 3, \dots, p$ ;  $i = 1, 2, \dots, n$  son las muestras cuasi ordenadas en las dimensiones  $j = 2, 3, \dots, p$ , de forma condicional con el ordenamiento marginal de la primera dimensión. Estas componentes en realidad no son ordenadas, simplemente son enlistadas acorde con las componentes calificadas.

En el ordenamiento parcial (P-ordenamiento) los subconjuntos de datos son agrupados conjuntamente formando cercos convexos mínimos. El primer cerco convexo se forma de manera tal que el perímetro contenga un número mínimo de puntos y el cerco resultante todos los otros puntos del conjunto dado. Los puntos a lo largo del perímetro son etiquetados como grupo 1 c-ordenado. Estos puntos forman el grupo más extremo. Posteriormente, se descartan los puntos del perímetro y se repite nuevamente el proceso. Los nuevos puntos del perímetro resultante son denotados como del grupo 2 c-ordenado, y así sucesivamente.

En el ordenamiento reducido (de agregación) o R-ordenamiento, se reduce cada observación multivariable  $\mathbf{x}_i$  a un valor escalar por medio de alguna combinación de los valores de las componentes. Los valores escalares resultantes pueden ser ordenados mediante técnicas monovariable.

Dentro de los filtros de ordenamiento más comunes se encuentran los filtros mediana, máximo, mínimo, de punto medio y el alfa-recortador. En la Tabla 3–2, se muestran las ecuaciones de estos algoritmos. Por otra parte, los filtros promediadores se basan en obtener un entorno para posteriormente aplicar una ecuación de promediado. En esta categoría de filtros de promediado, ver Tabla 3–3, se encuentran los de media aritmética, geométrica, armónica y contra armónica.

Tabla 3–2. Ecuaciones de filtros de ordenamiento.

Filtro	Ecuación
Mediana	$\hat{f}(x, y) = \text{mediana}_{(s,t) \in H_{xy}}\{g(s, t)\}$
Máximo	$\hat{f}(x, y) = \text{máximo}_{(s,t) \in H_{xy}}\{g(s, t)\}$
Mínimo	$\hat{f}(x, y) = \text{mínimo}_{(s,t) \in H_{xy}}\{g(s, t)\}$
Punto medio	$\hat{f}(x, y) = \frac{1}{2} \left[ \text{máximo}_{(s,t) \in H_{xy}}\{g(s, t)\} + \text{mínimo}_{(s,t) \in H_{xy}}\{g(s, t)\} \right]$
Alfa-recortador	$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in H_{xy}} g_r(s, t)$

Tabla 3–3. Ecuaciones de filtros de promediado

Filtro	Ecuación
Media aritmética	$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in H_{xy}} g(s, t)$
Geométrico	$\hat{f}(x, y) = \left[ \prod_{(s,t) \in H_{xy}} g(s, t) \right]^{\frac{1}{mn}}$
Armónico	$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in H_{xy}} \frac{1}{g(s, t)}}$
Contra armónico	$\hat{f}(x, y) = \frac{\sum_{(s,t) \in H_{xy}} [g(s, t)]^{Q+1}}{\sum_{(s,t) \in H_{xy}} [g(s, t)]^Q}, Q \in \mathbb{Z}$

### 3.3.3 Segmentación

La segmentación subdivide una imagen en sus regiones u objetos constituyentes; en base al problema a resolver es la forma en que se determina el nivel de subdivisión al que se debe llegar. Las técnicas de segmentación se dividen en cuatro grandes categorías:

- a. Segmentación basada en píxeles.
- b. Segmentación basada en bordes.
- c. Segmentación basada en regiones.
- d. Segmentación basada en modelos.

### 3.3.3.1 Segmentación basada en píxeles

Es la técnica más simple para realizar una segmentación y básicamente consiste en extraer un conjunto de características que representen a un objeto de una forma única, antes de aplicar un procedimiento de segmentación. Inicialmente se aplican algoritmos que permiten optimizar la iluminación de la escena, usando por ejemplo, la modificación o ecualización del histograma; tomando como base el histograma, se trata de determinar el umbral  $k$  que permite hacer una separación entre los píxeles de fondo y los de objeto, ver Figura 3–7. En consecuencia, esta técnica se basa en el uso de algoritmos de umbralado para realizar una segmentación en dos regiones: la de los píxeles de fondo y la de los objetos. Uno de los algoritmos base para esta técnica es el método de Otsu, el cual trata de modelar los histogramas como una distribución normal.

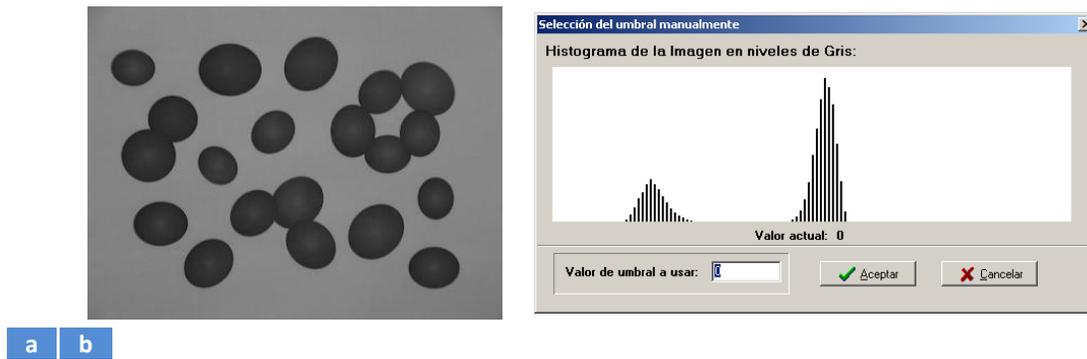


Figura 3–7. (a) Imagen en niveles de gris; (b) Histograma correspondiente.

### 3.3.3.2 Segmentación basada en bordes

Estas técnicas se basan en la detección de bordes, la cual es una discontinuidad en las intensidades de niveles de gris de la imagen. El principal problema que presenta esta técnica es que los bordes en una imagen no siempre indican las fronteras de los objetos, ya que la ubicación de una fuente luminosa puede generar bordes. Estos métodos de segmentación se basan en el hecho de que la ubicación de un borde está representado por un máximo o un mínimo en la primera derivada (Roberts, Prewitt y Sobel), o bien, por un cruce por cero en la segunda derivada (Laplaciano).

Matemáticamente, para detectar los bordes usando la primera derivada de una imagen, se usa el gradiente, el cual se define como un vector de la siguiente forma:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}, \quad \nabla f = \text{mag}(\nabla f) = \sqrt{G_x^2 + G_y^2} \quad (3.15)$$

Por otra parte, el Laplaciano permite determinar la segunda derivada de una imagen:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.16)$$

En la Tabla 3-4 se enlistan el conjunto de máscaras más comunes para determinar respectivamente el gradiente y el Laplaciano de una imagen.

Tabla 3-4. Máscaras más comunes para la segmentación basada en bordes.

Filtro	Máscaras
Roberts	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Laplaciano	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ o bien $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

### 3.3.3.3 Segmentación basada en regiones

Sea  $R$  la representación de la imagen completa como una región; las técnicas de esta categoría se basan en particionar  $R$  en un conjunto de  $n$ -regiones,  $R_1, R_2, \dots, R_n$ , tal que:

- $\bigcup_{i=1}^n R_i = R$ .
- $R_i$  es una región conectada,  $\forall i = 1, 2, \dots, k$ .
- $R_i \cap R_j = \emptyset$ , para todo  $i \neq j, i = 1, 2, \dots, k; j = 1, 2, \dots, k$ .
- $P(R_i) = \text{VERDADERO}, \forall i = 1, 2, \dots, k$ .
- $P(R_i \cap R_j) = \text{FALSO}, \forall i \neq j$ .

La primera condición estipula que la unión de todas las regiones es igual a la región a particionar  $R$ , con ello, todos los elementos de la región original, al final del algoritmo deben de estar clasificados en alguna de las nuevas  $n$ -regiones. En el segundo punto se tiene que cumplir que cada nueva región esté conectada conforme se estableció en la Definición 3.4. La tercera condición indica que las regiones no comparten elementos en común, es decir, pixeles. Posteriormente se define una condición para la cual el predicado de que una región exista es verdadero, por lo que no es posible definir regiones vacías. Por último, se agrega una condición que señala que no es posible encontrar áreas comunes entre regiones diferentes, es decir, un mismo elemento no puede estar asociado a dos regiones diferentes.

Existen dos algoritmos ampliamente usados dentro de esta clasificación: el crecimiento de regiones y el algoritmo de división y fusión de regiones (*split and merge*). A continuación describiremos brevemente estos dos algoritmos.

**Crecimiento de regiones.** Es un algoritmo que agrupa pixeles o subregiones en regiones más grandes basándose en un criterio conocido *a priori*. La idea básica consiste en iniciar con un conjunto aleatorio de pixeles conocidos como semillas, y a partir de estos pixeles hacer crecer las regiones agregando a cada semilla los pixeles vecinos que tienen propiedades similares a las de la semilla en particular.

Existen tres problemas importantes asociados con esta técnica que son: la selección de la cantidad y ubicación de las semillas, el criterio de similitud y la condición de finalización. En lo que respecta a la cantidad de semillas generalmente se elige un número igual a la cantidad de regiones que se desea obtener; cuando este dato se conoce antes de iniciar el algoritmo, la selección de este parámetro no presenta gran dificultad; sin embargo, cuando este dato no se conoce *a priori*, la elección de la cantidad de semillas puede dar origen a una sub-clasificación o sobre-clasificación. Para la ubicación de las semillas generalmente se usa un generador de números aleatorios con alguna densidad de probabilidad tal como la uniforme.

El criterio de similitud es el que da el desempeño al algoritmo de segmentación, ya que al usar un criterio de similitud muy estricto ocasionará que solamente los pixeles con alto grado de semejanza con las características de bajo nivel de las semillas sean clasificados, mientras que un

umbral bastante holgado, podría tener como consecuencia que algunos píxeles puedan ser clasificados en cierta región a la que no pertenecen. Como condición para detener este algoritmo se usa el criterio de detener el algoritmo cuando no haya más píxeles por clasificar.

**División y fusión de regiones.** Este algoritmo divide la imagen original en un número predeterminado de regiones, para posteriormente inspeccionar cada una de ellas. Al estar analizando una determinada región, se cuantifica alguna(s) característica(s) de bajo nivel de la imagen, como puede ser la media del nivel de gris, varianza, textura, etc.; en caso de que no se cumpla un criterio dado de homogeneidad, entonces se procede a dividir esta región. El proceso de división se alterna con una etapa de análisis para determinar si es posible realizar una fusión de regiones vecinas, cuando se cumple un criterio también preestablecido, y se pueda concluir que las regiones tienen un alto grado de similitud y deben unirse. Este algoritmo se detiene cuando no se hacen más divisiones o fusiones. Su buen desempeño radica en la buena selección de los criterios para realizar las divisiones o fusiones de las regiones de la imagen.

### 3.3.3.4 Segmentación basada en modelos

A diferencia de las otras técnicas de segmentación, en donde la información contenida en la imagen es el criterio que se usa para realizar la segmentación de la misma, la segmentación basada en modelos trata de encontrar patrones específicos en la imagen. Dentro de los patrones que se pueden usar se encuentran líneas, polígonos, regiones, etc. Para el caso más simple (que son las líneas) generalmente se usa la Transformada de Hough, ver Figura 3–8.



a b

Figura 3–8. Uso de la Transformada de Hough; (a) Imagen original; (b) Detección de los vértices formados por las líneas.

### 3.3.4 Agrupamiento (*clustering*)

Un agrupamiento (*cluster*) está compuesto por un conjunto de objetos similares recolectados o agrupados conjuntamente. La definición de agrupamiento por Everitt es la siguiente (Everitt, 1974):

- Un agrupamiento es un conjunto de entidades que son similares o iguales (*alike*), mientras que las entidades de agrupamientos diferentes no son similares o iguales (*not alike*).
- Un agrupamiento es una agregación de puntos en el espacio de prueba tal que la distancia entre dos puntos de un agrupamiento es menor que la distancia entre cualquier punto del agrupamiento y otro que no está en él.
- Los agrupamientos pueden ser descritos como regiones conectadas en un espacio multi-dimensional que contiene una densidad alta de puntos, separadas de otras regiones por una región que contiene una baja densidad de puntos.

Con esta definición, Everitt supone que los objetos a ser clasificados están representados por medio de puntos en el espacio de medición. Pero como plantea Jain, el principal problema no es la definición conceptual de lo que es un agrupamiento, si no su definición operacional (Jain & Dubes, 1988). Lo anterior se debe a que una distribución de objetos puede ser clasificada en diferentes formas con base en los requisitos que se impongan. Para apreciar de mejor manera este punto, se pueden considerar los objetos de la Figura 3–9. En este caso, al usar un grado de similitud alto, se obtendría un total de 4 agrupamientos, pero si se usara un grado de similitud bajo, se obtendría un total de 12 agrupamientos.

El problema crucial en la identificación de agrupamientos es la forma de especificar la proximidad y cómo cuantificarla o medirla, ya que la proximidad es un problema que depende de la aplicación. Las técnicas de clasificación ofrecen varias ventajas en comparación con un proceso manual de agrupamiento. Primero, un programa de clasificación puede aplicar objetivamente el criterio que haya sido especificado para formar los grupos.

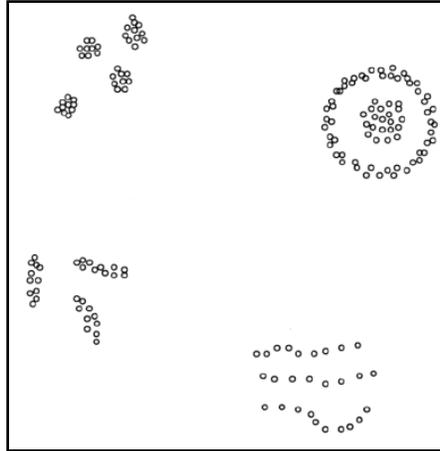


Figura 3–9. Agrupamiento de patrones de puntos en dos dimensiones.

Sin embargo, un grupo de personas puede obtener resultados diferentes aplicando una clasificación manual, porque aunque los humanos podemos realizar procesos de clasificación con resultados óptimos, tanto en 2 como en 3 dimensiones, cada persona posee experiencias y conocimientos *a priori* diferentes. La segunda ventaja sería la velocidad de procesamiento y la consistencia de los resultados.

Los algoritmos de clasificación por agrupamiento son reconocedores que requieren una definición previa de un vector de características, y en ocasiones el número de clases que se desea obtener. Una vez establecido el vector de características, los procedimientos de agrupación de clases reciben como datos de entrada los objetos a clasificar, representados mediante vectores que contienen valores que cuantifican sus propiedades, de modo que a partir de estos datos de entrada el algoritmo agrupa estos vectores en clases (Gomez-Allende, 1994).

Estas técnicas se utilizan cuando no existe un conocimiento *a priori* suficiente acerca de las clases en que se pueden distribuir los objetos de interés. No obstante, también son de gran interés práctico en otras áreas (análisis automático de voz, visión artificial, clasificación de señales, etc.) en las que existe un conocimiento completo de las clases y por tanto, se pueden aplicar los métodos supervisados. En estos casos, una vez diseñada la etapa de reconocimiento automático, los algoritmos de agrupación se utilizan como herramienta de verificación y prueba de la calidad del reconocedor y, en particular, del vector de características elegido.

En efecto, si se introducen como datos de entrada del algoritmo de agrupación los vectores del conjunto de muestras que se emplearon para el diseño del correspondiente clasificador supervisado, entonces los resultados que se obtengan como salida del algoritmo de agrupación indicarán la calidad del vector de características. Si la agrupación efectuada por aquél coincide con las clases reales, entonces el vector está bien escogido; en caso contrario (si obtienen solapamientos o agrupaciones incorrectas) se trata de un vector de características con propiedades indeseables. Por último, estas técnicas de agrupación se han utilizado como alternativa completa para el diseño supervisado, aunque en general, los resultados no son los mejores.

Los algoritmos de agrupación varían entre sí por el mayor o menor grado de reglas heurísticas que utilizan, y por el nivel de los procedimientos formales involucrados. Todos ellos se basan en el empleo sistemático de las distancias entre los vectores, así como entre las clases o grupos que se van haciendo o deshaciendo a lo largo del proceso de actuación del algoritmo concreto. Generalmente, para lograr los agrupamientos se emplea la distancia Euclidiana entre vectores:

$$d_E(\vec{X}_i, \vec{X}_j) = \sqrt{\sum_{k=1}^n (X_{ik} - X_{jk})^2} \quad (3.17)$$

Los algoritmos más comunes que pertenecen a esta categoría son:

- Algoritmo de las distancias encadenadas.
- Algoritmo max-min.
- Algoritmo  $k$ -medias y  $c$ -medias difuso.
- Algoritmo ISODATA y su variante el algoritmo K-SODATA.

Debido a que principalmente se hará referencia a los algoritmos  $k$ -medias,  $c$ -medias difuso e ISODATA, éstos serán los únicos que se explicarán, para mayor información puede revisarse [\(Gomez-Allende, 1994\)](#).

### 3.3.4.1 Algoritmo $k$ -medias (Gomez-Allende, 1994)

Este algoritmo es probablemente uno de los algoritmos de agrupación más empleados dentro de muchas áreas, como el PDI, donde se hace referencia a que existen  $k$  clases o patrones, por lo cual es necesario conocer anticipadamente el número de clases existentes. Es un algoritmo sencillo, pero muy eficiente, siempre que el número de clases se conozca con exactitud.

Partiendo de un conjunto de objetos a clasificar  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p$ , el algoritmo de  $k$ -medias consta de los siguientes pasos:

1. Estableciendo previamente el número exacto de clases existentes, digamos  $k$ , se escogen al azar entre los elementos a agrupar un total de  $k$  vectores, de forma que van a constituir los centroides de las  $k$  clases, esto es:

$$\alpha_1: \mathbf{Z}_1(1); \alpha_2: \mathbf{Z}_2(1); \dots; \alpha_k: \mathbf{Z}_k(1) \quad (3.18)$$

Donde  $\alpha_i$  es la  $i$ -ésima clase,  $\mathbf{Z}_i$  representa el centroide de la  $i$ -ésima clase y el número entre paréntesis indica el número de la iteración ( $n$ ), la que en un inicio se establece en el valor uno.

2. Debido a que es un proceso iterativo donde  $n$  denota el número de iteración, en el paso  $n$  se distribuyen todas las muestras  $\{\mathbf{X}\}_{1 \leq j \leq p}$  entre las  $k$  clases, de acuerdo con la siguiente regla:

$$\mathbf{X} \in \alpha_j(n) \text{ si y solo si } \|\mathbf{X} - \mathbf{Z}_j(n)\| < \|\mathbf{X} - \mathbf{Z}_i(n)\| \quad \forall i = 1, 2, \dots, k, i \neq j \quad (3.19)$$

3. Una vez redistribuidos los elementos a agrupar entre las diferentes clases, es preciso recalcular o actualizar los centroides de las clases. El objetivo en el cálculo de los nuevos centroides es minimizar el índice de rendimiento de la ecuación (3.20):

$$J_i = \sum_{\mathbf{X} \in \alpha_i(n)} \|\mathbf{X} - \mathbf{Z}_i(n)\|^2 \quad i = 1, 2, \dots, k \quad (3.20)$$

4. Este índice se minimiza utilizando la media muestral o aritmética de  $\alpha_i(n)$ :

$$\mathbf{z}_i(n+1) = \frac{1}{N_i(n)} \sum_{\mathbf{x} \in \alpha_i(n)} \mathbf{x} \quad i = 1, 2, \dots, k \quad (3.21)$$

Donde  $N_i(n)$  es el número de elemento de la clase  $\alpha_i$  en la iteración  $n$ .

5. Se comprueba si el algoritmo ha alcanzado una posición estable, lo cual ocurre cuando se cumple la siguiente condición:

$$\mathbf{z}_i(n+1) = \mathbf{z}_i(n) \quad i = 1, 2, \dots, k \quad (3.22)$$

6. Si se cumple el punto 5, el algoritmo ha finalizado; en caso contrario se repite desde el paso 2.

### 3.3.4.2 Algoritmo c-medias difuso

Este algoritmo plantea que un objeto puede pertenecer a distintos grupos, como en la lógica difusa (de ahí su nombre), en lugar de pertenecer completamente a una sola clase. Fue definido por Dunn en 1973 (Dunn 1973) y mejorado por Bezdeck en 1981 (Bezdeck, 1981) y es usado ampliamente en el reconocimiento de patrones. En consecuencia, el conjunto de objetos que están ubicados en el límite de la clase, tienen un menor grado de pertenencia a dicha clase, que aquellos objetos que se encuentran en el centro de la misma. Para cada objeto  $\mathbf{X}$  se determina su grado de pertenencia a la  $k$ -ésima clase, esto se denota como  $u_k(\mathbf{X})$ . Comúnmente la suma de todos los coeficientes de pertenencia para un objeto con respecto a todas las clases es igual a 1, es decir:

$$\sum_{k=1}^n u_k(\mathbf{X}) = 1 \quad (3.23)$$

Donde  $n$  es el número de clases o *clusters*.

En esta técnica de clasificación, el centroide de una clase es la media de todos los puntos, ponderados por el grado de pertenencia a una clase:

$$centroide_k = \frac{\sum_{\mathbf{X}} u_k(\mathbf{X})^m \mathbf{X}}{\sum_{\mathbf{X}} u_k(\mathbf{X})^m} \quad (3.24)$$

El grado de pertenencia está relacionado con el inverso de la distancia de un objeto a la clase, es decir:

$$u_k(\mathbf{X}) = \frac{1}{d(centroide_k, \mathbf{X})} \quad (3.25)$$

Entonces, los coeficientes son normalizados y asociados de forma difusa con un parámetro real  $m > 1$ , lo cual está dado por la siguiente expresión:

$$u_k(\mathbf{X}) = \frac{1}{\sum_{j=1}^n \left( \frac{d(centroide_k, \mathbf{X})}{d(centroide_j, \mathbf{X})} \right)^{2/(m-1)}} \quad (3.26)$$

Para  $m = 2$ , esta ecuación es equivalente a normalizar linealmente el coeficiente haciendo que todas las sumas sean igual a 1.

### 3.3.4.3 Algoritmo ISODATA (Gomez-Allende, 1994)

El algoritmo ISODATA (*Iterative Self-Organizing Data Analysis Techniques*) es un método de agrupación de datos cuya base es el algoritmo  $k$ -medias, y consiste de varios pasos.

Las variables usadas en este algoritmo tienen el siguiente significado:

- $N_c$  Es el número actual en una iteración de agrupamientos, grupos o clases
- $k$  Es el número deseado o estimado de clases *a priori*
- $\theta_N$  Número mínimo de miembros o elementos de una clase para constituirse como tal
- $\theta_s$  Desviación típica máxima. Sirve para aplicar un criterio de división en dos de un grupo o clase
- $\theta_c$  Es un parámetro de unión de dos clases
- $L$  Umbral para determinar el número de clases a considerar para una fusión
- $I$  Número máximo de iteraciones que se desea ejecute el algoritmo

1. **Inicialización.** Se inicializan los parámetros del ISODATA, para lo cual, se recomienda hacer  $N_c = k$ . En esta fase de inicialización se escogen aleatoriamente  $k$  elementos entre los  $p$  objetos a clasificar:  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p$  formándose con cada uno de ellos una clase inicial. Se tendrán entonces  $K = N_c$  primeros centroides  $\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_{N_c}$ .
2. **Distribución de los elementos entre las diferentes clases.** Se agrupan los elementos o muestras  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_p$  entre los  $N_c$  grupos ya formados según el principio de la mínima distancia Euclidiana.

$$\mathbf{X}_j \in \alpha_i \text{ si y solo si } \|\mathbf{X}_j - \mathbf{Z}_i\| \text{ es mínima} \quad (3.27)$$

$$\forall j = 1, 2, \dots, p; \quad \forall i = 1, 2, \dots, N_c$$

3. **Eliminación de clases con un número insuficiente de miembros.** Se eliminan las clases con un número de elementos inferior a  $\theta_c$ . Por ende, habrá que actualizar el valor del parámetro  $N_c$ .
4. **Actualización de los centroides de las clases.** Esta actualización se realiza calculando la media muestral de cada grupo o clase, ver ecuación (3.28).

$$\mathbf{Z}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{X}_j \quad ; \quad i = 1, 2, \dots, N_c \quad (3.28)$$

Donde  $N_i$  es el número de elementos de la clase  $\alpha_i$ .

5. **Cálculo de la distancia Euclidiana media de cada clase.** Se obtiene para cada clase la distancia Euclidiana media de sus elementos respecto al correspondiente centroide.

$$\bar{D}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \|\mathbf{X}_j - \mathbf{Z}_i\| \quad ; \quad i = 1, 2, \dots, N_c \quad (3.29)$$

Este parámetro da una medida de la dispersión de las muestras de cada clase con respecto a su media y se utilizará para una posible división de un grupo.

6. **Cálculo de la distancia media de todas las clases.** Para obtener este valor, se aplica la siguiente ecuación:

$$\bar{D} = \frac{1}{N_c} \sum_{i=1}^{N_c} N_i \bar{D}_i \quad (3.30)$$

7. **Comprobación de bifurcaciones.** En este caso se comprueba en primer lugar si se trata de la última iteración, en cuyo caso se hace  $\theta_c = 0$  y se prosigue con el paso 11.

En segundo lugar, se realiza un test de posible unión de clases, viendo si  $N_c \geq 2k$ ; en caso de cumplirse, también se pasa al paso 11.

De no cumplirse la condición anterior, se continúa con los siguientes pasos que sirven para tratar de realizar una división de clases.

8. **Cálculo del vector de desviaciones típicas de cada grupo.** Dado que se trabaja con un vector de características  $n$ -dimensional, las clases presentarán un vector  $n$ -dimensional de desviaciones típicas, como se muestra en la ecuación (3.31).

$$\sigma_i = \begin{pmatrix} \sigma_{i1} \\ \sigma_{i2} \\ \dots \\ \sigma_{in} \end{pmatrix}; \quad \sigma_{ij} = \sqrt{\frac{1}{N_i} \sum_{k=1}^{N_i} (X_{kj} - Z_{ij})^2} \quad (3.31)$$

$i = 1, 2, \dots, N_c$  clases  
 $j = 1, 2, \dots, n$  características  
 $k = 1, 2, \dots, N_i$  elementos de la clase  $\alpha_i$

9. **Obtención de las desviaciones típicas máximas de cada grupo.** Se selecciona en cada clase la componente mayor del correspondiente vector de desviaciones típicas, formándose el conjunto:

$$\sigma_{max} = \{\sigma_{1 max}, \sigma_{2 max}, \dots, \sigma_{N_c max}\} \quad (3.32)$$

10. **Posible división de clases.** Para una clase cualquiera, por ejemplo  $\alpha_j$ , si se cumple  $\sigma_{j max} > \theta_s$  y además se satisfaga una de las siguientes dos condiciones:

a)  $D_j > D$  y  $N_j > 2(\theta_n + 1)$  (3.33)

$$b) N_c \leq k/2$$

La condición a) significa que la dispersión media de la clase  $\alpha_j$ , candidata a dividirse en dos clases, es superior a la media de las dispersiones de todas las clases y, además, que el número de sus elementos es al menos superior al doble del número mínimo para formar una clase. La segunda condición significa que el número de clases en la iteración actual es menor o igual al valor medio de clases deseadas.

Para dividir el grupo, existen varias alternativas; una de ellas consiste en crear dos nuevos centroides  $\mathbf{Z}_j^+$  y  $\mathbf{Z}_j^-$  a partir de  $\mathbf{Z}_j$ , tales que todas las componentes de los nuevos centroides coincidan con las de  $\mathbf{Z}_j$ , excepto la componente con máxima dispersión, por ejemplo  $Z_k$ , con dispersión  $\sigma_{j \max}$ , por lo que quedaría:

$$\begin{aligned} Z_{jk}^+ &= Z_{jk} + \gamma \sigma_{j \max} \\ Z_{jk}^- &= Z_{jk} - \gamma \sigma_{j \max} \end{aligned} \quad (3.34)$$

Para  $0 < \gamma \leq 1$ . Otra alternativa de división es obtener las dos muestras de la clase  $\alpha_j$  más alejadas entre sí con respecto al centroide; estas dos muestras forman los nuevos centroides:

$$\mathbf{z}_{j1} = \frac{(\mathbf{z}_j^+ + \mathbf{z}_j)}{2} \quad \mathbf{z}_{j2} = \frac{(\mathbf{z}_j^- + \mathbf{z}_j)}{2} \quad (3.35)$$

**11. Cálculo de la distancia entre clases.** Para realizar la unión de dos clases, se calculan previamente todas las distancias entre parejas de clases usando:

$$\begin{aligned} D_{ij} = D_{ji} &= \|\mathbf{z}_i - \mathbf{z}_j\| \\ i &= 1, 2, \dots, N_c - 1; j = i + 1, i + 2, \dots, N_c \end{aligned} \quad (3.36)$$

**12. Posible unión.** Se comparan las distancias  $D_{ij}$  con el parámetro  $\theta_c$ , de forma que se toman las  $L$  más pequeñas en orden creciente en caso de existir:

$$\{D_1, D_2, \dots, D_L\} \text{ con } D_1 < D_2 < \dots < D_L \quad (3.37)$$

**13. Proceso de unión.** Comenzando con las parejas de clases con las menores distancias, se procede como se indica a continuación. Suponiendo que se van a unir las clases  $i, j$ , cuya

distancia  $D_{ij}$  se encuentra dentro del conjunto definido por la ecuación (3.37), si y solo si, ninguna de estas dos clases ha sido previamente fusionada con otra en esta misma iteración, entonces se forma una nueva clase cuyo centroide está dado por:

$$\mathbf{Z}_{ij} = \frac{1}{N_i + N_j} \times (N_i \mathbf{Z}_i + N_j \mathbf{Z}_j) \quad (3.38)$$

Donde  $N_i$  y  $N_j$  son el número de muestras de las clases,  $\alpha_i$  y  $\alpha_j$  respectivamente, antes de la fusión. En caso de realizar alguna fusión, se debe actualizar el parámetro  $N_c$ .

**14. Comprobación de última iteración.** Se determina si se ha llegado a la iteración  $I$ . En caso de no cumplirse, se debe repetir desde el procedimiento 2.

### 3.3.5 Etiquetado de componentes conexas

Formalmente, se define que dos píxeles de intensidad 1,  $p$  y  $q$  pertenecen a la misma componente conexas  $C$  si existe una secuencia de píxeles con valor de 1 ( $p_0, p_1, \dots, p_n$ ) de  $C$  donde  $p_0 = p, p_n = q$  y  $p_i$  un vecino de  $p_{i-1}$  para  $i = 1, 2, \dots, n$ . En consecuencia, la definición de componente conexas depende de la definición de vecindad que se desee emplear. Cuando se usan cuatro vecinos, los vecinos horizontales y verticales para un determinado píxel, se obtienen regiones llamadas 4 – conexas. Por el contrario cuando se toman en cuenta los ocho vecinos de un píxel, los citados anteriormente más los cuatro vecinos en diagonal, entonces el resultado son regiones 8 – conexas.

Indistintamente a la definición de vecinos que sea utilizada, se dice que los vecinos de un píxel son adyacentes al píxel del que son vecinos. El borde de una componente conexas de píxeles de objeto, con valor 1, es el subconjunto de píxeles que pertenecen a la región conexas que son adyacentes a píxeles de fondo, de intensidad 0. Análogamente, el borde de una componente conexas constituida de píxeles de fondo es el subconjunto de píxeles que pertenecen a la componente que son adyacentes a los píxeles del objeto. Existen diversos algoritmos para realizar el etiquetado de las componentes conexas, cada uno de los cuales aumenta gradualmente su complejidad de comprensión y de implantación, pero favorecen el desempeño y su posible uso en aplicaciones de tiempo real.

Todos los algoritmos procesan una fila de la imagen de entrada en un instante de tiempo y asignan nuevas etiquetas al primer píxel encontrado por cada componente e intentan propagar la etiqueta de un píxel a todos sus vecinos en un orden de arriba hacia abajo y de izquierda a derecha (conocido como análisis *top – down*) o de abajo hacia arriba y de derecha a izquierda (análisis *bottom – up*).

El algoritmo secuencial iterativo desarrollado en 1981 por (Haralick, 1992) no utiliza almacenamiento secundario para producir la imagen simbólica etiquetada. Es muy útil en aplicaciones donde se tienen limitaciones del medio de almacenamiento, pero su desventaja principal es el tiempo requerido para realizar el procesamiento de la imagen de entrada. Los pasos requeridos por el algoritmo son los que se enlistan en la Figura 3–10.

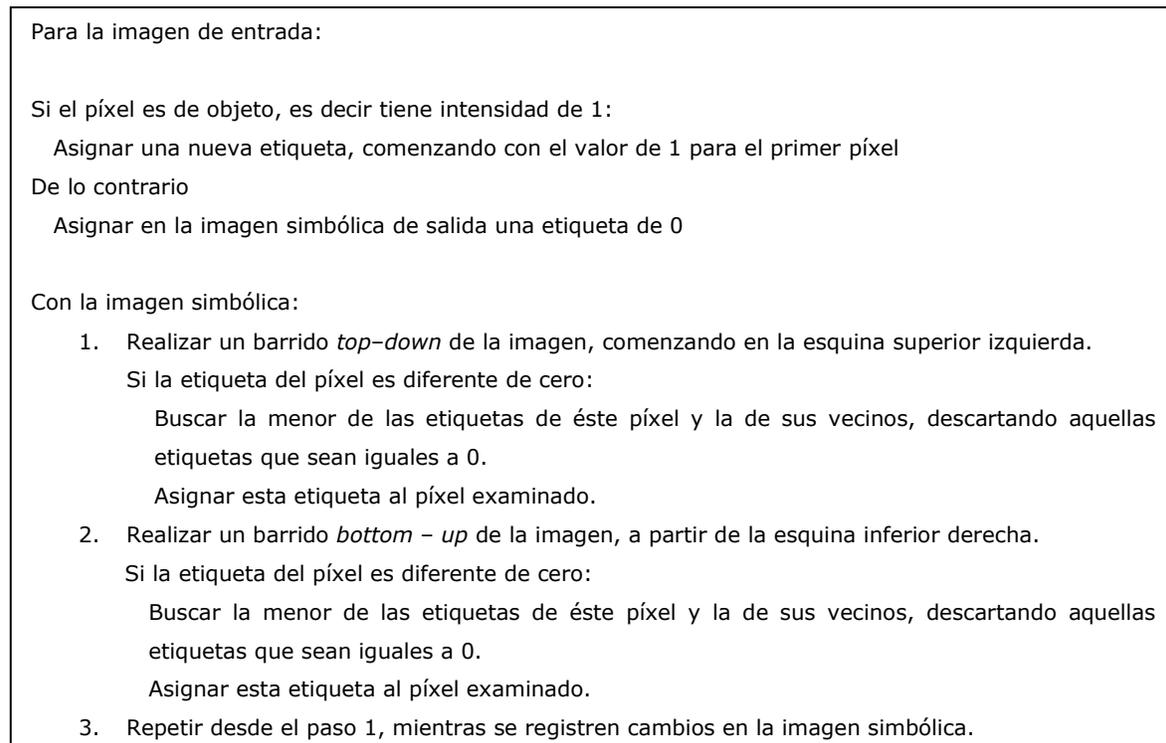


Figura 3–10. Algoritmo de etiquetado de componentes conexas.

Este algoritmo examina cada píxel de objeto y trata de asignar la menor de las etiquetas, inspeccionando el valor asignado a los vecinos. No lleva un registro de las equivalencias, realiza una propagación de las etiquetas en cada análisis que realiza en la imagen simbólica. Los análisis son alternados, comenzando con uno de tipo *top – down*, para hacer a continuación uno de tipo

*bottom – up*, finalizando cuando después de realizar un análisis, no se modifican los valores de las etiquetas en la imagen de salida.

### 3.3.6 Esqueleto de una imagen

Una de las técnicas que permite generar buenos resultados es la de (Zou et al., 2001). Considerando el segmento de un objeto cualquiera en el dominio continuo como aparece en la Figura 3–11, existen dos curvas que describen los contornos, llamadas  $\beta_1$  y  $\beta_2$ . Si se traza un círculo  $C$  de tal forma que únicamente dos puntos pertenecientes a la circunferencia hicieran intersección con  $\beta_1$  y  $\beta_2$ , se forman dos puntos de intersección, que serán llamados  $p_1$  y  $p_2$ , cuyas tangentes aparecen en la misma figura,  $t_1$  y  $t_2$ . La bisectriz del ángulo formado por los puntos  $p_1$  y  $p_2$  es en realidad un “espejo” que refleja una tangente en el otro extremo del segmento. Se dice que  $p_1$  y  $p_2$  forman una simetría local con respecto a la bisectriz  $M$ , o de manera breve, hacen una simetría local. Blum por su parte define el punto simétrico de una simetría local como el centro de la circunferencia  $C$  (Blum, 1973) (Blum, 1978). Por el contrario Brady lo definió como el punto medio de un segmento de línea recta que une a  $p_1$  y  $p_2$  nombrándolo también como una Simetría Local Suavizada (*Smoothed Local Symmetry - SLS*) (Brady, 1983) (Brady, 1984).

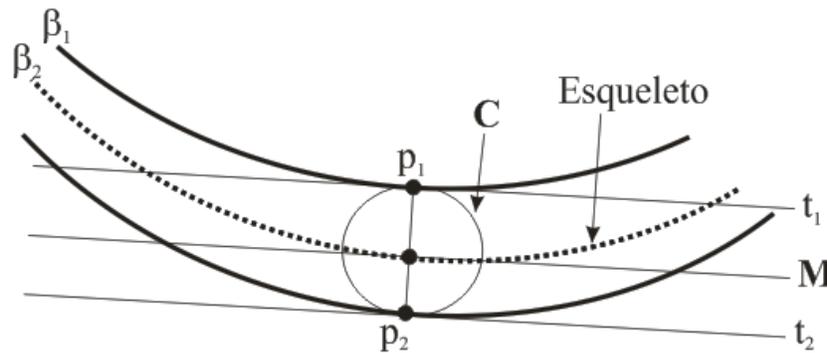


Figura 3–11. Una simetría local en el dominio del espacio continuo.

La principal dificultad de este concepto se presenta en las imágenes digitales. Tratando de aplicar lo anterior, por ejemplo, en la Figura 3–12 no es posible encontrar la pareja de puntos  $p_1$  y  $p_2$ . En conclusión, para algunos píxeles del contorno de un objeto en el espacio discreto, puede que no sea posible determinar sus contrapartes simétricas en el contorno del lado opuesto.

Sea un conjunto de píxeles que describen una imagen digital; se trazan segmentos de línea recta para unir puntos del contorno adyacentes, para así obtener contornos completamente cerrados. Un contorno digital está compuesto de dos tipos de elementos: píxeles y segmentos de contorno. Un elemento de contorno tipo A (CE-A) es un píxel de contorno. En contraparte, un elemento de contorno tipo B (CE-B) es un segmento. Si únicamente se consideran estos dos tipos de elementos para la construcción de pares ordenados, existen tres casos de estudio: dos píxeles de contorno, un píxel y un segmento de contorno, y finalmente dos segmentos de contorno. La primera posible combinación es más difícil de presentarse en una figura digital discreta.

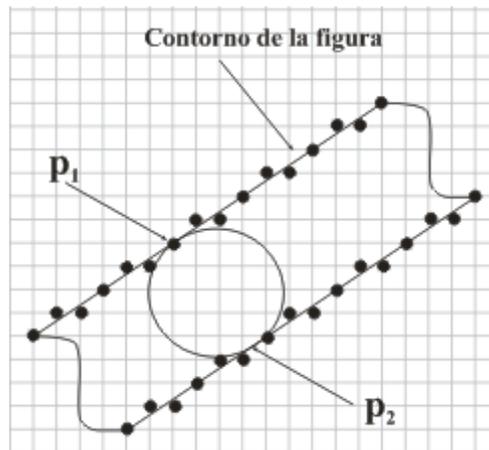


Figura 3–12. Línea recta discretizada donde no es posible encontrar para un punto cualquiera su simetría local.

Con lo anterior, es posible definir el término de Simetría Local Discreta (*Discrete Local Symmetry - DLS*): Un píxel y un segmento de contorno forman una simetría local discreta si se cumplen estas dos condiciones:

1. El círculo circunscrito al triángulo  $T$  formado por la unión del píxel con ambos extremos del segmento de contorno, no contiene en su interior ningún otro píxel de contorno visible desde cualquiera de los vértices de  $T$ .
2. El triángulo  $T$  se encuentra completamente dentro del cuerpo geométrico.

Esto se puede consultar en la Figura 3–13. Conceptualmente, una simetría local discreta es equivalente a una simetría local para el caso continuo. Conforme el segmento de contorno tiende

a cero, se convertirá en un píxel de contorno, quedando la simetría discreta conformada por dos píxeles de contorno, lo cual claramente es idéntico a la definición establecida para una simetría local suavizada.

Para poder obtener las DLS se emplea la técnica de Triangulación de Delaunay Restringida (*Constrained Delaunay Triangulation – CDT*). Los contornos de las figuras en una imagen digital pueden ser representados por un Grafo Plano de Líneas Rectas (*Planar Straight – Line Graph – PSLG*) donde los vértices y bordes son píxeles y segmentos de contorno respectivamente. Dado un grafo  $G = (V, E)$  que describe un contorno, su triangulación es el grafo  $G' = (V, E')$ , donde  $E \subseteq E'$ , tal que no es posible agregar ningún borde, el cual una dos vértices de  $V$  al grafo  $G'$  sin intersectar un borde existente. La triangulación de Delaunay de un grafo  $G$  es su grafo de triangulación donde los círculos circunscritos de cada triángulo no contienen en su interior ningún otro vértice de  $G$  que sea visible desde cualquier vértice del triángulo.

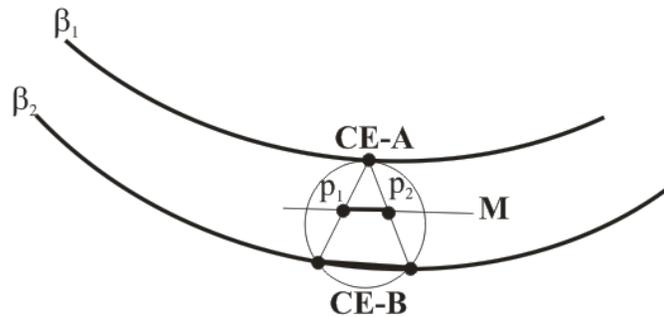


Figura 3–13. Dos elementos de contorno CE-A y CE-B de una simetría local discreta.

Se dice que dos vértices del  $G$  son visibles desde cualquier otro, si un segmento de línea recta que conecte ambos vértices no intersecta un borde de  $G$  en un punto interior. Comparando las definiciones de CDT y DLS, se puede observar que un triángulo de CDT de una imagen es un triángulo simétrico si se cumple que:

1. Al menos uno de sus bordes es del tipo CE-B, y
2. Queda completamente dentro del fondo de la imagen.

A partir de los triángulos obtenidos mediante la técnica de Delaunay se obtiene el esqueleto de un objeto.

### 3.4 Ontologías

Anteriormente, el término ontología se empleaba únicamente dentro de la Filosofía, pero su uso ha crecido significativamente dentro de la Inteligencia Artificial, la Lingüística Computacional y la Teoría de Base de Datos. Se ha reconocido su importancia y utilidad en campos tales como la ingeniería del conocimiento, representación del conocimiento, modelado cualitativo, ingeniería del lenguaje, diseño de base de datos, modelado de información, integración de la información, análisis orientado a objetos, obtención y extracción de información, medicina, comercio electrónico, sistemas de información geográfica, entre otros (Guarino, 1998).

Este término es usado indistintamente por muchos autores, lo cual puede causar una ambigüedad en su interpretación; además, no existe un consenso o definición general aceptada por la comunidad científica de qué es una ontología. Por ello, Corcho realizó una investigación de las diferentes definiciones que se han propuesto para ontología (Corcho et al., 2003), las cuales son:

- La palabra ontología fue tomada de la Filosofía; dentro de esta rama del conocimiento una ontología es la explicación sistemática del ser.
- Neches: Una ontología define los términos básicos y relaciones que comprenden o abarcan el vocabulario de un tópico de determinada área, así como las reglas para combinar los términos y relaciones para definir extensiones al vocabulario (Neches et al., 1991).
- Gruber: Una ontología es una especificación explícita de una conceptualización (Gruber, 1993).
- Borst: Las ontologías se definen como una especificación formal de una conceptualización compartida (Borst, 1997).
- Studer, et al.: Una conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo donde se han identificado los conceptos relevantes de dicho fenómeno (Studer et al., 1998).
- Guarino, et al.: Una ontología es una teoría lógica que da una cuenta parcial explícita de una conceptualización (Guarino et al., 1995).

- Bernaras, et al.: Una ontología proporciona los medios para describir explícitamente una conceptualización detrás del conocimiento representado en una base de conocimientos (Bernaras, 1996).
- Swartout: Una ontología es un conjunto de términos estructurado jerárquicamente para describir un dominio que puede ser usado como esqueleto para una base de conocimientos (Swartout et al., 1997).
- Uschold y Jasper: Una ontología puede tomar una gran variedad de formas, pero necesariamente debe incluir un vocabulario de términos y alguna especificación de su significado (Uschold & Jasper, 1999).

De estas definiciones, uno de los pocos autores, si no el único en proponer una definición matemática formal de ontología es Guarino, quien establece primeramente una separación en la interpretación del término ontología (Guarino, 1998). Desde el punto de vista filosófico, una ontología es un sistema particular de categorías que da cuenta de una cierta visión del mundo. Como tal, este sistema no depende de un lenguaje particular.

Por otra parte, en el campo de la Inteligencia Artificial, una ontología se refiere a un objeto de ingeniería constituido por un vocabulario específico, usado para describir cierta realidad, más un conjunto de suposiciones explícitas en relación con el significado previsto en las palabras del vocabulario. Usualmente, este conjunto de suposiciones tienen la forma de lógica de primer orden, donde las palabras del vocabulario aparecen como nombres de predicado unarios o binarios, llamados respectivamente, conceptos y relaciones. En el caso más simple, una ontología describe una jerarquía de conceptos vinculados por relaciones, pero en casos más complicados, se agregan axiomas para expresar otras relaciones entre los conceptos y para restringir su interpretación.

Genesereth y Nilsson definen una conceptualización como una estructura  $\langle \mathbf{D}, \mathbf{R} \rangle$ , donde  $\mathbf{D}$  es un dominio y  $\mathbf{R}$  es un conjunto de relaciones relevante en  $\mathbf{D}$  (Genesereth & Nilsson, 1987). Esta definición sirvió como base a Gruber para definir una ontología como “una especificación de una conceptualización”. Sin embargo, Guarino comenta que esta noción de conceptualización refleja un estado particular de un evento (Guarino, 1995). Por ejemplo, se puede considerar un conjunto

de bloques sobre una tabla como se muestra en la Figura 3–14. En este caso, la conceptualización podría ser la siguiente:  $(\{a, b, c, d, e\}, \{en, sobre, vacío, tabla\})$

Donde  $\{a, b, c, d, e\}$  es un conjunto llamado universo de discurso, mientras que  $\{en, sobre, vacío, tabla\}$  es el conjunto de relaciones entre estos bloques, donde las dos primeras son relaciones binarias y las últimas dos son unarias. Sin embargo, si la disposición de los bloques en la Figura 3–14 fuera diferente, se podría obtener una diferente conceptualización, por lo que la definición dada por Genesereth y Nilsson puede ser más adecuada para representar un estado del evento que una conceptualización.

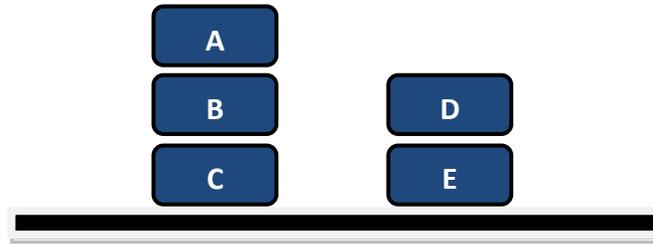


Figura 3–14. Un estado particular de un conjunto de bloques sobre una mesa.

Para poder comprender la definición formal de ontología propuesta por este autor es necesario citar previamente otras de sus definiciones.

*El espacio de dominio es una estructura del tipo  $\langle D, W \rangle$ , donde  $D$  es un dominio y  $W$  es el máximo conjunto de estados de tal dominio (también conocido como mundos posibles).*

*Dado un espacio de dominio  $\langle D, W \rangle$ , una relación conceptual  $n$ -aria sobre  $\langle D, W \rangle$  es una función total del tipo  $\rho^n: W \rightarrow 2^{D^n}$ .*

*Una conceptualización para  $D$  es una terna del tipo  $\mathcal{C} = \langle D, W, \mathfrak{R} \rangle$ , donde  $\mathfrak{R}$  es un conjunto de relaciones conceptuales en el espacio de dominio  $\langle D, W \rangle$ .*

*Sea  $L$  un lenguaje lógico con vocabulario  $V$ ; un modelo para  $L$  es una estructura del tipo  $\langle S, I \rangle$ , donde  $S = \langle D, R \rangle$  es una estructura del mundo e  $I = V \rightarrow D \cup R$  es una función de interpretación que asigna elementos de  $D$  a los símbolos constantes de  $V$ , así como elementos de  $R$  a los predicados de símbolos de  $V$ .*

Una interpretación intencional se define como una estructura del tipo  $\langle C, \mathfrak{I} \rangle$ , donde  $C = \langle D, W, \mathfrak{R} \rangle$  es una conceptualización y  $\mathfrak{I}: V \rightarrow D \cup \mathfrak{R}$  es una función que asigna elementos de  $D$  con símbolos constantes de  $V$ , y elementos de  $\mathfrak{R}$  a símbolos de predicado de  $V$ . Esta interpretación intencional es el compromiso ontológico para  $L$ .

Una ontología es un conjunto de axiomas lógicos diseñados para dar cuenta de un significado intencional de un vocabulario. Dado un lenguaje  $L$  con un compromiso ontológico  $K$ , una ontología para  $L$  es el conjunto definido de axiomas tales que el conjunto de sus modelos aproxima de la mejor forma posible el conjunto de modelos intencionales de  $L$  acorde a  $K$ .

Por otra parte, Guarino menciona que es posible definir diferentes tipos de ontología acordes, a su nivel de generalización, como se muestra en la Figura 3–15.

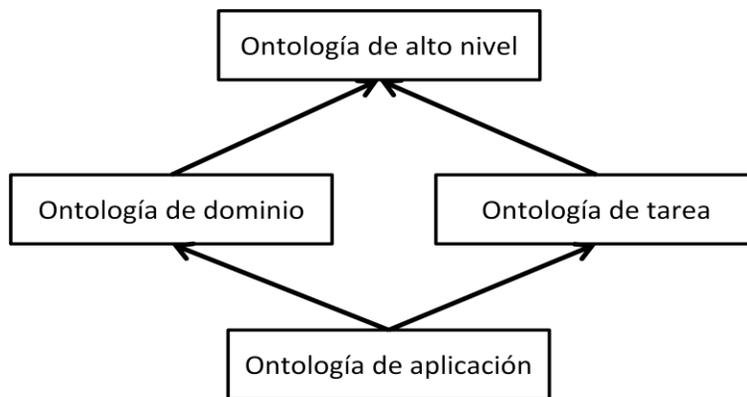


Figura 3–15. Tipos de ontologías, acorde a su nivel de generalización.

- Ontologías de alto nivel. Describen conceptos muy generales, tales como espacio, tiempo, materia, objeto, evento, acción, etc., los cuales son independientes de un problema o dominio en particular; en consecuencia, para comunidades grandes de usuarios, es razonable contar con ontologías unificadas de alto nivel.
- Ontologías de dominio y de tarea. Describen el vocabulario relacionado con un dominio genérico (como medicina, técnica automotriz), o una tarea o actividad genérica (como diagnóstico), respectivamente, por medio de una especialización de los términos introducidos en una ontología de alto nivel relacionada.

- Ontologías de aplicación. Describen conceptos que dependen tanto de un dominio como de una tarea en particular. Estas ontologías son más especializadas que las citadas anteriormente.

Por todo lo anterior, se puede concluir que una ontología es un mecanismo formal de describir y compartir conocimiento y que se caracteriza por tener conceptos y relaciones. La generación de una ontología es un proceso que inicia con una conceptualización, que es una abstracción de algún evento o fenómeno de la realidad. La conceptualización generalmente es expresada en términos de lenguaje natural, por lo que descrita en estos términos, el conocimiento podría no ser entendible por todas las personas. Al describir una ontología de dicha conceptualización, se obtienen los componentes vitales de la conceptualización: conceptos y relaciones, generando una forma estandarizada de describir una conceptualización que puede ser comprendida sin ningún inconveniente. Para el resto del trabajo, entenderemos que una ontología es aquella que cumple con la Definición 3–3.

**Definición 3–3.** Una ontología es un esquema que permite representar una conceptualización, integrado por un vocabulario y de los siguientes elementos básicos:

- Concepto. Es una palabra específica del vocabulario empleado en la ontología, que se refiere a una entidad física o abstracta de nuestra realidad y que tiene significado propio.
- Clase. Es un concepto general dentro de un determinado contexto que permite agrupar diferentes conceptos que tienen características comunes. Por ejemplo, el concepto carretera puede interpretarse como una clase, ya que agrupa diferentes tipos como son carretera federal o carretera de cuota.
- Contexto. Es un concepto que por su naturaleza permite delimitar el espacio de dominio. Como ejemplos de contexto en el área de la geocomputación tenemos vías de comunicación, hidrología, área urbana, etc.
- Relaciones. Es un concepto que describe la forma de vincular diferentes clases o conceptos, por ejemplo: *es un, contiene*.
- Axiomas. Es una proposición clara y evidente que no necesita demostración.

- Restricciones. Son axiomas que restringen la interacción de las instancias de la ontología.
- Atributos. Son las propiedades de un concepto que en su conjunto permiten caracterizar y distinguir de manera única las instancias de una clase.
- Instancias. Es un caso u ocurrencia de una clase.

Pero posiblemente surge la interrogante: ¿Por qué usar una ontología?<sup>12</sup> Existen muchas razones que permiten justificar el uso de una ontología en este trabajo, tal como han expresado muchos autores. Fonseca comenta que el desarrollo de ontologías del mundo geográfico (geo-ontologías) es importante para poder compartir datos geográficos entre diferentes comunidades de usuarios (Fonseca, 2002). Por otra parte, Uschold parte del hecho que las personas, organizaciones y sistemas de software deben comunicarse entre ellos mismos (Uschold & Gruninger, 1996). Sin embargo, debido a las diferentes necesidades y contextos de fondo, cada uno puede tener sus puntos de vista y suposiciones acerca del mismo tema. Asimismo, cada uno usa diferente terminología técnica, por lo que los conceptos pueden ser diferentes, traslaparse o desconocerse. En consecuencia, esto ocasiona:

- Una pobre comunicación dentro de una organización y entre las personas.
- En el contexto del diseño de un sistema de información, se dificulta la identificación de requerimientos y la definición de la especificación de un sistema.
- Los diferentes métodos de modelado, paradigmas, lenguajes y herramientas de software limitan seriamente la interoperabilidad y el potencial de compartir y reutilizar un sistema.

Estos problemas pueden ser minimizados mediante el uso de ontologías, ya que con ello se puede mejorar la comunicación, interoperabilidad e ingeniería de sistemas, su especificación, confiabilidad y reusabilidad (Uschold & Gruninger, 1996). Natalya Noy también expone diferentes razones para desarrollar y usar una ontología (Noy & McGuinness, 2001):

---

<sup>12</sup> En el siguiente capítulo se muestra el uso de una ontología para resolver ambigüedades de clasificación.

- Compartir el entendimiento común de la estructura de información entre personas o agentes de software.
- Permitir la reutilización del conocimiento de un dominio.
- Explicitar suposiciones de un dominio.
- Separar el conocimiento del dominio del conocimiento operacional.
- Analizar el conocimiento dentro de un dominio.

Adicionalmente, una ontología puede extenderse para agregar más conceptos, clases y relaciones. Al incrementar el nivel de profundidad de una ontología, se obtiene un mayor detalle de los conceptos descritos, con lo cual, el nivel de descripción de la ontología, se acerca más a la forma en que una persona percibe la realidad.

En lo que concierne a las herramientas para construir ontologías existen también una gran variedad de propuestas. Algunos de los programas más utilizados para definir ontologías son **(Corcho et al., 2003)**: DUET, IOLEd, Onto Edit Professional, Ontolingua, OntoSaurus, Protégé 2000, WebODE, WebOnto. De ellos, solamente Ontosaurus y Protégé 2000 usan una política de código libre, pero OntoSaurus tiene el inconveniente de usar una arquitectura cliente / servidor, mientras que Protégé está basado en Java y puede ser descargado e instalado. Por ello, se usará Protégé 2000 para desarrollar la ontología empleada en este trabajo.

Finalmente, los principales lenguajes que permiten describir una ontología son: KIF, Ontolingua, OCML, LOOM, FLogic, XOL, SHOE, RDF(s), OIL, DAML y OWL. Protégé permite exportar a una gran variedad de lenguajes, entre los que podemos citar XML, RDF, Esquema XML, FLogic, CLIPS, y Java HTML. Dentro de nuestra rama, numerosos investigadores prefieren usar OWL como lenguaje para describir ontologías, por lo cual, en este trabajo también se hará uso de OWL.

### 3.5 Árboles dinámicos

El uso de árboles dinámicos está creciendo en respuesta a los problemas de segmentación de imágenes. Uno de los principales motivos es la necesidad de construir modelos de dichas imágenes. Las técnicas de modelado tradicional tienden a ser no-probabilísticas; en consecuencia son un tanto inflexibles, aunque se puede alcanzar un desempeño adecuado si los métodos intentan representar o describir la estructura de la imagen. Por otra parte, los modelos probabilísticos comúnmente toman la forma de campos aleatorios de Markov, los cuales poseen una estructura plana, y aunque capturan algo de la coherencia local de la imagen, no cuentan con una estructura jerárquica. Las estructuras basadas en árboles prometen proporcionar dicha jerarquía, al mismo tiempo que permiten capturar de alguna forma el concepto de un objeto (Storkey Amos, 2007).

Los modelos probabilísticos de árboles cuaternarios fueron desarrollados por Chris Williams y Xiaojuan Feng. Los nodos-hoja de un árbol cuaternario representan las etiquetas de las clases locales de una imagen (para la segmentación). Usando una red neuronal se modela el mapa de probabilidades de las clases de los nodos-hoja a los píxeles de la imagen; esto se puede apreciar en la Figura 3–16.

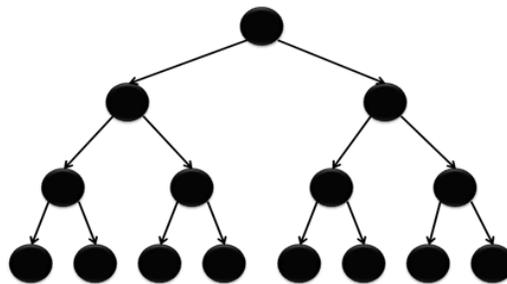


Figura 3–16. Ejemplo de un árbol cuaternario.

Los árboles cuaternarios tienen una ventaja en su estructura de árbol. Dado un conjunto de datos, la propagación de creencias puede usarse para obtener la probabilidad marginal de cada nodo, para así determinar la probabilidad (*likelihood*) del árbol. Los resultados prácticos son aceptables en problemas reales de segmentación, pero su principal desventaja es que la estructura es fija. Por esto, los árboles cuaternarios producen objetos grandes en la segmentación debido al hecho de

que dos nodos-hoja adyacentes (espacialmente) pueden estar conectados por un camino, solamente a través de un vértice superior lejano en la jerarquía de árbol.

Una forma de dar solución a esto es usando árboles dinámicos, ver Figura 3–17. Son una combinación de estructuras de árbol la cual permite que cada vértice pueda elegir a su nodo padre. Esto reduce el problema citado anteriormente para los árboles cuaternarios, debido a que dos nodos hoja pueden estar conectados a cualquier nivel de la jerarquía.

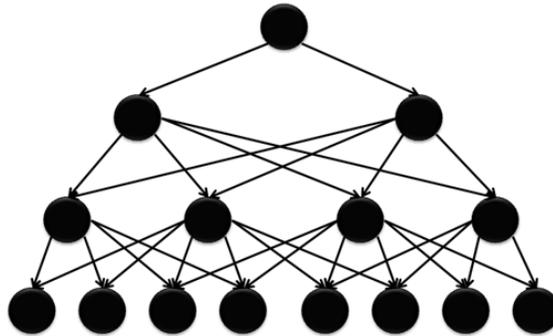


Figura 3–17. Ejemplo de árbol dinámico.

En teoría de grafos, un **árbol** es un grafo en el que dos vértices están conectados por un solo camino. Un **bosque** es un grafo en el que dos vértices cualquiera están conectados como máximo por un camino. Formalmente, se puede decir que un **árbol** es un grafo simple unidireccional  $G$  que satisface alguna de las siguientes condiciones equivalentes (Wikipedia, 2007):

- $G$  es conexo y no tiene ciclos simples.
- $G$  no tiene ciclos simples y, si se añade algún arco se forma un ciclo simple.
- $G$  es conexo y si se le quita algún arco deja de ser conexo.
- $G$  es conexo y el grafo completo de 3 vértices  $K_3$  no es un menor de  $G$ .
- Dos vértices cualquiera de  $G$  están conectados por un único camino simple.

Si  $G$  tiene muchos vértices ( $n$ ) entonces las definiciones anteriores son también equivalentes a cualquiera de las siguientes condiciones:

- $G$  es conexo y tiene  $n - 1$  arcos.

- $G$  no tiene arcos simples y tiene  $n - 1$  arcos.

Un grafo unidireccional simple  $G$  recibe el nombre de **bosque** si no tiene ciclos simples. Un árbol direccional es un grafo direccional que sería un árbol si se hiciera caso omiso de las direcciones de los arcos. Algunos autores restringen la frase al caso en el que todos los arcos se dirigen a un vértice particular, o todas sus direcciones parten de un vértice particular.

Un árbol recibe el nombre de árbol con raíz si cada vértice ha sido designado raíz, en cuyo caso los arcos tienen una orientación natural hacia o desde la raíz. Los árboles con raíz, a menudo con estructuras adicionales como orden de los vecinos de cada vértice, son una estructura clave en informática. Un árbol etiquetado es un árbol en el que cada vértice tiene una única etiqueta. Los vértices de un árbol etiquetado de  $n$  vértices reciben normalmente las etiquetas  $\{1, 2, \dots, n\}$ . Un árbol regular (homogéneo) es un árbol en el que cada vértice tiene el mismo grado.

Mientras que otros algoritmos clásicos del PDI se basan en usar árboles regulares, el algoritmo propuesto emplea un árbol dinámico de datos, el cual sirve para describir los segmentos que componen una geoimagen. Dichos segmentos son descritos por medio de sub-árboles (ramas) con arcos indexados; cuando se disuelve un segmento solamente es necesario eliminar los arcos cuyos índices son mayores al nivel  $i$  en cuestión. Como definición de árbol dinámico se puede tomar como base la establecida en [\(Adams & Williams, 2003\)](#).

*Un árbol dinámico es un grafo dirigido, integrado por un conjunto de nodos padre y nodos hijos, en el que cada nodo hijo puede estar conectado a un nodo padre en un nivel superior, donde para cada nivel del árbol existe un nodo padre nulo, de tal forma que cada nodo hijo conectado a dicho padre nulo puede convertirse en una nueva raíz del árbol.*

## Capítulo 4

# Metodología propuesta

### ***Resumen***

En este capítulo se describe el algoritmo general para la descomposición semántica de geoimágenes. Se describe el dominio del problema y restricciones de las geoimágenes que pueden usarse en la metodología de descomposición semántica, así como los fundamentos matemáticos del algoritmo de compresión semántica. Posteriormente, se detalla el funcionamiento de la metodología desarrollada, la cual está integrada por dos grandes etapas: análisis y síntesis. Finalmente, se explica la forma de resolver el problema de la especialización de los objetos geográficos mediante el uso de una ontología de aplicación del dominio geográfico.



## 4.1 Introducción

En este capítulo se presentan los detalles concernientes al algoritmo desarrollado en la presente tesis para realizar descomposición semántica de geoimágenes. Se comentarán algunos detalles de la metodología propuesta y posteriormente se explicarán las definiciones del algoritmo. La metodología está dividida en dos grandes etapas: análisis y síntesis. La primera tiene el objetivo de reducir el detalle de la geoimagen original para que su procesamiento posterior sea más sencillo, mediante un algoritmo de compresión semántica. La salida de esta etapa es un conjunto de objetos agrupados por intensidades de color uniforme. La etapa de síntesis persigue extraer, reconocer y especializar estos objetos, asociando etiquetas en lenguaje natural a cada uno de ellos. El conjunto resultante de etiquetas de la etapa de síntesis es la semántica de la geoimagen original.

## 4.2 Dominio del problema y restricciones

El dominio del problema está relacionado con la definición del tipo de información que un determinado algoritmo puede procesar. Con la definición de geoimagen citada en el capítulo anterior, se permite trabajar con un conjunto particular de datos geográficos. Sin embargo, aquella definición no es suficiente, ya que como se verá en esta sección, la información geográfica puede ser muy variable y con ello, afectar la eficiencia de la metodología, por lo cual, se requiere restringir correctamente el dominio del problema<sup>13</sup>.

Cada tipo de objeto geográfico se puede describir mediante un patrón el cual está formado por sus características de bajo nivel. Por ejemplo, una imagen de un bosque generalmente contiene valores de intensidad en un cierto rango del espacio RGB; lo mismo sucede con un cuerpo hidrológico, etc., ver Figura 4-1.

---

<sup>13</sup> Lo expuesto en esta sección pretende cubrir la mayor parte de los casos que pueden afectar las características de una geoimagen; sin embargo, pueden existir casos no contemplados que también pueden tener consecuencias en la eficiencia de la metodología propuesta.



Figura 4–1. Ejemplo de una geoimagen.

En la Figura 4–1 se pueden identificar tres tipos de objetos geográficos: nubes, áreas verdes y ríos. Uno de los problemas que se puede detectar en esta geoimagen es la presencia de las nubes, que son considerados como objetos geográficos dinámicos. Se dice que es dinámico porque en un instante de tiempo  $t_1$  pueden estar presente en un determinado lugar, pero puede existir un tiempo  $t_2$  ( $t_2 > t_1$ ) tal que al volver a visualizar la misma región, dicho objeto no esté presente o que su ubicación haya cambiado. El problema de este objeto geoespacial radica en que elimina información de otros objetos terrestres (que están debajo de la nube) y que podría ser útil o vital para la identificación de los objetos afectados. Por ello, es deseable que las geoimágenes no tengan objetos geográficos dinámicos.

Otro inconveniente es la ocurrencia de un fenómeno natural en una región de estudio. En la etapa de síntesis, el algoritmo de reconocimiento procesa cada objeto geográfico asociándole una etiqueta general; esto se consigue buscando en su base de conocimientos el patrón característico con el que el objeto geográfico tiene mayor grado de semejanza. Generalmente, los algoritmos de reconocimiento se diseñan para permitir variaciones en los vectores característicos, de tal forma que sigan reconociendo con éxito un determinado objeto. Sin embargo, conforme las variaciones en un vector característico son mayores, se pueden presentar dos casos:

- a. Error de clasificación. Sucede cuando un vector  $\bar{X}$  es clasificado en la clase  $\alpha_i$ , a pesar de que pertenece a la clase  $\alpha_j$ ; es decir, el vector  $\bar{X}$  tiene mayor similitud con el prototipo de la clase  $\alpha_i$  que con el de la clase  $\alpha_j$ .

- b. No se reconoce el objeto. En este caso, no existe ninguna clase con la que pueda ser asociado el vector  $\bar{X}$ .

Los fenómenos naturales (terremotos, inundaciones, erupciones volcánicas, huracanes, etc.) pueden ocasionar los dos casos mencionados previamente. Como ejemplo de ello, considérese la geoimagen que aparece en la Figura 4–2. Esta geoimagen corresponde a la misma ubicación geográfica de la geoimagen que aparece en la Figura 4–1. Con base en las características de bajo nivel, las geoimágenes presentan un alto grado de diferencia. Solamente empleando técnicas muy avanzadas de procesamiento sería posible determinar que ambas geoimágenes representan los mismos objetos geográficos. Dadas las características de la geoimagen de la Figura 4–2, lo más probable es que solamente se identifique un solo objeto geográfico, lo cual no es un error del sistema, ya que un programa de cómputo se basa en la información de la imagen que se está analizando, y en este caso nosotros estamos haciendo uso de información previa que no contiene la imagen de la Figura 4–2.

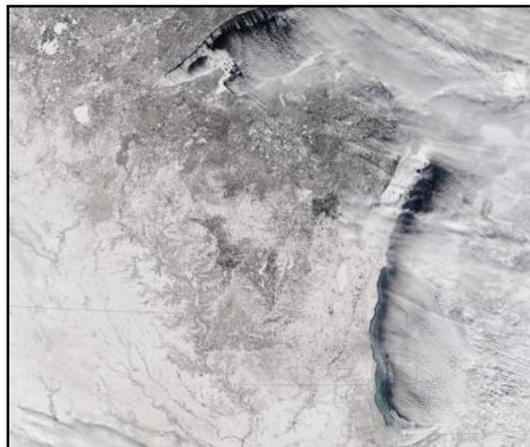


Figura 4–2. Ejemplo de una geoimagen donde las características de bajo nivel de los objetos geográficos están alteradas por un fenómeno natural.

Sin detallar excesivamente, otras situaciones que pueden producir resultados adversos son:

1. Iluminación de los objetos geográficos. La mayor parte de las imágenes satelitales son capturadas con sensores pasivos, por lo que se requiere de la presencia de un medio de iluminación natural o artificial. La cantidad de energía necesaria para poder iluminar

artificialmente una región geográfica es muy elevada; en consecuencia, el uso de la energía solar como medio de iluminación es la opción más conveniente. Sin embargo, la cantidad de energía solar en una región específica es variable y se ajusta al factor temporal, así como a las condiciones climáticas. Por lo anterior, estos dos parámetros están directamente relacionados con las características de una geoimagen.

2. Objetos geográficos representados parcialmente. En este caso, la región geográfica del objeto no es capturado en su totalidad en la geoimagen. Cuando el porcentaje faltante es relativamente pequeño existe alta probabilidad de reconocer correctamente dicho objeto geográfico; sin embargo, pueden existir geoimágenes donde uno o más objetos geográficos no puedan ser reconocidos exitosamente, ya que los datos de estos objetos es mínima.

Por todo lo anterior, el dominio del problema de la metodología aquí propuesta está integrado por geoimágenes que cumplan con las siguientes restricciones:

- a. Condiciones de iluminación adecuadas que permitan preservar las características de bajo nivel típicas de los objetos geográficos.
- b. Ausencia de objetos geográficos dinámicos.
- c. Que las características topológicas, geométricas, de textura y de intensidad de los objetos geográficos, no estén alteradas por la presencia de fenómenos naturales<sup>14</sup>.
- d. Si se está interesado en uno o más objetos geográficos en particular, que la región geográfica que delimita a dicho objeto esté capturada en su totalidad en la geoimagen.

Antes de describir la compresión semántica se requieren dos definiciones importantes, ya que son citadas ampliamente en el resto del capítulo.

**Definición 4–1.** Un segmento, es un conjunto de píxeles que se denotará por  $s$  y está compuesto por una serie de  $n$  puntos, es decir:

---

<sup>14</sup> Esto queda a juicio del usuario, ya que solamente él puede conocer de antemano si existe la posibilidad de que los objetos geográficos fueron afectados por un fenómeno natural.

$$s = \{p_1(x, y), p_2(x, y), \dots, p_n(x, y)\} \text{ ó} \quad (4.1)$$

$$s = \{p_1, p_2, \dots, p_n\}$$

**Definición 4–2.** Se dice que dos segmentos  $s_1, s_2$  son adyacentes si y solo si existe al menos un punto del primer segmento que forma parte del conjunto de  $k$ -vecinos del segundo segmento, esto es:

$$s_1 \text{ adyacente } s_2 \text{ si y solo si } \{\exists(p_1 \in s_1 \text{ y } p_2 \in s_2) | (p_2 \in N_k(p_1))\} \quad (4.2)$$

Donde  $k$  es el tipo de vecindad entre pixeles y que puede tomar el valor 4, 8, 6R ó 6L. Para los fines de este trabajo se usó  $k = 8$ .

### 4.3 Compresión semántica

Los objetos geoespaciales se caracterizan por presentar altas variaciones en sus características de bajo nivel, por lo que al aplicar las técnicas tradicionales de segmentación, comúnmente se producen imágenes con subsegmentación o sobresegmentación, esto es, se detectan menos clases o más clases que las deseadas o de las que se obtendrían realizando una segmentación manual. Por ejemplo, considerando la imagen de la Figura 4–3(a), se puede aplicar un agrupamiento  $k$ -medias indicando un total de 16 clases usando la herramienta comercial PCI-Geomática<sup>15</sup>, con la que se obtienen los resultados que aparecen en la Figura 4–3(b).

Al analizar nuevamente el ejemplo de la Figura 4–3, surgen dos interrogantes:

1. ¿Cómo elegir el número adecuado de clases para un algoritmo de clasificación?
2. ¿La correspondencia entre los pixeles agrupados y la información original es adecuada?

Inicialmente, puede surgir la idea de tratar de usar un algoritmo avanzado de segmentación, el cual mediante un análisis profundo de la información geográfica trate de encontrar el número de clases más conveniente según la cantidad de objetos geográficos que están en la geoimagen. Pero

<sup>15</sup> Para todos los casos donde sea mencionado, el resultado de hacer un agrupamiento  $k$ -medias,  $c$ -medias difuso o ISODATA fue obtenido con este programa de cómputo.

como se expresó previamente, no se ha podido encontrar una metodología adecuada especialmente para las imágenes que describen objetos geográficos, debido a la complejidad y heterogeneidad de la información contenida en las geoimágenes.

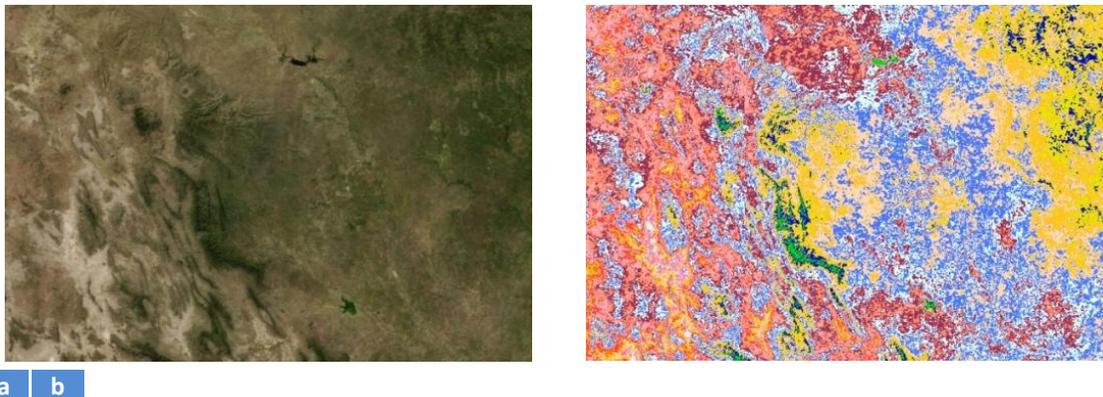


Figura 4–3. Ejemplo de una geoimagen (a) Resultado de la clasificación usando el algoritmo k-medias con la herramienta comercial PC-Geomática.

Los problemas de los trabajos citados en el Capítulo 2, radican en el hecho de que emplean algunas o una sola característica para analizar las imágenes digitales y están enfocados a realizar procesamiento con el color. Sin embargo, emplear una sola característica como criterio de segmentación, no es suficiente sobre todo al trabajar con geoimágenes, debido a que dos regiones pueden no tener similitud en su intensidad (por lo que no se deben agrupar), pero sí en otra propiedad como su tamaño, área, etc. en cuyo caso sí deben fusionarse, pero este tipo de propiedades no se puede medir empleando el color de los píxeles.

Dado un conjunto de objetos geográficos, el proceso de cuantificación de estas propiedades (tamaño, área, largo, ancho) es sencillo solamente cuando los objetos geográficos están completamente aislados, o bien, se pueden detectar los contornos que delimitan las regiones de cada objeto. Sin embargo, en una geoimagen todos sus elementos discretos tienen que ser asociados a un cierto objeto geográfico tratando de respetar las características geométricas y topológicas de los objetos geoespaciales, es decir, hacer una partición de la geoimagen.

Realizar la partición del contenido geográfico de una región es precisamente el problema al que más investigadores han tratado de dar solución, porque existen varias interrogantes al respecto:

1. ¿Cómo delimitar un objeto geográfico?
2. ¿Cómo medir el tamaño (largo, ancho, área) de cada objeto?

Por todo lo anterior, este trabajo propone un **algoritmo de compresión semántica**. Este algoritmo cuantifica diferentes características de un conjunto de segmentos en un espacio isotrópico que se representa mediante un árbol dinámico, partiendo del hecho de que el número de segmentos es igual al número de elementos discretos presentes en la geoimagen (píxeles). Esta suposición tiene que hacerse por el hecho de que no se tiene conocimiento sobre la estructura o forma de los objetos geográficos. Cuando se cumple una condición de adyacencia y que una propiedad o característica es similar entre un par de segmentos, se procede a realizar la fusión de los segmentos en el árbol dinámico. El término de **compresión semántica** para este algoritmo, se justifica por el hecho de que se reduce la cantidad inicial de segmentos, describiendo las regiones resultantes por medio de colores uniformes; esto significa que si un conjunto de píxeles vecinos tienen el mismo color, se interpretará que estos píxeles pertenecen a la misma región.

Este algoritmo persigue encontrar una representación de una geoimagen por medio de una jerarquía dinámica y compacta de objetos. Esta representación consiste en diversos segmentos de imagen que difieren en forma y tamaño y está diseñada para poderse ejecutar tanto en un modo automático como en uno iterativo. Mediante la ejecución iterativa del algoritmo se realiza una fusión de los segmentos de imagen semejantes con base en ciertas propiedades y atributos.

#### 4.3.1 Jerarquía compacta de los segmentos de una imagen

Se mencionó anteriormente que el algoritmo de compresión semántica se basa en generar y actualizar un árbol dinámico que describe los segmentos de la geoimagen. Esta jerarquía de segmentos se genera por medio de una integración paso a paso de las áreas semejantes, es decir, de manera iterativa. La selección del criterio que permite fusionar los segmentos está limitada a un conjunto de características que se calculan para cada segmento y que sirve para hacer una comparación entre segmentos vecinos.

### 4.3.2 Propiedades y atributos de la compresión semántica

Las características de los segmentos de una geoimagen pueden dividirse en dos grupos: atributos y propiedades. Los atributos pueden entenderse como un conjunto primario de características de los segmentos, estimados y almacenados dinámicamente, para todos los segmentos a cualquier nivel de la representación de una geoimagen. Por otro lado, las propiedades pueden ser vistas como características numéricas de un segmento, que se obtienen como salida de la conversión de datos y son seleccionadas como una función de la etapa de procesamiento y del contexto del problema.

El conjunto de características que pueden ser de utilidad en nuestro contexto, ordenados ascendentemente con base en la complejidad de sus cálculos, es el siguiente:

1. Características globales (para la imagen completa).
2. Características locales (dentro de la vecindad de un segmento).
3. Intensidad integral (la suma de las intensidades).
4. Número de píxeles.
5. Momentos de primer o segundo orden (calculados relativamente al origen de un segmento).
6. Perímetro no aditivo.
7. Descripción de segmentos adyacentes por medio de una relación binaria.

En concreto, esta lista proporciona una estimación de las propiedades geométricas y de intensidad entre las que podemos mencionar: rango de intensidades de los píxeles, intensidad promedio, momentos invariantes, etc. Para poder determinar las expresiones numéricas que nos permitirán obtener las propiedades de los segmentos para la construcción de la jerarquía de los objetos, es necesario trabajar en un plano diferente por medio de una transformación de los objetos que permita convertirlos en objetos isotrópicos<sup>16</sup> (Levachkine, 2003), (Levachkine et al., 2001).

---

<sup>16</sup> En física, la isotropía se refiere al hecho de que ciertas magnitudes vectoriales medibles dan resultados idénticos con independencia de la dirección escogida para la medida. Cuando una determinada magnitud no

Suponiendo que es posible detectar la orientación de un objeto, entonces es posible definir un nuevo sistema de coordenadas adecuado al objeto. Usando este sistema de coordenadas, se puede realizar una igualación de las escalas de los ejes. En dicho sistema de coordenadas, los objetos tipo área están caracterizados por una variable invariante, igual a la raíz media cuadrática del tamaño.

Si un conjunto de objetos puntuales no se intersectan con una línea recta, éstos formarán una figura no degenerativa, la que por medio de una transformación lineal se convierte en una figura isotrópica que tiene un valor uniforme (para la raíz media cuadrática del tamaño). Una figura isotrópica formada por  $n$  puntos está definida por una regla: que la raíz media cuadrática de la distancia de los puntos de la figura isotrópica al centro de gravedad de cualquier línea, no dependa de la pendiente de la línea.

En un espacio Euclidiano bidimensional  $(u, v)$ , inducido por combinaciones lineales (de columnas) de los sistemas de coordenadas  $(x, y)$ , los momentos de segundo orden calculados relativamente al centro de inercia se obtienen como productos escalares:

$$I_x \equiv (u, u), I_y \equiv (v, v), I_{xy} \equiv (u, v) \quad (4.3)$$

Donde las figuras isotrópicas están representadas como un par de vectores ortogonales de igual longitud o en términos equivalentes de momentos de segundo orden, como se expresa en la siguiente ecuación:

$$\left. \begin{array}{l} (u, v) = 0 \\ (u, u) = (v, v) \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} I_{xy} = 0 \\ I_x = I_y \end{array} \right. \quad (4.4)$$

Introduciendo los parámetros independientes  $\Delta$  y  $\gamma$  que están relacionados con los momentos de segundo orden por las ecuaciones:

---

presenta isotropía decimos que presenta anisotropía. En matemáticas, la isotropía se refiere a una propiedad geométrica de invarianza en una variedad diferenciable [Wikipedia, 2007].

$$\Delta, \gamma = \begin{cases} \cos \Delta = \frac{I_{xy}}{\sqrt{I_x I_y}} \\ \sinh \gamma = -\frac{1}{2} \left( \sqrt{\frac{I_x}{I_y}} - \sqrt{\frac{I_y}{I_x}} \right) \end{cases} \quad (4.5)$$

La expresión del seno hiperbólico puede ser desarrollada usando las propiedades de las raíces, para llegar a una expresión equivalente:

$$\begin{aligned} \sinh \gamma &= \frac{1}{2} \left( \sqrt{\frac{I_y}{I_x}} - \sqrt{\frac{I_x}{I_y}} \right) = \frac{1}{2} \left( \frac{\sqrt{I_y}}{\sqrt{I_x}} - \frac{\sqrt{I_x}}{\sqrt{I_y}} \right) = \frac{1}{2} \left( \frac{\sqrt{I_y} \sqrt{I_y} - \sqrt{I_x} \sqrt{I_x}}{\sqrt{I_x} \sqrt{I_y}} \right) \\ \sinh \gamma &= \frac{1}{2} \left( \frac{I_y - I_x}{\sqrt{I_x I_y}} \right) \end{aligned} \quad (4.6)$$

Recordando que  $(\sin \theta)^2 + (\cos \theta)^2 = 1$  y que  $(\cosh \theta)^2 - (\sinh \theta)^2 = 1$ . Para la variable  $\Delta$  se tiene lo siguiente:

$$\begin{aligned} (\sin \Delta)^2 + (\cos \Delta)^2 &= 1 \\ \sin \Delta &= \sqrt{1 - (\cos \Delta)^2} \\ \sin \Delta &= \sqrt{1 - \frac{(I_{xy})^2}{I_x I_y}} \end{aligned} \quad (4.7)$$

Mientras que para las funciones trigonométricas hiperbólicas:

$$\begin{aligned} (\cosh \gamma)^2 - (\sinh \gamma)^2 &= 1 \\ \cosh \gamma &= \sqrt{1 + (\sinh \gamma)^2} \\ \cosh \gamma &= \sqrt{1 + \frac{1}{4} \frac{(I_y - I_x)^2}{I_x I_y}} \end{aligned} \quad (4.8)$$

Las figuras no degradadas son aquellas que tienen asociado un valor de  $\sinh \gamma$  distinto de cero. Por otra parte, un valor cercano a cero para una nueva variable denotada por  $d$  caracteriza a las figuras isotrópicas, donde el valor de  $d$  está dado por la siguiente ecuación:

$$d = \sqrt{(\cos \Delta)^2 + (\sinh \gamma)^2} \quad (4.9)$$

Cualquier figura no isotrópica no degradada será transformada en una figura isotrópica por medio de una transformación no lineal  $W$  la cual, en el espacio  $u \times v$ , se reduce a una ortogonalización e igualación de las longitudes de los vectores  $u$  y  $v$ , que para el plano inicial  $x \times y$  significa un estiramiento de uno de los ejes principales junto con la compresión del otro eje. La transformación  $W$  está determinada por una ecualización de los momentos  $I_x$  e  $I_y$  y asignando un valor igual a cero a la unión de los momentos, es decir:

$$\left. \begin{aligned} W\{u\} &= e^{\theta}(u \cos \varphi - v \sin \varphi) \\ W\{v\} &= e^{-\theta}(-u \sin \varphi - v \cos \varphi) \end{aligned} \right\} \Rightarrow \begin{cases} \tilde{I}_{xy} = 0 \\ \tilde{I}_x = \tilde{I}_y \end{cases} \quad (4.10)$$

El ángulo  $\varphi$  que determina la transformación de los vectores  $u$  y  $v$  asociado con la rotación del plano inicial, se encuentra por medio de la expresión:

$$\varphi = \begin{cases} \sin 2\varphi = \sigma \frac{\cos \Delta}{d} \\ \cos 2\varphi = -\sigma \frac{\sinh \gamma}{d} \end{cases}, \quad \sigma = \pm 1 \quad (4.11)$$

Estas fórmulas del ángulo  $\varphi$ , determinan la orientación de los ejes de las figuras no isotrópicas relativas a la configuración inicial. En el caso particular de figuras simétricas, nos permite encontrar la dirección de los ejes de simetría.

Determinada ya una expresión para calcular  $\varphi$ , se define ahora el parámetro hiperbólico  $\theta$ , el cual está dado por el cociente:

$$\theta = \tanh 2\theta = -\sigma \frac{d}{\coth \gamma} \quad (4.12)$$

Este parámetro ( $\theta$ ) describe en escala logarítmica el radio de las dimensiones lineales, es decir:

$$\theta = \frac{1}{2} \sigma \ln(\text{ancho} \times \text{largo}) \quad (4.13)$$

Las dimensiones de las figuras se calculan indistintamente a partir de la orientación como la media cuadrática de las distancias de los puntos a los ejes:

$$\begin{aligned} \text{largo}^2 &= \frac{\sqrt{I_x I_y}}{n} (\cosh \gamma + d) \\ \text{ancho}^2 &= \frac{\sqrt{I_x I_y}}{n} (\cosh \gamma - d) \end{aligned} \quad (4.14)$$

El cuadrado del tamaño lineal invariante de una figura isotrópica determina su área, que corresponde con el área de la figura inicial y es igual al producto de la variable *largo* por el de *ancho*, esto es:

$$\begin{aligned} \text{tamaño}^2 &= \text{area} = \frac{\sqrt{I_x I_y}}{n} |\sinh \Delta| \\ \text{tamaño} &= \sqrt{\frac{\sqrt{I_x I_y}}{n} |\sinh \Delta|} \end{aligned} \quad (4.15)$$

El número de puntos, las variables *area*, *largo*, *ancho*, el *tamaño* invariante de una figura y los parámetros trigonométricos e hiperbólicos que permiten transformar las figuras en isotrópicas, se estiman para la geoimagen en su conjunto para una región o al hacer combinación de regiones.

Para establecer que la transformación *W* del espacio bidimensional  $u \times v$  coincide con la transformación del plano inicial  $x \times y$ , es necesario mostrar solamente que el producto de la transformación ortogonal *V* con el conjunto *L* (que es su simétrico conjugado), permite considerar estos planos como un conjunto de vectores. Los puntos de una figura se describen por medio de los vectores  $w_i$ :

$$w_i = x_i e_x - y_i e_y \quad (4.16)$$

Donde  $e_x$  y  $e_y$  son vectores ortonormales definidos como:

$$e_x, e_y = \begin{cases} e_x = \frac{W\{v\}}{\sqrt{\sqrt{I_x I_y} |\sin \Delta|}} \\ e_y = \frac{W\{u\}}{\sqrt{\sqrt{I_x I_y} |\sin \Delta|}} \end{cases} \quad (4.17)$$

Los vectores  $u$  y  $v$  en las bases  $e_x, e_y$  se pueden expresar como se muestra en la Ecuación (4.18):

$$\begin{aligned} u &= -\sqrt{n} \cdot size(e_x \cdot e^\theta \cdot \sin \varphi + e_y \cdot e^{-\theta} \cdot \cos \varphi) \\ v &= -\sqrt{n} \cdot size(e_x \cdot e^\theta \cdot \cos \varphi - e_y \cdot e^{-\theta} \cdot \sin \varphi) \end{aligned} \quad (4.18)$$

La rotación ( $V$ ) y la transformación de Lorentz ( $L$ ) están definidas por:

$$\begin{aligned} V &= \begin{cases} V(e_x) = e_x \cos \varphi - e_y \sin \varphi \\ V(e_y) = -e_x \sin \varphi + e_y \cos \varphi \end{cases} \\ L = L\{w\} &= e^{-\theta}(w, e_x)e_x - e^\theta(w, e_y)e_y \end{aligned} \quad (4.19)$$

Donde  $w$  es un vector arbitrario de los planos  $x \times y$  o  $u \times v$ . Entonces, en el caso de una interpretación pasiva de la transformación de coordenadas  $W$ , se tiene:

$$w_i = x_i e_x - y_i e_y \equiv (w_i, W^{+^{-1}}\{e_x\})W(e_x) + (w_i, W^{+^{-1}}\{e_y\})W(e_y) \quad (4.20)$$

$V$  y  $L$  son las componentes de la descomposición que se desean obtener:

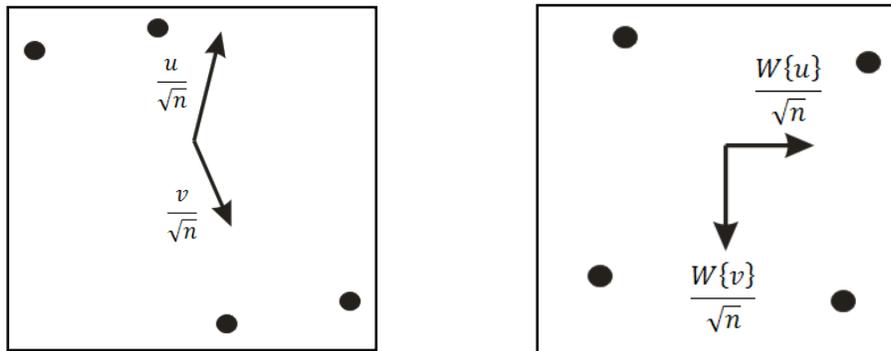
$$W = VL = \begin{cases} V^+ = V^{-1} \\ L^+ = L \end{cases} \quad (4.21)$$

Donde el signo “+” denota el conjugado, mientras que el exponente “-1” designa el inverso. Las nuevas coordenadas  $(w_i, VL^{-1}\{e_x\})$ ,  $(w_i, VL^{-1}\{e_y\})$  se forman como resultado de la deformación del plano y describen las componentes multidimensionales  $W\{u\}$ ,  $W\{v\}$ . Para una interpretación activa de la transformación de coordenadas  $W$ , la expresión anterior cumple con la transformación inversa.

Entonces, cualquier figura no isotrópica que consista en un conjunto de  $n$  puntos en el plano  $x \times y$  se compara en el mismo plano con los pares de vectores  $u, v$  que son tratados como columnas en el mismo espacio de coordenadas ( $u \rightarrow x, v \rightarrow y$ ). Los vectores  $u, v$  son determinados con una exactitud de  $\pi/2$ . Estos vectores se transforman en vectores ortogonales de la misma longitud empleando la transformación que convierte las figuras en isotrópicas, ver Figura 4-4.

En resumen, el conjunto de características y parámetros usado por la compresión semántica, es el siguiente:

1. Característica de *distancia* – ( $d$ ).
2. Característica de *largo* – ( $L$ ).
3. Característica de *ancho* – ( $W$ ).
4. Característica de *tamaño* – ( $Sz$ ).
5. Característica de *área* – ( $S$ ).
6. Característica de *intensidad* – ( $I$ ).
7. Característica de *intensidad promedio* – ( $ABS$ ).
8. Parámetro  $\sin \Delta$  - ( $Ss$ ).
9. Parámetro  $\cos 2\gamma$  – ( $C2f$ ).
10. Parámetro  $\sin 2\gamma$  – ( $S2f$ ).



a b

Figura 4–4. Ortogonalización y homogeneización de la longitud de los vectores  $u, v$  aplicando la transformación para  $n=4$ ; (a) Datos originales; (b) Imagen isotrópica.

### 4.3.3 Construcción de la jerarquía de figuras isotrópicas

Mediante un algoritmo que genere una estructura jerárquica compacta con base en los atributos geométricos se puede obtener la representación de objetos-adaptativos para una imagen real con textura, patrones, entre otras. Se deben fusionar las figuras que tienen un alto grado de semejanza cuando se comparan sus correspondientes representaciones isotrópicas<sup>17</sup>, realizando la unión lógica de los segmentos adyacentes. Para seleccionar los pares de los segmentos a fusionar, se debe realizar una estimación del grado de diferencia entre una imagen compuesta y una

<sup>17</sup> Esto es controlado por el umbral de similitud  $k_{cs}$

isotrópica. Para ello, se puede emplear el parámetro  $d$  definido en la Ecuación (4.9). El resultado de estos cálculos es una jerarquía regularizada de todos los segmentos.

Consideremos ahora el caso de cuando se realiza una fusión de dos figuras integrada por  $n_1$  y  $n_2$  puntos respectivamente. Los momentos de segundo orden  $I_x, I_y, I_{xy}$  cumplen con las siguientes propiedades:

$$\begin{aligned} I_x &= I_x^{(1)} + I_x^{(2)} + \frac{n_1 n_2}{n_1 + n_2} \delta x^2 \\ I_y &= I_y^{(1)} + I_y^{(2)} + \frac{n_1 n_2}{n_1 + n_2} \delta y^2 \\ I_{xy} &= I_{xy}^{(1)} + I_{xy}^{(2)} + \frac{n_1 n_2}{n_1 + n_2} \delta x \delta y \end{aligned} \quad (4.22)$$

Donde los índices 1 y 2 indican los números de segmentos y  $\delta x, \delta y$  describen el desplazamiento relativo de los centros de inercia, es decir, entre cada uno de los centros de los segmentos.

En el proceso de construcción de la jerarquía compacta, el valor de  $d$  está expresado por los parámetros  $\Delta$  y  $\varphi$  que se obtienen usando los nuevos momentos, conforme a la Ecuación (4.22). Si se cumple que  $I_{xy} = 0$  e  $I_x = I_y$ , entonces la nueva figura está descrita por  $d = 0$  y aparentará ser una figura isotrópica. La superposición de dos figuras isotrópicas resulta en una nueva figura isotrópica si y solo si coinciden los centros de inercia.

Para analizar los objetos, éstos deben ser estudiados realizando primeramente una división en fragmentos; para la descripción de objetos se puede usar la superposición de fragmentos con los centros de inercia alineados.

#### 4.3.4 Algoritmo de compresión semántica

El algoritmo de compresión semántica recibe como entrada una geoimagen y obtiene como salida un conjunto de geoimágenes comprimidas semánticamente, acorde con una característica o parámetro. A partir de la geoimagen original, el algoritmo de compresión semántica aplica un

conjunto de pasos para realizar la fusión de segmentos, seguido de una etapa de normalización y por último generando el conjunto de geoimágenes comprimidas (ver Figura 4–5).

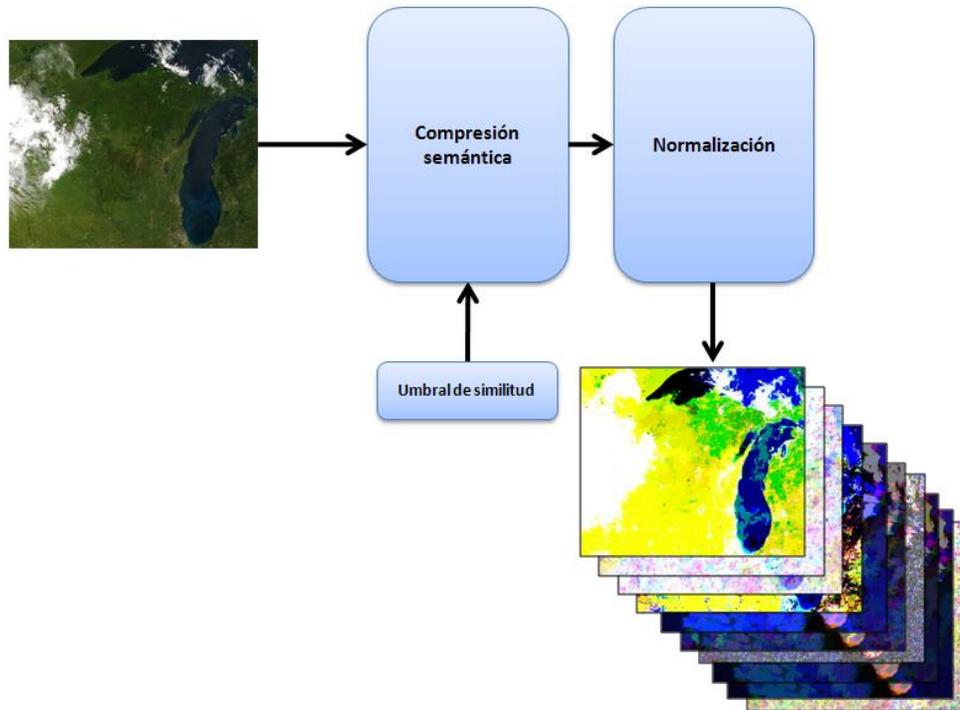


Figura 4–5. Esquema general del algoritmo de compresión semántica.

La fusión de segmentos se basa en que exista una similitud (acorde al umbral de compresión semántica<sup>18</sup>) en alguna característica o parámetro de compresión semántica, por lo que la fusión tiene que procesarse de forma independiente para cada característica o parámetro ya que dos segmentos pueden presentar similitud en la característica de **área**, pero no en la de **largo**. Lo anterior es una ventaja del algoritmo de compresión semántica, ya que permite hacer la simplificación de la geoimagen acorde con una característica o parámetro. Si se usa la característica de **tamaño**, el resultado es una geoimagen que contiene los segmentos fusionados acorde a una similitud en su **tamaño**. Al usar el conjunto completo de características y parámetros, se obtiene un total de diez geoimágenes, cada una describiendo la simplificación correspondiente según la variable de compresión semántica empleada.

De forma general el algoritmo de compresión semántica con una característica o parámetro de simplificación, consiste de los siguientes pasos:

<sup>18</sup> El umbral de compresión semántica también es referido como umbral de similitud.

1. **Creación e inicialización del árbol dinámico.** Se genera en un árbol dinámico  $B$  con un nodo raíz y con un número de nodos-hoja (segmentos) igual al número de píxeles de la geoimagen, es decir:

$$card(B) = M \times N \quad (4.23)$$

Donde  $M$  y  $N$  denotan el número de columnas y filas de la geoimagen original respectivamente. A cada nodo hijo se le asigna un índice de región como lo indica la siguiente ecuación (inicialización por número de píxel):

$$indice = ((x + 1) \times M) + ((y + 1) \times N) \quad (4.24)$$

Lo anterior genera un árbol dinámico con un solo nodo padre, y un total de  $n = M \times N$  segmentos, ver Figura 4–6. Es importante resaltar que el nodo raíz es únicamente simbólico, no representa un segmento de la geoimagen.

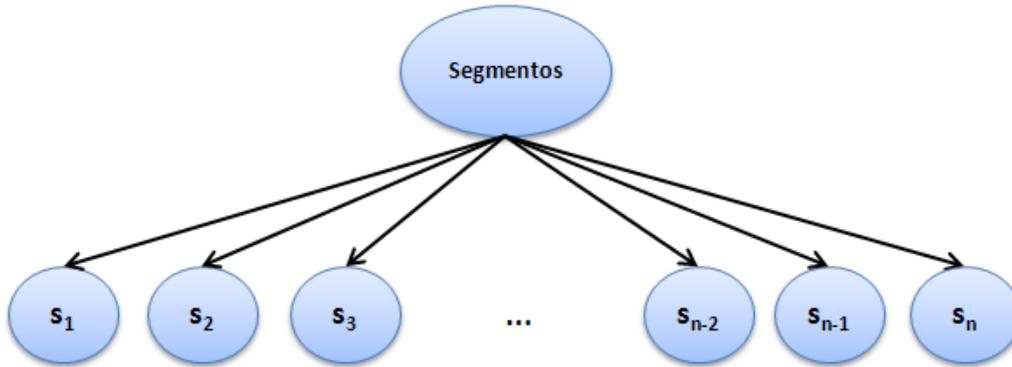


Figura 4–6. Árbol dinámico para iniciar el proceso de compresión semántica.

2. **Fusión de segmentos homogéneos.** Se fusionan los segmentos adyacentes con el mismo nivel de color, creando un nuevo nodo padre y agregando como hijos, todos los segmentos con la misma intensidad de color (ver Figura 4–7), es decir:

$$s_l = s_i \cup s_j = \{p_1, p_2, \dots, p_m\} \cup \{q_1, q_2, \dots, q_n\} \quad (4.25)$$

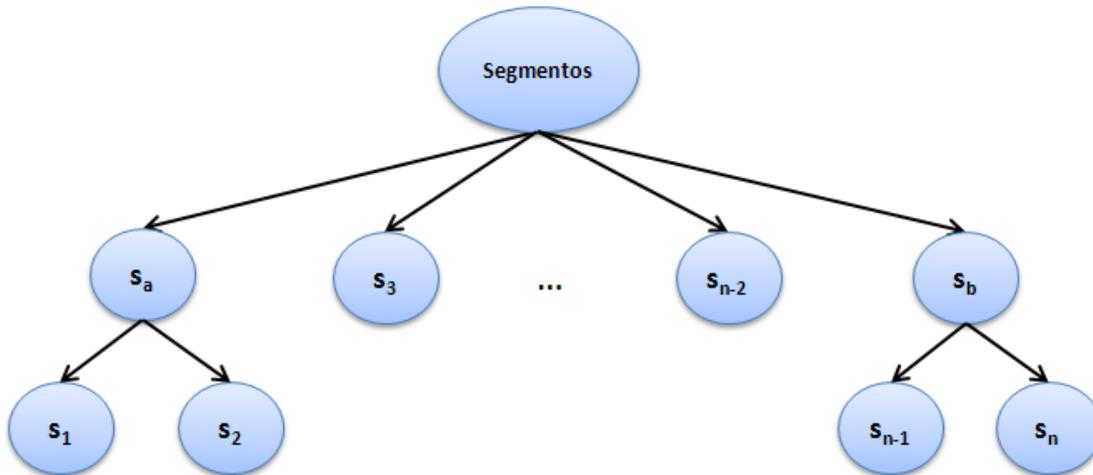


Figura 4–7. Ejemplo de fusión de segmentos en el árbol dinámico.

3. **Condición de repetición.** Se aplica un proceso iterativo a partir del punto 4 hasta que el número de fusiones de segmentos es igual a cero.
4. **Cuantificación de característica o parámetro.** Sea un árbol dinámico ( $B$ ) que describe los segmentos de la geoimagen con base en alguna característica o parámetro. Para cada segmento que no esté asociado con otro segmento (es decir, que no tengan un nodo padre común) y para cada segmento que describe una fusión, se obtiene la cuantificación de la característica o parámetro ( $c_s$ ). Esto no se realiza para los segmentos que fueron fusionados, ya que al calcular  $c_s$  para un segmento resultado de una fusión, todos los segmentos que lo integran contribuyen en la cuantificación de  $c_s$ , esto se ilustra en la Figura 4–8.

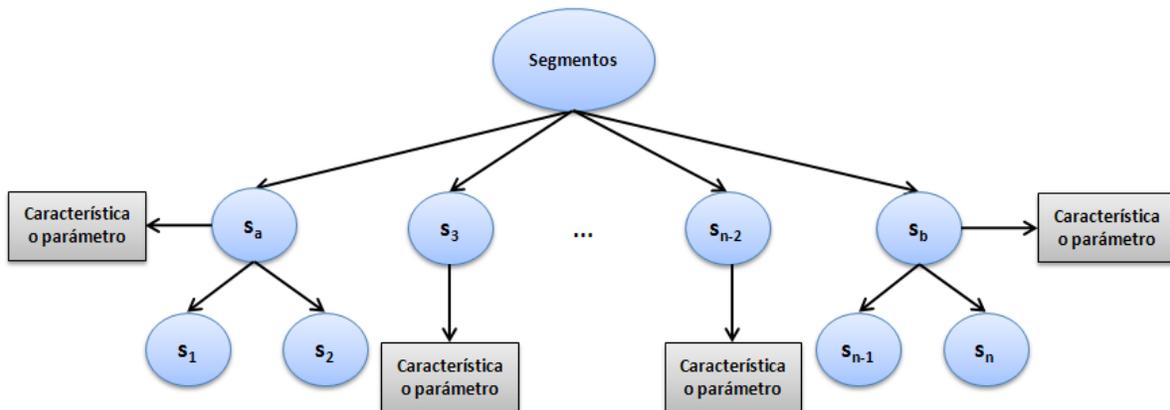


Figura 4–8. Cuantificación de una característica o parámetro.

5. **Diferencia máxima.** Se encuentran los valores máximo y mínimo de  $c_s$  entre las características cuantificadas para todos los segmentos:

$$\begin{aligned} c_{s\_max} &= \text{máximo}(c_{s_1}, c_{s_2}, \dots, c_{s_n}) \\ c_{s\_min} &= \text{mínimo}(c_{s_1}, c_{s_2}, \dots, c_{s_n}) \end{aligned} \quad (4.26)$$

Con estos valores se encuentra la diferencia máxima entre características, esto se muestra en la siguiente ecuación:

$$\Delta c_{max} = c_{s\_max} - c_{s\_min} \quad (4.27)$$

La multiplicación del umbral de compresión semántica ( $k_{cs} \in [0 \dots 100]$ ) junto con este nuevo valor ( $\Delta c_{max}$ ), permite determinar la tolerancia máxima entre la característica de dos segmentos:

$$k_c = \Delta c_{max} \times k_{cs} \quad (4.28)$$

6. **Fusión.** Se buscan pares de segmentos que cumplan el criterio de adyacencia y que la diferencia de la característica de los segmentos sea menor a  $k_c$ . En caso de que existan estos dos nodos  $s_i, s_j$  se fusionan generando un nuevo nodo padre  $s_l$  dentro del árbol dinámico que contiene los segmentos  $s_i, s_j$  fusionados, lo cual se indica asociando estos nodos como hijos de  $s_l$ .

$$s = s_i \cup s_j, \quad \forall i \neq j \mid (|c_{s_i} - c_{s_j}| < k_c) \quad (4.29)$$

7. **Normalización.** Se genera una geoimagen normalizando los valores que contiene el árbol dinámico ( $B$ ) al rango de intensidades de color  $[0, \dots, 255^3]$ .

El número de regiones es igual a la cantidad de nodos que son hijos directos del nodo raíz del árbol dinámico  $B$ . Con este algoritmo, se realiza una fusión de segmentos de la geoimagen con base en la similitud de una característica o parámetro entre pares de segmentos adyacentes. El resultado del algoritmo es un conjunto de geoimágenes que pueden diferir en la asociación de los

segmentos. Es por ello, que el usuario tiene que elegir la geoimagen que proporciona mejores resultados.

#### 4.3.5 Modo iterativo del algoritmo de compresión semántica

Generalmente después de terminada una ejecución del algoritmo de compresión semántica, no se llega a una simplificación adecuada, debido a la complejidad de los objetos geoespaciales. Por lo anterior, puede aplicarse el algoritmo de compresión semántica de forma iterativa. El objetivo es repetir la compresión semántica hasta que una intensidad de color homogénea corresponda con cada objeto geográfico que se desea obtener de la geoimagen.

Este modo de operación del algoritmo consiste en tomar una geoimagen de salida como geoimagen de entrada para aplicar nuevamente el algoritmo de compresión y así sucesivamente, un total de  $i_{CS}$  veces. Sin embargo, es necesario hacer una selección manual de la geoimagen a procesar porque el algoritmo de compresión semántica genera un total de  $n_{CCS}$  geoimágenes ( $n_{CCS} \leq 10$ ) con base en las características de distancia (d), largo (L), ancho (W), tamaño (Sz), área (S), intensidad (I), intensidad promedio (ABS), o los parámetros  $\sin \Delta$  (Ss),  $\cos 2\gamma$  (C2f),  $\sin 2\gamma$  (S2f). En consecuencia, es necesario definir cuál será la geoimagen, de este conjunto de salida, que servirá como geoimagen de entrada para la siguiente iteración del algoritmo, ver Figura 4–9. Para cada iteración del algoritmo se debe proporcionar un umbral de similitud, pudiendo ser el mismo en todas las iteraciones u otro valor.

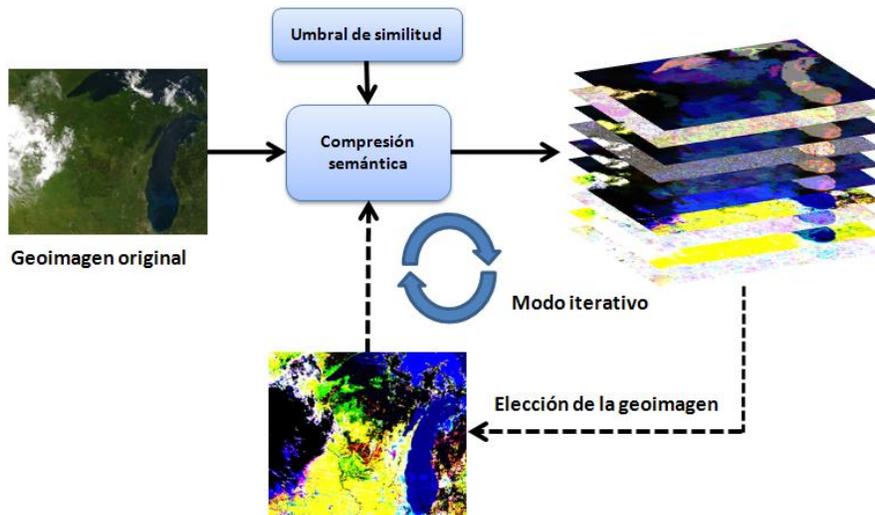


Figura 4–9. Modo iterativo del algoritmo de compresión semántica.

Desde la perspectiva de la teoría de la complejidad de algoritmos, cuando se tiene una técnica que recibe un conjunto de  $m$  datos y genera  $m^2$  resultados y que puede aplicarse de forma iterativa, se dice que el algoritmo tiene una complejidad exponencial. Para nuestro caso, suponiendo un modo iterativo de tres niveles, se tendría un total de 1000 nuevas geoimágenes; para cinco niveles, serían 100,000 geoimágenes, lo cual teóricamente implicaría un costo computacional bastante elevado.

El algoritmo de compresión semántica produce resultados constantes después de 6 ó 7 iteraciones, lo cual significa que no es posible simplificar más la geoimagen, por lo que aún aplicando el algoritmo iterativamente sobre todo el conjunto de geoimágenes, con los equipos de cómputo actuales es posible realizarlo sin ningún inconveniente; el problema se centra en que visualmente sería difícil (si no es que imposible) comparar todo este conjunto de geoimágenes.

En la primera iteración del algoritmo de compresión semántica, la geoimagen original siempre es la imagen<sup>19</sup> de entrada y el conjunto de salida son las geoimágenes comprimidas semánticamente acorde con alguna característica o parámetro la que es cuantificada en los segmentos que componen la geoimagen a procesar. Generando un árbol de este conjunto de geoimágenes se tiene la Figura 4–10.

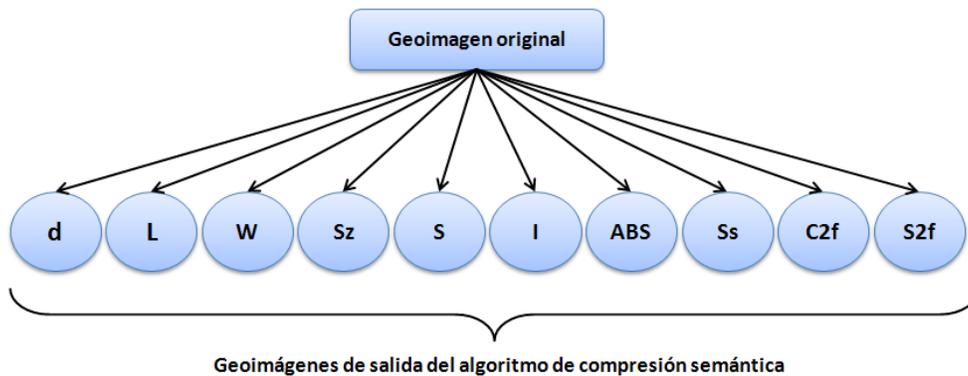


Figura 4–10. Representación de la primera iteración del algoritmo de compresión semántica.

En el primer nivel de esta gráfica se representa simbólicamente la geoimagen de entrada del algoritmo (imagen original), y cada nodo hijo es una geoimagen comprimida semánticamente

<sup>19</sup> De aquí en adelante se usa el término de geoimagen e imagen de forma indistinta, con el objeto de evitar una excesiva repetición del primer término; esto no debe causar confusión alguna al lector.

acorde con alguna característica o propiedad, describiendo cada segmento con un color de intensidad uniforme. Por ejemplo, considerando el nodo hijo de la izquierda, la geoimagen de distancia  $d$  refleja el resultado de las fusiones de segmentos, considerando la similitud en la característica de distancia de los segmentos de la geoimagen; lo mismo aplica al resto de geoimágenes generadas por el algoritmo.

Cuando se ejecuta otra iteración del algoritmo, se selecciona una de las diez geoimágenes generadas la cual sirve como entrada para la siguiente iteración del algoritmo. Todas las nuevas geoimágenes generadas, describen el resultado de la fusión de segmentos acorde con la característica o parámetro de la compresión semántica sobre la geoimagen de entrada de esta nueva iteración. Después de terminar esta iteración del algoritmo, el experto puede determinar si necesita una mayor simplificación de la geoimagen, en cuyo caso selecciona una geoimagen y aplica nuevamente el algoritmo. En la Figura 4–11 se muestra el resultado después de tres iteraciones tomando como geoimagen de entrada para cada nueva iteración, la imagen de intensidad.

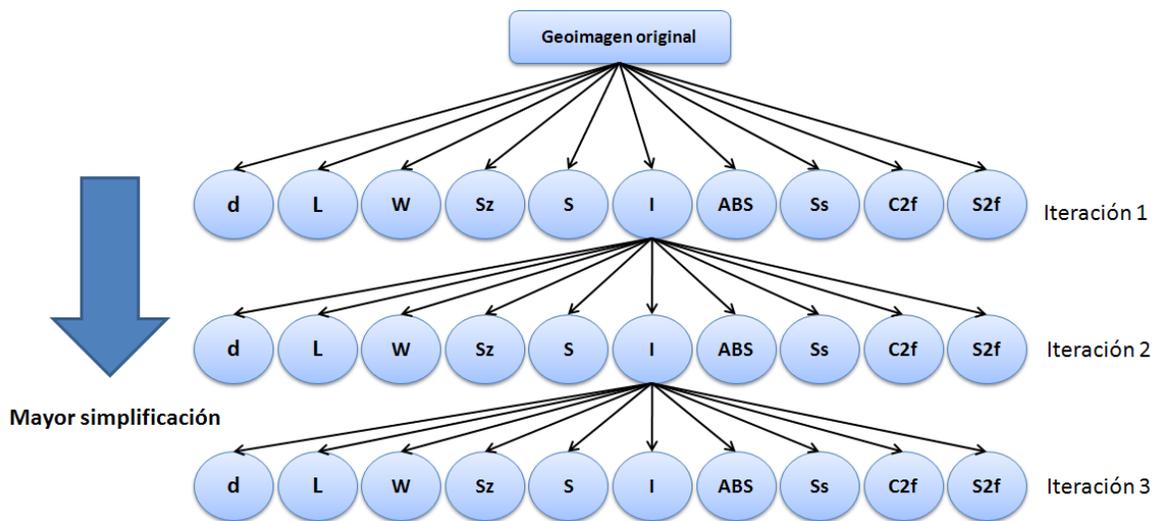


Figura 4–11. Ejecución del algoritmo de compresión semántica con tres iteraciones tomando como base la geoimagen de intensidad.

Conforme se realizan más iteraciones del algoritmo, se realiza una mayor simplificación de los objetos geográficos, es decir, las regiones de los objetos geográficos se describen con menos intensidades homogéneas de color. En el ejemplo de la figura anterior, en la segunda y tercera iteración se tomó como base la geoimagen de intensidad resultante del primero y segundo ciclo

de comprensión semántica, respectivamente. Como en estas dos iteraciones se usó la misma característica para hacer la simplificación, una forma de abreviar esta descripción es diciendo que: **se simplificó la geoimagen de intensidad**, o bien, **se aplicó una compresión semántica sobre la geoimagen de intensidad**, entendiéndose que siempre en la primera iteración, se toma como entrada la imagen original y en las posteriores, la geoimagen que describe los resultados de la fusión con base en la característica de intensidad. Si usamos esta misma frase para la característica de tamaño ( $Sz$ ), se diría que se simplificó la geoimagen de la característica  $Sz$ , lo cual significa que en todas las iteraciones ( $i \geq 2$ ) se usó como entrada del algoritmo, la geoimagen de la característica de tamaño obtenida en la iteración anterior.

En cada iteración del algoritmo se debe especificar un valor para el umbral de similitud,  $k_{CS}$ ; para ello existen dos opciones: usar el mismo umbral de similitud en todas las iteraciones, o bien, emplear un valor diferente, lo cual queda a juicio y elección del usuario. En modo iterativo, el algoritmo de compresión semántica no está limitado a usar siempre la misma geoimagen de característica, es decir, en una iteración podemos emplear la geoimagen de intensidad promedio (ABS) y en una posterior, la geoimagen de distancia (d), como se puede observar en la Figura 4–12. Es el usar diferentes características y parámetros de simplificación lo que permite simplificar cualquier tipo de geoimagen, ya que se involucra el color, la geometría y la topología de los objetos geoespaciales.

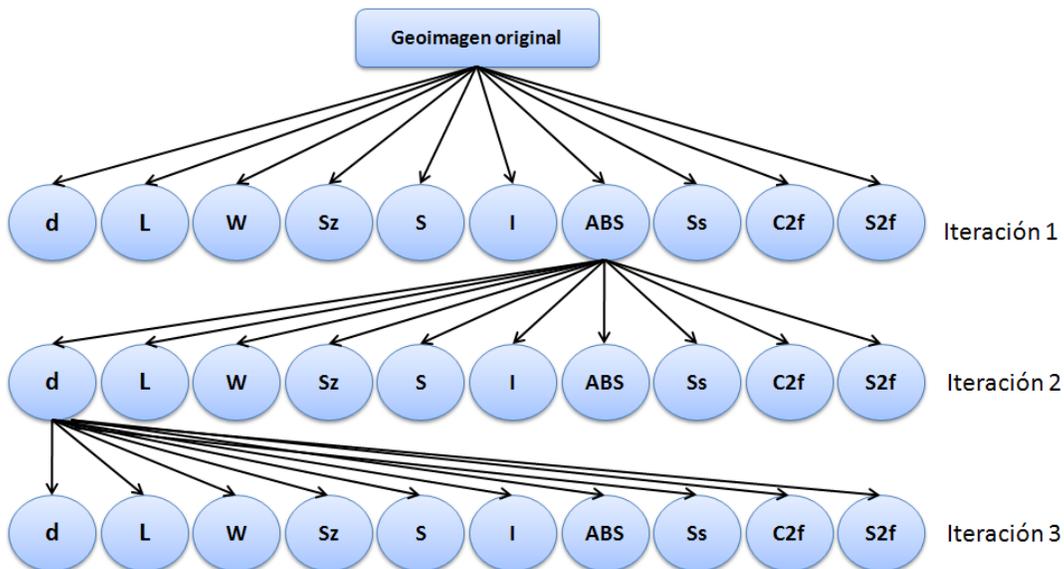


Figura 4–12. Uso de las geoimágenes de intensidad promedio (ABS) y de distancia (d) en el algoritmo de compresión semántica.

Terminada la última iteración de la técnica de compresión semántica, se realiza una nueva elección de geoimagen, entre el conjunto de  $n_{ccs}$  imágenes generadas después de terminar todas las iteraciones. Como la geoimagen que sirve de entrada para cada nueva iteración así como los umbrales de similitud pueden ser diferente, se propone la siguiente definición con el objeto de poder expresar todos los pasos que fueron empleados para simplificar una geoimagen dada.

**Definición 4–3.** Una cadena o secuencia de compresión semántica, se define como el conjunto de geoimágenes y umbrales de similitud que se usa en la simplificación de una geoimagen  $I_o$ , cuyo resultado final es una geoimagen  $I_{ccs}$ , la cual se elige del conjunto de geoimágenes que se obtiene en la  $i_{cs}$ -ésima iteración del algoritmo de compresión semántica:

$$C_{cs}(I_o, I_{ccs}) = \left\{ (I_o, k_{cs1}), (I_1, ccs_1, k_{cs2}), (I_2, ccs_2, k_{cs3}), \dots, (I_{i_{cs}-1}, ccs_{i_{cs}-1}, k_{cs_{i_{cs}}}) \right\} \quad (4.30)$$

Donde:

$(I_{i-1}, ccs_{i-1}, k_{cs_i})$  Son los parámetros de entrada en la  $i$ -ésima iteración.

$k_{cs_i}$  Es el umbral de similitud usado en la  $i$ -ésima iteración.

$ccs_{i-1}$  Es la característica de compresión semántica usada para simplificar la geoimagen  $I_{i-1}$ .

$I_o$  Es la geoimagen de entrada al algoritmo de compresión semántica.

$I_{ccs}$  Es la geoimagen elegida del conjunto de imágenes generadas después de la  $i_{cs}$ -ésima iteración, es decir:  $\{I_d, I_L, I_W, I_{Sz}, I_S, I_{intensidad}, I_{ABS}, I_{Ss}, I_{C2f}, I_{S2f}\}$ .

Con esta notación,  $(I_1, ccs_1, k_{cs2})$  representa que en la segunda iteración del algoritmo, se usó la geoimagen que se obtuvo en la primera iteración ( $I_1$ ) la cual fue simplificada semánticamente con la característica o parámetro  $ccs_1$ . Los valores que puede tomar  $ccs_i$  son:  $\{d, L, W, Sz, S, intensidad, ABS, Ss, C2f, S2f\}$ .<sup>20</sup> Cuando se usa el mismo umbral de similitud y la misma característica o parámetro, se puede usar la siguiente notación para una cadena de compresión semántica:

$$C_{cs}(I_o, I_{ccs}) = \{(I_o, k_{cs}), (ccs, k_{cs})_{i_{cs}-1}\} \quad (4.31)$$

<sup>20</sup> En la expresión de cadena de compresión semántica la característica de intensidad no se abrevia para evitar cualquier confusión con el símbolo de una geoimagen.

El conjunto propuesto de características y parámetros permite que, cuando se emplean en su totalidad, poder hacer una simplificación adecuada de las regiones que describen los objetos geográficos. Esto implica que se debe experimentar con el número de iteraciones, umbral de similitud y la geoimagen de entrada en cada iteración para poder extraer los objetos geográficos.

#### 4.4 Descomposición semántica

Definir una metodología general de descomposición que permita trabajar con un conjunto amplio de geoimágenes, es una tarea muy difícil, aun estableciendo un conjunto de restricciones (dominio del problema). Es por ello, que no se puede plantear una regla única que siempre permita obtener los resultados deseados con cualquier tipo de geoimagen. Lo que se propone a continuación es una metodología completa con la que se puede hacer una correcta extracción de los objetos geográficos que componen una geoimagen, la que requiere de la manipulación de los parámetros de los algoritmos, principalmente el de compresión semántica; por ello estos valores dependen ampliamente del tipo de geoimagen a trabajar.

Esto quiere decir que los umbrales con los que se procese correctamente una geoimagen, posiblemente no sean los mismos a emplear con otra geoimagen de características diferentes; lo más probable es que se tengan que modificar. Sin embargo, la compresión semántica tiene la característica de que en algún momento va a converger en la simplificación de los objetos geográficos.

El algoritmo general de descomposición semántica procesa una geoimagen y genera como resultado un conjunto de etiquetas de los objetos geográficos que componen la geoimagen original. Está integrado por dos etapas: análisis y síntesis. La primera de esta etapa recibe una geoimagen y genera como resultado un conjunto de regiones descritas con intensidad de color uniforme. Por otra parte, la etapa de síntesis procesa cada una de estas regiones, para generar el conjunto especializado de etiquetas de los objetos geográficos que componen la geoimagen. En este trabajo, se entenderá que dicho conjunto de etiquetas es la semántica de la geoimagen. El algoritmo de descomposición semántica propuesto se puede observar en la Figura 4–13.

Descrito de manera informal, el algoritmo de descomposición semántica está integrado por los siguientes pasos:

1. Debido a que la geoimagen original contiene muchos detalles y sobre todo una gran cantidad de objetos semánticamente no importantes, el objetivo de la etapa de análisis consiste en hacer una simplificación de la misma, por medio de un algoritmo de compresión semántica. Este algoritmo emplea un conjunto de variables mediante las cuales se realiza la fusión de segmentos<sup>21</sup>, cuando se cumple una condición de adyacencia y que una determinada característica o parámetro de los segmentos a fusionar es similar.

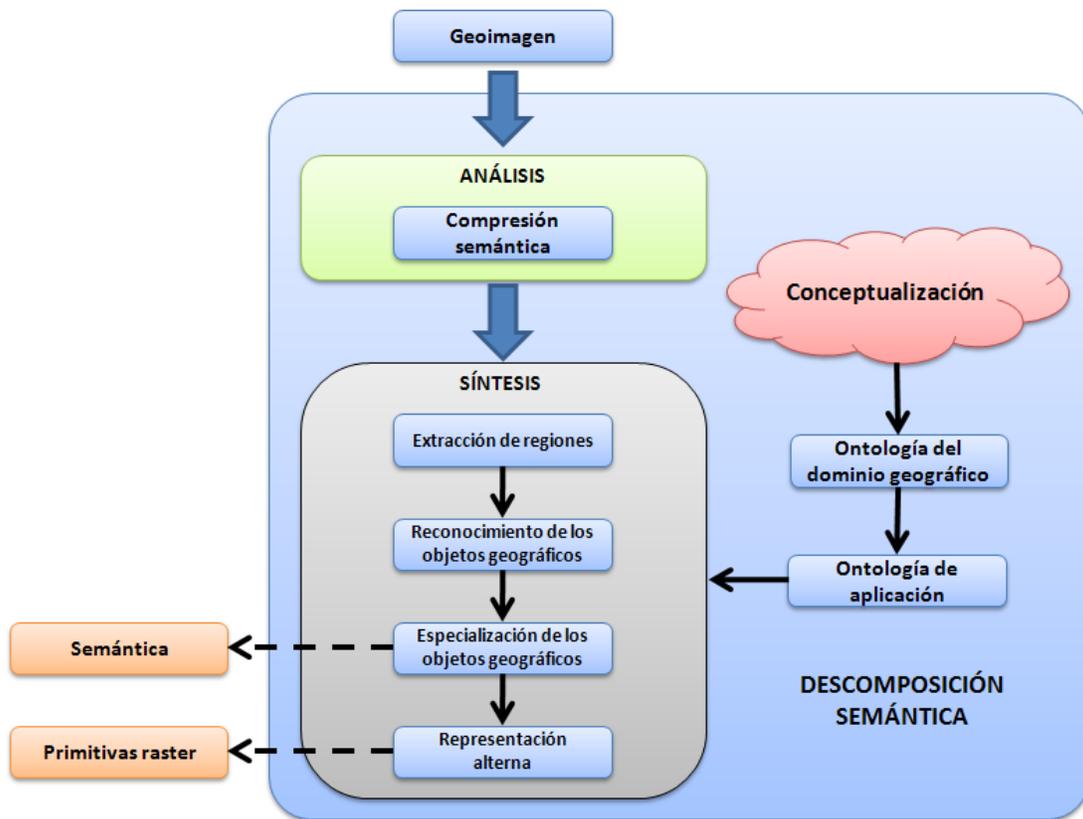


Figura 4–13. Algoritmo general de descomposición semántica.

Estas operaciones se van realizando sobre un árbol dinámico que equivale a eliminar las variaciones de bajo nivel que describe cada objeto: textura, borde, intensidad, forma

<sup>21</sup> El conjunto inicial de segmentos es igual al número de píxeles que componen la geoimagen.

geométrica. Al finalizar este algoritmo, se tiene un conjunto de segmentos (conjunto de píxeles) descritos con una intensidad de color uniforme.

2. Como se está interesado en reconocer los objetos que componen la geoimagen, se necesita extraer un conjunto de parámetros de bajo nivel, por lo cual se procede a generar una estructura que enlista los objetos y el valor de sus propiedades cuantificadas.
3. El sistema de reconocimiento toma en cuenta las características de bajo nivel de los objetos para poder asociar etiquetas generales a cada uno de ellos.
4. Se realiza una especialización de los objetos geoespaciales haciendo uso de una ontología de aplicación del dominio geográfico<sup>22</sup>.
5. Cuando no sea posible especializar un objeto, se debe repetir la descomposición semántica con diferentes parámetros, ya que el nivel de simplificación no fue el adecuado.
6. Una vez que se tienen especializados los objetos geográficos, se puede generar una representación alterna de los mismos. Lo anterior es posible ya que al conocer con precisión los objetos geográficos, éstos se pueden describir mediante una serie de primitivas tales como: puntos, líneas y áreas.

Es difícil percibir la simplificación de la información geoespacial por medio de características tales como el tamaño, el ancho, la longitud, el área, etc.; debido a que tradicionalmente se está acostumbrado a realizar procesamiento basado en color. En la práctica, en el contexto de un experto, por ejemplo un oceanógrafo, el cual trabaje con geoimágenes muy similares, podría definirse el rango del umbral de similitud, número de iteraciones o características de compresión semántica, que proporcionen resultados adecuados a sus necesidades.

---

<sup>22</sup> Las etiquetas especializadas describen la semántica de la geoimagen.

## 4.5 Etapa de análisis

Para poder hacer la simplificación adecuada de la geoimagen, se debe aplicar el algoritmo de compresión semántica en modo iterativo, siendo necesaria únicamente la manipulación de tres parámetros:

1. Umbral de compresión semántica.
2. Número de iteraciones.
3. Geoimagen de entrada en cada nueva iteración.

Terminada la última iteración del algoritmo, se elige la geoimagen que servirá de entrada para la etapa de síntesis. En consecuencia, la etapa de análisis consiste en encontrar la geoimagen simplificada  $I_{CCS}$  a partir de una imagen original  $I_o$  aplicando una secuencia de compresión semántica:

$$C_{CS}(I_o, I_{CCS}) = \left\{ (I_o, k_{CS1}), (I_1, CCS_1, k_{CS2}), (I_2, CCS_2, k_{CS3}), \dots (I_{i_{CS}-1}, CCS_{i_{CS}-1}, k_{CS_{i_{CS}}}) \right\}$$

Donde la geoimagen  $I_{CCS}$  está descrita por un total de  $n_{r_{CS}}$  intensidades de color uniforme y cada intensidad de color representa una clase (temática). Una de las ventajas del algoritmo de compresión semántica es que terminada la simplificación de una geoimagen, las regiones que son similares, aunque no estén adyacentes, son descritas por la misma intensidad de color. Por ejemplo, en una geoimagen que contiene tres cuerpos de agua y una región de suelo, es probable que los tres cuerpos de agua sean descritos con la misma intensidad de color, mientras que la región de suelo con otra intensidad. En caso de que cada región tenga una intensidad de color diferente, el número de intensidades es igual al número de objetos geográficos extraídos de la geoimagen original.

## 4.6 Etapa de síntesis

La segunda etapa de la metodología de descomposición semántica es la etapa de síntesis. Su objetivo es obtener la semántica de la geoimagen seleccionada después de aplicar el algoritmo de compresión semántica. Esta etapa está integrada por los siguientes algoritmos (ver Figura 4–14):

1. Extracción de regiones.
2. Reconocimiento de los objetos geográficos.
3. Especialización de los objetos geográficos.
4. Representación alterna.

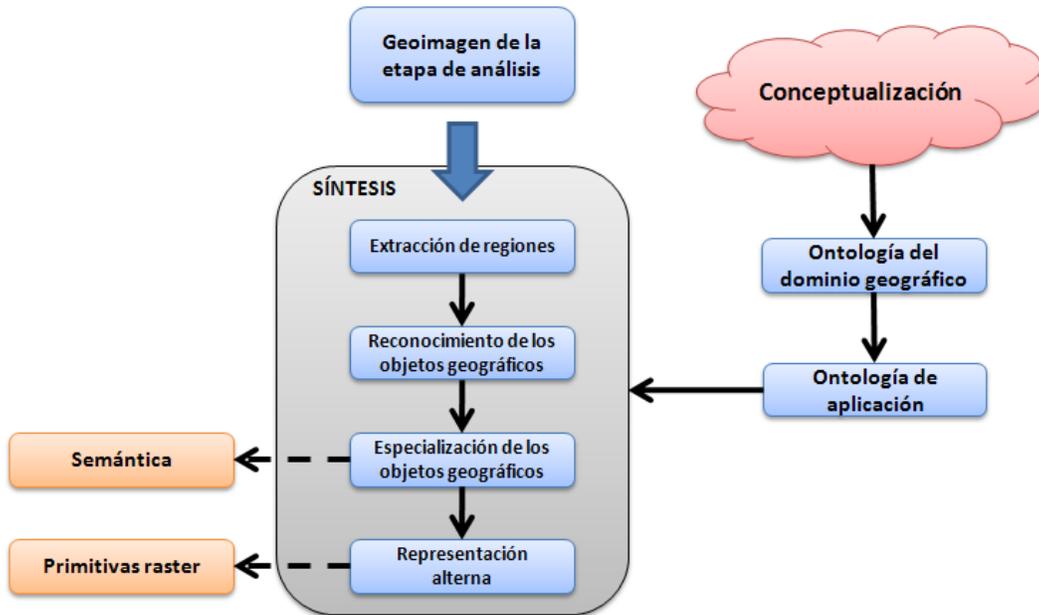


Figura 4–14. Diagrama general de la etapa de síntesis.

#### 4.6.1 Extracción de regiones

El resultado de la etapa de análisis es una geoimagen simplificada en un total de  $n_{rcs}$  clases que están descritas por algún color de intensidad uniforme. Sin embargo, con esta información no es posible hacer un reconocimiento de los objetos geoespaciales, por lo que es necesario recuperar los datos de bajo nivel de la geoimagen original. Se puede definir al algoritmo de extracción de regiones como una función que permite crear una correspondencia entre las intensidades de las geoimágenes original y la agrupada, la cual genera como resultado una lista de geoimágenes donde cada una contiene todas las regiones de los objetos geográficos del mismo tipo con su intensidad de color original. Este procedimiento consiste en realizar los siguientes pasos para cada identificador de clase obtenido en la clasificación:

1. Sea  $i$  un identificador de número de clase, donde  $i = \{1, 2, \dots, n_{rcs}\}$ , se extraen todas las regiones homogéneas que están clasificadas en la clase  $i$  en una nueva geoimagen  $I_{T_i}$ , es decir, aquellas que satisfacen la condición:

$$I_{T_i} = \{(x, y) \in I_A \mid N_c(x, y) = i\} \quad (4.32)$$

Donde  $N_c(x, y)$  es una función que indica el número de clase a la que pertenece el elemento discreto  $p$  de coordenadas  $(x, y)$  en  $I_A$ , en consecuencia  $Rango(N_c) = \{1, 2, \dots, n_{rcs}\}$ .

Después de ejecutar este punto para el identificador de clase 1, se tiene una geoimagen que contiene al menos una región homogénea<sup>23</sup> compuesta de todos los pixeles que fueron agrupados en la clase 1; lo mismo sucede con el resto de las clases  $\{2, 3, \dots, n_{rcs}\}$ .

2. Sustituir la intensidad de color de cada elemento<sup>24</sup> en  $I_{T_i}$  por la correspondiente en la geoimagen original  $I_O$ , es decir:

$$I_{T_i}(x, y) = \langle R(I_O(x, y)), G(I_O(x, y)), B(I_O(x, y)) \rangle \forall p(x, y) \in I_{T_i} \quad (4.33)$$

Tanto la cantidad de geoimágenes uniformes como el número de regiones no se altera, sino únicamente se hace un reemplazo del color de cada píxel. Dado que en cada geoimagen se agregan todos los pixeles de las regiones asociadas a la misma clase, se puede reemplazar la intensidad uniforme por la correspondiente en la geoimagen original, sin que esto genere una pérdida de la información de agrupamiento.

Terminado este punto, se obtienen  $n_{rcs}$  geoimágenes  $I_T = \{I_{T_1}, I_{T_2}, \dots, I_{T_{n_{rcs}}}\}$ , donde cada geoimagen contiene todos los objetos geoespaciales que pertenecen a una misma temática. Teniendo las temáticas separadas, es posible hacer un reconocimiento y asignar la etiqueta del primer nivel que le corresponde. La forma de conseguirlo se describe a continuación.

<sup>23</sup> Se afirma que al menos hay una región homogénea por las restricciones del algoritmo de clasificación.

<sup>24</sup> Los pixeles del fondo de la imagen no son reemplazados, solamente aquellos que están asociados a una región homogénea en  $I_A$ .

#### 4.6.2 Reconocimiento de objetos geográficos

La etapa de síntesis persigue obtener la semántica de la geoimagen como un conjunto de etiquetas en lenguaje natural, donde cada etiqueta corresponde al nombre de un objeto geográfico.

Tradicionalmente, los algoritmos de reconocimiento e interpretación (y en general los algoritmos orientados a píxeles) emplean únicamente las características de bajo nivel del objeto que se está procesando para poder asociarlo a cierta clase. Por ejemplo, en la Figura 4–15 se puede observar que existen dos objetos geográficos.



Figura 4–15. Ejemplo de objetos geográficos.

La dificultad se presenta cuando se trata de especializar la identidad de un objeto geográfico, ya que se tienen dos opciones:

1. Usar un conjunto de objetos de entrenamiento muy amplio. Posiblemente sería la solución más fácil, sobre todo si se está usando una escala donde el nivel de detalle de los objetos geográficos es muy grande. Sin embargo, conforme la escala aumente, la similitud entre distintos objetos geográficos es mayor y aunque se haya empleado una gran cantidad de geoimágenes de entrenamiento, es muy probable que se presenten ambigüedades de clasificación. Además, el conjunto de entrenamiento tendría que estar integrado por objetos

bastante representativos para que existan variaciones en los vectores característicos resultantes.

2. Usar información global. En este caso se requiere de datos adicionales a los contenidos en las características locales del objeto geográfico. Por ejemplo, un individuo que inspeccione la geoimagen de la Figura 4–15, analizaría el entorno o vecindad sobre el que se encuentra el objeto geográfico. Con esta información, junto con la incluida en el propio objeto geográfico, como son intensidades, textura, dimensiones, etc., la persona es capaz de especializar el objeto geográfico.

Como aportación de esta tesis, se propone que para poder obtener una descripción correcta del contenido de una geoimagen, se realice primero un reconocimiento general de los objetos geográficos se clasifiquen en temáticas y posteriormente se llegue a una especialización de éstos mediante el uso de una ontología de aplicación.

El algoritmo de reconocimiento tiene el objetivo de hacer un **reconocimiento general** de todas las regiones que describen los objetos geográficos detectados en una geoimagen. A diferencia de las metodologías de otros autores, en este trabajo de investigación se propone que el algoritmo de reconocimiento realice una asignación más general de las etiquetas de los objetos geoespaciales, tomando como base un conjunto de capas temáticas.

Lo anterior permite que sea posible usar un algoritmo de reconocimiento simple, porque la especialización se consigue tomando en cuenta la geometría y topología de los objetos geográficos. Con base en características de bajo nivel, puede afirmarse que una determinada región representa por ejemplo un cuerpo de agua, mientras que otra región es tierra o suelo, pero lo que aporta mayor significado en una identificación de objeto, es su configuración o distribución en un determinado contexto geográfico.

Esto es, un cuerpo de agua aislado siempre será un cuerpo de agua, sus características propias son las que permiten asignarle una etiqueta de cuerpo de agua; sin embargo, en un contexto donde ese cuerpo de agua esté *rodeado por* otro objeto geográfico como es el suelo, es cuando adquiere

más especialización, pues ya no es solamente un cuerpo de agua, sino que es un lago o una laguna.

Desde la perspectiva de los sistemas de información geográfica, una capa temática está integrada por datos que poseen los mismos atributos asociados y una misma forma de representación, por ejemplo, topografía, geología, geomorfología, áreas verdes, áreas urbanas, etc. (ver Figura 4–16). Esta interpretación de capa temática no se basa únicamente en las características físicas de la información geoespacial, por lo que no es una definición adecuada para este trabajo.

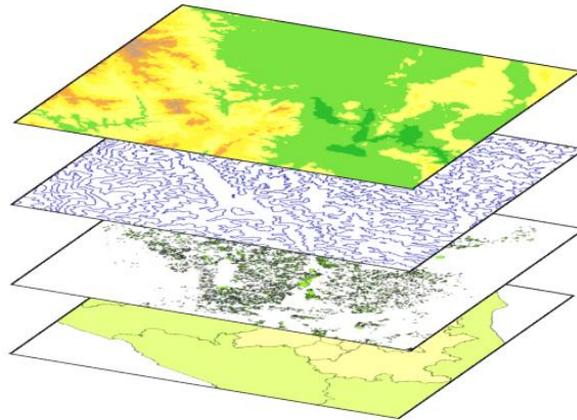


Figura 4–16. Ejemplo de capas temáticas.

**Definición 4–4.** Una capa temática ( $t_i$ ) es una etiqueta que describe un conjunto de objetos geográficos que tienen similitud en alguna(s) característica(s) de bajo nivel.

Se requiere de esta definición porque el algoritmo de reconocimiento solamente dispondrá de la información de intensidad de las regiones que describen los objetos geográficos; en consecuencia, es necesario restringir la ambigüedad en la asignación de etiquetas de tal forma que un objeto geoespacial no pueda ser asociado con más de una temática. Actualmente no existe un conjunto único de capas temáticas, sino, las necesidades específicas de un caso de estudio son las que influyen significativamente en la definición de dicho conjunto; en algún proyecto en particular, se puede usar el conjunto: hidrología, topografía, geología, vías de comunicación, áreas verdes, geomorfología, mientras que el Instituto Nacional de Estadística, Geografía e Informática (INEGI) define las siguientes capas: aeronáutica, climatología, edafología, geología, hidrología de aguas subterráneas, hidrología de aguas superficiales, humedad del suelo, topografía, entre otras. En

consecuencia, según la aplicación y el problema que se esté resolviendo, se puede definir un conjunto de capas temáticas que cumplan con la restricción de la definición anterior.

**Definición 4–5.** El conjunto de capas temáticas está integrado por todas las etiquetas de capas temáticas, es decir:

$$T = \{t_1, t_2, t_3, \dots, t_{n_T}\} \quad (4.34)$$

Donde  $n_T$  es el número de temáticas definido para la etapa de síntesis. Con este conjunto de temáticas, se persigue hacer un etiquetado de escala gruesa (*coarse scale*) de los objetos geoespaciales. Al terminar esta asignación de etiquetas se conoce el contenido de objetos geográficos de la geoimagen.

En síntesis, el algoritmo de reconocimiento tiene que determinar primeramente un conjunto de vectores prototipo que modelen cada temática  $t_i$ , para lo cual es necesario contar con una serie de geoimágenes donde cada geoimagen describa un objeto geográfico de una sola temática. Teniendo los vectores prototipo se cuantifican las mismas propiedades en cada geoimagen  $I_{T_i}$  y aplicando algún algoritmo de reconocimiento, como el de distancia mínima, se asocia  $I_{T_i}$  con la etiqueta cuyo vector prototipo esté más cercano. El resultado es un conjunto de etiquetas en lenguaje natural  $T_{I_o} = \{eg_1, eg_2, \dots, eg_{n_{rcs}}\}$  de las temáticas reconocidas en la geoimagen  $I_o$ .

### 4.6.3 Especialización de los objetos geográficos

Como su nombre lo indica, este algoritmo tiene el objetivo de realizar una especialización en el nombre o etiqueta de cada objeto geoespacial. Por ejemplo, dado un objeto geográfico clasificado en el algoritmo de reconocimiento como un cuerpo de agua, con esta técnica se pretende llegar a la conclusión de si ese cuerpo de agua es un río, lago, laguna, mar, océano, etc. Otro caso es el de indicar si una región de suelo es una isla o bahía.

Cada una de las regiones  $\{R_1, R_2, \dots, R_j\}$  que pertenecen a una misma geoimagen de temática  $I_{T_i}$  describe un objeto geográfico  $\{O_1, O_2, \dots, O_j\}$ . Con el etiquetado general se concluye que todas las regiones  $\{R_1, R_2, \dots, R_j\}$  tienen la misma etiqueta general  $eg_i$ . La especialización debe asociar una

etiqueta adicional  $ee_m$  a cada  $(R_z \in I_{T_i}) \forall I_{T_i} \in I_T$ , considerando las propiedades geométricas y topológicas de los objetos geoespaciales. El conjunto de etiquetas específicas  $\{ee_1, ee_2, \dots, ee_{n_o}\}$  debe ser igual al número de objetos geográficos  $n_o$  que pudieron ser extraídos de la geoimagen original.

**Definición 4–6.** La semántica de una geoimagen  $I_O$  es el conjunto de etiquetas específicas que se obtienen después de hacer una especialización de los objetos geográficos:

$$S_{I_o} = \{ee_1, ee_2, \dots, ee_{n_o}\} \quad (4.35)$$

Donde  $n_o$  es el número de objetos reconocidos en  $I_O$  y  $ee_i$  es una etiqueta especializada para el  $i$ -ésimo objeto geográfico.

Para ello, es necesario contar con información cuantitativa de cada uno de los objetos geográficos, así como de las relaciones de un objeto geográfico con su entorno. Sin embargo, existen dos problemas claramente identificados con estos dos aspectos: la definición o concepto de un objeto geográfico y las relaciones entre ellos, los cuales se explican en la siguiente sección.

#### 4.6.3.1 Definición de los objetos geográficos

Mientras que la definición de ciertos objetos (en general, no únicamente objetos geográficos) es clara y no presenta ambigüedad alguna, al tratar de particularizar la identidad de un objeto geoespacial existe el inconveniente de que su definición utilice términos no cuantificables, o bien, se base en el incumplimiento en la identidad de otro cuerpo geográfico. Por ejemplo, se puede considerar la definición de algunos cuerpos de agua, como se muestra en la Tabla 4.3.

Lo anterior implica que para poder determinar ciertos atributos geoespaciales es necesario contar con la escala de la geoimagen, es decir, conocer la métrica real que representa cada píxel; con el valor de la escala (que se solicita al usuario) sería posible determinar el tamaño real (descrito en la geoimagen) de cada objeto geográfico.

Pero esto no es factor confiable totalmente, ya que como el reconocimiento se hace con base en la información que representa la geoimagen, a excepción de que toda el área que delimita el objeto geoespacial esté contenida en la geoimagen, ésta sería de un valor relativo<sup>25</sup>.

Tabla 4–1. Definiciones de algunos cuerpos de agua.

Cuerpo de Agua	Definición
<b>Río</b>	Corriente de agua continua y más o menos caudalosa que va a desembocar en otra, en un lago o en el mar <sup>26</sup> ( <b>WordReference, 2007</b> ) ( <b>Real Academia, 2007</b> )
	Un río es una corriente natural de agua que fluye con continuidad. Posee un caudal determinado y desemboca en el mar, en un lago o en otro río, en cuyo caso se denomina afluente. La parte final de un río es su desembocadura. Algunas veces terminan en zonas desérticas donde sus aguas se pierden por infiltración y evaporación: es el caso de los ríos alóctonos (llamados así porque sus aguas proceden de otros lugares con clima más húmedo), como el caso del Okavango en el falso delta donde desemboca o numerosos wadis (uadis) del Sáhara y de otros desiertos. Cuando el río es corto y estrecho recibe el nombre de riacho, riachuelo o arroyo ( <b>Wikipedia, 2007</b> )
<b>Lago</b>	Gran masa de agua, normalmente dulce, acumulada en depresiones de terreno ( <b>WordReference, 2007</b> )
	Gran masa permanente de agua depositada en depresiones del terreno ( <b>Real Academia, 2007</b> )
	Un lago es un cuerpo de agua dulce o salada, más o menos extensa, que se encuentra alejada del mar, y asociada generalmente a un origen glaciar. El aporte de agua a los lagos viene de los ríos y el afloramiento de aguas freáticas. Los lagos más grandes se forman aprovechando depresiones creadas por fallas. Otros se forman por la obstrucción de valles debido a avalanchas en sus laderas o por la acumulación de morrenas glaciares. También se pueden formar lagos artificialmente por la construcción de una presa ( <b>Wikipedia, 2007</b> )
<b>Laguna</b>	Depósito natural de agua menor que el lago ( <b>WordReference, 2007</b> )
	Depósito natural de agua, generalmente dulce y de menores dimensiones que el lago ( <b>Real Academia Española, 2007</b> )
	Es la denominación que recibe cualquier extensión natural de agua estancada, sea ésta dulce o salada. La diferencia con los lagos no es muy precisa, salvo que se supone que una laguna tiene menor extensión y profundidad. Las lagunas que se encuentran cercanas al litoral y están asociadas a un origen marino se llaman marismas.

<sup>25</sup> Esto es uno de los inconvenientes al manejar geoimágenes de escala muy fina, debido a que se pueden apreciar con muy buen nivel de detalle los objetos de tamaño relativamente pequeño o mediano y sin embargo, los objetos más grandes, como pueden ser los continentes, mares, océanos, no son capturados en su totalidad.

<sup>26</sup> Esta definición coincide textualmente entre las dos referencias consultadas.

Tabla 4–1. (Cont...) Definiciones de algunos cuerpos de agua.

Cuerpo de Agua	Definición
<b>Laguna</b>	No obstante, existen lagunas más grandes y profundas que algunos lagos, motivo por el que la diferencia se limita exclusivamente a la toponimia o a la denominación tradicional de un área lacustre. Otra característica habitual de las lagunas es que se presentan agrupadas en aquellos lugares en los que los cursos fluviales no pueden desembocar en otros ríos mayores o en el mar, y desaguan en depresiones de suelos impermeables, que contienen el agua sin filtrarla, desecándose por evaporación. Este fenómeno se denomina endorreísmo. También se utiliza para describir vacíos, como pasajes perdidos en documentos, o las lagunas de memoria <b>(Wikipedia, 2007)</b>
<b>Mar</b>	Masa de agua salada que cubre gran parte de la superficie terrestre <b>(WordReference, 2007)</b>
	Masa de agua salada que cubre la mayor parte de la superficie de la Tierra <b>(Real Academia, 2007)</b>
	Lago de cierta extensión <b>(Real Academia Española, 2007)</b>
	Un mar es una masa de agua salada de tamaño inferior al océano, así como también el conjunto de la masa de agua salada que cubre la mayor parte del planeta Tierra, incluyendo océanos y mares menores <b>(Wikipedia, 2007)</b>

En la Figura 4–17 se puede apreciar una geoimagen y una porción ampliada; al trabajar con cada imagen por separado se obtendrían diferentes resultados aún conociendo el valor de la escala de los objetos, ya que la información en cada una de ellas está incompleta. Aún con esta desventaja, el uso de la escala aparenta ser la opción más conveniente, pero la conceptualización definida en este trabajo permite que la especialización se pueda obtener usando información implícita en la ontología de los objetos geográficos, en lugar de un atributo explícito que cuantifique el tamaño basado en la escala de un objeto geoespacial.

#### 4.6.3.2 Relaciones entre los objetos geográficos

Las características propias de los objetos geográficos no son el factor determinante para poder hacer una especialización de su tipo, ya que las relaciones entre estos objetos geográficos también deben tomarse en cuenta. Esto es lo que caracteriza principalmente a una metodología orientada a objetos.

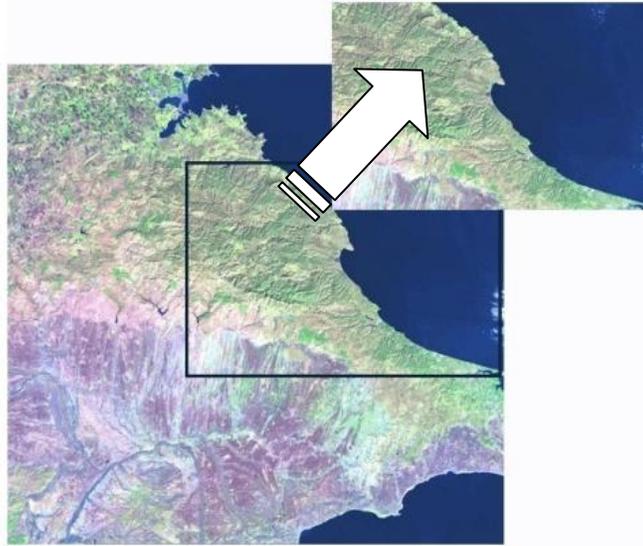


Figura 4–17. Ejemplificación de información geográfica incompleta. Fuente: Galería gratuita de LANDSAT (LANDAST Satellite Image Gallery, 2007).

Por ejemplo, sean dos objetos geográficos representados simbólicamente como se muestra en la Figura 4–18, que describen suelo y agua respectivamente; el algoritmo de reconocimiento indicaría que el objeto A pertenece a la capa temática **suelo**, mientras que el objeto geográfico B es un **cuerpo de agua**. Tomando en cuenta información cuantitativa de las intensidades de cada objeto, posiblemente se puede especializar más el nombre de cada objeto geoespacial, pero lo que realmente proporciona más información es la forma en que están relacionados estos dos objetos, ya que el objeto A (suelo) está **dentro de** o **rodeado por** el objeto geográfico B (cuerpo de agua). Este conocimiento acerca de las relaciones entre los objetos geográficos permite llegar a una especialización que sería una **isla** o **islote**.



Figura 4–18. Relación entre dos objetos geográficos.

Consideremos la definición de una isla: es una porción de tierra menor que un continente que se encuentra totalmente **rodeada de** agua (Wikipedia, 2007). Con lo indicado anteriormente, sabemos que existe un objeto geográfico tipo **suelo** y un **cuerpo de agua**, pero falta determinar dos datos adicionales; primeramente, que el objeto A es una porción de tierra menor que un continente y que la tierra se encuentra totalmente **rodeada del** objeto B. La relación **rodeado por** cuenta con diversas interpretaciones, como se puede apreciar en la Figura 4–19(a)<sup>27</sup> donde el objeto C está rodeado de objeto D, cuando la relación es entre dos objetos únicamente; sin embargo, analizando la Figura 4–19(b), se podría concluir que el objeto E está rodeado de F, G y H e I, pero en esta configuración, están involucrados varios objetos. En conclusión, se deben definir las relaciones entre objetos geográficos tomando en consideración el contexto, estipulando claramente la interpretación que se está haciendo de una relación en particular.

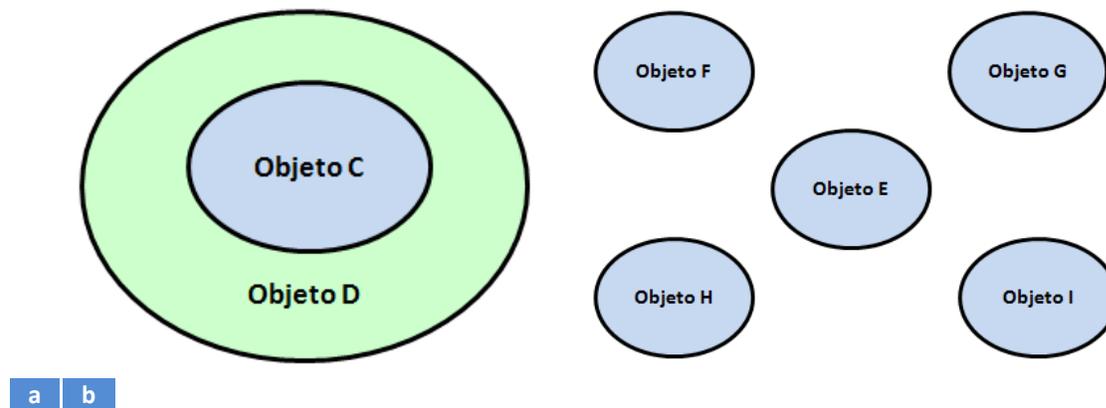


Figura 4–19. Configuraciones diferentes donde se cumple la relación *rodeado por*.

#### 4.6.4 Representación alterna

En la mayoría de los casos, los especialistas prefieren trabajar con los datos en formato vectorial. Lo anterior se debe principalmente a una mayor precisión al realizar operaciones de análisis espacial, así como una mayor flexibilidad para el manejo de la información. La vectorización consiste en, dado un conjunto de datos en formato raster, obtener las representaciones mediante primitivas vectoriales de dicho conjunto de datos. Para ello, se utilizan tres tipos de primitivas básicas: puntos, líneas y polígonos.

<sup>27</sup> En caso de usar la relación **dentro de**, en lugar de, **rodeado por**, no se generaría la ambigüedad citada en este ejemplo. Sin embargo, si se examinan las definiciones de objetos geográficos, como la de isla, se dice que es una porción de Tierra **rodeada por** un cuerpo de agua.

El proceso de vectorización es un problema bastante complejo para el que ya se han propuesto diferentes soluciones; en este trabajo de investigación solamente se proponen las técnicas que permiten iniciar un proceso de vectorización, integrado por la generación mediante primitivas de los objetos geoespaciales.

Teniendo la especialización de los objetos geográficos, para cada objeto geográfico que forma parte de la ontología se agrega una propiedad que nos permite saber la forma en que se debe representar el objeto. El rango de valores para esta propiedad es precisamente el de las primitivas básicas. Tiene que ser una propiedad de cada objeto y no de la temática, porque no siempre se cumple que todos los objetos de una determinada temática se representan mediante la misma primitiva. Con esta propiedad, es posible generar una primitiva raster de descripción alterna  $P_{r_i}$  para cada objeto geográfico de  $I_O$ .

Para modelar objetos simples, las primitivas básicas de abstracción son punto, línea y polígono (región), definidos como (Güting, 1994):

- Un punto representa un aspecto geométrico de un objeto para el cual solamente es relevante su ubicación en el espacio, pero no su extensión. Por ejemplo, aunque una ciudad describe u ocupa cierta región espacial, se puede representar puntualmente, tomando como referencia el centro de la ciudad.
- Una línea es una curva en el espacio<sup>28</sup>, representada usualmente por una poli-línea (secuencia de segmentos), que es una abstracción básica para desplazarse a través del espacio, o generar conexiones en él. Las líneas se pueden usar para representar, por ejemplo, ríos, medios de comunicación como carreteras, líneas de transmisión de comunicación (electricidad, red telefónica, etc.) y otras.
- Una región es una abstracción para un objeto que tiene una extensión en el espacio bidimensional. Se caracteriza porque puede contener huecos o estar integrada por distintas piezas aisladas o separadas, pero que juntas representan una sola región.

---

<sup>28</sup> En este contexto, siempre entenderemos por línea una curva en el espacio, aunque en otras áreas como la geometría analítica no se aplique esta interpretación.

#### 4.6.4.1 Representación de objetos geográficos mediante puntos

Si la estructura geométrica de todos los objetos geográficos fueran de regiones cóncavas, se puede usar el momento de primer orden (Gonzalez & Woods, 2002). Para generar una representación puntual de una región  $R$  se tiene que identificar primeramente si es un objeto convexo o cóncavo. Para ello se puede determinar el cerco convexo utilizando la técnica del mismo nombre. Cuando se cumple que el número de elementos que pertenecen a la diferencia entre la región del cerco convexo y la región original es mayor que un umbral dado, entonces se dice que la región  $R$  es convexa; en caso contrario se interpretará como una región cóncava. Con este resultado se aplica uno de los siguientes métodos:

1. Objeto geográfico convexo. Para este objeto geográfico, el elemento puntual que lo representa está dado por su centro geométrico.
2. Objeto geográfico cóncavo. En esta situación, se determina el punto medio del esqueleto de la región (ver siguiente sección).

En específico para objetos geográficos, generalmente la representación mediante puntos se aplica a regiones convexas.

#### 4.6.4.2 Representación de objetos geográficos mediante líneas

Para generar la representación lineal es necesario tratar de preservar las características topológicas y geométricas de un objeto geográfico, lo cual se puede lograr determinando el esqueleto del objeto geográfico.

#### 4.6.4.3 Representación de objetos geográficos mediante bordes

Para este caso se requiere extraer el conjunto de píxeles que describen el contorno del objeto geoespacial de interés y eliminar aquellos elementos discretos que sean redundantes<sup>29</sup>, según el

<sup>29</sup> Un píxel es redundante cuando puede ser removido del contorno y no se rompe la conectividad en el contorno del objeto.

tipo de conectividad empleada. Para poder determinar el borde de una región  $R_j$  que describe un objeto geográfico  $O_j$  se usa la Definición 4.4.

**Definición 4-7.** Un conjunto de puntos  $S = \{r(i, j) \mid i, j \in \mathbb{Z}\}, S \subset \mathbb{Z}^2$  se dice que están  $k$ -conexos, si existe un camino que comienza en el punto  $p_1$  y finaliza en el punto  $p_2$  para todo  $p_1, p_2 \in \mathbb{Z}^2$ . Donde  $k$  es el tipo de vecindad entre pixeles y que puede tomar el valor 4, 8, 6R ó 6L.

**Definición 4-8.** El contorno de un objeto geográfico, denotado por  $\alpha(O_i)$ , es el conjunto de todos los pixeles que pertenecen a la región  $R_i$  del objeto geográfico  $O_i$  con al menos uno de sus ocho vecinos que no está en  $R_i$ .

Para determinar los elementos del objeto que deben formar parte de su contorno, simplemente se barre la imagen si y solo si se cumplen dos condiciones: que el píxel en cuestión pertenece al objeto geográfico en cuestión; y si al menos uno de sus  $k$ -vecinos no pertenece al objeto, en cuyo caso es marcado como de contorno.

Al aplicar un detector que permita obtener el contorno de una forma, se pueden generar pixeles redundantes. Se puede demostrar que los pixeles redundantes son aquellos que cumplen con alguna de las máscaras de la Figura 4-20, por lo que después de aplicar la detección de contorno se puede hacer un podado de los pixeles redundantes.

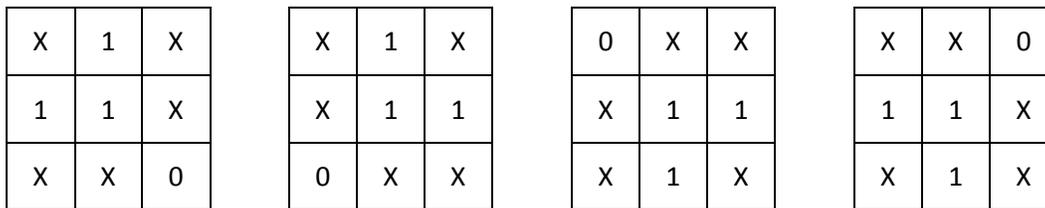


Figura 4-20. Máscaras para la detección de pixeles redundantes.

En la Figura 4-21 se muestra un tramo de un contorno antes y después de eliminar los pixeles redundantes.

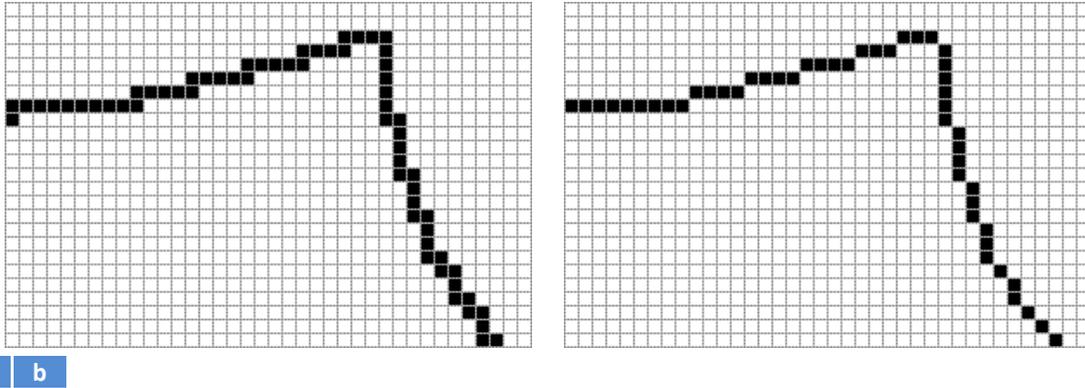


Figura 4–21. Resultados de la eliminación de pixeles redundantes (a) Segmento de contorno original; (b) Resultado después de eliminar los pixeles redundantes.

El resumen de los algoritmos de la etapa de síntesis se describe en la Tabla 4–2.

Tabla 4–2. Resumen de la etapa de síntesis.

Síntesis	
<b>Extracción de regiones</b>	
Opcional	No
Entrada	$I_{CCS}$
Parámetros	No requiere
Salida	$\{I_{T_1}, I_{T_2}, \dots, I_{T_{n_{rCS}}}\}$
<b>Reconocimiento de objetos geográficos</b>	
Opcional	No
Entrada	$\{I_{T_1}, I_{T_2}, \dots, I_{T_{n_{rCS}}}\}$
Parámetros	No requiere
Salida	$T_{I_o} = \{eg_1, eg_2, \dots, eg_{n_{rCS}}\}$
<b>Especialización de los objetos geográficos</b>	
Opcional	No
Entrada	$\{I_{T_1}, I_{T_2}, \dots, I_{T_{n_{rCS}}}\}$ $\{eg_1, eg_2, \dots, eg_{n_{rCS}}\}$
Parámetros	Nivel de detalle de la geoimagen
Salida	$S_{I_o} = \{ee_1, ee_2, \dots, ee_{n_o}\}$

Tabla 4–2. (Cont...) Resumen de la etapa de síntesis.

Síntesis	
Representación alterna	
Opcional	Si
Entrada	$\{I_{T_1}, I_{T_2}, \dots, I_{T_{n_{rcs}}}\}$
Parámetros	No requiere
Salida	$\{P_{r_1}, P_{r_2}, \dots, P_{r_{n_o}}\}$
<b>Resultado final</b>	$S = \{ee_1, ee_2, \dots, ee_{n_o}\}$ $RA = \{P_{r_1}, P_{r_2}, \dots, P_{r_{n_o}}\}$

## 4.7 Ontología para la especialización de los objetos geográficos

Definir una ontología de algún dominio no es una tarea fácil, incluso para un experto. Es un trabajo que requiere realizar mejoras a un prototipo inicial que permita ir refinándola, lo cual indica que es susceptible a cambios o ajustes posteriores. De hecho, una característica importante de las ontologías es que permiten agregar más conocimientos conforme el experto adquiere nuevas habilidades.

Algunos autores como **Noy & McGuinness, 2001** han definido una serie de pasos con el propósito de ayudar a terceras personas en el diseño de ontologías. Estos autores retoman algunas ideas del diseño orientado a objetos (**Rumbaugh et al., 1991; Booch et al., 1997**), así como su experiencia en el uso de entornos de edición de ontologías como Protégé-2000 (**Protege, 2007**), Ontolingua (**Ontolingua, 1997**), Chimaera (**Chimaera, 2000**). Es importante resaltar que acorde con estos autores no se hace una separación entre una conceptualización y una ontología.

### 4.7.1 Preguntas de competencia

El primer paso para definir una ontología, es tratar de dar respuesta a las siguientes interrogantes:

1. *¿Cuál es el dominio que cubrirá la ontología?*

El dominio de la ontología será el de los objetos geográficos.

2. *¿Para qué se usará la ontología?*

Para especializar el tipo de un objeto geográfico por medio de una conceptualización de las propiedades geométricas y topológicas de los objetos.

Esta ontología permitirá conceptualizar el conocimiento que poseen los seres humanos para hacer una especialización de los objetos geográficos. En este trabajo, el realizar una conceptualización completa de todo dominio geográfico está fuera de su alcance; otros trabajos como el de **Torres, 2007** abordan detalladamente este problema. Una de las aportaciones de este trabajo es el mecanismo para asociar una ontología de dominio geográfico en el problema de la especialización de los objetos geográficos.

3. *¿Para qué tipos de preguntas la información en la ontología deberá proporcionar respuestas?*

Sea un conjunto de objetos geográficos etiquetados por el algoritmo de reconocimiento. La ontología debe ser capaz de responder lo siguiente:

- a. ¿Cuál es la relación o relaciones que puede existir entre estos objetos geográficos?
- b. ¿Cuál es el nivel de detalle de un objeto geográfico dado?
- c. ¿Cuáles son las operaciones de procesamiento digital de imágenes que describen una determinada relación topológica?

4. *¿Quién usará y dará mantenimiento a la ontología?*

El uso de la metodología en general queda abierto a cualquier persona con conocimientos generales del área de la geocomputación; sin embargo, el mantenimiento de la ontología es responsabilidad de un especialista, ya que la forma de describir y relacionar objetos geográficos requiere conocimientos más específicos.

#### 4.7.2 Reutilización de ontologías existentes

Para este punto, recomiendan considerar otras ontologías que se han definido y verificar si es posible refinar y extender recursos existentes para un dominio y tarea en particular. Reutilizar ontologías existentes puede ser un requerimiento si un sistema necesita interactuar con otras aplicaciones que trabajan con ontologías particulares o vocabularios controlados. Actualmente,

existen muchas ontologías disponibles en forma electrónica y pueden ser importadas dentro de un entorno de desarrollo de ontologías, como son las ontologías públicas de Ontolingua, DAML u otras comerciales entre las que destacan UNSPSC, RosettaNet y DMOZ. Comúnmente el formalismo en el cual está expresada una ontología no interesa, puesto que muchos sistemas de representación de conocimiento pueden importar y exportar ontologías.

En este trabajo se plantea un nuevo mecanismo que permite particularizar la identidad de los objetos mediante un etiquetado general, junto con el empleo de una ontología que permite describir las propiedades geométricas y topológicas de estos objetos, lo cual no ha sido analizado previamente por otros trabajos. Por lo anterior, este punto en particular se considera un trabajo futuro.

Sin embargo, con respecto a la conceptualización del dominio geográfico, existen trabajos existentes que, aunque no están representados por medio de ontologías, fueron de utilidad para la definición de la ontología propuesta en este trabajo:

1. Wikipedia. Es una enciclopedia libre plurilingüe basada en la tecnología wiki. Wikipedia se escribe de forma colaborativa por voluntarios, permitiendo que la gran mayoría de los artículos sean modificados por cualquier persona con acceso mediante un navegador Web ([Wikipedia, 2007](#)). Tiene la ventaja de incluir definiciones mucho más extensas que otros diccionarios, así como ejemplos ilustrativos de cualquier tipo de objeto geográfico.
2. Especificación de codificación de datos ambientales (*Environmental Data Coding Specification – EDS*). El estándar ISO/IEC 18025:2005(E) es una norma ISO de la categoría de tecnologías de información que proporciona mecanismos para especificar de forma clara y sin ambigüedades a los objetos empleados para modelar conceptos ambientales. Cuenta con una base de datos en formato de Microsoft Access, así como en Excel ([EDCS, 2005](#)).
3. Diccionario de datos del INEGI. Los diccionarios de datos del INEGI, disponibles en escala de 1:50,000, 1:250,000 y 1:1,000,000 contienen la definición de objetos geográficos. Desafortunadamente tiene dos inconvenientes, la primera es que estos diccionarios están enfocados a usarse con representaciones vectoriales (los datos están descritos como punto,

líneas o áreas), y por otra parte, omite la inclusión de características geométricas en la definición de los objetos geoespaciales (INEGI, 2007).

4. Diccionarios de la Real Academia de la Lengua Española y WordNet. Son la base en la búsqueda de los conceptos de objetos geográficos, pero como ya se explicó en párrafos anteriores, existe ambigüedad en sus definiciones de ciertos objetos (Real Academia, 2007; WordNet, 2007).

#### 4.7.3 Términos importantes para la ontología

Consiste en escribir una lista con todos los términos con los que se quisieran hacer enunciados o dar explicación a un usuario. ¿Cuáles son los términos de los cuales se desea hablar? ¿Qué propiedades tienen esos términos?

En este punto se tiene que comenzar a hacer una abstracción de los elementos (clases) que se desea formen parte de la ontología, la forma en que están relacionados (relaciones) y los atributos que los caracterizan (propiedades).

Se puede entender que esta parte consiste en hacer la conceptualización de los objetos geográficos y que son descritos formalmente por las siguientes dos etapas.

#### 4.7.4 Definición de las clases y jerarquías de clases

Se tiene que hacer una definición de las clases y de la jerarquía de clases. Hay varios enfoques para desarrollar una jerarquía de clases (Uschold and Gruninger, 1996):

- Un proceso de desarrollo de arriba hacia abajo (*top-down*) comienza con la definición de los conceptos más generales en el dominio seguida de una especialización de los conceptos.

- Un proceso de desarrollo de abajo hacia arriba (*bottom-up*) comienza con la definición de las clases más específicas (las hojas de la jerarquía) con el subsecuente agrupamiento de esas clases en conceptos más generales.
- Un proceso de desarrollo combinado es el resultado de una combinación de los enfoques *top-down* y *bottom-up*: primero definimos los conceptos más sobresalientes y luego los generalizamos y especializamos apropiadamente.

Se analizó previamente que existen dos cuestiones importantes a tomar en cuenta para la conceptualización: las características topológicas y las geométricas. Por ello, se requiere que la conceptualización esté integrada por diversas clases que permitan hacer la especialización involucrando ambas características. La ontología del dominio geográfico estará integrada en un primer nivel por clases abstractas de dos tipos:

1. **Nivel de detalle.** Debido a que se persigue hacer una representación implícita de la identidad o tipo de un objeto geoespacial, acorde con sus propiedades geométricas como el tamaño, esta clase permite generalizar la clasificación de los objetos geográficos según su escala y cuenta con las siguientes subclases: escala muy gruesa, escala gruesa, escala media, escala fina y escala muy fina.

Para un experto, es más intuitivo proporcionar información de forma cualitativa y no cuantitativa acerca del detalle de una geoimagen a procesar. Por ejemplo, observando la Figura 4–17, si se cuestiona al usuario acerca de la escala, posiblemente no pueda responder a esta pregunta o proporcione un dato muy fidedigno; sin embargo, indicar la granularidad de los datos geográficos es tanto más preciso como fácil de determinar<sup>30</sup>.

2. **Temática.** El propósito de esta clase será el de describir los objetos geográficos acorde con el etiquetado que hará el algoritmo de reconocimiento. Asimismo, servirá como base para permitir la especialización de los datos geoespaciales por medio de la descripción de las

---

<sup>30</sup> Se asume que el usuario es experto en el manejo de datos espaciales, por lo que determinar la granularidad de los datos geográficos no es algo complicado o desconocido para él.

relaciones topológicas. Las subclases de esta clase es el conjunto de temáticas  $T = \{t_1, t_2, t_3, \dots, t_{n_T}\}$ .

La ramificación de la ontología que especializa los objetos geográficos con base en el nivel de detalle, permite hacer una descripción de las propiedades geométricas de los objetos, mientras que la clasificación por temática considerará las propiedades topológicas (relaciones) de los mismos. Hasta este punto, la ontología tendrá el aspecto que se ilustra en la Figura 4–22.

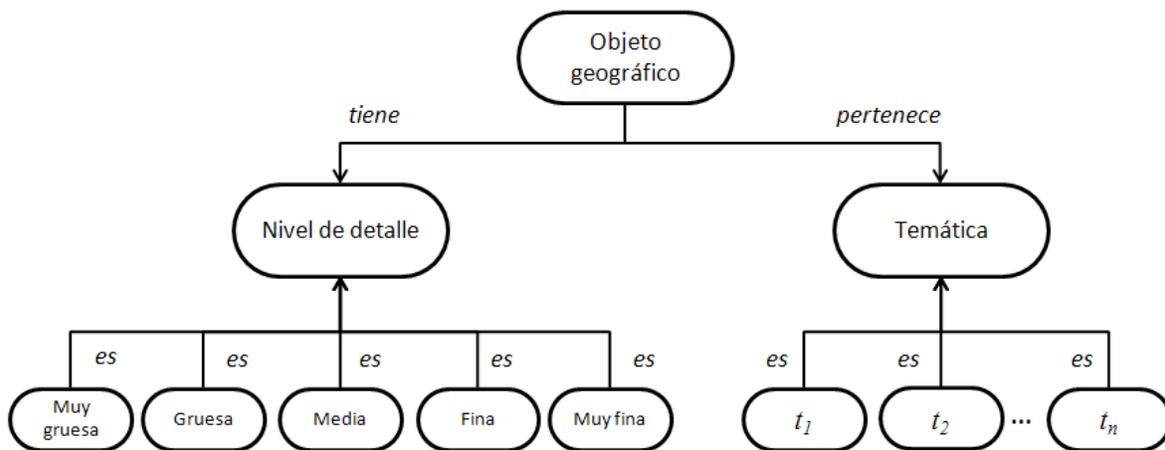


Figura 4–22. Ontología para la especialización de objetos geográficos.

La forma en que se interpretará esta ontología es que un **objeto geográfico** tiene un **nivel de detalle** y pertenece a una cierta **temática**. Para cada una de las temáticas ( $\forall t_i \in T$ ), se agregan como una nueva clase todos los objetos geográficos que pertenecen a esta temática; esto se puede apreciar en la Figura 4–23.

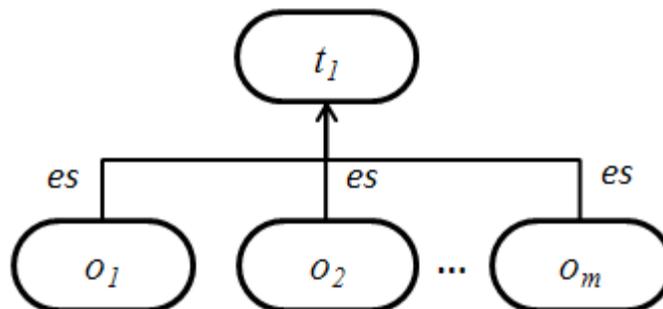


Figura 4–23. Adición de los objetos geográficos a una temática.

En algunos casos puede ser necesario agregar clases intermedias que permiten generalizar otros tipos de objetos geográficos, por ejemplo, los lagos y lagunas son clasificados como objetos tipo *INLAND*. Al agregar una clase intermedia, lo que se hace es añadir más nivel de descripción al momento de especializar un objeto geográfico, ver Figura 4–25 y Figura 4–26.

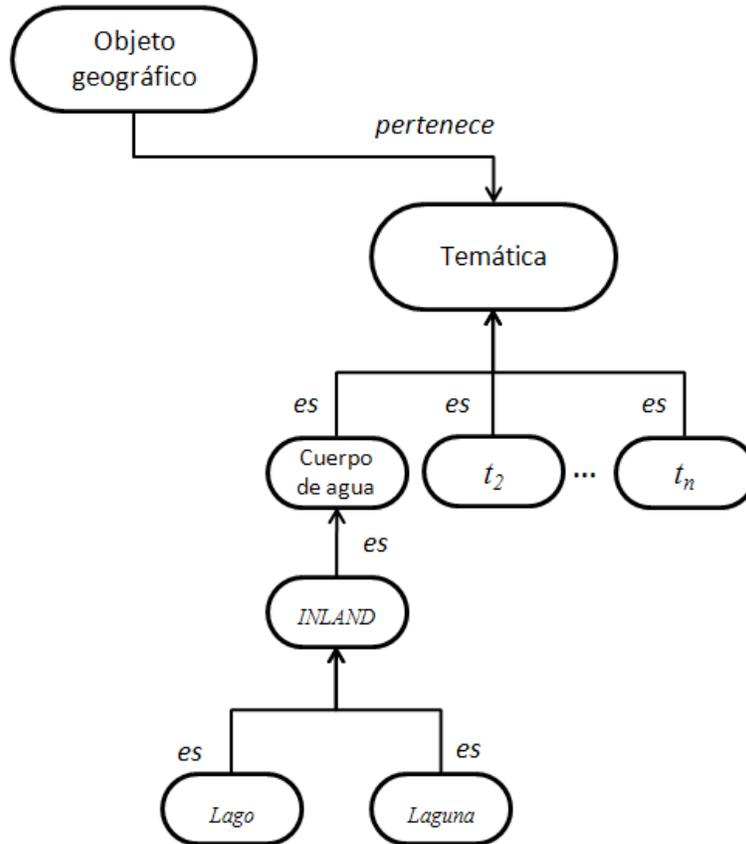


Figura 4–24. Ontología con diferentes clases de especialización, caso 1.

Considerando la especialización de un lago y una laguna con la clase *INLAND*, Figura 4–24, y sin la clase *INLAND*, Figura 4–25, las descripciones del objeto podrían ser correspondientemente: “es un objeto geográfico que pertenece a la temática de cuerpo de agua...”, mientras que agregando la clase *INLAND* se tendría: “es un objeto geográfico que pertenece a la temática de cuerpo de agua, el cual es un *INLAND*”.

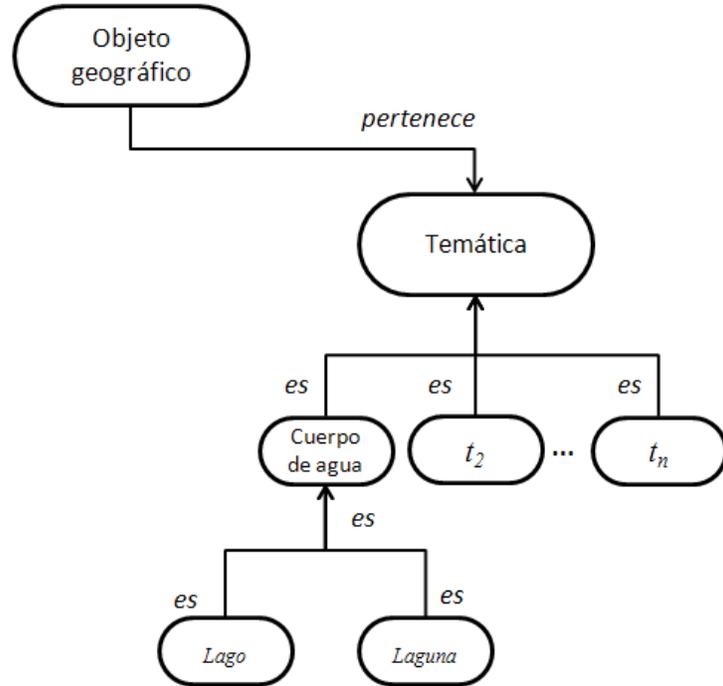


Figura 4–25. Ontología con diferentes clases de especialización, caso 2.

Es necesario definir las relaciones entre los objetos geográficos, con las cuales se describan las condiciones topológicas en las que es posible encontrar dichos objetos en un determinado contexto. Una relación está descrita por medio de un concepto, por ejemplo, carretera **cerca de** áreas verdes o edificios **dentro de** área urbana. La forma general de las relaciones es:

$$\begin{aligned}
 & o_i \mathcal{R} o_j \\
 & o_i \mathcal{R} t_k \\
 & t_k \mathcal{R} t_l \\
 & \forall i \neq j; i, j < n_T; k, l < n_{ont}
 \end{aligned}
 \tag{4.36}$$

Donde  $n_T$  es el número de temáticas y  $n_{ont}$  es igual al número de objetos geográficos en la ontología. Para poder agregar las relaciones entre objetos geográficos dentro de la ontología, se debe generar una nueva clase por cada nueva relación. Por ejemplo, un cuerpo de agua tipo lago o laguna siempre está relacionado con un objeto geográfico tipo suelo por medio de la relación **rodeado**, tal como se muestra en la Figura 4–26.

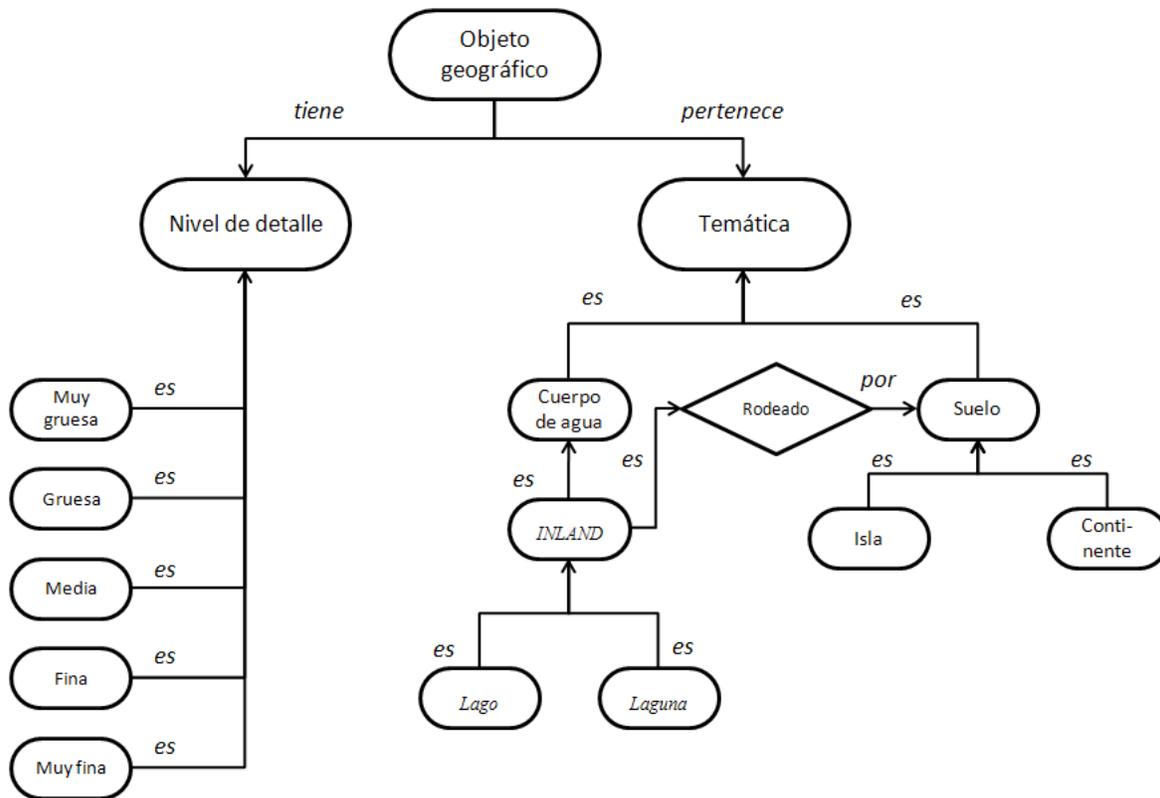


Figura 4–26. Ejemplo de una ontología con la relación *rodeado*.

En consecuencia, con la ontología es posible identificar algunos tipos de accidentes geográficos. Un accidente geográfico es una unidad geomorfológica. Los accidentes geográficos se clasifican por características tales como elevación, pendiente, orientación, estratificación, exposición de roca y tipo de suelo; como ejemplos de accidentes geográficos son los montes, acantilados, valles, etcétera. Los océanos y continentes son los accidentes de orden máximo. Los elementos de los accidentes geográficos son aquellas partes que pueden identificarse en ellos. Los elementos genéricos de los accidentes geográficos son los fosos, picos, canales, crestas, pasos, estanques, llanos, etcétera, y pueden extraerse a menudo de modelos digitales de elevación usando algunas técnicas automáticas o semiautomáticas (Wikipedia, 2007).

Con la información disponible en una geoimagen no es posible hacer un reconocimiento e identificación de todos los tipos de accidentes geográficos; por ejemplo, considérese una imagen de los acantilados de Mohar en Irlanda; en la Figura 4–27 se puede apreciar una captura de esta región desde un satélite. Si se hiciera una descomposición semántica de esta geoimagen sería casi imposible especializar la región de suelo e indicar que es un acantilado, debido a que la

información de bajo nivel de la geoimagen no proporciona detalles de la elevación o altura del objeto geográfico. Por lo anterior, es necesario contar con datos geospaciales que describan la característica de elevación como son los Modelos Digitales de Elevación (*DEM – Digital Elevation Model*).

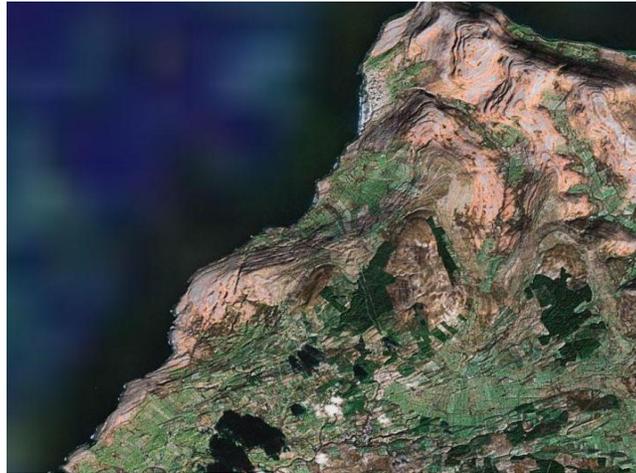


Figura 4–27. Geoimagen de los Acantilados de Mohar, Irlanda.

Independientemente del tipo de relación, para que el algoritmo de especialización pueda hacer uso de la ontología, es necesario determinar cuándo los objetos geográficos o capas temáticas pueden vincularse por medio de una determinada relación. Por ello, una relación debe poder ser descrita como un conjunto de operadores o funciones de las existentes en el procesamiento digital de imágenes o del análisis topológico.

#### 4.7.5 Definición de las propiedades de las clases: *slots*

Las clases aisladas no proporcionan información suficiente para responder las preguntas de competencia del paso 1. Una vez que se han definido algunas de las clases, se debe describir la estructura interna de los conceptos.

Las relaciones permiten vincular dos objetos geográficos, un objeto geográfico con una temática, o bien, dos conceptos tipo temática. Los casos prácticos mencionados, nos permiten concluir que los conceptos que describen relaciones, pueden o no ser ambiguos, es decir, la relación **dentro de** no puede presentar algún tipo de ambigüedad, ya que es fácil determinar cuándo un segmento

está dentro de otro, haciendo uso de la operación intersección. Para la relación **cerca de**, puede existir un inconveniente, ya que una persona puede considerar dos segmentos cerca, cuando la distancia (empleando algún tipo de función distancia) es menor a 200, mientras que para otro individuo un valor de 500 es más adecuado. Dentro del conjunto de atributos que pueden tener las clases están los que permiten describir cuantitativamente una relación.

## 4.8 Descomposición semántica iterativa

El objetivo de la comprensión semántica es hacer una simplificación de los objetos geográficos, para que su reconocimiento y especialización sean más fáciles de lograr. Por ejemplo, considerando la geoimagen de la siguiente figura, podemos observar que existen muchas variaciones de color, textura, entre otros. Sin embargo, existen tres objetos importantes o relevantes: el objeto geográfico que describe el suelo (la República Mexicana), así como dos cuerpos de agua. Esto sería una descomposición aceptable, aunque se puede presentar la situación de que un experto requiera mayor especialización de la geoimagen, es decir, tal vez le interese obtener los objetos geográficos acorde con el tipo de suelo.



Figura 4–28. Geoimagen de la República Mexicana.

En consecuencia se requiere hacer una mayor descomposición semántica de la geoimagen. Cuando se presentan estas situaciones, el optar desde un inicio por hacer una simplificación de menor nivel de las regiones de los objetos geográficos, conlleva a una mayor complejidad para las etapas de reconocimiento y especialización. Retomando la primera descomposición propuesta, es más simple (incluso para un ser humano) primero determinar estas tres regiones y luego sobre

cada región obtener más detalles o información; en el caso del cuerpo de agua del lado izquierdo de la República Mexicana (océano Pacífico) una vez que se tiene la región que describe este objeto geoespacial, considerando únicamente esta región, se puede aplicar la metodología de descomposición semántica para especializar lo contenido por este cuerpo de agua. Por lo tanto, ahora se obtendrían un cuerpo de agua profunda y las islas, por citar algunos ejemplos. Lo mismo aplica para los otros objetos geográficos.

En resumen, la metodología de descomposición semántica también puede aplicarse iterativamente de tal forma que el usuario puede obtener el nivel deseado de especialización de los objetos geográficos, logrando un mayor nivel de descomposición semántica. Esto es, en cada iteración de la metodología completa, se obtienen más conocimientos sobre los objetos geográficos, pero es importante resaltar que esto implica que el algoritmo de reconocimiento sea más fino con cada nivel de descomposición semántica, así como una ontología de aplicación más completa.

## 4.9 Comentarios y observaciones

El algoritmo de descomposición semántica se basa en dos grandes etapas, una de análisis y otra de síntesis. Aplicando una técnica de compresión semántica se realiza una simplificación de la información geoespacial original, generando un conjunto de regiones de intensidad de color homogénea. El conjunto completo de características y parámetros definidos en el algoritmo de compresión semántica, permiten hacer una simplificación de los objetos geográficos sin requerir de otro tipo de algoritmo adicional (preprocesamiento o posprocesamiento). Pero para esta adecuada simplificación, se requiere manipular el umbral de similitud, el número de iteraciones y la geoimagen de entrada en cada nueva iteración.

La etapa de síntesis tiene el propósito de generar un listado de etiquetas en lenguaje natural de los objetos geográficos identificados en una geoimagen. Un algoritmo de reconocimiento o etiquetado general permite asociar etiquetas de temática a cada objeto geográfico, por medio de un algoritmo sencillo basado en vectores prototipos determinados a partir de las características de intensidad de color de regiones representativas de cada temática. Terminado el reconocimiento

general, se tiene una lista en lenguaje natural de las temáticas presentes en la geoimagen, ejemplo: cuerpo de agua, suelo, etc.

Por último, el algoritmo de especialización realiza una consulta en una ontología de aplicación del dominio geográfico para tratar de hacer una especialización en la identidad de los objetos geográficos, considerando las propiedades geométricas y topológicas entre los objetos. Con ello se responde a la interrogante ¿Qué objetos geográficos hay en la geoimagen?, como respuesta puede ser: una isla, un islote, uno o más lagos, etc.

Asimismo, se definió un algoritmo de representación alterna para poder generar primitivas raster tipo punto, línea o área para la descripción de objetos geográficos.

## Capítulo 5

### Resultados experimentales

#### ***Resumen***

A continuación se mostrarán los resultados obtenidos aplicando el algoritmo de descomposición semántica a un conjunto de geoimágenes que describen diferentes accidentes geográficos, explicando cada una de las etapas de la metodología propuesta. Se muestra la capacidad del algoritmo de compresión semántica de poder hacer una simplificación de los objetos geográficos sin la necesidad de una etapa previa de preprocesamiento ni posprocesamiento, así como la utilidad de las diferentes características propuestas para poder fusionar los segmentos de cada objeto geográfico. Al final del capítulo se muestra una comparación con una herramienta comercial que permite hacer segmentación basada en conocimiento, con lo que se podrán determinar las diferencias, mejoras y áreas de oportunidad del algoritmo de descomposición semántica.



## 5.1 Introducción

En este capítulo se muestran los resultados obtenidos con el sistema de descomposición semántica de geoimágenes. Se eligió hacer una comparación de los resultados del algoritmo de descomposición semántica en un caso en particular (una isla) con la herramienta comercial eCognition, dado que es un programa que tiene varios años de desarrollo y está enfocado al procesamiento orientado a objetos de datos geoespaciales, para poder ejemplificar claramente las ventajas de la metodología desarrollada en esta tesis. Se podrá observar que nuestros resultados superan claramente a lo que existe actualmente en varios puntos:

1. **Identificación de la semántica.** A diferencia de otros trabajos afines, la metodología de descomposición semántica parte de una geoimagen y genera como resultado final, un conjunto de etiquetas que describe la semántica de la geoimagen. Otros sistemas, solamente realizan parte de lo propuesto en este trabajo, produciendo como salida una imagen digital, es decir, el resultado sigue siendo un conjunto de datos raster.
2. **Uso de diferentes características para la compresión semántica.** Emplear características de color para analizar información geoespacial, no permite obtener resultados adecuados en todos los casos, sobre todo por la heterogeneidad de la información a procesar, tal como ocurre con las geoimágenes que se analizarán en una sección de este capítulo. El conjunto propuesto de 10 características y parámetros (distancia, largo, ancho, tamaño, área, intensidad, intensidad promedio,  $\sin \Delta$ ,  $\cos 2\gamma$ ,  $\sin 2\gamma$ ) para la compresión semántica permite obtener muy buenos resultados para todas las geoimágenes.
3. **Menor cantidad de parámetros para controlar la fusión de regiones.** Uno de los inconvenientes que tiene eCognition es su elevada cantidad de información *a priori* que debe proporcionarse para poder obtener resultados satisfactorios. En esta propuesta, el algoritmo de compresión semántica requiere menor intervención y experiencia del usuario para poder extraer las regiones que describen los objetos geográficos.

## 5.2 Sistema desarrollado

Se trabajó con las herramientas C++ Builder y MATLAB para poder llevar a cabo las pruebas correspondientes de la metodología de descomposición semántica. La siguiente tabla indica la herramienta utilizada para cada algoritmo propuesto.

Tabla 5–1. Herramientas usadas para los algoritmos de la metodología de descomposición semántica.

Algoritmo	C++ Builder	MATLAB
Compresión semántica	X	
Extracción de regiones	X	
Reconocimiento		X
Especialización		X
Representación alterna	X	

Asimismo, el sistema cuenta con diversas opciones típicas de las aplicaciones de sistemas operativos de interfaz gráfica, como son: abrir archivo, guardar, guardar como, copiar al portapapeles, pegar del portapapeles, organizar ventanas en cascada, tiras horizontales o tiras verticales, imprimir. La pantalla principal de la aplicación de descomposición semántica, se muestra en la Figura 5–1.

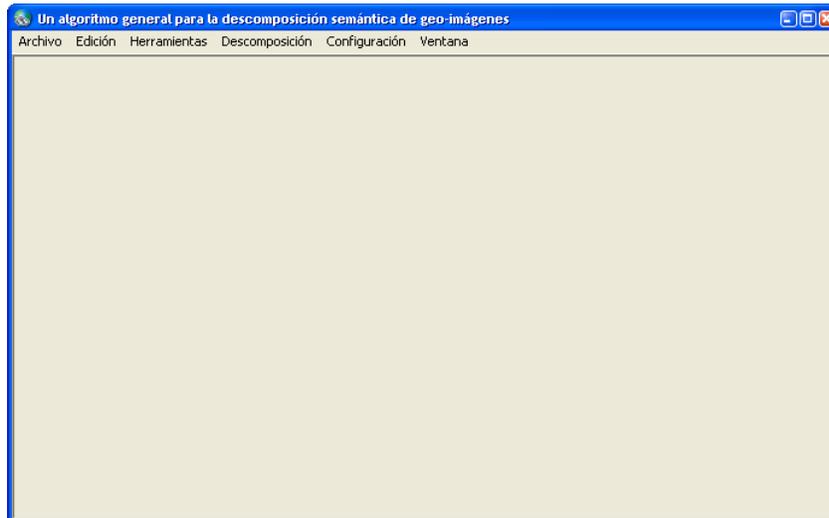


Figura 5–1. Formulario principal de la aplicación de descomposición semántica.

Tal como ya fue expuesto, el algoritmo de compresión semántica genera un total de diez geoimágenes. En la Figura 5–2 aparece la geoimagen original (la que se encuentra en el extremo

superior izquierdo) y el conjunto de geoimágenes que se obtienen después de una iteración del algoritmo de compresión semántica, con un umbral de similitud predeterminado<sup>31</sup> ( $k_{CS} = 50$ ). Por cuestiones prácticas, en los resultados que se muestran posteriormente se agrega el contenido de las geoimágenes por separado, omitiendo el contenido de la interfaz gráfica de la aplicación para poder apreciar de mejor forma las geoimágenes generadas.

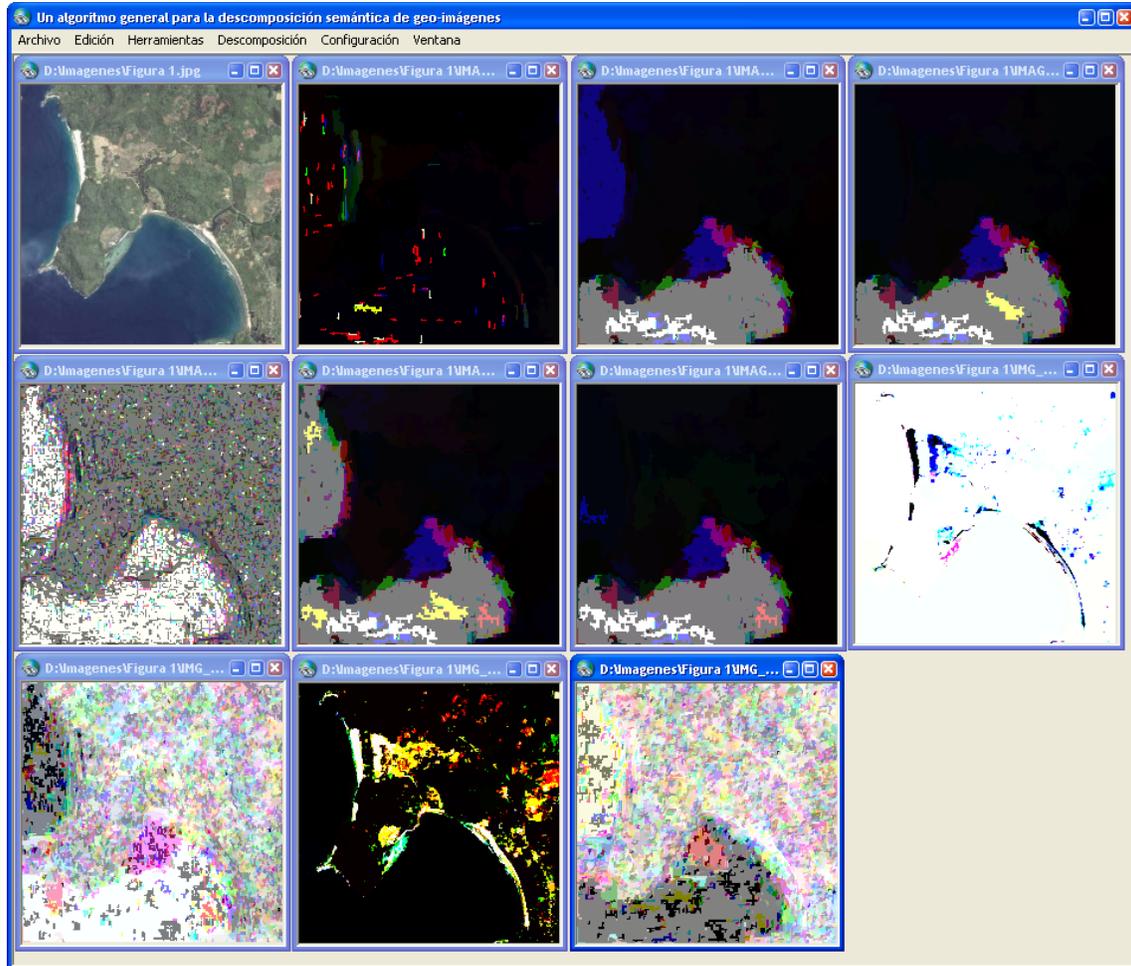


Figura 5–2. Conjunto de imágenes de salida del algoritmo de compresión semántica.

<sup>31</sup> Este valor predeterminado del umbral de similitud fue elegido al azar, aunque también puede ser personalizado dentro de las opciones del sistema implementado.

### 5.3 Algoritmo de reconocimiento

Para poder realizar las pruebas correspondientes de los casos de estudio, se definió una ontología de aplicación que servirá de base para ejemplificar el funcionamiento de la descomposición semántica con dos temáticas diferentes. Se usó un algoritmo de reconocimiento generalizado al espacio de color RGB basado en la técnica de distancias Euclidianas mínimas. El algoritmo de distancias mínimas parte de una etapa inicial<sup>32</sup> en la que se determinan los centroides de las clases en las que se puede clasificar un vector. Para la definición de los centroides, generalmente se utiliza un conjunto de muestras en las que se conoce de antemano, la clase a la que pertenece cada muestra. Esto se realiza con el objeto de modelar las clases, siendo el número de muestras una elección personal. Se propuso el siguiente conjunto de temáticas:

$$T = \{\text{cuerpo de agua, suelo}\}$$

Donde cuerpo de agua es la primera temática ( $t_1$ ) mientras que suelo es la segunda temática ( $t_2$ ). Se empleó un conjunto de 12 geoimágenes con un tamaño de 200 x 200 pixeles que contienen pixeles exclusivamente de cuerpos de agua, así como 24 regiones del mismo tamaño para la temática de suelo. Para el caso de la temática de suelo, el primer grupo describe regiones de suelo claras, como puede ser tierra, rocas, entre otros, mientras que el segundo grupo es para describir zonas verdes, la cuales también son categorizadas en la temática de suelo.

Las regiones se pueden observar en las siguientes tres figuras<sup>33</sup>, las que se obtuvieron de un conjunto aleatorio de geoimágenes satelitales. Se utilizaron regiones rectangulares porque en este caso las características geométricas de los objetos geográficos no son relevantes, sino solamente se está interesado en la información de intensidad de los objetos geoespaciales.

---

<sup>32</sup> Comúnmente esta etapa es conocida como etapa de entrenamiento.

<sup>33</sup> Este conjunto fue suficiente para poder hacer una definición adecuada de vectores prototipo para las pruebas de este capítulo.

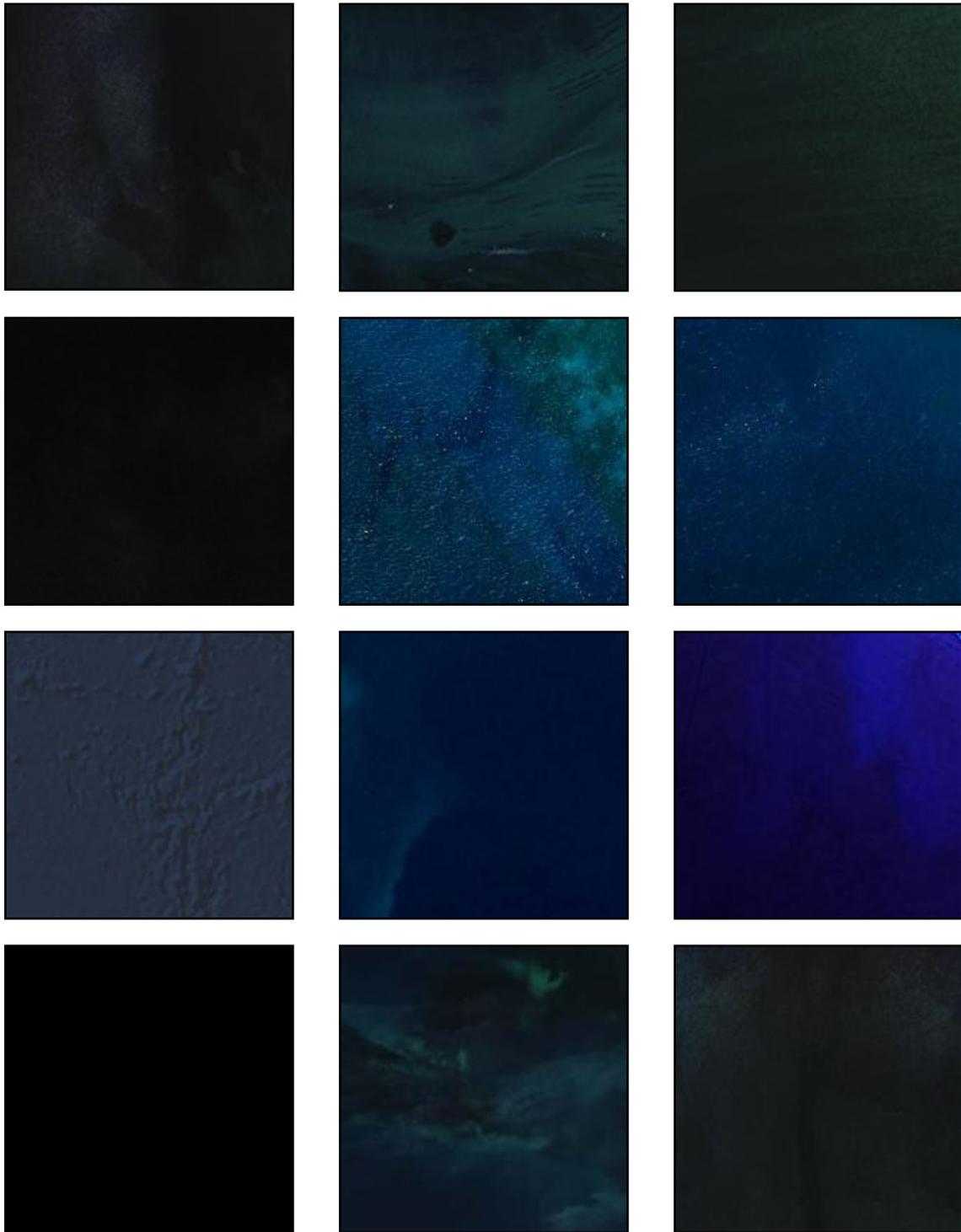


Figura 5-3. Regiones para la determinación de vectores prototipo para la temática de cuerpo de agua.

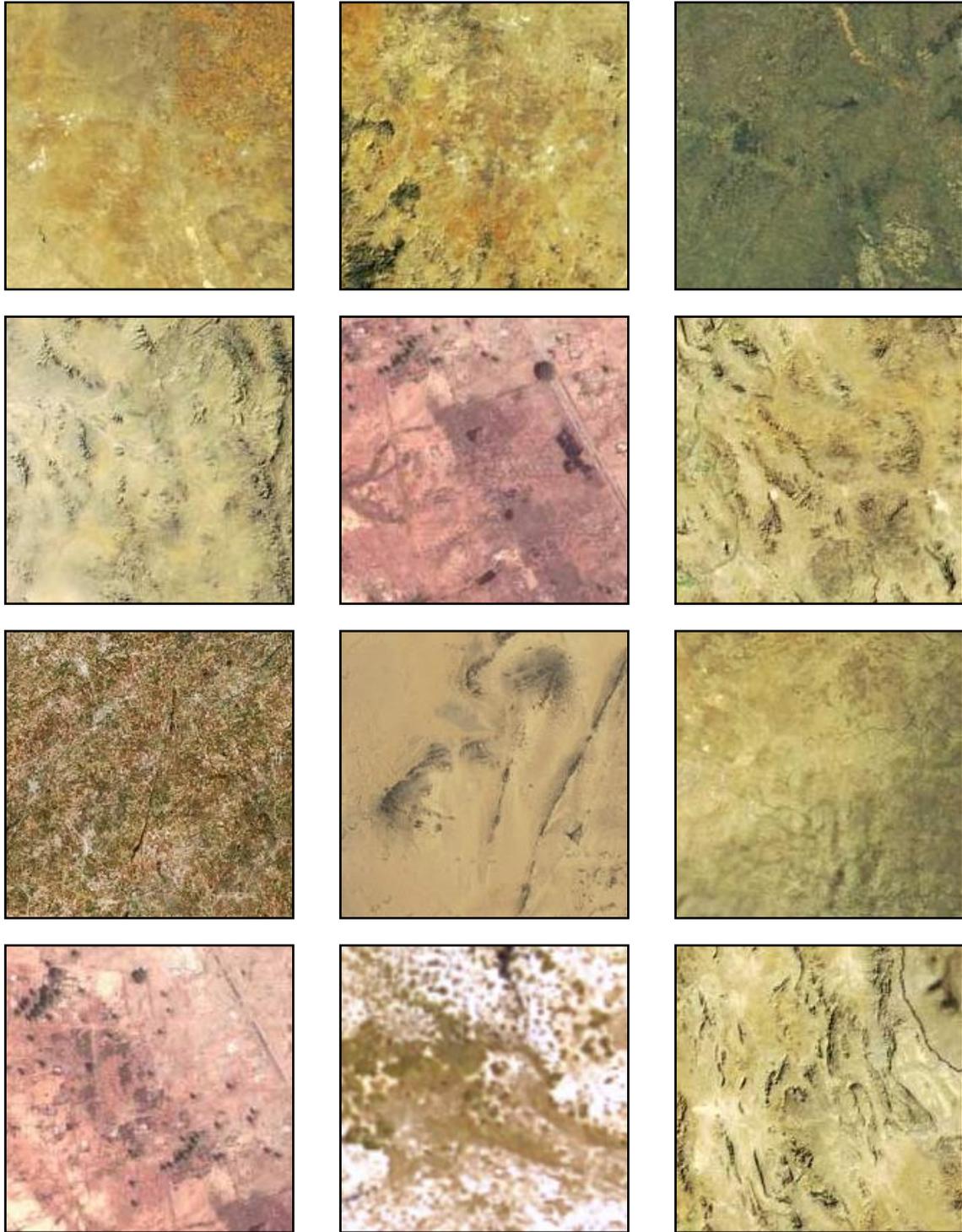


Figura 5-4. Regiones para la determinación de vectores prototipo para la temática de suelo, variante 1.



Figura 5-5. Regiones para la determinación de vectores prototipo para la temática de suelo, variante 2.

El código en MATLAB que fue empleado para determinar los vectores prototipo se ilustra en la Figura 5–6.

```
function [red, green, blue] = Stats(fileName)
    path('d:\imagenes\programas\matlab', path);
    %Leer la imagen a procesar.
    image = imread(fileName, 'bmp');
    color = [178, 178, 178];

    items = Items(image, color);
    [red, green, blue] = FindStats(image, items - 1, color);
end

function items = Items(image, color)
    [rows, cols, depth] = size(image);
    redBgnd = color(1);
    greenBgnd = color(2);
    blueBgnd = color(3);
    count = 1;

    for i = 1 : rows
        for j = 1 : cols
            red = image(i, j, 1);
            green = image(i, j, 2);
            blue = image(i, j, 3);
            if((red~=redBgnd) && (green~=greenBgnd) && (blue~=blueBgnd))
                count = count + 1;
            end
        end
    end
    items = count;
end

function [red, green, blue] = FindStats(image, items, color)
    % Obtener el numero de renglones y de columnas
    [rows, cols, depth] = size(image);
    redArray = zeros(items, 1, 'single');
    greenArray = zeros(items, 1, 'single');
    blueArray = zeros(items, 1, 'single');
    redBgnd = color(1);
    greenBgnd = color(2);
    blueBgnd = color(3);
    index = 1;

    for i = 1 : rows
        for j = 1 : cols
            red = image(i, j, 1);
            green = image(i, j, 2);
            blue = image(i, j, 3);
```

Figura 5–6. Código en MATLAB para la determinación de la media y la mediana.

```

        if((red~=redBgnd) && (green~=greenBgnd) && (blue~=blueBgnd))
            redArray(index) = red;
            greenArray(index) = green;
            blueArray(index) = blue;
            index = index + 1;
        end
    end
end

red = [mean(redArray), median(redArray), std(redArray)];
green = [mean(greenArray), median(greenArray), std(greenArray)];
blue = [mean(blueArray), median(blueArray), std(blueArray)];
end

```

Figura 5–6. (Cont...) Código en MATLAB para la determinación de la media y la mediana.

En este código la función principal *Stats* recibe como parámetro el nombre de la geoimagen sobre la que se deben calcular los valores de media, mediana y desviación estándar. Primeramente, con ayuda de la función *Items* se encuentra el número de píxeles que son diferentes de color de fondo. Con el resultado determinado, se invoca a la función *FindStats* la cual recibe como parámetros la matriz de valores de las intensidades RGB de la geoimagen, el número de elementos discretos con una intensidad distinta de fondo y el color que representa fondo. Con esta información se realiza un análisis de arriba hacia abajo y de izquierda a derecha (*top-down*), determinando primeramente si el píxel de coordenadas  $(x,y)$  es píxel de objeto; en caso de ser verdadero, lo agregan al arreglo de la componente de color correspondiente. Terminado el barrido de la imagen se procede a determinar la característica de media, mediana y desviación estándar. Los resultados que se obtuvieron para las regiones de la temática de cuerpo de agua aparecen en la Tabla 5–2.

En este caso se puede observar que los valores de media y mediana tienen resultados en un rango de 0 a 80, debido principalmente a que las componentes de color de los cuerpos de agua no presentan altas intensidades. De igual forma, los valores para las desviaciones estándar son reducidos.

Tabla 5–2. Cuantificación de la media, mediana y desviación estándar para las regiones muestra de la temática de cuerpo de agua.

Región	Componente	Media	Mediana	Desviación estándar
1	Rojo	22.37	22	04.67
	Verde	24.48	24	05.77
	Azul	27.06	25	07.87
2	Rojo	18.30	18	02.68
	Verde	37.00	37	08.52
	Azul	42.45	44	08.30
3	Rojo	21.48	21	02.36
	Verde	32.86	21	06.47
	Azul	30.41	30	05.59
4	Rojo	11.52	11	01.60
	Verde	11.07	11	02.24
	Azul	10.98	11	02.38
5	Rojo	15.48	15	03.25
	Verde	30.62	28	09.31
	Azul	40.60	40	10.20
6	Rojo	06.13	04	08.19
	Verde	49.40	47	14.09
	Azul	77.02	75	14.59
7	Rojo	02.43	01	03.74
	Verde	35.52	34	07.20
	Azul	67.69	67	09.15
8	Rojo	39.67	39	02.20
	Verde	48.25	48	03.56
	Azul	65.99	66	06.46
9	Rojo	01.19	01	00.99
	Verde	19.55	17	05.52
	Azul	52.55	51	04.86
10	Rojo	17.30	16	05.06
	Verde	09.95	09	04.32
	Azul	77.01	68	26.86
11	Rojo	00.00	00	00.00
	Verde	00.00	00	00.01
	Azul	00.00	00	00.02
12	Rojo	23.40	23	03.28
	Verde	27.99	27	04.38
	Azul	29.36	28	06.63

En lo que respecta a la temática de suelo, los resultados se enlistan en la Tabla 5–3 y en la Tabla 5–4. En contraste con los resultados obtenidos anteriormente, los centros de los vectores prototipo de esta temática tendrán un mayor valor para la media, mediana y desviación estándar.

Tabla 5-3. Cuantificación de la media, mediana y desviación estándar para las regiones muestra de la temática de suelo, variante 1.

Región	Componente	Media	Mediana	Desviación estándar
1	Rojo	186.66	188	17.97
	Verde	160.21	161	20.95
	Azul	090.93	091	18.10
2	Rojo	182.97	189	28.03
	Verde	159.68	164	26.94
	Azul	088.51	089	22.03
3	Rojo	111.72	112	15.01
	Verde	114.07	114	13.41
	Azul	078.80	079	10.98
4	Rojo	195.23	201	28.53
	Verde	182.53	189	27.49
	Azul	136.59	139	26.74
5	Rojo	194.88	198	29.06
	Verde	144.44	144	25.51
	Azul	134.72	134	21.52
6	Rojo	191.86	197	26.39
	Verde	174.58	180	26.91
	Azul	113.60	115	24.97
7	Rojo	148.09	147	32.81
	Verde	123.33	122	31.81
	Azul	084.85	083	32.84
8	Rojo	187.29	193	19.72
	Verde	161.13	165	17.16
	Azul	116.43	119	11.92
9	Rojo	173.39	178	26.95
	Verde	159.34	162	23.90
	Azul	099.08	099	17.11
10	Rojo	213.61	220	30.61
	Verde	165.88	168	30.92
	Azul	151.94	154	27.06
11	Rojo	203.63	207	35.55
	Verde	179.26	178	43.09
	Azul	153.47	150	57.91
12	Rojo	197.13	204	33.02
	Verde	182.43	189	32.80
	Azul	124.86	128	31.01

Tabla 5-4. Cuantificación de la media, mediana y desviación estándar para las regiones muestra de la temática de suelo, variante 2.

Región	Componente	Media	Mediana	Desviación estándar
1	Rojo	71.7189	072	18.0834
	Verde	80.8221	083	20.2617
	Azul	52.0636	052	16.2565
2	Rojo	80.5740	078	34.0028
	Verde	85.1783	083	34.2560
	Azul	78.2866	076	30.0657
3	Rojo	46.9373	046	21.0379
	Verde	45.3674	045	20.6976
	Azul	33.9175	033	18.1429
4	Rojo	27.0851	027	10.8840
	Verde	38.1141	038	12.2160
	Azul	27.9165	028	12.5200
5	Rojo	63.9708	062	22.7062
	Verde	67.0885	065	27.1953
	Azul	35.6940	030	19.0554
6	Rojo	92.0500	091	17.0833
	Verde	84.3672	085	15.1460
	Azul	79.6385	080	14.6894
7	Rojo	87.0243	085	22.4581
	Verde	93.7306	094	20.6574
	Azul	64.6340	063	20.4141
8	Rojo	61.0095	061	20.2767
	Verde	60.2419	060	20.4801
	Azul	48.1749	048	19.6822
9	Rojo	107.1506	110	32.3702
	Verde	116.0049	121	29.6136
	Azul	100.7814	105	27.0488
10	Rojo	134.2475	132	43.0372
	Verde	139.8713	140	39.8777
	Azul	118.2815	121	33.7737
11	Rojo	104.3223	105	17.4223
	Verde	113.4818	115	17.9852
	Azul	102.5222	104	16.4872
12	Rojo	55.4109	055	17.5921
	Verde	61.1749	061	17.9062
	Azul	43.6481	044	16.8121

Tomando en cuenta toda esta información, se determinaron los vectores prototipo para cada una de las temáticas, calculando el promedio (media) de las características cuantificadas en las componentes de color, quedando el resultado que se muestra en la Tabla 5–5.

Tabla 5–5. Vectores prototipo para las temáticas de cuerpo de agua y suelo.

Temática	Componente	Media	Mediana	Desviación estándar
1	Rojo	014.94	014.25	03.17
	Verde	027.22	025.25	05.95
	Azul	043.43	042.08	08.58
2	Rojo	182.21	186.17	26.97
	Verde	158.91	161.33	26.74
	Azul	114.48	115.00	25.18
2	Rojo	077.63	077.00	23.08
	Verde	082.12	082.50	23.02
	Azul	065.46	065.33	20.41

## 5.4 Ontología de aplicación

Cada una de las temáticas está integrada por los siguientes conceptos geográficos:

$$\{océano, mar, río, riachuelo, lago, laguna\} \in t_1$$

$$\{isla, islote, continente\} \in t_2$$

Las definiciones tomadas como base para el diseño de la ontología de aplicación aparecen en la Tabla 5–6 (Wikipedia, 2007).

Tabla 5–6. Definiciones de los objetos geográficos para la ontología de aplicación

Objeto geográfico	Definición
<i>Lago</i>	Un lago es un cuerpo de agua dulce o salada, más o menos extensa, que se encuentra alejada del mar y asociada generalmente a un origen glaciar
<i>Laguna</i>	Es la denominación que recibe cualquier extensión natural de agua estancada, sea ésta dulce o salada. La diferencia con los lagos no es muy precisa, salvo que se supone que una laguna tiene menor extensión y profundidad
<i>Río</i>	Es una corriente natural de agua que fluye con continuidad. Posee un caudal determinado y desemboca en el mar, en un lago o en otro río, en cuyo caso se denomina afluente
<i>Riachuelo</i>	Es un río corto y estrecho

Tabla 5–6. (Cont...) Definiciones de los objetos geográficos para la ontología de aplicación

Océano	Se denomina océano a la parte de la superficie terrestre ocupada por el agua marina
Mar	Es una masa de agua salada de tamaño inferior al océano
Continente	Un continente es una gran extensión de Tierra que se diferencia de otras menores o sumergidas por conceptos geográficos y culturales como océanos y etnografía
Isla	Una isla es una porción de Tierra menor que un continente que se encuentra totalmente rodeada de agua. Toda su superficie tomada a la misma altura sobre el nivel del mar está sometida a un clima similar, a diferencia de lo que ocurre en los continentes
Islote	Los islotes son pequeñas islas donde normalmente no viven seres humanos por su pequeño tamaño

Agregando tanto los objetos geoespaciales como las relaciones topológicas por medio de clases, así como los niveles de detalle, la ontología resultante sería la que se aprecia en la Figura 5–7. Es importante señalar que las clases que aparecen en color gris no describen nuevos conceptos, sino solo se agregaron para poder hacer más comprensible y completo el diagrama de la ontología.

En esta ontología de aplicación existe una relación denotada por el término de **rodeado**. Cuando se añade una nueva relación en la ontología, es necesario describirla por medio de un conjunto de operadores de álgebra de imágenes o de topología<sup>34</sup>. Se dirá que el objeto  $O_j$  es rodeado por el objeto  $O_i$  cuando se cumpla:

$$(O_i \cap O_j) \subset O_i \text{ y}$$

$$(O_i \cap O_j) - O_j \neq \emptyset$$

La primera condición estipula que la intersección de los dos objetos debe ser un subconjunto del objeto que rodea, mientras que la segunda es propiamente una restricción mediante la cual se obliga a que el resultado de la diferencia de la intersección con el objeto que rodea sea diferente del conjunto vacío. Esta restricción es necesaria porque en caso de omitirla se permitiría que un objeto estuviera rodeado y rodeara a otro objeto geográfico.

<sup>34</sup> Un ejemplo de una relación empleada en el análisis topológico es la de adyacencia, la cual tiene una diferente interpretación en el álgebra de imágenes.

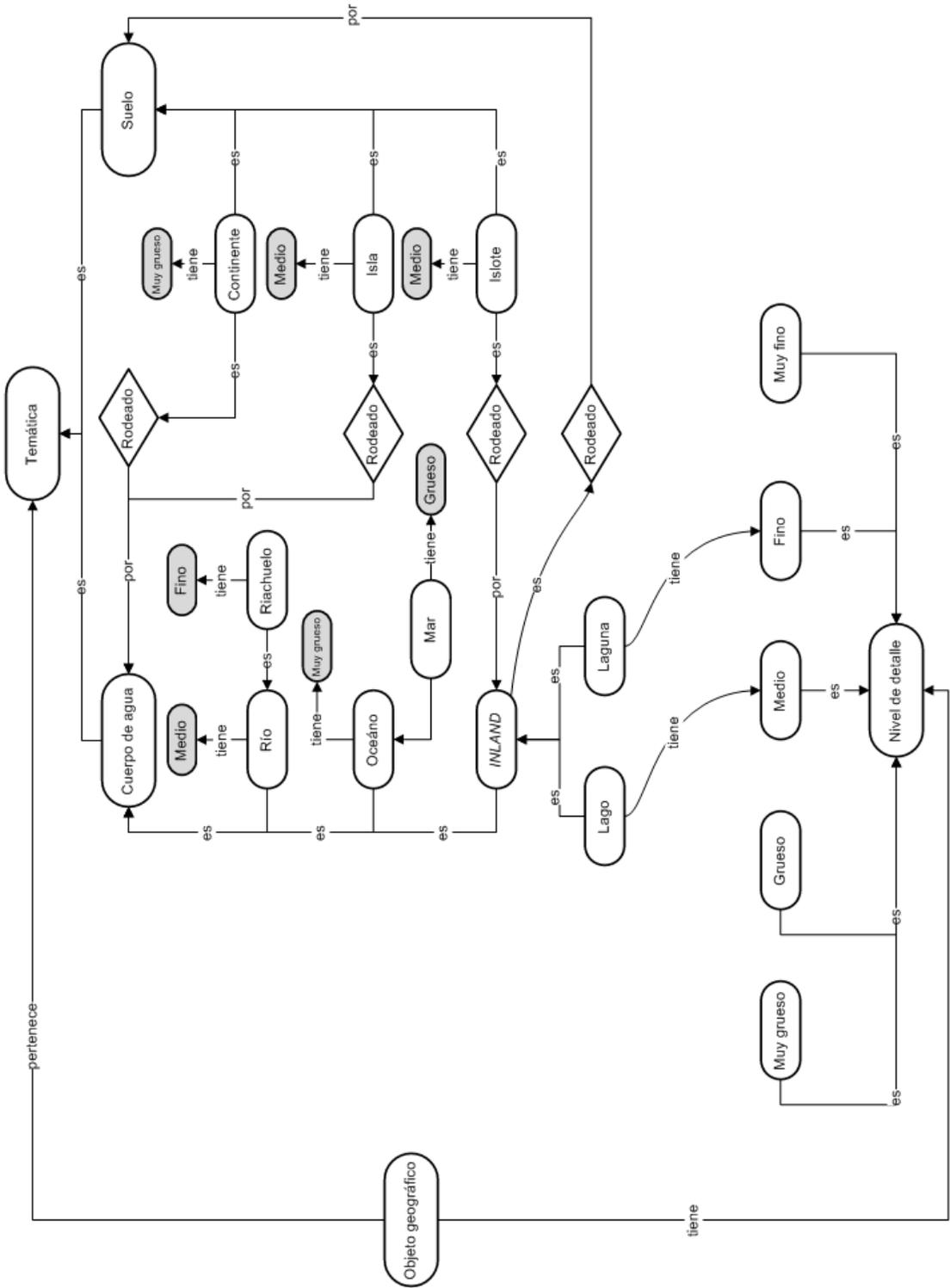


Figura 5-7. Ontología de aplicación para la especialización de objetos geográficos.

## 5.5 Casos de estudio para la descomposición semántica

En este apartado se examinarán varias geoimágenes con el objetivo de llegar a una especialización de los objetos geográficos contenidos en las imágenes, haciendo uso de la metodología de descomposición semántica propuesta. Para las fases de reconocimiento y de especialización se utilizarán, respectivamente, los vectores prototipo así como la ontología de aplicación del dominio geográfico que fueron expuestos en el apartado anterior.

El propósito del algoritmo de compresión semántica es generar, con la participación del experto, una secuencia de compresión semántica para una geoimagen dada. Dado que las imágenes que describen objetos geográficos son muy diversas, no es posible definir una cadena de compresión semántica única que se pueda aplicar a todas las geoimágenes. Con el objeto de mostrar que se encontró al menos una cadena de compresión semántica para la correcta simplificación de los objetos geoespaciales, se explicará la secuencia que permitió obtener la simplificación deseada para cada caso de estudio. Los cuatro casos analizados se diferencian significativamente en escala y el tipo de objetos geográficos, lo cual permitirá exponer la capacidad del algoritmo de compresión de adaptarse a cualquier tipo de geoimagen.

### 5.5.1 Caso de estudio 1: Geoimagen de una isla, ejemplo 1

La primera geoimagen que será procesada es una imagen digital donde se persigue extraer dos cuerpos geográficos que pertenezcan a las temáticas de cuerpo de agua y suelo. El objetivo es que la metodología de descomposición semántica realice primeramente una extracción de las regiones que describen cada uno de los objetos geoespaciales, seguido de un reconocimiento general en temáticas, y finalmente se asigne una etiqueta más específica a cada uno de los objetos geográficos. Con base en la ontología de aplicación se podrá determinar que por las propiedades geométricas y topológicas de ambos objetos, dicha geoimagen representa una isla rodeada por un océano o mar. La geoimagen original se muestra en la Figura 5–8.



Figura 5–8. Geoimagen original que representa una isla.

Para poder tener una mejor percepción de la complejidad de esta geoimagen, se puede consultar el histograma de cada componente de color en la Figura 5–9. Haciendo un análisis de los histogramas se puede observar que no existe un nivel de intensidad (al menos fácil de determinar) que permita aislar perfectamente, las dos regiones de los objetos geoespaciales.

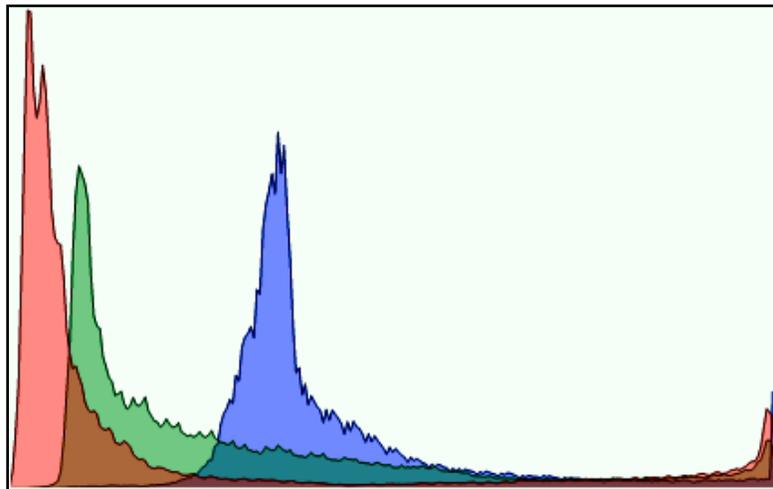


Figura 5–9. Histograma de las componentes roja, verde y azul de la geoimagen de la isla 1.

La idea general de esta metodología es tratar de hacer una simplificación del contenido visual de una geoimagen, respetando el contenido semántico de la misma. Esto es, aunque las características de bajo nivel (textura, intensidad, por citar algunas) de la geoimagen de la isla son muy variables, el obtener las regiones de cada objeto geográfico representativo y asignar una etiqueta especializada es suficiente para muchas necesidades.

### 5.5.1.1 Compresión semántica

La cadena de compresión semántica que permitió obtener las regiones de los objetos geográficos de la geoimagen original se describen en la siguiente tabla. Se realizaron cinco iteraciones del algoritmo de compresión semántica, con el mismo umbral de similitud ( $k_{CS} = 50$ ), ver Tabla 5–7. Formalmente, la secuencia de compresión semántica que se generó fue:

$$C_{CS}(I_o, I_{SS}) = \left\{ (I_o, 50), (I_1, ABS, 50), (I_2, S2f, 50), \right. \\ \left. (I_3, L, 50), (I_4, Ss, 50) \right\}$$

Tabla 5–7. Parámetros del algoritmo de compresión semántica para el caso de estudio 1.

Número de iteración	Geoimagen de entrada	Umbral de similitud
1	Imagen original	$k_{CS}=50$
2	Intensidad promedio (ABS)	$k_{CS}=50$
3	Parámetro sin $2\gamma - (S2f)$	$k_{CS}=50$
4	Característica de largo (L)	$k_{CS}=50$
5	Parámetro sin $\Delta - (Ss)$	$k_{CS}=50$

El conjunto de geoimágenes que se obtuvieron después de la primera iteración del algoritmo de compresión semántica aparece en la Figura 5–10. En este conjunto de imágenes se puede observar que solamente las geoimágenes que se obtienen con las características de intensidad e intensidad promedio tienen una correspondencia con el contenido de la geoimagen original, el resto de características y parámetros no muestra una simplificación acorde con las regiones de los dos objetos geográficos (tierra y agua). Todavía no es posible detectar la geometría y topología de los objetos geoespaciales porque la información aún es heterogénea. Cuando sucede esto, es recomendable seguir aplicando una simplificación tomando como base alguna de las geoimágenes obtenidas con característica de color (intensidad o intensidad promedio). Por esta razón, se tomó la geoimagen de intensidad promedio como entrada para la segunda iteración del algoritmo. Usando el mismo umbral de similitud, después de terminar este segundo ciclo de compresión semántica se obtuvieron las geoimágenes que aparecen en la Figura 5–11.

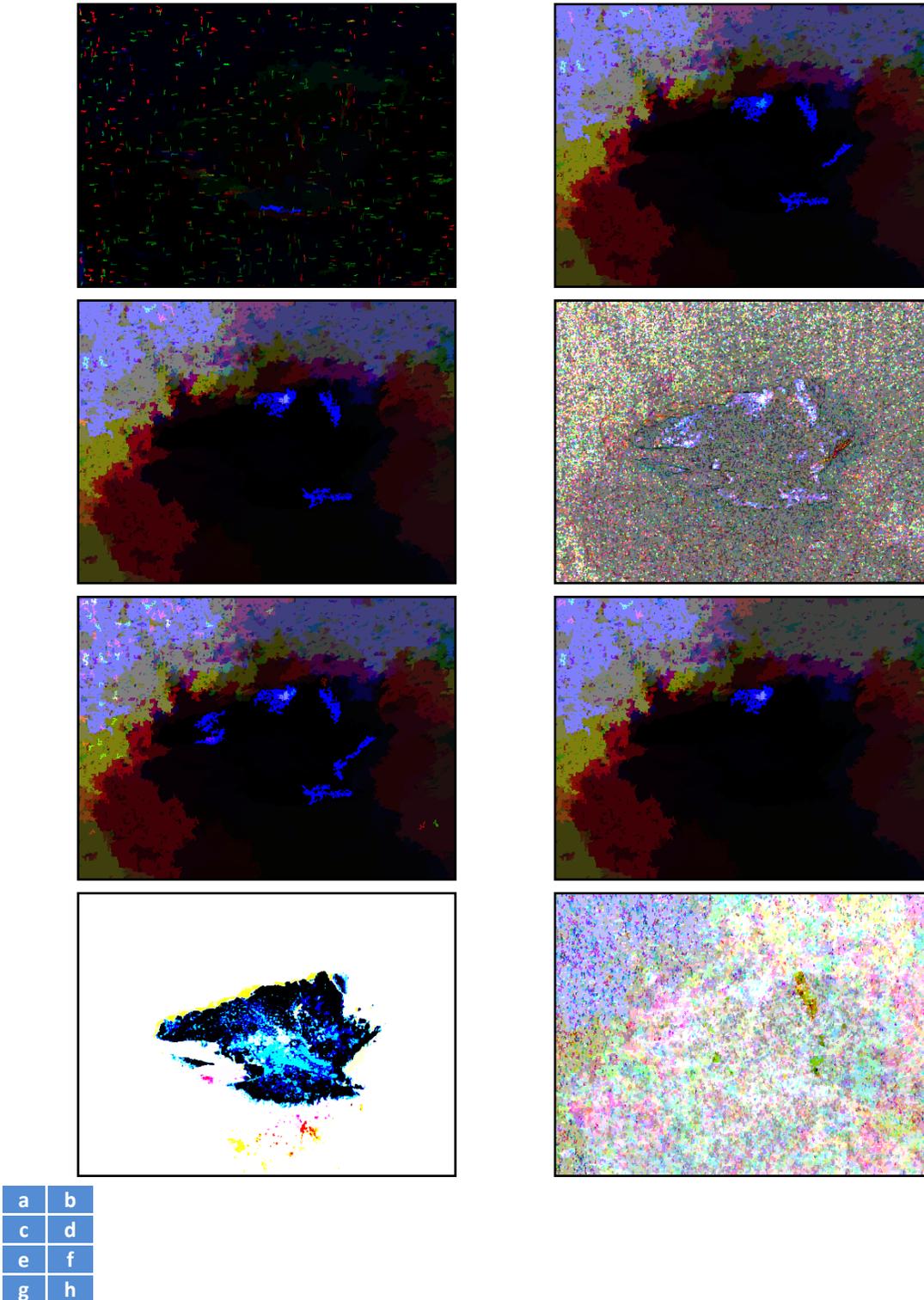
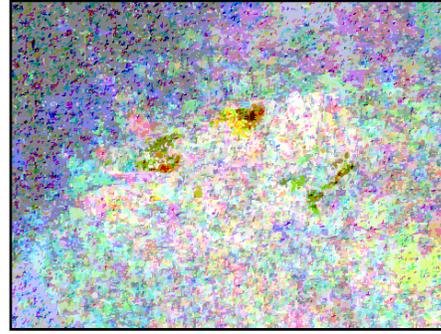
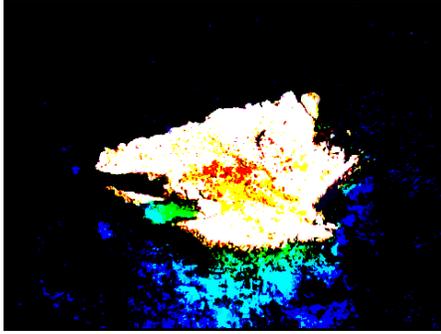
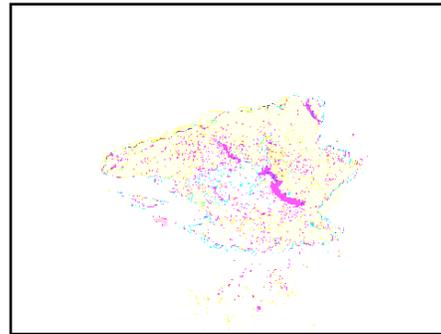
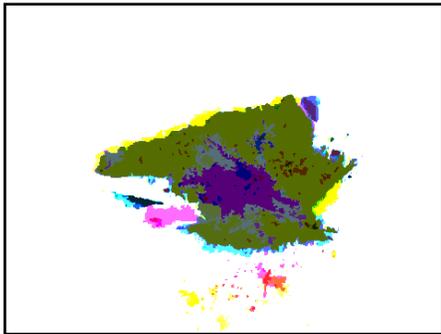
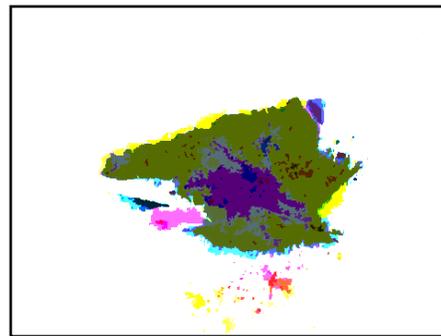
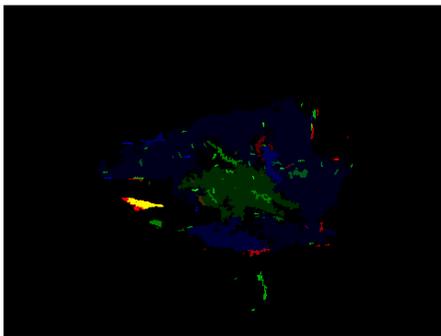


Figura 5–10. Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



i j

Figura 5–10. (Cont...) Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



a b  
c d

Figura 5–11. Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad promedio de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

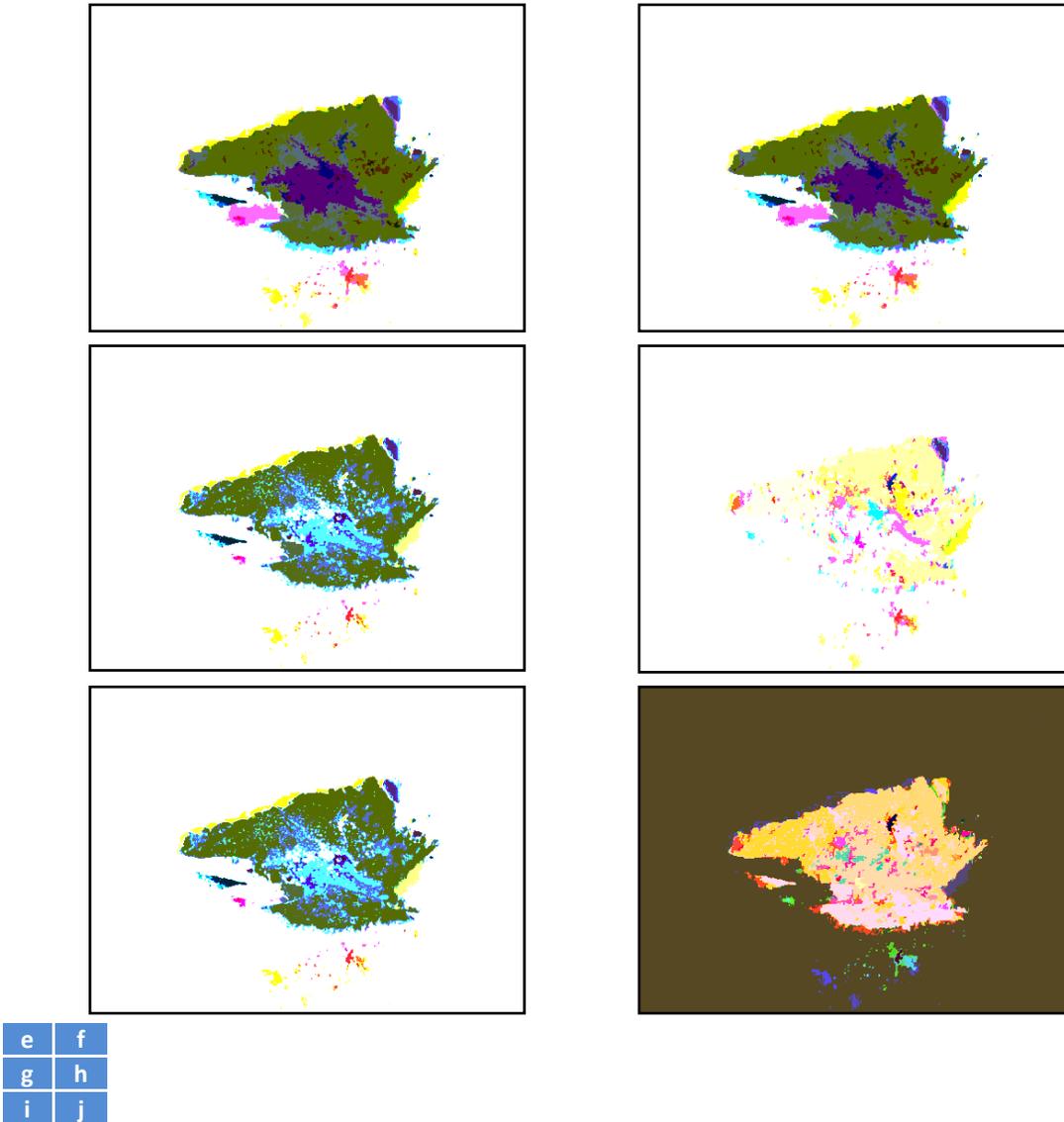


Figura 5–11. (Cont...) Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad promedio de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica l; (j) Parámetro S2f.

En este grupo de imágenes (Figura 5–11) ya comienza a detectarse una geometría y topología entre los segmentos de los objetos geográficos, esto se puede observar en los resultados de las geoimágenes simplificadas con las características de largo, área, tamaño, ancho, así como con el parámetro  $S2f$ . Usando la geoimagen del parámetro  $S2f$  se realizó una tercera iteración obteniendo las geoimágenes que se muestran en la Figura 5–12.

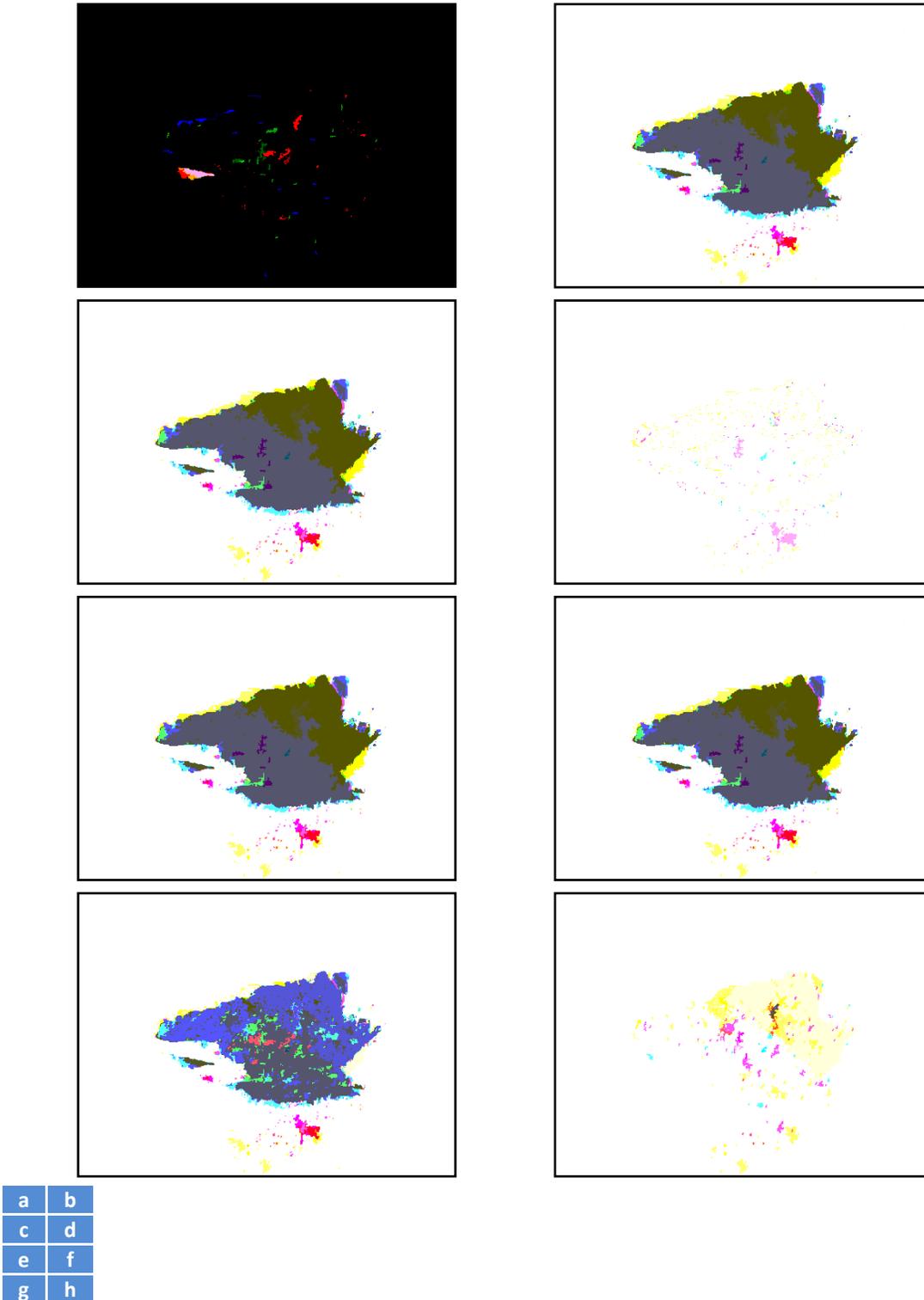


Figura 5–12. Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen del parámetro S2f de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

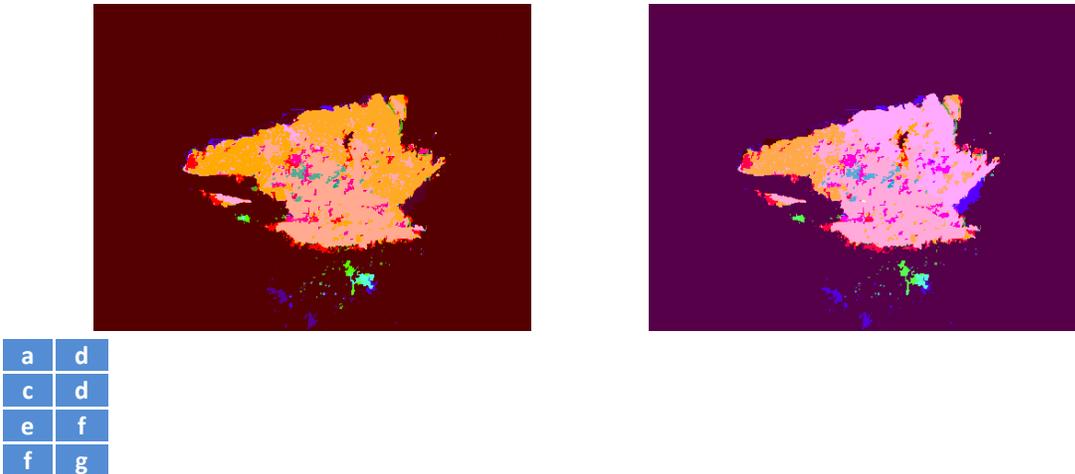
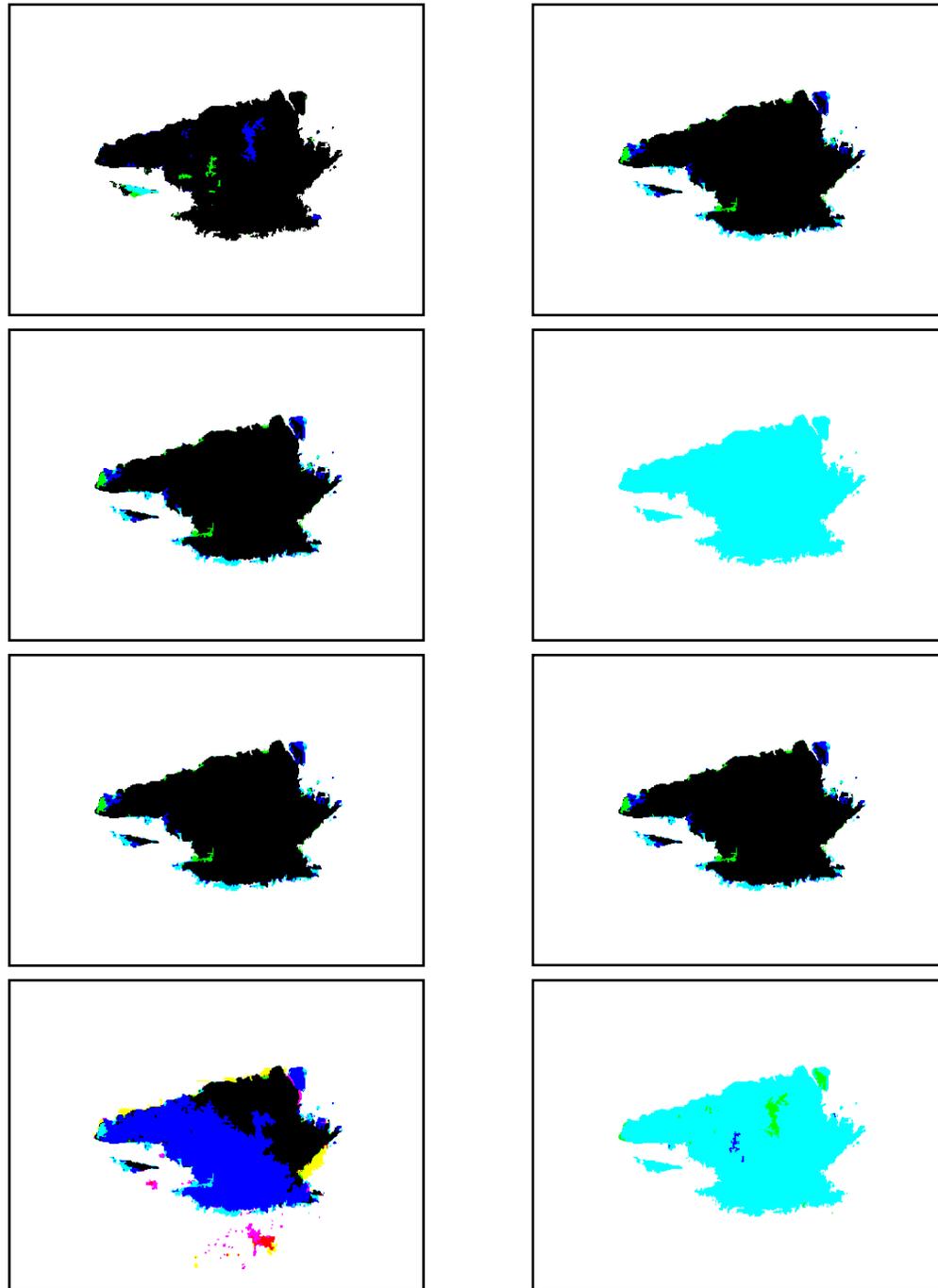


Figura 5–12. (Cont...) Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen del parámetro S2f de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Terminada la tercera iteración (como se puede apreciar en la Figura 5–12) las geoimágenes de las características de largo, área, tamaño y ancho son muy similares entre sí en comparación con el resto de las geoimágenes; además de que la geometría de los segmentos de cada objeto geográfico que se desea extraer está más diferenciada entre sí (los segmentos de menor tamaño en su mayoría están dentro de la región de suelo). Con este tipo de resultados, se puede optar por cualquiera de las características o parámetros que están diferenciando los objetos geográficos y los resultados serán prácticamente los mismos. Para este caso de estudio, sería alguna de las geoimágenes simplificada con los atributos que ya fueron mencionadas: largo, área, tamaño o ancho. Como geoimagen de entrada para la cuarta iteración se eligió la geoimagen simplificada con la característica de largo.

Los resultados aparecen en la Figura 5–13. Para este grupo de geoimágenes, la que es simplificada usando el parámetro  $\sin \Delta$  es la que tiene menos regiones (cada intensidad de color uniforme representa una región). Aplicando una quinta y última iteración del algoritmo de compresión semántica sobre esta geoimagen se obtiene lo mostrado en la Figura 5–14. Cabe recordar que en todas las iteraciones se usó un umbral de similitud igual a 50.



a	b
c	d
e	f
g	h

Figura 5–13. Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de la característica de largo de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

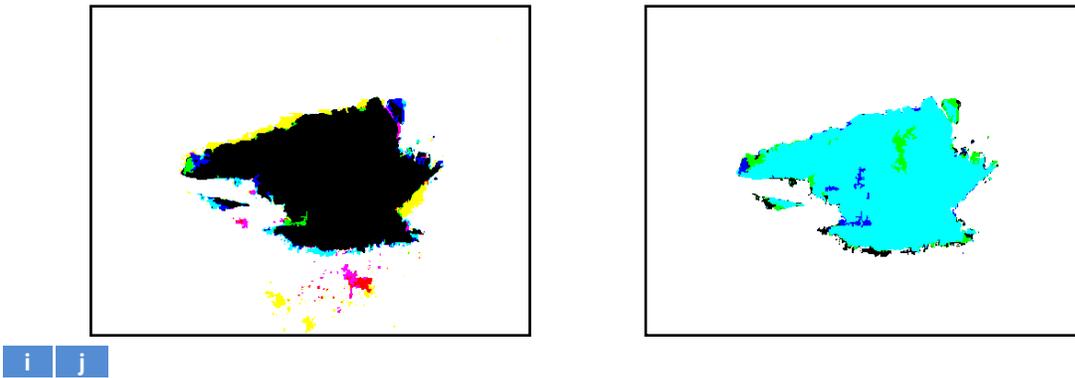


Figura 5–13. (Cont...) Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de la característica de largo de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

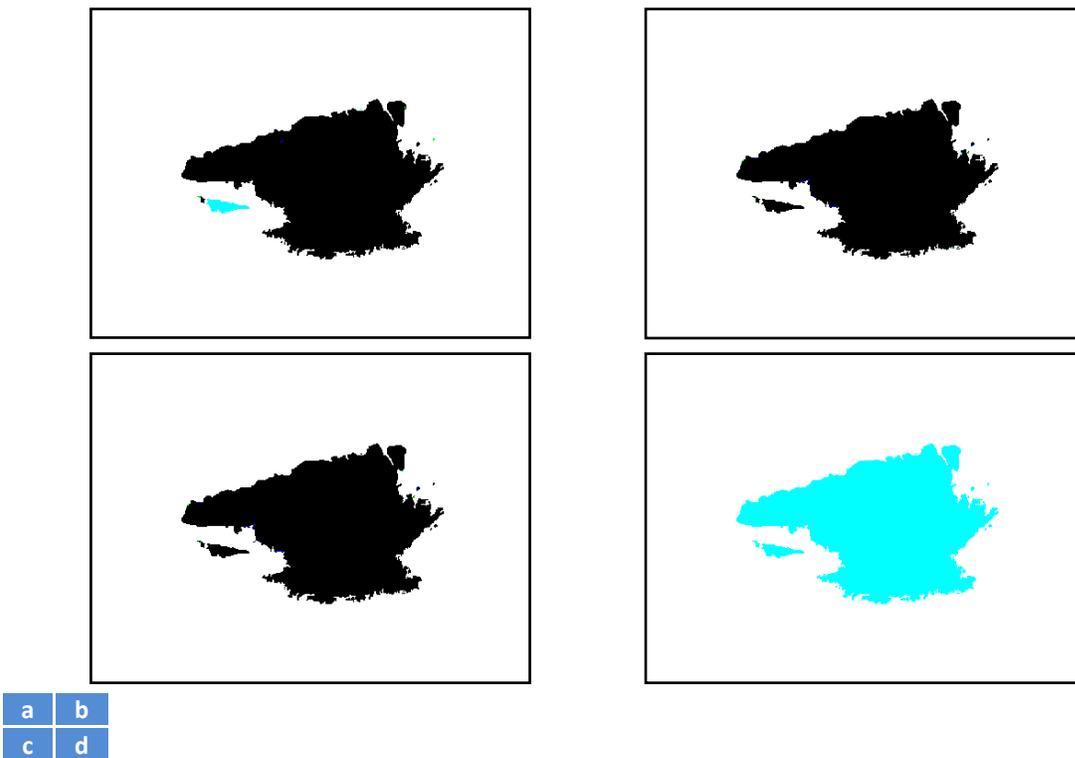


Figura 5–14. Conjunto de geoimágenes que se obtienen después de la quinta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen del parámetro Ss de la iteración 4; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

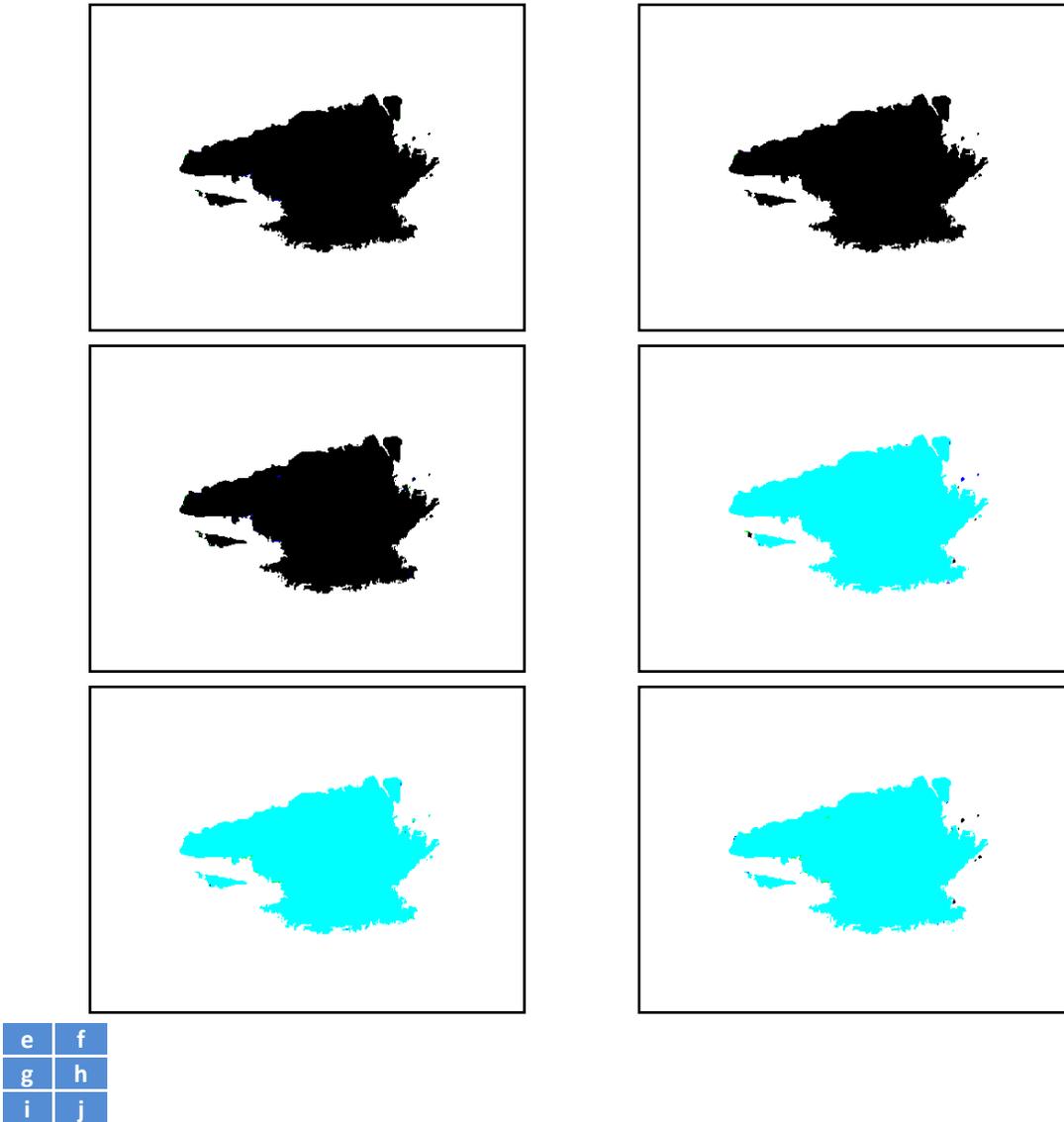
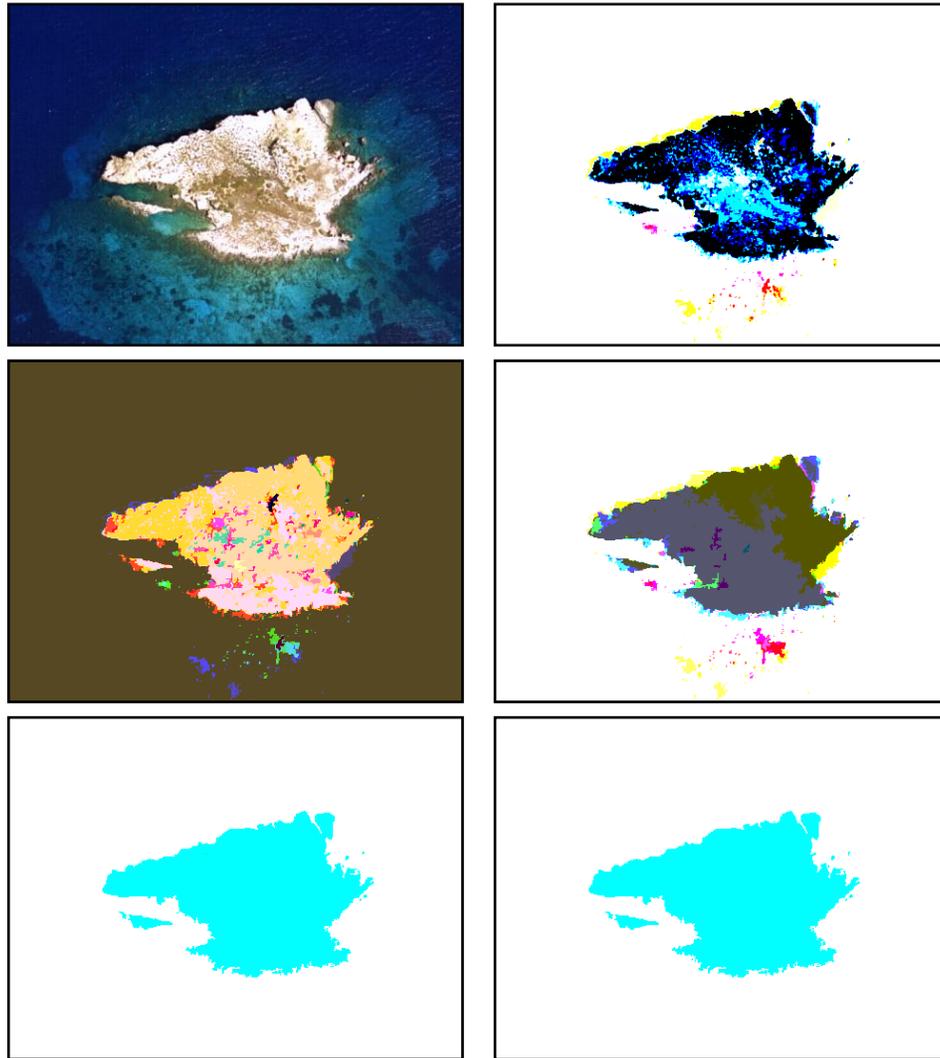


Figura 5–14. (Cont...) Conjunto de geoimágenes que se obtienen después de la quinta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen del parámetro  $S_s$  de la iteración 4; (a) Característica  $d$ ; (b) Característica  $L$ ; (c) Característica  $S$ ; (d) Parámetro  $S_s$ ; (e) Característica  $S_z$ ; (f) Característica  $W$ ; (g) Característica  $ABS$ ; (h) Parámetro  $C2f$ ; (i) Característica  $I$ ; (j) Parámetro  $S2f$ .

Se compararon las geoimágenes del parámetro  $S_s$  obtenidas en la cuarta y quinta iteración, y no se detectó una fusión adicional de segmentos; lo que implica que la geoimagen es invariante, es decir, no es posible simplificarla más. Adicionalmente, en las imágenes simplificadas con el resto de características y parámetros hubo más fusión de segmentos, principalmente en la zona ubicada entre los límites de la isla con el cuerpo de agua. Para la etapa de síntesis se usará como entrada la geoimagen del parámetro  $S_s$ .

En resumen, la geoimagen original se simplificó de la forma como se describe en la Figura 5–15.



a	b
c	d
e	f

Figura 5–15. Simplificación de la geoimagen del caso de estudio 1; (a) Geoimagen original; (b) Geoimagen de intensidad promedio (iteración 1); (c) Geoimagen del parámetro S2f (iteración 2); (d) Geoimagen de la característica L (iteración 3); (e) Geoimagen del parámetro Ss (iteración 4); (f) Geoimagen del parámetro Ss (iteración 5).

Es importante mencionar que pueden existir diferentes secuencias de compresión semántica y llegar al mismo resultado (desde el punto de vista semántico). Por ejemplo, usando la cadena:

$$C_{cs}(I_o, I_{Ss}) = \{(I_o, 50), (I_1, ABS, 50), (I_2, S, 50)\}$$

Se obtuvieron las simplificaciones que se resumen en la siguiente figura.

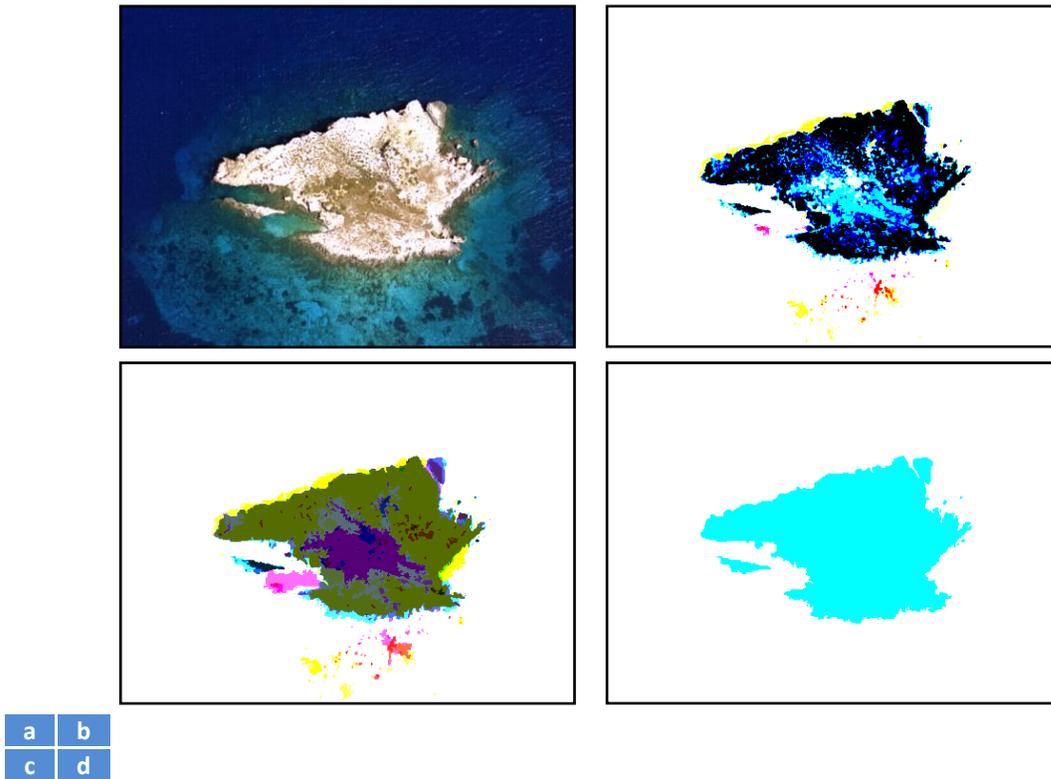


Figura 5–16. Simplificación de la geoimagen del caso de estudio 1 con otra cadena de compresión semántica; (a) Geoimagen original; (b) Geoimagen de intensidad promedio (iteración 1); (c) Geoimagen de la característica de área (iteración 2); (d) Geoimagen del parámetro Ss (iteración 3).

Esta secuencia de compresión semántica requiere menos iteraciones para llegar al resultado buscado<sup>35</sup>. En consecuencia, generalmente se busca la primera secuencia de compresión semántica que simplifica una geoimagen dada. A partir de ésta secuencia, se pueden emplear otras características o parámetros de compresión y umbrales de similitud, para determinar una cadena más reducida, como se hizo en este ejemplo.

Retomando lo obtenido con la primera secuencia de compresión semántica, la imagen de la Figura 5–15(f) fue usada como entrada para la siguiente etapa de la metodología de descomposición semántica (síntesis).

<sup>35</sup> En realidad, la primera cadena de compresión semántica solamente requiere de cuatro iteraciones, pero se realizó la quinta iteración para mostrar que la geoimagen del parámetro Ss era invariante.

### 5.5.1.2 Extracción de regiones

El primer paso en la etapa de síntesis consiste en extraer las regiones de la geoimagen original mediante la función de mapeo. El resultado se muestra en la Figura 5–17; el color gris claro representa el fondo de la imagen.

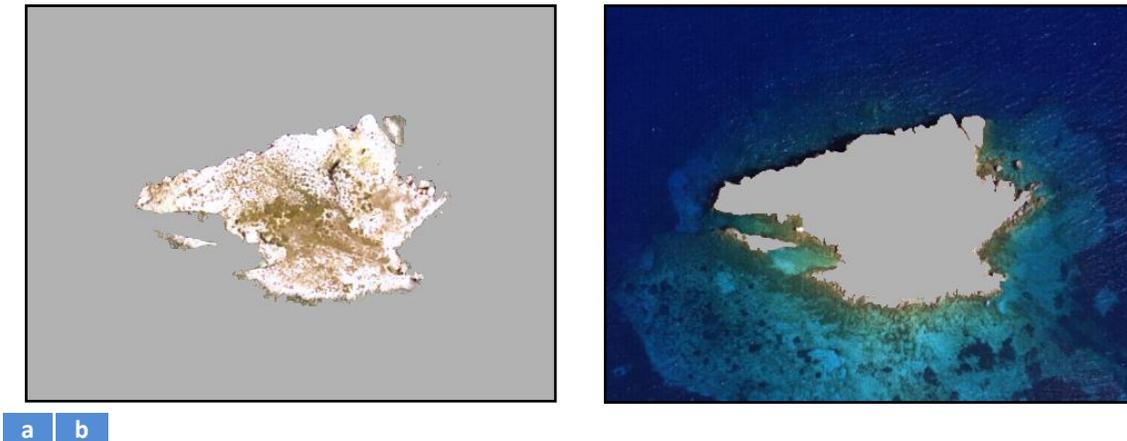


Figura 5–17. Extracción de las regiones de los objetos geográficos (a) Geoimagen de la clase 1; (b) Geoimagen de la clase 2.

### 5.5.1.3 Reconocimiento

La geoimagen de la Figura 5–17(a) contiene todos los píxeles que fueron descritos con la misma intensidad de color homogénea según el algoritmo de compresión semántica; en consecuencia, todos los elementos discretos que tengan un color distinto al color de fondo, corresponden a elementos de la misma temática, visualmente se puede observar que es toda la región de tierra de la isla. Para el caso de la Figura 5–17(b), esta geoimagen describe otra temática, que es la de cuerpos de agua.

En este punto es necesario asignar una etiqueta general indicando para cada región la etiqueta de si corresponde a  $t_1$  (cuerpo de agua) o a  $t_2$  (suelo). Primeramente, se determinó la información cuantitativa de la media, mediana y desviación estándar de las geoimágenes de cada temática, ver Tabla 5–8.

Tabla 5–8. Resultados para cada una de las regiones del caso de estudio 1.

Geoimagen	Componente	Media	Mediana	Desviación estándar
1	Rojo	217.1619	230	036.6150
	Verde	199.3082	209	045.5054
	Azul	184.0881	193	059.7738
2	Rojo	020.8962	013	022.8430
	Verde	054.4544	042	034.8318
	Azul	095.2111	090	022.2705

Donde:

Geoimagen 1 – Imagen que contiene los pixeles de la clase 1, ver Figura 5–17(a).

Geoimagen 2 – Imagen que contiene los pixeles de la clase 2, ver Figura 5–17(b).

Se puede interpretar la terna formada por la media, mediana y desviación estándar, como un punto en un espacio tridimensional (*media* → *x*, *mediana* → *y*, *desviación estándar* → *z*). Ahora, es necesario calcular la distancia Euclidiana de estos vectores con la componente de color correspondiente de los vectores prototipo que modelan las temáticas de suelo y cuerpo de agua, obteniéndose lo que aparece en la Tabla 5–9.

Tabla 5–9. Distancias entre los vectores a reconocer y los vectores prototipos de las temáticas.

Geoimagen	Componente	Vectores prototipos		
		Temática 1	Temática 2a	Temática 2b
1	Rojo	297.5909	056.8835	207.5123
	Verde	254.8392	065.2425	173.8992
	Azul	212.5614	110.1182	178.6666
2	Rojo	020.5928	236.7003	085.5265
	Verde	043.0863	158.7958	050.4495
	Azul	071.8682	031.6979	038.6936

Como las tres componentes de color de la geoimagen 1 están más cerca de la temática 2a, entonces la geoimagen 1 corresponde a objetos geoespaciales de la temática de suelo, mientras que en los valores calculados para la geoimagen 2, las componentes de color están más cerca de la temática 1, en consecuencia, son objetos de la temática de cuerpo de agua, es decir:

$$T_{I_0} = \{suelo, cuerpo\ de\ agua\}$$

5.5.1.4 Especialización

Hasta este punto, se tiene conocimiento de que la temática 1 es suelo, mientras que la temática 2 es un cuerpo de agua, lo cual es verdadero. Esto es un resultado válido del algoritmo de descomposición semántica. Para tratar de obtener una especialización de los objetos geográficos de cada temática, es necesario consultar la ontología de la Figura 5–7 y buscar las relaciones topológicas o geométricas, con el objeto de obtener una etiqueta especializada para los objetos geográficos.

Considerando únicamente el objeto geográfico tipo suelo de mayor tamaño y el otro que representa el cuerpo de agua, quedarían dos objetos geográficos. Denotando como  $O_1$  al objeto geoespacial tipo suelo y  $O_2$  al objeto cuerpo de agua, se tiene la especialización que se muestra en la Tabla 5–10.

Tabla 5–10. Especialización de los objetos geográficos del caso de estudio 1.

Relación	Conclusión	Especialización
$O_1$ es rodeado por $O_2$	Verdadero	$O_1$ es una isla, islote o continente $O_2$ es un <i>INLAND</i> , océano o mar
$O_2$ es rodeado por $O_1$	Falso	Ninguna

Como se cumple la relación  $O_1$  es rodeado por  $O_2$  entonces existen diferentes opciones de especialización para cada objeto geoespacial.  $O_1$  no puede ser un islote porque si se revisa la ontología de aplicación, se requiere que el cuerpo de agua que rodea al islote sea de tipo *INLAND* (objeto que está rodeado por otro objeto tipo suelo), pero esto no se cumple porque no hay otro objeto geográfico con el que el cuerpo de agua pudiera relacionarse, o lo que es lo mismo, rodee al cuerpo de agua. En consecuencia, solo resta diferenciar entre un objeto tipo isla y continente; para lo cual la ontología de aplicación establece que el nivel de detalle de estos dos objetos geográficos es: muy grueso para un continente, mientras que una isla tiene un nivel de detalle medio. Un experto puede concluir que para esta geoimagen el nivel de detalle es medio, por lo que  $O_1$  es un objeto tipo isla.

En lo que respecta al segundo objeto geográfico, ya se determinó que no es de tipo *INLAND*, por lo que quedan las opciones de océano y mar; sin embargo, no existe una coincidencia en el nivel de

detalle de un experto con los dos objetos geográficos; por lo tanto,  $O_2$  es un objeto geográfico tipo océano o mar.

La temática de suelo contiene una gran cantidad de objetos geográficos, para los cuales se cumple que están rodeados por el cuerpo de agua. Por lo anterior, a todos estos objetos de la temática de suelo, se les asignaría una etiqueta especializada igual a la del objeto geográfico suelo más grande que es una isla. Apegándonos a la definición de una isla (región de suelo rodeada por un cuerpo de agua), todas las pequeñas regiones tipo suelo satisfacen la definición; sin embargo, como hay un objeto tipo isla mucho más grande, todos estos objetos geoespaciales de menor tamaño, forman parte de la isla de mayor tamaño y no son islas propiamente, si no, rocas o roquetas que pertenecen a una isla. La especialización que se puede obtener de la geoimagen depende de las necesidades del experto; un caso sería que para él todos los objetos geográficos isla muy pequeños forman parte de la isla más grande, o bien, en la ontología de aplicación se puede asociar un nuevo concepto, tal como roca sobre el mar.

Este conocimiento no está dentro de la ontología de aplicación; pero lo importante de la frase anterior es que los objetos geográficos se pueden especializar con base en sus propiedades geométricas. De la misma forma que los seres humanos analizamos un problema de esta índole, ya no se está manejando información de color, ni un patrón, firma, etc., como se realiza en una técnica orientada a píxeles. En consecuencia, estos objetos tipo suelo pueden ser etiquetados como otros. Con esto, la especialización de los objetos geográficos es:

$$S_{I_0} = \{isla, océano o mar, otros\}$$

#### 5.5.1.5 Descomposición semántica iterativa

Cuando se requiere obtener mayor nivel de descomposición en una geoimagen, se puede aplicar la metodología de descomposición semántica en forma iterativa. Retomando nuevamente el caso de estudio de la isla, se puede aplicar el algoritmo de compresión semántica sobre cada geoimagen de las regiones extraídas: la de cuerpo de agua y la de suelo.

Después de aplicar el algoritmo de compresión semántica sobre la región de cuerpo de agua, con la cadena de compresión semántica  $C_{cs}(I_o, I_{intensidad}) = \{(I_o, 25), (I_1, intensidad, 2.5)\}$  se obtuvo la geomimagen simplificada que se muestra en la siguiente figura.

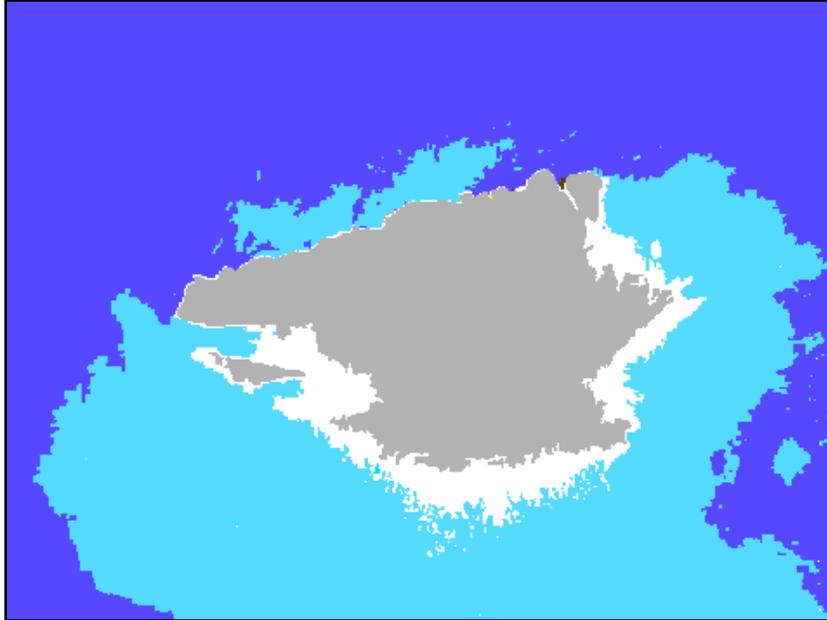


Figura 5–18. Resultado de la compresión semántica sobre la región extraída que representa un cuerpo de agua.

Comparando visualmente, la geomimagen de la Figura 5–18 con la versión original de la isla, Figura 5–8, se puede observar que las intensidades de color blanco representan la parte límite entre la porción de la isla con el cuerpo de agua, mientras que la intensidad de color cian describe la parte de la isla que está a baja profundidad, o mejor dicho, el arrecife bajo el agua que se forma alrededor de la isla. Finalmente, la parte de color azul se corresponde con la porción más profunda del cuerpo de agua.

Con estas nuevas regiones se podría hacer una asignación de etiquetas que proporcione mayor **particularización** de la identidad de los objetos geográficos que se obtuvieron del cuerpo de agua. Primeramente, se tiene que definir un algoritmo de reconocimiento que permita asignar etiquetas generales a cada uno de estos objetos geoespaciales, seguido del algoritmo de especialización, que personalice cada una de las etiquetas generales, esto se propone como trabajo futuro. No obstante, se puede adelantar que una forma de resolver esto sería: la región que está descrita como color blanco es la que está **más cerca** de la isla y puede ser especializada como el límite

entre la isla y el cuerpo de agua y la parte del arrecife está **entre** la parte baja y la parte profunda del cuerpo de agua. Es decir, la forma de asignar las nuevas etiquetas, corresponde con la forma en que los individuos realizamos un análisis de una imagen. No estamos buscando únicamente un patrón, sino que usamos un conocimiento que se basa en relaciones geométricas y topológicas para poder especializar los objetos geográficos. Aplicando la cadena de compresión semántica  $C_{CS}(I_o, I_{ABS}) = \{(I_o, 18), (I_1, ABS, 10), (I_2, intensidad, 10)\}$  en la región extraída de la temática de suelo, se obtuvo la simplificación que aparece en la Figura 5–19.

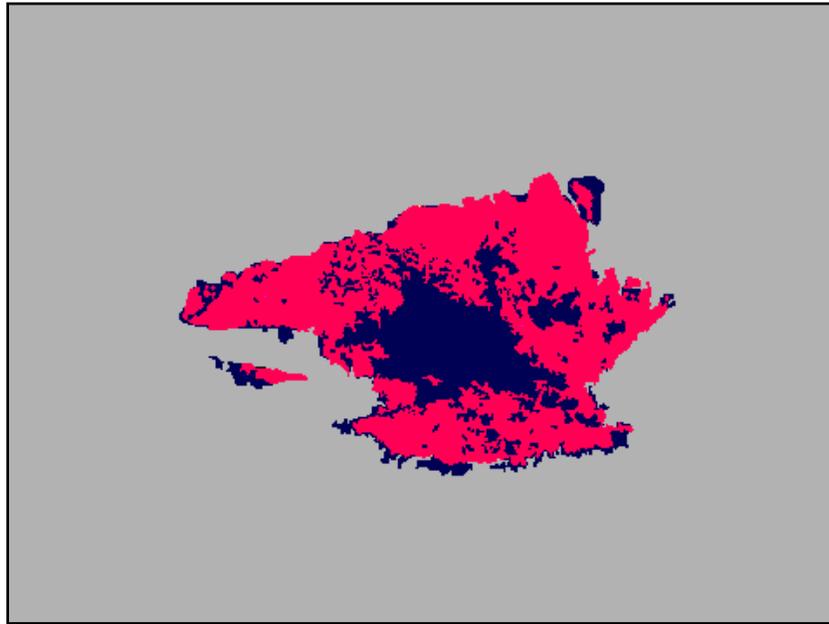


Figura 5–19. Resultado de la compresión semántica sobre la región extraída que representa suelo.

### 5.5.2 Caso de estudio 2: Geoimagen de una isla, ejemplo 2

Se realizó la prueba con otra geoimagen que describe el mismo accidente geográfico del caso de estudio anterior, pero en esta geoimagen, se tiene mayor complejidad en las características de los objetos geográficos (Figura 5–20). El histograma de cada componente de color de esta geoimagen se puede consultar en la Figura 5–21. El objetivo fue obtener dos objetos geográficos, una isla y un cuerpo de agua tipo océano o mar.

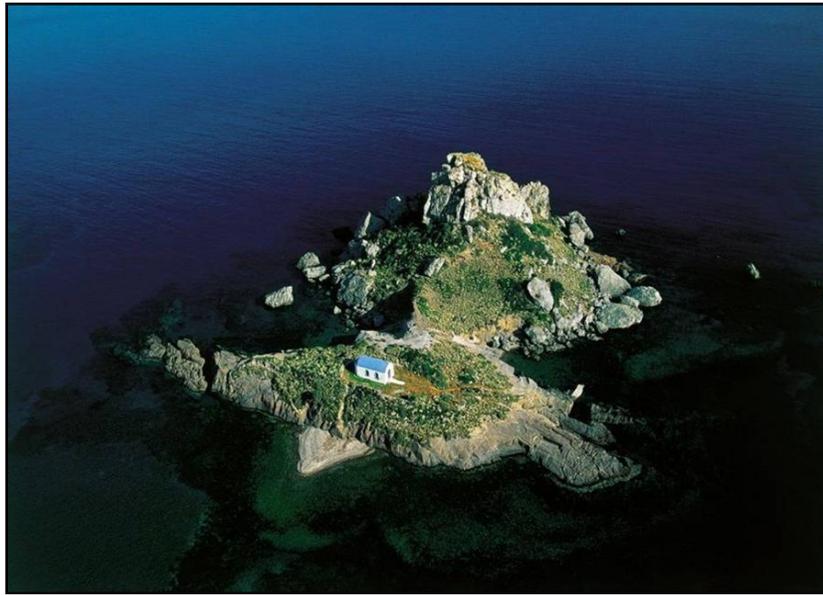


Figura 5–20. Geoimagen de otro accidente geográfico tipo isla.

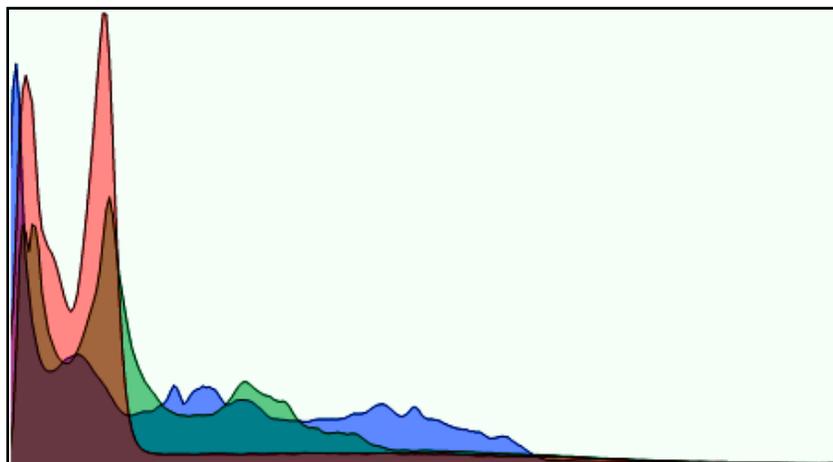


Figura 5–21. Histogramas de las componentes roja, verde y azul del accidente geográfico tipo isla.

### 5.5.2.1 Compresión semántica

En la siguiente tabla aparecen los parámetros que proporcionaron los resultados más adecuados.

Tabla 5–11. Parámetros para el algoritmo de compresión semántica para el caso de estudio 2.

Número de iteración	Geoimagen de entrada	Umbral de similitud
1	Imagen original	$k_{cs}=21.5$
2	Intensidad (I)	$k_{cs}=22.5$
3	Área (S)	$k_{cs}=22.5$
4	Área (S)	$k_{cs}=22.5$

Con esta información la cadena de compresión semántica sería:

$$C_{cs}(I_o, I_{ss}) = \{(I_o, 21.5), (I_1, intensidad, 22.5), (I_2, S, 22.5), (I_3, S, 22.5)\}$$

Después de realizar la primera iteración del algoritmo de compresión semántica se obtuvieron las geoimágenes que aparecen en la Figura 5–22.

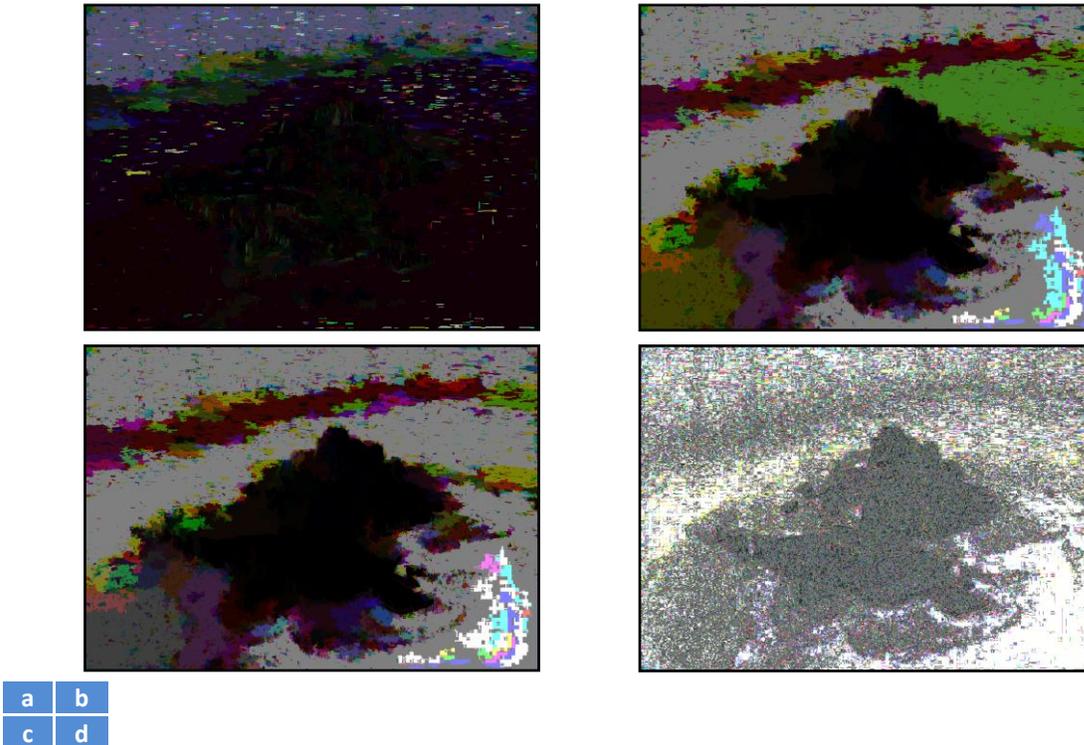
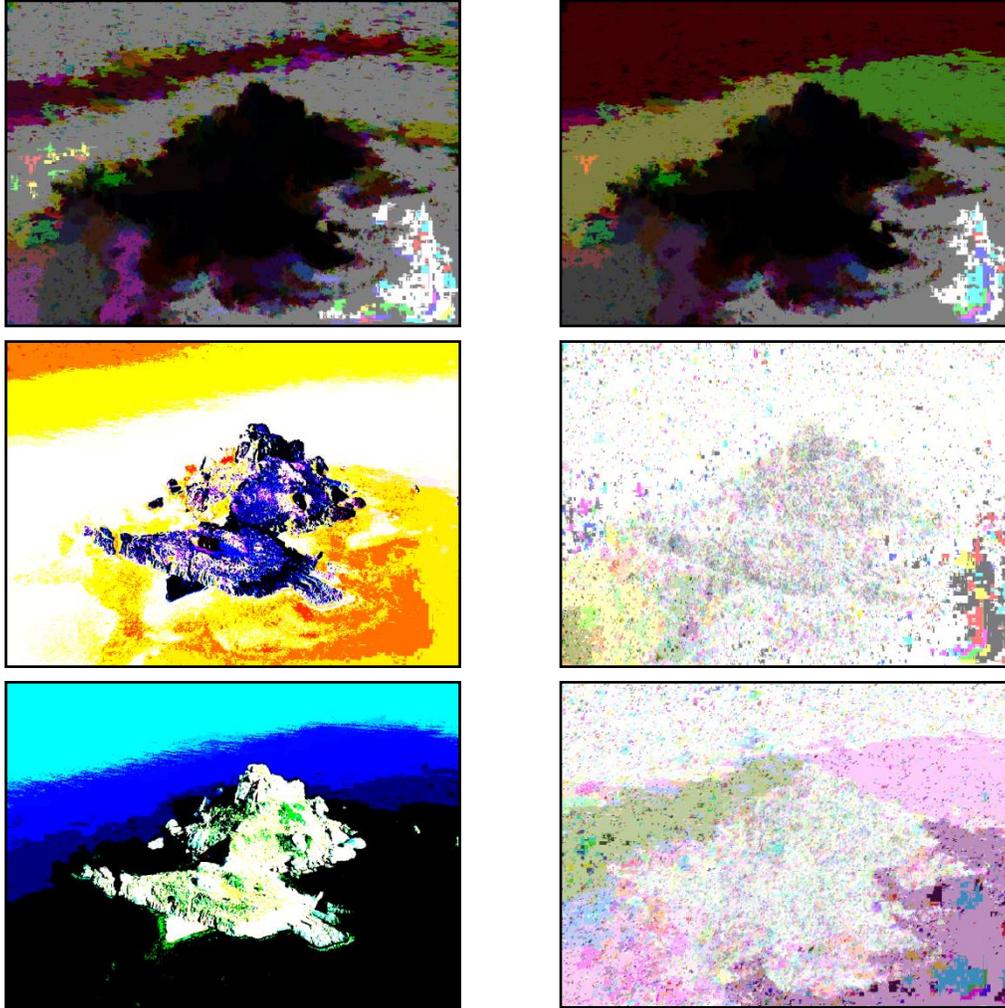


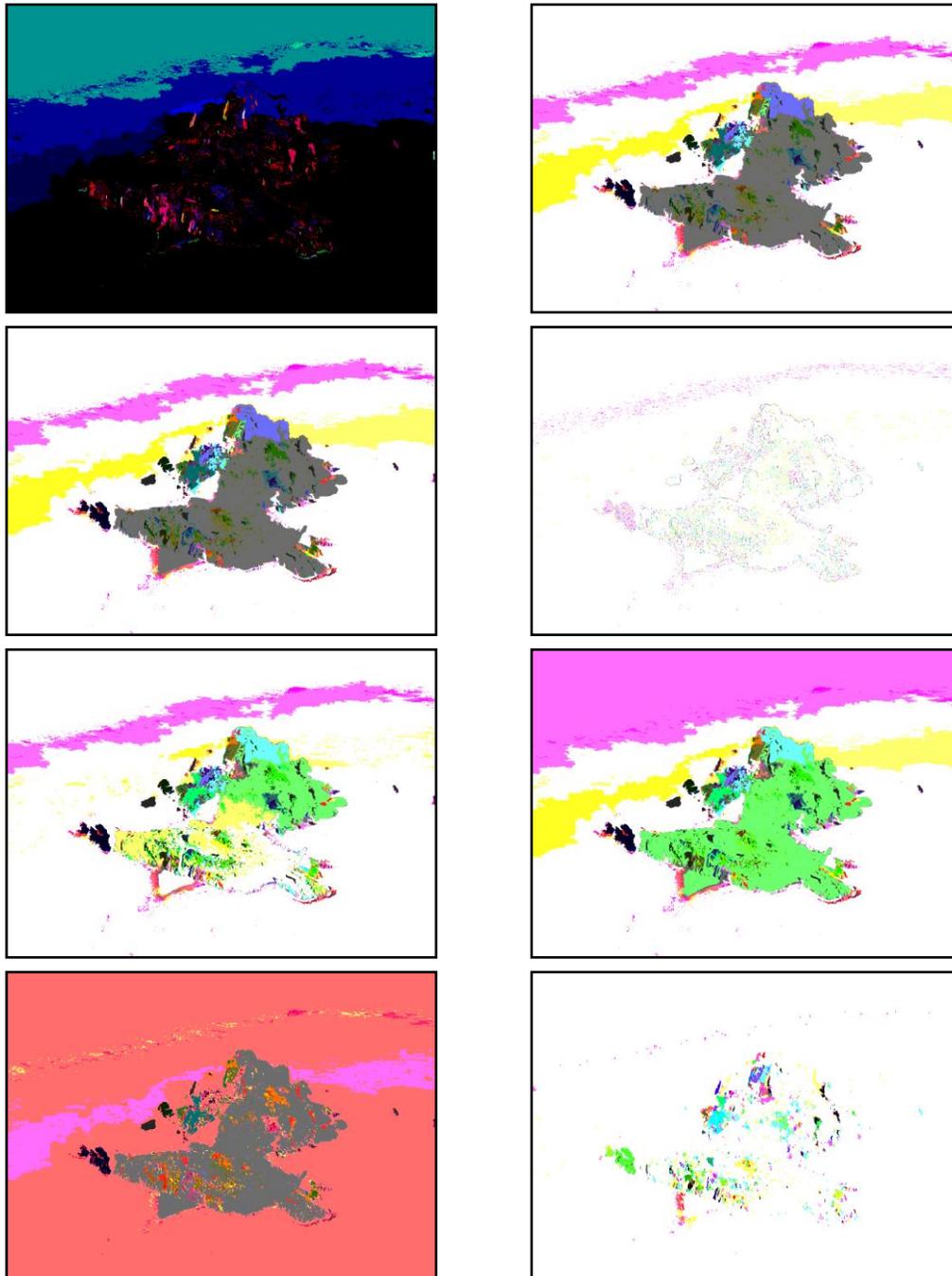
Figura 5–22. Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica I; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



e	f
g	h
i	j

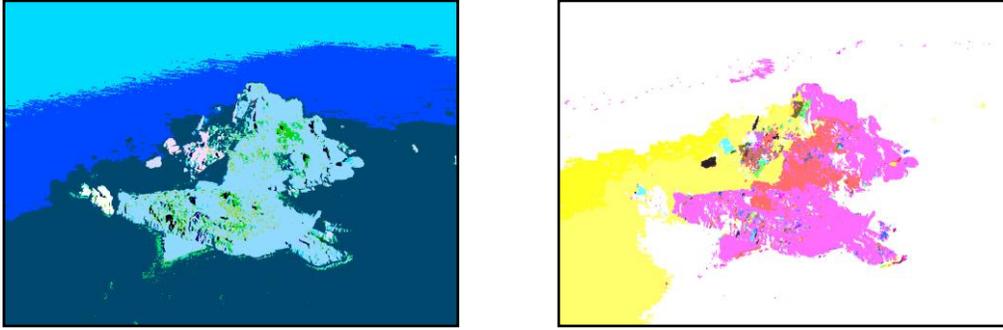
Figura 5–22. (Cont...) Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

En esta geoimagen, los resultados de la simplificación tienen un comportamiento como el explicado en el caso de estudio 1 (la isla anterior), solamente las simplificaciones por color son adecuadas a este nivel de iteración. Por ello, para la siguiente iteración del algoritmo se usó la geoimagen de intensidad. La Figura 5–23 muestra las geoimágenes que se obtuvieron al terminar la segunda iteración, como umbral de similitud se usó un valor un poco diferente al empleado en la iteración 1. La razón de esto es que después de realizar las pruebas con varios umbrales, éste valor permitió llegar a la simplificación buscada.



a	b
c	d
e	f
g	h

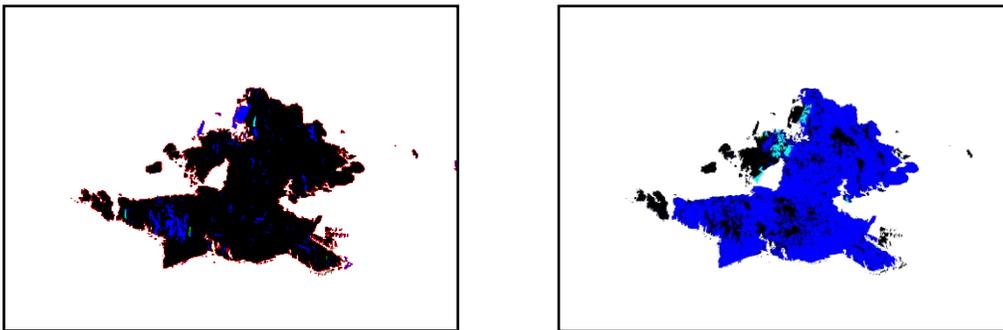
Figura 5–23. Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



i j

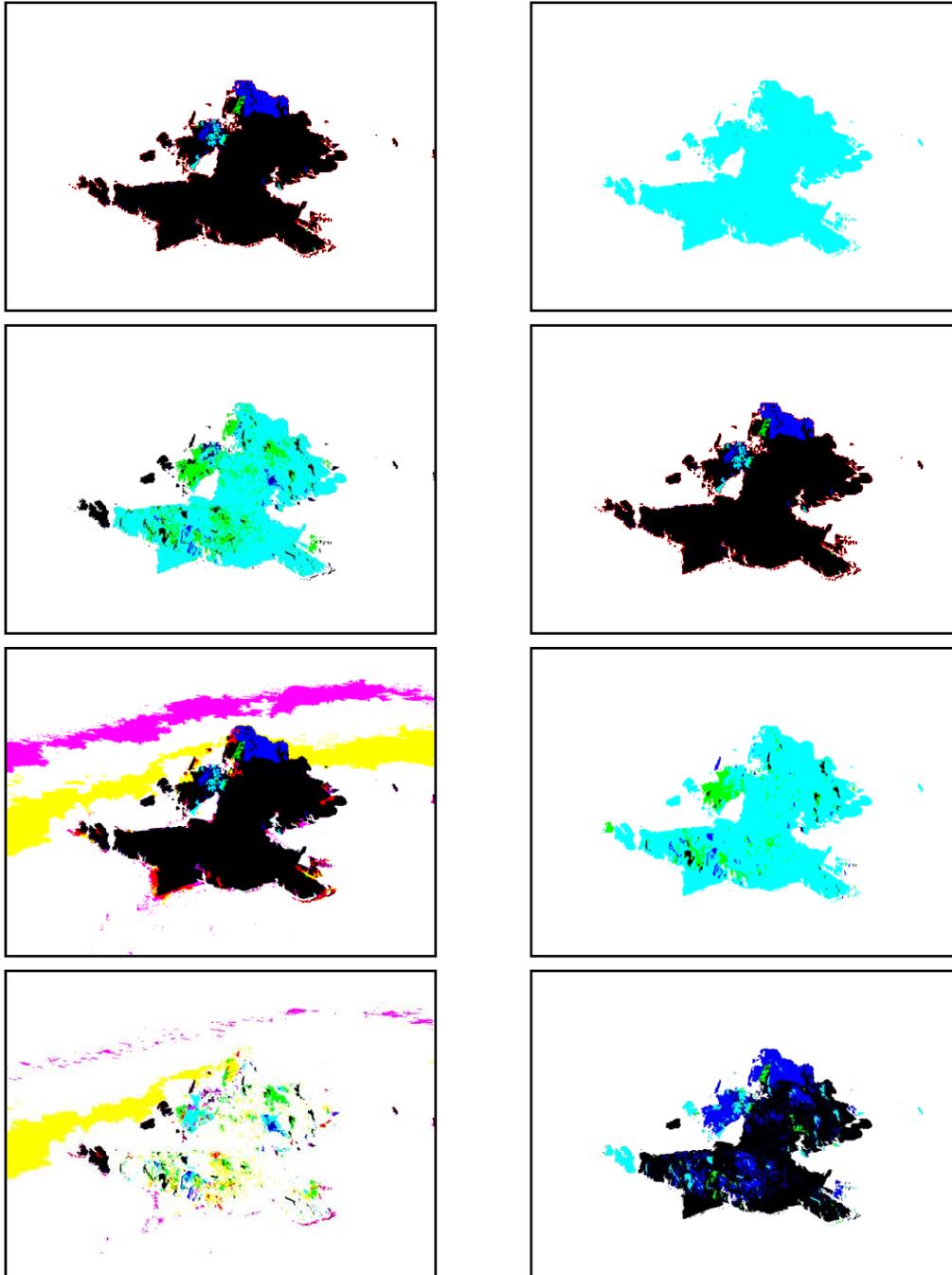
Figura 5–23. (Cont...) Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Las simplificaciones obtenidas con las características de largo y área son más aptas para la siguiente iteración, porque la cantidad de segmentos es menor y éstos tienen más correspondencia con los objetos geográficos a extraer. La Figura 5–27 contiene las geoimágenes simplificadas después de la tercera iteración del algoritmo, usando un umbral de similitud igual a 22.5.



a b

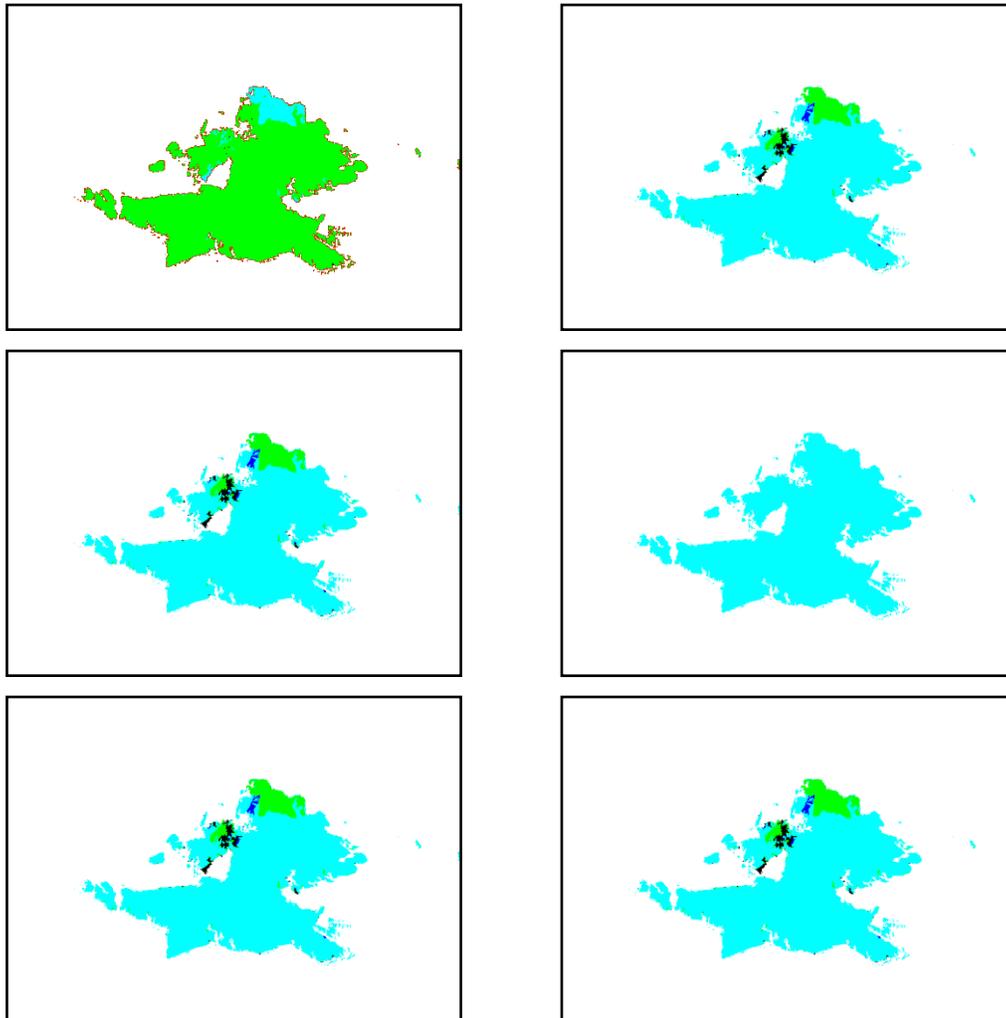
Figura 5–24. Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de área de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



c	d
e	f
g	h
i	j

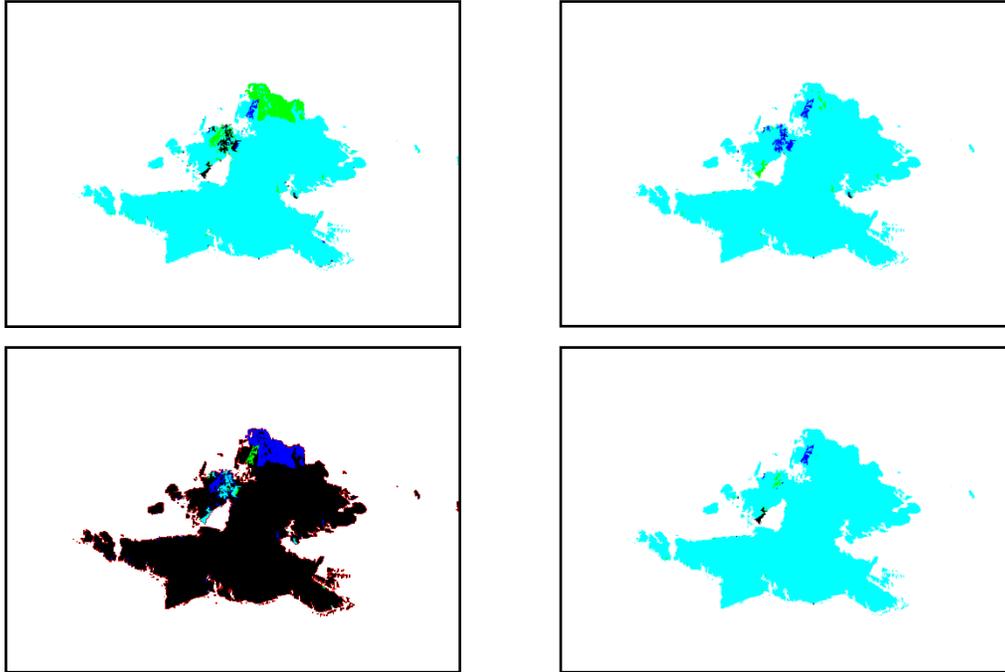
Figura 5–24. (Cont...) Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de área de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

El cuerpo de agua ya está descrito con una sola intensidad de color uniforme, en lo que respecta a la zona de la suelo, todavía se requiere aplicar una nueva iteración de la compresión semántica; para lo cual se usó la geoimagen simplificada con la característica de área. Es importante observar que la geoimagen simplificada con el parámetro  $S_s$  contiene menos segmentos. También se hizo la prueba con esta geoimagen, pero se obtuvo mejor simplificación con la otra característica propuesta.



a	b
c	d
e	f

Figura 5–25. Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de área de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro  $S_s$ ; (e) Característica  $S_z$ ; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



g	h
i	j

Figura 5–25. (Cont...) Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de área de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

### 5.5.2.2 Extracción de regiones

Usando la geoimagen del parámetro  $S_s$  se realizó la extracción de regiones, obteniendo las dos geoimágenes que aparecen a continuación. La Figura 5–26 contiene la intensidad de color original de los píxeles que fueron descritos con una de las intensidades de color homogéneas en la geoimagen obtenida con el algoritmo de compresión semántica; mientras que la Figura 5–27 representa la información para la segunda intensidad de color.



Figura 5–26. Geoimagen de los píxeles que fueron agrupados con la primera intensidad de color en la geoimagen del parámetro  $S_s$ .



Figura 5–27. Geoimagen de los píxeles que fueron agrupados con la segunda intensidad de color en la geoimagen del parámetro  $S_s$ .

### 5.5.2.3 Reconocimiento

Se determinó la información de la media, mediana y desviación estándar para cada geoimagen generada con el algoritmo de extracción de regiones. Los resultados de estas variables están contenidos en la Tabla 5–12.

Tabla 5–12. Resultados para cada una de las regiones del caso de estudio 3.

Geoimagen	Componente	Media	Mediana	Desviación estándar
1	Roja	018.4674	019	010.9967
	Verde	036.7330	031	027.3805
	Azul	052.1167	039	047.0074
2	Roja	107.7964	106	055.6347
	Verde	119.7726	119	048.9239
	Azul	092.3005	088	048.2364

Donde:

Geoimagen 1 – Imagen que contiene la información original de los píxeles con la intensidad de color 1, ver Figura 5–26.

Geoimagen 2 – Imagen que contiene la información original de los píxeles con la intensidad de color 2, ver Figura 5–27.

Con respecto a las distancias entre las temáticas y los vectores prototipo, se obtuvieron los resultados que aparecen en la Tabla 5–13.

Tabla 5–13. Distancias entre los vectores a reconocer y los vectores prototipos de las temáticas.

Geoimagen	Componente	Vectores prototipos		
		Temática 1	Temática 2a	Temática 2b
1	Rojo	009.8113	234.5477	083.7271
	Verde	012.4510	188.2363	078.9734
	Azul	024.7787	115.3531	049.1236
2	Rojo	140.6873	113.0764	053.0172
	Verde	132.4957	073.6753	055.0789
	Azul	093.2107	038.7352	060.1269

El conjunto de etiquetas generales fue igual a  $T_{I_0} = \{\text{cuerpo de agua, suelo (área verde)}\}$

### 5.5.2.4 Especialización

En la temática de cuerpo de agua, solamente hay un solo objeto geográfico y suponiendo un nivel de detalle de la geoimagen igual a medio, tal como se hizo con la otra geoimagen de la isla, se tendría que el cuerpo geoespacial es de tipo océano o mar. Asimismo, se puede considerar

únicamente el objeto tipo suelo de mayor tamaño. Entonces, quedarían las siguientes tres etiquetas especializadas:

$$S_{I_0} = \{océano\ o\ mar, isla, otros\}$$

### 5.5.3 Caso de estudio 3: Geoimagen de los Grandes Lagos

El siguiente caso de estudio corresponde a la descomposición semántica de una geoimagen de los Grandes Lagos, la cual se puede observar en la Figura 5–28.



Figura 5–28. Geoimagen de los Grandes Lagos.

Los histogramas de cada componente de color de esta geoimagen se muestran en la Figura 5–29.

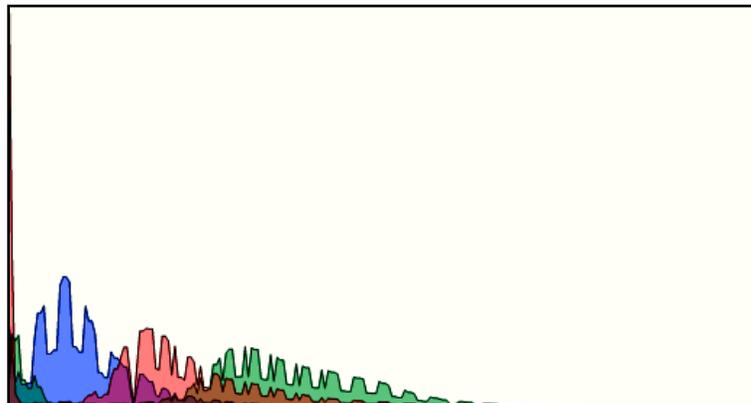


Figura 5–29. Histogramas de las componentes de color roja, verde y azul de la geoimagen de la República Mexicana.

### 5.5.3.1 Compresión semántica

La cadena de compresión semántica que permitió simplificar esta geoimagen fue:

$$C_{CS}(I_0, I_{intensidad}) = \left\{ \begin{array}{l} (I_0, 20), (I_1, intensidad, 20), (I_2, intensidad, 20), \\ (I_3, intensidad, 20), (I_4, intensidad, 20), \\ (I_5, intensidad, 05) \end{array} \right\}$$

En la Tabla 5–14 se desglosa la secuencia de compresión semántica.

Tabla 5–14. Parámetros del algoritmo de compresión semántica para el caso de estudio 3.

Número de iteración	Geoimagen de entrada	Umbral de similitud
1	Imagen original	$k_{CS}=20$
2	Intensidad	$k_{CS}=20$
3	Intensidad	$k_{CS}=20$
4	Intensidad	$k_{CS}=20$
5	Intensidad	$k_{CS}=05$

Después de aplicar la primera iteración del algoritmo de compresión semántica, se obtuvieron las geoimágenes que aparecen en la Figura 5–30.

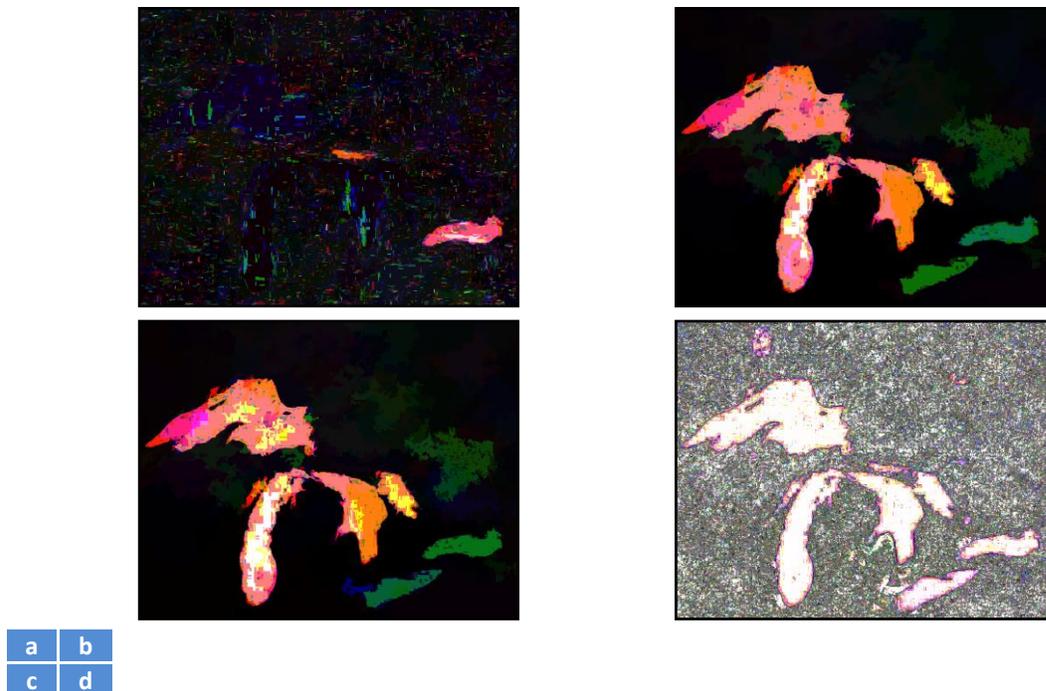
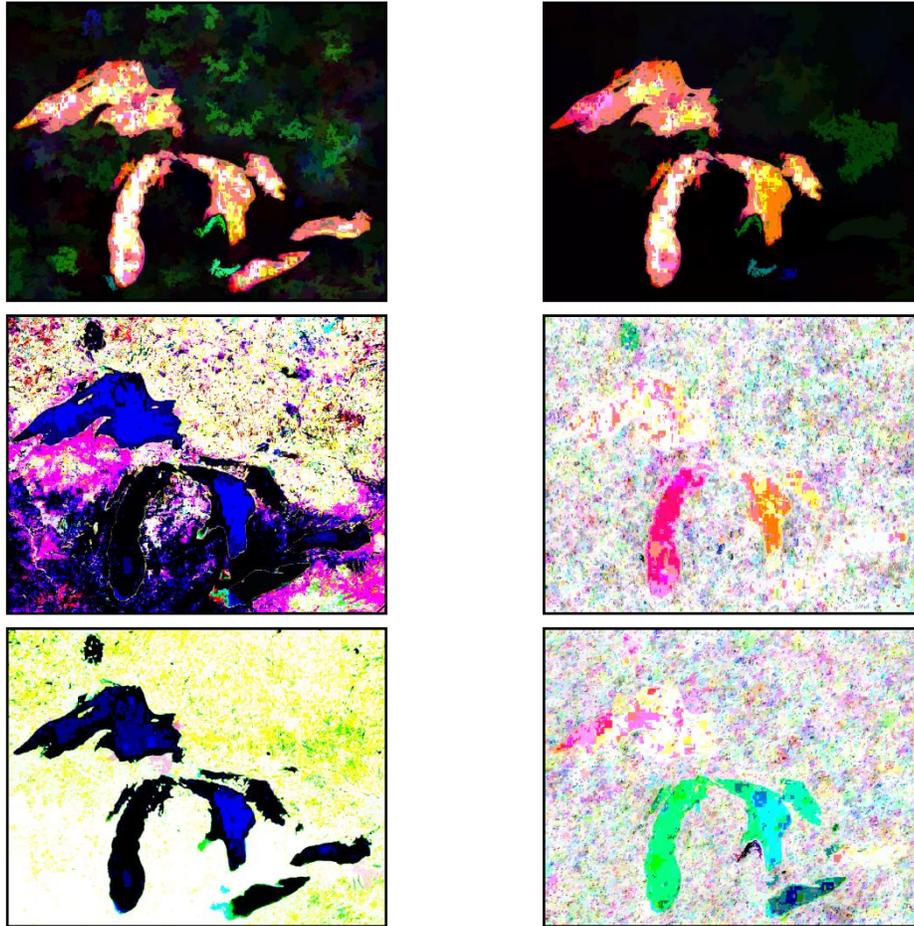


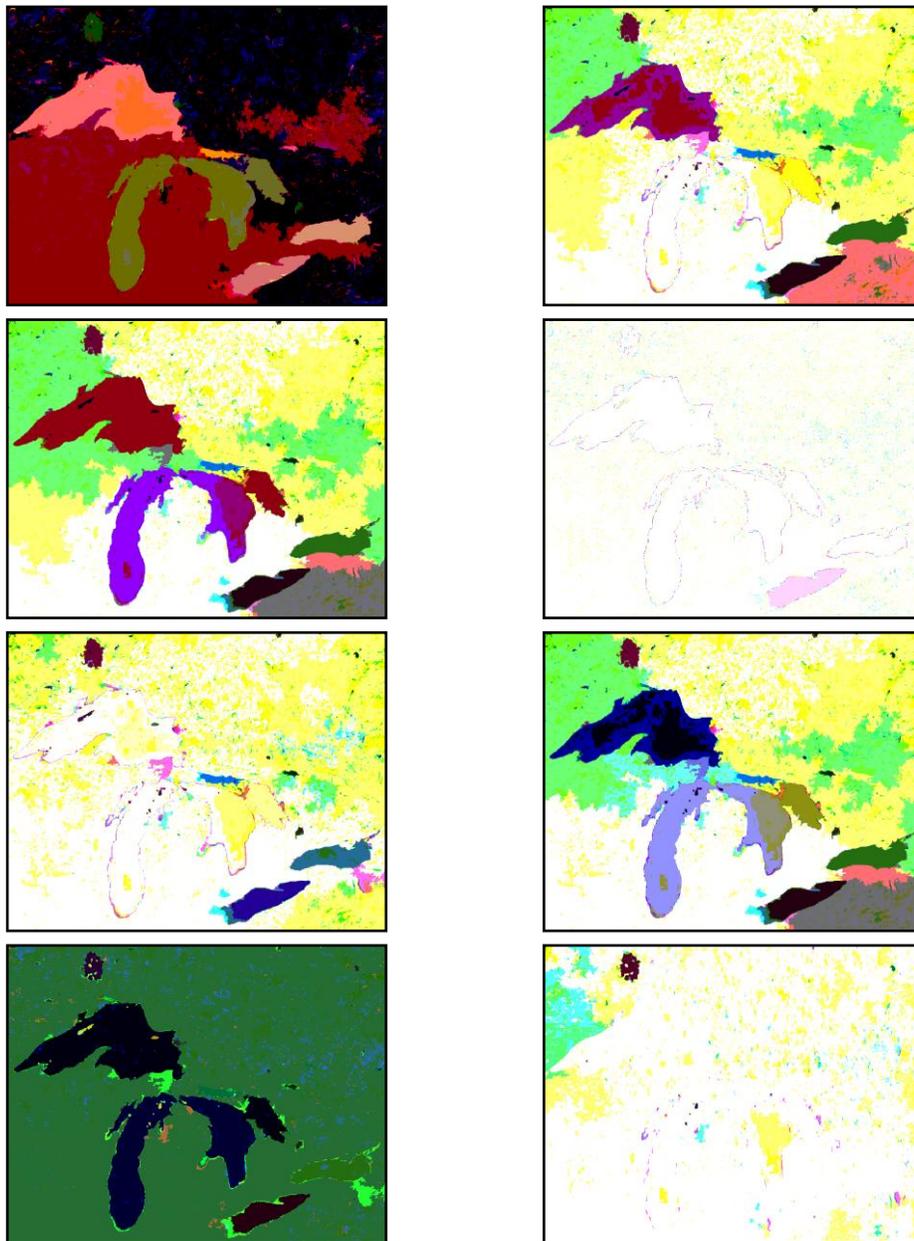
Figura 5–30. Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



e	f
g	h
i	j

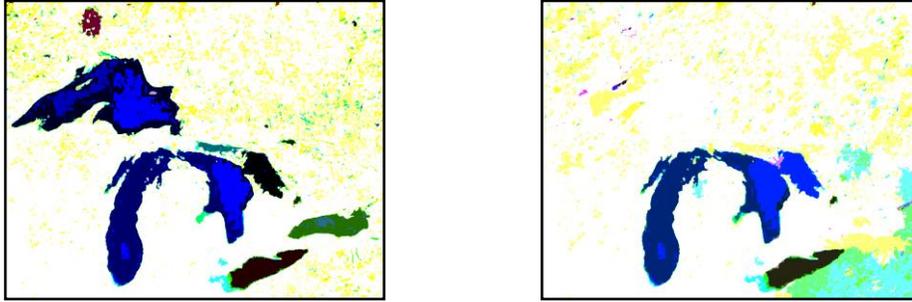
Figura 5–30. (Cont...) Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Tal como se ha comentado, en las primeras iteraciones de este algoritmo se requiere hacer una simplificación de las geoimágenes, aplicando iterativamente la compresión semántica tomando como entrada ya sea la geoimagen de intensidad, o bien, la de intensidad promedio. Por este motivo, se aplicó el algoritmo con un umbral de similitud igual a 20 sobre la geoimagen de intensidad. La siguiente figura contiene las diez geoimágenes simplificadas por cada característica y parámetro de compresión semántica.



a	b
c	d
e	f
g	h

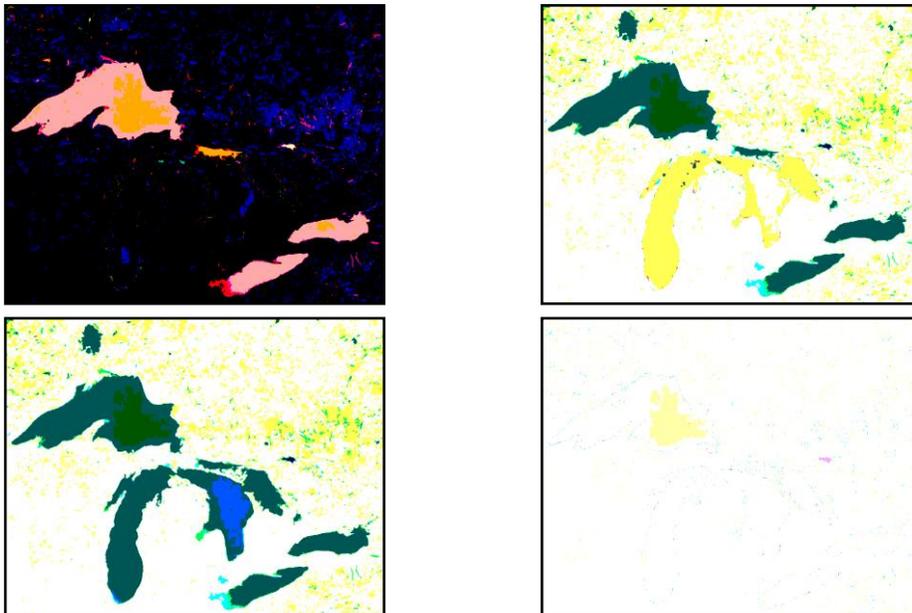
Figura 5–31. Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



i j

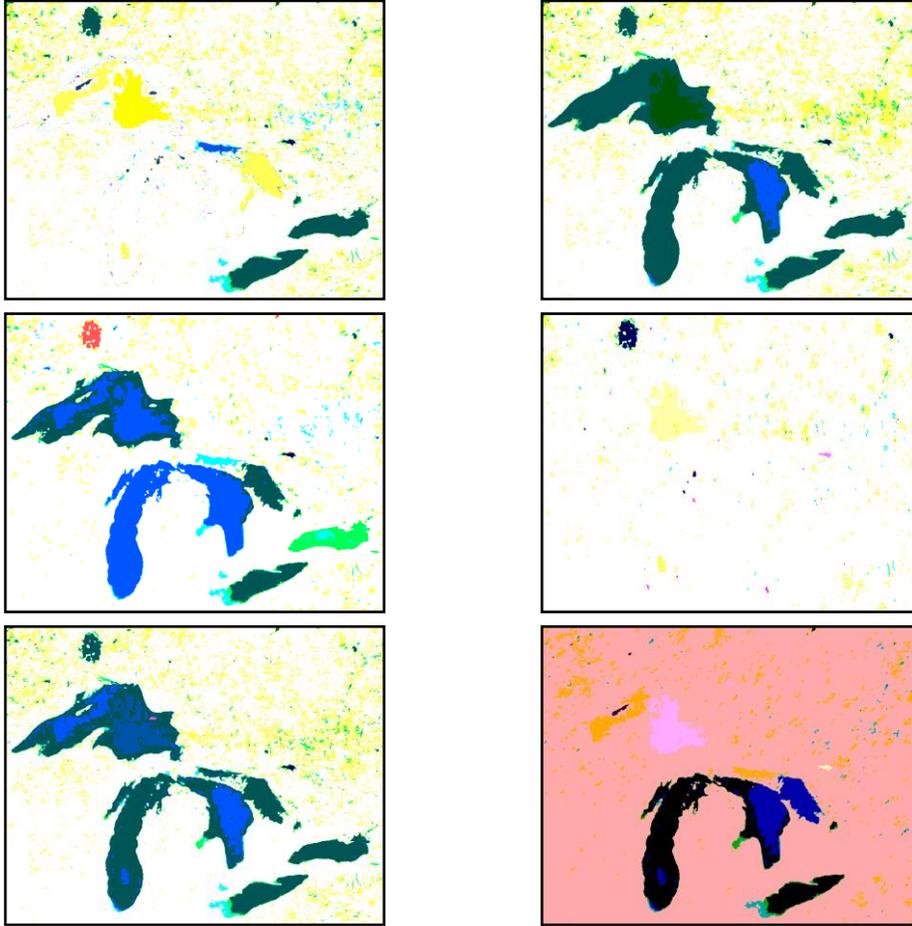
Figura 5–31. (Cont...) Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Todavía no es de utilidad los resultados de compresión usando las características y parámetros distintos a la intensidad e intensidad promedio; por ello, se aplicó una tercera iteración del algoritmo usando como geoimagen de entrada la generada en la segunda iteración con la característica de intensidad.



a b  
c d

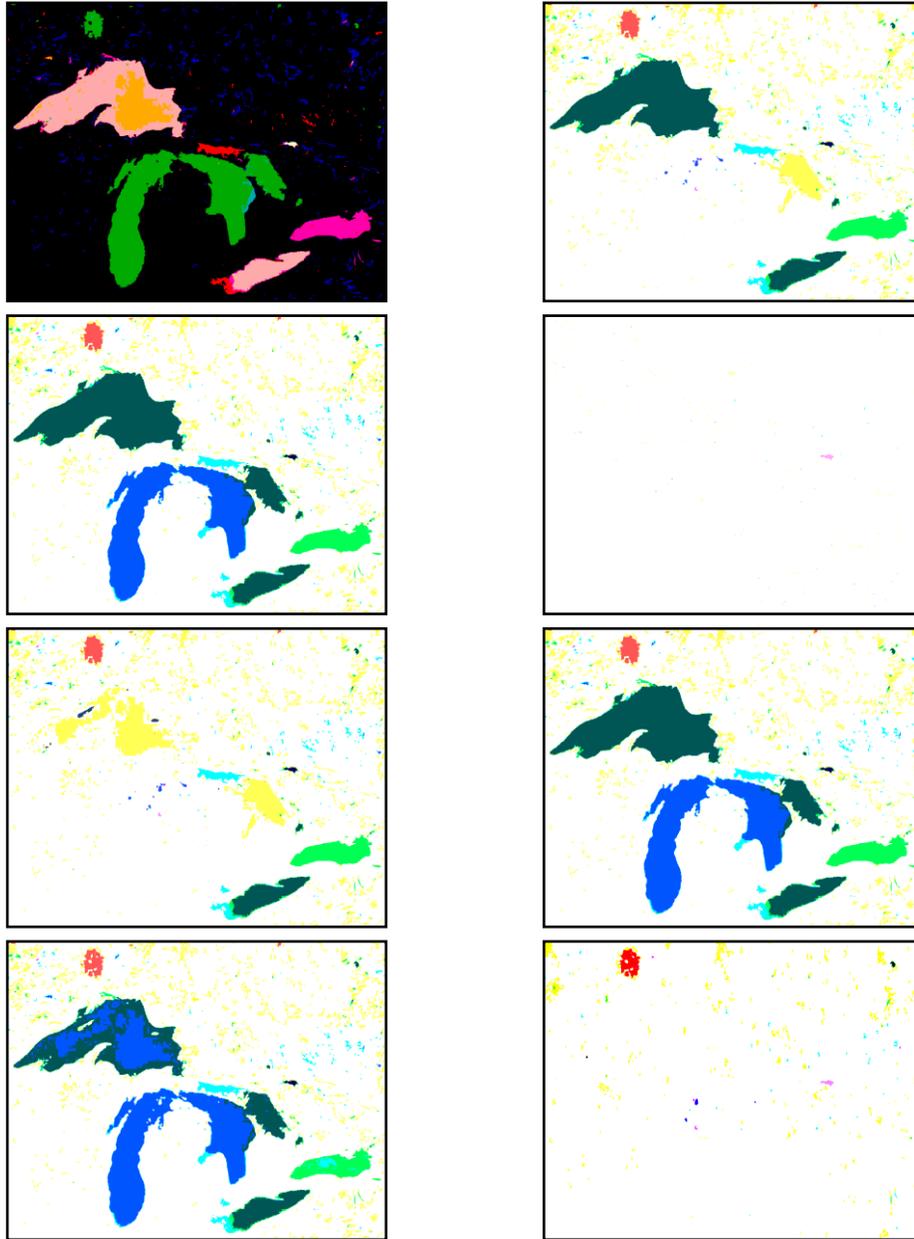
Figura 5–32. Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



e	f
g	h
i	j

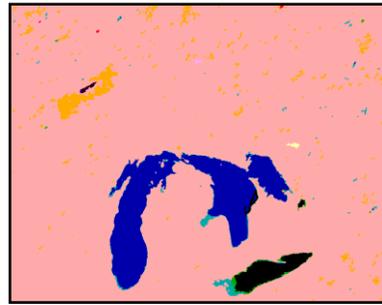
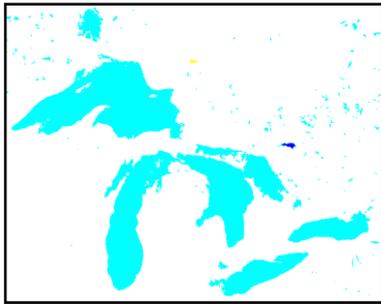
Figura 5–32. (Cont...) Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Se aplicó nuevamente otra iteración del algoritmo de compresión semántica usando el mismo umbral de similitud y tomando como entrada la geoimagen simplificada por intensidad. La detección de geometría en algunas características como el área y el ancho también son muy similares a las regiones que se describen en la geoimagen simplificada por intensidad. Las geoimágenes que se generaron después de terminar la cuarta iteración se muestran en la siguiente figura.



a	b
c	d
e	f
g	h

Figura 5–33. Conjunto de geomágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geomagen de intensidad de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



i j

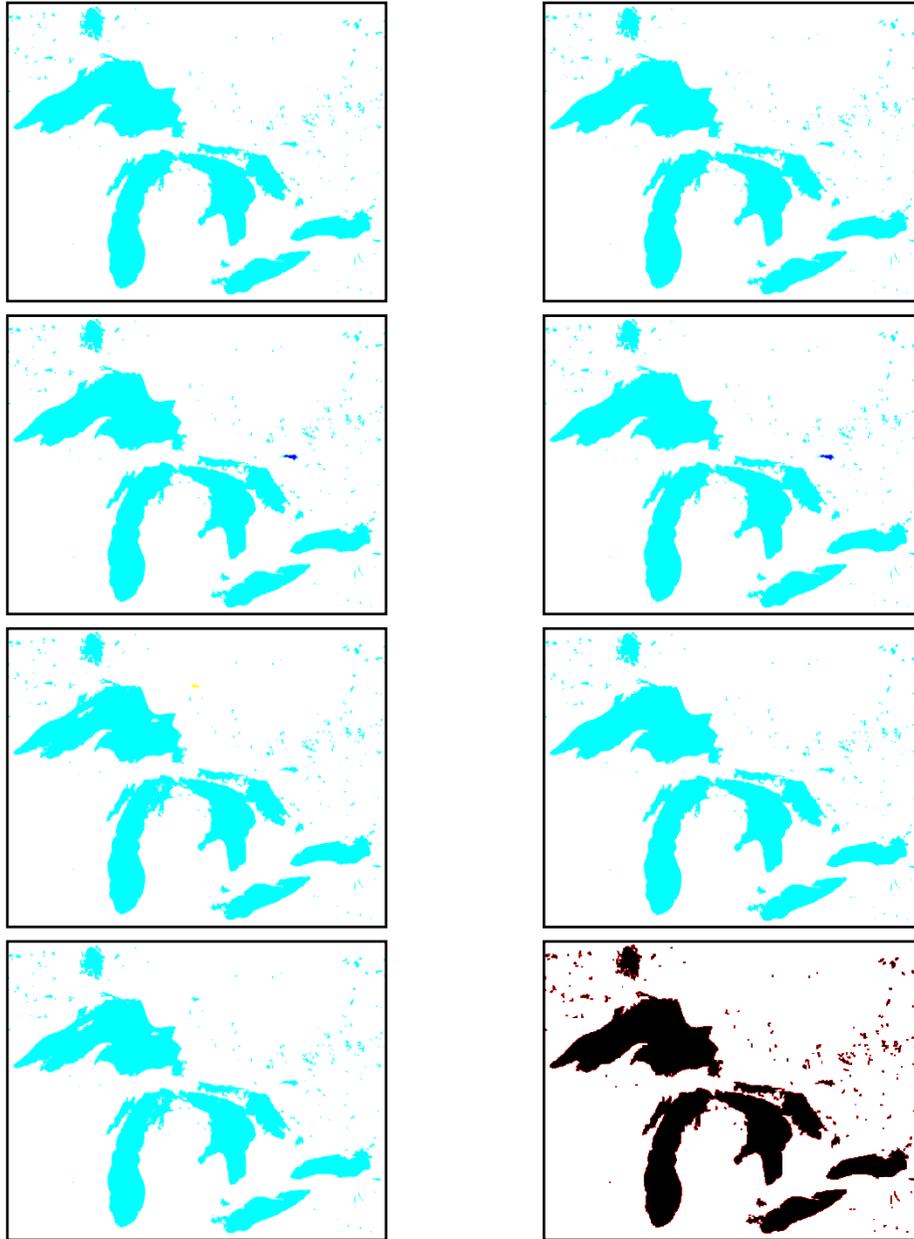
Figura 5–33. (Cont...) Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Se encontró que aplicando una nueva iteración del algoritmo de compresión semántica, haciendo una variación en el umbral de similitud ( $k_{cs} = 05$ ) y tomando nuevamente como entrada la geoimagen de intensidad obtenida en la iteración anterior (cuarta), se pudo obtener la fusión de los segmentos buscada en las regiones de los cuerpos de agua y de suelo. Los resultados aparecen en la Figura 5–34.



a b

Figura 5–34. Conjunto de geoimágenes que se obtienen después de la quinta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 4; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



c	d
e	f
g	h
i	j

Figura 5–34. (Cont...) Conjunto de geoimágenes que se obtienen después de la quinta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 4; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

La geoimagen comprimida semánticamente con la característica de intensidad, permite extraer los objetos geográficos de forma correcta. Es importante resaltar que con la característica de área, se puede obtener otro tipo de simplificación, ya que las regiones de suelo dentro de los cuerpos de agua (islotos), son fusionadas con los cuerpos de agua. En la Figura 5–35 se muestra la geoimagen con un mayor tamaño (se hizo un cambio de los colores de las regiones para poder observarlas con mayor facilidad).



Figura 5–35. Geoimagen simplificada con la característica de área, obtenida en la quinta iteración.

Esto no es un error del algoritmo de compresión semánticamente, ya que en la característica de área no se utiliza el color de los segmentos como variable a cuantificar, sino el área que describen los segmentos. Esta es una de las características del algoritmo, ya que permite controlar el nivel de fusión, obteniendo resultados acorde a las necesidades particulares de un experto. Por ejemplo, algún usuario puede desear obtener las regiones de los grandes lagos, sin importarle los objetos geográficos que haya dentro de cada uno de los cuerpos de agua. Como geoimagen de entrada, se usó la obtenida en la quinta iteración con la característica de intensidad, ver Figura 5–34(i).

### 5.5.3.2 Extracción de regiones

Después de hacer la extracción de regiones (existen dos intensidades de color uniforme) se generaron las geoimágenes que aparecen en la Figura 5–36 y Figura 5–37.



Figura 5–36. Extracción de regiones de la intensidad de color 1.



Figura 5–37. Extracción de regiones de la intensidad de color 2.

### 5.5.3.3 Reconocimiento

Los valores determinados para la media, mediana y desviación estándar para las geoimágenes de cada clase aparecen en la Tabla 5–15.

Tabla 5–15. Resultados para cada una de las geoimágenes extraídas del caso de estudio 2.

Geoimagen	Componente	Media	Mediana	Desviación estándar
1	Rojo	57.6642	51	23.5079
	Verde	88.7490	85	23.1485
	Azul	26.6240	23	15.9633
2	Rojo	01.6257	00	05.2403
	Verde	08.9472	04	13.4381
	Azul	10.4462	10	07.4152

Donde:

Geoimagen 1 – Imagen que contiene los pixeles de la clase 1, ver Figura 5–36.

Geoimagen 2 – Imagen que contiene los pixeles de la clase 2, ver Figura 5–37.

La Tabla 5–16 muestra los resultados numéricos de los cálculos de las distancias Euclidianas de cada geoimagen de temática, a los vectores prototipo de las temáticas propuestas en la metodología. Con estos resultados el algoritmo de reconocimiento proporcionó las siguientes etiquetas generales:

$$T_{I_0} = \{suelo (área verde), cuerpo de agua\}$$

Tabla 5–16. Distancias entre los vectores a reconocer y los vectores prototipos de las temáticas.

Geoimagen	Componente	Vectores prototipos		
		Temática 1	Temática 2a	Temática 2b
1	Rojo	59.9128	183.8330	032.7844
	Verde	87.4738	103.7388	007.0859
	Azul	26.4765	127.5446	057.6181
2	Rojo	19.6117	260.2732	109.6536
	Verde	29.0091	217.7579	107.7419
	Azul	46.0263	148.8745	079.0999

#### 5.5.3.4 Especialización

El número de objetos geográficos en este ejemplo es mayor que los obtenidos previamente; en consecuencia se mostrarán en geoimágenes distintas los resultados del algoritmo de especialización. La geoimagen de la Figura 5–38 contiene los objetos geográficos que son especializados como tipo *INLAND*, es decir, lago o laguna, que son aquellos que están rodeados por un objeto geográfico tipo suelo.



Figura 5–38. Objetos geográficos especializados como tipo *INLAND*.

Los objetos geográficos tipo suelo que son especializados como islotes, por estar rodeados por cuerpos de agua tipo *INLAND*, son los que se muestran en la Figura 5–39. Con el conocimiento existente en la ontología no se puede distinguir entre un objeto lago o laguna, por lo que solamente podemos especializar los cuerpos de agua como tipo *INLAND*. Los propios diccionarios como el de la Real Academia de la Lengua Española citan que una laguna es un lago de menor tamaño, pero el concluir que un objeto es de menor tamaño que otro, requiere del uso de una métrica cuantitativa, por ejemplo el área. Aquí es donde existe una ambigüedad entre estos dos tipos de objetos geoespaciales, ya que no existe una definición del área mínima para un lago. En consecuencia, se requiere consultar con un experto para poder agregar el conocimiento necesario en la ontología para poder resolver la especialización entre lago y laguna. Considerando únicamente con fines ilustrativos, una extracción de los objetos *INLAND* que tienen un área

(cuantificada como el número de píxeles) mayor a 100 para un lago, se tiene lo que aparece en la Figura 5–40.

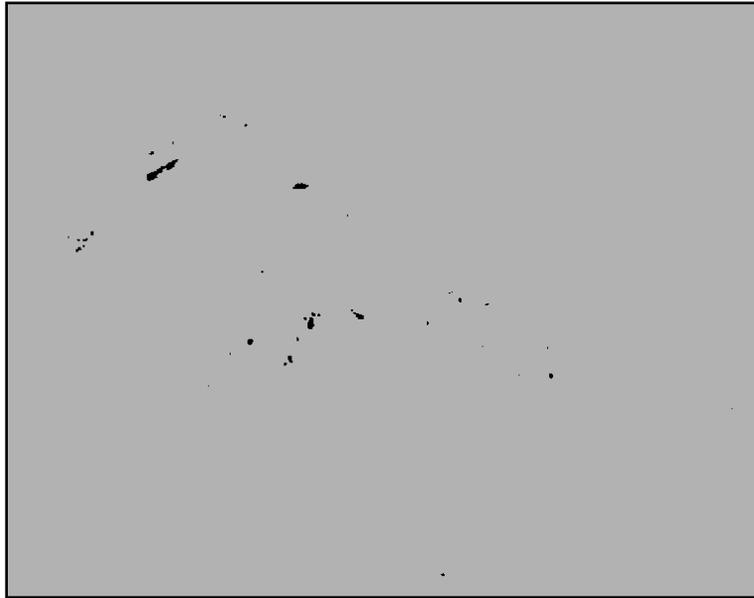


Figura 5–39. Objetos geográficos especializados como tipo islote.



Figura 5–40. Objetos geográficos *INLAND* especializados como lagos.

Los objetos geográficos que no cumplen con la condición de lagos pueden ser interpretados como lagunas, ver Figura 5–41. Lo anterior no significa que esta sea la forma de resolver la especialización entre lago y laguna, ya que como se comentó anteriormente, es el experto el único

que puede establecer la forma correcta de hacerlo, definiendo una relación geométrica (el tamaño de cada objeto) y a su criterio particular.

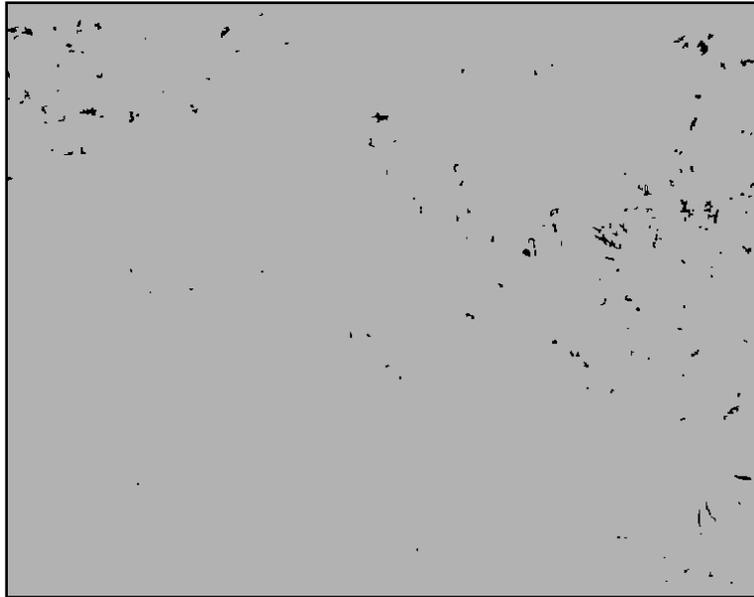


Figura 5–41. Objetos geográficos *INLAND* especializados como lagunas.

Finalmente, considerando una sola instancia de las etiquetas especializadas, se tiene que la semántica de la geoimagen es:  $S_{I_0} = \{suelo, INLAND, islote\}$ .

#### 5.5.4 Caso de estudio 4: Geoimagen de la República Mexicana

El último caso de estudio corresponde a la descomposición semántica de una geoimagen de la República Mexicana, la cual se puede observar en la Figura 5–42.



Figura 5–42. Geoimagen de la República Mexicana.

Los histogramas de cada componente de color de esta geoimagen se muestran en la Figura 5–43.

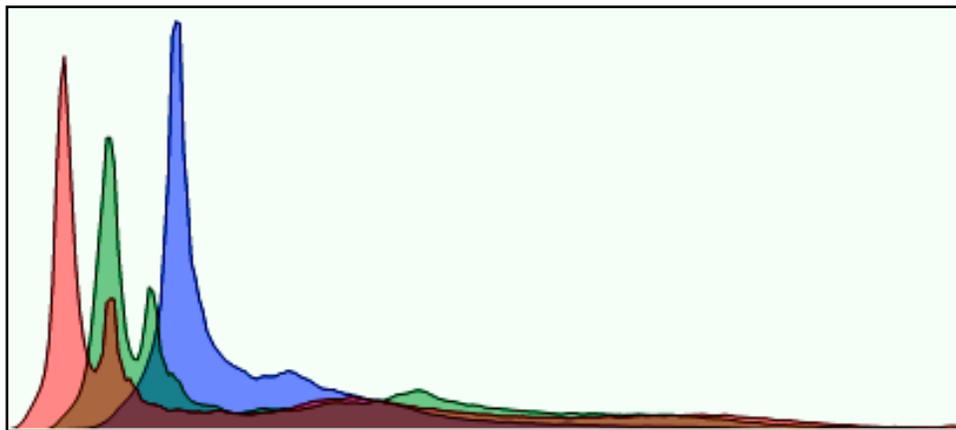


Figura 5–43. Histogramas de las componentes de color roja, verde y azul de la geoimagen de la República Mexicana.

5.5.4.1 Compresión semántica

La cadena de compresión semántica que permitió simplificar esta geoimagen fue:

$$C_{cs}(I_o, I_L) = \{ (I_0, 21.5), (I_2, intensidad, 2.5), \} \\ \{ (I_3, intensidad, 1.5), (I_4, D, 90.0) \}$$

En la Tabla 5–14 se desglosa la secuencia de compresión semántica.

Tabla 5–17. Parámetros del algoritmo de compresión semántica para el caso de estudio 4.

Número de iteración	Geoimagen de entrada	Umbral de similitud
1	Imagen original	$k_{cs}=21.5$
2	Intensidad (I)	$k_{cs}=02.5$
3	Intensidad (I)	$k_{cs}=01.5$
4	Distancia (d)	$k_{cs}=90.0$

La Figura 5–44 contiene las geoimágenes simplificadas después de terminar la primera iteración del algoritmo de compresión semántica.

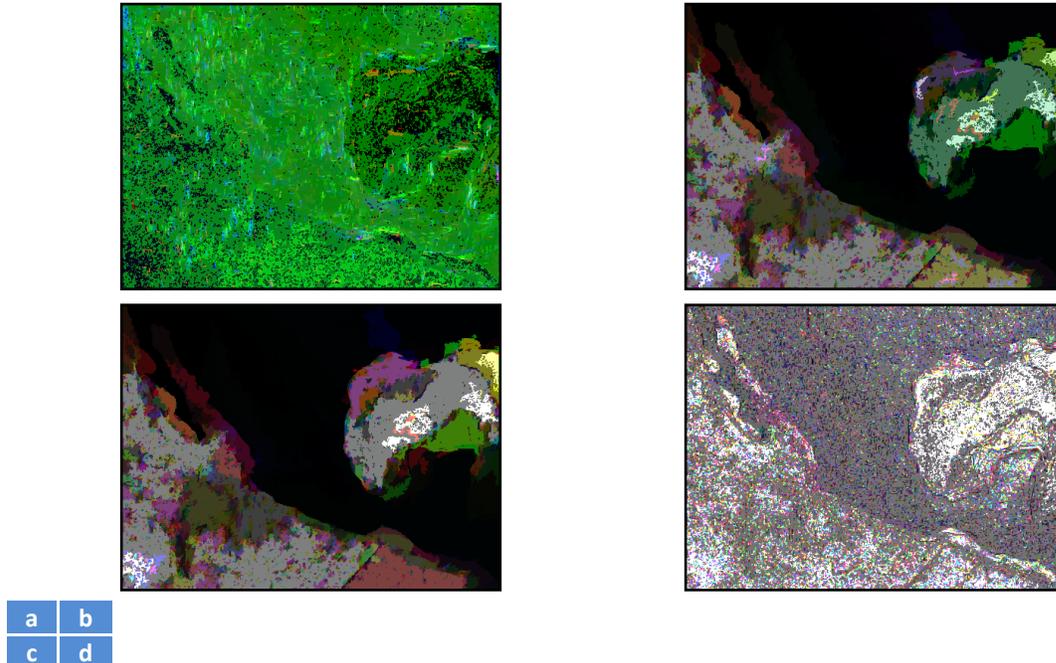
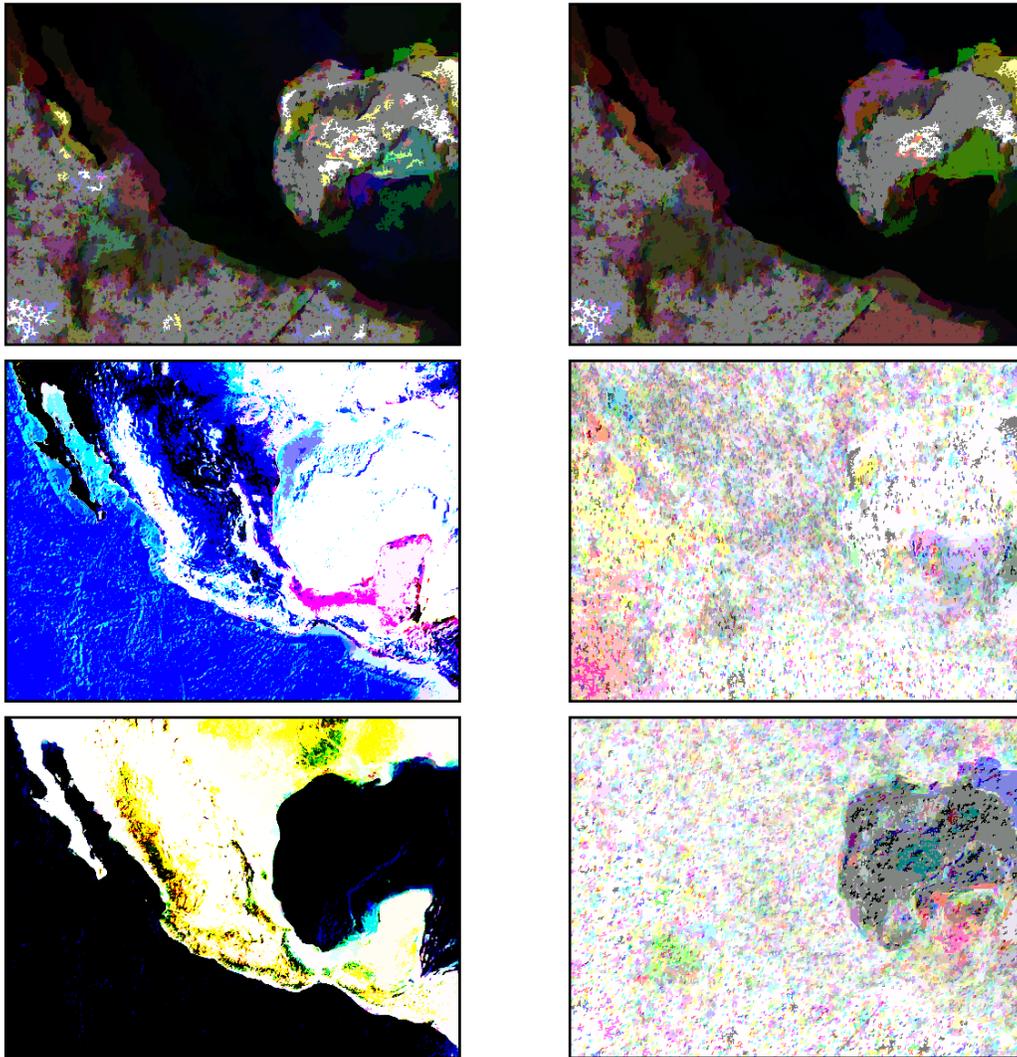


Figura 5–44. Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

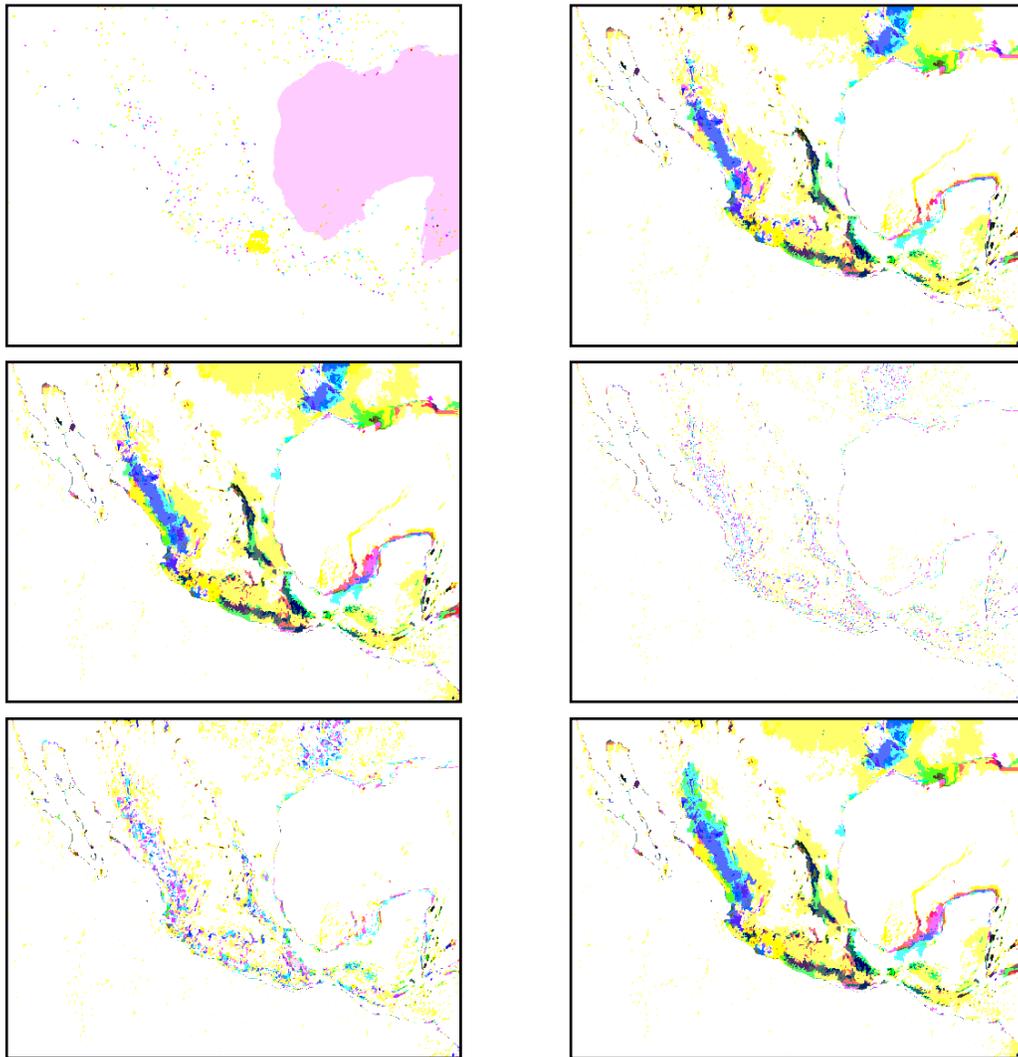


e	f
g	h
i	j

Figura 5–44. (Cont...) Conjunto de geoimágenes que se obtienen después de la primera iteración del algoritmo de compresión semántica; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Las regiones que se obtienen con las características de largo, área, tamaño y ancho permiten describir la geometría de los objetos geográficos más significativos de la imagen original; sin embargo, las regiones todavía no están muy bien delimitadas. Por esta razón, se aplicó nuevamente el algoritmo tomando la geoimagen simplificada con la característica de intensidad

con un umbral de similitud igual a 2.5. Las geoimágenes resultantes de la segunda iteración aparecen a continuación, ver Figura 5–45. Se escogió un valor pequeño para el umbral de similitud porque empleando uno como el que se usó en la iteración anterior ( $k_{cs} = 21.5$ ) no se obtuvo una fusión adecuada.



a	b
c	d
e	f

Figura 5–45. Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

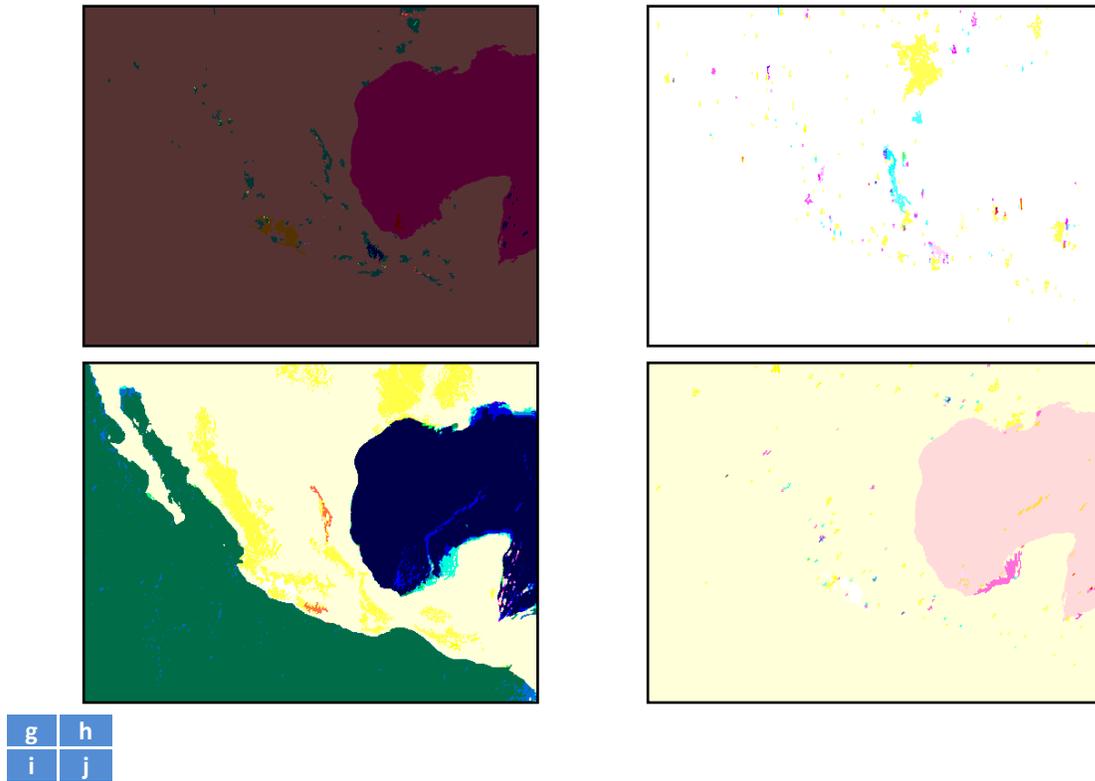
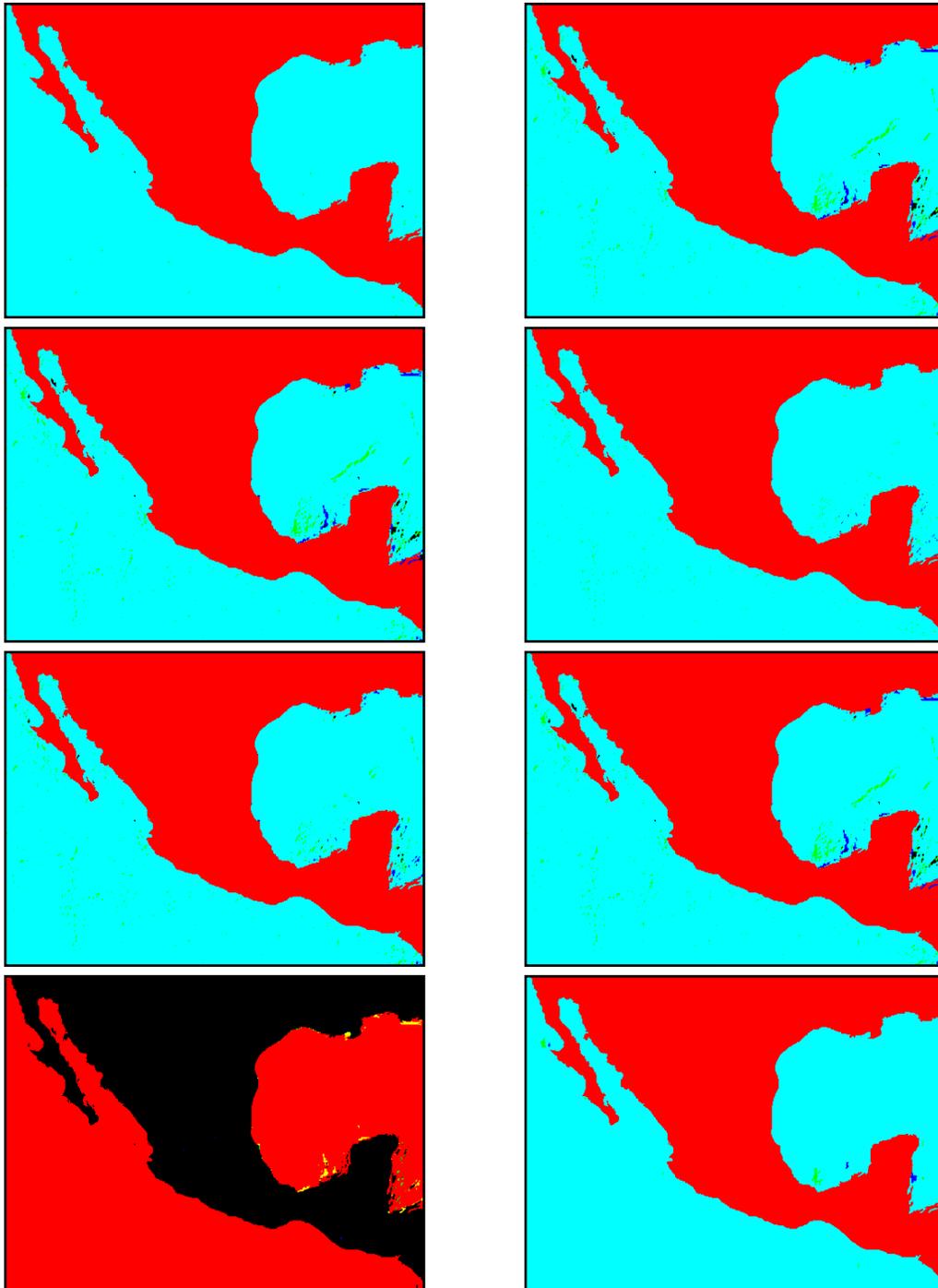


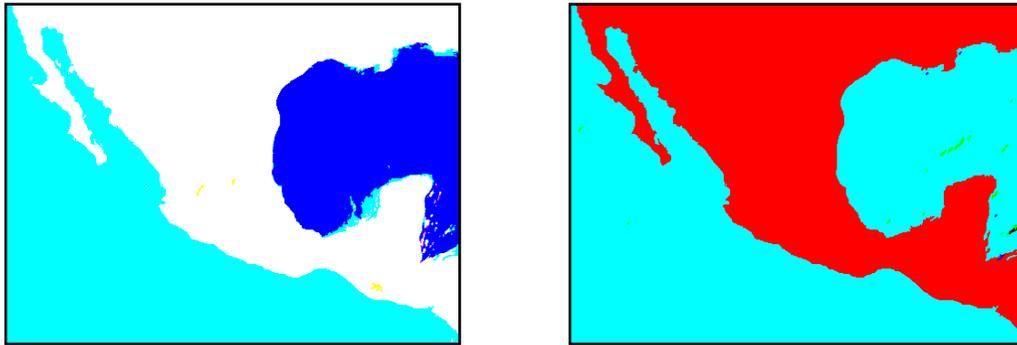
Figura 5–45. (Cont...) Conjunto de geoimágenes que se obtienen después de la segunda iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 1; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Nuevamente, la geoimagen de intensidad tiene una mejor simplificación a comparación de los resultados obtenidos con el resto de características y parámetros. Se usó esta geoimagen como entrada para una tercera iteración del algoritmo de compresión semántica, especificando un umbral de similitud pequeño ( $k_{cs} = 1.5$ ) para eliminar las variaciones que existen en las intensidades de color y se generaron las geoimágenes que aparecen en la Figura 5–46.



a	b
c	d
e	f
g	h

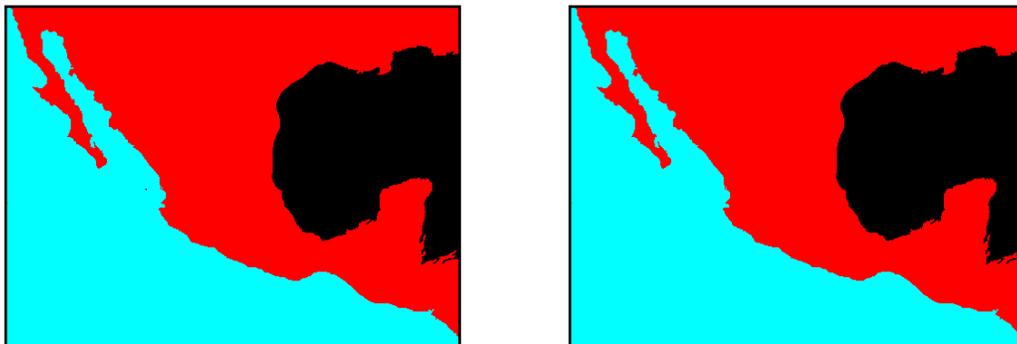
Figura 5–46. Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



i j

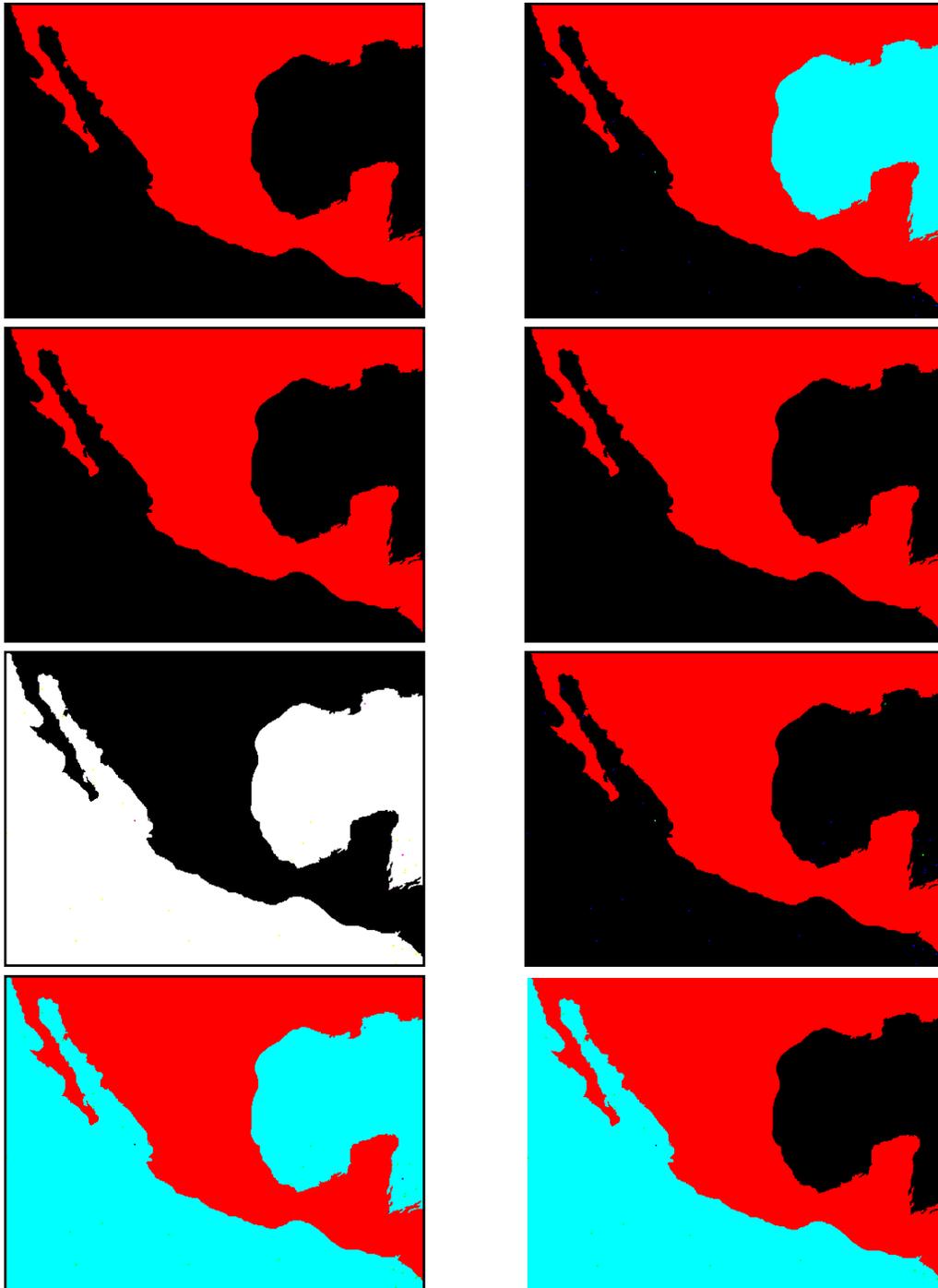
Figura 5–46. (Cont...) Conjunto de geoimágenes que se obtienen después de la tercera iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de intensidad de la iteración 2; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

En estas geoimágenes se puede apreciar una mayor detección de geometría y topología de las regiones de los objetos geográficos. Sin embargo, todavía faltan regiones que fusionar, principalmente en los cuerpos de agua. Se realizaron pruebas haciendo uso de la geoimagen de intensidad pero ya no se obtuvieron mejores resultados; por esta razón se utilizaron otras geoimágenes. Con la geoimagen de distancia y un alto umbral de similitud ( $k_{cs} = 90$ ) se llegó a la simplificación deseada. La siguiente figura contiene las geoimágenes obtenidas después de terminar la cuarta iteración de la compresión semántica.



a b

Figura 5–47. Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de distancia de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.



c	d
e	f
g	h
i	j

Figura 5–47. (Cont...) Conjunto de geoimágenes que se obtienen después de la cuarta iteración del algoritmo de compresión semántica, usando como entrada la geoimagen de distancia de la iteración 3; (a) Característica d; (b) Característica L; (c) Característica S; (d) Parámetro Ss; (e) Característica Sz; (f) Característica W; (g) Característica ABS; (h) Parámetro C2f; (i) Característica I; (j) Parámetro S2f.

Con relación a la figura anterior, en todas las geoimágenes resultantes existe una simplificación muy alta, acorde a los tres objetos geográficos con menor nivel de detalle (más grandes). Nuevamente, es el experto el que tiene un control sobre el nivel de compresión semántica (agrupamiento) de los objetos geoespaciales. Por esta razón, la metodología propuesta es adaptativa, ya que puede ser usada por una gran cantidad de especialistas, según los resultados que sean necesarios obtener.

Como geoimagen de entrada para la etapa de síntesis, se usó la imagen obtenida en la cuarta iteración simplificada con la característica de largo; esta geoimagen se puede observar a continuación.

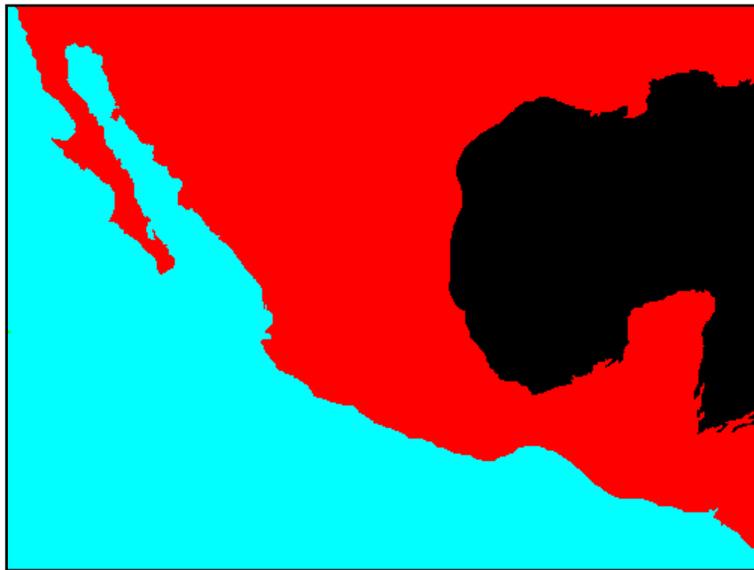


Figura 5–48. Geoimagen simplificada con la característica de largo.

#### 5.5.4.2 Extracción de regiones

Esta geoimagen está descrita por medio de tres regiones, cada una con una intensidad de color uniforme, por lo que deben generarse tres geoimágenes diferentes. En la Figura 5–49 aparecen las geoimágenes de regiones que se extraen de la Figura 5–48. Para este caso de estudio, el número de geoimágenes no corresponde todavía con el número de temáticas que se desea extraer, pues recordemos que son 2, la de cuerpo de agua y la de suelo. Cuando se realice el reconocimiento, el número de etiquetas generales diferentes será igual al número de temáticas buscado.



- a
- b
- c

Figura 5-49. Resultado de la extracción de regiones de la geoimagen de la característica de largo.

### 5.5.4.3 Reconocimiento

Se determinó la información de la media, mediana y desviación estándar para cada geoimagen de intensidad de color uniforme encontrada en el punto anterior. La información numérica puede consultarse en la Tabla 5–18.

Tabla 5–18. Resultados para cada una de las regiones del caso de estudio 4.

Geoimagen	Componente	Media	Mediana	Desviación estándar
1	Rojo	024.8513	024	006.1375
	Verde	036.6432	036	053.0000
	Azul	006.1375	007	006.5902
2	Rojo	130.9768	119	048.1519
	Verde	130.0658	123	039.5071
	Azul	093.2756	087	031.3378
3	Rojo	040.0932	037	009.4188
	Verde	051.4974	048	010.8699
	Azul	056.2263	055	008.7394

Donde:

Geoimagen 1 – Imagen que contiene la información original de los píxeles que fueron agrupados en la clase 1, ver Figura 5–49(a).

Geoimagen 2 – Imagen que contiene la información original de los píxeles que fueron agrupados en la clase 2, ver Figura 5–49(b).

Geoimagen 3 – Imagen que contiene la información original de los píxeles que fueron agrupados en la clase 3, ver Figura 5–49 (c).

Con la información de la Tabla 5–19, el algoritmo de reconocimiento de los objetos geográficos determinó que las geoimágenes 1 y 3 son cuerpos de agua, mientras que las tercera geoimagen es tipo suelo, como son dos etiquetas diferentes, el resultado es:

$$T_{I_o} = \{suelo, cuerpo de agua\}$$

Tabla 5–19. Distancias entre los vectores a reconocer y los vectores prototipos de las temáticas.

Geoimagen	Componente	Vectores prototipos		
		Temática 1	Temática 2 <sup>a</sup>	Temática 2b
1	Rojo	014.2163	226.9248	076.6919
	Verde	014.3383	176.1935	066.9713
	Azul	015.0459	088.8678	021.9994
2	Rojo	162.6668	087.0938	072.3774
	Verde	145.8026	049.6405	064.8912
	Azul	070.8542	035.6587	036.9150
3	Rojo	034.4861	206.7775	056.5300
	Verde	033.6327	156.9491	047.7034
	Azul	018.1851	085.2278	018.1155

#### 5.5.4.4 Especialización

Esta geoimagen tiene un nivel de detalle mayor que todas las analizadas en los otros casos de estudio; un experto también coincidiría en que su nivel de detalle es muy grueso. Para estos tres objetos geográficos no se cumple ninguna relación topológica, en específico la de **rodeado**, por lo que considerando únicamente el nivel de detalle, se tendría la especialización de que los dos cuerpos de agua son tipo océano, mientras que el objeto geográfico tipo suelo quedaría con esa misma etiqueta, pudiéndose concluir únicamente que es un suelo tipo tierra o rocoso; en consecuencia:

$$S_{I_0} = \{océano, suelo\}$$

## 5.6 Comparación con eCognition

La imagen de la Figura 5–8 es la que acompaña el ejemplo adjunto a la versión de prueba de eCognition que disponemos. Los pasos que se indican en la guía de uso de este programa son primeramente crear un proceso para poder aplicar una segmentación multi-resolución a la imagen

original. En la Figura 5–46 se pueden apreciar los parámetros que se deben usar en la segmentación multi-resolución<sup>36</sup>.

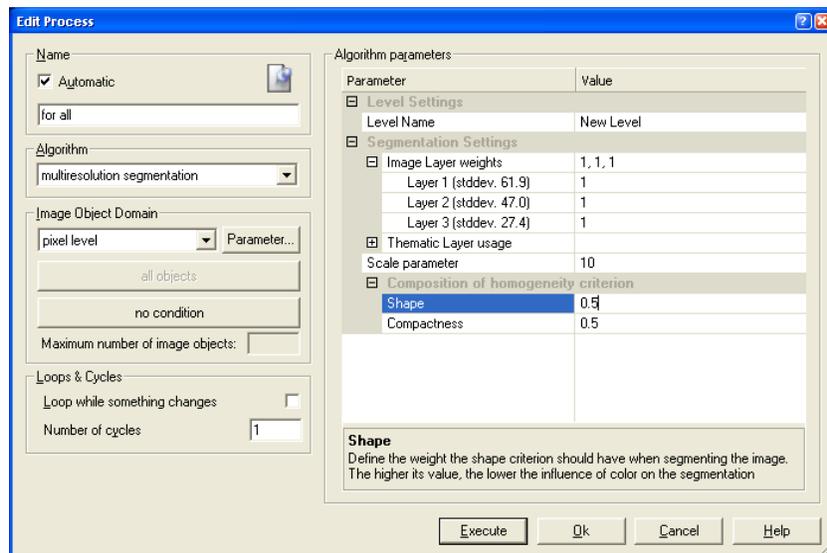


Figura 5–50. Parámetros de para el algoritmo de segmentación multi-resolución.

El resultado de la segmentación se muestra en la Figura 5–51.



Figura 5–51. Resultado de la segmentación multiresolución.

Se puede observar que la segmentación multi-resolución genera muchas clases, debido principalmente a las características de la imagen original. En consecuencia es necesario aplicar una

<sup>36</sup> Esta información fue obtenida de la guía de uso del programa e-Cognition; desafortunadamente no se da una explicación que justifique el por qué se usan esos valores. Esto mismo aplica para la segmentación basada en conocimiento que se describe posteriormente.

segmentación basada en conocimiento, para lo cual se deben definir los criterios que usará esta técnica de segmentación (Figura 5–52). Finalmente, la imagen resultante es la que aparece en la Figura 5–53.

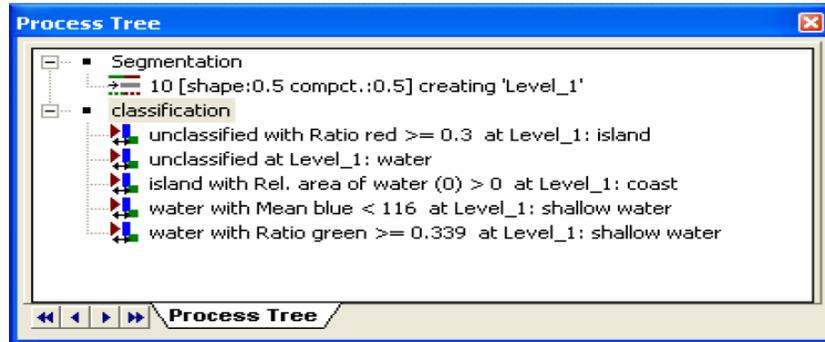


Figura 5–52. Criterios para la segmentación basada en conocimiento.

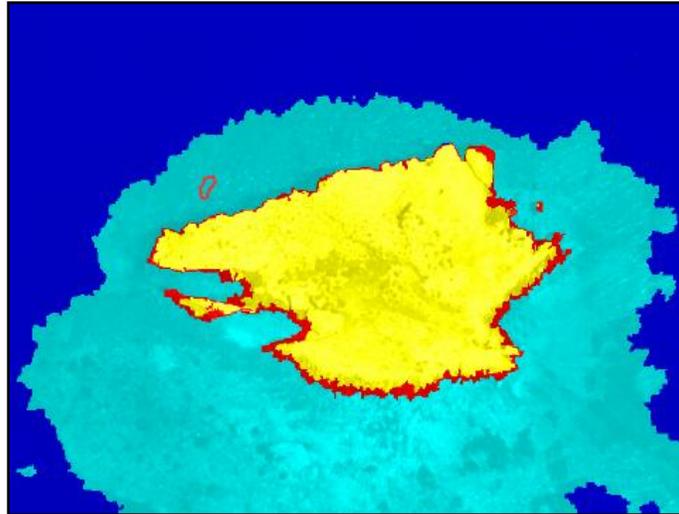


Figura 5–53. Geoimagen final segmentada.

La segmentación final tiene una adecuada correspondencia con tres elementos: parte profunda del cuerpo de agua, zona rocosa debajo del agua y la parte de la isla; esto se debe en gran medida a la segmentación basada en conocimiento. Sin embargo, observando los parámetros de este algoritmo, para cada clase resultante es necesario especificar el valor de las propiedades geométricas que deben satisfacer los segmentos para ser incluidos en una clase, lo cual implica que el usuario tiene que guiar adecuadamente la segmentación para poder obtener resultados satisfactorios. Adicionalmente, no es fácil determinar los parámetros, como son el caso de una *media* menor que 116, o un *radio* mayor o igual que 0.3.

El primer paso, acorde con lo explicado en el manual de ayuda que acompaña al programa de eCognition, consiste en realizar un agrupamiento de los píxeles que tienen un radio en la componente de color roja, mayor o igual que 0.3 para clasificarlo como tipo isla, el resultado aparece en la Figura 5–54.



Figura 5–54. Clasificación de objetos con radio mayor o igual que 0.3 en la componente roja.

Modificando el parámetro de radio en la componente de color roja, de 0.3 a 0.4 se obtiene el resultado que aparece en la Figura 5–55, mientras que usando un valor de 0.25, se obtuvo lo que se muestra en la Figura 5–56. Se puede observar que con esta ligera modificación al valor del parámetro, la clasificación sufrió cambios en ambos casos, en el primero muy drástico. Adicionalmente, el procedimiento de clasificación basado en conocimiento de esta herramienta, realiza una clasificación etapa por etapa, es decir, después de terminar de clasificar los píxeles en una región (que en este caso sería lo que se clasificó como isla) se emplea el conjunto de píxeles que no fueron clasificados y sobre este conjunto se definen nuevas reglas para agruparlos en otra clase. Este proceso se repite sucesivamente, lo cual es una forma de guiar completamente el mecanismo de clasificación, dejando como responsabilidad del usuario todo el control de la clasificación.

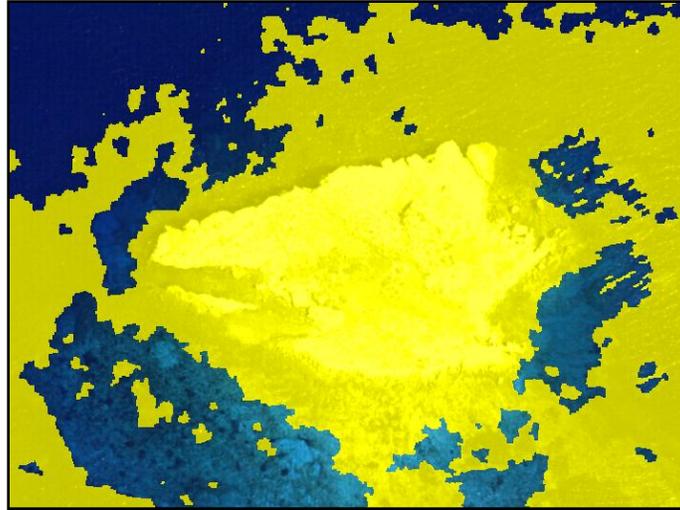


Figura 5–55. Clasificación de objetos con radio mayor o igual que 0.4 en la componente roja.



Figura 5–56. Clasificación de objetos con radio mayor o igual que 0.25 en la componente roja.

Por último, otra de las principales desventajas de esta herramienta de cómputo es que el conjunto de parámetros de clasificación es variable según la imagen a procesar. Por ejemplo, con esta geoimagen se usaron entre otros parámetros la mediana y el radio en una componente de color; mientras que en otro caso como el citado por Mueller se usan la dirección de bordes, ángulo, longitud, proximidad entre bordes, etc. (Mueller et al., 2004). En consecuencia, los parámetros a emplear están estrechamente ligados con la imagen a clasificar. En contraste, el algoritmo de compresión semántica usa los mismos parámetros: número de iteraciones, umbral de similitud y la geoimagen de entrada en cada nueva iteración.

## 5.7 Enfoque propuesto vs enfoque clásico

Para mostrar que el procesamiento basado en el color no es suficiente para poder hacer una simplificación de cualquier tipo de geoimagen, se hizo una comparación de los resultados obtenidos entre el enfoque propuesto (compresión semántica que utiliza color, geometría y topología para simplificar una geoimagen) con un análisis basado únicamente en color, que es el método clásico usado por las técnicas orientadas a píxeles.

Se dice que el enfoque tradicional se basa únicamente en el color, porque para determinar alguna semejanza entre píxeles, solamente se usa la información proporcionada por las intensidades de color, es decir, no se cuantifica alguna propiedad geométrica entre los píxeles, como es el tamaño, el largo, el ancho, el área; lo cual si está considerado en el conjunto completo de propiedades y características de compresión semántica.

Este análisis clásico consistió en aplicar la compresión semántica simplificando las geoimágenes utilizando las características de intensidad e intensidad promedio, así como otras etapas clásicas del PDI, como es el preprocesamiento. Adicionalmente, se aplicó un algoritmo de agrupamiento y otro de eliminación de regiones parásitas. Se usaron los cuatro casos de estudio analizados previamente, en dos de ellos si se pudo obtener la simplificación adecuada, mientras que en las otras dos geoimágenes, un análisis orientado a color no fue suficiente para extraer las regiones de los objetos geoespaciales.

### 5.7.1 Agrupamiento

Esta fase consiste en aplicar un algoritmo de agrupamiento, que puede ser el algoritmo  $k$ -medias, el  $c$ -medias difuso o el algoritmo ISODATA, por su amplio uso en procesamiento de imágenes digitales. Este algoritmo requiere que se especifique manualmente el número de clases o agrupamientos que se generarán ( $n_s$ ). Para los ejemplos que se estudiaron, este parámetro consistió en indicar el número de temáticas en el que se debían clasificar los píxeles de la geoimagen que se obtiene a la salida de la compresión semántica.

Partiendo de la geoimagen que se elige después de aplicar la compresión semántica (solamente puede ser la geoimagen de intensidad o intensidad promedio), se aplica el algoritmo de agrupamiento en la geoimagen, buscando generar un total de  $n_s$  clases. El resultado es una geoimagen segmentada, ver Figura 5–57, donde cada elemento discreto o píxel de la imagen está clasificado en alguna clase o región ( $R_i$ ) que satisface las condiciones citadas en la Sección 3.3.3.3 para  $n = n_s$ .

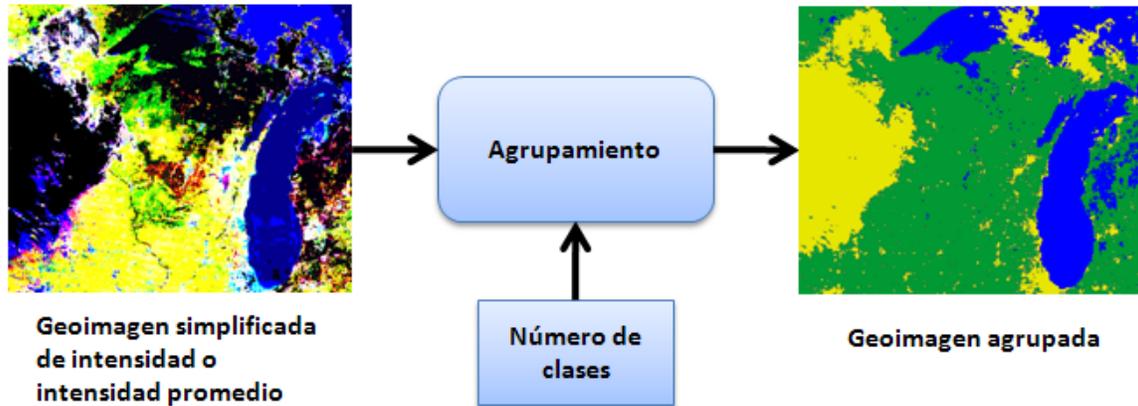


Figura 5–57. Algoritmo de clasificación.

### 5.7.2 Eliminación de regiones parásitas

Como siguiente paso puede ser necesario eliminar las regiones pequeñas o parásitas. Para ello se definió el siguiente algoritmo, el cual se debe aplicar a la geoimagen obtenida con el algoritmo de segmentación, consistente en el etiquetado de las regiones y una etapa de eliminación de regiones parásitas. Una vez que se tienen identificadas todas las regiones mediante el algoritmo de etiquetado de Haralick, se eliminan las regiones parásitas acorde con el siguiente criterio:

Sea  $R$  una región cuya área es igual a  $A_R$ <sup>37</sup>, si  $A_R$  es menor a un umbral  $k_{rp}$  esta región se elimina, reemplazando la intensidad de todos los píxeles de  $R$  por la intensidad de la región en la que está contenida.

El análisis orientado a procesamiento del color quedaría como se muestra en la Figura 5–58.

<sup>37</sup> El área está dada por el número de píxeles que conforman la región.

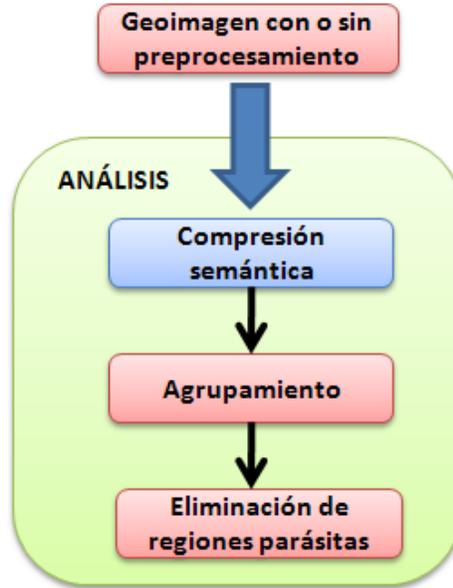


Figura 5–58. Análisis tradicional basado en procesamiento del color.

### 5.7.3 Comentarios de los resultados

Se realizaron diversas pruebas sobre las cuatro geoimágenes que se han analizado en el capítulo, haciendo simplificaciones de la geoimagen de intensidad e intensidad promedio con diferentes umbrales de similitud. Para la primera isla se repitieron seis iteraciones del algoritmo de compresión semántica, mientras que para el resto de las pruebas, solamente se hicieron tres iteraciones. Solamente en dos de los cuatro casos de estudio, con ayuda del algoritmo de agrupamiento se pudo encontrar una simplificación similar a la obtenida anteriormente (haciendo una compresión semántica con todo el conjunto de características y parámetros). A continuación se describen brevemente los mejores resultados obtenidos en cada uno de los casos. En el Anexo A, el lector puede encontrar todas las geoimágenes que se obtuvieron, por si es de su interés el consultarlas.

### 5.7.4 Caso de estudio 1: Geoimagen de una isla, ejemplo 1.

Con esta geoimagen, se pudo llegar a una buena clasificación, aplicando primeramente la compresión semántica con un umbral de similitud igual a 30, haciendo una simplificación de la geoimagen de intensidad promedio. En la siguiente figura se muestra la simplificación de la

geoimagen original. Después de la tercera iteración, la geoimagen fue invariante, esto puede verificarse observando los resultados.

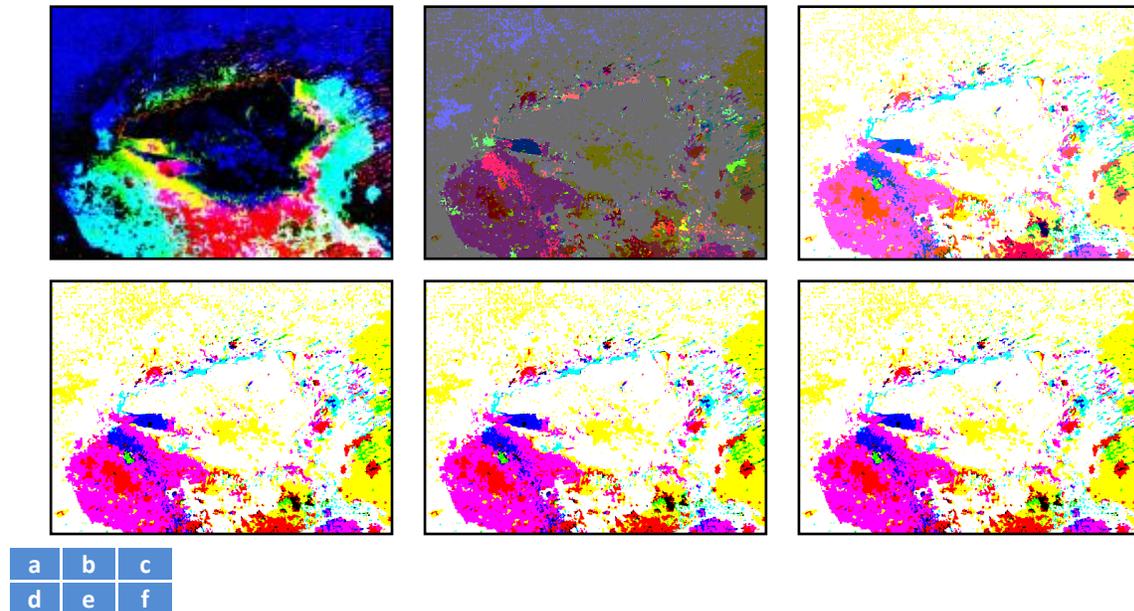


Figura 5–59. Resultados de cada iteración para la simplificación de la geoimagen de intensidad promedio con un umbral de similitud igual a 30; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3; (d) Iteración 4; (e) Iteración 5; (f) Iteración 6.

De los resultados que se obtuvieron con otros umbrales de similitud se observó el siguiente comportamiento:

1. A partir de la tercera iteración ocurría una subclasificación, ya que las regiones de los dos objetos geográficos se fusionaban en una sola, lo cual repercutió en que ya no se podían recuperar los objetos geoespaciales; o bien
2. Se realizaba una simplificación en las regiones, sin embargo, no fue posible igualar los resultados obtenidos cuando se usó todo el conjunto de características y parámetros.

Por esta razón, tuvo que usarse una técnica de agrupamiento, para poder hacer una mayor simplificación de las regiones de los objetos geográficos. Después de aplicar el algoritmo de

agrupamiento  $k$ -medias a cada una de las geoimágenes que aparecen en la Figura 5–59 se generaron las siguientes imágenes agrupadas.

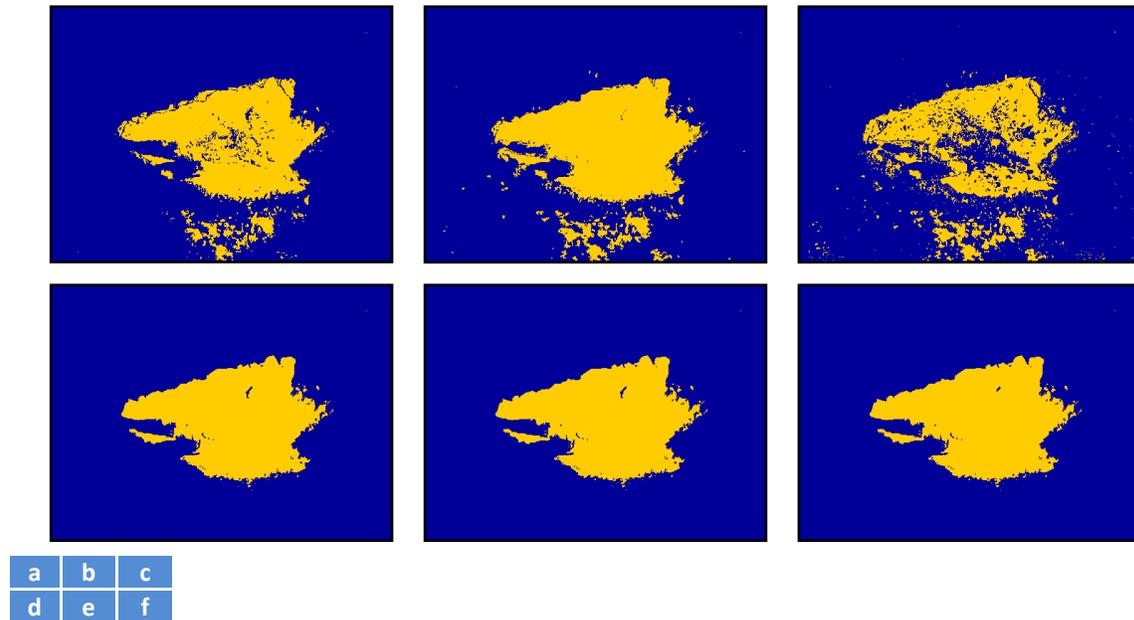


Figura 5–60. Resultados del agrupamiento para la simplificación de la geoimagen de intensidad promedio con un umbral de similitud igual a 30; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3; (d) Iteración 4; (e) Iteración 5; (f) Iteración 6.

El algoritmo de agrupamiento permitió obtener solamente dos clases, una para cada objeto geográfico, ya que la compresión semántica con las características de intensidad o intensidad promedio generaba más regiones y los resultados eran invariantes después de la sexta iteración. Sin embargo, no en todos los casos se obtuvieron regiones acorde al resultado buscado; en algunos, la geoimagen agrupada preservaba los objetos geoespaciales originales, mientras que en otras, el algoritmo de agrupamiento proporcionaba una geoimagen clasificada la cual no tenía correspondencia alguna con la imagen original. Como ejemplo de esto último, considérese la siguiente figura que muestra la geoimagen que se obtuvo después de la sexta iteración, haciendo simplificación de la geoimagen de intensidad promedio con un umbral de similitud igual a 25, en la Figura 5–61(b) aparece el resultado después de aplicar el algoritmo de agrupamiento  $k$ -medias.

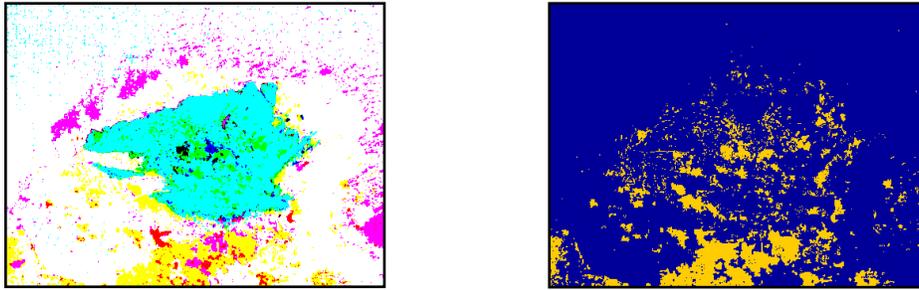


Figura 5–61. Resultados de la simplificación de la geoimagen de intensidad promedio; (a) Geoimagen de salida del algoritmo de compresión semántica; (b) Resultado del algoritmo de agrupamiento.

Como se comentó, aunque es posible distinguir la parte de la isla en la geoimagen simplificada, ver Figura 5–61(a), la imagen que se genera después de aplicar el algoritmo de agrupamiento no contiene esta región, Figura 5–61(b). Otro inconveniente que tiene el usar una técnica de agrupamiento radica en que se pueden generar regiones parásitas; siendo necesario, aplicar un algoritmo de podado. En la Figura 5–62(a) se muestra la geoimagen original agrupada, elegida de la compresión semántica basada en color, mientras que en la Figura 5–62(b) aparece el resultado del algoritmo de poda con un umbral de eliminación de regiones parásitas, igual a 50.

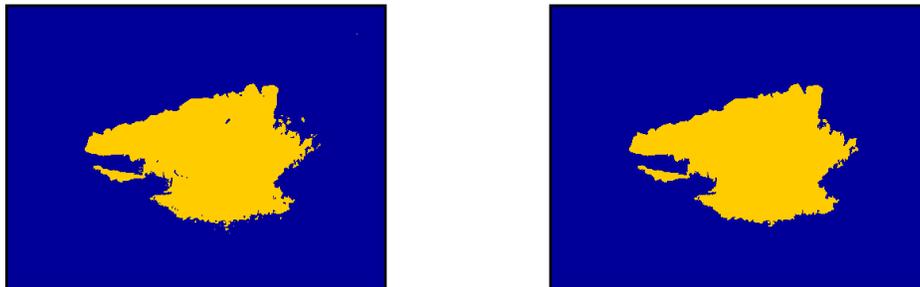


Figura 5–62. Eliminación de regiones parásitas de la geoimagen agrupada; (a) Imagen original; (b) Resultado del algoritmo de poda.

La geoimagen de la Figura 5–62(b) es semánticamente muy similar al resultado que se obtuvo con la compresión semántica orientada a objetos. Es importante la frase *semánticamente similar*, ya que desde el punto de vista del procesamiento digital de imágenes, los resultados son diferentes, pero la metodología de descomposición semántica no genera como resultado final una imagen, sino un conjunto de etiquetas especializadas. En consecuencia, lo que se debe comparar es el conjunto de etiquetas especializadas, es decir, la semántica de las imágenes. Las etiquetas que se obtuvieron en la sección 5.5.1 fueron “suelo o mar, isla, otros” mientras que para la geoimagen de

la Figura 5–62(b) sería “suelo o mar, isla”. Como los objetos denotados como *otros* pueden ser objetos geoespaciales muy pequeños, o bien, forman parte de otro objeto geográfico, la similitud semántica de ambas geoimágenes podría ser muy alta, o incluso, idéntica para algún experto.

### 5.7.5 Caso de estudio 2: Geoimagen de una isla, ejemplo 2.

Con el segundo caso de estudio no fue posible llegar a una simplificación adecuada usando únicamente la información de color. Las pruebas realizadas consistieron en simplificar la geoimagen de intensidad e intensidad promedio con seis umbrales de similitud diferentes: 10, 15, 20, 25, 30 y 35, usando el algoritmo de compresión semántica en modo iterativo con un total de tres iteraciones. Asimismo se aplicó el algoritmo de agrupamiento c-medias difuso a cada geoimagen obtenida en cada iteración. En las siguientes dos figuras se muestran, primeramente, la simplificación con un umbral de similitud igual a 15 de la intensidad promedio (Figura 5–63), mientras que en la Figura 5–64 aparecen los resultados del agrupamiento c-medias difuso. Estas geoimágenes fueran las que se acercaron más al resultado buscado, se logró hacer una delimitación aproximada de la región que delimita el suelo de la isla; sin embargo existen muchos huecos, principalmente en las zonas de la versión original, donde hay sombras o poca iluminación, lo que hace a la geoimagen visualmente muy compleja. Es importante señalar que esta geoimagen no satisface completamente los requisitos del dominio de problema planteados en el Capítulo 4, ya que las características de bajo nivel de los objetos geográficos no están preservadas completamente, ocasionado por el ángulo de adquisición de la geoimagen (foto aérea a baja altura) que provoca la presencia de muchas sombras.

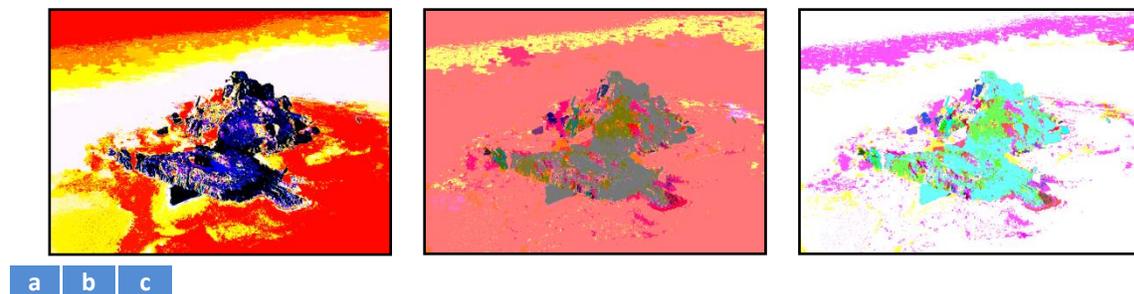


Figura 5–63. Resultados de la compresión semántica de la geoimagen de intensidad promedio, usando un umbral de similitud igual a 15; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.

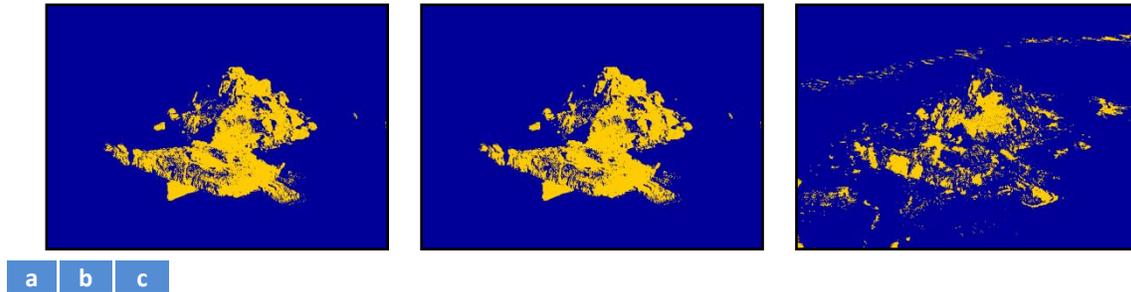


Figura 5–64. Resultados del agrupamiento *c*-medias difuso de las geoimágenes obtenidas en la compresión semántica de la Figura 5–63; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.

### 5.7.6 Caso de estudio 3: Geoimagen de los Grandes Lagos

En esta geoimagen, también se aplicó el algoritmo de compresión semántica simplificando únicamente la intensidad e intensidad promedio. Se usaron los siguientes umbrales de compresión semántica: 05, 10, 15, 20, 25 y 30; aplicando sobre cada geoimagen resultado, el algoritmo de agrupamiento *k*-medias. Para este caso de estudio, si fue posible obtener las regiones de los objetos geográficos. En la Figura 5–65 se muestra el resultado de la compresión semántica que estuvo más acorde con los objetos geográficos originales, aplicando compresión semántica en la geoimagen de intensidad con un umbral de similitud igual a 10. La Figura 5–66 contiene los resultados después de aplicar un agrupamiento *k*-medias.

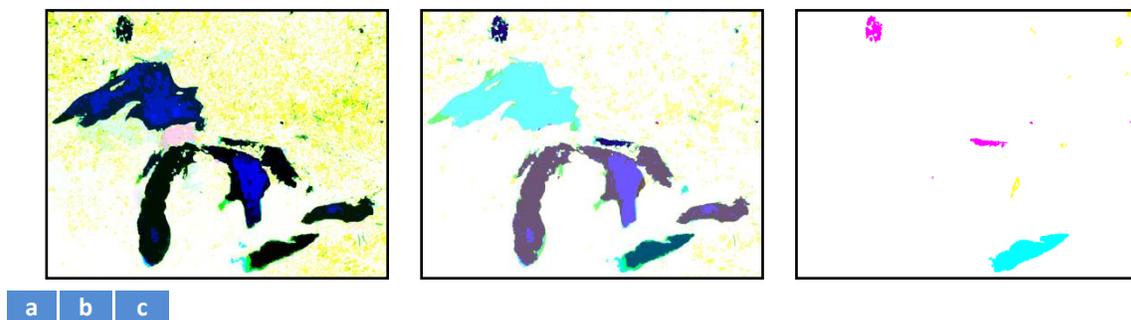


Figura 5–65. Resultados de la compresión semántica de la geoimagen de intensidad, usando un umbral de similitud igual a 10; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.



Figura 5–66. Resultados del agrupamiento  $k$ -medias de las geoimágenes obtenidas en la compresión semántica de la Figura 5–65; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.

Es importante mencionar que observando los resultados del agrupamiento, la mejor geoimagen es la obtenida después de la primera iteración, ya que en las agrupaciones de las iteraciones posteriores se pierden los cuerpos de agua, es decir, ocurre una subclasificación de las regiones.

#### 5.7.7 Caso de estudio 4: Geoimagen de la República Mexicana

Con la geoimagen de la República Mexicana no fue posible hacer una simplificación de los tres objetos geográficos más importantes: dos cuerpos de agua y la zona de suelo. Los resultados más cercanos a la simplificación buscada aparecen a continuación. Al igual que en las explicaciones previas, la primera figura corresponde con los mejores resultados del algoritmo de compresión semántica, mientras que la segunda contiene los resultados aplicando un agrupamiento. Para esta geoimagen se usó el algoritmo de agrupamiento ISODATA. Los diferentes umbrales de similitud empleados con este caso de estudio fueron: 12.5, 15, 17.5, 20, 22.5 y 25, haciendo simplificación de la intensidad e intensidad promedio, con un modo iterativo de tres iteraciones.

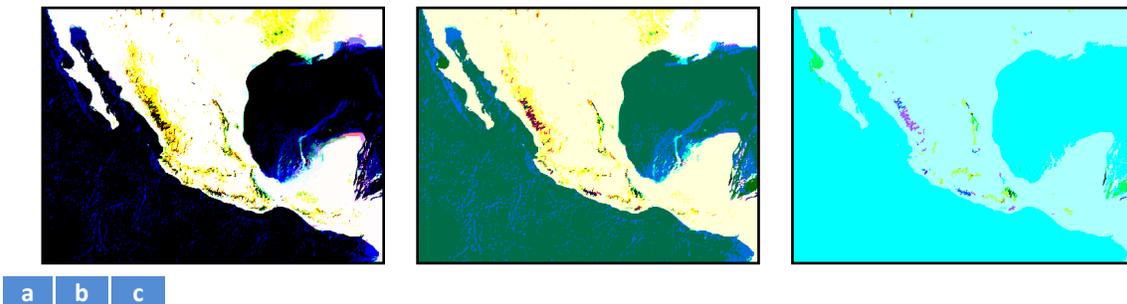


Figura 5–67. Resultados de la compresión semántica de la geoimagen de intensidad, usando un umbral de similitud igual a 20; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.



Figura 5–68. Resultados del agrupamiento ISODATA de las geoimágenes obtenidas en la compresión semántica de la Figura 5–67; (a) Iteración 1; (b) Iteración 2; (c) Iteración 3.

### 5.7.8 Uso de preprocesamiento

En el PDI, la mayoría de las metodologías incorporan una etapa de preprocesamiento, con el objeto de reducir las posibilidades de error en los resultados de un algoritmo dado. Por lo anterior, puede surgir la inquietud respecto a que sucede cuando se aplica un preprocesamiento previo a una compresión semántica basada en color: la respuesta es que solamente se requiere una menor cantidad de iteraciones, obteniéndose los mismos resultados.

La elección de los métodos de preprocesamiento en el dominio espacial o en el dominio de la frecuencia es un proceso subjetivo, ya que depende del conocimiento y experiencia de cada persona, así como de las características que presenta una imagen dada. El efecto que tiene el aplicar adecuadamente una etapa de preprocesamiento es lograr una convergencia más rápida del algoritmo de compresión semántica. Por ejemplo, se mostrarán los efectos de un algoritmo de descomposición en componentes aplicado a la geoimagen del caso de estudio 1. En la Figura 5–69 se muestran los resultados de este algoritmo de preprocesamiento.

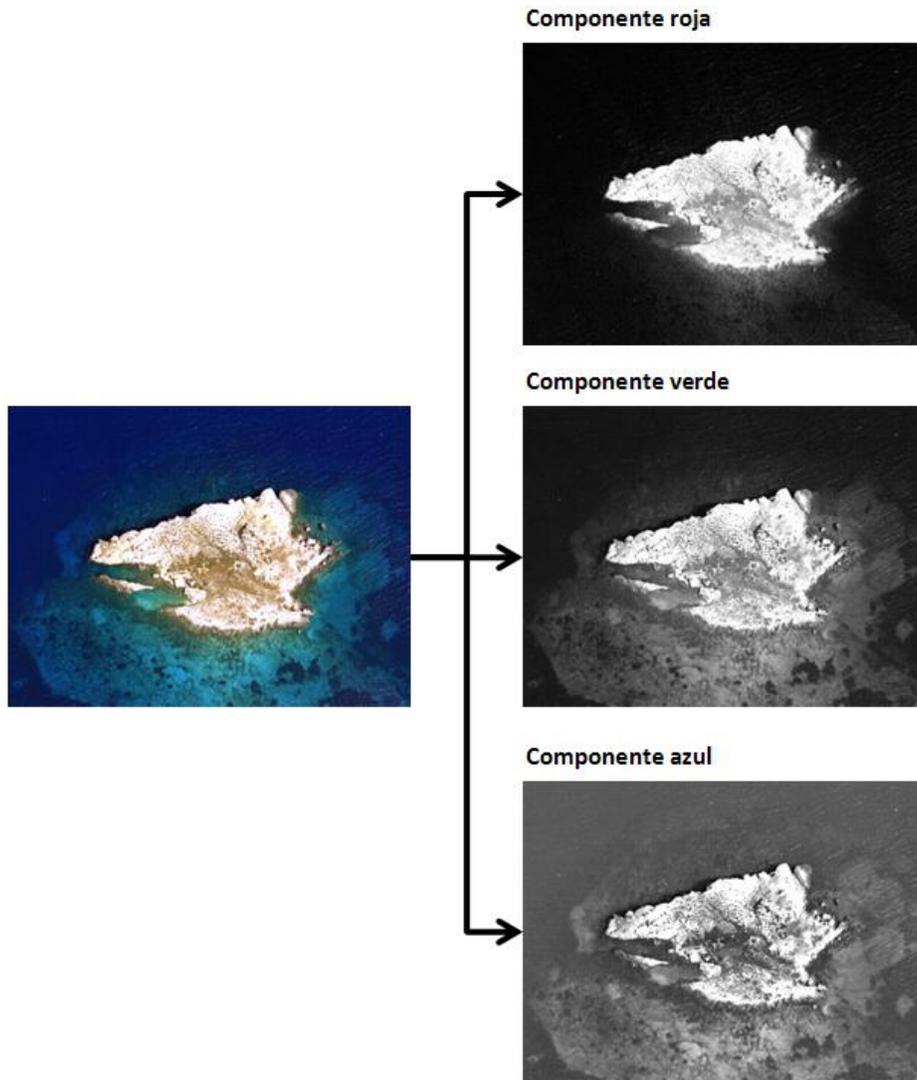
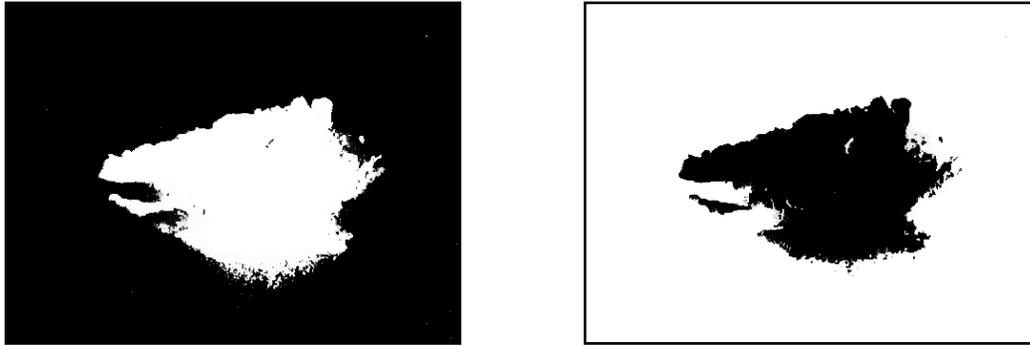


Figura 5–69. Descomposición en componentes de color de la geoimagen del caso de estudio 1.

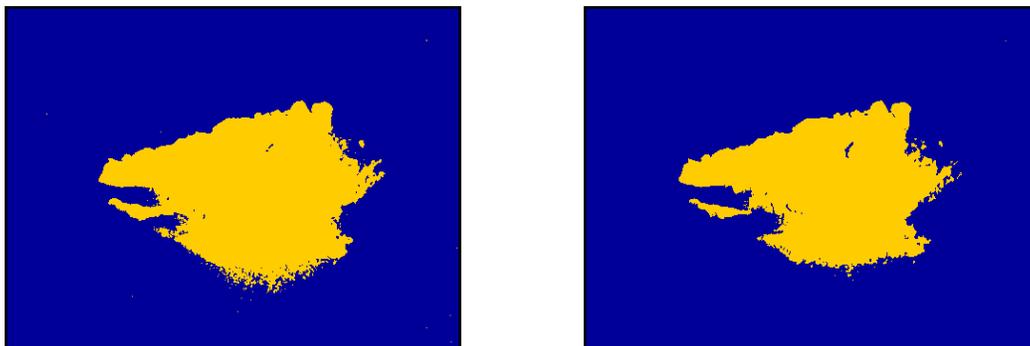
Haciendo una comparación de las tres geoimágenes que se obtienen, la que describe la componente roja es donde se obtiene una mejor separación entre los dos cuerpos geográficos, por lo que será la utilizada en el siguiente punto. Se aplicará el algoritmo de compresión semántica usando el umbral de similitud que proporcionó mejores resultados anteriormente, es decir,  $k_{cs} = 30$ . Aplicando una sola iteración, se obtienen las geoimágenes de intensidad e intensidad promedio que aparecen en la Figura 5–70.



a b

Figura 5–70. Resultados del algoritmo de compresión semántica; (a) Geoimagen de intensidad; (b) Geoimagen de intensidad promedio.

Se procede a realizar la clasificación de la geoimagen de intensidad promedio mediante el algoritmo  $k$ -medias, obteniendo como resultado la Figura 5–71.



a b

Figura 5–71. Resultados de la clasificación usando  $k$ -medias; (a) Geoimagen de intensidad clasificada; (b) Geoimagen de intensidad promedio clasificada.

Por último es necesario hacer una comparación entre la geoimagen clasificada de la característica de intensidad promedio con  $k_{CS} = 30$ , donde no se usó preprocesamiento, con la geoimagen que se acaba de obtener en la Figura 5–71(b) que contó con un preprocesamiento. En la Figura 5–72 puede consultarse el resultado de la sustracción de intensidades de color; no se hace una reducción de la misma para poder apreciar los píxeles de diferencia. En conclusión, el uso de un algoritmo de preprocesamiento no alteró significativamente el resultado del algoritmo de compresión semántica, pero sí auxilió para que este algoritmo convergiera más rápidamente, ya que solamente fue necesario aplicar una sola iteración del mismo.

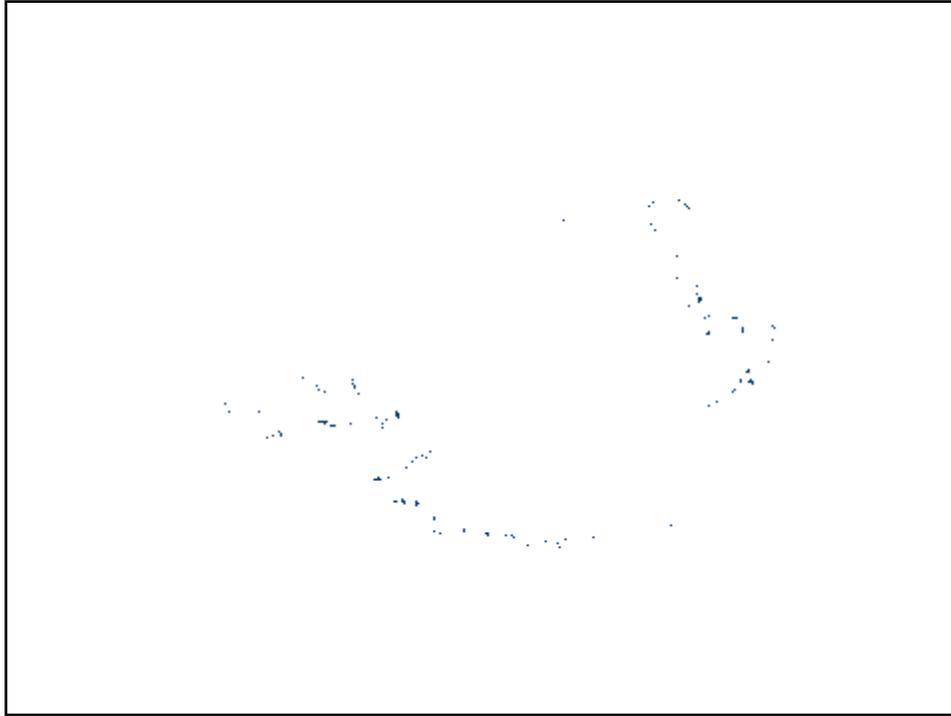


Figura 5–72. Diferencia entre las dos geoimágenes clasificadas.

Sin embargo, es importante resaltar que los algoritmos de preprocesamiento tienen que ser aplicados de forma cautelosa. Como se mencionó, es un proceso subjetivo y se requiere que el usuario esté familiarizado con estas técnicas, ya que puede ser contraproducente y producirse resultados muy divergentes a los que se obtendrían sin una etapa de preprocesamiento, por lo que el usuario interpretaría que el algoritmo de compresión semántica no funciona adecuadamente, cuando en realidad, es el algoritmo de preprocesamiento elegido el responsable de las repercusiones que puede tener sobre los resultados que genera el algoritmo de compresión semántica.

## 5.8 Generación de geoimágenes invariantes

El algoritmo de compresión semántica en modo iterativo produce una geoimagen invariante cuando más a la sexta o séptima iteración, independientemente de la cadena de compresión semántica empleada. En la Figura 5–73 se muestra una gráfica del número de segmentos que se obtuvieron en cada iteración del algoritmo de compresión semántica para una de las cadenas de compresión semántica que fueron usadas en esta sección de pruebas.

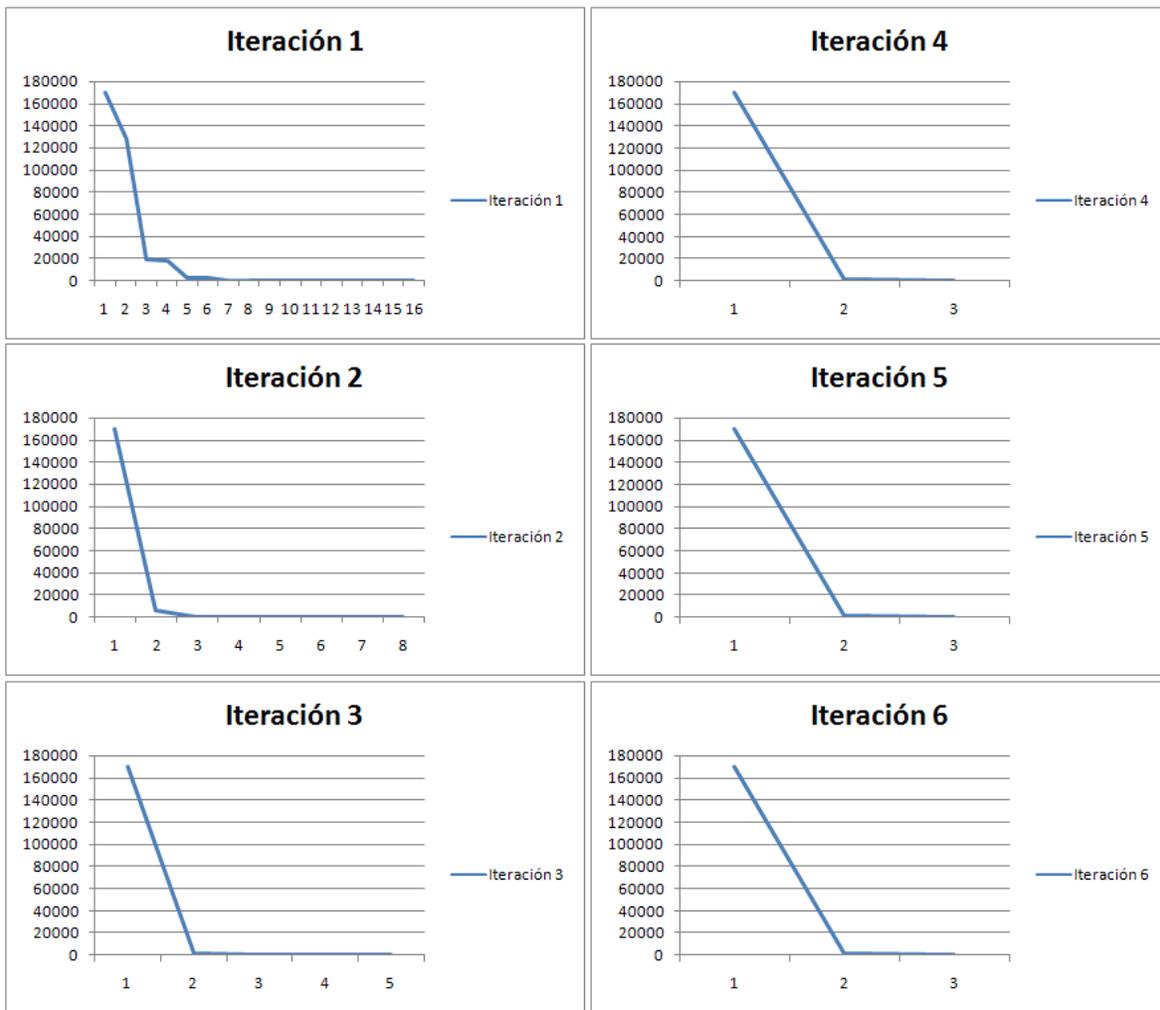


Figura 5-73. Número de segmentos de cada iteración de una cadena de compresión semántica.

En cada iteración, el algoritmo de compresión semántica inicia con la generación del árbol dinámico para realizar un ciclo que consiste en fusionar segmentos hasta el punto en que no se realizan más fusiones, con las condiciones que fueron expuestas en el capítulo 4. De la figura anterior, se puede observar que en la primera iteración, el número de ciclos que requiere el algoritmo es cercano a dieciséis, pero este número de ciclos decae rápidamente en la segunda iteración, lo cual se cumple en las iteraciones posteriores. Observando la iteración número cuatro y posteriores, el número de ciclos es invariante (igual a 3) lo cual implica que ya no se realizan

fusiones adicionales entre segmentos<sup>38</sup>. En conclusión la fusión de segmentos se produce rápidamente en las primeras iteraciones del algoritmo de compresión semántica.

Lo anterior es un aspecto muy importante del algoritmo de compresión semántica ya que son muy escasos los algoritmos de procesamiento de información geoespacial que tienen un comportamiento similar, independientemente de su propósito.

## 5.9 Comentarios y observaciones

Los resultados obtenidos en este capítulo permitieron comprobar que el uso del color, la geometría y la topología de los segmentos es fundamental para poder hacer una correcta simplificación de geoimágenes. El algoritmo de compresión semántica propuesto no requiere de una etapa de preprocesamiento ni de posprocesamiento, ya que al usar el conjunto completo de parámetros y características se van realizando fusiones de los segmentos con base en un umbral de similitud. Esto es, el algoritmo de compresión semántica posee flexibilidad, ya que todos estos algoritmos son realizados intrínsecamente por la compresión semántica. Al usar el algoritmo de compresión semántica en modo iterativo se obtiene mayor nivel de simplificación al terminar una nueva iteración.

La metodología de descomposición semántica permitió llegar a un conjunto de etiquetas especializadas, partiendo del etiquetado en temáticas proporcionado por el algoritmo de reconocimiento. Considerando las propiedades geométricas y topológicas de los objetos geográficos descritos en una ontología de aplicación, fue posible llegar a asignar etiquetas en todos los casos de estudio analizados. Obviamente, para poder llegar a un mayor nivel de descomposición, se requiere de más conocimiento, pero todo esto puede ser representado por medio de conceptos, relaciones y restricciones en una ontología, con lo que la metodología puede ser empleada en amplias áreas mediante el uso de ontologías particulares (de tarea).

---

<sup>38</sup> Recuérdese que el primer ciclo siempre consiste en asociar los píxeles con la misma intensidad de color, ya que al generar el árbol dinámico, el número de segmentos es igual al número de píxeles en la geoimagen.

En el caso de estudio 1 de la isla, se pudo apreciar que el programa eCognition realiza un buen agrupamiento de objetos con base en la definición de parámetros cuantitativos de los segmentos obtenidos con la segmentación multi-resolución, lo cual resulta difícil por la habilidad que se exige del usuario para obtener resultados satisfactorios. Para acertarle al valor de 0.3 que proporciona un buen agrupamiento de las regiones de la isla, se requieren muchas pruebas, pues se observó que al variar muy poco este parámetro los resultados fueron muy distintos.

La empresa propietaria de eCognition hace énfasis en que el programa cuenta con una clasificación basada en conocimiento, pero surge la duda si una isla puede definirse como las regiones que se obtienen de una segmentación multi-resolución con un radio en componente roja mayor o igual a 0.3.

Desde nuestra perspectiva, esto no es conocimiento, una persona dirá que una isla es una región de tierra rodeada por agua, lo cual coincide con la definición que se puede encontrar en un diccionario. Esta forma de representar conocimiento, es como se realiza en la metodología de descomposición semántica a través de una ontología de aplicación del dominio geográfico.

En contraste, nuestra metodología está orientada a emular los conocimientos de un experto que percibe conceptualmente (no numéricamente) el mundo geográfico como un conjunto de objetos geoespaciales etiquetados relacionados entre sí. Añadir más conocimientos a la metodología de descomposición semántica se simplifica notablemente, ya que la representación de los objetos geoespaciales en una ontología de aplicación por medio de conceptos, relaciones y restricciones, coincide con la forma en que una persona percibe los objetos geográficos.

Aunque el algoritmo de compresión semántica requiere tres parámetros, el conjunto (no los valores) que se usan para cualquier geoimagen es el mismo, logaron con esto el propósito planteado de definir una metodología general de descomposición semántica. Por otra parte, eCognition requiere parámetros muy diferentes, por ejemplo, en la clasificación basada en conocimiento se usaron 4 parámetros, mientras que acorde con uno de los artículos analizados en el estado del arte, se necesitan más de 20 parámetros para hacer la clasificación en el caso de estudio de dicha publicación.

Si bien es cierto que existen algoritmos para procesar información geoespacial, éstos no están enfocados a trabajar de forma general, o dicho de otra forma, con cualquier tipo de geoimagen, casi siempre analizan un subconjunto específico del dominio de objetos geográficos, por ejemplo: extracción de zonas de cultivo en imágenes de alta resolución, diferenciación entre zonas de nubes y nieve volcánica, etc. En consecuencia, lo que se permite solucionar un problema no se puede aplicar con otros casos o tipos de geoimágenes.

Cuando se hace una simplificación basada en color junto con un algoritmo de agrupamiento y de eliminación de regiones parásitas, se pueden extraer los objetos geográficos en geoimágenes visualmente no tan complejas, específicamente cuando los patrones de color de cada objeto geográfico es distinto. Sin embargo, en geoimágenes de alta complejidad, el enfoque tradicional no es suficiente para describir las regiones de los objetos geoespaciales. El uso de una etapa de preprocesamiento conduce a los mismos resultados, permitiendo únicamente que el algoritmo de compresión requiera menos iteraciones.

## **Capítulo 6**

### **Conclusiones**

#### *Resumen*

En este capítulo se muestran las conclusiones científicas del presente trabajo, así como las posibles líneas de investigación a seguir como trabajo futuro.



## 6.1 Introducción

Nuestro sistema visual humano realiza el reconocimiento de objetos de una forma muy sencilla y con gran precisión. Desafortunadamente, todavía no existe una teoría demostrada que haya descifrado completamente el funcionamiento de este proceso. Por ejemplo, cuando reconocemos a una persona, aunque su rostro no sea percibido completamente por nuestros sensores visuales (por malas condiciones de iluminación, porque se encuentre a gran distancia, porque lo observamos de perfil, entre otros) es posible que lleguemos a reconocer a dicha persona sin ningún problema, inclusive después de un largo periodo de tiempo y cambios significativos en sus rasgos físicos. Existen muchos factores que se involucran en este proceso, entre los que destacan el conocimiento *a priori* y la inteligencia.

El objetivo de todas las metodologías, teorías y algoritmos analizados en esta tesis (incluyendo nuestro trabajo) persiguen el tratar de simular este proceso mediante sistemas de cómputo. Aunque existen técnicas como la función de calidad de una imagen estudiada en el Capítulo 2, la calificación de una técnica depende todavía de la evaluación de un individuo.

A continuación, se enlistarán las conclusiones de esta tesis y de igual manera se plantean un conjunto de posibles líneas de investigación que se pueden seguir como trabajo futuro.

## 6.2 Conclusiones

Los algoritmos orientados a pixeles se basan en tomar en cuenta una o muy pocas características de bajo nivel de la imagen a procesar, lo cual no ha aportado avances significativos en la resolución de este problema. Las publicaciones que siguen este enfoque dan como resultado algoritmos de mayor complejidad tanto en sus definiciones matemáticas como en el tiempo de procesamiento.

Sin lugar a dudas el algoritmo de compresión semántica es una de las principales aportaciones del presente trabajo. El costo por definir un algoritmo general es que el usuario tiene que hacer pruebas del algoritmo con diferentes umbrales de similitud, número de iteraciones y la selección

de la geoimagen que sirva como entrada para cada nueva iteración, pero esto representa un gran avance en el contexto de las metodologías orientadas a objetos, ya que con estos tres parámetros se puede encontrar la cadena de compresión semántica de una geoimagen, sin importar su escala, resolución o complejidad.

La simplificación no implica pérdida de objetos geográficos, ya que el experto puede ejecutar la descomposición semántica de forma iterativa cuantas veces lo requiera, con el objeto de obtener un mayor nivel de descomposición de los objetos geográficos, siempre y cuando los conocimientos estén representados en la ontología de aplicación, y se cuente con un algoritmo de reconocimiento particularizado para cada nivel de descomposición.

Existen ambigüedades que no son posibles de resolver utilizando únicamente los datos (píxeles) de una geoimagen, por lo que el conocimiento a priori y un mecanismo de resolución de ambigüedades son necesarios, lo cual se logra con el empleo de una ontología de aplicación del dominio geográfico.

Todos los objetivos particulares planteados para la presente tesis fueron cumplidos satisfactoriamente:

1. Se definió el concepto de geoimagen como una superclase de imágenes digitales que describen los objetos geográficos.
2. La semántica de una geoimagen quedó expresada como un conjunto de etiquetas en lenguaje natural de los objetos geográficos contenidos en la geoimagen.
3. Se propuso un conjunto de características y parámetros para poder hacer la compresión semántica de una geoimagen.
4. Se definió e implementó un algoritmo de compresión semántica que permite describir con intensidades de color homogéneas a las regiones de cada objeto geográfico.

5. Se propuso una etapa de síntesis para poder obtener etiquetas en lenguaje natural de los objetos geográficos contenidos en una geoimagen.
6. Se definió una ontología de aplicación para la especialización de los objetos geográficos.

Entre las principales aportaciones del presente trabajo, se tienen:

1. Definición de un algoritmo de compresión semántica adaptativo que permite hacer fusiones de segmentos de los objetos geográficos considerando características geométricas de los objetos geográficos, lo cual es novedoso en nuestro campo de investigación.
2. Metodología que requiere pocos parámetros o poco conocimiento *a priori* para poder hacer la descomposición de los objetos geográficos contenidos en una geoimagen. A diferencia de otras metodologías en las que el usuario tiene que proporcionar una gran cantidad de información al sistema para poder hacer una adecuada descomposición.
3. Definición de la semántica de una geoimagen como un conjunto de etiquetas especializadas en lenguaje natural de los objetos representativos de una geoimagen.
4. Definición de dos etapas de etiquetado que permiten especializar la identidad de los objetos geográficos, donde el primer algoritmo de etiquetado puede ser sencillo, ya que solamente busca clasificar los objetos geográficos en temáticas, y el algoritmo de especialización, el que considerando las propiedades geométricas y topológicas de los mismos, hace una asignación de la identidad de los objetos geoespaciales.
5. Un mecanismo para asociar una ontología de dominio geográfico en el problema de la especialización de los objetos geográficos, la cual describe las propiedades geométricas y topológicas de los objetos geoespaciales.

### 6.3 Trabajo futuro

Existen varios problemas que quedan por resolver dentro de esta línea de investigación, los cuales citamos como trabajo futuro:

1. Demostrar que el conjunto de parámetros y características de compresión semántica es el conjunto mínimo necesario para realizar la compresión semántica de geoimágenes.
2. Delimitar los parámetros de compresión semántica para contextos geográficos específicos. Se considera que en contextos delimitados, es posible establecer un rango de umbrales de similitud así como la secuencia de características y parámetros de compresión semántica con los que se pueden extraer correctamente los objetos geográficos de ese contexto.
3. Enriquecimiento del conjunto de temáticas. El conjunto de temáticas que se empleó para la etapa de reconocimiento puede ser extendido para contar con un conjunto más amplio y no solamente realizar un etiquetado general tipo suelo o cuerpo de agua.
4. Reconocimiento de los objetos geográficos mediante otras técnicas. Para hacer visualización de ciertas geoimágenes se pueden emplear técnicas como la de falso color, sin embargo, la representación visual de color para cada objeto geográfico no corresponde con las del objeto real. Se pueden emplear y adaptar otros algoritmos de reconocimiento que trabajen con otra representación en pseudocolor de los objetos geográficos.
5. Extensión de la metodología de descomposición para el trabajo con objetos no únicamente geográficos, como pueden ser imágenes satelitales que incluyen objetos del espacio.
6. Uso de similitud semántica. Con esta información se puede obtener una medida cuantitativa de la similitud del conjunto de etiquetas especializadas que se generan con diferentes cadenas de compresión semántica aplicadas a una misma geoimagen.
7. Consulta de información no aportada por la geoimagen. Como se explicó, existen algunos accidentes geográficos (colinas, acantilados) en los que se requiere información adicional que

no está descrita por las intensidades de color de una geoimagen, como es el caso de la elevación. Contando con este tipo de información podría hacerse una extensión del algoritmo de especialización para poder trabajar con un conjunto más amplio de accidentes geográficos.

8. Extensión de la ontología de aplicación. En este punto se pueden considerar los trabajos publicados relacionados con la conceptualización del dominio geográfico, con el objeto de agregar más conocimientos a la ontología de aplicación.
9. Descomposición semántica iterativa. Se planteó que al usar la descomposición semántica en forma iterativa, es posible obtener mayor nivel de descomposición de los objetos geográficos. Por esta razón, se puede definir un mecanismo de comunicación entre la etapa de análisis y síntesis, para que después de asignar un conjunto de etiquetas a una geoimagen original, cada región extraída sea procesada por el algoritmo de compresión semántica y se lleve a cabo la etapa de síntesis en las nuevas regiones detectadas, empleando procesos de reconocimiento y de especialización más finos que permitan generar etiquetas con mayor nivel de descomposición semántica.



## **Referencias**



**(Adams & Williams, 2003)**

Nicholas J. Adams, Christopher K. I. Williams, Dynamic trees for image modeling, in Image Vision Computing, Vol. 21, No. 10, pp. 865 – 877, 2003.

**(Angulo & Serra, 2003)**

Jesús Angulo & Jean Serra, Mahemathical Morphology in Color Spaces Applied to the Analysis of Cartographic Images, in Proceedings of the Second International Workshop on Semantic Processing of Spatial Data, 2003.

**(Béjar, 1993)**

Javier Béjar, En busca del concepto perdido, Universidad Politècnica de Catalunya, 1993, <http://www-lsi.upc.es/~bejar/reports/LSI-93-12.ps.gz>.

**(Bernaras et al, 1996)**

A. Bernaras, I. Laresgoiti, J. Corera, Building and reusing ontologies for electrical network applications, in Proceedings of the European Conference on Artificial Intelligence, pp. 298 – 302, 1996

**(Bezdeck, 1981)**

J. C. Bezdeck, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.

**(Blum, 1978)**

H. Blum, R. N. Nagel, Shape description using weighted symmetric axis features, Pattern Recognition, Vol. 10, No. 3, pp. 205 – 287, 1978.

**(Blum, 1973)**

H. Blum, Biological shape and visual science (Part 1), Journal of Theoretical Biology, No. 38, No. 2, pp. 205 – 287, 1973.

**(Booch, 1997)**

G. Booch, J. Rumbaugh & and I. Jacobson, The Unified Modeling Language user guide: Addison-Wesley, 1997.

**(Borst, 1997)**

Construction of Engineering Ontologies, Ph.D. Thesis, University of Tweenty, Enschede, NL – Centre for Telematica and Information Technology, 1997.

**(Brady, 1984)**

M. Brady, H. Asada, Smoothed local symmetries and their implementation, International Journal of Robotics Research, Vol. 3, pp. 36 – 61, 1984.

**(Brady, 1983)**

M. Brady, Criteria for representation of shape, in Human and Machine Vision, J. Beck, B. Hope, A. Rosenfeld (Editors), Academic Press, New York, pp. 39 – 84, 1983.

**(Bradshaw, 2000)**

Ben Bradshaw, Semantic Based Image Retrieval: A Probabilistic Approach, ACM Multimedia, pp. 167 – 176, 2000.

**(Byung-Gyu et al., 2003)**

Kim Byung-Gyu, Shim Jae-Ick & Park Dong-Jo, Fast image segmentation based on multi-resolution analysis and wavelets, in Pattern Recognition Letters, Vol. 24, pp. 2995 – 3006, 2003.

**(Calvard, 1978)**

S. Calvard, Picture thresholding using an iterative selection method, in IEEE Trans. on Systems, Man & Cybernet, Vol. 8, No. 8, pp. 629–632, 1978.

**(CCRS, 2007)**

Fundamentals of Remote Sensing, Canada Centre for Remote Sensing, 2007,  
[http://ccrs.nrcan.gc.ca/resource/tutor/fundam/index\\_e.php](http://ccrs.nrcan.gc.ca/resource/tutor/fundam/index_e.php)

**(Chen et al., 2005)**

J. Chen, T. N. Pappas, A. Mojsilovic, & B. E. Rogowitz, Adaptive Perceptual Color-Texture Image Segmentation, in IEEE Transactions on Image Processing, pp. 1524 – 1536, 2005.

**(Chimaera, 2000)**

Chimaera Ontology Environment, 2000  
<http://www.ksl.stanford.edu/software/chimaera>

**(Chuvieco, 1990)**

E. Chuvieco, Fundamentos de la Teledetección Espacial, Colección Monografías y Tratados GER, Ediciones Rialp, S.A., Madrid, pp 205 – 211, 1990.

**(Corcho et al., 2002)**

Oscar Corcho, Mariano Fernández-López & Asunción Gómez-Pérez, Methodologies, tools and languages for building ontologies. Where is the meeting point?, in Data & Knowledge Engineering, Vol. 46, pp. 41 – 64, 2002.

**(Definiens, 2007)**

Definiens Professional 5, Take the Plunge, Guided Tour, Definiens AG, 2007.

**(Depalov et al., 2006)**

Dejan Depalov, Thrasyvoulos Pappas, Dongge Li & Bhavan Gandhi, Perceptually Based Techniques for Semantic Image Classification and Retrieval, in Human Vision and Electronic Imaging XI, pp. 354 – 363, 2006.

**(Din-Yuen et al., 2005)**

Chan Din-Yuen, Lin Chih-Hsueh & Hsieh Wen, Image Segmentation with Fast Wavelet-Based Color Segmentation and Directional Region Growing, in IEEE Transactions Informatics and Systems, Vol. E88-D, No. 10, pp. 2249 – 2259, 2005.

**(Dorado & Izquierdo, 2003)**

Andres Dorado & Ebroul Izquierdo, Semantic Labeling of Images Combining Color, Texture and Keywords, in Proceedings of the International Conference on Image Processing, Vol. III, pp. 9 – 12, 2003.

**(Dudani et al., 1977)**

S. Dudani, K. Breeding & R. McGbee, Aircraft Identification by Moment Invariants, in IEEE Transactions on Computers, Vol. C-26, No. 1, pp. 39 – 45, 1977.

**(Dumains, 1999)**

Dumains S.T., Improving the retrieval of information from external sources, Behavior Research Methods, in Instruments & Computers, Vol. 23, No. 2, pp. 229 – 236, 1999.

**(Dunn, 1973)**

J. C. Dunn, A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, Journal of Cybernetics Vol. 3, pp. 32 – 57.

**(Ecognition, 2007)**

<http://www.definiens.com/>

**(EDCS, 2005)**

ISO/IEC 18025:2005(E), Environmental Data Coding Specification, SEDRIS Standards, Available at online resource in <http://standards.sedris.org/>

**(Everitt, 1974)**

Everitt, B.S., Cluster Analysis, John Wiley & Sons, New York, 1974.

**(Fonseca, 2002)**

Frederico Fonseca, Max Egenhofer, Clodoveu Davis & Gilberto Câmara, Semantic Granularity in Ontology-Driven Geographic Information Systems, in AMAI Annals of Mathematics and Artificial Intelligence – Special Issue on Spatial and Temporal Granularity, Vol. 36 No. 1 y 2, pp. 121 – 151, 2002.

**(Fung & Loe, 1999)**

Cheong Yiu Fung & Kia Fock Loe, Learning Primitive and Scene Semantics of Images of Classification and Retrieval, ACM Multimedia, pp. 9 – 12, 1999.

**(Genesereth & Nilsson, 1987)**

M. R. Genesereth & H. J. Nilsson, Logical Foundation of Artificial Intelligence, Morgan Kaufmann, 1987.

**(Gómez-Allende, 1994)**

Dario Maravall Gomez-Allende, Reconocimiento de formas y vision artificial, Addison-Wesley Iberoamericana, ISBN 0-201-64183-6, 1994.

**(Gonzalez & Woods, 2002)**

Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Second Edition, Prentice Hall, New Jersey, ISBN 0-201-18075-8, 2002.

**(Gruber, 1993)**

T. R. Gruber, A translation approach to portable ontology specification, in Knowledge Acquisition, Vol. 5, pp. 199 – 220, 1993.

**(Guarino et al., 1995)**

N. Guarino, M. Carrara & P. Giaretta, Ontologies and knowledge bases: towards a terminological clarification, in Towards Very Large Knowledge Bases, Knowledge Building and Knowledge Sharing, N. Mars (Editor), IOS Press, Amsterdam, pp. 25 – 32, 1995.

**(Guarino, 1998)**

Nicola Guarino, Formal Ontology and Information Systems, in Proceedings of FOIS'98, pp. 3 – 15, 1998.

**(Güting 1994)**

Ralf Harthmunt Güting, An Introduction to Spatial Database Systems, in Special Issue on Spatial Database Systems, VLDB Journal, Vol. 3, No. 4, 1994.

**(Guzmán et al., 2006a)**

G. Guzmán, R. Quintero, M. Torres and M. Moreno, “Applying a-priori Knowledge for Compressing Digital Elevation Models”, in B. Gabrys, R.J. Howlett and L.C. Jain, Lecture Notes in Artificial Intelligence, Vol. 4251, Springer-Verlag, Berlin Heidelberg, ISSN: 0302-9743, pp. 614-622, 2006.

**(Guzmán et al., 2006b)**

G. Guzmán, S. Levachkine, M. Torres, R. Quintero & M. Moreno, “Un algoritmo general para la descomposición semántica de geoimágenes”, SELPER 2006.

**(Guzmán & Quintero, 2006)**

G. Guzmán & R. Quintero, “Compressing Digital Elevation Models based on Run Length Encoding Approach”, in Special Issue: Semantics in Geoinformatics Applications, Marco Moreno & Serguei Levachkine (Eds.), Research on Computing Science Vol. 25, ISSN: 1870-4069, 2006.

**(Hakimpour & Timpf, 2002)**

Farshad Hakimpour, Sabine Timpf, “A Step Towards Geodata Integration using Formal Ontologies”, in 5th AGILE Conference on Geographic Information Science, 2002.

**(Hanbury, 2001)**

A. Hanbury, Mathematical Morphology in the HLS Colour Space, in Proceedings of 12th BMVC, British Machine Vision Conference, Vol. II, pp. 451 – 460, 2001.

**(Haralick, 1992)**

R. M. Haralick, L. G. Shapiro, Computer and robot vision, Vol 1, Ed. Addison-Wesley, 1992.

**(Huang et al., 1998)**

Jing Huang, S. Ravi Kumar & Ramih Zabih, An Automatic Hierarchical Image Classification Scheme, ACM Multimedia, pp. 219 – 228, 1998.

**(INEGI, 2007)**

Diccionarios de Datos del Instituto Nacional de Estadística, Geografía e Informática, 2007, <http://mapserver.inegi.gob.mx/geografia/espanol/normatividad/diccio/datos.cfm>.

**(Jähne, 2002)**

Bernard Jähne, Digital Image Processing: Concepts, Algorithms, and Scientific Applications, Springer, ISBN 3-540-67754, 2002.

**(Jain & Dubes, 1988)**

Jain, Anil K., Dubes Richard C., Algorithms for Clustering Data, Prentice Hall, ISBN: 0-13-122278-X, 1998.

**(Kittler & Illingworth, 1986)**

J. Kittler, J. Illingworth 1986, Minimum error thresholding, in Pattern Recognition, Vol. 19, No. 1, pp. 41 – 47, 1986.

**(LANDAST Satellite Image Gallery, 2007)**

[http://www.landsat.org/landsat\\_gallery/EP119R37D030100.html](http://www.landsat.org/landsat_gallery/EP119R37D030100.html)

**(Levachkine & Alexandrov, 2003)**

Serguei Levachkine & Victor Alexandrov, Semantic-Mind Analysis and Objected-Oriented Data Integration of Visual Information, in Proceedings of the Second International Workshop on Semantic Processing of Spatial Data, 2003.

**(Levachkine, 2003)**

Serguei Levachkine, Raster to Vector Conversion of Color Cartographic Maps, in Proceedings of 5th International Workshop Graphics Recognition, Recent Advances and Perspectives, GREC 2003, pp. 50 – 62, 2003.

**(Levachkine et al., 2001)**

Serguei Levachkine, Aurelio Velázquez, Victor Alexandrov, Mikhail Kharinov, Semantic Analysis and Recognition of Raster-Scanned Color Cartographic Images, in Proceedings of 3th International Workshop Graphics Recognition, Recent Advances and Perspectives, GREC 2001, pp. 178 – 189, 2001.

**(Lin et al., 2003)**

C. Y. Lin, B. L. Tseng & J.R. Smith, Video collaborative annotation forum: Establishing ground-truth labels on large multimedia datasets, NIST TRECVID, 2003.

**(Lin & Chang, 2001)**

C.Y. Lin & S. F. Chang, SARI: Self Authentication and Recovery Watermarking System, ACM Multimedia, 2001.

**(Lin & Tseng, 2002)**

Ching-Yung Lin & Belle L. Tseng, Segmentation, Classification and Watermarking for Image/Video Semantic Authentication, in IEEE Workshop on Multimedia Signal Processing, pp. 359 – 362, 2002.

**(Liu & Yang, 1994)**

J. Liu, Y. H. Yang, Multiresolution Color Image Segmentation, in IEEE Transactions on Pattern Analysis and Machine Intelligence, No. 16, pp. 689 – 700, 1994.

**(Lu et al., 2000)**

Lu Y., Hu CH., Zhu XQ. Zhang HJ & Yang Q, A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval, in Proceedings of the 8<sup>th</sup> ACM International Conference on Multimedia, pp. 31 – 38, 2000.

**(Martínez, 2001)**

Martínez Trinidad José Francisco, Herramienta para la Estructuración Conceptual de Espacios, Resumen de Tesis Doctoral, Computación y Sistemas, Vol. 4, No. 3, pp. 264 – 269, 2001.

**(Michalski, 1986)**

R. Michalski, R. E. Steep, Conceptual Clustering: Inventing goal oriented classifications of structured objects, in Machine Learning II. Ed., Tioga, Palo Alto, California, 1986.

**(Michalski, 1983)**

R. Michalski, R. E. Steep, Learning from observation: Conceptual Clustering, in Machine Learning II. Ed., Tioga, Palo Alto, California, 1983.

**(Meyer, 1977)**

F. Meyer, Contrast Features Extraction, Cuantitative Analysis of Microstructures in Materials Science, Biology and Medicine, pp. 374 – 380, 1977.

**(Mueller et al., 2004)**

Marina Mueller, Karl Segl & Hermann Kaufmann, Edge- and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery, in Pattern Recognition, Vol. 37, No. 8, pp. 1619 – 1628, 2004.

**(Noy & McGuinness, 2001)**

Natalya F. Noy & Deborah L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, 2001.

**(Neches et al., 1991)**

R. Neches, R. E. Fikes, T. Finin, T.R. Gruber, T. Senator & W.R. Swartout, Enabling technology for knowledge sharing, AI Magazine, Vol. 3, No. 12, pp. 36 – 56, 1991.

**(Ontolingua, 1997)**

Ontolingua System Reference Manual,

<http://www-ksl-svc.stanford.edu:5915/doc/frame-editor/index.html>

**(Otsu, 1979)**

N. Otsu, A threshold selection method from grey-level histograms, in IEEE Trans. On Systems, Man & Cybernet, Vol. 9, No. 1, pp. 62 – 66, 1979.

**(Pappas, 1992)**

T. N. Pappas, An Adaptive Clustering Algorithm for Image Segmentation, in IEEE Transactions on Signal Processing, pp. 901 – 914, 1992.

**(Phillips, 1999)**

Dwayne Phillips, Image Processing in C, R & D Publications, ISBN 0-13-104548-2, 1999.

**(Plataniotis & Venetsanopoulos, 2000)**

Konstantinos N. Plataniotis & Anastasios N. Venetsanopoulos, Color Image Processing and Applications, Springer, ISBN: 3-540-66953-1, 2000.

**(Pratt, 2001)**

Pratt William, Digital Image Processing PIKS Inside, Third Edition, John Wiley & Sons, ISBN 0-471-22132-5, 2001.

**(Protege, 2007)**

Protégé Software, Stanford Medical Informatics,  
<http://protege.stanford.edu/download/release/full/>

**(Real Academia, 2007)**

Diccionario de la Real Academia de la Lengua Española, 2007, <http://buscon.rae.es/drae/>.

**(Rumbaugh et al., 1991)**

J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy & W. Lorenzen, Object oriented modeling and design. Englewood Cliffs, New Jersey, Prentice Hall, 1991.

**(Salitchev, 1979)**

Salitchev, K.A. Mapas geográficos y cartografía. Editorial Pueblo y Educación. La Habana, Cuba. 1979.

**(Samet, 1995)**

Hanan Samet, Spatial Data Structures, in Modern Database Systems: The Object Model, Interoperability, and Beyond, Addison Wesley, pp. 361 – 385, 1995.

**(Space Imaging, 2002)**

Space Imaging, Top Ten IKONOS Satellite Images for 2002,  
[http://www.spaceimaging.com/gallery/top10\\_2002/](http://www.spaceimaging.com/gallery/top10_2002/)

**(Storkey Amos, 2007)**

<http://homepages.inf.ed.ac.uk/amos/dynamic.html>

**(Studer et al., 1998)**

R. Studer, V.R. Benjamins, D. Fensel, Knowledge engineering: principles and methods, in Data and Knowledge Engineering, Vol. 25, pp. 161 – 197, 1998.

**(Swartout et al., 1997)**

B. Swartout, P. Ramesh, K. Knight & T. Russ, Toward Distributed Use of Large-Scale Ontologies, in AAAI Symposium on Ontological Engineering, 1997.

**(Tamura, & Yamawaki, 1978)** H. Tamura, S. Mori & T. Yamawaki, Texture Features Corresponding to Visual Perception, in IEEE Transactions on Systems, Man, and Cybernetics, Vol. 8, No. 6, 1978.

**(TeleDet, 2007)**

TeleDet, 2007, <http://www.teledet.com.uy>.

**(Torres, 2007)**

Miguel Torres, Tesis Doctoral, "Representación ontológica basada en descriptores semánticos aplicada a objetos geográficos", Centro de Investigación en Computación, Noviembre 2007.

**(Uschold & Jasper, 1999)**

M. Uschold, R. Jasper, A Framework for Understanding and Classifying Ontology Applications, in Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods, 1999.

**(Uschold & Gruninger, 1996)**

M. Uschold & M. Gruninger, Ontologies: Principles, Methods and Applications, in Knowledge Engineering Review, Vol. 11, No. 2, 1996.

**(WebSeek, 2007)**

WebSeek: A Content-Based Image and Video Search and Catalog Tool for the Web, 2007, <http://persia.ee.columbia.edu:8008/>.

**(Wikipedia, 2007)**

Wikipedia, The Free Library, 2007, <http://es.wikipedia.org/wiki/Portada>

**(WordReference, 2007)**

Diccionario Word Reference, 2007, <http://www.wordreference.com/es/index.htm>.

**(Zhang & Fu 2005)**

Tong-Zhen Zhang & Yong-Gang Fu, An Image Semantic Retrieval System Design and Realization, in Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, pp. 5284 – 5289, Agosto 2005.

**(Zhong et al., 2002)**

Jiwei Zhong, Haiping Zhu, Jianming Li & Yong Yu, Conceptual Graph Matching for Semantic Search, in Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces, pp. 192-196, 2002

**(Zhuang et al., 1999)**

Y. T. Zhuang, Y. H. Pan & Y. Rui, Using Semantic Association to Support Content-Based Video Queries, in Journal of Computer Reseca & Development, Vol. 36, No. 5, pp. 613 – 616, 1999.

**(Zou, 2001)**

J. J. Zou, H.H. Chang, Hong Yan, "Shape skeletonization by identifying discrete local symmetries", Pattern Recognition, Vol. 34, No.10 pp. 1895 – 1905, 2001.



## **Anexo A**

### **Resultados del enfoque clásico**



## A.1 Resultados de la etapa de análisis con enfoque clásico

En este apéndice se encuentran todas las geoimágenes que se obtuvieron con los cuatro casos de estudio haciendo un análisis basado únicamente en el color de los objetos geográficos. Recuérdese que este tipo de análisis consiste en hacer una simplificación con las características de intensidad e intensidad promedio, seguido de un algoritmo de agrupamiento como el  $k$ -medias, el  $c$ -medias difuso o el ISODATA, y de forma opcional un algoritmo de podado.

El objetivo de este apéndice es que el lector pueda constatar que solamente en dos de los casos se pudieron obtener resultados satisfactorios (casos de estudio 1 y 3) mientras que en los otros casos de estudio (2 y 4) no fue posible obtener correctamente las regiones de los objetos geográficos. Para cada geoimagen original, se mostrarán los resultados de cada iteración del algoritmo de compresión semántica en modo iterativo. En los casos aquí ejemplificados, se generaron seis cadenas de compresión semántica diferentes de intensidad promedio y seis para la característica de intensidad, es decir, se usaron seis valores diferentes del umbral de similitud. Para cada cadena de compresión semántica, se mantuvo constante el umbral en todas las iteraciones. Aunque se hicieron pruebas adicionales variando los parámetros para la generación de la cadena de compresión semántica, no se obtuvieron mejores resultados.

En la siguiente tabla aparecen todas las cadenas de compresión semántica empleadas con cada caso de estudio.

Tabla A-1. Cadenas de compresión semántica usadas en el análisis con enfoque clásico.

Caso de estudio 1
Cadenas de compresión semántica de intensidad promedio
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 15), (I_1, ABS, 15), (I_2, ABS, 15), (I_3, ABS, 15), (I_4, ABS, 15), (I_5, ABS, 15)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 20), (I_1, ABS, 20), (I_2, ABS, 20), (I_3, ABS, 20), (I_4, ABS, 20), (I_5, ABS, 20)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 25), (I_1, ABS, 25), (I_2, ABS, 25), (I_3, ABS, 25), (I_4, ABS, 25), (I_5, ABS, 25)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 30), (I_1, ABS, 30), (I_2, ABS, 30), (I_3, ABS, 30), (I_4, ABS, 30), (I_5, ABS, 30)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 35), (I_1, ABS, 35), (I_2, ABS, 35), (I_3, ABS, 35), (I_4, ABS, 35), (I_5, ABS, 35)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 40), (I_1, ABS, 40), (I_2, ABS, 40), (I_3, ABS, 40), (I_4, ABS, 40), (I_5, ABS, 40)\}$

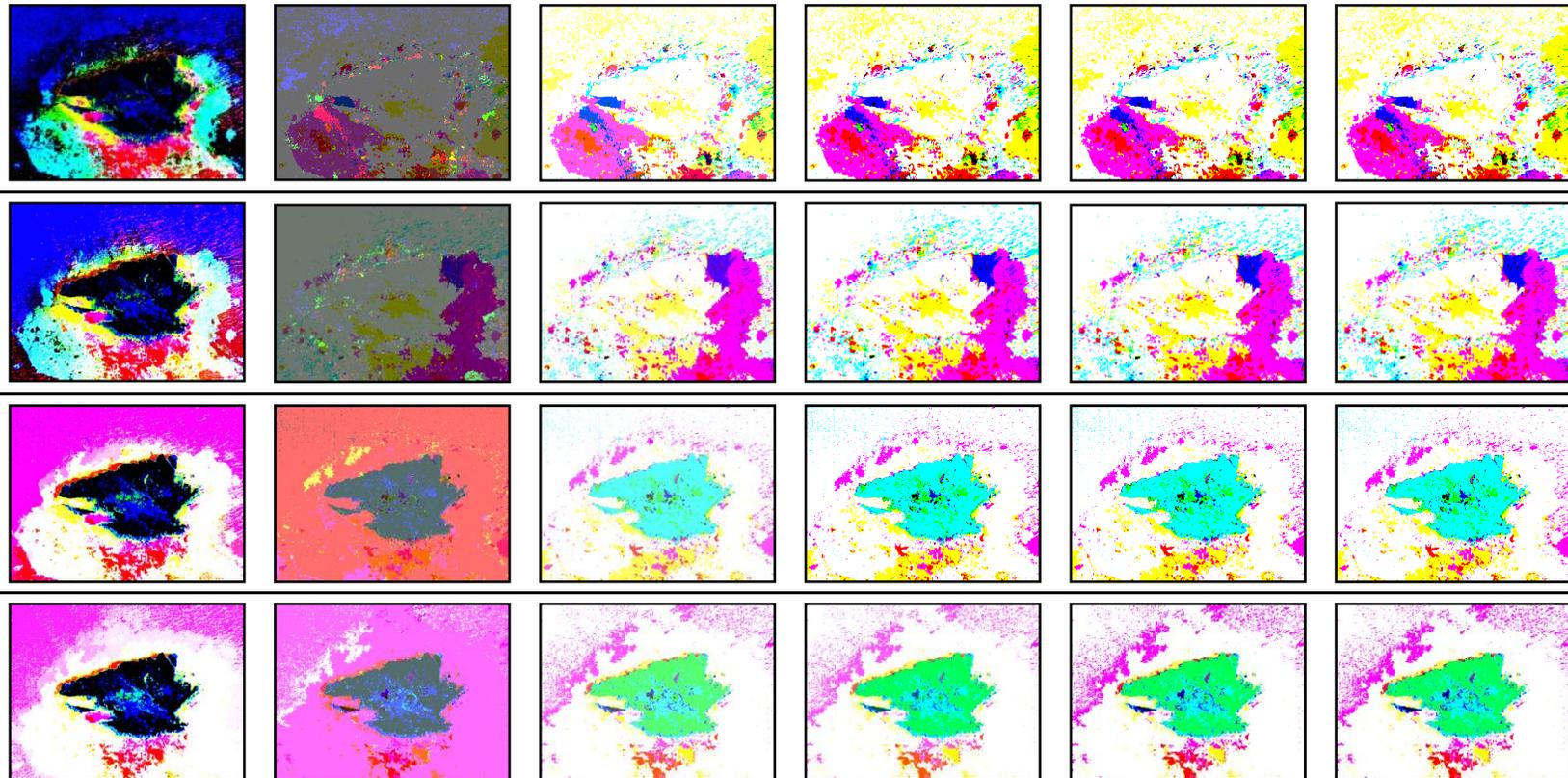
Tabla A-1. (Cont...) Cadenas de compresión semántica usadas en el análisis con enfoque clásico.

Cadenas de compresión semántica de intensidad	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 15), (I_1, intensidad, 15), (I_2, intensidad, 15), (I_3, intensidad, 15), (I_4, intensidad, 15), (I_5, intensidad, 15)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 20), (I_1, intensidad, 20), (I_2, intensidad, 20), (I_3, intensidad, 20), (I_4, intensidad, 20), (I_5, intensidad, 20)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 25), (I_1, intensidad, 25), (I_2, intensidad, 25), (I_3, intensidad, 25), (I_4, intensidad, 25), (I_5, intensidad, 25)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 30), (I_1, intensidad, 30), (I_2, intensidad, 30), (I_3, intensidad, 30), (I_4, intensidad, 30), (I_5, intensidad, 30)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 35), (I_1, intensidad, 35), (I_2, intensidad, 35), (I_3, intensidad, 35), (I_4, intensidad, 35), (I_5, intensidad, 35)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 40), (I_1, intensidad, 40), (I_2, intensidad, 40), (I_3, intensidad, 40), (I_4, intensidad, 40), (I_5, intensidad, 40)\}$	
Caso de estudio 2	
Cadenas de compresión semántica de intensidad promedio	
$C_{cs}(I_o, I_{ABS}) = \{(I_o, 10), (I_1, ABS, 10), (I_2, ABS, 10)\}$	
$C_{cs}(I_o, I_{ABS}) = \{(I_o, 15), (I_1, ABS, 15), (I_2, ABS, 15)\}$	
$C_{cs}(I_o, I_{ABS}) = \{(I_o, 20), (I_1, ABS, 20), (I_2, ABS, 20)\}$	
$C_{cs}(I_o, I_{ABS}) = \{(I_o, 25), (I_1, ABS, 25), (I_2, ABS, 25)\}$	
$C_{cs}(I_o, I_{ABS}) = \{(I_o, 30), (I_1, ABS, 30), (I_2, ABS, 30)\}$	
$C_{cs}(I_o, I_{ABS}) = \{(I_o, 35), (I_1, ABS, 35), (I_2, ABS, 35)\}$	
Cadenas de compresión semántica de intensidad	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 10), (I_1, intensidad, 10), (I_2, intensidad, 10)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 15), (I_1, intensidad, 15), (I_2, intensidad, 15)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 20), (I_1, intensidad, 20), (I_2, intensidad, 20)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 25), (I_1, intensidad, 25), (I_2, intensidad, 25)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 30), (I_1, intensidad, 30), (I_2, intensidad, 30)\}$	
$C_{cs}(I_o, I_{intensidad}) = \{(I_o, 35), (I_1, intensidad, 35), (I_2, intensidad, 35)\}$	

Tabla A-1. (Cont...) Cadenas de compresión semántica usadas en el análisis con enfoque clásico.

Caso de estudio 3
Cadenas de compresión semántica de intensidad promedio
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 05), (I_1, ABS, 05), (I_2, ABS, 05)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 10), (I_1, ABS, 10), (I_2, ABS, 10)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 15), (I_1, ABS, 15), (I_2, ABS, 15)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 20), (I_1, ABS, 20), (I_2, ABS, 20)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 25), (I_1, ABS, 35), (I_2, ABS, 25)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 30), (I_1, ABS, 30), (I_2, ABS, 30)\}$
Cadenas de compresión semántica de intensidad
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 05), (I_1, intensidad, 05), (I_2, intensidad, 05)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 10), (I_1, intensidad, 10), (I_2, intensidad, 10)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 15), (I_1, intensidad, 15), (I_2, intensidad, 15)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 20), (I_1, intensidad, 20), (I_2, intensidad, 20)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 35), (I_1, intensidad, 25), (I_2, intensidad, 25)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 30), (I_1, intensidad, 30), (I_2, intensidad, 30)\}$
Caso de estudio 4
Cadenas de compresión semántica de intensidad promedio
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 12.5), (I_1, ABS, 12.5), (I_2, ABS, 12.5)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 15.0), (I_1, ABS, 15.0), (I_2, ABS, 15.0)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 17.5), (I_1, ABS, 17.5), (I_2, ABS, 17.5)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 20.0), (I_1, ABS, 20.0), (I_2, ABS, 20.0)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 22.5), (I_1, ABS, 22.5), (I_2, ABS, 22.5)\}$
$C_{CS}(I_o, I_{ABS}) = \{(I_o, 25.0), (I_1, ABS, 25.0), (I_2, ABS, 30.0)\}$
Cadenas de compresión semántica de intensidad
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 12.5), (I_1, intensidad, 12.5), (I_2, intensidad, 12.5)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 15.0), (I_1, intensidad, 15.0), (I_2, intensidad, 15.0)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 17.5), (I_1, intensidad, 17.5), (I_2, intensidad, 17.5)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 20.0), (I_1, intensidad, 20.0), (I_2, intensidad, 20.0)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 22.5), (I_1, intensidad, 22.5), (I_2, intensidad, 22.5)\}$
$C_{CS}(I_o, I_{intensidad}) = \{(I_o, 25.0), (I_1, intensidad, 25.0), (I_2, intensidad, 25.0)\}$

### A.1.1 Caso de estudio 1: Geoimagen de una isla, ejemplo 1



a  
b  
c  
d

Figura A-1. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.

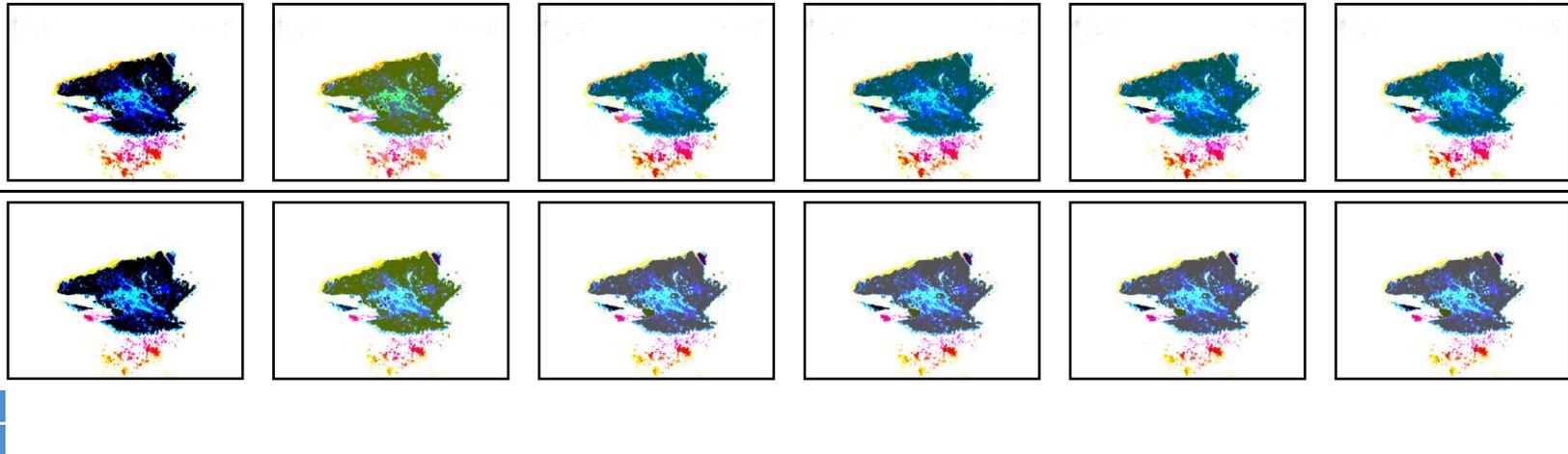


Figura A-1. (Cont...) Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.

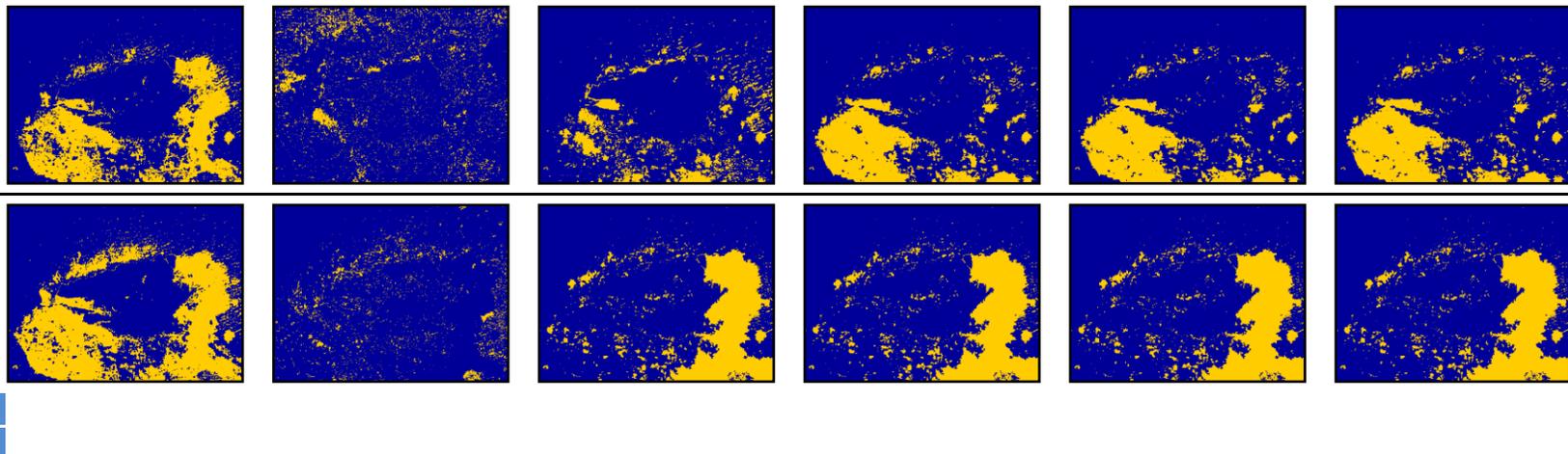
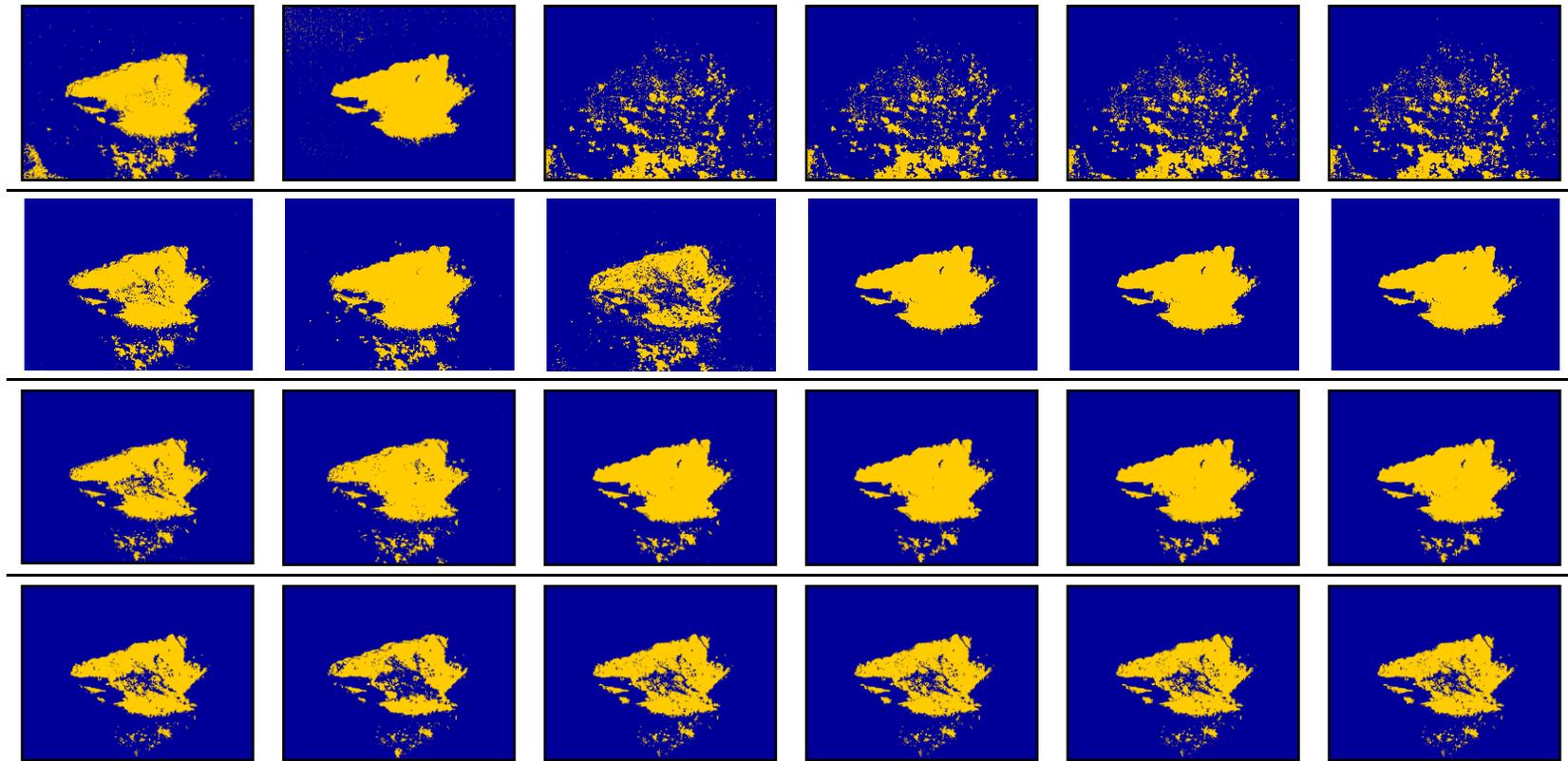


Figura A-2. Resultados del algoritmo de agrupamiento  $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.



c  
d  
e  
f

Figura A-2. (Cont...) Resultados del algoritmo de agrupamiento  $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.

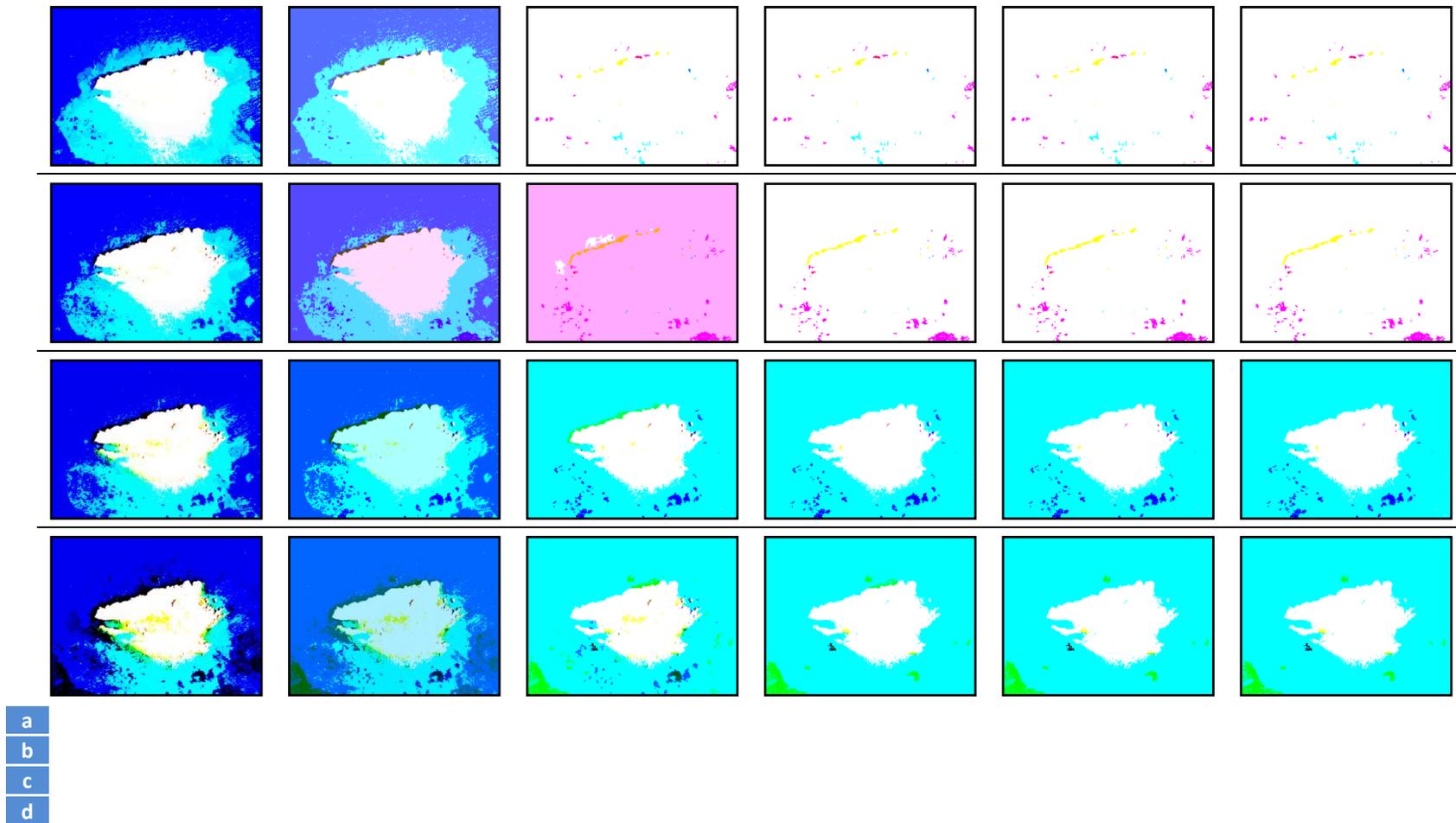
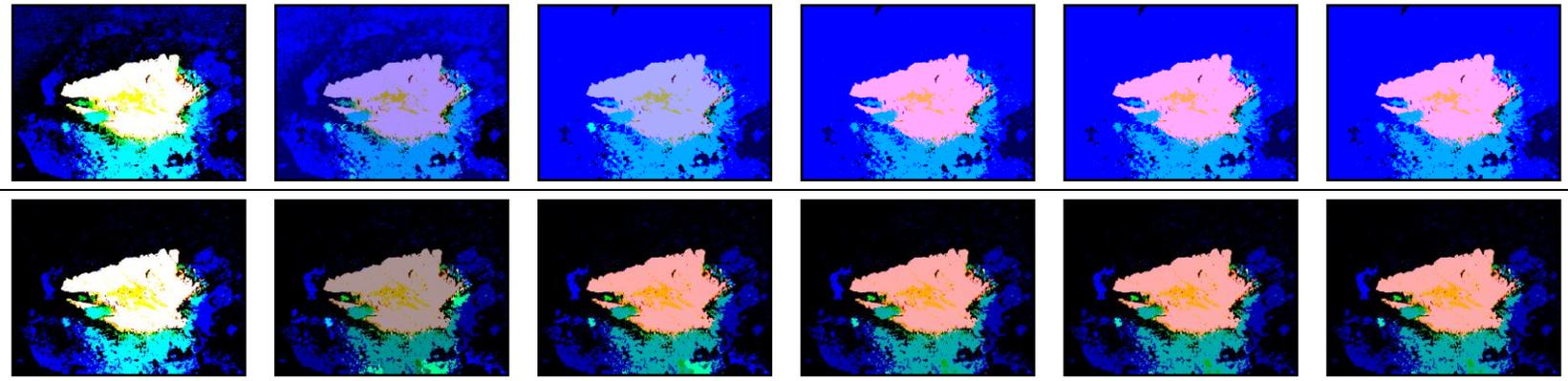


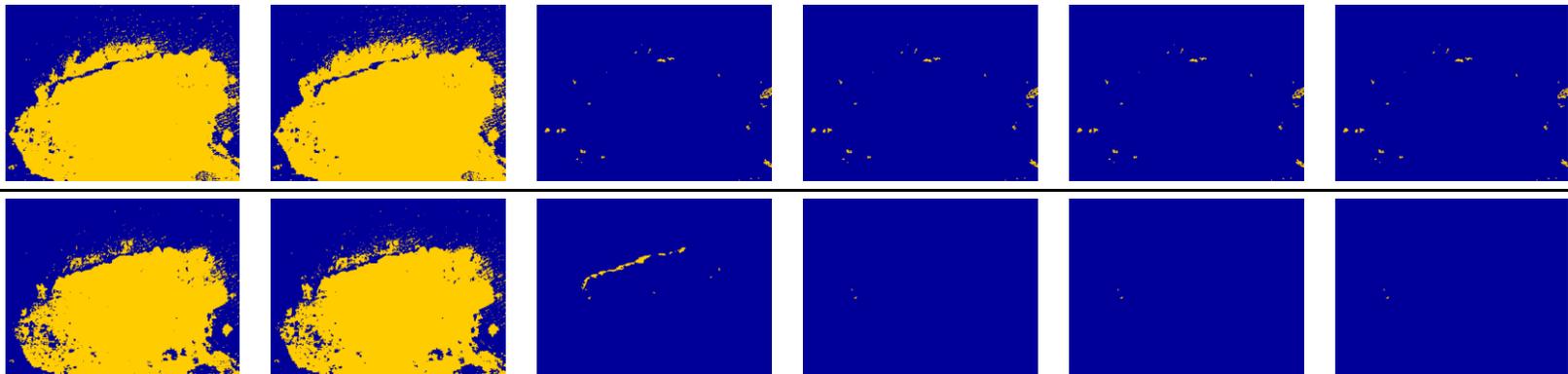
Figura A–3. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.



e

f

Figura A-3. (Cont...) Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.



a

b

Figura A-4. Resultados del algoritmo de agrupamiento  $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.

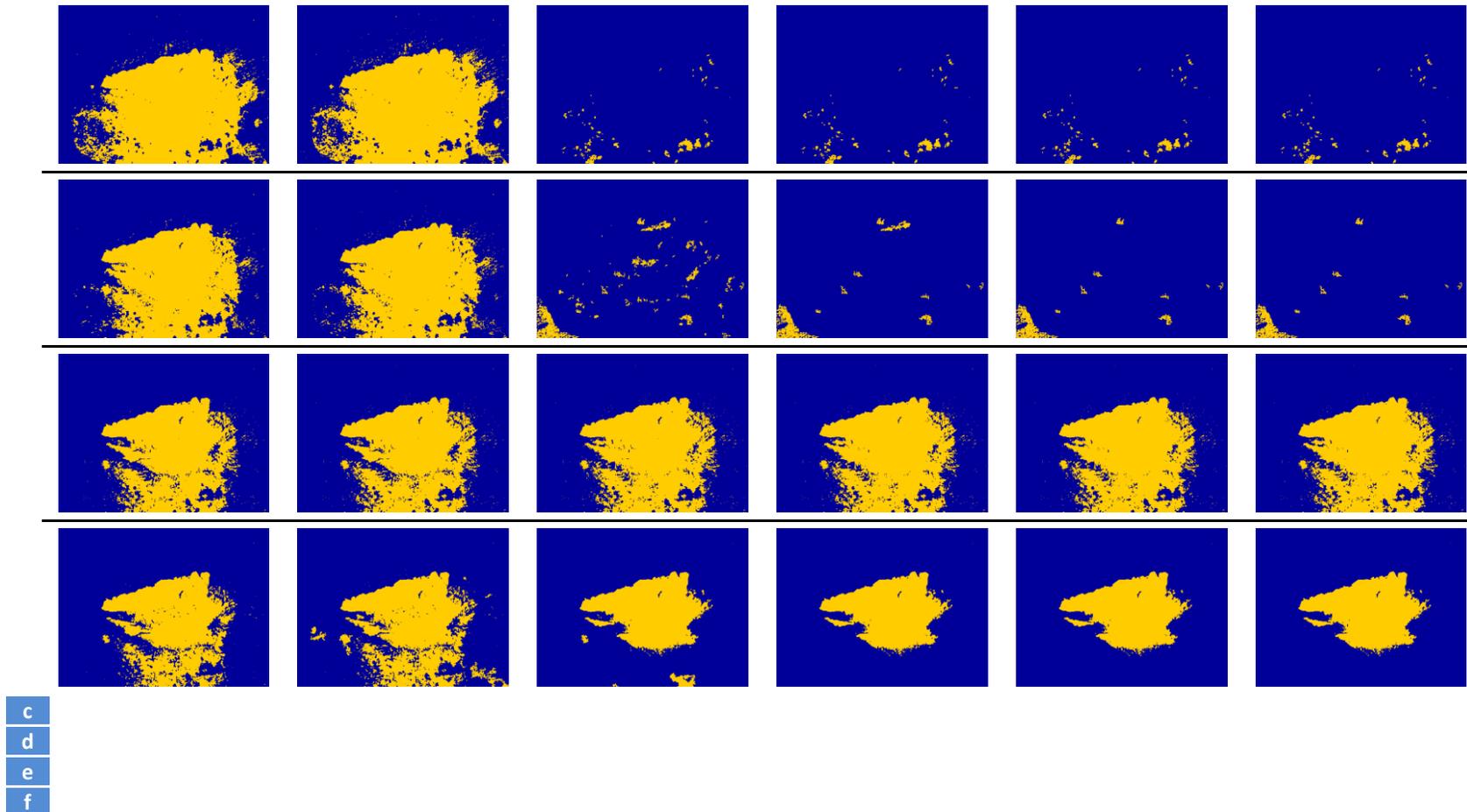


Figura A-4. (Cont...) Resultados del algoritmo de agrupamiento  $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 15; (b) 20; (c) 25; (d) 30; (e) 35; (f) 40.

### A.1.2 Caso de estudio 2: Geoimagen de una isla, ejemplo 2

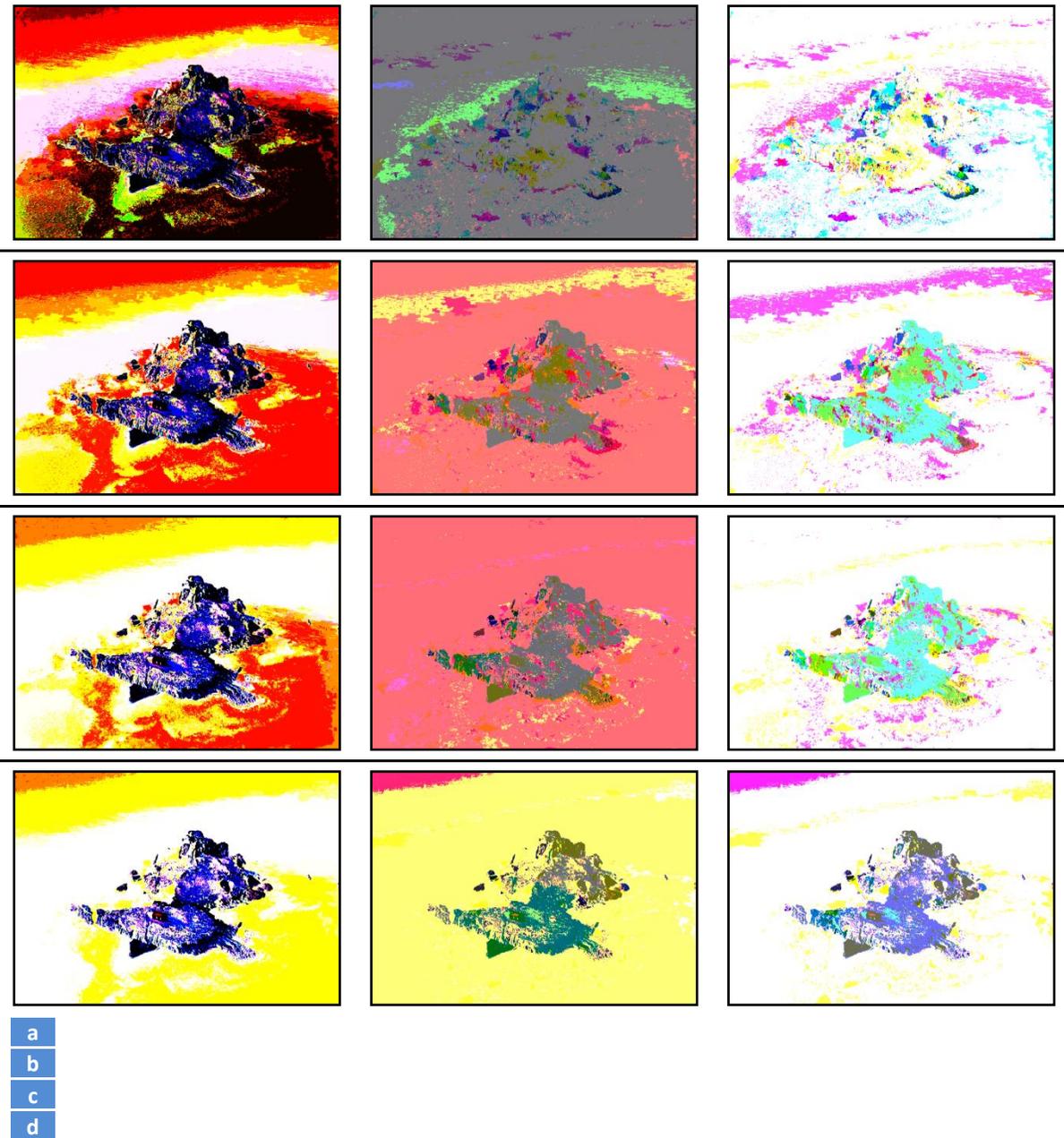


Figura A-5. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.

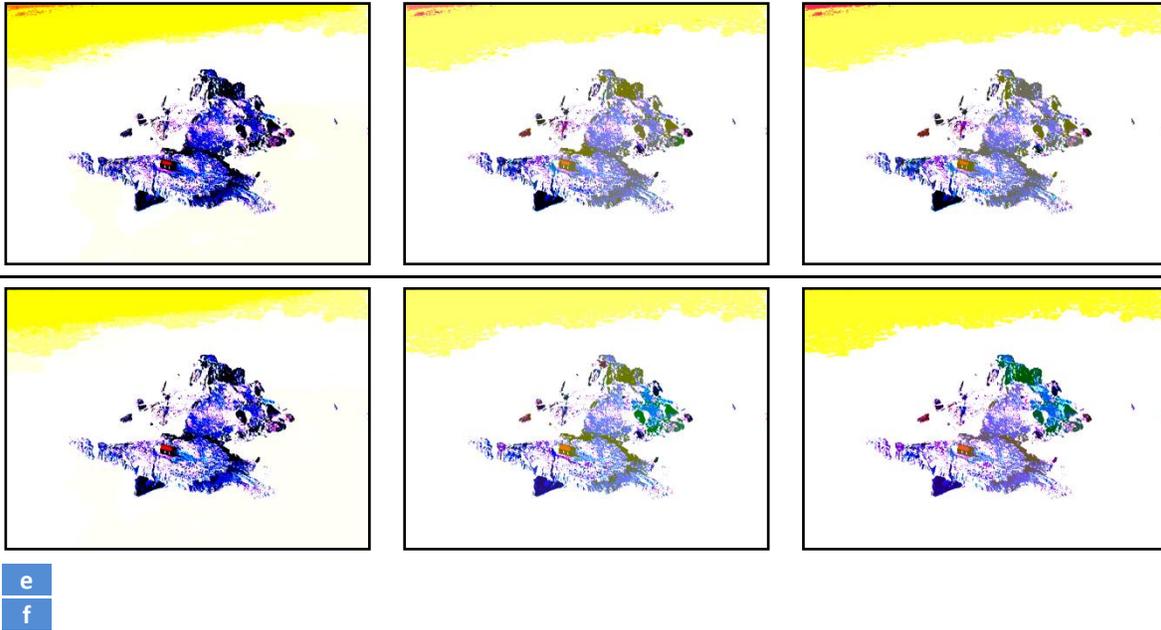


Figura A-5. (Cont...) Resultados de cada iteración de la compresión semántica de la geoinmagen de intensidad promedio con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.

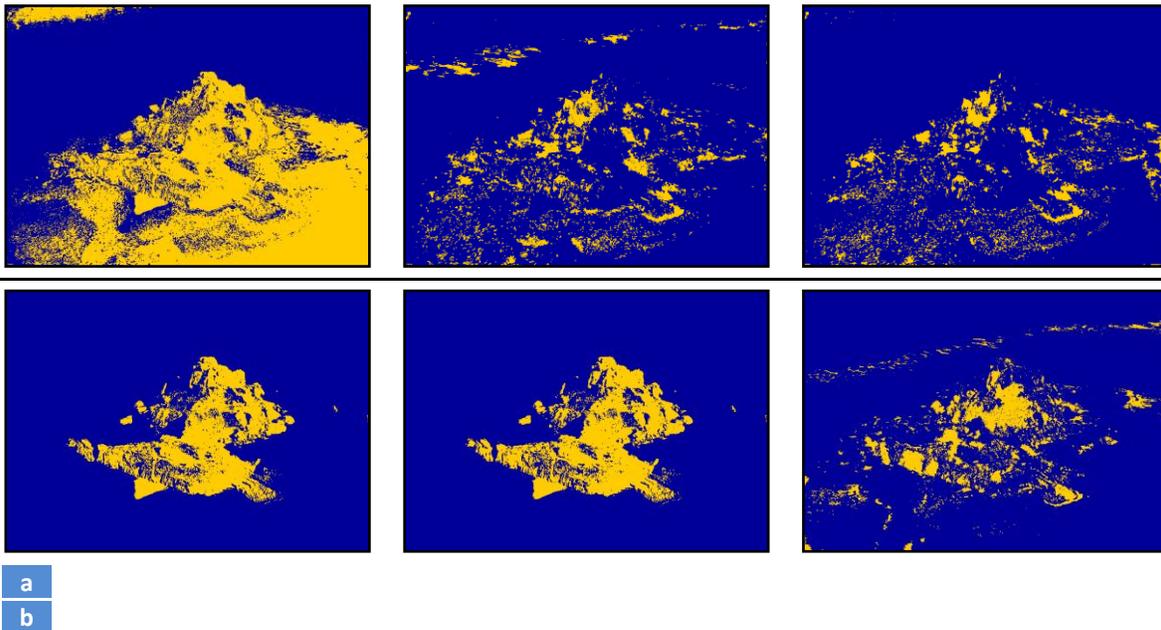


Figura A-6. Resultados del algoritmo de agrupamiento c-medias difuso aplicado a las geoinmágenes que se obtienen en cada iteración de la compresión semántica de la geoinmagen de intensidad promedio con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.

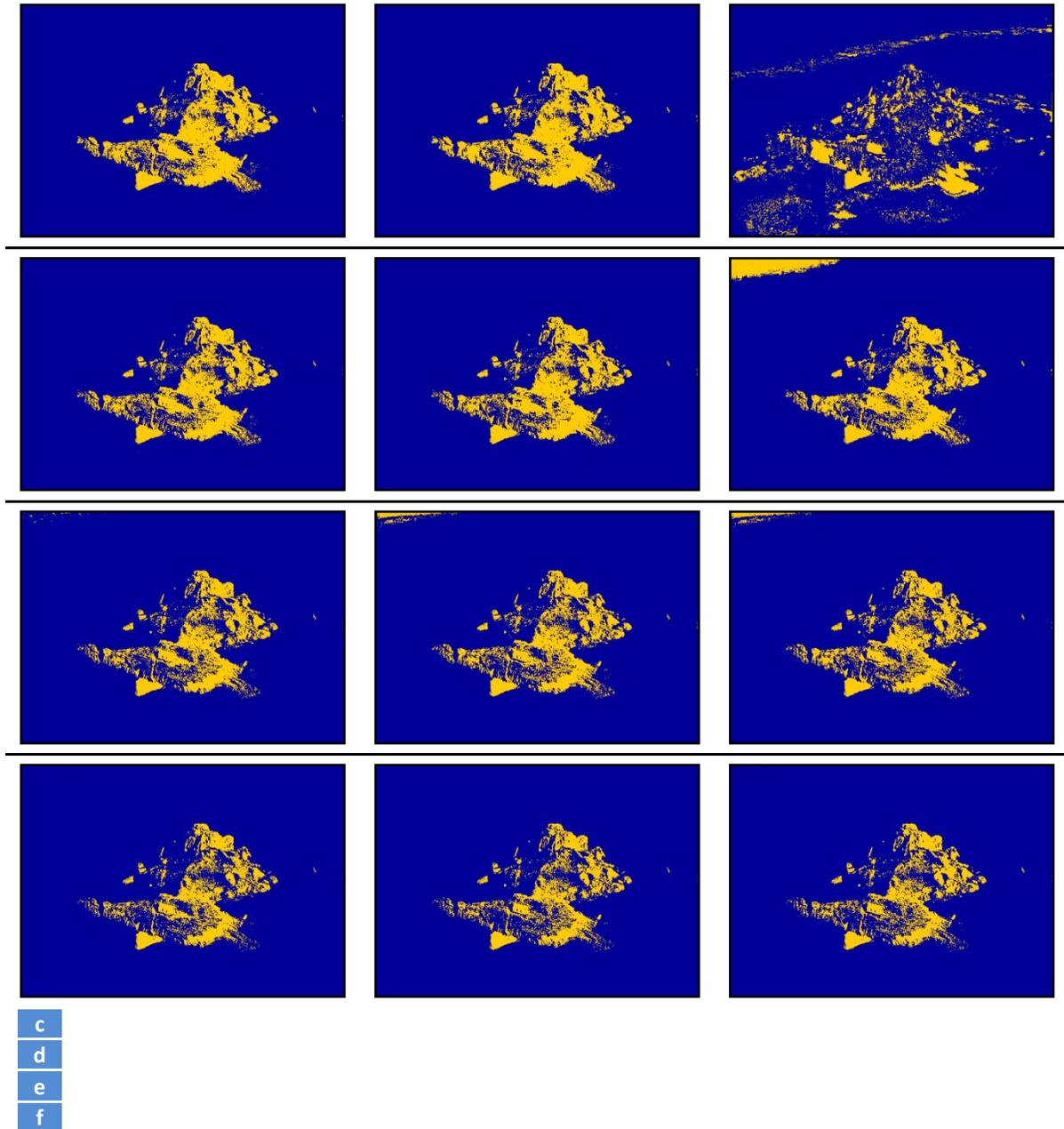


Figura A-6. (Cont...) Resultados del algoritmo de agrupamiento c-medias difuso aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.

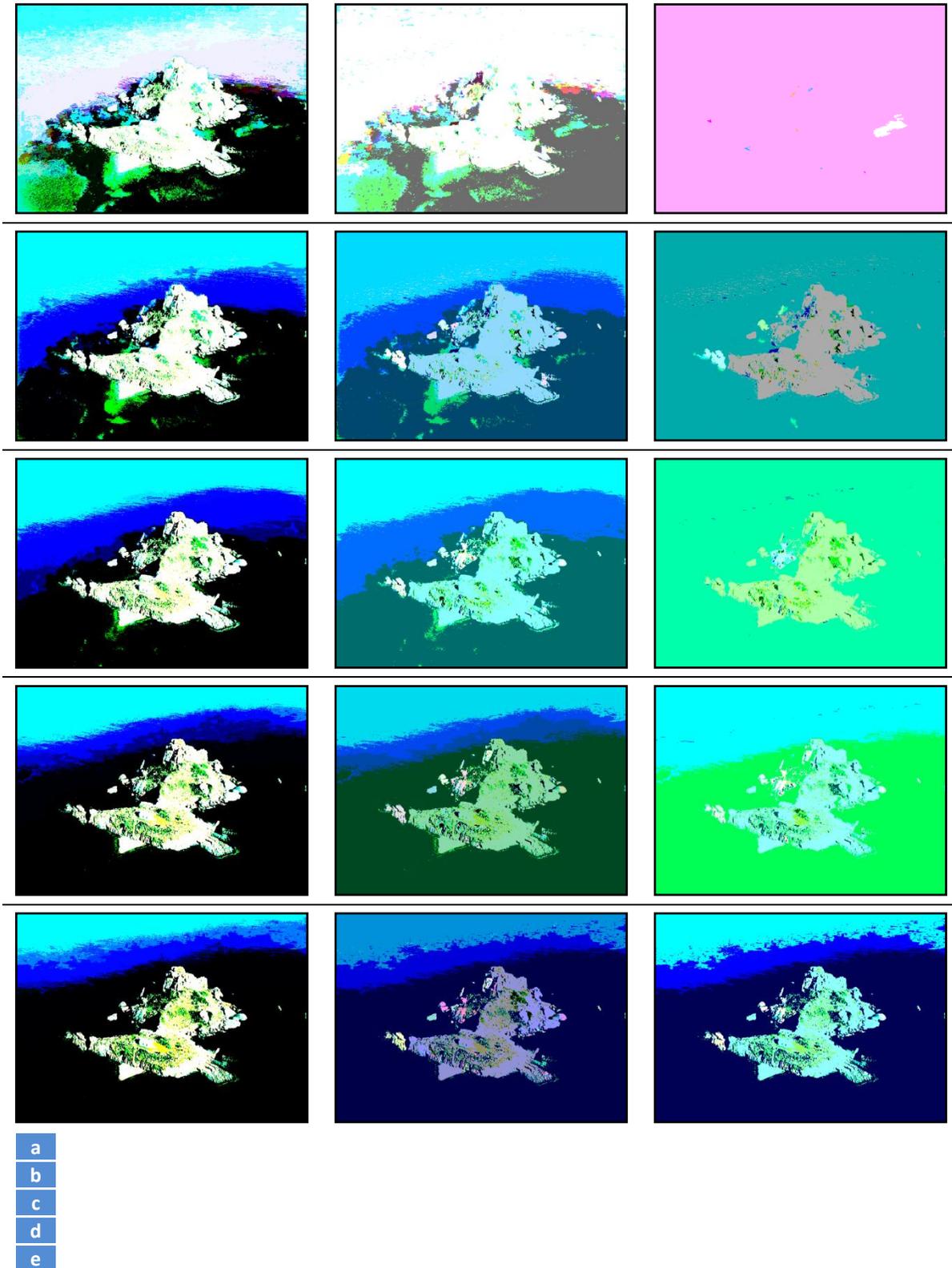
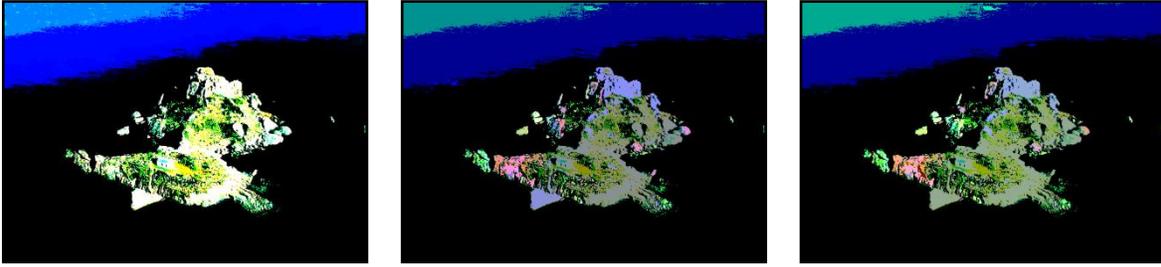
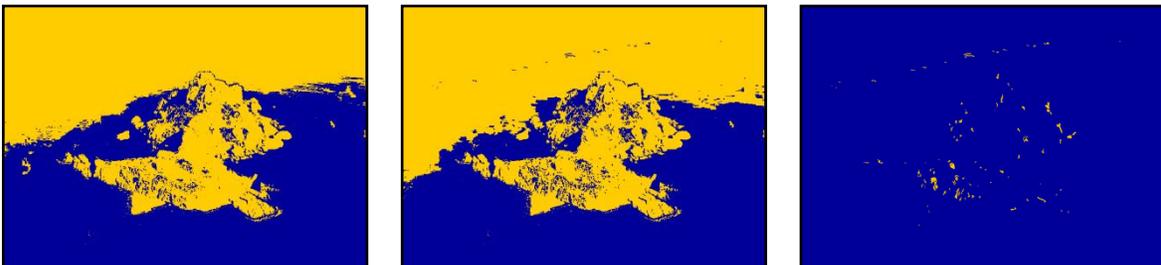
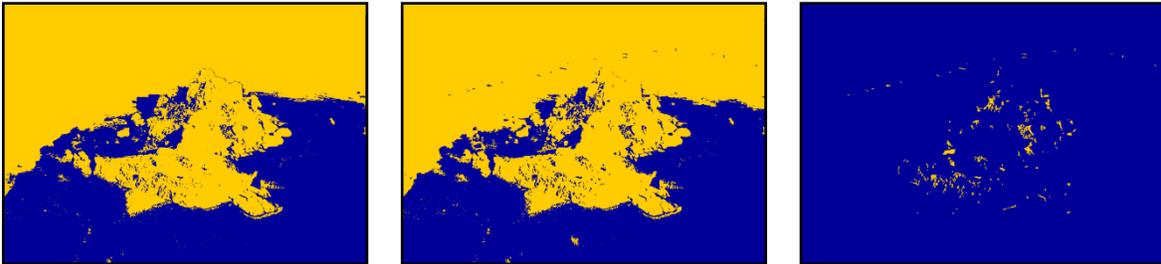
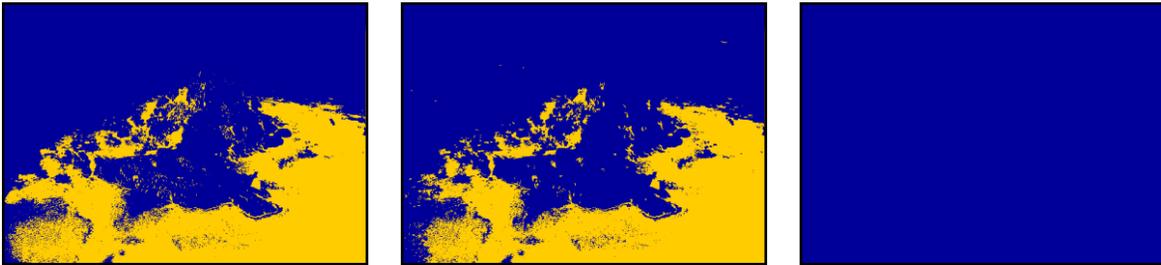


Figura A-7. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.



f

Figura A-7. (Cont...) Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.



a

b

c

Figura A-8. Resultados del algoritmo de agrupamiento c-medias difuso aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.

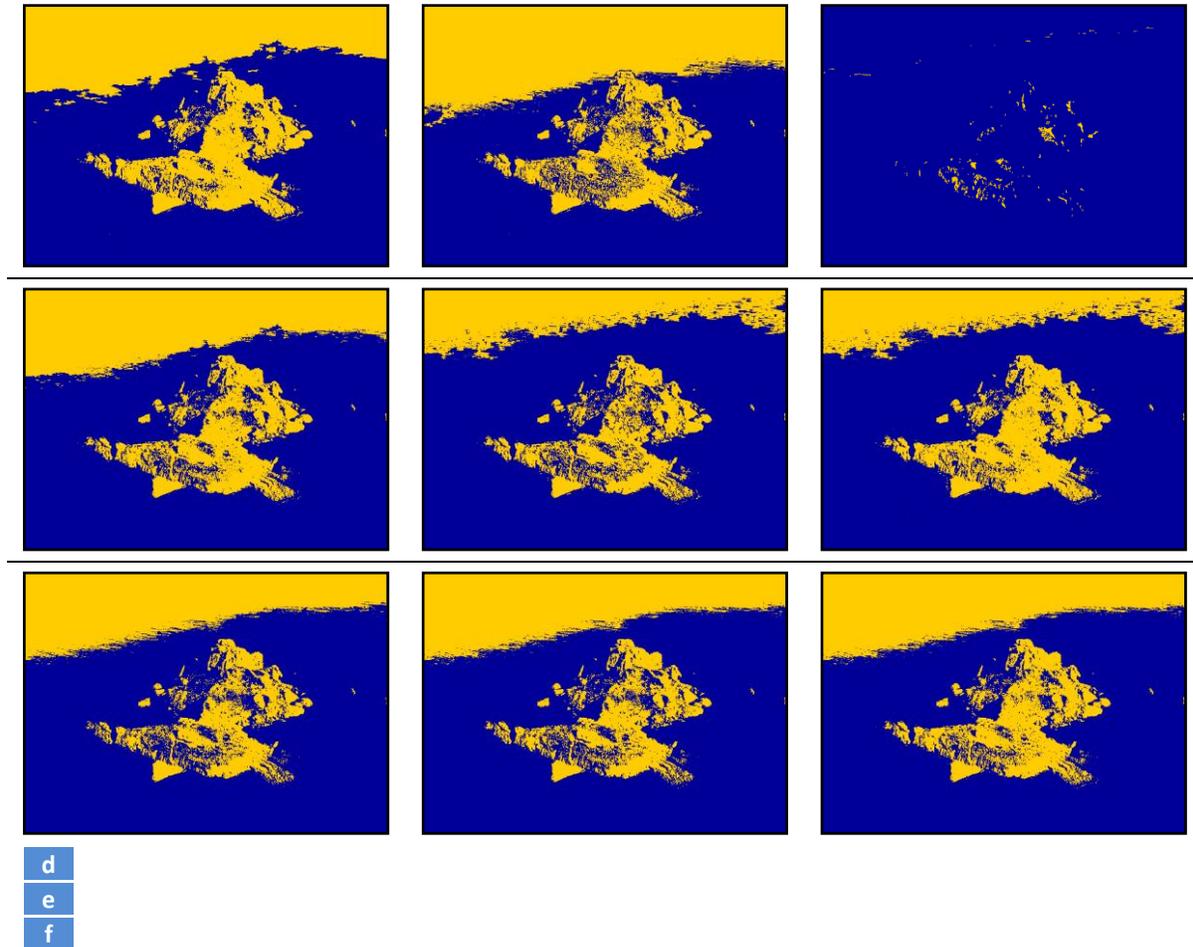


Figura A-8. (Cont...) Resultados del algoritmo de agrupamiento c-medias difuso aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 10; (b) 15; (c) 20; (d) 25; (e) 30; (f) 35.

### A.1.3 Caso de estudio 3: Geoimagen de los Grandes Lagos

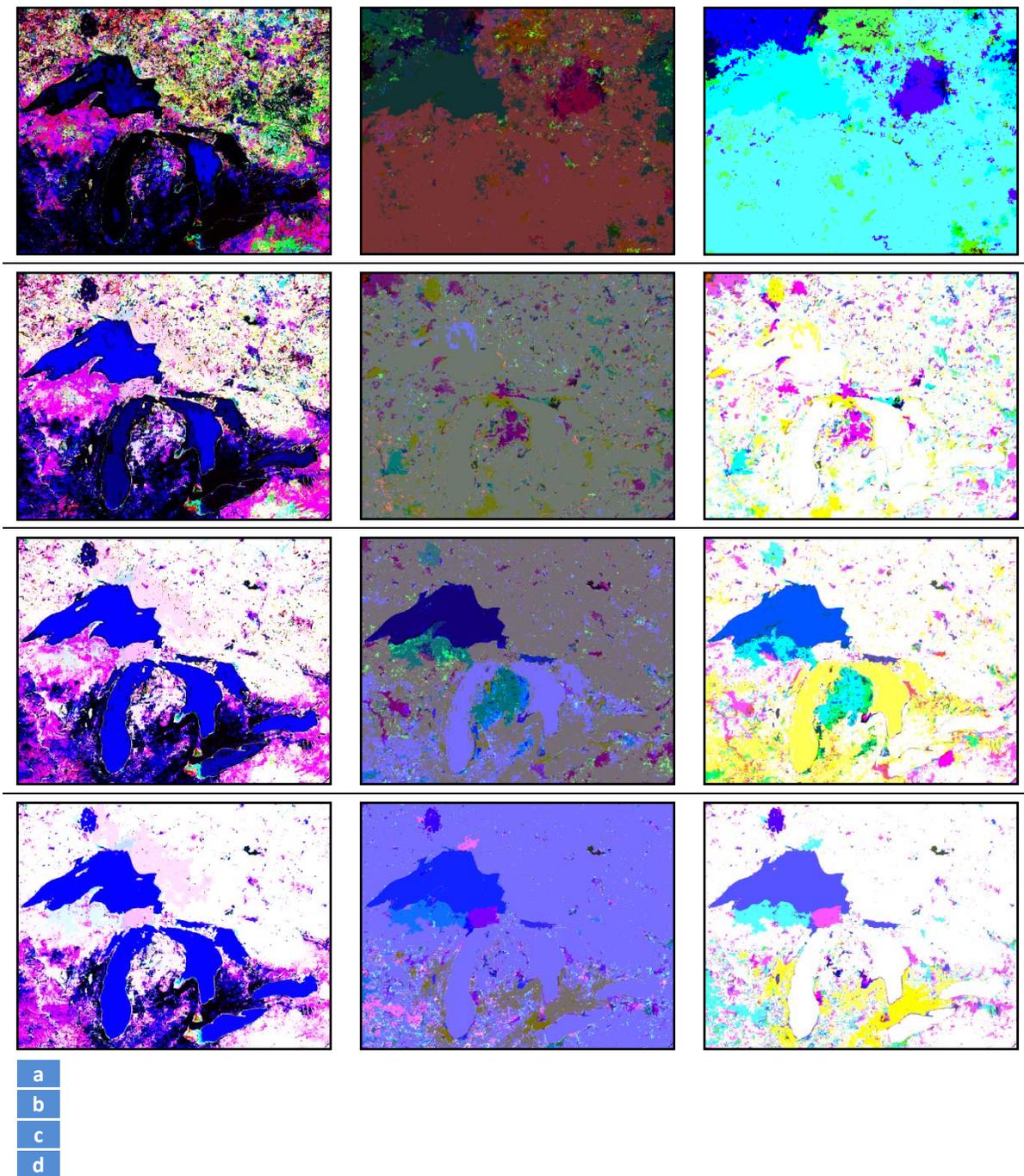


Figura A-9. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.

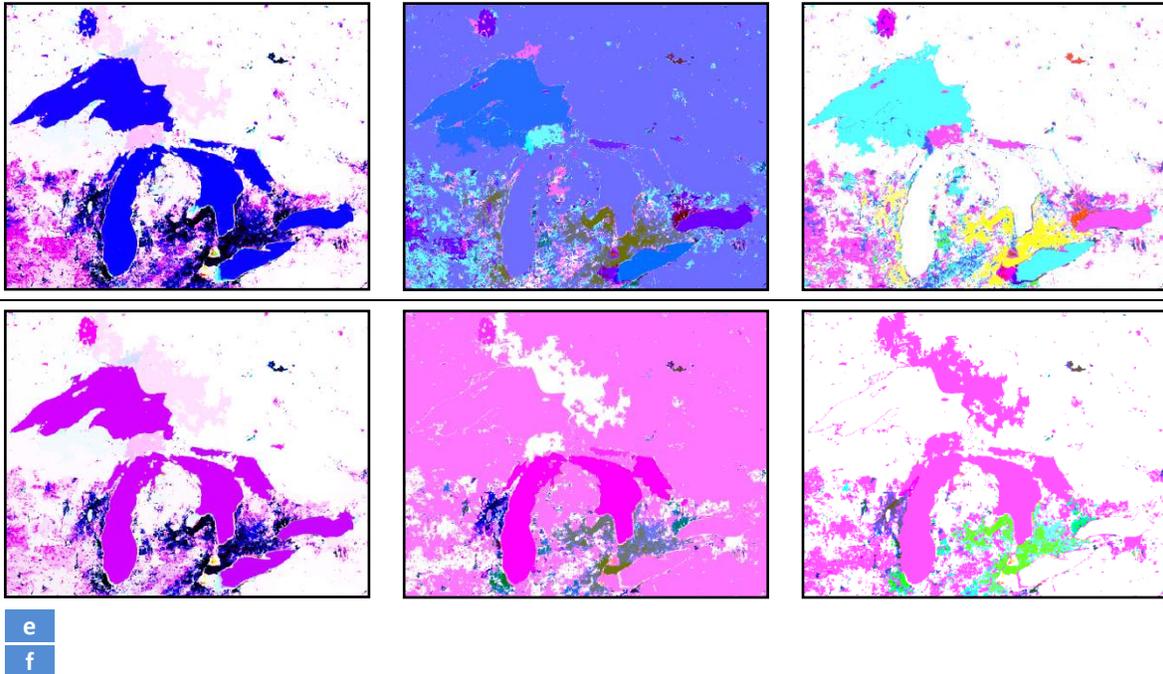


Figura A-9. (Cont...) Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.

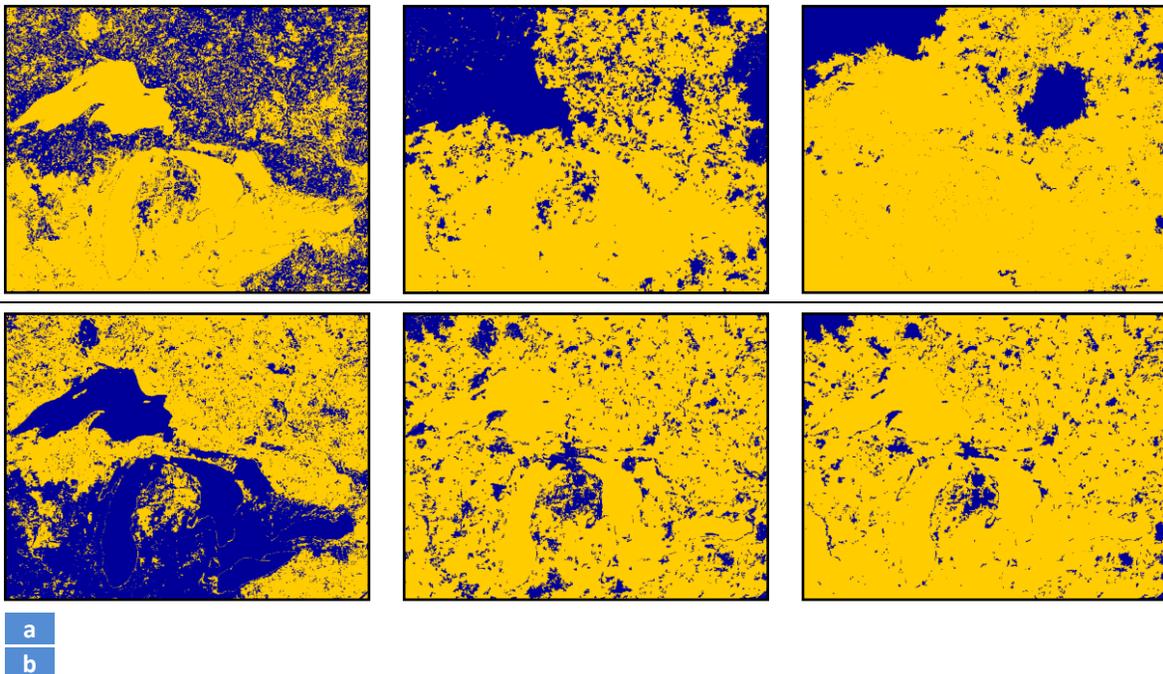


Figura A-10. Resultados del algoritmo de agrupamiento  $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.

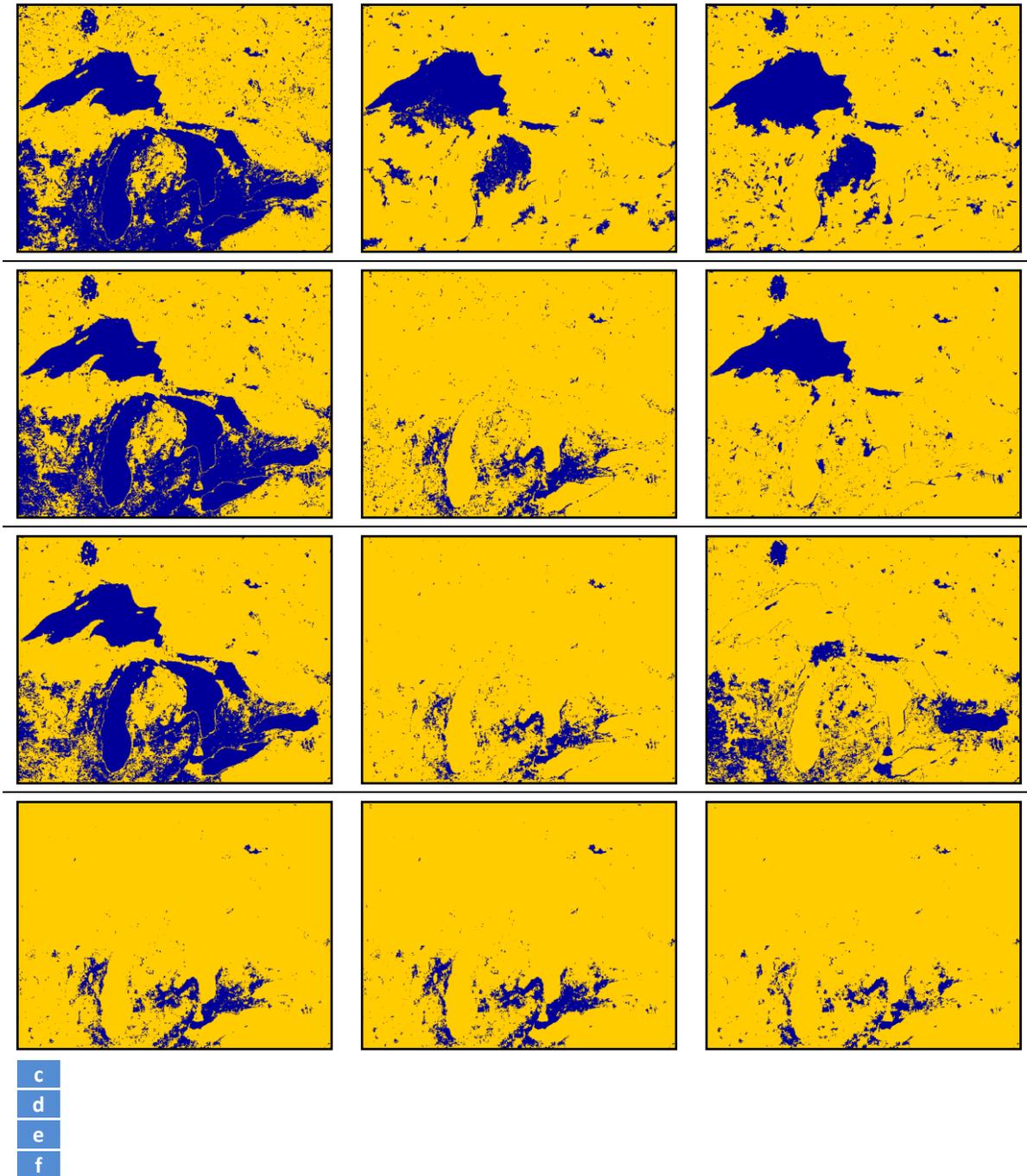


Figura A-10. (Cont...) Resultados del algoritmo de agrupamiento  $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.

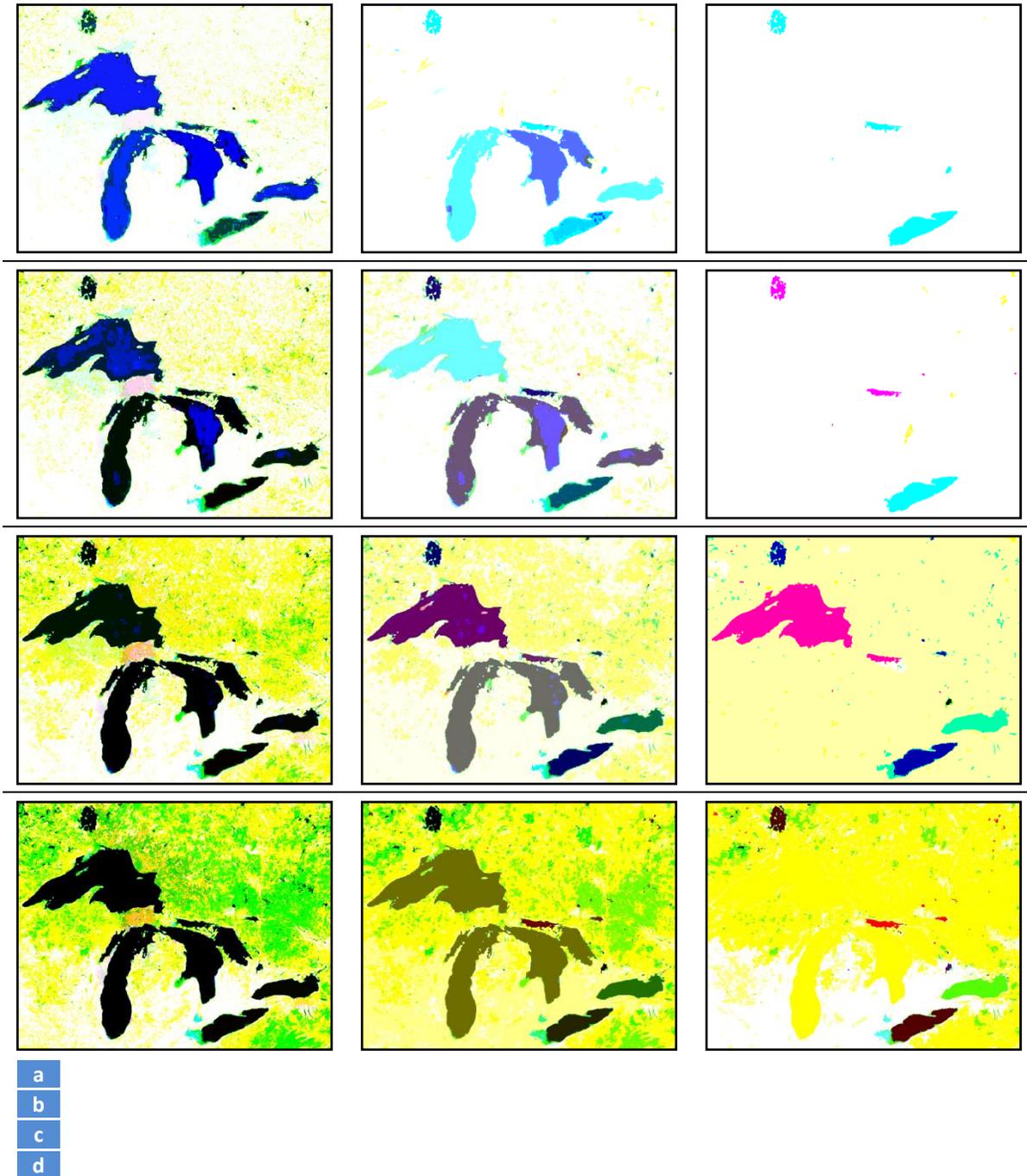
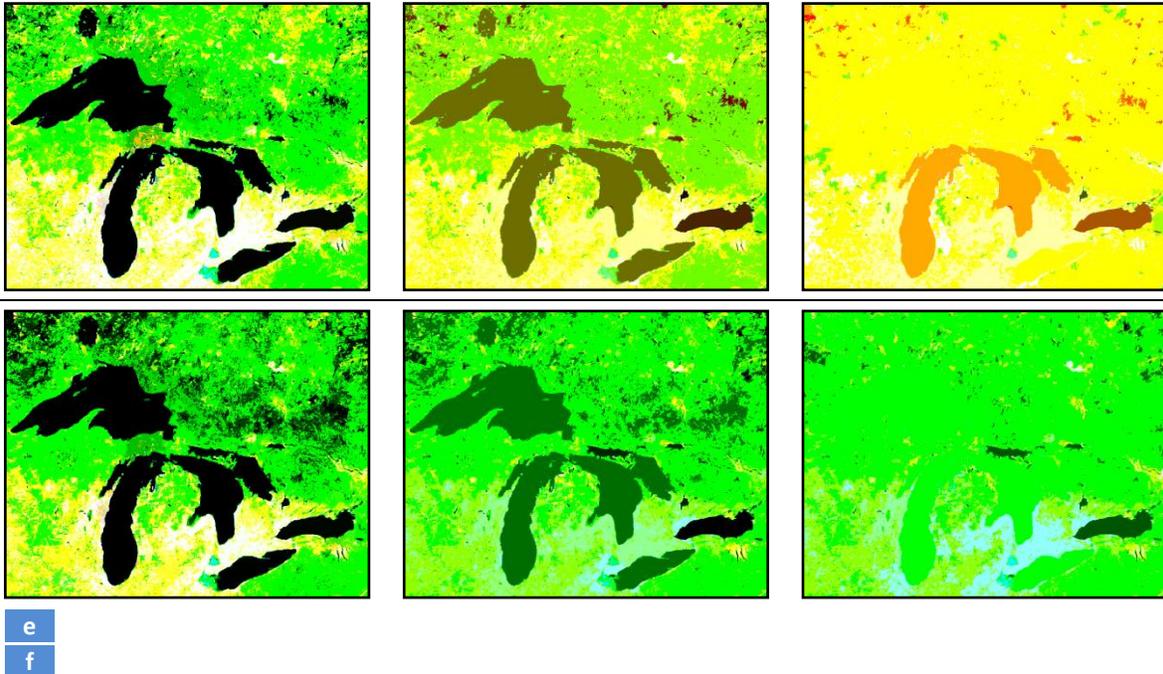


Figura A-11. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.



¡Error! No se encuentra el origen de la referencia.¡Error! No se encuentra el origen de la referencia.Figura A-11. (Cont...) Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.

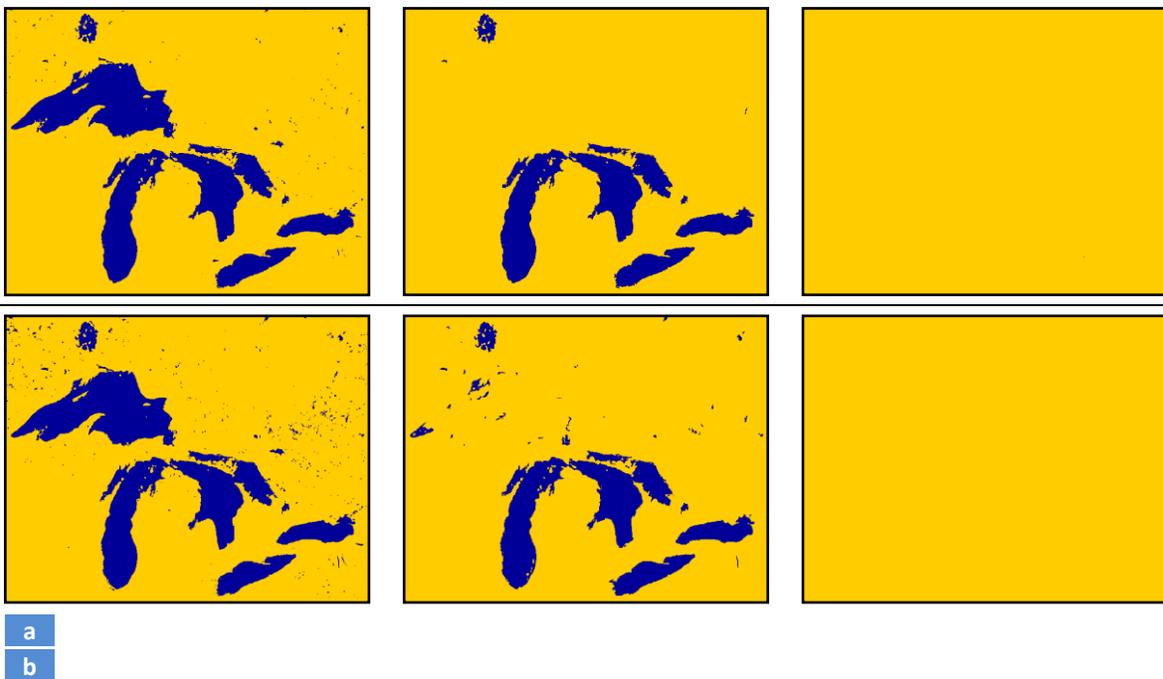


Figura A-12. Resultados del algoritmo de agrupamiento  $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.

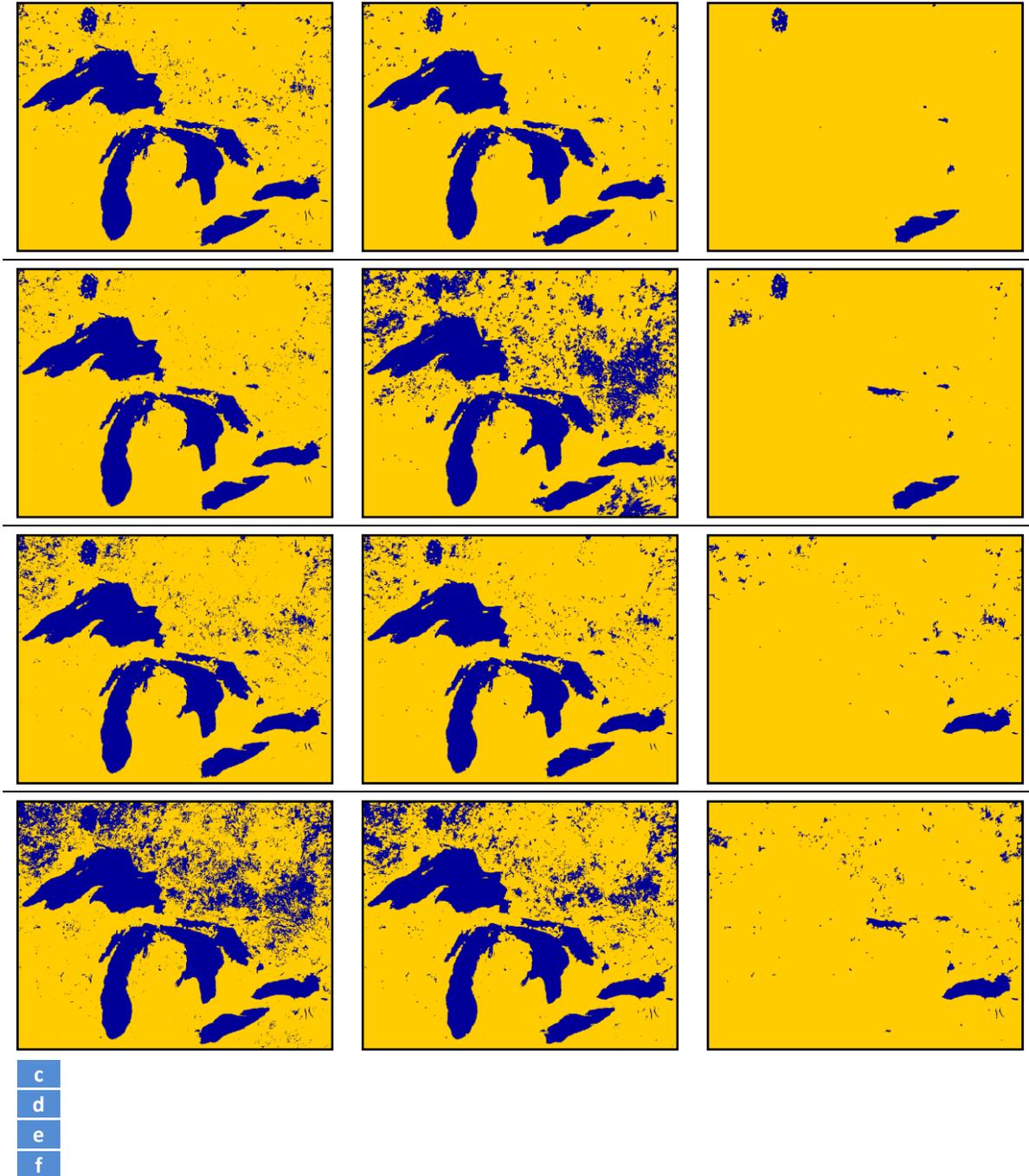


Figura A-12. (Cont...) Resultados del algoritmo de agrupamiento  $k$ -medias aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 05; (b) 10; (c) 15; (d) 20; (e) 25; (f) 30.

### A.1.4 Caso de estudio 4: Geoimagen de la República Mexicana

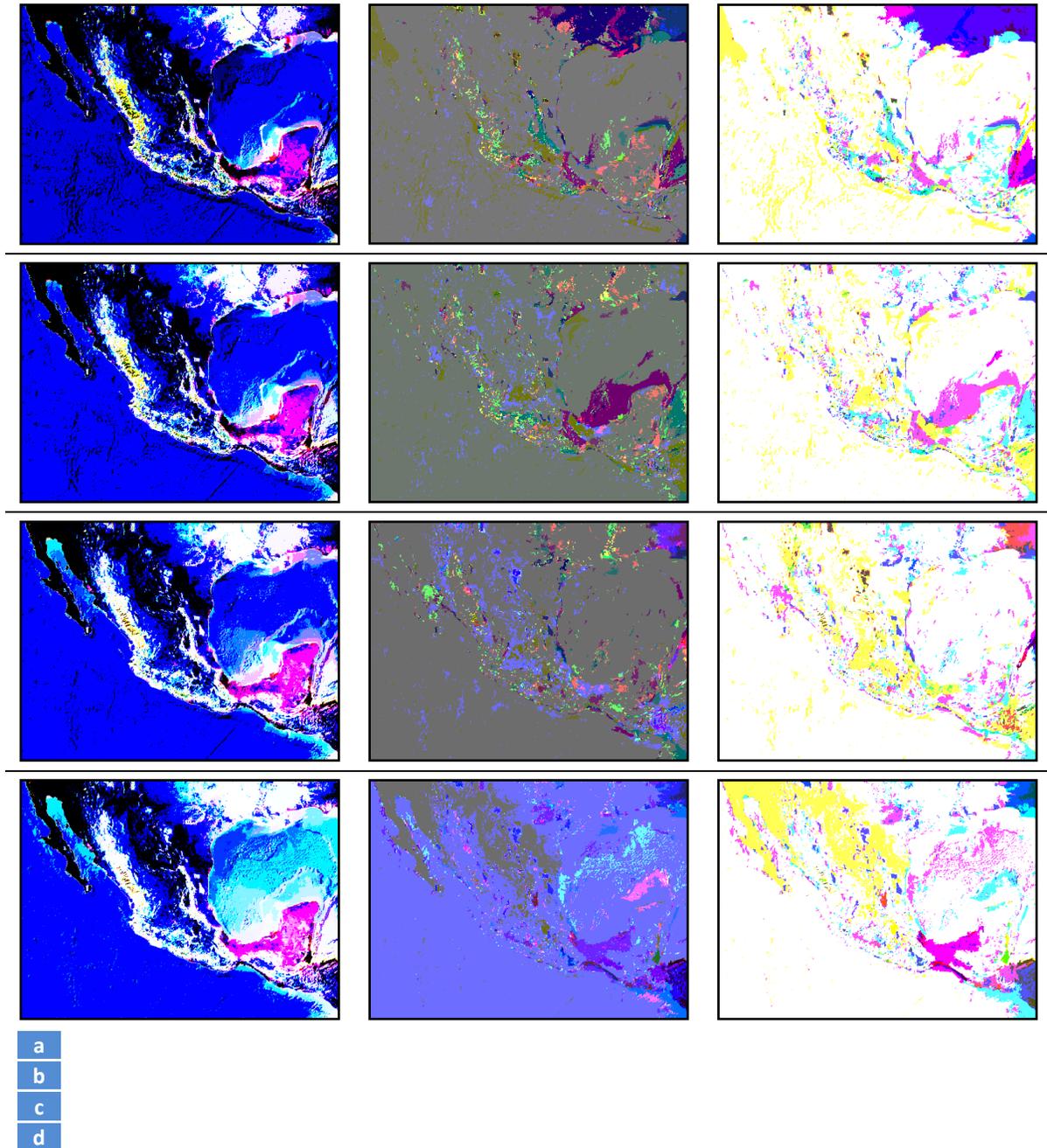


Figura A-13. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 20; (e) 22.5; (f) 25.

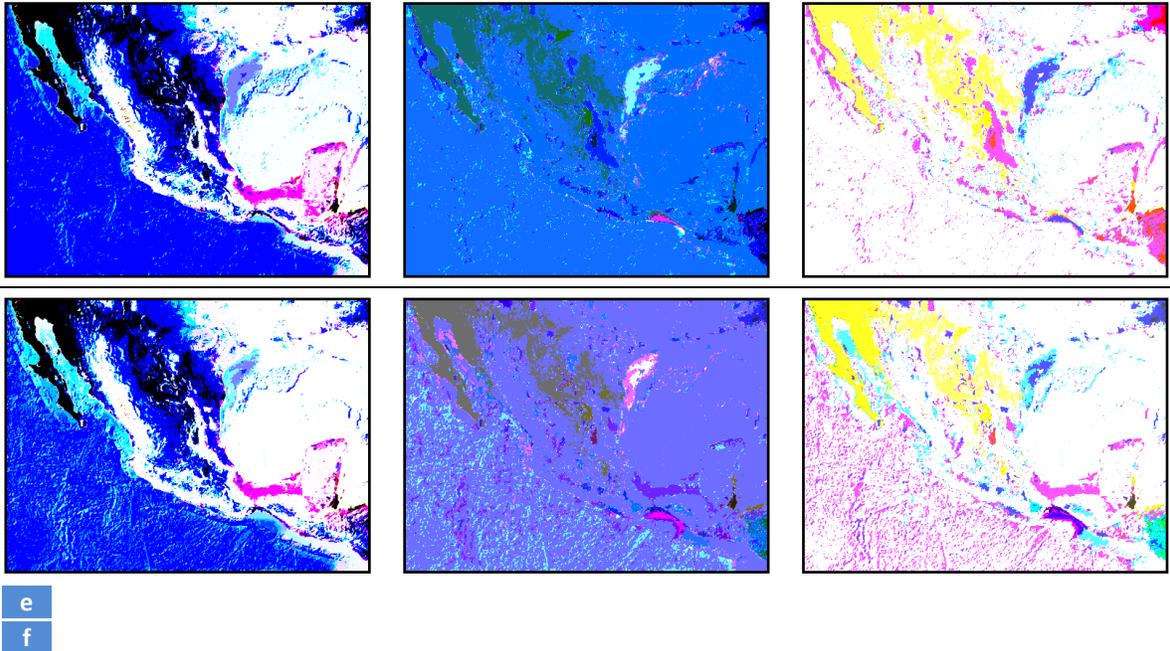


Figura A-13. (Cont...) Resultados de cada iteración de la compresión semántica de la geomagen de intensidad promedio con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 20; (e) 22.5; (f) 25.

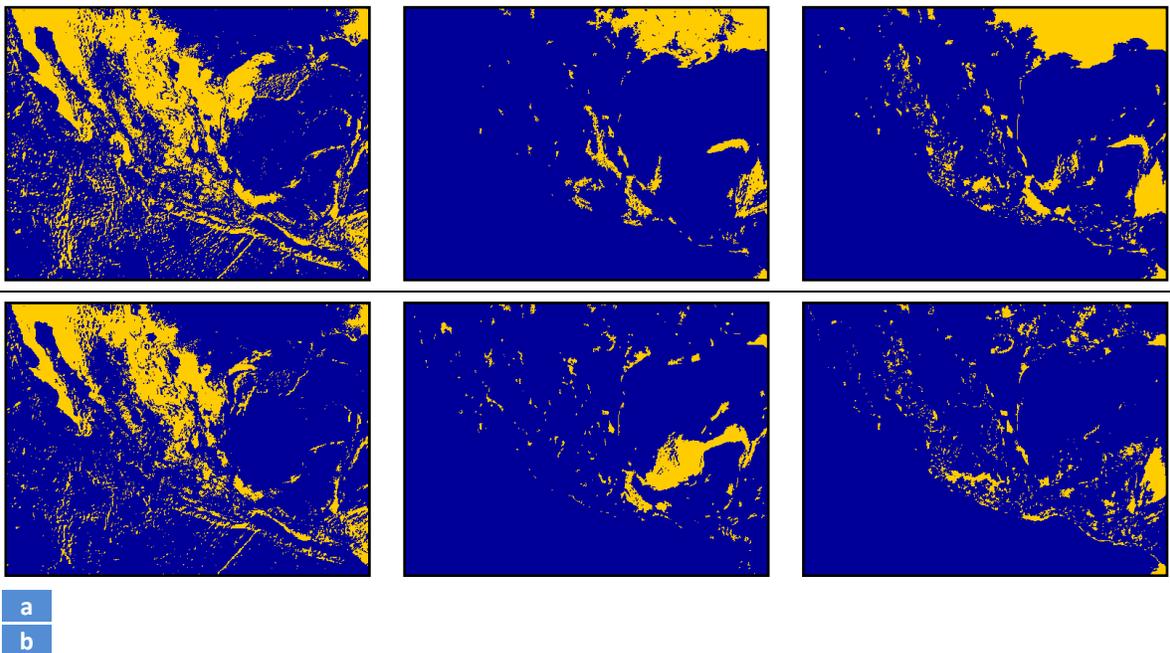


Figura A-14. Resultados del algoritmo de agrupamiento ISODATA aplicado a las geomágenes que se obtienen en cada iteración de la compresión semántica de la geomagen de intensidad promedio con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 30; (e) 32.5; (f) 35.

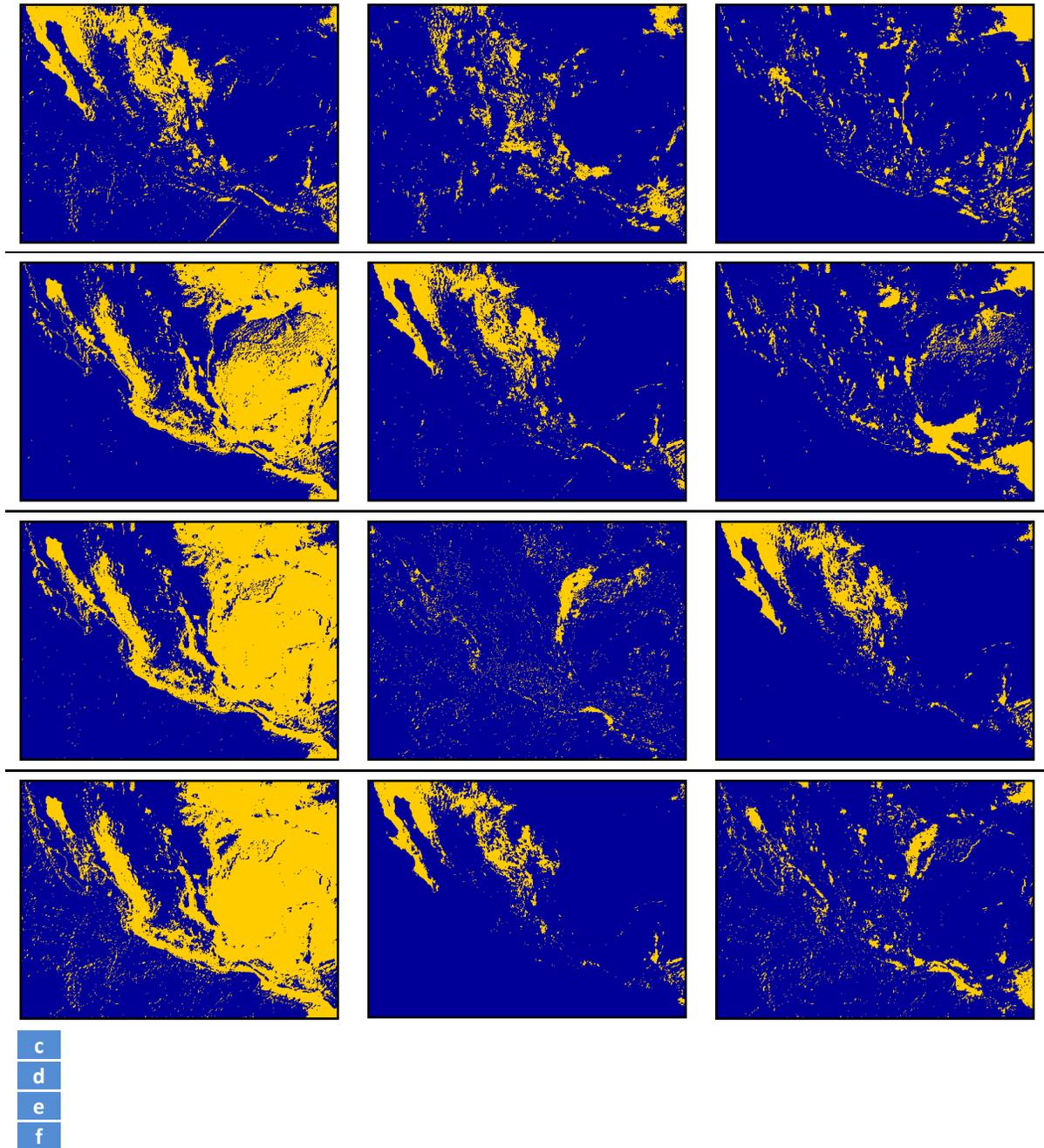


Figura A-14. (Cont...) Resultados del algoritmo de agrupamiento ISODATA aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad promedio con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 30; (e) 32.5; (f) 35.

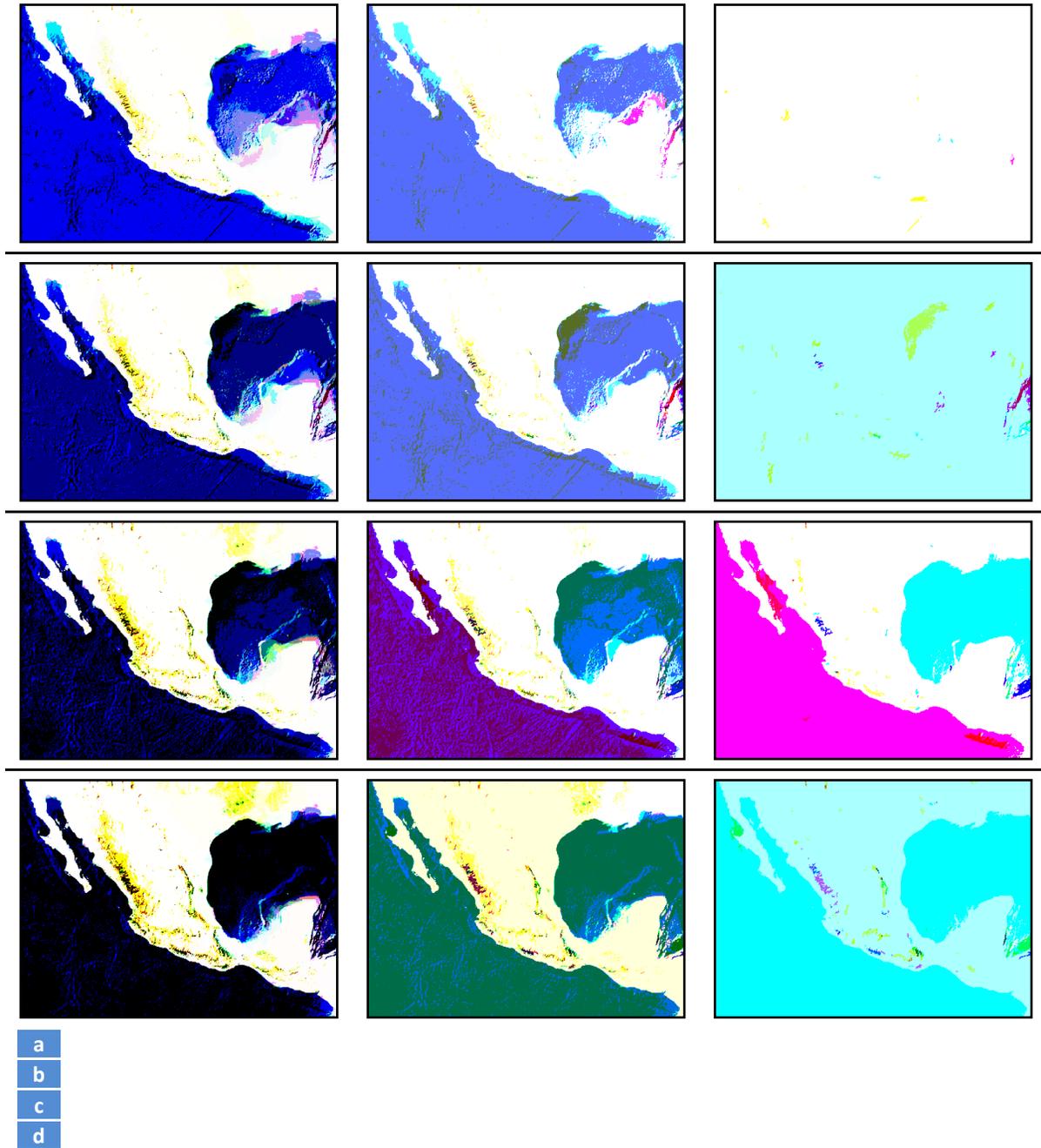


Figura A–15. Resultados de cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 20; (e) 22.5; (f) 25.

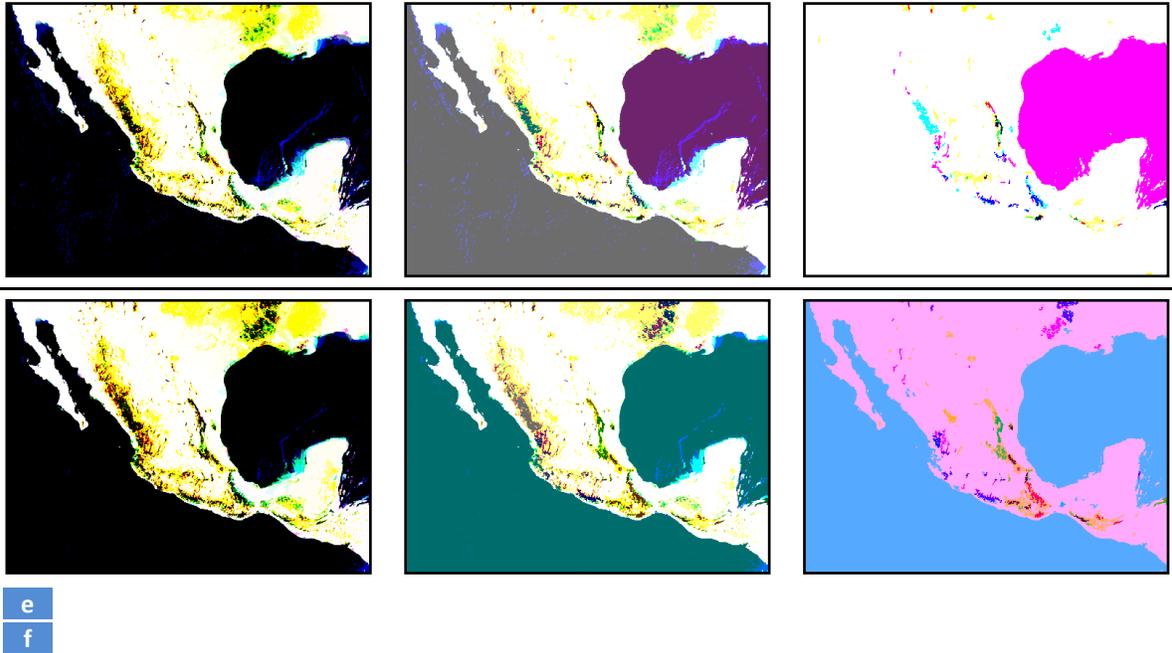


Figura A-15. (Cont...) Resultados de cada iteración de la compresión semántica de la geoinmagen de intensidad con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 20; (e) 22.5; (f) 25.

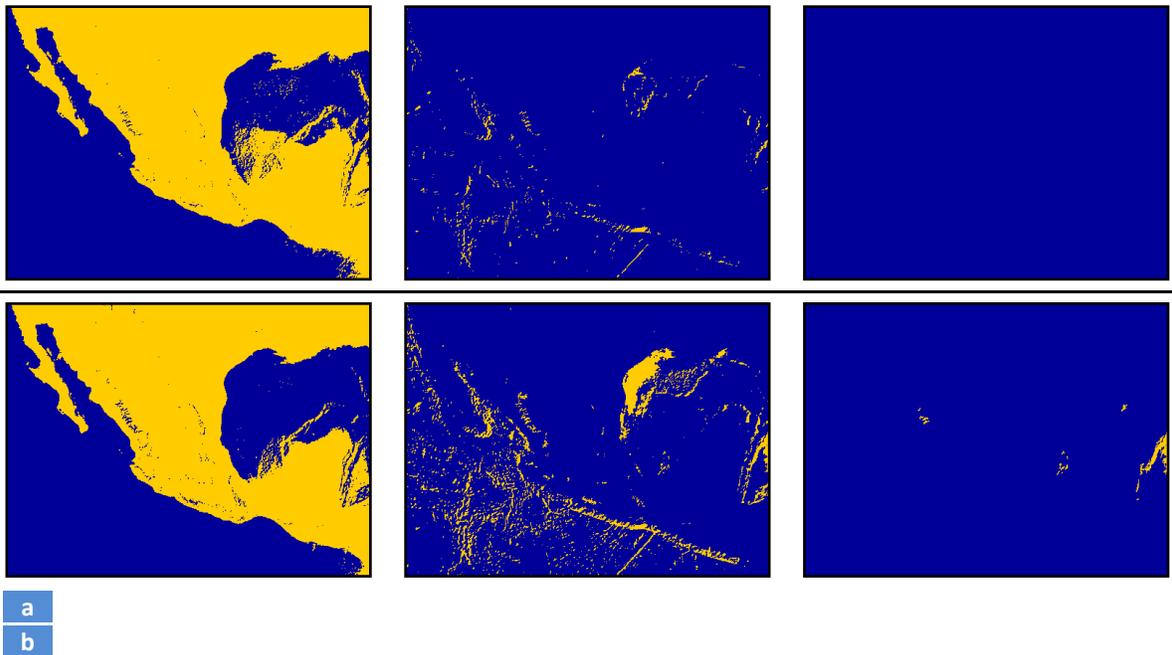


Figura A-16. Resultados del algoritmo de agrupamiento ISODATA aplicado a las geoinmágenes que se obtienen en cada iteración de la compresión semántica de la geoinmagen de intensidad con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 30; (e) 32.5; (f) 35.

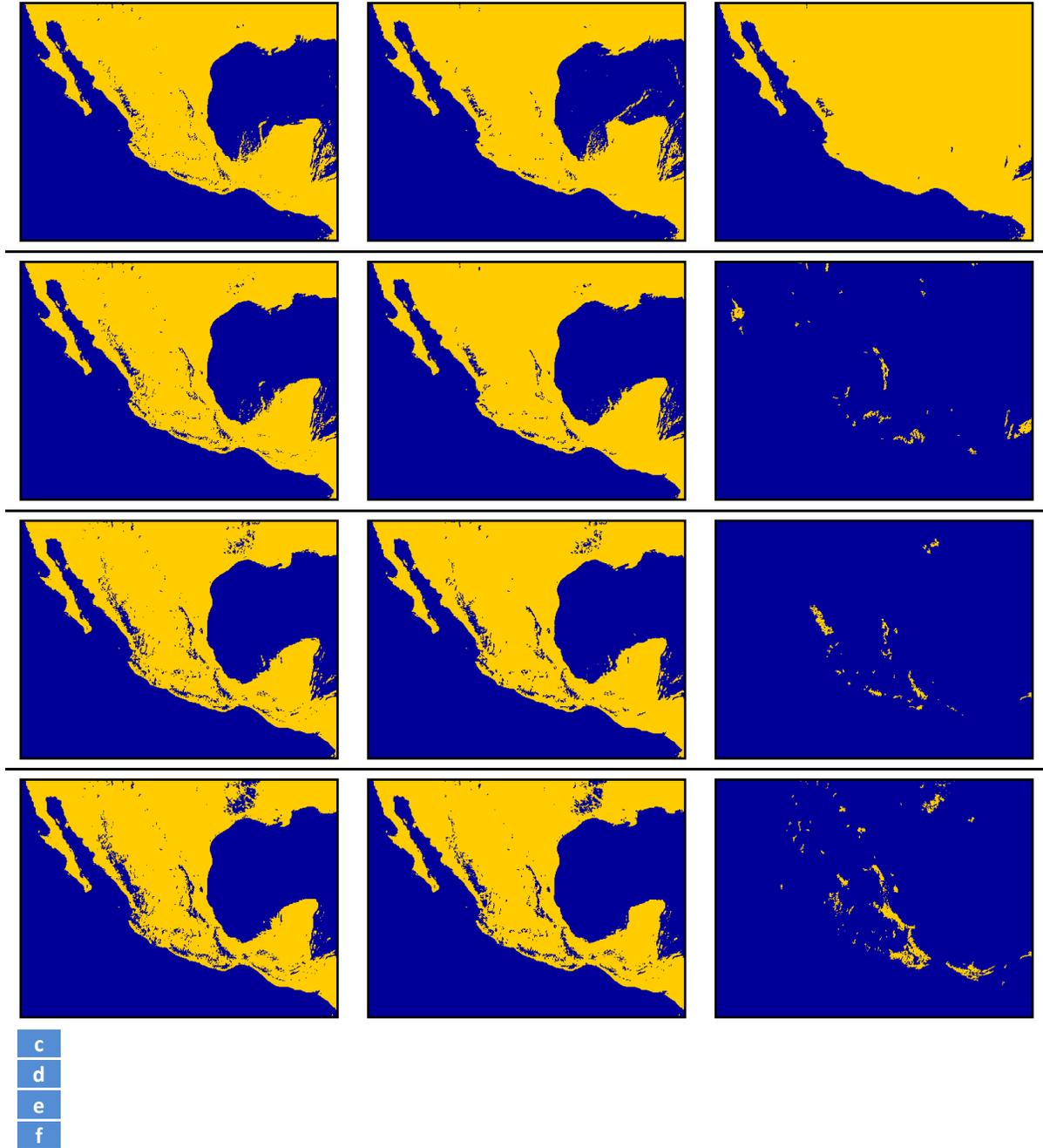


Figura A-16. (Cont...) Resultados del algoritmo de agrupamiento ISODATA aplicado a las geoimágenes que se obtienen en cada iteración de la compresión semántica de la geoimagen de intensidad con diferentes umbrales de similitud; (a) 12.5; (b) 15; (c) 17.5; (d) 30; (e) 32.5; (f) 35.



**Anexo B**

**Código fuente**



## B.1 Código fuente del algoritmo de compresión semántica

```

#pragma hdrstop
#include "UnitDecomposition.h"
#pragma package(smart_init)
void __fastcall WriteByteToFile(AnsiString file, BYTE* array, int items)
{
    FILE* data;

    data = fopen(file.c_str(), "wt");
    for(register int i = 0; i < items; i++)
        fprintf(data, "%u\n", array[i]);
    fclose(data);
}

void __fastcall WriteByteToFile(AnsiString file, BYTE array)
{
    FILE* data;

    data = fopen(file.c_str(), "wt");
    fprintf(data, "%u\n", array);
    fclose(data);
}

void __fastcall WriteIntToFile(AnsiString file, int* array, int items)
{
    FILE* data;

    data = fopen(file.c_str(), "wt");
    for(register int i = 0; i < items; i++)
        fprintf(data, "%d\n", array[i]);
    fclose(data);
}

void __fastcall WriteIntToFile(AnsiString file, int array)
{
    FILE* data;

    data = fopen(file.c_str(), "wt");
    fprintf(data, "%d\n", array);
    fclose(data);
}

void __fastcall WriteInt64ToFile(AnsiString file, __int64* array, int items)
{
    FILE* data;

    data = fopen(file.c_str(), "wt");
    for(register int i = 0; i < items; i++)
        fprintf(data, "%Ld\n", array[i]);
    fclose(data);
}

void __fastcall WriteInt64ToFile(AnsiString file, __int64 array)
{
    FILE* data;

    data = fopen(file.c_str(), "wt");
    fprintf(data, "%Ld\n", array);
    fclose(data);
}

void __fastcall WriteDoubleToFile(AnsiString file, double* array, int items)
{
    FILE* data;

    data = fopen(file.c_str(), "wt");
    for(register int i = 0; i < items; i++)
        fprintf(data, "%.5E\n", array[i]);
    fclose(data);
}

/***** CLASE TMXPROP *****/
__fastcall TMxProp::TMxProp(int items)
{
    InitToNull();
    if(InitData(items) == false)
        TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
}

__fastcall TMxProp::~TMxProp()
{
}

```

```

    ReleaseData();
}

void __fastcall TMxProp::InitToNull(void)
{
    this->items = 0;
    this->absRep = NULL;
    this->vseabsRep = NULL;
    this->intRep = NULL;
    this->vseintRep = NULL;
    this->sqRep = NULL;
    this->vsesqRep = NULL;
    this->ssRep = NULL;
    this->vsessRep = NULL;
    this->szRep = NULL;
    this->vseszRep = NULL;
    this->dRep = NULL;
    this->vsedRep = NULL;
    this->wRep = NULL;
    this->vsewRep = NULL;
    this->lRep = NULL;
    this->vselRep = NULL;
    this->s2fRep = NULL;
    this->vses2fRep = NULL;
    this->c2fRep = NULL;
    this->vsec2fRep = NULL;
    this->released = true;
    InitVars();
}

void __fastcall TMxProp::InitVars(void)
{
    this->absMin = DBL_MAX;
    this->intMin = DBL_MAX;
    this->sqMin = DBL_MAX;
    this->ssMin = DBL_MAX;
    this->szMin = DBL_MAX;
    this->dMin = DBL_MAX;
    this->wMin = DBL_MAX;
    this->lMin = DBL_MAX;
    this->s2fMin = DBL_MAX;
    this->c2fMin = DBL_MAX;

    this->absMax = -DBL_MAX;
    this->intMax = -DBL_MAX;
    this->sqMax = -DBL_MAX;
    this->ssMax = -DBL_MAX;
    this->szMax = -DBL_MAX;
    this->dMax = -DBL_MAX;
    this->wMax = -DBL_MAX;
    this->lMax = -DBL_MAX;
    this->s2fMax = -DBL_MAX;
    this->c2fMax = -DBL_MAX;
}

bool __fastcall TMxProp::ReleaseData(void)
{
    AnsiString message;

    try
    {
        if(this->released)
            return true;
        delete[] this->absRep;
        delete[] this->vseabsRep;
        delete[] this->intRep;
        delete[] this->vseintRep;
        delete[] this->sqRep;
        delete[] this->vsesqRep;
        delete[] this->ssRep;
        delete[] this->vsessRep;
        delete[] this->szRep;
        delete[] this->vseszRep;
        delete[] this->dRep;
        delete[] this->vsedRep;
        delete[] this->wRep;
        delete[] this->vsewRep;
        delete[] this->lRep;
        delete[] this->vselRep;
        delete[] this->c2fRep;
        delete[] this->vsec2fRep;
        delete[] this->s2fRep;
        delete[] this->vses2fRep;
        this->released = true;
        InitToNull();
    }
    catch (Exception& e)

```

```

    {
        message = AnsiString(RELEASE_ERROR_TEXT) + "\n" + e.Message;
        return TMxUtils::ErrorMessageBox(message);
    }
    return true;
}

bool __fastcall TMxProp::InitData(int items)
{
    try
    {
        this->items = items;

        if((this->absRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vseabsRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->intRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vseintRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->sqRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vsesqRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->ssRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vsessRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->szRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vseszRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->dRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vsedRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->wRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vsewRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->lRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vselRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->s2fRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vses2fRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->c2fRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->vsec2fRep = new BYTE[this->items]) == NULL)
            return false;
        if((this->released = new BYTE[this->items]) == NULL)
            return false;

        // Inicializar a cero todos los arreglos.
        memset(this->absRep, 0, sizeof(BYTE) * this->items);
        memset(this->vseabsRep, 0, sizeof(BYTE) * this->items);
        memset(this->intRep, 0, sizeof(BYTE) * this->items);
        memset(this->vseintRep, 0, sizeof(BYTE) * this->items);
        memset(this->sqRep, 0, sizeof(BYTE) * this->items);
        memset(this->vsesqRep, 0, sizeof(BYTE) * this->items);
        memset(this->ssRep, 0, sizeof(BYTE) * this->items);
        memset(this->vsessRep, 0, sizeof(BYTE) * this->items);
        memset(this->szRep, 0, sizeof(BYTE) * this->items);
        memset(this->vseszRep, 0, sizeof(BYTE) * this->items);
        memset(this->dRep, 0, sizeof(BYTE) * this->items);
        memset(this->vsedRep, 0, sizeof(BYTE) * this->items);
        memset(this->wRep, 0, sizeof(BYTE) * this->items);
        memset(this->vsewRep, 0, sizeof(BYTE) * this->items);
        memset(this->lRep, 0, sizeof(BYTE) * this->items);
        memset(this->vselRep, 0, sizeof(BYTE) * this->items);
        memset(this->c2fRep, 0, sizeof(BYTE) * this->items);
        memset(this->vsec2fRep, 0, sizeof(BYTE) * this->items);
        memset(this->s2fRep, 0, sizeof(BYTE) * this->items);
        memset(this->vses2fRep, 0, sizeof(BYTE) * this->items);
    }
    catch (Exception& e)
    {
        return false;
    }
    return true;
}

__fastcall TMxSegments::TMxSegments(int items, int width, int height)
{
    InitToNull();
}

```

```

    if(InitData(items, width, height) == false)
        TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
}

__fastcall TMxSegments::~TMxSegments()
{
    ReleaseData();
}

void __fastcall TMxSegments::InitToNull(void)
{
    this->items = 0;
    this->rows = 0;
    this->columns = 0;
    this->intSum = NULL;
    this->absSum = NULL;
    this->iiSum = NULL;
    this->xSum = NULL;
    this->ySum = NULL;
    this->xxSum = NULL;
    this->yySum = NULL;
    this->xySum = NULL;
    this->trees = NULL;
    this->trees1 = NULL;
    this->sTrees = NULL;
    this->members = NULL;
    this->MT = NULL;
    this->TM = NULL;
    this->props = NULL;

    this->allintSum = 0;
    this->allxSum = 0;
    this->allySum = 0;
    this->allxxSum = 0;
    this->allyySum = 0;
    this->allxySum = 0;
    this->allabsSum = 0;
}

void __fastcall TMxSegments::InitAllSumVars(void)
{
    this->allintSum = 0;
    this->allxSum = 0;
    this->allySum = 0;
    this->allxxSum = 0;
    this->allyySum = 0;
    this->allxySum = 0;
    this->allabsSum = 0;
    this->props->InitVars();
}

bool __fastcall TMxSegments::InitData(int items, int width, int height)
{
    try
    {
        if((this->items = items) <= 0)
            return false;
        this->rows = height;
        this->columns = width;
        if((this->intSum = new int[this->items]) == NULL)
            return false;
        if((this->absSum = new int[this->items]) == NULL)
            return false;
        if((this->iiSum = new __int64[this->items]) == NULL)
            return false;
        if((this->xSum = new __int64[this->items]) == NULL)
            return false;
        if((this->ySum = new __int64[this->items]) == NULL)
            return false;
        if((this->xxSum = new __int64[this->items]) == NULL)
            return false;
        if((this->yySum = new __int64[this->items]) == NULL)
            return false;
        if((this->xySum = new __int64[this->items]) == NULL)
            return false;
        if((this->trees = new int[this->items]) == NULL)
            return false;
        if((this->trees1 = new int[this->items]) == NULL)
            return false;
        if((this->sTrees = new int[this->items]) == NULL)
            return false;
        if((this->members = new int[this->items]) == NULL)
            return false;
        if((this->props = new TMxProp(this->items)) == NULL)
            return false;
    }
    catch(Exception& e)

```

```

    {
        return false;
    }
    return true;
}

bool __fastcall TMxSegments::ReleaseData(void)
{
    AnsiString message;

    try
    {
        if(this->intSum != NULL)
            delete[] this->intSum;
        if(this->absSum != NULL)
            delete[] this->absSum;
        if(this->iiSum != NULL)
            delete[] this->iiSum;
        if(this->xSum != NULL)
            delete[] this->xSum;
        if(this->ySum != NULL)
            delete[] this->ySum;
        if(this->xxSum != NULL)
            delete[] this->xxSum;
        if(this->yySum != NULL)
            delete[] this->yySum;
        if(this->xySum != NULL)
            delete[] this->xySum;
        if(this->trees != NULL)
            delete[] this->trees;
        if(this->trees1 != NULL)
            delete[] this->trees1;
        if(this->sTrees != NULL)
            delete[] this->sTrees;
        if(this->members != NULL)
            delete[] this->members;
        if(this->props != NULL)
            delete[] this->props;
        InitToNull();
    }
    catch(Exception& e)
    {
        message = AnsiString(RELEASE_ERROR_TEXT) + "\n" + e.Message;
        return TMxUtils::ErrorMessageBox(message);
    }
    return true;
}

int __fastcall TMxSegments::InitSegment(int i, BYTE* matrix)
{
    try
    {
        this->sTrees[i] = this->trees[i] = this->trees1[i] = i;
        this->members[i] = 1;
        this->intSum[i] = (int)matrix[i];
        this->iiSum[i] = (__int64)matrix[i] * (__int64)matrix[i];
        this->xSum[i] = i % this->columns;
        this->ySum[i] = i / this->columns;
        this->xxSum[i] = this->xSum[i] * this->xSum[i];
        this->xySum[i] = this->xSum[i] * this->ySum[i];
        this->yySum[i] = this->ySum[i] * this->ySum[i];

        return this->intSum[i];
    }
    catch(Exception& e)
    {
        return (int)(TMxUtils::ErrorMessageBox(e.Message));
    }
}

int __fastcall TMxSegments::StartTreesSw2(int i, int segments, int* pSegments, BYTE* matrix)
{
    try
    {
        if(this->TM[i] > this->TM[i-1])
        {
            if((matrix[i] == matrix[i-1]))
            {
                if(TreeMerge(this->sTrees, i, i - 1))
                    (*pSegments)--;
                if((this->trees1[i] != this->trees1[i - 1]) && SwMergeTrees(this->trees, i, i - 1))
                    segments--;
            }
        }
        if(this->MT[i] > this->MT[i - 1])
        {
            if(matrix[this->MT[i]] == matrix[this->MT[i - 1]])

```

```

        {
            if((this->trees1[this->MT[i]] != this->trees1[this->MT[i - 1]]) &&
                SwMergeTrees(this->trees, this->MT[i], this->MT[i - 1]))
                segments--;
        }
    }
}
catch(Exception& e)
{
    return TMxUtils::ErrorMessageBox(e.Message);
}
return segments;
}

bool __fastcall TMxSegments::TreeMerge(int* trees, int a, int b)
{
    int index_a;
    int index_b;

    for(index_a = trees[a]; index_a != trees[index_a]; index_a = trees[index_a]);
    for(index_b = trees[b]; index_b != trees[index_b]; index_b = trees[index_b]);

    if(index_a < index_b)
    {
        trees[index_b] = index_a;
        return true;
    }
    else if(index_a > index_b)
    {
        trees[index_a] = index_b;
        return true;
    }
    return false;
}

bool __fastcall TMxSegments::SwMergeTrees(int* trees, int a, int b)
{
    int content;
    int index_a;
    int index_b;

    for(index_a = trees[a]; index_a != trees[index_a]; index_a = trees[index_a]);
    for(index_b = trees[b]; index_b != trees[index_b]; index_b = trees[index_b]);

    if(index_a < index_b)
    {
        content = trees[index_a];
        trees[index_b] = index_a;
        this->members[content] += this->members[index_b];
        this->intSum[content] += this->intSum[index_b];
        this->iiSum[index_a] += this->iiSum[index_b];
        this->xSum[content] += this->xSum[index_b];
        this->ySum[content] += this->ySum[index_b];
        this->xxSum[content] += this->xxSum[index_b];
        this->xySum[content] += this->xySum[index_b];
        this->yySum[content] += this->yySum[index_b];
        return true;
    }
    else if(index_a > index_b)
    {
        content = trees[index_b];
        trees[index_a] = index_b;
        this->members[content] += this->members[index_a];
        this->intSum[content] += this->intSum[index_a];
        this->iiSum[index_b] += this->iiSum[index_a];
        this->xSum[content] += this->xSum[index_a];
        this->ySum[content] += this->ySum[index_a];
        this->xxSum[content] += this->xxSum[index_a];
        this->xySum[content] += this->xySum[index_a];
        this->yySum[content] += this->yySum[index_a];
        return true;
    }
    return false;
}

void __fastcall TMxSegments::SetMatrixStructWS_1(int i, BYTE *matrix)
{
    this->sTrees[i] = this->sTrees[this->sTrees[i]];
    if(this->trees[i] == i)
    {
        this->trees1[i] = this->trees[i];
        matrix[i] = (BYTE)((this->intSum[i] + (this->members[i] >> 1)) / this->members[i]);
    }
    else if(this->trees[i] < i)
    {
        this->trees1[i] = this->trees1[this->trees[i]];
        matrix[i] = matrix[this->trees[i]];
    }
}

```

```

}
}

void __fastcall TMxSegments::InitMapMinDif(int i, BYTE *originMap, double *mapMinDif)
{
    if((i - this->sTrees[i]) >= 0)
        originMap[i] = 0;
    else
        originMap[i] = 255;
    this->sTrees[i] = i;
    mapMinDif[i] = DBL_MAX;
    if(this->trees1[i] == i)
        this->allintSum += this->intSum[i];
}

int __fastcall TMxSegments::SetMapMinDifWS1(int i, int seg, BYTE* matrix, BYTE* originMap, double* mapMinDif)
{
    BYTE byteData;
    double doubleData;

    if(this->TM[i] > this->TM[i - 1])
    {
        if(matrix[i] >= matrix[i-1])
            byteData = matrix[i] - matrix[i-1];
        else
            byteData = matrix[i - 1] - matrix[i];
        doubleData = (double)byteData;
        if(byteData != 0)
        {
            if(doubleData < mapMinDif[i])
                mapMinDif[i] = doubleData;
            if(doubleData < mapMinDif[i - 1])
                mapMinDif[i - 1] = doubleData;
        }
    }
    if(this->MT[i] > this->MT[i-1])
    {
        if((matrix[this->MT[i]] == matrix[MT[i - 1]]) && (originMap[this->MT[i]] == originMap[MT[i - 1]]))
            if(TreeMerge(this->sTrees, this->MT[i], this->MT[i - 1]))
                seg--;
        if(matrix[this->MT[i]] >= matrix[this->MT[i - 1]])
            byteData = matrix[this->MT[i]] - matrix[this->MT[i - 1]];
        else
            byteData = matrix[this->MT[i - 1]] - matrix[this->MT[i];
        doubleData = (double)byteData;
        if(byteData != 0)
        {
            if(doubleData < mapMinDif[this->MT[i]])
                mapMinDif[this->MT[i]] = doubleData;
            if(doubleData < mapMinDif[this->MT[i - 1]])
                mapMinDif[this->MT[i - 1]] = doubleData;
        }
    }
    return seg;
}

void __fastcall TMxSegments::SetAdditives(int i, double* mapMinDif)
{
    int j = this->items - i - 1;
    TMxDaiMetric metric;
    double dval;

    this->sTrees[j] = this->sTrees[this->sTrees[j]];

    if(this->trees[i] != i)
    {
        if(mapMinDif[this->trees[i]] > mapMinDif[i])
            mapMinDif[this->trees[i]] = mapMinDif[i];
    }
    else
    {
        this->absSum[i] = (int) (((__int64)this->intSum[i] - ((__int64)this->allintSum *
            (__int64)this->members[i] + (__int64)(this->items / 2)) / this->items));
        if(this->absSum[i] < 0)
            this->absSum[i] = -(this->absSum[i]);

        this->allabsSum += this->absSum[i];
        this->allxSum += this->xSum[i];
        this->allySum += this->ySum[i];
        this->allxxSum += this->xxSum[i];
        this->allxySum += this->xySum[i];
        this->allyySum += this->yySum[i];

        dval = (double)this->absSum[i] / (double)this->members[i];
        if(this->props->absMin > dval)
            this->props->absMin = dval;
        if(this->props->absMax < dval)

```

```

        this->props->absMax = dval;

    dval = (double)this->intSum[i] / (double)this->members[i];
    if(this->props->intMin > dval)
        this->props->intMin = dval;
    if(this->props->intMax < dval)
        this->props->intMax = dval;

    dval = (double)members[i];
    if(this->props->sqMin > dval)
        this->props->sqMin = (double)dval;
    if(this->props->sqMax < dval)
        this->props->sqMax = (double)dval;

    DaiMetric(i, &metric);

    if(this->props->ssMin > metric.ss)
        this->props->ssMin = metric.ss;
    if(this->props->ssMax < metric.ss)
        this->props->ssMax = metric.ss;

    if(this->props->szMin > metric.size)
        this->props->szMin = metric.size;
    if(this->props->szMax < metric.size)
        this->props->szMax = metric.size;

    if(this->props->dMin > metric.d)
        this->props->dMin = metric.d;
    if(this->props->dMax < metric.d)
        this->props->dMax = metric.d;

    if(this->props->wMin > metric.width)
        this->props->wMin = metric.width;
    if(this->props->wMax < metric.width)
        this->props->wMax = metric.width;

    if(this->props->lMin > metric.length)
        this->props->lMin = metric.length;
    if(this->props->lMax < metric.length)
        this->props->lMax = metric.length;

    if(this->props->s2fMin > metric.s2f)
        this->props->s2fMin = metric.s2f;
    if(this->props->s2fMax < metric.s2f)
        this->props->s2fMax = metric.s2f;

    if(this->props->c2fMin > metric.c2f)
        this->props->c2fMin = metric.c2f;
    if(this->props->c2fMax < metric.c2f)
        this->props->c2fMax = metric.c2f;
    }
}

int __fastcall TMxSegments::Normalize(double min, double max, double val)
{
    double middle;
    int result;

    middle = ((double)SMALLEST * (max - val)) + ((double)LARGEST * (val - min));
    if(min == max)
    {
        if(middle == 0.)
            return 0;
        else if(middle > 0.)
            return LARGEST;
        return SMALLEST;
    }
    result = (int)((middle + (max - min) / 2) / (max - min));
    return result;
}

void __fastcall TMxSegments::SpreadWSv(int i, int iter, BYTE* originMap, double* mapMinDif)
{
    TMxDaiMetric metric;
    double dval;

    if((i - this->sTrees[i]) < 0)
        originMap[i] = 255;
    this->sTrees[i] = i;

    if(this->trees[i] == i)
    {
        dval = (double)this->members[i];
        if(Normalize(this->props->sqMin, this->props->sqMax, dval) < this->percent)
            this->props->sqRep[i] = 0;
        else

```

```

        this->props->sqRep[i] = 255;

dval = (double)this->absSum[i] / (double)this->members[i];
if(Normalize(this->props->absMin, this->props->absMax, dval) < this->percent)
    this->props->absRep[i] = 0;
else
    this->props->absRep[i] = 255;

dval = (double)this->intSum[i] / (double)this->members[i];
if(Normalize(this->props->intMin, this->props->intMax, dval) < this->percent)
    this->props->intRep[i] = 0;
else
    this->props->intRep[i] = 255;

DaiMetric(i, &metric);

if(Normalize(this->props->ssMin, this->props->ssMax, metric.ss) < this->percent)
    this->props->ssRep[i] = 0;
else
    this->props->ssRep[i] = 255;

if(Normalize(this->props->szMin, this->props->szMax, metric.size) < this->percent)
    this->props->szRep[i] = 0;
else
    this->props->szRep[i] = 255;
if(Normalize(this->props->dMin, this->props->dMax, metric.d) < this->percent)
    this->props->dRep[i] = 0;
else
    this->props->dRep[i] = 255;

if(Normalize(this->props->wMin, this->props->wMax, metric.width) < this->percent)
    this->props->wRep[i] = 0;
else
    this->props->wRep[i] = 255;

if(Normalize(this->props->lMin, this->props->lMax, metric.length) < this->percent)
    this->props->lRep[i] = 0;
else
    this->props->lRep[i] = 255;

if(Normalize(this->props->s2fMin, this->props->s2fMax, metric.s2f) < this->percent)
    this->props->s2fRep[i] = 0;
else
    this->props->s2fRep[i] = 255;

if(Normalize(this->props->c2fMin, this->props->c2fMax, metric.c2f) < this->percent)
    this->props->c2fRep[i] = 0;
else
    this->props->c2fRep[i] = 255;
}
else if(this->trees[i] < i)
{
    mapMinDif[i] = mapMinDif[this->trees[i]];
    this->props->absRep[i] = this->props->absRep[this->trees[i]];
    this->props->intRep[i] = this->props->intRep[this->trees[i]];
    this->props->sqRep[i] = this->props->sqRep[this->trees[i]];
    this->props->ssRep[i] = this->props->ssRep[this->trees[i]];
    this->props->szRep[i] = this->props->szRep[this->trees[i]];
    this->props->dRep[i] = this->props->dRep[this->trees[i]];
    this->props->wRep[i] = this->props->wRep[this->trees[i]];
    this->props->lRep[i] = this->props->lRep[this->trees[i]];
    this->props->s2fRep[i] = this->props->s2fRep[this->trees[i]];
    this->props->c2fRep[i] = this->props->c2fRep[this->trees[i]];
}

if(iter <= 8)
{
    this->props->vs2fRep[i] *= 2;
    if(this->props->s2fRep[i] == 255)
        (this->props->vs2fRep[i])++;
    this->props->vsec2fRep[i] *= 2;
    if(this->props->c2fRep[i] == 255)
        (this->props->vsec2fRep[i])++;
    this->props->vselRep[i] *= 2;
    if(this->props->lRep[i] == 255)
        (this->props->vselRep[i])++;
    this->props->vsewRep[i] *= 2;
    if(this->props->wRep[i] == 255)
        (this->props->vsewRep[i])++;
    this->props->vsedRep[i] *= 2;
    if(this->props->dRep[i] == 255)
        (this->props->vsedRep[i])++;
    this->props->vseszRep[i] *= 2;
    if(this->props->szRep[i] == 255)
        (this->props->vseszRep[i])++;
    this->props->vsessRep[i] *= 2;
    if(this->props->ssRep[i] == 255)

```

```

        (this->props->vsessRep[i])++;
        this->props->vsesqRep[i] *= 2;
        if(this->props->sqRep[i] == 255)
            (this->props->vsesqRep[i])++;
        this->props->vseabsRep[i] *= 2;
        if(this->props->absRep[i] == 0)
            (this->props->vseabsRep[i])++;
        this->props->vseintRep[i] *= 2;
        if(this->props->intRep[i] == 255)
            (this->props->vseintRep[i])++;
    }
}

int __fastcall TMxSegments::SetTreesWS1(int i, int segments, BYTE *matrix, double *mapMinDif)
{
    BYTE byteData;
    double doubleData;

    if(this->TM[i] > this->TM[i-1])
    {
        if(matrix[i] >= matrix[i-1])
            byteData = matrix[i] - matrix[i - 1];
        else
            byteData = matrix[i - 1] - matrix[i];
        doubleData = (double)(byteData);

        if((doubleData == mapMinDif[i]) || (doubleData == mapMinDif[i - 1]))
            if(TreeMerge(this->trees, i, i-1))
                segments--;
    }
    if(this->MT[i] > this->MT[i-1])
    {
        if(matrix[this->MT[i]] >= matrix[this->MT[i-1]])
            byteData = matrix[this->MT[i]] - matrix[this->MT[i - 1]];
        else
            byteData = matrix[this->MT[i - 1]] - matrix[this->MT[i]];
        doubleData = (double)byteData;
        if((doubleData == mapMinDif[this->MT[i]]) || (doubleData == mapMinDif[this->MT[i-1]]))
            if(TreeMerge(this->trees, this->MT[i], this->MT[i - 1]))
                segments--;
    }
    return segments;
}

void __fastcall TMxSegments::SetMatrixStructWS10(int i)
{
    if(this->trees[i] < i)
    {
        if(this->trees1[i] == i)
        {
            this->trees1[i] = this->trees1[this->trees[i]];
            this->members[this->trees1[i]] += this->members[i];
            this->intSum[this->trees1[i]] += this->intSum[i];
            this->iiSum[this->trees1[i]] += this->iiSum[i];
            this->xSum[this->trees1[i]] += this->xSum[i];
            this->ySum[this->trees1[i]] += this->ySum[i];
            this->xxSum[this->trees1[i]] += this->xxSum[i];
            this->xySum[this->trees1[i]] += this->xySum[i];
            this->yySum[this->trees1[i]] += this->yySum[i];
        }
        else
            this->trees1[i] = this->trees1[this->trees[i]];
    }
}

void __fastcall TMxSegments::SetMatrixStructWS11(int i, BYTE* matrix)
{
    if(this->trees[i] == i)
        matrix[i] = (BYTE)((this->intSum[i] + (this->members[i] >> 1)) / this->members[i]);
    else if(this->trees[i] < i)
        matrix[i] = matrix[this->trees[i]];
}

void __fastcall TMxSegments::SetIntRep(__int64 sqMax, BYTE value)
{
    for(int i = 0; i != this->items; i++)
    {
        if(this->members[this->trees[i]] == sqMax)
        {
            this->props->vses2fRep[i] =
            this->props->vsec2fRep[i] =
            this->props->vsewRep[i] =
            this->props->vselRep[i] =
            this->props->vsedRep[i] =
            this->props->vseszRep[i] =
            this->props->vsessRep[i] =
            this->props->vsesqRep[i] = value;
        }
    }
}

```

```

    }
}

__fastcall TMxCompImgRep::TMxCompImgRep(TMxImage* srcImage, TMxComponent component)
{
    InitToNull();
    InitData(srcImage, component);
    SetValues(srcImage, component);
}

__fastcall TMxCompImgRep::~TMxCompImgRep()
{
    ReleaseData();
}

void __fastcall TMxCompImgRep::InitToNull(void)
{
    this->items = 0;
    this->rows = 0;
    this->columns = 0;
    this->percent = 100;
    this->segments = NULL;
    this->mapMinDif = NULL;
    this->source = NULL;
    this->matrix = NULL;
    this->repres = NULL;
    this->originMap = NULL;
    this->vseRepres = NULL;
    this->MT = NULL;
    this->TM = NULL;
}

bool __fastcall TMxCompImgRep::InitData(TMxImage* srcImage, TMxComponent component)
{
    AnsiString message;

    try
    {
        this->rows = srcImage->Height;
        this->columns = srcImage->Width;
        this->items = this->rows * this->columns;
        this->percent = 100;
        this->component = component;

        if((this->segments = new TMxSegments(this->items, this->columns, this->rows)) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->mapMinDif = new double[this->items]) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->source = new BYTE[this->items]) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->matrix = new BYTE[this->items]) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->repres = new BYTE[this->items]) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->originMap = new BYTE[this->items]) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->vseRepres = new BYTE[this->items]) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
    }
    catch(Exception& e)
    {
        message = AnsiString(MEMORY_ERROR_TEXT) + "\n" + e.Message;
        return TMxUtils::ErrorMessageBox(message);
    }
    return true;
}

bool __fastcall TMxCompImgRep::ReleaseData(void)
{
    AnsiString message;

    try
    {
        if(this->segments != NULL)
            delete[] this->segments;
        if(this->mapMinDif != NULL)
            delete[] this->mapMinDif;
        if(this->source != NULL)
            delete[] this->source;
        if(this->matrix != NULL)
            delete[] this->matrix;
        if(this->repres != NULL)
            delete[] this->repres;
        if(this->originMap != NULL)
            delete[] this->originMap;
        if(this->vseRepres != NULL)

```

```

        delete[] this->vseRepres;
        InitToNull();
    }
    catch(Exception& e)
    {
        message = AnsiString(RELEASE_ERROR_TEXT) + "\n" + e.Message;
        return TMxUtils::ErrorMessageBox(message);
    }
    return true;
}

bool __fastcall TMxCompImgRep::SetValues(TMxImage* srcImage, TMxComponent component)
{
    AnsiString message;
    BYTE grayLevel;
    int i = 0;

    try
    {
        for(register int y = 0; y < srcImage->Height; y++)
        {
            for(register int x = 0; x < srcImage->Width; x++)
            {
                grayLevel = srcImage->GetLevel(component, x, y);
                this->source[i] = grayLevel;
                this->matrix[i] = grayLevel;
                this->segments->InitSegment(i++, this->matrix);
            }
        }
    }
    catch(Exception& e)
    {
        message = AnsiString(INIT_ALGORITHM_ERROR_TEXT) + "\n" + e.Message;
        return false;
    }
    return true;
}

int __fastcall TMxCompImgRep::StartTreesSw2(int i, int segments, int* pSegments)
{
    return this->segments->StartTreesSw2(i, segments, pSegments, this->matrix);
}

void __fastcall TMxCompImgRep::SetMatrixStructWS_1(int i)
{
    this->segments->SetMatrixStructWS_1(i, this->matrix);
}

void __fastcall TMxCompImgRep::InitMapMinDif(int i)
{
    this->segments->InitMapMinDif(i, this->originMap, this->mapMinDif);
}

int __fastcall TMxCompImgRep::SetMapMinDifWS1(int i, int segments)
{
    return this->segments->SetMapMinDifWS1(i, segments, this->matrix, this->originMap, this->mapMinDif);
}

void __fastcall TMxCompImgRep::SetAdditives(int i)
{
    this->segments->SetAdditives(i, this->mapMinDif);
}

void __fastcall TMxCompImgRep::SpreadWSv(int i, int iter)
{
    this->segments->SpreadWSv(i, iter, this->originMap, this->mapMinDif);
}

int __fastcall TMxCompImgRep::SetTreesWS1(int i, int segments)
{
    return this->segments->SetTreesWS1(i, segments, this->matrix, this->mapMinDif);
}

void __fastcall TMxCompImgRep::SetMatrixStructWS10(int i)
{
    this->segments->SetMatrixStructWS10(i);
}

void __fastcall TMxCompImgRep::SetMatrixStructWS11(int i)
{
    this->segments->SetMatrixStructWS11(i, this->matrix);
}

void __fastcall TMxCompImgRep::SetIntRep(__int64 sqMax)
{
    if(this->component == rgbRed)
        this->segments->SetIntRep(sqMax, 255);
}

```

```

else
    this->segments->SetIntRep(sqMax, 255);
}

void __fastcall TMxCompImgRep::InitToNullAbsSum(void)
{
    memset(this->segments->absSum, 0, sizeof(int) * this->items);
}

void __fastcall TMxCompImgRep::InitToNullAllIntSum(void)
{
    this->segments->allIntSum = 0;
}

void __fastcall TMxCompImgRep::InitAllSumVars(void)
{
    this->segments->InitAllSumVars();
}

__fastcall TMxRGBCompImgRep::TMxRGBCompImgRep(TMxImage* srcImage) : TMxMulImgAlgorithm(srcImage)
{
    InitToNull();
    InitData();
}

__fastcall TMxRGBCompImgRep::~TMxRGBCompImgRep()
{
    ReleaseData();
}

void __fastcall TMxRGBCompImgRep::InitToNull(void)
{
    this->items = 0;
    this->columns = 0;
    this->rows = 0;
    this->percent = 100;
    this->redIntSum = 0;
    this->greenIntSum = 0;
    this->blueIntSum = 0;

    this->redCompImgRep = NULL;
    this->greenCompImgRep = NULL;
    this->blueCompImgRep = NULL;
    this->MT = NULL;
    this->TM = NULL;
}

bool __fastcall TMxRGBCompImgRep::InitData(void)
{
    AnsiString message;

    try
    {
        this->items = this->width * this->height;
        this->rows = this->height;
        this->columns = this->width;

        // Arreglos de la clase.
        if((this->MT = new int[this->items]) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->TM = new int[this->items]) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        // Objetos que contiene la compresión semántica por cada componente de color.
        if((this->redCompImgRep = new TMxCompImgRep(this->srcImage, rgbRed)) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->greenCompImgRep = new TMxCompImgRep(this->srcImage, rgbGreen)) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        if((this->blueCompImgRep = new TMxCompImgRep(this->srcImage, rgbBlue)) == NULL)
            return TMxUtils::ErrorMessageBox(MEMORY_ERROR_TEXT);
        return SetValues();
    }
    catch(Exception& e)
    {
        message = AnsiString(MEMORY_ERROR_TEXT) + "\n" + e.Message;
        return TMxUtils::ErrorMessageBox(message);
    }
}

bool __fastcall TMxRGBCompImgRep::SetValues(void)
{
    register int i;

    try
    {
        for(int i = 0; i < this->items; i++)
        {
            this->MT[i] = (i % this->rows) * this->columns + i / this->rows;
            this->TM[i] = (i % this->columns) * this->rows + i / this->columns;
        }
    }
}

```

```

    }

    this->redCompImgRep->SetMT(this->MT);
    this->greenCompImgRep->SetMT(this->MT);
    this->blueCompImgRep->SetMT(this->MT);

    this->redCompImgRep->SetTM(this->TM);
    this->greenCompImgRep->SetTM(this->TM);
    this->blueCompImgRep->SetTM(this->TM);

    this->redCompImgRep->Percent = this->percent;
    this->greenCompImgRep->Percent = this->percent;
    this->blueCompImgRep->Percent = this->percent;
}
catch(Exception& e)
{
    return TMxUtils::ErrorMessageBox(e.Message);
}
return true;
}

bool __fastcall TMxRGBCompImgRep::ReleaseData(void)
{
    AnsiString message;

    try
    {
        if(this->redCompImgRep != NULL)
            delete this->redCompImgRep;
        if(this->greenCompImgRep != NULL)
            delete this->greenCompImgRep;
        if(this->blueCompImgRep != NULL)
            delete this->blueCompImgRep;
        if(this->MT != NULL)
            delete[] this->MT;
        if(this->TM != NULL)
            delete[] this->TM;
        InitToNull();
    }
    catch(Exception& e)
    {
        message = AnsiString(RELEASE_ERROR_TEXT) + "\n" + e.Message;
        return TMxUtils::ErrorMessageBox(message);
    }
    return true;
}

bool __fastcall TMxRGBCompImgRep::InitAlgorithm(void)
{
    return true;
}

bool __fastcall TMxRGBCompImgRep::Algorithm(void)
{
    int segmentsR, segments1R, oldSegmentsR;
    int segmentsG, segments1G, oldSegmentsG;
    int segmentsB, segments1B, oldSegmentsB;
    int redSegments, greenSegments, blueSegments;
    int VSEsegMax[256], iter = 0;
    int* tree, *members;
    register int i;
    AnsiString message;
    __int64 sqMax;

    try
    {
        WriteByteToFile("matrix_g.txt", this->greenCompImgRep->matrix, this->items);

        // tree = new int[this->items];
        // members = new int[this->items];
        segmentsR = segmentsG = segmentsB = this->items;
        while((segmentsR != 1) || (segmentsG != 1) || (segmentsB != 1))
        {
            iter++;
            oldSegmentsR = segmentsR;
            oldSegmentsG = segmentsG;
            oldSegmentsB = segmentsB;

            this->redCompImgRep->InitToNullAbsSum();
            this->greenCompImgRep->InitToNullAbsSum();
            this->blueCompImgRep->InitToNullAbsSum();

            DebugFiles("Init", iter);

            segments1R = segments1G = segments1B = this->items;
            for(int i = this->items - 1; i != 0; i--)
            {

```

```

        segmentsR = this->redCompImgRep->StartTreesSw2(i, segmentsR, &segments1R);
        segmentsG = this->greenCompImgRep->StartTreesSw2(i, segmentsG, &segments1G);
        segmentsB = this->blueCompImgRep->StartTreesSw2(i, segmentsB, &segments1B);
    }
    DebugFiles("StartTreesSw2", iter);

    if(segmentsR < segmentsG)
        VSEsegMaX[iter - 1] = max(segmentsG, segmentsB);
    else
        VSEsegMaX[iter - 1] = max(segmentsR, segmentsB);
    for(i = 0; i != this->items; i++)
    {
        this->redCompImgRep->SetMatrixStructWS_1(i);
        this->greenCompImgRep->SetMatrixStructWS_1(i);
        this->blueCompImgRep->SetMatrixStructWS_1(i);
    }
    DebugFiles("SetMatrixStructWS_1", iter);

    this->redCompImgRep->InitAllSumVars();
    this->greenCompImgRep->InitAllSumVars();
    this->blueCompImgRep->InitAllSumVars();
    for(i = 0; i != this->items; i++)
    {
        this->redCompImgRep->InitMapMinDif(i);
        this->greenCompImgRep->InitMapMinDif(i);
        this->blueCompImgRep->InitMapMinDif(i);
    }
    redSegments = greenSegments = blueSegments = this->items;
    for(i = this->items - 1; i != 0; i--)
    {
        redSegments = this->redCompImgRep->SetMapMinDifWS1(i, redSegments);
        greenSegments = this->greenCompImgRep->SetMapMinDifWS1(i, greenSegments);
        blueSegments = this->blueCompImgRep->SetMapMinDifWS1(i, blueSegments);
    }
    for(i = this->items - 1; i != -1; i--)
    {
        this->redCompImgRep->SetAdditives(i);
        this->greenCompImgRep->SetAdditives(i);
        this->blueCompImgRep->SetAdditives(i);
    }
    sqMax = this->redCompImgRep->GetSQMax();

    for(i = 0; i != this->items; i++)
    {
        this->redCompImgRep->SpreadWSv(i, iter);
        this->greenCompImgRep->SpreadWSv(i, iter);
        this->blueCompImgRep->SpreadWSv(i, iter);
    }
    for(i = this->items - 1; i != 0; i--)
    {
        segmentsR = this->redCompImgRep->SetTreesWS1(i, segmentsR);
        segmentsG = this->greenCompImgRep->SetTreesWS1(i, segmentsG);
        segmentsB = this->blueCompImgRep->SetTreesWS1(i, segmentsB);
    }
    for(i = 0; i != items; i++)
    {
        this->redCompImgRep->SetMatrixStructWS10(i);
        this->greenCompImgRep->SetMatrixStructWS10(i);
        this->blueCompImgRep->SetMatrixStructWS10(i);
    }
    Application->ProcessMessages();
}
}
catch(Exception& e)
{
    return TMxUtils::ErrorMessageBox(e.Message);
}
return true;
}

void __fastcall TMxRGBCompImgRep::DebugFiles(AnsiString path, int iter)
{
    try
    {
        // Depuracion
        AnsiString name = path + AnsiString("\\matrix") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->greenCompImgRep->matrix, this->items);

        name = path + AnsiString("\\originMap") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->greenCompImgRep->originMap, this->items);
        name = path + AnsiString("\\mapMinDif") + AnsiString(iter) + ".g.txt";
        WriteDoubleToFile(name.c_str(), this->greenCompImgRep->mapMinDif, this->items);

        name = path + AnsiString("\\intsum") + AnsiString(iter) + ".g.txt";
        WriteIntToFile(name.c_str(), this->greenCompImgRep->segments->intSum, this->items);
        name = path + AnsiString("\\iisum") + AnsiString(iter) + ".g.txt";
        WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->iiSum, this->items);
    }
}

```

```

name = path + AnsiString("\\xsum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->xSum, this->items);
name = path + AnsiString("\\ysum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->ySum, this->items);
name = path + AnsiString("\\xysum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->xySum, this->items);
name = path + AnsiString("\\xxsum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->xxSum, this->items);
name = path + AnsiString("\\yysum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->yySum, this->items);
name = path + AnsiString("\\members") + AnsiString(iter) + ".g.txt";
WriteIntToFile(name.c_str(), this->greenCompImgRep->segments->members, this->items);

name = path + AnsiString("\\allintSum") + AnsiString(iter) + ".g.txt";
WriteIntToFile(name.c_str(), this->greenCompImgRep->segments->allintSum);
name = path + AnsiString("\\allabsSum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->allabsSum);
name = path + AnsiString("\\allxSum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->allxSum);
name = path + AnsiString("\\allySum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->allySum);
name = path + AnsiString("\\allxxSum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->allxxSum);
name = path + AnsiString("\\allxySum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->allxySum);
name = path + AnsiString("\\allyySum") + AnsiString(iter) + ".g.txt";
WriteInt64ToFile(name.c_str(), this->greenCompImgRep->segments->allyySum);

name = path + AnsiString("\\trees_") + AnsiString(iter) + ".g.txt";
WriteIntToFile(name.c_str(), this->greenCompImgRep->segments->trees, this->items);
name = path + AnsiString("\\strees_") + AnsiString(iter) + ".g.txt";
WriteIntToFile(name.c_str(), this->greenCompImgRep->segments->sTrees, this->items);
name = path + AnsiString("\\trees1_") + AnsiString(iter) + ".g.txt";
WriteIntToFile(name.c_str(), this->greenCompImgRep->segments->trees1, this->items);
}
catch(Exception& e)
{
    TMxUtils::ErrorMessageBox("Se produjo un error al escribir los archivos");
}
}

void __fastcall TMxProp::DebugFiles(AnsiString path, int iter)
{
    AnsiString name;

    try
    {
        name = path + AnsiString("\\absRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->absRep, this->items);
        name = path + AnsiString("\\intRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->intRep, this->items);
        name = path + AnsiString("\\sqRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->sqRep, this->items);
        name = path + AnsiString("\\ss") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->ssRep, this->items);
        name = path + AnsiString("\\szRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->szRep, this->items);
        name = path + AnsiString("\\dRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->dRep, this->items);
        name = path + AnsiString("\\wRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->wRep, this->items);
        name = path + AnsiString("\\lRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->lRep, this->items);
        name = path + AnsiString("\\s2fRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->s2fRep, this->items);
        name = path + AnsiString("\\c2fRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->c2fRep, this->items);

        name = path + AnsiString("\\vseabsRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vseabsRep, this->items);
        name = path + AnsiString("\\vseintRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vseintRep, this->items);
        name = path + AnsiString("\\vsesqRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vsesqRep, this->items);
        name = path + AnsiString("\\vsess") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vsessRep, this->items);
        name = path + AnsiString("\\vseszRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vseszRep, this->items);
        name = path + AnsiString("\\vsedRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vsedRep, this->items);
        name = path + AnsiString("\\vsewRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vsewRep, this->items);
        name = path + AnsiString("\\vselRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vselRep, this->items);
        name = path + AnsiString("\\vses2fRep") + AnsiString(iter) + ".g.txt";
        WriteByteToFile(name.c_str(), this->vses2fRep, this->items);
        name = path + AnsiString("\\vsec2fRep") + AnsiString(iter) + ".g.txt";
    }
}

```

```
        WriteByteToFile(name.c_str(), this->vsec2fRep, this->items);
    }
    catch (Exception& e)
    {
        TmxUtils::ErrorMessageBox("Se produjo un error al escribir los archivos");
    }
}
```