



INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

**Segmentación en tiempo real del
movimiento en secuencias de imágenes
digitales**

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

Ing. René Montoya Flores

DIRECTOR DE TESIS

Dr. Edgardo Manuel Felipe Riverón



Ciudad de México
Enero de 2016





SIP-14

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 10:00 horas del día 15 del mes de diciembre de 2015 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

Segmentación en tiempo real del movimiento en secuencias de imágenes digitales

Presentada por el alumno:

MONTOYA
Apellido paterno

FLORES
Apellido materno

RENÉ
Nombre(s)

Con registro:

B	1	3	0	1	1	3
---	---	---	---	---	---	---

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de Tesis

Dr. Edgardo Manuel Felipe Riverón

Dr. Oleksiy Pogrebnyak

Dr. René Luna García

Dr. Ricardo Barrón Fernández

Dr. Jesús Guillermo Figueroa Nazuno

Dr. Marco Antonio Moreno Ibarra

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 14 del mes Enero del año 2016, el (la) que suscribe René Montoya Flores alumno (a) del Programa de Maestría en ciencias de la computación con número de registro B130113, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Edgardo Manuel Felipe Riverón y cede los derechos del trabajo intitulado Segmentación en tiempo real del movimiento en secuencias de imágenes digitales, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección ranarene@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

René Montoya Flores 

Nombre y firma

Resumen

Este trabajo de tesis, ha sido realizado con el fin de crear un sistema de videovigilancia, que permite segmentar el movimiento detectado en secuencias de video; para ello fue necesario el uso de algunas herramientas como la morfología matemática, codificación y decodificación de video, visión por computadora además de algunas otras disciplinas;. Así mismo, también se desarrolló un algoritmo para generar un reporte, en donde se detallan las características de cada objeto móvil detectado.

En este trabajo, se presenta un nuevo método para la segmentación del movimiento y caracterización de los objetos detectados en secuencias de imágenes. Para segmentar el movimiento, el primer paso es generar la imagen de fondo que será utilizada para comparar la secuencia de imágenes, después se realiza una diferencia de imágenes entre cada imagen de la secuencia y la imagen de fondo en cada capa de color, y se aplica un umbral de diferencia, teniendo como resultado tres matrices binarias. Al trabajar cada capa por separado también se le puede asignar un peso mayor o menor a alguna capa si las condiciones así lo requiriesen. El siguiente paso es utilizar un proceso morfológico en cada matriz para realizar la supresión de los artefactos; a continuación se realiza una operación OR entre las tres imágenes para obtener la ubicación de los pixeles pertenecientes a objetos en movimiento. Los datos de los objetos detectados son almacenados en un archivo de texto y después son procesados para generar un reporte acerca de ellos; se analizan y se correlacionan estos datos para poder realizar el seguimiento de cada uno de los objetos móviles detectados; finalmente se genera un reporte con los datos encontrados.

Este algoritmo de segmentación de movimiento se presenta con aplicaciones de videovigilancia, pero tiene un amplio número de otras aplicaciones. Los resultados demuestran que es posible la implementación del algoritmo en tiempo real para la segmentación del movimiento y escritura de los datos generados; Además es compatible con la mayoría de los sistemas actuales de videovigilancia

Abstract

This thesis work has been developed as a video surveillance system, it achieves motion segmentation, and it uses some tools as mathematical morphology, video codecs, computer vision and some other disciplines; accordingly, an algorithm to generate a report about each detected element has been developed.

In this thesis, a new method to achieve motion segmentation and objects characterization in image sequences is presented. The first step to get motion segmentation, is to generate the background image which will be used to compare the image sequence, next, an image difference between each image sequence and the background image is carried through, in this image difference a threshold is set in order to get better results, this way, three binaries matrixes are resulted from the process. So, by processing each layer, it is possible to weight each one of these if necessary. Then, a morphological process is used to suppress noise. The next step is to put the three binary layers together with an OR operation in order to detect the pixels that belong to moving objects. The specific data belonging to these objects are saved in a text file to be processed later. So this information is analyzed and correlated to be able to generate the tracking of each one of these detected objects, then, a report is generated.

As already mentioned, this motion segmentation algorithm is shown with video surveillance applications, but a wide applicability range is in fact possible. Results show that its implementation in real time for achieving motion segmentation and data writing is possible. It is also compatible with most of the actual surveillance systems.

Agradecimientos

Principalmente quiero agradecer a mi casa de estudios, el Instituto Politécnico Nacional, por haberme preparado desde joven hasta ahora que termino mi maestría. Gracias IPN por ser la más bella casa de estudios.

Quiero dar gracias al Centro de Investigación en Computación, por la oportunidad que me ha dado para seguir mejorando mi preparación académica y por permitirme expandir mis conocimientos. También quiero agradecer a los profesores del CIC. Gracias a su motivación, al arduo trabajo que nos ponen como ejemplo todos los días, a su profesionalismo y el empeño que ponen en prepararnos como mejores profesionistas.

Connie 1.0, gracias por todo.

Con todo mi corazón, dedico este trabajo a Male, Ramos, Rube y Lili, *in Memoriam* a mi padre.

También dedico este trabajo a todas las personas que sin distinción de raza, nacionalidad o credo, hacen de este mundo un mejor lugar para vivir

Índice general

SIP-14	i
Carta cesión de derechos de autor	ii
Resumen	iii
Abstract	iv
Agradecimientos	v
Índice general	vi
Lista de figuras	ix
Lista de siglas y acrónimos	xii
Lista de tablas	xiii
Glosario de términos	xiv
Acerca de los términos “tiempo real” y “segmentación de movimiento”	xvi
Capítulo 1. Introducción a la segmentación del movimiento	1
1.1. El problema a resolver	2
1.2. Justificación	2
1.3. Hipótesis	3
1.4. Objetivos	4
1.4.1. Objetivo general	4
1.4.2. Objetivos particulares	4
1.5. Consideraciones generales	5
1.6. Medios utilizados	5
1.7. Organización del documento	5
Capítulo 2. Estado del arte	7
2.1. Introducción	8
2.2. Comparación de Imágenes	10
2.2.1. Diferencia de imágenes (FD)	10
2.2.2. Sustracción del fondo	12
2.3. Trabajos relacionados con la segmentación de movimiento	14
2.3.1. Segmentación de movimiento por sustracción de fondo y análisis temporal	14
2.3.2. Diferencia de imágenes con sustracción del fondo basado en diferencia dinámica	16
2.3.3. Uso de herramientas morfológicas	19

2.3.4.	Detección humana basada en la detección de bordes de Canny y FD	20
2.3.5.	Segmentación de video con FD de 3 imágenes y operador de Kirsch	23
2.3.6.	Implementación de técnicas convencionales con modificaciones	25
2.3.7.	Segmentación de movimiento y rastreo usando cortes normalizados.	26
2.3.8.	Una revalorización de FD en detección de movimiento rápida y robusta.	30
2.4.	Conclusiones del capítulo	32
Capítulo 3.	Conceptos básicos y definiciones	33
3.1.	Procesamiento digital de imágenes	34
3.2.	Historia	34
3.3.	Algunos conceptos importantes	34
3.3.1.	El ojo humano	35
3.3.2.	Modelo de color RGB	35
3.4.	Procesamiento en el dominio espacial	36
3.4.1.	Procesamiento del color	36
3.4.2.	Restauración digital de imágenes	37
3.4.3.	Reconocimiento de patrones (clasificación)	37
3.4.4.	Morfología matemática	38
3.5.	Procesamiento en el dominio de la frecuencia	40
3.6.	Segmentación del movimiento	40
3.6.1.	Antecedentes de la segmentación del movimiento	41
3.6.2.	Segmentación del movimiento basada en discontinuidades de flujo óptico	42
3.6.3.	Segmentación del movimiento basada en la detección de cambios	42
3.6.4.	Segmentación del movimiento basada en la estructura del movimiento (SFM)	43
3.7.	Conclusiones del capítulo	45
Capítulo 4.	Solución del problema	47
4.1.	Introducción	48
4.2.	Plataforma de programación	48
4.3.	Descripción general del algoritmo	49
4.4.	Obtención de la secuencia de imágenes / Decodificación de video	50
4.5.	Comparación de las imágenes	51
4.5.1.	Generación de la imagen de fondo	52

4.5.2.	Resta y umbralado de las imágenes.....	53
4.5.3.	Supresión de los artefactos por capa de color	54
4.5.4.	Recomposición de la imagen	55
4.5.5.	Etiquetado	56
4.5.6.	Generación de las imágenes de salida	56
4.6.	Escritura de datos	57
4.7.	Correlación de los objetos detectados	59
4.8.	Interfaz gráfica.....	62
4.9.	Pseudocódigo del algoritmo	65
4.10.	Conclusiones del capítulo	66
Capítulo 5.	Evaluación de los resultados.....	67
5.1.	Experimentos.....	68
5.1.1.	Variación del umbral de diferencia entre pixeles	69
5.1.2.	Cambio en el umbral de apertura por área.....	71
5.1.3.	Desplazamiento alejándose de la cámara.....	75
5.1.4.	Desplazamiento acercándose hacia la cámara	77
5.1.5.	Variación del área del sujeto a la misma distancia	79
5.1.6.	Generación del reporte	80
5.2.	Limitaciones del sistema.....	82
5.2.1.	Influencia de las sombras en el área de los objetos.....	82
5.2.2.	Problemas de iluminación	83
5.3.	Conclusiones del capítulo	85
Capítulo 6.	Conclusiones y trabajos futuros.....	87
6.1.	Conclusiones.....	88
6.2.	Recomendaciones	89
6.3.	Trabajos futuros	89
6.3.1.	Agregar Histograma en el reconocimiento de objetos	90
6.3.2.	Agregar “cortes de grafos” para delinear mejor objetos detectados	90
6.4.	Conclusión final	92
	Referencias.....	93

Lista de figuras

Figura 1-1. Pasos a seguir para la segmentación del movimiento y caracterización de los objetos detectados.	4
Figura 2-1. Diagrama de flujo para obtener la diferencia de imágenes FD.....	11
Figura 2-2. Ejemplo de FD. La imagen (c) es el resultado de restar la imagen (b) de la imagen (a).....	11
Figura 2-3. Diagrama de flujo de la sustracción del fondo con una operación de umbralado.	13
Figura 2-4. Ejemplo de sustracción del fondo. La imagen (c) es el resultado de restar la imagen (b) de la imagen (a).....	13
Figura 2-5. Algoritmo BSFD con diferencia dinámica.	18
Figura 2-6. Comparación de los resultados con un algoritmo existente y el algoritmo BSFD con diferencia dinámica en interiores y exteriores.	19
Figura 2-7. Segmentación del movimiento en un video tomado en una carretera, utilizando morfología para eliminar los artefactos, (a) nos muestra 3 imágenes: a la izquierda el cuadro 12 del video, en el centro el cuadro 14 y hasta la derecha la imagen de fondo. En (b) se muestra sustracción de fondo del cuadro 12. En (c) se muestra sustracción de fondo del cuadro 14. (d) nos muestra la diferencia de imágenes con umbral 54. (e) es la diferencia de imágenes con umbral 47. (f) es la imagen binaria con máscara de movimiento.	20
Figura 2-8. Ejemplo de la ecualización de una imagen.	21
Figura 2-9. Ejemplo del resultado al aplicar el algoritmo de detección de bordes de canny y FD.....	22
Figura 2-10. Proceso FD de tres imágenes, (a), (b) y (c) son 3 imágenes sucesivas en un video. (d) es el resultado de FD entre (b) y (a); (e) es el resultado FD entre (c) y (b); (f) es el resultado de restar (d) y (e).	24
Figura 2-11. Valores de la máscara del operador de sobel.	26
Figura 2-12. Máscaras de convolución de sobel extendidas a 5x5 píxeles.	26
Figura 2-13. Resultados de algoritmo de segmentación utilizando cortes normalizados. El renglón a) muestra una secuencia de una carrera de carl lewis, nótese que el fondo se desplaza hacia la izquierda, por el movimiento de la cámara que mantiene al corredor en el centro de la imagen, en este caso la sustracción de fondo no funcionaría como técnica de segmentación. Las dimensiones de la imagen original es de 200 x 190, los parches de la imagen que se utilizan para crear los cortes del grafo son de 3 x 3. Cada uno de los parches de la imagen se conecta con otros parches que se encuentran a menos de 3 cuadros de diferencia. B) muestra los píxeles del corredor, c) es el fondo en movimiento y d) es parte de la pierna izquierda, lo cual resulta de que esta parte sufre de una rotación hacia arriba en la secuencia de imágenes.	29
Figura 2-14. Ejemplificación de inconvenientes en FD y sustracción de fondo.	31
Figura 3-1. Esquema básico del ojo humano.	35
Figura 3-2. Los 3 planos de color del modelo de color RGB.	36
Figura 3-3. Dilatación y erosión. En (a) se muestra la imagen original. En (b) se muestra el resultado de la dilatación de (a). En (c) se muestra el resultado de la erosión de (a).	39
Figura 3-4. Imagen dividida en cinco regiones.	41
Figura 3-5. Línea y plano epipolar.	44
Figura 4-1. Diagrama a bloques del proceso general de detección de movimiento	49
Figura 4-2. Lectura de archivos.	50
Figura 4-3. Secuencia de pasos en el procesamiento de imágenes.....	51

Figura 4-4. Composición de la imagen de fondo. En la imagen a) se han reemplazado los pixeles en los rectángulos rojo y verde por los pixeles de la imagen b) que están en la misma posición, resultando la imagen c) que será utilizada como imagen de fondo.	52
Figura 4-5. Promediado de imágenes. En a) se muestra una secuencia de imágenes, (b) es el promediado de esas imágenes, es el mapa de bits que se utiliza como imagen de fondo.	53
Figura 4-6. Supresión de los artefactos. En a) se muestra una imagen con artefactos. En b) se muestra la misma imagen después de habérsele suprimido los artefactos.	55
Figura 4-7. Recomposición de la imagen. En las imágenes a, b y c, se muestran las imágenes binarias correspondientes a las capas roja, verde y azul respectivamente, en d) se muestra la imagen resultante de combinar estas tres capas.	55
Figura 4-8. Ejemplo de etiquetado. En a) se muestra una imagen binaria con dos objetos, después del proceso de etiquetado, se puede ver la misma imagen pero con los elementos etiquetados utilizando diferentes colores para ver mejor resaltados los dos objetos.	56
Figura 4-9. Pasos seguidos durante la comparación de imágenes en el algoritmo propuesto.	58
Figura 4-10. Ejemplo del suceso “se convirtió en” de la “imagen a” a la “imagen b”.	59
Figura 4-11. Ejemplo del suceso “se fusionó con” de la “imagen a” a la “imagen b”.	60
Figura 4-12. Ejemplo del suceso “son producto de la separación” de la “imagen a” a la “imagen b”.	60
Figura 4-13. Ejemplo del suceso “apareció” de la “imagen a” a la “imagen b”.	60
Figura 4-14. Ejemplo del suceso “desapareció” de la “imagen a” a la “imagen b”.	61
Figura 4-15. Ejemplo del archivo de salida <i>observer.txt</i>	61
Figura 4-16. Ejemplo de la ventana de salida del algoritmo desarrollado.	63
Figura 4-17. Ventana de Figuras.....	64
Figura 5-1. Muestra de la variación del umbral FD en la misma imagen. El área roja muestra los pixeles que fueron detectados como objetos en movimiento.	70
Figura 5-2. Relación entre el umbral FD y los objetos detectados en el experimento anterior.	71
Figura 5-3. Imágenes obtenidas al variar el umbral de la apertura por área. En cada imagen se muestran en la parte superior los objetos detectados sobrepuestos a la imagen original; en la parte inferior la imagen nos muestra cada objeto con su centroide en un rectángulo color ciano. En la secuencia de imágenes se nota cómo al aumentar el umbral disminuye el número de objetos detectados.....	74
Figura 5-4. Relación entre el umbral de apertura y el número de objetos detectados.....	75
Figura 5-5. Análisis del movimiento de una persona alejándose de la cámara.....	76
Figura 5-6. Análisis de un objeto que se aleja de la cámara. En el círculo verde se nota un aumento en el área del objeto, a pesar de que éste se aleja de la cámara.	77
Figura 5-7. Secuencia de imágenes de sujeto acercándose a la cámara.....	78
Figura 5-8. Gráfica de la variación de área del sujeto que se acerca a la cámara, donde se puede apreciar en la marca verde que hay dos decrementos sucesivos en el valor del área.	79
Figura 5-9. Comparación del mismo sujeto a la misma distancia en diferente posición. En a) se ocupa un área de 21,523 pixeles, en b) ocupa un área de 13119 pixeles.	79
Figura 5-10. Comparación del área del mismo sujeto con sombra y sin sombra. En a) el sujeto con sombra es resaltado con una marca verde, en b) el mismo sujeto, ya sin sombra, también resaltado con una marca verde.	83
Figura 5-11. Imagen con iluminación muy desigual.....	84
Figura 6-1. Ejemplo de algoritmo de corte de grafos. En la imagen a), se hacen dos garabatos rojos sobre el objeto que se quiere segmentar, y se dibujaron otros garabatos azules sobre el fondo de la imagen, b) es la imagen binaria obtenida después de aplicar el algoritmo de corte de grafos.....	90

Figura 6-2. Método de corte de grafos aplicado recursivamente. En a) se muestran en color rojo los pixeles que pertenecen al objeto y en color azul los pertenecientes al fondo, en b) es nuevamente la imagen original, ahora con el borde más definido. 91

Lista de siglas y acrónimos

ACM.- Asociación para sistemas informáticos, del inglés *Association for Computing Machinery*.

ADP.- Imagen de diferencia acumulativa, del inglés *Accumulative Difference Picture*.

aprox.- Aproximadamente.

AVI.- Formato contenedor de video, del inglés *Audio Video Interleave*.

BG.- Plano de fondo, del inglés *Background*.

BS.- Sustracción de fondo, del inglés *Background Subtraction*.

BSFD.- Diferencia de imágenes con sustracción de fondo, del inglés *Background Subtraction With Frame Difference*.

Códec.- Codificador-decodificador.

DB.- Fondo dinámico de la imagen, del inglés *Dinamic Background*.

EM.- Maximización de la expectativa, del inglés *Expectation-maximization*.

FA.- Apertura del primer plano, del inglés *Foreground Apperture*.

FD.- Diferencia de imágenes, del inglés *Frame Difference*.

FG.- Primer plano, del inglés *Foreground*.

FHD.- Alta definición completa, del inglés *Full High Definition*.

FLV.- Formato contenedor de video, del inglés *Flash Video*.

fps.- *Frames per second* - Fotogramas por segundo.

GH.- Efecto de fondo; gh se toma de las dos primeras letras de la palabra en inglés *Ghost*.

Matlab.- Significa laboratorio de matrices, del inglés *Matrix Laboratory*; es un software de alto nivel para cálculos numéricos y simbólicos.

MPEG.- Grupo de expertos en imágenes en movimiento, del inglés *Moving Picture Experts Group*.

SFM.- Estructura a partir del movimiento, del inglés *Structure From Motion*.

STR.- Sistema de tiempo real.

TFD.- Diferencia de tres imágenes, del inglés *Three Frame Difference*.

WP.- Persona caminando, del inglés *Walking Person*.

Lista de tablas

Tabla 5-1. Relación del umbral de diferencia y objetos detectados.....	69
tabla 5-2. Relación del umbral de apertura por área y el número de objetos detectados.....	72

Glosario de términos

Algoritmo de Canny.- Operador de detección de borde que utiliza un algoritmo multietapa para detectar una amplia gama de bordes en imágenes digitalizadas.

AND.- Operación booleana en la cual se multiplican lógicamente bit a bit los conjuntos definidos por dos o más números binarios.

Arista.- Línea que resulta de la intersección de dos superficies, considerada por la parte exterior del ángulo que forman.

Booleano.- Símbolos referentes al álgebra booleana que son utilizados para la relación entre términos matemáticos, variables, etc.

Distancia Euclidiana.- Es la distancia "ordinaria" (que se mediría con una regla) entre dos puntos de un espacio euclídeo, la cual se deduce a partir del teorema de Pitágoras.

Entropía de bloque.- Término utilizado para describir la complejidad de la información en un conjunto de bits.

Filtro de Sobel.- Algoritmo para detectar bordes utilizado en el procesamiento de imágenes digitales, que crea una imagen haciendo hincapié en los bordes.

Filtro Gaussiano.- Es un efecto de suavizado para mapas de bits generado por algoritmos de edición gráfica digital.

Ground truth.- Conjunto de medidas que se sabe son más exactas que las que se están obteniendo con algún sistema de pruebas.

Heurística.- Método basado en la experiencia que puede utilizarse como ayuda para resolver problemas de diseño a base de prueba y error; puede incluir la elaboración de medios auxiliares, principios, reglas, estrategias y programas que faciliten la búsqueda de vías de solución al problema.

Híbrido.- Referencia a todo aquello que sea el resultado de la mezcla de dos o más elementos de diferente naturaleza o tipo.

Histéresis.- En el proceso Canny, es la identificación de bordes considerados fuertes mediante el seguimiento y conexión de bordes similares.

Iteración.- Acto de reiterar varias veces determinados pasos.

MOV.- Formato de video desarrollado para QuickTime.

mpeg.- Formato de imágenes digitales desarrollado por MPEG.

Oclusiones.- Acción y resultado de cerrar u obstruir algo.

Ondeletas.- Es un tipo especial de transformada matemática que representa una señal en términos de versiones trasladadas y dilatadas de una onda finita (denominada óndula madre).

Operador de Kirsch.-Algoritmo para detectar bordes utilizado en el procesamiento de imágenes digitales; se toma una sola máscara kernel y se gira en incrementos de 45 grados a través de los 8 puntos cardinales.

OR.- Operación booleana en la cual se suman lógicamente bit a bit los conjuntos definidos por dos o más números binarios.

Teoría estadística.- Herramienta que estudia usos y análisis provenientes de una muestra representativa de datos.

Umbralado.- Cantidad más reducida de un valor que se tiene que dar para que sea advertida durante un proceso.

Ud.- Abreviatura de usted, tratamiento de cortesía.

Acerca de los términos “tiempo real” y “segmentación de movimiento”

El título de este trabajo de tesis incluye los términos “*segmentación de movimiento*” y “*tiempo real*”; por lo tanto, ante todo considero que es necesario definirlos.

La “segmentación de movimiento” es un proceso mediante el cual, en una secuencia de imágenes, se separan los objetos que se considera que están en movimiento respecto del fondo, el cual es considerado estático. Esto es, en una imagen digital, se detectan y agrupan los píxeles que pertenecen a objetos en movimiento y se separan del fondo de la imagen.

La definición de “tiempo real” es un poco más complicada. Según David Gray [David Gray, 2001] los sistemas se pueden clasificar según sus acciones en:

Secuenciales.- Las acciones no pueden traslaparse en tiempo.

Concurrentes.- Puede haber coincidencia en tiempo de las acciones.

En tiempo real.- Estos son los más complicados porque pueden estar compuestos de varios subsistemas con diferentes comportamientos y con restricciones de tiempo.

Los sistemas en tiempo real se pueden dividir en “tiempo real duro” y “tiempo real suave”. Según W.S. Liu [Liu, 2000], la restricción de tiempo real dura, es cuando al haber un retardo puede tener consecuencias desastrosas, por ejemplo fallas en procesos críticos de reactores nucleares o en sistemas de aeronaves según Lee y Krishna [Lee y Krishna, 2010]. Los sistemas de tiempo real suave, admiten retardos en resultados que pueden tener consecuencias con malos resultados, pero que no son importantes. Por lo tanto, se dice que si un sistema de tiempo real no puede fallar, su tiempo de entrega entonces es “duro”, pero si ocasionalmente puede fallar se considera “suave”.

Así pues, un sistema en tiempo real (STR) es un sistema que interactúa activamente con un entorno de dinámica conocida que relaciona las entradas y produce salidas con restricciones de tiempo; su correcto funcionamiento va de acuerdo con los conceptos de predictibilidad, estabilidad y controlabilidad. No basta con que las acciones del proceso sean correctas, sino que además, tienen que ejecutarse dentro de un intervalo de tiempo determinado.

El concepto de STR no queda restringido a los sistemas digitales o de cómputo, ya que puede extenderse al mundo vivo: humanos, animales y plantas. Como ejemplo, considérese un animal mamífero que empieza con su concepción, su crecimiento como un feto por un tiempo determinado por la especie misma, después su nacimiento e inicio de su alimentación fuera del vientre, crecimiento, inicio del ciclo reproductivo, desarrollo, etc. Se puede asegurar que el proceso es un sistema de tiempo real en el ambiente y en las circunstancias en las cuales se desarrolla, ya que a estímulos del ambiente, responde dentro de sus restricciones de tiempo específicas.

En el presente trabajo de tesis, se tomará el término “tiempo real”, como un proceso de cómputo digital, con restricción de tiempo; en el cual se procesarán 12 fps (*frames per second* – fotogramas por segundo), debiéndose obtener de forma correcta y en un tiempo de 0.08 décimas de segundo los resultados de cada imagen, lo cual nos da un retardo prácticamente imperceptible para el ojo humano

Capítulo 1. Introducción a la segmentación del movimiento.

1.1.El problema a resolver

Para este trabajo de tesis será necesario desarrollar los algoritmos y programas que detecten, muestren y caractericen los objetos en movimiento en tiempo real, dada una secuencia de imágenes digitales capturadas con una cámara digital estática, con propósitos de videovigilancia.

Para realizar este trabajo es necesario procesar secuencias de imágenes, las cuales pueden ser leídas directamente como secuencia de imágenes desde un medio de almacenamiento masivo, o pueden ser adquiridas desde la lectura de un archivo de video, sustrayendo uno a uno los mapas de bits a ser procesados. Cuando se trata de lectura de imágenes sucesivas, no es mayor problema leer una a una la secuencia de imágenes, siempre y cuando, los nombres de archivo tengan una secuencia lógica; Si la lectura de las imágenes será desde un archivo de video, es entonces necesaria la implementación de un códec de video, el estudio de este tema que está directamente relacionado con la codificación/descodificación de la información y la compresión de datos, se tomará como un campo de estudio que se desarrollará de forma paralela.

El estudio de la codificación de video es un campo amplio y que tiene diferentes ramas, estándares, algoritmos, incluso marcas y procesos registrados. A simple vista, se puede vislumbrar que es un tema de extenso estudio.

Una vez resuelto el tema de la adquisición de la secuencia de imágenes, se realiza la segmentación del movimiento; en un vistazo rápido al estado del arte, se puede ver la gran cantidad de algoritmos que se han desarrollado con este fin y las grandes diferencias que existen entre ellos. Algunos utilizan vectores para comparar las imágenes, otros gradientes, otros la simple diferencia espacio-temporal entre píxeles y secuencias de imágenes, o utilizan la clasificación según el histograma, otros se basan en la búsqueda de puntos de interés etc. Hay muchos caminos que se han recorrido con esta misma finalidad.

Es también un propósito de este trabajo de tesis, realizar la caracterización de los objetos encontrados, es decir: realizar un reporte de los objetos analizados y el rastreo que surge del análisis de estos objetos, la dirección de su desplazamiento, el tamaño de los objetos etc. Hay documentos científicos completos acerca de este tema, por lo tanto la implementación de este proceso de rastreo, será también un proceso complicado.

Y es así como el camino a seguir para el desarrollo de esta tesis, ha sido trazado de esta manera. Se necesitarán conocimientos de codificadores de video, visión por computadora, análisis de imágenes, correlación de la información y clasificación.

1.2.Justificación

Dada la evolución que la inteligencia artificial ha tenido, y que las aplicaciones en computadora están evolucionando, se hace necesario el desarrollo de algoritmos y procesos que agilicen las tareas en varios campos de la ciencia y la tecnología, en la vida cotidiana del ser humano.

Un ejemplo de esto es la visión por computadora. Existen muchas tareas que el ser humano considera incómodas de realizar, o que no pueden ser llevadas a cabo de manera eficiente, a causa de la naturaleza misma de cómo funcionan nuestros cerebros. Un ejemplo de esto es el análisis de imágenes.

En ocasiones, el cerebro humano es el único capaz de reconocer patrones de manera acertada, o de realizar tareas intrincadas de manera correcta. Pero en otras situaciones, el uso de una computadora, agiliza el trabajo, lo realiza de manera más rápida y muy acertada, ejemplos de esto son el reconocimiento de placas de autos que viajan a gran velocidad o la adquisición y análisis de acontecimientos que suceden a larga distancia, además de otros ejemplos.

Es así como se presenta este trabajo de tesis con aplicaciones de videovigilancia, en donde se analizarán secuencias de imágenes con la finalidad de detectar los objetos en movimiento que existan en esta secuencia de imágenes.

Será así posible, ejecutar un programa en la computadora que analice videos tomados con fines de videovigilancia, y nos dé un reporte de los objetos detectados, ya sea las escenas en donde detectó movimiento o la frecuencia de los objetos detectados a ciertas horas del día, además de algunas otras aplicaciones a las que se podría extender el algoritmo. Todo esto realizado de manera automática, sin la intervención humana, ya que este tipo de trabajos, como se mencionó anteriormente, es considerado como de difícil ejecución para un ser humano, a causa del tedio que de manera natural, es generado en nosotros.

Otro de los intereses para presentar este trabajo de tesis, es iniciar un camino en la visión por computadora.

1.3.Hipótesis

Dada una secuencia de imágenes, la cual fue adquirida con una cámara estática, es posible generar algoritmos capaces de detectar los objetos en movimiento que existen en dicha secuencia de imágenes.

Este trabajo se basa en que al cambiar de posición un objeto en una secuencia de imágenes, es posible analizar el cambio de brillo y color en los píxeles que se ven afectados por el movimiento de este objeto, es entonces posible la clasificación de píxeles en objetos que pertenecen a un objeto en movimiento, y en píxeles que pertenecen al fondo de la imagen.

Estos píxeles que se mueven de manera conjunta, se presupone que pertenecen al mismo objeto, y se pueden agrupar para conformar en conjunto un objeto en movimiento, este objeto tiene una posición en la imagen, y un área, estos atributos se pueden correlacionar entre imágenes sucesivas para describir la ruta que siguió el objeto, y conjuntamente con todos estos datos, se puede generar un reporte que incluya toda la información relacionada a los objetos detectados.

1.4.Objetivos

1.4.1. Objetivo general

El objetivo general de este trabajo es realizar en tiempo real la segmentación del movimiento en una secuencia de imágenes. Estas imágenes pueden ser leídas directamente de una cámara conectada a la computadora, o pueden ser leídas de una secuencia de video, o del disco duro de la PC. El análisis de los resultados obtenidos se puede realizar posteriormente.

1.4.2. Objetivos particulares

Se muestra la secuencia de pasos a seguir en la figura 1-1. Para poder describir los objetivos particulares, el primer paso es la adquisición de imágenes, ya sea mediante la decodificación de video, o la lectura directa de la secuencia de imágenes, después se realizará la segmentación de los objetos en movimiento y finalmente el reporte acerca de la caracterización de los objetos.

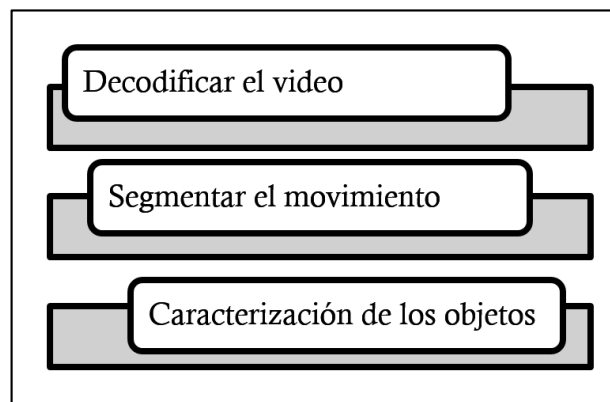


Figura 1-1. Pasos a seguir para la segmentación del movimiento y caracterización de los objetos detectados.

A continuación se listan los objetivos particulares

- a) Desarrollar una interfaz gráfica para el usuario final. -Con la finalidad de que el usuario tenga el control del algoritmo, es necesario desarrollar un ambiente gráfico con el que se pueda interactuar de manera cómoda e intuitiva, y que brinde la información necesaria para entender el funcionamiento de dicho algoritmo.
- b) Extraer imágenes sucesivas desde un archivo de video, para generar la secuencia de imágenes.- Existirán dos formas de adquirir la secuencia de imágenes: leyéndolas directamente una a una de algún medio de almacenamiento masivo, u obteniéndolas de un archivo de video, para lo cual será necesario hacer uso de un decodificador de video que nos alimente de manera secuencial las imágenes a ser procesadas.
- c) Desarrollar los algoritmos y programas necesarios para llevar a cabo la segmentación del movimiento.- Con el fin de realizar la segmentación del movimiento, será necesario analizar las posibles opciones para la comparación de las imágenes y detección de objetos en movimiento.

d) Implementar los algoritmos necesarios que caractericen los objetos en movimiento detectados.- Con base en los resultados anteriores, se realizará un reporte de los objetos detectados, este reporte se generará de manera automática, y contendrá la información obtenida en los pasos anteriores.

1.5.Consideraciones generales

Se da por hecho que el usuario final entiende el uso básico de una computadora personal, y que está familiarizado con la jerga computacional común, también se presupone que tiene conocimientos básicos de programación. La plataforma de programación será “Matlab r2012a”; se eligió dado que se presta mucho para la experimentación e investigación de visión por computadora, también se presta para otros campos de investigación y análisis.

Las imágenes a ser procesadas, deben ser tomadas con una cámara estática libre de vibración perceptible a simple vista, pueden ser en colores o en escala de grises, y preferentemente deben cumplir con los estándares actuales de videovigilancia.

La plataforma de desarrollo será ejecutada en una computadora personal de capacidades normales, es decir, no en servidores o en supercomputadoras.

Los videos de vigilancia que serán procesados, deberán tener una iluminación buena, o deben haber sido grabados con buena iluminación natural, si los videos tienen una iluminación deficiente, los resultados serán muy pobres.

Para procesar los videos, es necesario que ya estén como archivo de computadora, no como cintas de video; si este es fuera caso, es necesario primero convertirlos a archivo de computadora para poder ser procesados.

1.6.Medios utilizados

Los videos pueden ser grabados por cámaras de videovigilancia, teléfonos inteligentes, tabletas o cualquier video grabadora, siempre con la restricción de que el sensor debe permanecer estático durante la grabación.

Las computadoras en donde se probará el video, deben tener el suficiente espacio para funcionar de manera correcta, leer y escribir datos en velocidades normales, y con el sistema operativo funcionando también correctamente.

1.7.Organización del documento

Este documento de tesis está organizado en seis capítulos que se describen a continuación:

El Capítulo uno da una idea básica de la segmentación del movimiento, el problema a resolver y algunos pormenores a resolver en cuanto al tema, la hipótesis de cómo atacar el problema a resolver y la justificación del mismo; también incluye los objetivos del trabajo así como las consideraciones generales y los medios utilizados. En general es el primer vistazo al tema de esta tesis.

El capítulo dos, nos brinda una introducción al estado del arte, en él se explican brevemente algunos trabajos que se relacionan directamente con el que Vd. tiene en sus manos, o que nos dan una visión diferente acerca de cómo enfrentar los obstáculos, o ser más ingeniosos para resolverlos. También en el estado del arte, se pueden ver como las técnicas son muy diferentes unas de otras. Además se explican las técnicas que se toman como base para este trabajo de tesis, las cuales son: “Diferencia de Imágenes” y “Sustracción de Fondo”.

El capítulo tres, es un pequeño viaje por los conceptos básicos, se incluye la forma en cómo funciona el sistema de visión humana, la cual está relacionada con la forma en cómo se guardan los archivos de imágenes en las computadoras. También nos explica cómo es que las imágenes pueden ir del dominio espacial al dominio de la frecuencia y viceversa. Este capítulo trata también diferentes formas de conseguir la segmentación del movimiento.

El capítulo cuatro nos describe la solución del problema, nos muestra la interfaz gráfica además del pseudocódigo y la descripción general de cómo funciona el algoritmo desarrollado en este trabajo.

En el capítulo cinco se describen los resultados obtenidos después de la evaluación del algoritmo y una evaluación general del sistema.

En el capítulo seis se da una conclusión final del trabajo y algunas posibles mejoras que se pueden incorporar para mejorar el sistema o para adaptarlo con otras aplicaciones de este algoritmo.

Se pueden escribir sugerencias, recomendaciones y críticas al correo ranarene@hotmail.com las cuales serán leídas y en caso de ser necesario también serán contestadas.

Espero que el lector disfrute de esta lectura.

Capítulo 2. Estado del arte.

2.1.Introducción

En este capítulo, primero se muestra un recorrido por los principales obstáculos a los que se enfrentan los algoritmos de segmentación de movimiento, y también se muestra una clasificación de los diferentes métodos que se pueden utilizar para segmentar el movimiento. Después se explican 2 métodos que se utilizan para realizar la comparación de imágenes: “sustracción de fondo” y “diferencia de fotogramas” también conocida como FD por sus siglas en inglés (*Frame Difference*); más adelante se realiza un pequeño recorrido por algunos métodos que se han aplicado recientemente, también se analizan brevemente los pros y los contras de cada una de estas técnicas, así como los resultados obtenidos. El análisis de las técnicas que se abordan en este capítulo, influyen de manera directa en este trabajo de tesis, el cual toma como base la sustracción de fondo.

Según nos dicen Luca Zapella, Xavier Lladó y Joaquín Salvi [Zapella et al., 2008] en su tratado “*Motion segmentation: a review*”, la segmentación de movimiento es una tarea esencial en los algoritmos de visión por computadora, y se han publicado muchos trabajos para atacar este tema; en esta revisión de soluciones para la segmentación de movimiento, se estudian y analizan las técnicas más actuales e importantes de acuerdo a sus características y principios más importantes. Además, se mencionan sus puntos fuertes y débiles. A continuación una revisión rápida de estos algoritmos sugeridos y su clasificación.

Para realizar una segmentación del movimiento automática en secuencias de imágenes reales, se necesitan resolver algunos problemas particularmente importantes, por ejemplo: ruido, falta de información, borrosidad, movimiento de la imagen, desaparición y reaparición de objetos dentro de la escena y falta de conocimiento a priori. En el tratado de Zapella, se indica que los obstáculos principales con los que la segmentación de movimiento se puede encontrar son:

Oclusiones.- Es la habilidad de resolver las oclusiones de los objetos detectados.

Varios objetos.- Es cuando el algoritmo puede detectar más de un objetos de manera correcta.

Continuidad espacial.- Es la habilidad de explotar la continuidad espacial.

Selección de puntos de interés y/o métodos de densidad. En la selección de puntos de interés, los objetos son representados por un limitado número de puntos, por ejemplo esquinas o puntos salientes. Densidad de píxeles, realiza un análisis acerca de los movimientos de los píxeles.

Detención de objetos.- Es la habilidad de seguir reconociendo los objetos aun cuando después de realizar cierto movimiento estos se detienen.

Robustez.- Es la manera en cómo el algoritmo se sobrepone al ruido. En el caso de “selección de puntos de interés”, es que la posición de los puntos de interés se vean afectados sin que esto ocasiona errores en la asociación de los datos.

Secuencialidad.- Es la habilidad de trabajar incrementalmente, esto significa por ejemplo que el algoritmo pueda explotar correctamente la información que no estaba presente en el inicio de la secuencia.

Falta de información.- Es la manera en cómo se resuelven los problemas de falta de información.

Objetos no rígidos.- Es la habilidad de lidiar con los objetos no rígidos.

La literatura acerca de la segmentación del movimiento es amplia, para explicar mejor este análisis, se muestra una clasificación en la que se pueden dividir los algoritmos, algunos métodos pueden pertenecer a más de una clase, las cuales son:

Diferencia de Imágenes.- Se basa en los cambios espacio-temporales, no se puede utilizar cuando hay movimiento de cámara, es muy sensible al ruido, normalmente se mezcla con otras técnicas para mejorar los resultados.

Teoría estadística.- Es ampliamente usado en el campo de la segmentación de movimiento; de hecho segmentar el movimiento puede ser visto como un problema de clasificación, en donde cada pixel tiene que clasificarse en fondo u objeto en movimiento. Dependiendo del marco de referencia se pueden subdividir en: probabilidades máxima a posteriori, filtro de partículas y maximización expectativa.

Flujo óptico (FO).- Al igual que diferencia de imágenes, FO es un concepto antiguo que es ampliamente explotado en la visión por computadora. Se encuentra en el dominio de vector de movimiento, que describe la distribución de las velocidades aparentes de los patrones de brillo en la secuencia de imágenes. La idea de utilizar discontinuidades en el flujo óptico con el fin de segmentar movimiento es aún más antigua.

Ondeletas.- Este conjunto de métodos se basa en el análisis de las ondeletas para realizar el análisis de los diferentes componentes de las frecuencias en las imágenes, y después se analiza cada componente con una resolución de acuerdo a su escala.

Capas.- La idea principal de estos algoritmos es entender cuáles son las diferentes capas en profundidad en la imagen y que objetos (o que parte de objetos articulados) pertenecen a esa capa. Esta técnica se utiliza frecuentemente en estereovisión dado que es más fácil calcular la profundidad; de cualquier manera, sin calcular la profundidad, es posible estimar que objetos se mueven en planos similares, esto es muy útil y ayuda a resolver el problema de la oclusión.

Factorización.- Las técnicas de factorización se han vuelto muy populares debido a su simplicidad. Recobran la estructura y movimiento utilizando rastreo de características en una secuencia de imágenes.

En la clasificación anterior que nos muestra Zapella, se amplía la visión acerca de algunas de las técnicas que existen para segmentar el movimiento. En él, no se menciona específicamente el costo computacional de cada clasificación de técnicas; una de las metas principales de este trabajo es que se pueda implementar en tiempo real, a una velocidad de 12 fps, fue por tanto necesario realizar pruebas de velocidad de algunos de los métodos mencionados para lograr esta meta.

2.2.Comparación de Imágenes

A pesar de que la segmentación del movimiento ya tiene muchos años de estudio, no se ha podido crear una técnica para uso general con este fin, pues las técnicas desarrolladas todavía están lejos de poder conseguir esta meta. Por ende, este tema todavía tiene un gran futuro de desarrollo, estudio y aplicación en muchos campos de la ciencia, debido a que todavía genera muchos retos para obtener resultados satisfactorios. Cabe mencionar que hay aplicaciones específicas, en las cuales el desempeño automatizado de la segmentación del movimiento es, incluso, mejor que el de la visión humana, y hasta pueden ser más veloces y exactas, por ejemplo en el mundial de fútbol soccer en Brasil 2014, se utilizó por primera vez el programa *GoalControl*, un sistema desarrollado por una empresa Alemana en Renania del Norte, de nombre homónimo al software “GoalControl”, que termina con el problema de los goles fantasmas, pues según sus creadores, puede asegurar al cien por ciento, cuando se produce un gol y avisa al árbitro; en cada uno de los 12 estadios de este torneo, se instalaron catorce cámaras de video, estas transmiten las imágenes a un servidor que las procesa filtrando a los jugadores, al árbitro y otras posibles interferencias dejando sólo visible el balón, el sistema captura quinientas imágenes por segundo y permite conocer la posición del balón dentro del campo cada 2 milisegundos con una precisión de 5 milímetros. La aplicabilidad de la segmentación del movimiento, puede tener tantas aplicaciones como la imaginación nos lo permita.

Es recomendable preprocesar las imágenes para su comparación, pues un buen preprocesamiento nos arrojará mejores resultados. De manera inversa, un mal o inexistente preprocesamiento seguramente nos proporcionará resultados no satisfactorios.

2.2.1. Diferencia de imágenes (FD)

El método de segmentación por diferencia de imágenes o FD (por sus siglas en inglés del término *Frame Difference*), es muy utilizado actualmente combinado con otras técnicas. FD toma como base la diferencia de 2 imágenes sucesivas que se comparan pixel a pixel para detectar zonas de cambio entre ellas; de esta manera se realiza una comparación temporal y espacial entre las dos imágenes, para detectar las zonas donde no hubo cambios y donde sí los hubo, con el fin de detectar el objeto o los objetos en movimiento. La diferencia de imágenes se obtiene mediante la expresión:

$$FD(x) = S_k(x) - S_{k+1}(x)$$

Donde: $FD(x)$ Es la imagen resultante.

$S_k(x)$ Es la imagen de donde se sustraerán los valores pixel a pixel.

$S_{k+1}(x)$ Es alguna imagen consecutiva a $S_k(x)$

FD normalmente se aplica a toda la imagen y no sólo a un pixel o a un grupo de pixeles como en otras técnicas; cumple con la restricción de que la cámara que toma el video debe permanecer estática durante la grabación del mismo. En la figura 2-1, se muestra el diagrama de flujo para la obtención de FD.

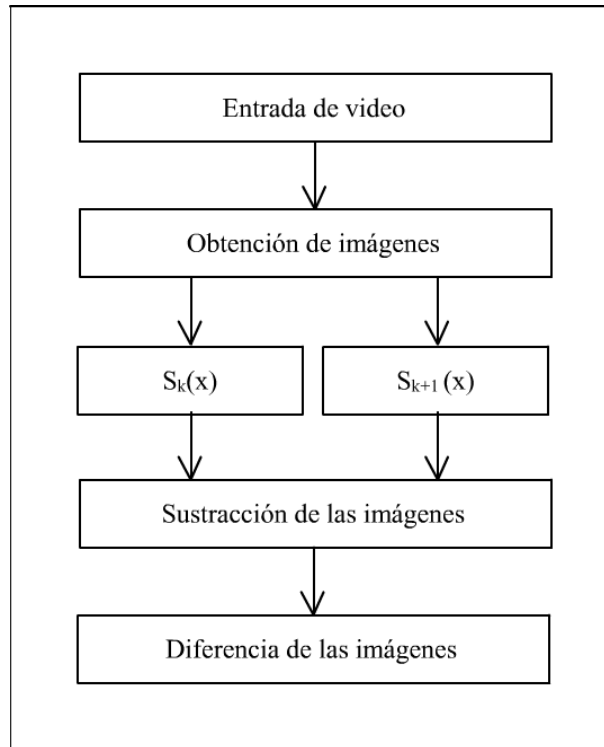


Figura 2-1. Diagrama de flujo para obtener la diferencia de imágenes FD.

En la figura 2-2, se muestra la imagen resultante de restar la imagen (b), de la imagen (a). Con imágenes de alta resolución se pueden apreciar los artefactos generados de esta simple sustracción de valores. En este ejemplo no se realiza ningún preprocesamiento ni mejoramiento de las imágenes iniciales ni del resultado obtenido.

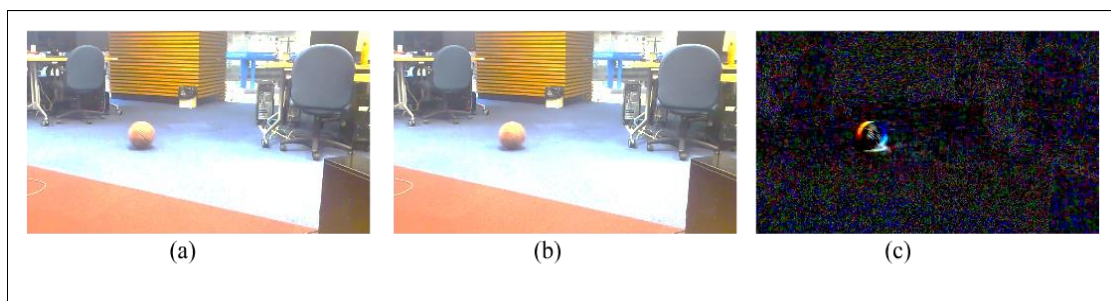


Figura 2-2. Ejemplo de FD. La imagen (c) es el resultado de restar la imagen (b) de la imagen (a).

Algunos de los problemas con los que hay que lidiar son: el método es muy sensible al ruido; los cambios repentinos de luz pueden ocasionar resultados desfavorables; las sombras pueden ser reconocidas como objetos en movimiento; los pequeños o grandes movimientos de la cámara generan resultados inesperados; es susceptible a errores cuando hay cambios ambientales, etc. Sin embargo, algunas buenas características son: es un algoritmo de fácil implementación, veloz y de

fácil modificación en su parametrización, el que al combinarlo con otras técnicas puede dar buenos resultados.

2.2.2. Sustracción del fondo

La técnica de sustracción de fondo o, simplemente BS por sus siglas en inglés (*Background Subtraction*), consiste en identificar los pixeles que pertenecen al fondo, con el fin de sustraerlos de la imagen, o de la secuencia de imágenes. Esta técnica es parecida a FD; la diferencia es que en FD se realiza la sustracción de dos imágenes consecutivas, y en BS se utiliza la imagen de fondo y la imagen que está siendo procesada para realizar la sustracción.

Supongamos que queremos obtener el resultado de la sustracción de fondo de un video; primero se obtiene la secuencia de imágenes del video, luego es necesario identificar los pixeles que conformarán la imagen de fondo B ; después para cada imagen f_k de la secuencia de video, se realiza la sustracción del fondo, y después se realiza una operación de umbralado para cada pixel; esta operación consiste en asignar un valor booleano a cada uno de los pixeles, basándose en el valor obtenido de cada uno de ellos, con la finalidad de detectar la posición de los pixeles que han cambiado su valor; a estos pixeles se les considerará como parte de objetos en movimiento y los demás pixeles serán considerados como fondo, esto es:

$$R_k(x, y) = f_k(x, y) - B(x, y)$$

$$f(x) = \begin{cases} 1, & \text{fondo } R_k(x, y) \leq T \\ 0, & \text{objeto } R_k(x, y) > T \end{cases}$$

Donde: R_k Es el resultado.

$f_k(x, y)$ Es la imagen procesada.

$B(x, y)$ Es la imagen de fondo.

$f(x)$ Es la función de sustracción de fondo.

En la figura 2-3, se muestra un diagrama de flujo del proceso explicado.

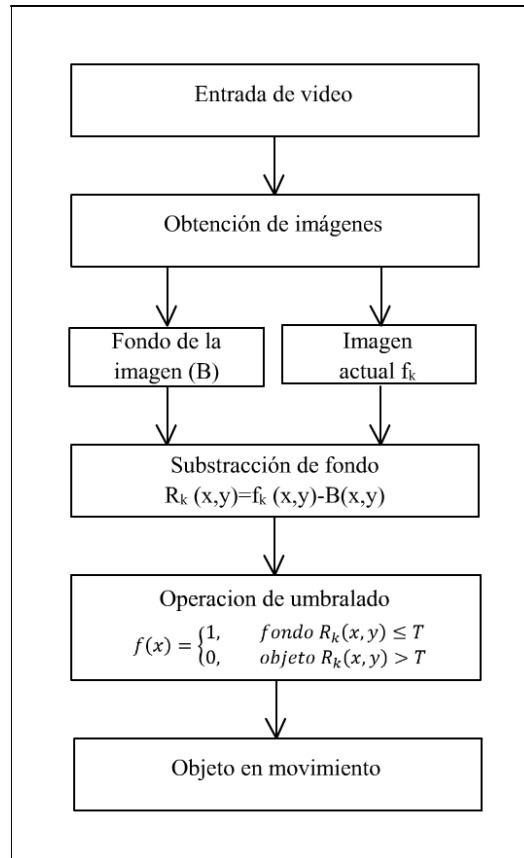


Figura 2-3. Diagrama de flujo de la sustracción del fondo con una operación de umbralado.

Es común realizar algunos procedimientos extras con la finalidad de obtener mejores resultados. Para eliminar los artefactos aditivos subyacentes en las imágenes, se puede realizar un proceso morfológico de apertura por área, también se pueden eliminar los artefactos, si antes de realizar la operación de sustracción, se aplica un filtro gaussiano para suavizar las imágenes. En la figura 2-4 se muestra un ejemplo real de la sustracción del fondo sin umbralado; se puede apreciar cómo los bordes y el relleno del objeto en movimiento, se definen un poco mejor que en FD.

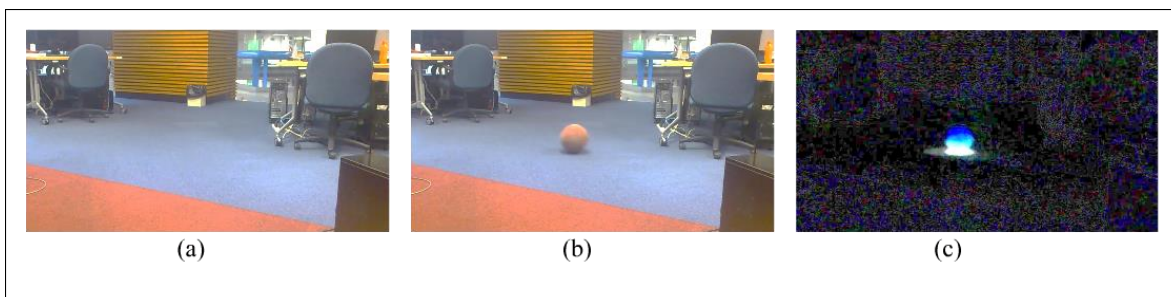


Figura 2-4. Ejemplo de sustracción del fondo. La imagen (c) es el resultado de restar la imagen (b) de la imagen (a).

2.3.Trabajos relacionados con la segmentación de movimiento

En la segmentación del movimiento es muy común utilizar el método de sustracción del fondo; detectar o segmentar el fondo de una imagen sigue siendo un gran reto para la visión por computadora. La sustracción de fondo se utiliza cuando se trabaja con cámaras fijas, es de fácil implementación y brinda una detección efectiva; una deficiencia de este método es que en ocasiones es difícil obtener el fondo de la imagen de manera precisa debido a los artefactos que intrínsecamente siempre se generan.

A continuación se hace una breve revisión de trabajos relacionados con la segmentación del movimiento y llevados a cabo en los últimos años, con la intención de ampliar la visión del lector acerca de este tema.

2.3.1. Segmentación de movimiento por sustracción de fondo y análisis temporal

En el documento [Spagnolo et al, 2005], se presenta un método de segmentación de objetos en movimiento que combina el análisis temporal de las imágenes contra una imagen de referencia de fondo, se toman en cuenta los problemas de cambio de iluminación en espacios abiertos debidos a la intensidad luminosa. En él se presenta un método de adaptación dinámica de fondo. Todos los píxeles en la imagen, incluso aquellos que son ocluidos por objetos en primer plano, son actualizados constantemente en el modelo de la imagen de fondo. Este método muestra un buen desempeño en espacios abiertos y cerrados. En este documento se menciona que los métodos de reconocimiento de fondo que tienen un umbral oportuno en la diferencia entre la secuencia de imágenes y el modelo de fondo, son reconocidos por la comunidad científica, como los que mejor resultados tienen en cuanto a desempeño y confiabilidad. También son los que mejor cumplen con integridad de la información obtenida y una mejor segmentación de los objetos.

El primer paso de cada sistema de análisis de la escena es la segmentación de objetos en primer plano del fondo. Esta tarea es un requisito previo crucial para la eficacia del sistema global. Un algoritmo de segmentación del primer plano debe poder hacer frente a un número de las situaciones críticas como las mencionadas anteriormente en el trabajo de Zapella [Zapella et al, 2008]. En particular, debe hacer frente a un continuo cambio repentino de luz, la variación temporal y permanente en objetos de fondo, y obviamente la presencia de ruido. Por otra parte, una limitación fundamental es que el algoritmo tiene que hacer que el sistema sea sustancialmente independiente de la presencia de objetos en primer plano además de su tamaño y velocidad. Este método se compone de varios pasos para obtener las formas correctas de los objetos en movimiento en cada imagen de la secuencia.

Muchos enfoques de diferencias de cuadros utilizados en la literatura, tiene el problema de producir imágenes resultantes que pueden ser incorrectas, esto en gran medida por el ruido si la selección del umbral no es óptima. Muchos autores sugieren el uso de un proceso de filtrado con el fin de eliminar estas regiones puntuales y obtener mejores resultados.

El primer paso en este tipo de algoritmos es calcular la diferencia pixel a pixel en dos imágenes, la cual se obtiene mediante:

$$R(I^t(x, y), I^{t-1}(x, y)) = \frac{m[W(I^t(x, y))W(I^{t-1}(x, y))] - m[W(I^t(x, y))]m[W(I^{t-1}(x, y))]}{\sqrt{v[W(I^t(x, y))]v[W(I^{t-1}(x, y))]}}$$

Donde: $m[W]$ Es la media de la intensidad de los píxeles evaluada en la ventana W .

$v[W]$ Es la varianza de la intensidad de los píxeles evaluada en la ventana W .

La similitud radiométrica varía en el intervalo (0,1) y es estimado en una ventana W , centrada en el píxel que se comparan entre las imágenes sucesivas $((I^t(x, y)I^{t-1}(x, y)))$. Mediante el uso de puntos vecinos en una ventana para comparar puntos correspondientes, se ha optado por dar una interpretación local en el concepto de diferencia, en lugar de uno basado en píxeles. Así, el algoritmo se vuelve más robusto frente al ruido: el efecto de un único píxel de ruido está limitado por los otros píxeles de la ventana W . El tamaño de la ventana tiene que ser una compensación razonable entre la capacidad de suavizar aglomerados de ruido de píxeles cuando se utilizan grandes ventanas y, la posibilidad de error de detectar puntos estáticos en el borde de objetos en movimiento.

Se decidió considerar dos puntos estáticos con un valor de similitud radiométrica de un umbral menor a σ_s ; valores mayores identifican píxeles en movimiento, tal como lo muestran las fórmulas en la sección 2.1.2 que se refiere a la sustracción de fondo.

$$I_M(x, y) = \begin{cases} 1, & \text{si } R(x, y) < \sigma_s \\ 0, & \text{de otra manera} \end{cases}$$

$$I_s(x, y) = \begin{cases} 1, & \text{si } R(x, y) > \sigma_s \\ 0, & \text{de otra manera} \end{cases}$$

Las opciones del umbral σ_s y el tamaño de la ventana W , realizan un aprendizaje experimental de los valores propios de diferentes imágenes de referencia. De esta manera, los resultados del análisis temporal son menos sensibles a la selección del umbral, ya que el efecto del ruido se suaviza por el análisis local de la similitud radiométrica.

Para la sustracción de fondo, en este trabajo, se ha utilizado la imagen I_m de detección de puntos móviles llevado a cabo por el análisis temporal de la imagen, la imagen de fondo de referencia se evaluará en el momento t . La ecuación para calcular la similitud radiométrica que ya ha sido introducida anteriormente, se utiliza de nuevamente para realizar la selección de los puntos en movimiento en I_m , seleccionándolos de aquellos que son diferentes a la imagen de modelo de fondo B^t . La matriz binaria resultante de primer plano se obtiene como sigue:

$$f^t(x, y) = \begin{cases} 1, & \text{si } R(I_m(x, y), B^t(x, y)) < \sigma_s \\ 0, & \text{de otra manera} \end{cases}$$

Para la actualización de la Imagen de Fondo, en este trabajo, consideramos que todos los píxeles de la imagen pertenecen al modelo de fondo, incluso los correspondientes a los puntos que están cubiertos por las regiones de primer plano, por lo tanto en ese momento no son visible. La idea

principal del enfoque propuesto, es actualizar cada píxel de acuerdo con las variaciones exhibidas por todos los píxeles de la imagen con el mismo valor de intensidad. Para cada píxel de la ganancia fotométrica se evalúa como sigue:

$$\Lambda^{t-1}(x, y) = \frac{I^t(x, y)}{B^{t-1}(x, y)}$$

Donde: $B^{t-1}(x, y)$ Es el modelo de fondo en el paso $t - 1$.

$I^t(x, y)$ Es la imagen actual.

Las ganancias fotométricas medidas en todos los píxeles estáticos que tienen el mismo valor de intensidad $B^{t-1}(x, y) = B_i$ son consideradas para la evaluación de una ganancia media fotométrica de la siguiente manera:

$$\mu(b_i) = \frac{1}{N(b_i)} \sum_{\{(x,y) \in I'_s | B^{t-1}(x,y)=b_i\}} \Lambda^{t-1}(x, y)$$

Donde: $\{b_i\}_{i=1...n}$ Son los diferentes valores de intensidad que puede tener un píxel.

$N(b_i)$ Es el número de píxeles en la imagen de fondo $B^{t-1}(x, y)$ con valor de intensidad b_i

La ecuación anterior, se evalúa con todos las posibles valores de intensidad. La regla para la actualización de la imagen de fondo es:

$$B^t(x, y) = B^{t-1}(x, y)\mu(B^{t-1}(x, y))$$

De esta manera se consigue la agrupación de los píxeles en los dos grupos, pertenecen al fondo o pertenecen a objeto en movimiento. Utilizando este método, se consigue un algoritmo robusto para la segmentación, el cual combina la sustracción de fondo con el análisis temporal de la imagen. El uso de la similitud radiométrica entre regiones para comparar píxeles, ha demostrado resolver los problemas ocasionados por el ligero movimiento de la vegetación, del cambio gradual en la iluminación y también la eliminación de los fantasmas que se producen por el movimiento de objetos que realmente pertenecen al fondo. La lectura de este documento, ha dado ideas para seguir el camino de la sustracción de fondo para conseguir el objetivo de segmentar el movimiento.

2.3.2. Diferencia de imágenes con sustracción del fondo basado en diferencia dinámica

La técnica de diferencia de imágenes con sustracción de fondo basado en diferencia dinámica o BSFD por sus siglas en inglés (*Background Subtraction with Frame Difference*), fue utilizada por Stalin Alex y Amitabh Wahi [Stalin y Amitabh, 2014], utilizando los métodos FD y de sustracción de fondo. BSFD presenta el concepto de fondo dinámico, para lo que se utilizan las imágenes f_i , f_{i-5} y f_{i+5} con el fin de sobreponerse a los problemas que presenta FD y así poder detectar los objetos que se mueven de manera lenta. En su documento, se indica que el primer paso es, realizar una actualización

dinámica del fondo de la imagen que se está procesando, esta se obtiene comparando las imágenes del fondo inicial f_i , contra las imágenes que estén cinco cuadros antes y cinco cuadros después f_{i-5} , y f_{i+5} respectivamente. Después se sustrae el fondo de la imagen donde se quiere detectar el objeto en movimiento. Por otro lado, para llevar a cabo esta técnica, se aplica previamente el filtro mediana a la imagen original, con la finalidad de suprimir el ruido sal y pimienta y los valores de pixel que están lejos de sus rangos normales.

En el método de sustracción del fondo basado en diferencia dinámica, se dispone de la imagen de fondo (que es estática), y los objetivos (los elementos que se desplazan); en él nos encontrarnos con objetos que después de un tiempo de permanecer estáticos, pasan a formar parte del fondo, por lo que la actualización dinámica del fondo es esencial para la detección de los elementos que se están desplazando, y así diferenciarlos de los que ahora permanecen estáticos. La actualización de la imagen de fondo, se puede lograr mediante el método FD.

Las tres imágenes f_i , f_{i-5} y f_{i+5} , se obtienen directamente del video, y con estas se generan otras 2 imágenes (matrices de diferencia), D_{i-5} y D_{i+5} mediante el método FD; después con estas dos matrices de diferencia se realiza la operación AND para obtener el objeto en movimiento M. Se realiza la sustracción de M en f_i para obtener el fondo de la imagen, el cual será B. Luego con una operación OR de B y la imagen de referencia, se produce la imagen de fondo dinámica DB; ahora la imagen de referencia se puede remplazar con DB para que sea la nueva imagen de fondo. Finalmente se realiza la sustracción entre DB y f_i , junto con una operación de umbralado para detectar los objetos en movimiento. La figura 2-5, muestra el diagrama de flujo de esta metodología.

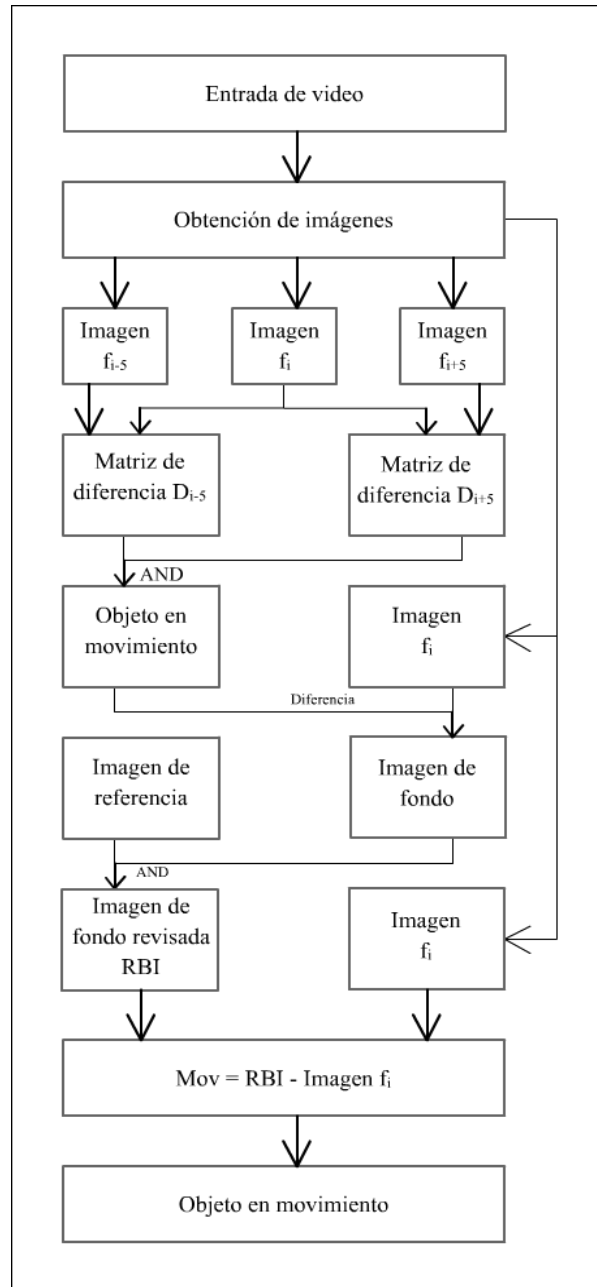


Figura 2-5. Algoritmo BSFD con diferencia dinámica.

Las pruebas de este experimento se realizaron tanto al aire libre como en lugares cerrados. En la figura 2-6 se muestran los resultados tanto del algoritmo propuesto como de un algoritmo existente, no mencionado en [Stalin y Amitabh, 2014]; se puede ver claramente que el algoritmo propuesto tiene una mejor efectividad y exactitud en detectar el objeto en movimiento.

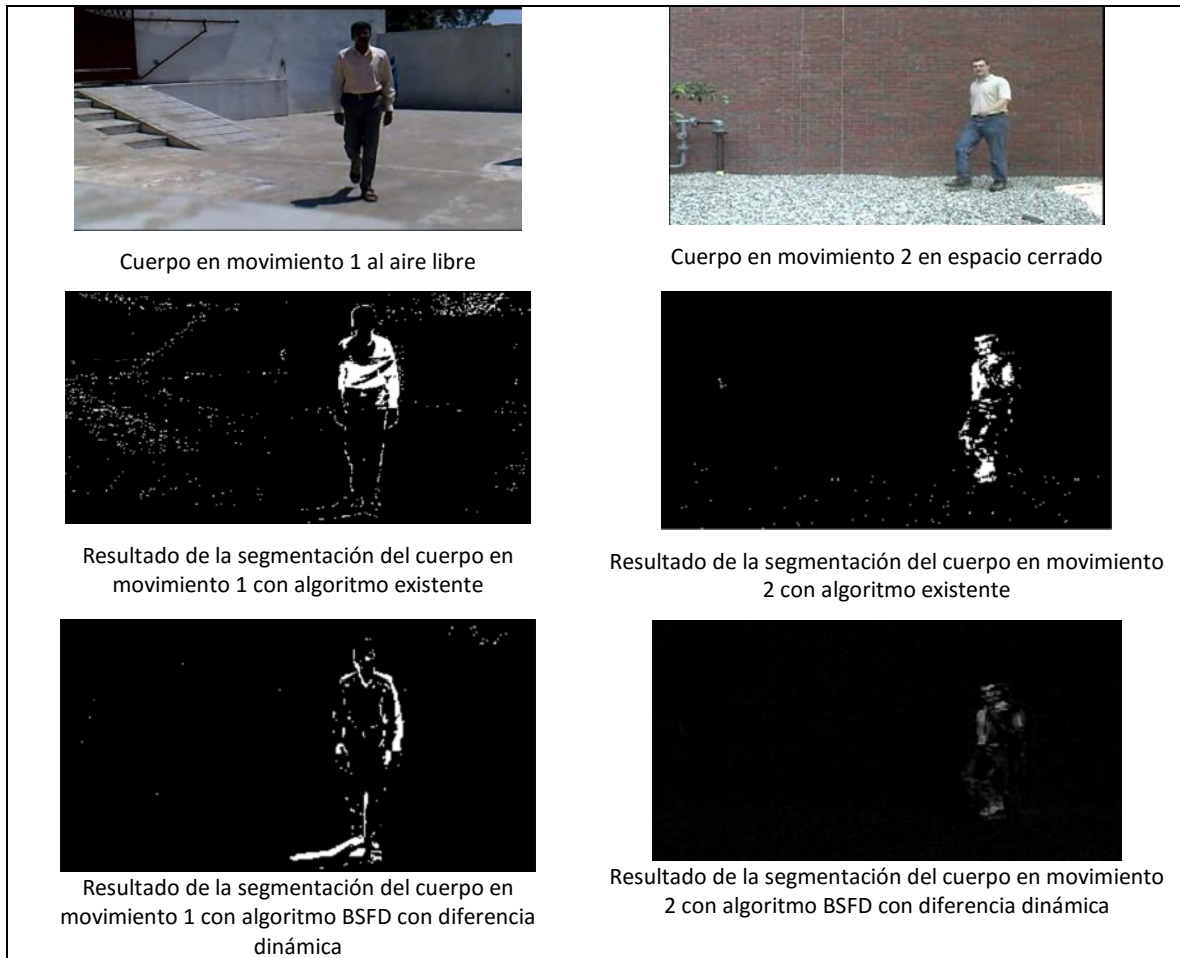


Figura 2-6. Comparación de los resultados con un algoritmo existente y el algoritmo BSFD con diferencia dinámica en interiores y exteriores.

El nuevo algoritmo propuesto en este ejemplo, muestra una mejoría respecto a los algoritmos existentes, como resultado de la mezcla de los algoritmos de FD y de sustracción de fondo, además del nuevo ingrediente que consiste en la actualización dinámica de la imagen de fondo. Se aprecia una mejoría notable en los resultados, tanto en interiores como en exteriores. Los resultados muestran que es posible su aplicación en tiempo real.

2.3.3. Uso de herramientas morfológicas

Se cita el documento [Jong y Hang, 2001], el cual tiene como título "*Efficient region-based motion segmentation for a video monitoring system*". Aquí se explica el procedimiento que se siguió para segmentar el movimiento en un sistema de monitoreo de vigilancia. Este método se divide en 2 etapas: la primera consiste en la detección del movimiento utilizando un método con un umbral que se adapta automáticamente, y la segunda etapa consiste en la segmentación del movimiento, donde los píxeles con intensidad similar e información del movimiento son segmentados utilizando un algoritmo de agrupamiento k -medias en las regiones binarias resultantes de la primera etapa. En el

documento se detalla el uso de herramientas morfológicas tanto para la supresión de artefactos, como para la segmentación de objetos, y a pesar de que el elemento de estructura es de 5x5 píxeles y de que la computadora que fue utilizada hacía uso de un procesador Pentium III a 850 MHz se obtuvieron buenos resultados al asignar valores de “1” para los píxeles que sufrieron algún cambio, y valor de “0” para píxeles donde no se detectaron cambios (fondo). En la figura 2-7, se muestra el proceso y el resultado de aplicar este método:

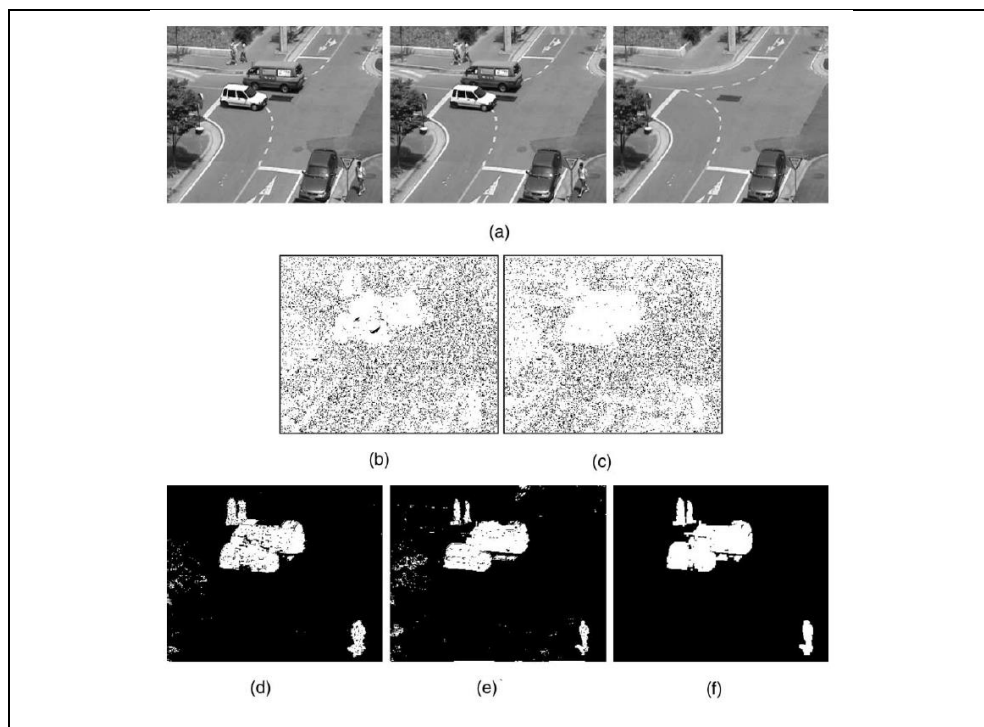


Figura 2-7. Segmentación del movimiento en un video tomado en una carretera, utilizando morfología para eliminar los artefactos, (a) Nos muestra 3 imágenes: a la izquierda el cuadro 12 del video, en el centro el cuadro 14 y hasta la derecha la imagen de fondo. En (b) se muestra sustracción de fondo del cuadro 12. En (c) se muestra sustracción de fondo del cuadro 14. (d) Nos muestra la diferencia de imágenes con umbral 54. (e) Es la diferencia de imágenes con umbral 47. (f) Es la imagen binaria con máscara de movimiento.

Con estos resultados, queda claro que la implementación de métodos morfológicos es posible para obtener una mejor segmentación del movimiento, al no haber un costo computacional tan alto, además de observar una clara mejoría en los resultados.

2.3.4. Detección humana basada en la detección de bordes de Canny y FD

En este método propuesto por Wen Jun-Qin [Wen, 2013], se mezclan las técnicas de FD y de detección de bordes de Canny [Canny, 1986] mediante la operación lógica AND.

El primer paso en el algoritmo de Canny es suavizar la imagen, para lo que se utiliza el desenfoque Gaussiano con el fin de eliminar el ruido que pudiera detectarse erróneamente como aristas. Después se aplica el filtro de Sobel [Gonzalez y Woods, 2008] para encontrar las zonas en donde la

variación de la intensidad es mayor, lo que se le conoce como localización de gradientes. El siguiente paso es eliminar los puntos que no son considerados como gradientes máximos resultantes del paso anterior. El paso 4 es conocido como umbralado doble, con el que los pixeles restantes se categorizan en fuertes, débiles y no bordes. Finalmente los puntos que se categorizaron como fuertes se quedan, manteniéndose únicamente los puntos débiles que estén tocando puntos fuertes; a este último paso se le conoce como seguimiento de bordes por histéresis.

En el método de detección humana basada en la detección de bordes de Canny y FD, el primer paso es realizar un preprocesamiento con la finalidad de reducir el costo computacional al mismo tiempo que se realzan algunos detalles de la imagen; ésta se convierte a escala de grises y se realiza una ecualización del histograma para obtener un mejor contraste entre el fondo y los objetos; de esta manera se mejora la eficiencia al requerirse una menor cantidad de información a ser procesada (de 24 a 8 bits de profundidad).

En la figura 2-8, se muestra un ejemplo de una imagen con el histograma ecualizado. En (a) se muestra la foto original; en (c) se aprecia el histograma de la imagen original; en (b) se ve la imagen ecualizada y finalmente en (d) se muestra su histograma.

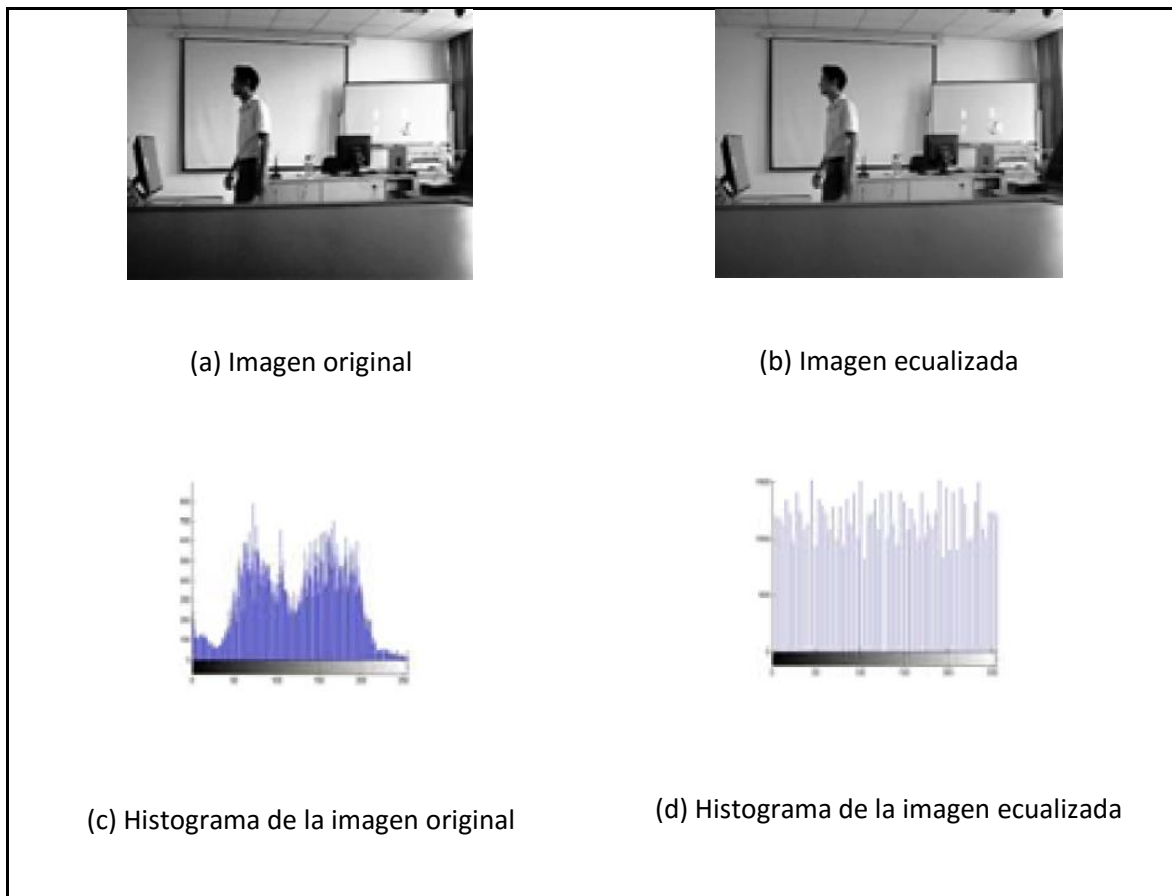


Figura 2-8. Ejemplo de la ecualización de una imagen.

Después de ecualizar las imágenes, se realiza una FD entre dos imágenes consecutivas para detectar las zonas donde hay movimiento y seguidamente se realiza una extracción de bordes utilizando el operador de Canny en la secuencia de imágenes preprocesadas.

En este método se utilizó el detector de bordes de Canny debido a su buen desempeño y precisión en ambiente de ruido, además de que puede capturar las características estructurales e información de bordes más efectivamente.

El paso final del método de Canny con FD consiste en llevar a cabo una operación lógica AND entre el resultado de FD y la extracción de bordes realizada con dicho operador. En la figura 2-9 se muestran (a) el resultado de aplicar FD; (b) el resultado de la extracción de bordes con el operador de Canny y (c) el resultado de la operación AND entre las dos imágenes anteriores.

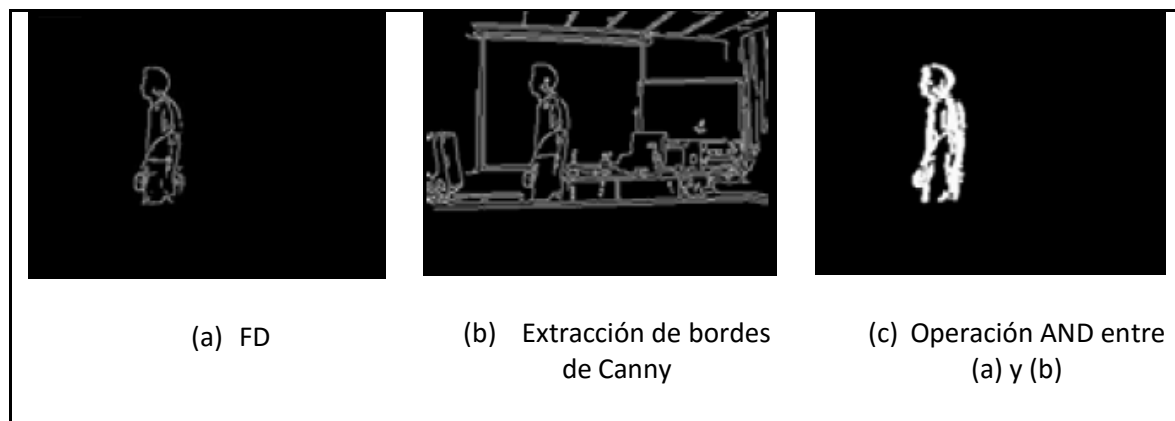


Figura 2-9. Ejemplo del resultado al aplicar el algoritmo de detección de bordes de Canny y FD.

El algoritmo resuelve el problema del pseudomovimiento y la supresión de artefactos que aparecen al ejecutar FD; también permite obtener de manera acertada la zona del movimiento. Los experimentos muestran la exactitud del método, la información de los bordes y la mejoría de la calidad del objeto en movimiento. Algunos de los resultados muestran que la exactitud de la detección depende de la velocidad del objeto en movimiento, así como se reconoce la importancia del preprocesamiento de las imágenes para obtener mejores resultados. El análisis de la región del objeto en movimiento es de particular importancia en este método. Se alcanza una muy buena supresión de artefactos y una mejor exactitud en la detección del movimiento. Una posible desventaja es su dificultad de funcionamiento en tiempo real, dado que necesita varias operaciones de filtros y la aplicación del algoritmo de Canny, por lo que existe una gran dependencia de los cuadros por segundo a ser procesados.

2.3.5. Segmentación de video con FD de 3 imágenes y operador de Kirsch

Este método realiza la segmentación del movimiento mediante la combinación del operador de Kirsch y la diferencia de tres cuadros TFD. En la diferencia de imágenes normalmente se utiliza FD (dos imágenes sucesivas), pero en este algoritmo se utilizan tres imágenes para realizar la comparación espacio-temporal entre los objetos en movimiento. TFD toma 3 imágenes consecutivas que son $f(x, y, t - 1)$, $f(x, y, t)$ y $f(x, y, t + 1)$ respectivamente. También se utilizan las imágenes $D_{t,t-1}(x, y, t)$ y $D_{t,t+1}(x, y, t)$ que se obtienen mediante:

$$D_{t,t-1}(x, y, t) = \begin{cases} 1, & \text{si } |f(x, y, t) - f(x, y, t - 1)| > T \\ 0, & \text{si } |f(x, y, t) - f(x, y, t - 1)| \leq T \end{cases}$$
$$D_{t,t+1}(x, y, t) = \begin{cases} 1, & \text{si } |f(x, y, t) - f(x, y, t + 1)| > T \\ 0, & \text{si } |f(x, y, t) - f(x, y, t + 1)| \leq T \end{cases}$$

Donde: T es un umbral para obtener la imagen binaria.

$D_{t,t-1}(x, y, t)$ Es el módulo de la diferencia de la imagen actual y la imagen anterior.

$D_{t,t+1}(x, y, t)$ Es el módulo de la diferencia de la imagen actual y la imagen posterior.

Se considera que los pixeles pertenecen a un objeto en movimiento si su valor es uno, y es parte del fondo si su valor es igual a cero.

En las imágenes obtenidas mediante FD se extrae del video el objeto en movimiento, pero no de manera exacta, porque el resultado de la diferencia entre dos imágenes no muestra únicamente el objeto en movimiento; el fondo cubierto por el objeto en movimiento en las imágenes previas aparecerá en las siguientes imágenes, y sus pixeles serán tomados erróneamente como objeto.

Para tener una mejor comprensión y análisis del movimiento, se utiliza la diferencia de tres imágenes en lugar de sólo dos imágenes (FD) para realizar la segmentación del movimiento. La fórmula en este caso puede ser escrita de la siguiente manera:

$$D_3(x, y, t) = D_{t,t-1}(x, y, t) - D_{t,t+1}(x, y, t)$$

De esta manera se puede obtener de manera rápida y un poco más precisa el contorno del objeto en movimiento; también se resuelve el problema de la reaparición del fondo y resuelve un poco el problema de las sombras generadas por la iluminación. En la figura 2-10, se muestra el proceso y el resultado de aplicar este método.

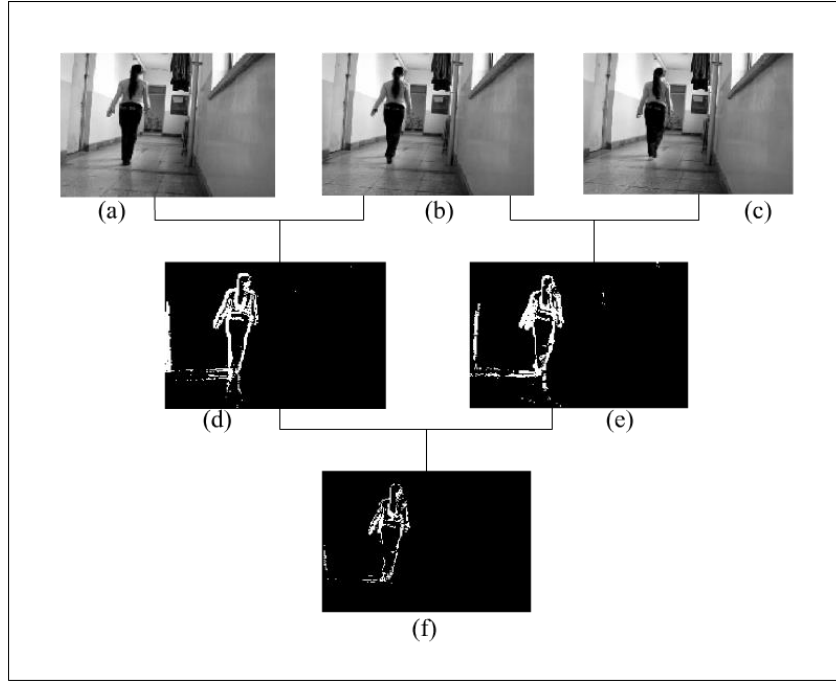


Figura 2-10. Proceso FD de tres imágenes, (a), (b) y (c) son 3 imágenes sucesivas en un video. (d) es el resultado de FD entre (b) y (a); (e) es el resultado FD entre (c) y (b); (f) es el resultado de restar (d) y (e).

Este método utiliza la detección de bordes basado en la entropía de bloques y el método de Kirsch adaptativo. La entropía de una imagen refleja cuánta información hay en esa imagen; si asumimos que una imagen está cuantificada por Q bits por pixel, entonces la imagen tiene $M = 2^Q$ niveles de gris; si el nivel de gris es $G_i = i$ ($i = 0, 1, 2, \dots, 2^Q - 1$), entonces la información propia de la imagen y la entropía queda:

$$p(G_i) = \frac{n_i}{n_{total}}$$

$$I(G_i) = \log_2 \left[\frac{1}{p(G_i)} \right]$$

$$H = \sum_{i=0}^{2^Q-1} p(G_i) I(G_i)$$

Donde: n_i Es la frecuencia del pixel de nivel de gris G_i en toda la imagen.

n_{total} Es el número total de pixeles en toda la imagen.

H Es la entropía.

La entropía indica la ecualización de la distribución del gradiente y la continuidad e integridad de los niveles de gris. La entropía es alta cuando hay demasiadas series de grises y están balanceadas en distribución. Debido a que los niveles de gris en cada parte de la imagen no están balanceados,

tal vez existan algunas zonas que tengan más detalles (grandes cantidades de información), mientras que otras partes no (menor cantidad de información). Cuando se utiliza el operador de Kirsch con el mismo umbral en toda la imagen, algunos detalles de la imagen pueden ser ignorados, lo que puede ocasionar algunas discontinuidades en los bordes, incluso podría no reconocerse todo el borde; para resolver este problema, se agrega como método de refuerzo una técnica morfológica adaptativa que conecta las líneas discontinuas con un elemento de estructura elíptico adaptativo para llevar a cabo una dilatación morfológica, tomando en cuenta la entropía en cada bloque procesado.

Los resultados muestran que mediante la técnica de diferencia de tres imágenes obtenemos robustez de una manera rápida y simple; al combinarla con el método propuesto de mejoramiento con el operador de Kirsch, se pueden superar los problemas de hoyos y fantasmas que se generan, mejorándose de esta manera los resultados. El umbral del operador de Kirsch no se puede definir de manera adaptativa, por lo que este método propone hallarlo utilizando el cálculo de la entropía de los bloques. Los resultados experimentales muestran que este algoritmo es capaz de responder a los cambios entre cuadros de manera rápida, por lo que la segmentación de los objetos en movimiento se realiza de manera efectiva.

Con este método se comprueba que también se puede procesar información posterior al cuadro de imagen que se está procesando, y no solamente la información actual y anterior. Al utilizar un método de segmentación basado en el reconocimiento de los bordes, es necesario implementar un método de rellenado de áreas para localizar los objetos en movimiento; este procedimiento es de fácil y rápida implementación.

2.3.6. Implementación de técnicas convencionales con modificaciones

En el artículo *Image segmentation using extended Edge operator for mammographic images*, que presentó el Dr. H. B. Kekre [Kekre, 2010], se describe el procedimiento que mejora la detección de tumores en imágenes mamográficas; en el documento se explica cómo se pueden modificar algunos filtros, y de esta manera obtener resultados más satisfactorios.

Generalmente los operadores Sobel, Prewitt y Kirsch [Gonzalez y Woods, 2008], utilizan máscaras de 3x3 píxeles para detectar bordes. Con el fin de detectar los tumores mamarios o para segmentar dichos tumores, se propone el uso de máscaras extendidas. Como ejemplo, el operador Sobel que realiza una transformación bidimensional sobre una imagen para localizar los gradientes en y y en x , G_y y G_x respectivamente. Este filtro utiliza un par de máscaras de 3x3 píxeles con las cuales se realiza la convolución con las imágenes. En la figura 2-11 se muestran las máscaras de Sobel correspondientes:

-1	-2	-1	-1	0	+1
0	0	0	-2	0	+2
+1	+2	+1	-1	0	+1
Gx			Gy		

Figura 2-11. Valores de la máscara del operador de Sobel.

Esto implica que el resultado tras aplicar el operador en algún punto de la imagen donde se localice una región de intensidad constante, es un vector cero, y en una región con borde será un vector con valores diferentes de cero. Para obtener una mejor segmentación, se hace un cambio en el tamaño de las máscaras de 3x3 a 5x5, con lo cual se muestra más claramente las regiones con bordes. Estas máscaras se muestran en la figura 2-12, donde también se puede apreciar que una máscara es simplemente la rotación en 90° de la máscara anterior.

+2	+2	+4	+2	+2
+1	+1	+2	+1	+1
0	0	0	0	0
-1	-1	-2	-1	-1
-2	-2	-4	-2	-2
Gx				

+2	+1	0	-1	-2
+2	+1	0	-1	-2
+4	+2	0	-2	-4
+2	+1	0	-1	-2
+2	+1	0	-1	-2
Gy				

Figura 2-12. Máscaras de convolución de Sobel extendidas a 5x5 píxeles.

Gracias a este artículo, podemos darnos cuenta de que es posible utilizar métodos tradicionales o comunes de procesamiento de imágenes, y/o al mismo tiempo modificarlos, para poder realizar análisis de imágenes de manera más exacta, o con resultados más satisfactorios. En este caso se puede ver cómo se amplía el análisis de la vecindad en los píxeles, dando como resultado una mejoría, con más nitidez, y una mejor demarcación, con lo que se logra una mejor segmentación de las áreas de interés.

2.3.7. Segmentación de movimiento y rastreo usando cortes normalizados.

En este documento se propone un algoritmo de segmentación de movimiento que divide una secuencia de imágenes en grupos de píxeles con los movimientos más sobresalientes. Se construye

un grafo ponderado sobre los píxeles de la secuencia de imágenes según la conexión que hay en su vecindario espacio-temporal. En cada píxel, definimos los vectores que capturan sus perfiles de movimiento en la distribución de probabilidad de la velocidad del grafo de la imagen. Las distancias entre perfiles de movimiento se utilizan para asignar un peso en los bordes de la imagen. Utilizando cortes normalizados encontramos las particiones más sobresalientes del grafo espacio temporal formado por la serie de imágenes. Para segmentar secuencias de imágenes demasiado largas, hemos desarrollado un procedimiento recursivo de actualización que incorpora el uso de segmentación de fotogramas anteriores, para encontrar de manera eficiente la correspondencia del grupo en el nuevo marco.

La agrupación basada en el movimiento común, o lo que en la escuela Gestalt se llama el factor del "destino común", es una de las señales más fuertes para obtener la segmentación de una secuencia de imágenes en objetos separados. Sin embargo, la implementación de esta capacidad perceptual ha demostrado ser muy desafiante. Para los sistemas de visión por computadora. Los primeros enfoques se basaron en tratar de estimar el flujo óptico, después se buscó en discontinuidades. Esto resultó difícil debido a una serie de razones:

La medición de flujo óptico es difícil, en las zonas de poca textura o donde la textura es principalmente unidimensional, y cualquier imagen real está obligada a tener grandes regiones con estas propiedades.

Para hacer frente a este problema, se propuso hacer cumplir las limitaciones para interpolar en el campo de flujo. Sin embargo, esto plantea el requisito de que uno debe saber primero la segmentación con el fin de evitar el suavizado a través de discontinuidades de movimiento. En los últimos años, un nuevo marco ha aparecido con base en la idea de la estimación simultánea de múltiples modelos globales de movimiento y sus soportes espaciales (las llamadas "capas"). Esta idea ha evolucionado a través de una serie de documentos, tal vez las corrientes más limpias se basan en el uso de la maximización expectativa o EM por sus siglas en inglés, (englobado en la teoría estadística). Por lo general los modelos de movimiento son modelos paramétricos 2D traslacional, afín o proyectiva.

En nuestra opinión, la debilidad principal del enfoque EM para la segmentación de movimiento en capas está en la fase de inicialización. ¿Cuántos modelos han de inicializarse, dónde y qué deben ser? y ¿Cómo se puede saber cuándo se ha alcanzado una solución óptima global? Uno de los primeros enfoques representativos propuesto por Ayer y Sawhney [Ayer y Sawhney, 1995], utiliza el principio de descripción de longitud mínima, para seleccionar el número de modelos. La inicialización se hace dividiendo la imagen en un número fijo de "mosaicos", y se estiman los parámetros de movimiento iniciales en estos mosaicos, luego se usan estos como condiciones iniciales para el algoritmo EM. Sin embargo, nuestra experiencia ha demostrado que la búsqueda de una buena inicialización, sigue siendo un problema persistente. Sin lugar a dudas, una mayor investigación en esta área proporcionará mejoras en esta área; aun así, en este trabajo se optó por

desarrollar un enfoque alternativo que incorpora información de movimiento a través de los vecindarios espaciales y temporales, y busca una solución global óptima de segmentación sin la dificultad de inicialización.

Para realizar los cortes normalizados, se procede de la siguiente manera:

Sea $G = (V, E)$ un grafo ponderado, el grafo G se puede dividir en dos conjuntos disjuntos, A y B , de tal forma que $A \cup B = V$, y $A \cap B = \emptyset$, mediante la simple remoción de bordes que unen ambas partes. Los pesos totales en esos bordes removidos reflejan el grado de disociación entre los conjuntos A y B . En el campo de teoría de grafos, se llama “corte”:

$$corte(A, B) = \sum_{u \in E} w(u, v)$$

Aunque hay algoritmos computacionales muy eficientes para encontrar las particiones que minimizan el valor del corte, este criterio favorece a las particiones que tienen tamaños pequeños. Se definen “cortes normalizados” (Ncut) como medida objetiva de disociación.

$$N \text{ corte}(A, B) = \frac{corte(A, B)}{asso(A, V)} + \frac{corte(A, B)}{asso(B, V)}$$

Donde: $asso(A, V) = \sum_{u \in A, t \in B} w(u, t)$ Es el total de conexiones de los nodos en A con todos los nodos del grafo.

$asso(B, V)$ Se define de manera similar, es el total de conexiones de los nodos en B con todos los nodos del grafo.

También podemos definir una medida objetiva para la asociación total dentro de los grupos para una partición determinada:

$$N \text{ asso}(A, B) = \frac{asso(A, A)}{asso(A, V)} + \frac{asso(B, B)}{asso(B, V)}$$

Donde $asso(A, A)$ y $asso(B, B)$ son los pesos totales de los bordes que conectan los nodos de A y B respectivamente.

$$N \text{ corte}(A, B) = 2 - N \text{ asso}(A, B)$$

De esta manera, la minimización de la disociación entre los grupos y la maximización de la asociación dentro del grupo, puede ser satisfecha simultáneamente.

Sea W el peso de la matriz del grafo, y D la matriz diagonal con $D(i, i) = \sum_j W(i, j)$. En [Shi y Malik, 1997] se demuestra que minimizar N corte se puede reducir a minimizar el cociente Rayleigh:

$$\min_y N \text{ corte} = \min_y \frac{y^T (D - W) y}{y^T D y}$$

Con la condición $y_i \in \{1, -1\}$ y $y^T D_1 = 0$. Dándole a y valores reales, podemos minimizar la ecuación anterior, con sus restricciones, resolviendo para el segundo vector propio más pequeño, del sistema de valores propios generalizados.

$$(D - W)y = \lambda Dy$$

El vector y se puede ver como un vector indicador para la partición. Aún más, los vectores propios subsecuentes son soluciones con valores reales que conforman la subpartición óptima.

El criterio del corte normalizado ha sido exitosamente utilizado para segmentar imágenes estáticas basándose en la información del brillo, color y textura. Para extender esta herramienta en la segmentación de movimiento, primero se necesita seleccionar una W apropiada según la noción del perfil de movimiento. En la figura 2-13 se puede ver un ejemplo tomado del documento que muestra la segmentación que logra este algoritmo.

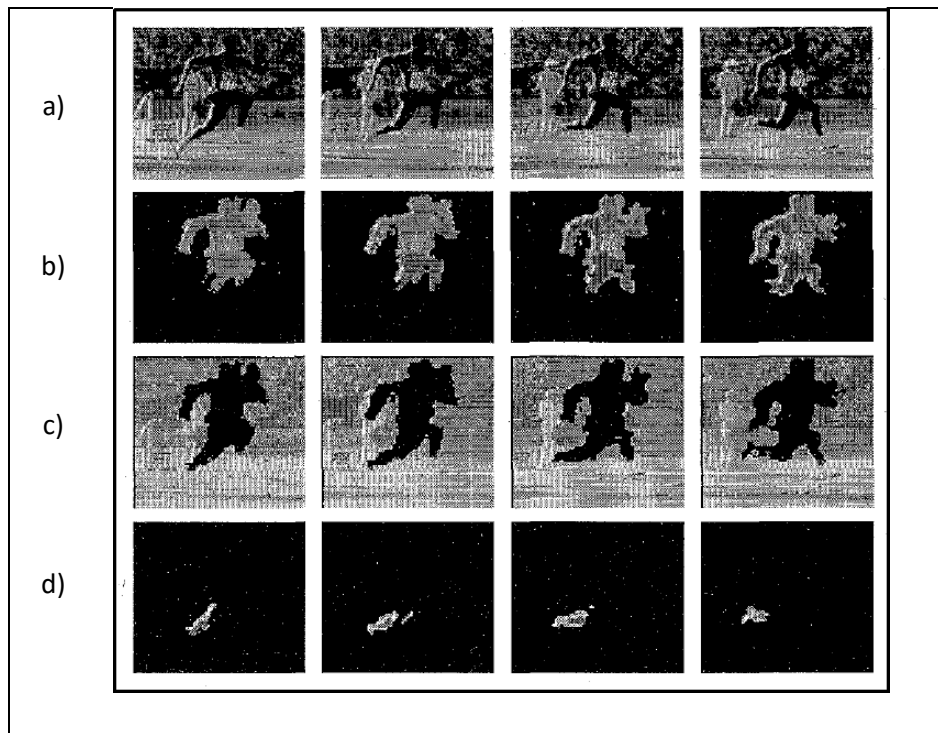


Figura 2-13. Resultados de algoritmo de segmentación utilizando cortes normalizados. El renglón a) muestra una secuencia de una carrera de Carl Lewis, nótese que el fondo se desplaza hacia la izquierda, por el movimiento de la cámara que mantiene al corredor en el centro de la imagen, en este caso la sustracción de fondo no funcionaría como

técnica de segmentación. Las dimensiones de la imagen original es de 200 x 190, los parches de la imagen que se utilizan para crear los cortes del grafo son de 3 x 3. Cada uno de los parches de la imagen se conecta con otros parches que se encuentran a menos de 3 cuadros de diferencia. b) muestra los píxeles del corredor, c) es el fondo en movimiento y d) es parte de la pierna izquierda, lo cual resulta de que esta parte sufre de una rotación hacia arriba en la secuencia de imágenes.

El desarrollo de esta técnica, utiliza la información espacio temporal en tres dimensiones y construye un grafo ponderado tomando cada pixel como un nodo, y conecta los píxeles espacio y temporalmente en su vecindario. Se define un vector de perfil de movimiento en cada pixel de la imagen que captura la probabilidad de distribución de velocidad de la imagen en ese punto.

La implementación de esta técnica es muy costosa computacional mente aún con imágenes en baja resolución, es por eso que después de algunos intentos por seguir este camino, e intentar optimizar el desempeño del algoritmo, se decidió tomar otra solución. Un resultado muy positivo de este algoritmo es que la segmentación de los cuerpos en movimiento es muy exacto, y al poner la restricción de que los píxeles tienen que estar cuatro conectados, la generación de ruido o artefactos es nula; Si el trabajo requiere una segmentación fina de los objetos en movimiento y no está sujeta al tiempo de procesamiento, esta opción sería de las más recomendables.

2.3.8. Una revalorización de FD en detección de movimiento rápida y robusta.

En este documento [Migliore et al, 2006] se propone un proceso para segmentar el movimiento, se dice que es veloz y robusto para la clasificación de los píxeles, utiliza una combinación de FD y sustracción de fondo, sin la necesidad de proceso de aprendizaje del fondo. Presenta el algoritmo de diferencia conjunta, este utiliza la información de FD para corregir la clasificación de los píxeles realizada por la sustracción de fondo, y va actualizando selectivamente el modelo de la imagen de fondo. La segmentación es también alcanzada en cambios del medio ambiente como la iluminación. El algoritmo funciona en 15fps con una resolución de 640 x 480 píxeles en imágenes a color.

Los métodos de sustracción de fondo y FD tienen dos problemas graves causados por la velocidad de los cuadros (demasiado lento o demasiado rápido), y la apertura o aparición de fantasmas. Para resolver esto se han intentado varias soluciones. En la siguiente imagen se muestra de manera gráfica, estos dos sucesos que son resultados de la diferencia de imágenes en el tiempo t y $t-1$. En la figura 2-14 se ejemplifican estos inconvenientes.

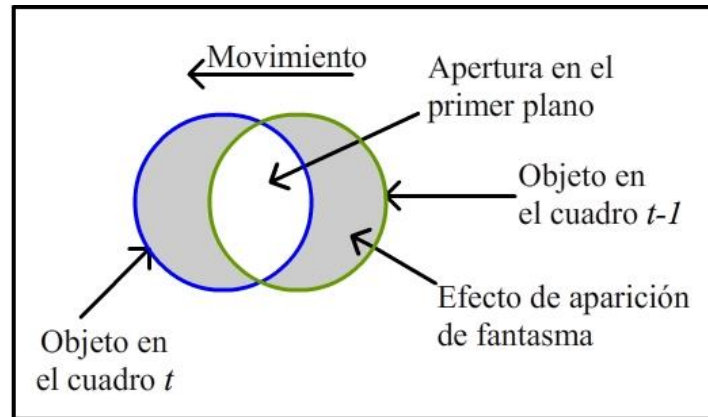


Figura 2-14. Ejemplificación de inconvenientes en FD y sustracción de fondo.

Para detectar un objeto en movimiento dentro del cuadro actual, se propone una técnica híbrida que utiliza tanto FD y sustracción de fondo. Para clasificar los píxeles de una manera sólida, en el momento t combinamos la diferencia entre el marco actual F_t y el anterior F_{t-1} , con la diferencia entre el marco actual y el modelo actual de fondo B_t . Para arrancar este proceso, la primera imagen de la secuencia es utilizada como modelo de fondo inicial, y se actualiza dinámicamente con el nuevo marco según la segmentación de movimiento. Se distinguen cinco clases que queremos detectar en este proceso: el primer plano (FG), el fondo (BG), el efecto fantasma (GH), la apertura de primer plano (FA), y persona caminando (WP). Como en los otros algoritmos, los píxeles del primer plano pertenecen a un objeto en movimiento, de lo contrario, pertenecen al fondo de la escena; las otras tres clases son más interesantes.

Para describir mejor el problema de efecto fantasma, sea p un píxel en primer plano en la posición (x, y) en F_{t-1} ; en F_t estará en un lugar diferente, por lo que la diferencia entre las dos imágenes en (x, y) tendrán un píxel de primer plano en lugar de un píxel de fondo. En la Figura 2-14 podemos notar que la pelota está moviéndose de derecha a izquierda, y en la diferencia de imágenes el efecto fantasma está presente a la derecha. Por otro lado, la apertura primer plano es debido a los píxeles con textura similar. En la Figura 2-14, en la zona blanca entre los dos círculos se nota este problema.

Los métodos utilizados en este proceso para realizar la agrupación de los píxeles, requiere de procesos computacionales no muy costosos, es de rápido funcionamiento y los resultados son muy aceptables, este trabajo muestra que el seguimiento de este camino es recomendable para llevar a cabo nuestro objetivo. Su implementación en tiempo real, el procesamiento de las tres capas de color, el costo computacional y el análisis del algoritmo fue alentador para realizar alguna implementación que tuviera características similares, pero no idénticas a este algoritmo.

Este nuevo método denominado diferencia conjunta, mezcla la diferencia de imágenes y el método FD. Los cuales según el documento, realiza el proceso de manera más rápida y con mejores resultados. El proceso fue desarrollado en Italia, y el documento fue aprobado en 2006 por ACM.

2.4.Conclusiones del capítulo

Así las cosas, con la velocidad de los procesadores actuales, con el estudio de los métodos mencionados y observando los resultados obtenidos, nos damos cuenta de que es posible implementar métodos morfológicos, diferencia de imágenes o sustracción de fondo, utilizando secuencias de imágenes en colores de resolución media, por ejemplo, a una tasa de 12 imágenes por segundo, con resolución de 640x480 pixeles, con 24 bits por pixel; esto se ajusta a los estándares de videovigilancia que se utilizan actualmente.

La mezclar técnicas para obtener una mejor segmentación, así como buscar métodos para reducir el costo computacional de los algoritmos, nos puede resultar en métodos efectivos de segmentación en tiempo real, de objetos en movimiento, lo cual constituye el objetivo principal de este trabajo de tesis.

Capítulo 3. Conceptos básicos y definiciones

Para entender mejor la segmentación del movimiento en imágenes digitales, es recomendable familiarizarse con algunos términos comúnmente utilizados en el análisis de imágenes digitales. En este capítulo, se describen algunos conceptos, y se intenta dar al lector una idea de la manera en que se procede con el análisis de las imágenes digitales.

3.1. Procesamiento digital de imágenes

El procesamiento digital de imágenes consiste en el uso de algoritmos informáticos para realizar mediante computadoras el procesamiento de imágenes digitalizadas, con la finalidad de realizar alguna mejora en estas imágenes, restaurarlas, etc., o realizar algún análisis en particular en estas imágenes. El procesamiento digital de imágenes está relacionado con el procesamiento de señales digitales.

3.2. Historia

Muchas de las técnicas de procesamiento digital de imágenes se desarrollaron en la década de los sesenta en muchos centros de investigación de todo el mundo, con aplicaciones en la visión por computadora, en imágenes de satélite, imágenes médicas, videoteléfono, reconocimiento óptico de caracteres, retoque fotográfico, etc. Con los equipos informáticos de esa época el costo del procesamiento era bastante alto. Eso comenzó a cambiar en la década de los años setenta, pues el procesamiento digital de imágenes proliferó con computadoras más baratas y el uso de hardware dedicado. Las imágenes entonces podían ser procesadas en tiempo real en algunos problemas específicos, tales como la conversión entre los estándares de televisión. Al hacerse más rápidas las computadoras digitales de propósito general, comenzó a acercarse su velocidad a la del hardware dedicado, por lo que los procesamiento pudieron ser más especializados e intensivos.

Ya en el siglo actual el pixel se ha abaratado muchísimo y con esto el procesamiento digital de imágenes; las computadoras pasaron a ser muy veloces con procesadores muy potentes a precios muy accesibles. El procesamiento digital de imágenes en las computadoras personales, se ha convertido en la forma más común del procesamiento de imágenes por computadora. Asimismo, las empresas transnacionales dedicadas al desarrollo de programas de procesamiento digital de imágenes, dedican sus esfuerzos al desarrollo de aplicaciones para funcionar mayoritariamente en computadoras personales.

En la actualidad, el procesamiento de imágenes en computadoras personales es más versátil y más barato. Un claro ejemplo es que actualmente existen anuncios en la televisión e Internet realizados con muy alta calidad que han sido grabados y editados con tabletas digitales.

3.3. Algunos conceptos importantes

Para que el procesamiento de imágenes se considere exitoso, se analizarán los resultados que se obtienen; gran parte de la calificación dependerá de los resultados visuales que obtengamos de ello, por lo tanto, es necesario explicar brevemente la forma en que funciona el sistema de visión humano y la manera en que se lleva a cabo el análisis y/o las transformaciones digitales.

3.3.1. El ojo humano

De los cinco sentidos externos que tiene el ser humano, el que más información aporta al cerebro es el sentido de la visión. Para poder realizar el análisis de imágenes es necesario entender algunos conceptos básicos que se relacionan con la forma en que los humanos percibimos el mundo a través del sentido de la vista. En la figura 3-1, se muestra un esquema básico de las partes del ojo humano.

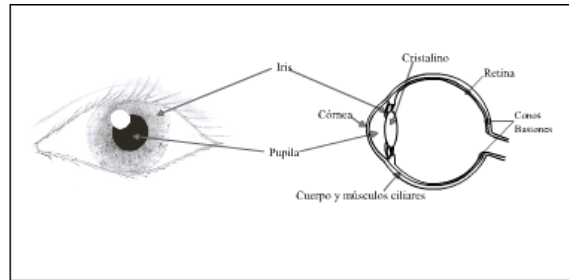


Figura 3-1. Esquema básico del ojo humano.

El ojo humano tiene varios tipos de células con funciones muy variadas. Entre otras están dos tipos de células fotosensibles llamadas bastoncillos y conos.

Los bastoncillos son células que reaccionan con la luminosidad; estas células se saturan en condiciones de mucha luz.

Las segundas, los conos, son las responsables de percibir la visión del color y la visión precisa. Existen tres tipos de conos: los sensibles al rojo, los sensibles al verde y los sensibles al azul. El cerebro interpreta la gama de colores a partir de la cantidad de estimulación que recibe de los conos, con base en estos tres colores básicos. Esta es la razón por la cual las imágenes digitalizadas de color utilizan al menos tres conjuntos de datos para guardar la información, cada uno de ellos asociado a un color diferente: rojo, azul y verde.

3.3.2. Modelo de color RGB

En las imágenes digitales en colores se asigna un plano a la intensidad del color rojo, uno a la intensidad del color verde y, finalmente uno a la intensidad del color azul de la imagen en colores que se captura. Este modelo de color es comúnmente conocido como RGB, por las siglas de los colores *Red*, *Green*, *Blue* – rojo, verde, azul en el idioma inglés.

Cada uno de los tres planos de las imágenes digitalizadas desplegadas en el modelo RGB, tienen el mismo número de puntos a lo ancho (eje *x*), y a lo alto (eje *y*); cada uno de los puntos compuestos por los tres niveles de colores se llama pixel. El formato más utilizado para el nivel de color de los pixeles en las imágenes digitales es de 8 bits para cada pixel en cada plano de color (en total 24 bits/pixel), con lo cual se puede asignar un valor para cada plano de color en el rango de 0 a 255, en donde 0 es el nivel más bajo (considerado el negro) y 255 el nivel más alto obtenible de ese color, el blanco. En este caso pueden distinguirse un total de 16.7×10^6 colores ($256 \times 256 \times 256$). El nivel

de color puede no estar limitado a 8 bits y el modelo RGB no es el único modelo de color que existe para desplegar y codificar imágenes.

Es también importante explicar que existen imágenes en colores, imágenes en niveles de gris e imágenes que solo tienen píxeles en blanco o en negro conocidas como imágenes binarias.

Las imágenes se pueden procesar en el dominio espacial o en el dominio de la frecuencia. Los objetivos de procesar imágenes digitales son muy variados en los diferentes campos de la ciencia y la tecnología. Existen muchos paquetes de programas que se pueden obtener de manera gratuita en revistas, Internet, etc., para el procesamiento de las imágenes digitales.

3.4. Procesamiento en el dominio espacial

El procesamiento de imágenes en el dominio espacial se realiza directamente sobre los píxeles de la imagen y el tipo de proceso que se realiza depende del objetivo a conseguir. Algunas de las metas comunes pueden ser las siguientes:

3.4.1. Procesamiento del color

Manejaremos la información del color de cada píxel según su ubicación en las imágenes digitales mediante la forma $(M \times N \times C)$ donde:

M es la coordenada del píxel en el eje de horizontal

N es la coordenada del píxel en el eje vertical

C es uno de los 3 planos de color.

En la figura 3-2 se ejemplifica esto:

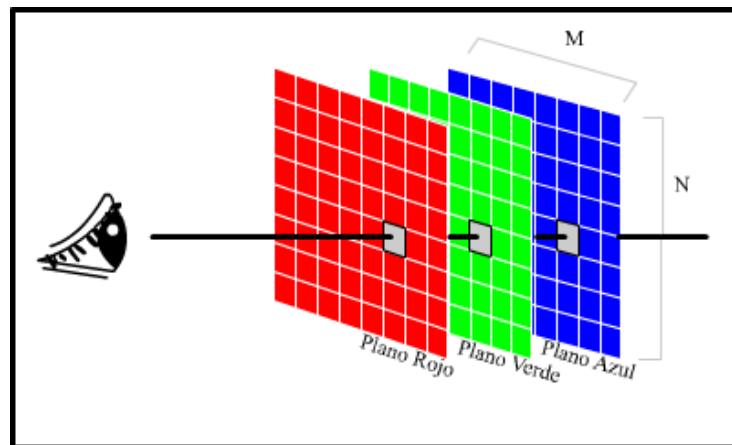


Figura 3-2. Los 3 planos de color del modelo de color RGB.

Cada uno de los 3 planos de color es una imagen en escala de grises asociada a cada color. Las imágenes RGB pueden ser procesadas aplicando máscaras o vecindades en cada plano de color por separado, considerando una matriz de 3 dimensiones $M \times N \times 3$. Para ajustar la intensidad del color en el caso de las imágenes RGB, se puede aplicar un mismo factor en los 3 planos, o seleccionar diferentes factores para cada plano del color que queremos modificar. De esta manera se pueden

trabajar los tres colores básicos de manera similar, o seleccionar únicamente la capa de color que deseamos modificar.

3.4.2. Restauración digital de imágenes

Como su nombre lo indica, la restauración es el proceso de reconstruir las partes perdidas o deterioradas de las imágenes o videos. Para el caso de las imágenes, esta labor puede ser realizada manualmente en forma intuitiva por un artista experto. Sin embargo, en el mundo de las imágenes y videos digitales este concepto ha cambiado, pues se han generado algoritmos de cierta complejidad que permiten hacer lo mismo automáticamente mediante software.

Una manera de restaurar imágenes es mediante la técnica de *inpainting* (repintado) que es el término usado en inglés para referirse a este proceso que actualmente constituye un campo muy activo de investigación.

La restauración de imágenes es una tarea clásica en el procesamiento de imágenes. Existe la posibilidad de abordar el problema desde diferentes ángulos, por ejemplo, la detección de bordes, la propagación de texturas y la utilización de ecuaciones diferenciales en algunas situaciones particulares.

La restauración de imágenes es sin duda un campo de especial interés tanto desde el punto de vista matemático como comercial. En las industrias del cine y la publicidad es frecuente restaurar imágenes que por el paso del tiempo y/o falta de cuidado, se han deteriorado. Dichas imágenes pueden tener un valor histórico. Algo similar pasa en el campo del arte, en el cual es frecuente la necesidad de restaurar determinadas obras artísticas. En este sentido existe una discusión abierta entre quienes consideran que una obra de valor artístico se debe restaurar y quienes consideran que solo se debe conservar en su estado actual.

3.4.3. Reconocimiento de patrones (clasificación)

La clasificación digital comprende el proceso de conversión de una imagen sin clasificación, en alguna otra categorizada temáticamente, a partir de la agrupación de niveles digitales espectralmente similares. Existen dos métodos de clasificación: supervisado y no supervisado.

La clasificación no supervisada contempla los procedimientos de agrupación de los píxeles de una imagen según su similitud espectral, sin conocimiento previo del contexto temático. En este método no se requiere conocimiento de los tipos de cobertura a priori. El proceso se basa en la elección de las bandas espectrales de la imagen a clasificar, definición del número de clases espectrales, selección de los criterios de similitud y de los algoritmos de agrupación.

La clasificación supervisada requiere de cierto conocimiento previo acerca de la información de la imagen, a través de una combinación de trabajo de campo, análisis de fotografías, informes técnicos, además de referencias profesionales. Con base en este conocimiento, se define y se delimita la clasificación. Las características son utilizadas para "entrenar" un algoritmo de clasificación, el cual calcula los parámetros estadísticos de cada imagen para luego evaluarse,

compararse y asignarse a una determinada clase. La clasificación supervisada pretende definir las imágenes temáticas que no tengan claro significado, considerada por esto como un método artificial.

3.4.4. Morfología matemática

La palabra morfología denota una rama de la biología que se ocupa de la forma y estructura de los animales y plantas. La morfología matemática es una herramienta que se utiliza para extraer componentes de una imagen, los que son útiles en la representación y descripción de regiones de interés como los bordes y los esqueletos. La meta es obtener una imagen resultante a partir de una imagen fuente; esto se logra mediante la manipulación aritmética de un conjunto de píxeles, conocido como elemento de estructura con cada posición de la imagen fuente. Para cada posición se plantea una cuestión relativa a la unión, la intersección o la inclusión del elemento de estructura en la imagen fuente. La respuesta solo puede ser positiva o negativa. El conjunto de los puntos pertenecientes a las respuestas positivas forman un nuevo conjunto que constituye la imagen resultante. La morfología matemática se puede aplicar a imágenes binarias o en niveles de gris. Algunas de las técnicas morfológicas básicas como la dilatación, erosión, apertura, clausura y apertura por área se describen a continuación.

Erosión.- La erosión de un conjunto X por un elemento de estructura B se denota como $\varepsilon_B(X)$ y se define como los puntos de B , tales que B está en X cuando su origen se localiza en x :

$$\varepsilon_B(X) = \{x \mid B_x \subseteq X\}$$

La ecuación anterior también se puede definir en términos de una intersección de translaciones de un conjunto determinadas por el elemento de estructura SE :

$$\varepsilon_B(X) = \bigcap_{b \in B} X_{-b}$$

Más adelante se muestra un ejemplo del resultado de esta operación.

Dilatación.- La dilatación es el operador dual de la erosión; se basa en la pregunta ¿El origen del elemento de estructura toca al conjunto analizado? El conjunto resultante son los puntos en donde la respuesta es afirmativa

La dilatación de un conjunto X , por un elemento de estructura B se denota como $\delta_B(X)$ y se define como los puntos B , tales que el origen de B toca los puntos de X

$$\delta_B(X) = \{x \mid B_x \cap X \neq \emptyset\}$$

La ecuación anterior también se puede definir en términos de la unión de translaciones de un conjunto determinadas por el elemento de estructura SE :

$$\delta_B(X) = \bigcup_{b \in B} X_{-b}$$

En la figura 3-3 se muestran los resultados de la erosión y la dilatación de una imagen binaria dada.

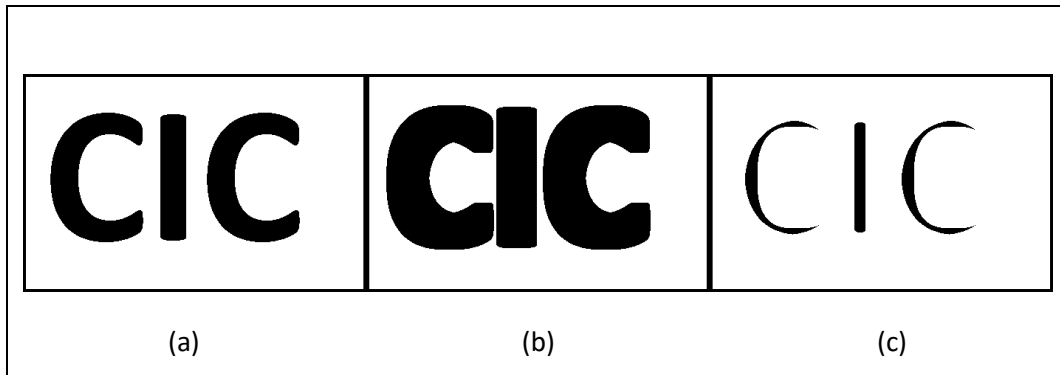


Figura 3-3. Dilatación y erosión. En (a) se muestra la imagen original. En (b) se muestra el resultado de la dilatación de (a). En (c) se muestra el resultado de la erosión de (a).

Apertura.- La apertura de A por el elemento de estructura B se obtiene por la erosión de A por B, seguida por la dilatación de la imagen resultante por el mismo elemento de estructura B, esto es:

$$A \circ B = (A \ominus B) \oplus B$$

Clausura.- La clausura de A por el elemento de estructura B se obtiene por la dilatación de A por B, seguida por la erosión de la imagen resultante por el mismo elemento de estructura B, esto es:

$$A \bullet B = (A \oplus B) \ominus B$$

Apertura por área.- Es la unión de todos los objetos 4 u 8 conectados que cumplen con un mínimo del área λ en píxeles.

$$r = x \in X \mid Area(Cx(X)) \geq \lambda$$

Donde: r Es el resultado de la apertura por área.

Cx Es el tipo de conectividad entre píxeles, normalmente 4 u 8 conectados.

X Son los objetos con conectividad Cx .

x Son los píxeles de los objetos X .

λ Es el umbral.

Estas herramientas morfológicas son muy útiles para la eliminación de artefactos, rellenado de áreas, definición de bordes, además de otros usos. Para una explicación más amplia de la morfología matemática se le refiere al lector al libro *Morphological Image Analysis: Principles and Applications* de Pierre Soille en su segunda edición [Soille, 2013].

3.5. Procesamiento en el dominio de la frecuencia

Para poder procesar una imagen en el dominio de la frecuencia es necesario hacer una transformación del dominio espacial al dominio de la frecuencia. Básicamente existen tres métodos para realizar esta transformación, entre otros.

- La transformada discreta del coseno
- La transformada de Fourier
- La transformada de ondeleta.

Estas herramientas tienen usos variados, por ejemplo, realizar mejoramiento de imágenes aplicando filtros de frecuencias, codificación de información, compresión, análisis de señales y otras.

En general el procesamiento digital de imágenes en el dominio de la frecuencia puede tener aplicación en cualquier ámbito de la ciencia. Es una herramienta muy útil que se desarrolla y aplica cada vez más en todo el mundo, por ejemplo actualmente la compresión de video es una herramienta que ha contribuido en gran parte, al gran crecimiento de la transmisión de secuencias de imágenes por Internet; esta compresión de las imágenes se basa principalmente en la reducción del tamaño de las imágenes en el dominio de la frecuencia; así mismo, varios de los formatos gráficos actuales para archivos de imágenes estáticas, aprovechan este tipo de compresiones en el dominio de la frecuencia, para optimizar el uso del espacio de almacenamiento en las computadoras.

3.6. Segmentación del movimiento

Es importante aclarar que aunque están muy relacionadas, la segmentación de imágenes y la segmentación del movimiento, son dos temas diferentes.

El tema de la segmentación de imágenes es fundamental para el análisis del movimiento en imágenes sucesivas. La segmentación de imágenes es un tema que ha sido abordado desde diferentes ángulos con diversas técnicas y métodos desde hace más de 50 años, según nos dice Yu-Jin Zhang [Yu-Jin Zhang, 2006] en su tratado de Segmentación del movimiento en los últimos 40 años.

Para realizar la segmentación del movimiento en imágenes sucesivas, es necesario analizar cada imagen de una secuencia de imágenes y detectar los objetos que están cambiando de posición y/o tamaño; además, se debe tomar en cuenta que estos objetos pueden aparecer o desaparecer de la escena.

La segmentación del movimiento en imágenes sucesivas, consiste en la identificación del conjunto de píxeles que semánticamente representan un objeto en una secuencia de imágenes. Dado que la segmentación de imágenes es una herramienta primordial para llevar a cabo la segmentación del movimiento, se dará una breve explicación acerca de este tema.

En la figura 3-4, se muestra una imagen dividida en cinco regiones:

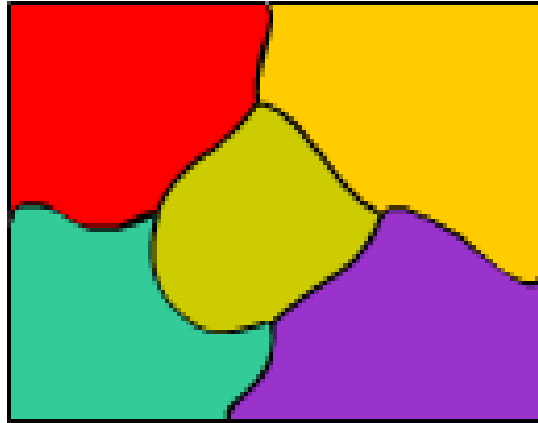


Figura 3-4. Imagen dividida en cinco regiones.

Formalmente, la segmentación de imágenes según Gonzalez y Woods [Gonzalez y Woods, 2008], puede ser definida como la unión de todos los conjuntos de píxeles en que puede ser dividida una imagen, con la restricción de que ningún píxel de la imagen, puede pertenecer a dos regiones o más al mismo tiempo, esto es:

$$X = \bigcup_{i=1}^N R_i ,$$

$$R_i \cap R_j = \emptyset \text{ for } i \neq j$$

Donde X Es la imagen completa.
 R_i y R_j Son regiones de la imagen.
 N Es el número de regiones de la imagen.

Es necesario entender a la segmentación de una imagen como la descomposición de esta imagen en sus partes constituyentes, las cuales están compuestas por los objetos de interés además del fondo.

Más adelante se explican algunas de las técnicas más utilizadas para la segmentación del movimiento, así como algunos de los métodos que se utilizan.

3.6.1. Antecedentes de la segmentación del movimiento

El primer desarrollo de las técnicas para la segmentación de imágenes se remonta a unos 50 años atrás. En 1965 Lawrence Roberts desarrolló el Operador de Roberts [Roberts, 1965], utilizado para detectar bordes (también se le conoció como el detector de bordes Roberts). El detector de Roberts fue el primer paso que se dio para la segmentación de imágenes en sus partes constituyentes.

Después de esto, se han desarrollado una gran cantidad de técnicas, algoritmos y métodos para la segmentación de las imágenes digitales.

Las técnicas que se siguen desarrollando son muy parecidas a las que ya existen, o son la mezcla de algunas técnicas ya desarrolladas; ahora bien, se ha buscado mejorar la segmentación de imágenes bidimensionales y tridimensionales. Con innumerables fines, la investigación y el desarrollo de este tema continúan fervientemente en todo el mundo.

A continuación se explican algunas técnicas comunes para lograr la segmentación del movimiento, con el fin de familiarizar al lector con este tema.

3.6.2. Segmentación del movimiento basada en discontinuidades de flujo óptico

Este método de segmentación está basado en el flujo óptico (*optic flow* en inglés). Consiste en la detección de píxeles en una imagen, para después realizar el reconocimiento y mapeo de estos píxeles en la siguiente imagen; los píxeles se agrupan según su desplazamiento, y de esta manera se crean los objetos que son detectados como objetos en movimiento. Es un algoritmo muy activo que requiere gran cantidad de procesamiento de datos.

Se utilizan medidas locales y se calculan velocidades de acuerdo a estas medidas. Se asume que todas las partes de un objeto tienen la misma velocidad cuando el objeto se encuentra en movimiento. Principalmente se toma en cuenta el movimiento de los bordes. Potter [Potter, 1975] reflexionó que dado que las imágenes sucesivas con las que trabajó tienen muy poca diferencia en sus tiempos de captura, sería fácil correlacionar los bordes de los objetos. En esta técnica cada píxel o punto de la imagen, se tiene que clasificar en uno de los tres grupos siguientes: sombra, objeto o fondo.

En el caso de que dos objetos cercanos estén en movimiento, al realizar el análisis del desplazamiento de los bordes se detectan dos diferentes velocidades en los grupos de píxeles, ya sea una mayor que la otra o una con diferencia de desplazamiento negativo, es decir, una discontinuidad en el flujo de los grupos de píxeles; de aquí es de donde este algoritmo toma su nombre.

Con este método es común obtener sobresegmentación de los objetos bajo ciertas circunstancias, como cuando no se tiene un movimiento rígido de los objetos. En estos casos es recomendable mezclar la información espacial con la temporal. Al utilizar conjuntamente la información espacio-temporal, se vuelve más activa y compleja la ejecución del algoritmo; también se procesa cada punto de la imagen, lo cual genera un sobreprocesamiento de datos, causado por los píxeles que conforman el fondo de la imagen que permanecen fijos, pero en cada iteración son procesados nuevamente.

3.6.3. Segmentación del movimiento basada en la detección de cambios

Este algoritmo trata de tomar ventaja de que es más económico y eficiente detectar los cambios en cada imagen y solo procesar esos puntos o áreas de cambio, lo cual también nos previene de

posibles errores futuros al no procesar datos que no deberían intervenir. La segmentación del movimiento debería ser un proceso multietapa o iterativo para obtener mejores resultados. También se toma ventaja de que el fondo es estático, por lo que podemos quitarlo para no procesarlo en cada imagen, además de que se elimina el cálculo de la diferencia de gradientes. Para detectar la diferencia de si un pixel pertenece a un objeto, al fondo, es brillo etc., se especifica un umbral de diferencia entre pixeles.

Sea la máscara $C(x)$ para indicar que hubo cambio entre dos imágenes:

Si $C(x) = 1$ indica que hubo cambios significativos.

Si $C(x) = 0$ indica que no hubo cambios significativos.

Este algoritmo y su efectividad depende de dos situaciones: la primera es el umbral y la segunda es la eliminación de pequeñas regiones que pudieran ser detectadas como falsos positivos.

Se han realizado estudios sobre esta base para crear el algoritmo que se denominó de imagen de diferencia acumulativa o ADP (por sus siglas en inglés *Accumulative Difference Picture*, Imagen de diferencia acumulativa). En la técnica ADP se lleva un registro $ADPk(x,y)$ de cada pixel, el que inicia en 0 y se va incrementando en 1 cada vez que haya cambios significativos, o se mantiene en el último valor si no hubo cambios en el umbral. La segmentación se lleva a cabo encontrando los valores de cambio más altos en el registro, los cuales corresponderían a los objetos que están realmente en movimiento.

En el algoritmo ADP podrían presentarse varios inconvenientes, por ejemplo, en la reconstrucción de las imágenes no se especifica la imagen de fondo, solo los objetos en movimiento; además, solo se puede aplicar bajo ciertas circunstancias (como la de utilizar un sensor fijo); aun así, al satisfacer algunas condiciones, puede constituir una manera muy conveniente de aislar los objetos en movimiento en una secuencia de imágenes. También se puede aplicar la detección de similitudes en lugar de la detección de cambios y enfocar de una manera contrapuesta el algoritmo dependiendo de las condiciones del proyecto, utilizando paneles o mosaicos de cada imagen.

3.6.4. Segmentación del movimiento basada en la estructura del movimiento (SFM)

La segmentación del movimiento a partir de la estructura, cuya sigla es SFM (del inglés *Structure From Motion* – Estructura a partir del movimiento), se aplica en la segmentación del movimiento en tres dimensiones según se explica en el documento *Segmentation of moving objects in Image Sequence* [Dengsheng, 2001]. Está basada en la geometría espacial tridimensional, la cual es la manera primitiva de cómo funciona el sistema de visión del ser humano. Se obtienen dos imágenes bidimensionales y a partir de éstas se obtiene una perspectiva espacial del entorno.

La tarea de obtener estructuras tridimensionales a partir de dos imágenes bidimensionales es una tarea difícil. Se da por hecho que los objetos se mueven de manera rígida porque se está trabajando en tres dimensiones, o equivalentemente, la cámara es la que se mueve, se toman más

en cuenta los objetos cercanos que los lejanos y a cada punto de una imagen le corresponde un punto en la otra imagen; estas asunciones no siempre se cumplen. Por estas y muchas otras cuestiones la geometría espacial común no se puede aplicar de manera correcta.

Hace algunos años, se desarrolló un esquema de segmentación del movimiento 3D, que utiliza la línea epipolar de la matriz fundamental, que se genera a partir de los datos de la generación tridimensional. En ambas imágenes, de acuerdo a la geometría epipolar, un punto p_1 se asocia con su punto correspondiente p_2 en la otra imagen mediante una matriz fundamental F , la cual encapsula toda la información acerca del movimiento de las cámaras, así como la información de traslación; esta matriz es de la forma:

$$p_1^t F p_2^t = 0$$

El punto correspondiente p_2^t solo puede estar en la línea epipolar p_1^t .

En la figura 3-5, se muestra un ejemplo de la geometría epipolar. En la imagen podemos observar cómo a partir de la visión estereoscópica, se forman una línea epipolar y un plano epipolar.

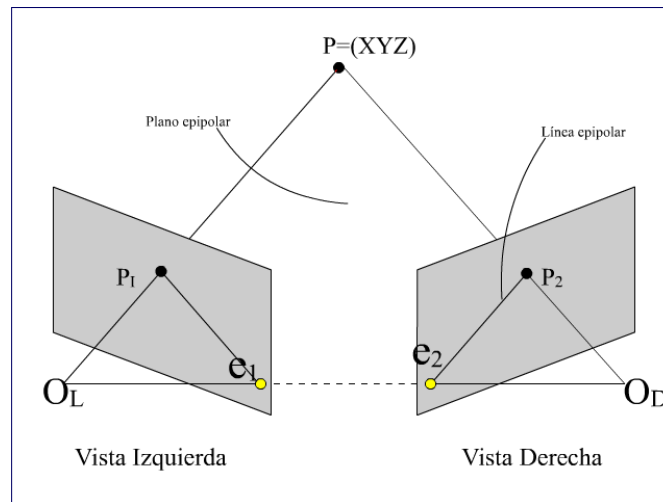


Figura 3-5. Línea y plano epipolar.

Este algoritmo necesita de cuatro pasos: correspondencia de características, generación del modelo, agrupamiento y pruebas de hipótesis múltiples para comprobar la segmentación correcta.

Una de las ventajas que nos ofrece esta técnica, es que se puede alcanzar una reconstrucción de las estructuras en las escenas gracias al tipo de segmentación; una debilidad es que se dan por hechas fuertes asunciones, además de que sólo se puede aplicar en limitadas situaciones. Si la complejidad del algoritmo es lineal, necesita un procesamiento de datos ligero, pero si su complejidad es no lineal, necesita un monto grande de procesamiento de datos.

3.7.Conclusiones del capítulo

Hasta aquí se han explicado brevemente algunas técnicas y algoritmos para la segmentación del movimiento. La intención fue aclarar un poco el panorama que envuelve a este tema, así como mostrar algunos de los problemas a los que nos tenemos que enfrentar. Se han desarrollado muchas técnicas, por lo que aportar nuevas ideas es difícil, pero no imposible, por ejemplo en 2001, Raanan Fattal [Raanan et al., 2001] introdujo el procesamiento de imágenes de dominio del gradiente, una nueva manera para procesar imágenes manipulando el gradiente entre los píxeles en lugar de utilizar los propios valores del píxel, lo cual nos da un enfoque un poco diferente para conseguir nuestro objetivo.

Los ejemplos anteriores son solo unos pocos algoritmos desarrollados para alcanzar la meta de segmentar el movimiento en tiempo real en imágenes sucesivas. En dependencia del objetivo a alcanzar, se puede optar por combinar algunas de las técnicas ya existentes y asignar valores estratégicos, o implementar alguno de los métodos ya existentes y seleccionar algunos parámetros personalizados para mejorar los resultados.

Capítulo 4. Solución del problema

4.1.Introducción

Actualmente no existe un programa que de manera general segmente el movimiento en secuencias de imágenes, por lo que se hace necesario desarrollar aplicaciones específicas, para cada tarea que tenga esta finalidad.

El trabajo de tesis que se presenta se muestra con aplicaciones de videovigilancia, por lo que las pruebas y experimentos que se describen se muestran principalmente con resultados obtenidos de videos de este tipo. Una importante restricción en este algoritmo es que las imágenes que van a ser comparadas, tienen que ser capturadas con una cámara estática, debido a que este algoritmo, se basa principalmente en la diferencia de imágenes (FD).

La mayoría de los sistemas de videovigilancia utiliza baja resolución en las imágenes, e incluso hay muchos que solo procesan una capa de color. El algoritmo desarrollado funciona en tiempo real con secuencias de imágenes en las 3 capas de color o solo en una, y funciona con una resolución espacial de 640x480 pixeles, lo cual da un amplio margen de aplicación con los sistemas actuales de videovigilancia.

También se destaca que el algoritmo desarrollado puede tener otras aplicaciones además de la videovigilancia; por ejemplo, se puede utilizar en puntos de monitoreo para obtener estadísticas de las horas pico de circulación de peatones o automóviles en vías públicas; en paradas de autobús o estaciones del metro, puentes peatonales o de vehículos etc.; también puede ser implementado para detectar migración de aves o simplemente como detección de movimiento.

4.2.Plataforma de programación.

El algoritmo fue implementado en “*Matlab R2012a*”, en Windows 10; por ello, el programa se puede exportar fácilmente a cualquier plataforma que trabaje sobre Unix, Windows, Mac OS X o GNU/Linux que soporte la versión 7.14 de Matlab.

Tomando en cuenta la velocidad actual de los procesadores y los estándares de los sistemas de videovigilancia que se utilizan hoy en día, este algoritmo es de fácil implementación y funciona con cualquier computadora personal que tenga Windows Vista o superior, y Linux kernel 2.4 o superior, además de las versiones similares de Unix.

Además de que el programa desarrollado tiene como restricción obtener las secuencias de imágenes mediante una videocámara estática, para obtener un mejor funcionamiento es importante tomar en cuenta que el algoritmo se ve seriamente afectado por los cambios repentinos en la iluminación, ya sea en toda la imagen o por zonas. Los siguientes son algunos de los aspectos que debemos tomar en cuenta:

- La luz generada por el sol genera sombras, tiene oclusiones a causa de las nubes y otros fenómenos ambientales, tiene variaciones de luminosidad, etc.
- Al comparar la misma escena con iluminación artificial y sin ésta, se detectan cambios muy significativos.

- En lugares abiertos no se debe utilizar la misma imagen de fondo a toda hora, a causa de los cambios en la iluminación natural.
- Movimientos accidentales de la cámara también generan cambios muy significativos.

Además de estas circunstancias, existen otras que dificultan el óptimo funcionamiento de este algoritmo. Sin embargo, el algoritmo cumple con los objetivos que nos hemos propuesto.

4.3.Descripción general del algoritmo

El algoritmo fue desarrollado basándonos en el objetivo principal de desarrollar un método que detecte en tiempo real objetos en movimiento dentro de una secuencia de imágenes, esto es, dada una imagen de fondo, el algoritmo detecta en tiempo real los objetos que no se encontraban presentes en dicha escena, para posteriormente generar un reporte acerca de estos objetos detectados. En la figura 4.1 se muestra un diagrama de bloques del algoritmo desarrollado.

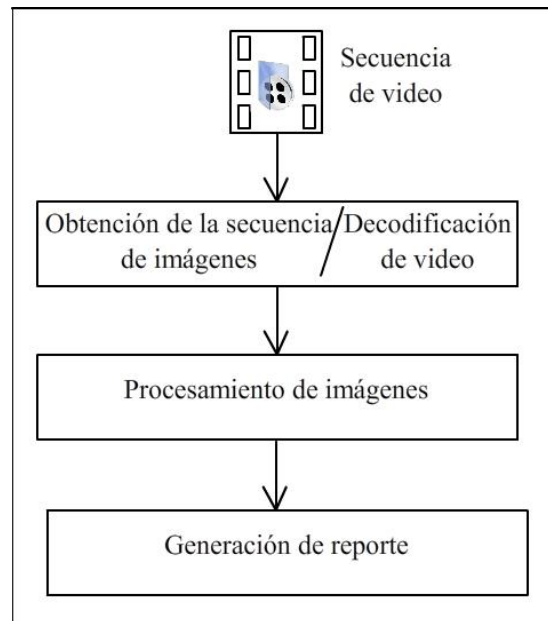


Figura 4-1. Diagrama a bloques del proceso general de detección de movimiento

Cada una de las tres partes que se programaron para cumplir con los objetivos, fue desarrollada de manera independiente de las otras, tan solo recibiendo la información generada del paso anterior; esto es, la obtención de la secuencia de imágenes o decodificación del video, toma como entrada la secuencia de video, y entrega al siguiente proceso la secuencia de imágenes; el procesamiento de las imágenes genera una secuencia de imágenes de salida con los objetos detectados, y también genera un archivo de texto con la información acerca de los objetos detectados; finalmente, el proceso de generación del reporte toma el archivo de texto generado y realiza un análisis de los datos para generar un reporte de cada uno de los objetos detectados. A continuación se describe cada uno de estos procesos.

4.4. Obtención de la secuencia de imágenes / Decodificación de video

Para el procesamiento de las imágenes se puede proceder de diferentes maneras; se da por hecho que las imágenes son secuenciales, que tienen un formato estandarizado similar, y que son compatibles con las computadoras personales actuales.

Las imágenes se pueden procesar directamente en formato de mapa de bits desde el disco duro de la computadora, o desde algún otro medio de almacenamiento masivo; también se pueden leer y procesar las imágenes que tengan algún formato con compresión (JPG, PNG, TIFF u otro formato estándar), o se pueden obtener y analizar directamente desde una cámara conectada a la computadora, o desde una secuencia de video que se obtenga de Internet.

El algoritmo tiene un codificador y un decodificador de video, implementados con las herramientas incorporadas en Matlab; por lo tanto, si se va a procesar una secuencia de archivos de imagen, se puede convertir la secuencia de imágenes a un archivo de video y posteriormente analizar el archivo de video codificado. El tema de decodificación de video fue por mucho el tema más complicado y el que llevó más tiempo de investigación, a pesar de no ser la parte central de este trabajo; se analizaron temas como la transformada discreta del coseno, correlación de la información, flujo serial de datos, cuantificación de información, compresión de la información etc.; no hubo aportaciones al estado del arte en este campo, tan solo análisis de métodos de compresión y descompresión de video, además de la implementación de algunas técnicas de este campo.

Las secuencias de video también se pueden leer desde un archivo AVI, MOV, FLV, MPEG o algún otro formato de compresión de video estándar. Matlab acepta una gran variedad de formatos de video para su decodificación y proceso. En la figura 4-2 se muestra un diagrama que explica visualmente el proceso de obtención de las imágenes

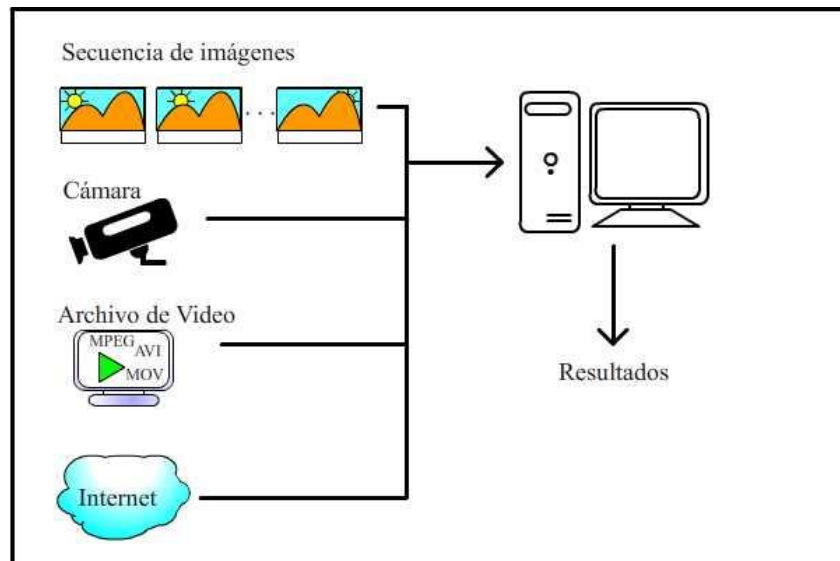


Figura 4-2. Lectura de archivos.

4.5.Comparación de las imágenes

La segunda parte de este trabajo de tesis es la parte medular de la investigación. Consistió en el análisis de la secuencia de imágenes para detectar los objetos en movimiento. La primera parte de esta sección del algoritmo, consiste en determinar la imagen de fondo contra la que se comparará la secuencia de imágenes; después se realiza la sustracción de fondo y umbralado en cada imagen, con el fin de detectar los conjuntos de píxeles que pertenecen a objetos en movimiento, y cuyo resultado son tres imágenes binarias; inmediatamente después, mediante el proceso morfológico de “apertura por área”, se elimina el ruido o artefactos que pudieran aparecer en las imágenes resultantes; después, para correlacionar los píxeles que pertenecen al mismo objeto, se realiza un etiquetado de ellos; con esta información en las imágenes binarias se calcula el área, el centroide y la posición de cada objeto detectado; estos datos se envían a un archivo de texto para realizar la tercera parte del proceso. En la figura 4-3 se muestra un diagrama general de la segunda parte de este trabajo.

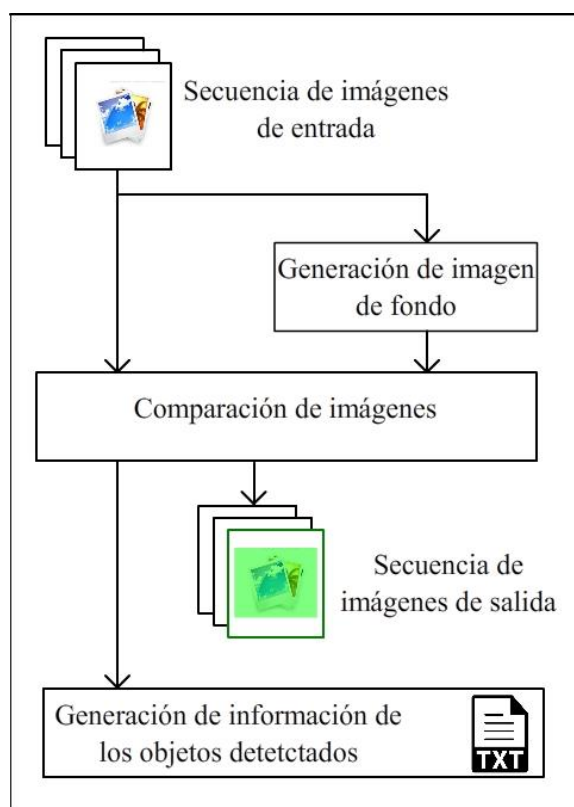


Figura 4-3. Secuencia de pasos en el procesamiento de imágenes.

En la comparación de imágenes, el primer paso consiste en realizar una resta de imágenes en cada una de las tres capas de la imagen de fondo y en las tres capas de la imagen actual de la secuencia de imágenes; dependiendo del resultado con cada píxel, se aplica un umbral para decidir si ese píxel pertenece al fondo o a un objeto en movimiento, tal como se explicó en la sección 2.2.2, dando como resultado 3 imágenes binarias.

El siguiente paso es remover el ruido o los artefactos que pudieran aparecer en cada una de las imágenes binarias; en la morfología matemática este proceso es conocido como “apertura por área”; es un método algebraico que consiste en detectar y contar los pixeles cuatro u ocho conectados, si la suma de pixeles de estos conjuntos es menor que el área establecida, ese conjunto de pixeles es eliminado de la matriz que conforma la imagen binaria. Si es mayor, el objeto se mantiene dentro de la imagen.

A continuación se revisa cada uno de los procesos de la segunda parte del algoritmo.

4.5.1. Generación de la imagen de fondo

La imagen de fondo puede obtenerse de varias maneras, la imagen de fondo juega un papel importante en el algoritmo por lo que su generación es crucial.

Si se planea analizar videos de lugares donde la iluminación es constante o tiene muy poca variación, por ejemplo, en interiores donde no interviene de manera notoria la luz solar, se recomienda utilizar una imagen que haya sido capturada sin ningún agente (u objeto) extraño, es decir, solo la imagen de fondo misma; en caso de que no se pueda capturar esta imagen de manera directa, la imagen se puede componer, mediante la unión de los pixeles que conforman el fondo. En la figura 4-4 se muestra un ejemplo de la composición del fondo.

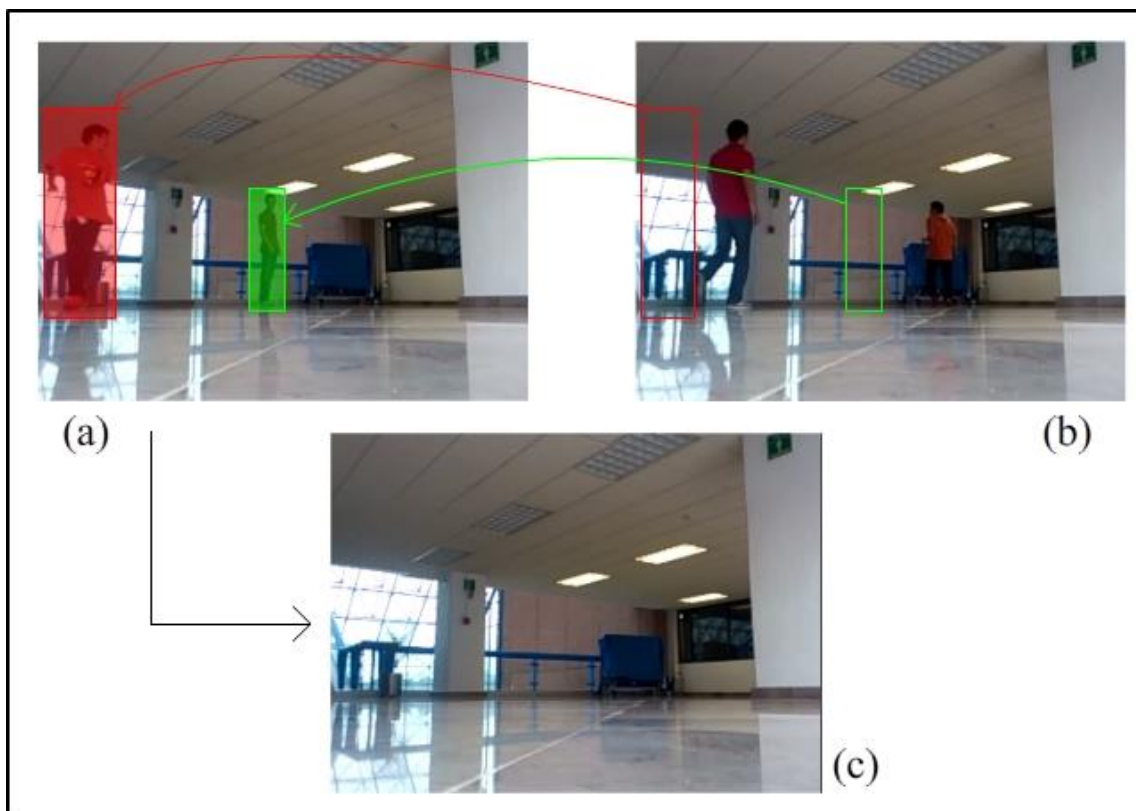


Figura 4-4. Composición de la imagen de fondo. En la imagen a) se han reemplazado los pixeles en los rectángulos rojo y verde por los pixeles de la imagen b) que están en la misma posición, resultando la imagen c) que será utilizada como imagen de fondo.

Si se va a analizar un video grabado en exteriores o donde la luz natural afecta de manera notoria, se recomienda realizar un promediado de la secuencia de imágenes, es decir, se realiza un promedio escalado pixel a pixel de todas las imágenes; el número de imágenes que se utilizará para realizar este promediado dependerá de los resultados que se quieran obtener. De esta forma, cuando un objeto en movimiento llega a la escena analizada, es detectado, pero si permanece inmóvil durante el tiempo que dura el promediado, pasa a formar parte del fondo y ya no se detecta como objeto en movimiento. En la figura 4-5 se muestra un ejemplo del promediado de imágenes:

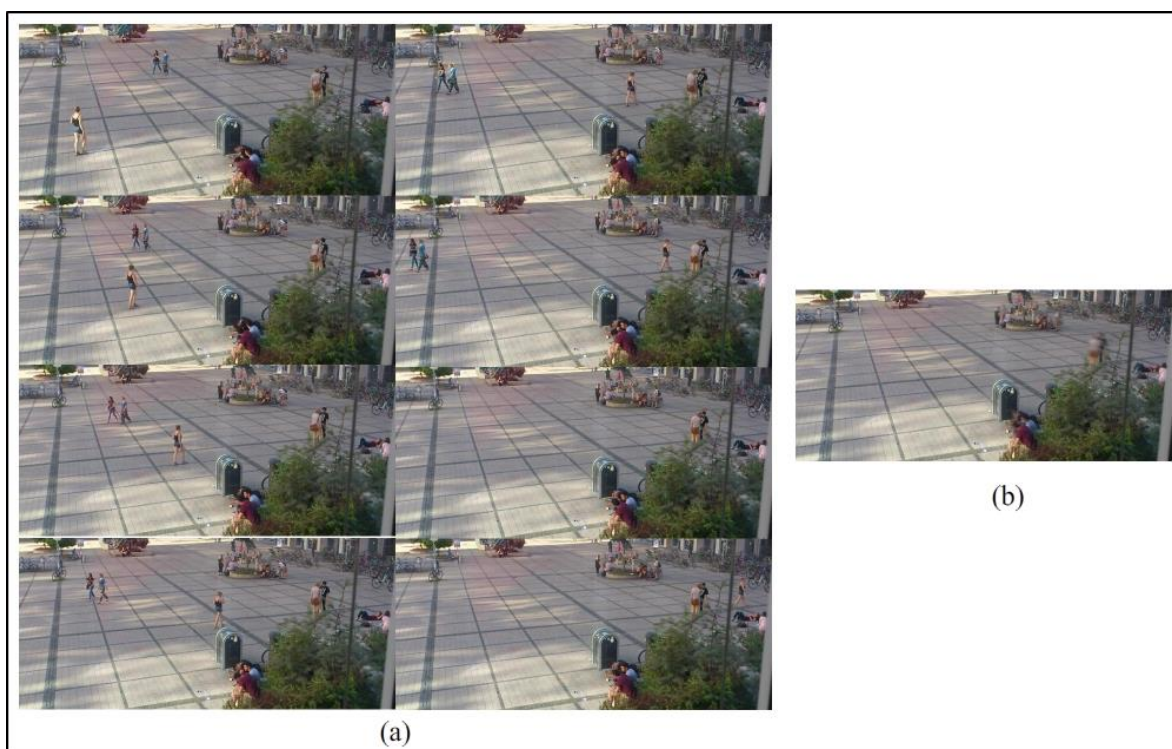


Figura 4-5. Promediado de imágenes. En a) se muestra una secuencia de imágenes, (b) es el promediado de esas imágenes, es el mapa de bits que se utiliza como imagen de fondo.

4.5.2. Resta y umbralado de las imágenes

Este procedimiento que consiste en restar y umbralar aparentemente en un solo paso, es frecuentemente utilizado, realiza la resta de dos imágenes pixel a pixel en cada capa de color, y se toma el valor absoluto, esto nos da como resultado una matriz únicamente con valores positivos en cada capa de color; después, también en cada capa de color se analiza si el resultado es mayor o igual que un umbral predefinido, en dado caso, se le asigna un valor igual a uno a ese pixel, de otra forma se le asigna un valor igual a cero; de esta manera, para cada capa de color, los pixeles pertenecientes al fondo de la imagen tendrán solamente valores iguales a cero, mientras que los pixeles pertenecientes a objetos en movimiento tendrán un valor igual a uno.

La resta de dos imágenes que nos da como resultado tres matrices solo con valores positivos, se realiza mediante la siguiente ecuación:

$$R_c(x, y) = \text{abs} \left(\sum_{x=1}^{\text{ancho}} \sum_{y=1}^{\text{alto}} \text{Img1}_c(x, y) - \text{Img2}_c(x, y) \right)$$

Donde: $R(x, y)$ Es la matriz resultante de la capa roja verde o azul.

Img1_c Misma capa de la imagen resultante en la Imagen 1.

Img2_c Misma capa de la imagen resultante en la imagen 2.

Para obtener la imagen binaria de cada capa de color, se aplica un umbral en cada una de las matrices resultantes $R_c(x, y)$, de la siguiente manera:

$$U_c(x, y) = \begin{cases} 1, & R_c(x, y) \geq u \\ 0, & \text{en otro caso} \end{cases}$$

Donde: $U_c(x, y)$ Es la matriz umbralada resultante de la capa roja, verde o azul de la matriz R.

El resultado obtenido son tres matrices binarias bidimensionales; cada matriz contendrá la información binaria resultante de cada capa de color. Se ha elegido esta manera de proceder por conveniencia para el siguiente paso que es la supresión de los artefactos.

4.5.3. Supresión de los artefactos por capa de color

La supresión de los artefactos consiste en eliminar los pixeles de los objetos que no aportan información relevante, como se muestra en la figura 4-6. Este paso es de gran importancia, pues uno de los puntos débiles en FD, es que se generan muchos artefactos, y por lo tanto los resultados pueden ser muy desalentadores.

La supresión de artefactos es mucho más veloz en imágenes binarias, que en imágenes en colores que están compuestas por imágenes en niveles de gris en cada capa. Esto es así porque al realizar el escaneo y el procesamiento de los datos en imágenes binarias, cada pixel de cada capa de color solo puede tener dos valores posibles; en cambio, en las imágenes en colores, en cada capa de color cada pixel puede tener 2^n posibles valores, donde n es la cantidad de bits utilizado para representar cada valor de pixel.

Para este paso se optó por utilizar el procedimiento morfológico de “apertura por área”, con el que se eliminan todos los objetos que no cumplen con un área en pixeles igual o mayor que un umbral predefinido. Es necesario especificar si el análisis se realizará para objetos cuatro u ocho conectados. En la figura 4-6 se ejemplifica el proceso de supresión de artefactos.

En los trabajos revisados en el estado del arte, no se encontró ningún algoritmo que realizara la segmentación del movimiento tomando en cuenta cada una de las capas de color. Al realizarlo de esta manera, se tiene un método nuevo para la segmentación del movimiento. Es intuitivo pensar que al analizar cada capa de color, se requiere de un mayor tiempo en procesamiento, pero al realizarlo en imágenes binarias, el costo computacional es bajo, lo que hace posible implementarlo en tiempo real en el algoritmo propuesto.

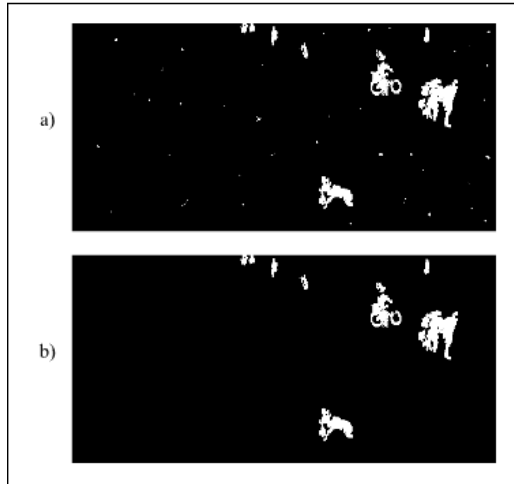


Figura 4-6. Supresión de los artefactos. En a) se muestra una imagen con artefactos. En b) se muestra la misma imagen después de habersele suprimido los artefactos.

4.5.4. Recomposición de la imagen

El resultado del paso anterior nos genera tres imágenes binarias con la representación de los objetos detectados en cada capa de color; para obtener la recomposición de la imagen con los resultados de las tres capas de color, éstas se combinan mediante una operación OR, dándonos como resultado otra imagen binaria. Esto se hace para llevar a cabo ahora el etiquetado de esa imagen resultante. A continuación, se muestra un ejemplo de cómo se combinan las tres imágenes binarias, con el fin de obtener otra imagen con la cual se trabajará posteriormente. La figura 4-7 ilustra el proceso de combinado de las tres imágenes en una sola, mediante la operación OR.

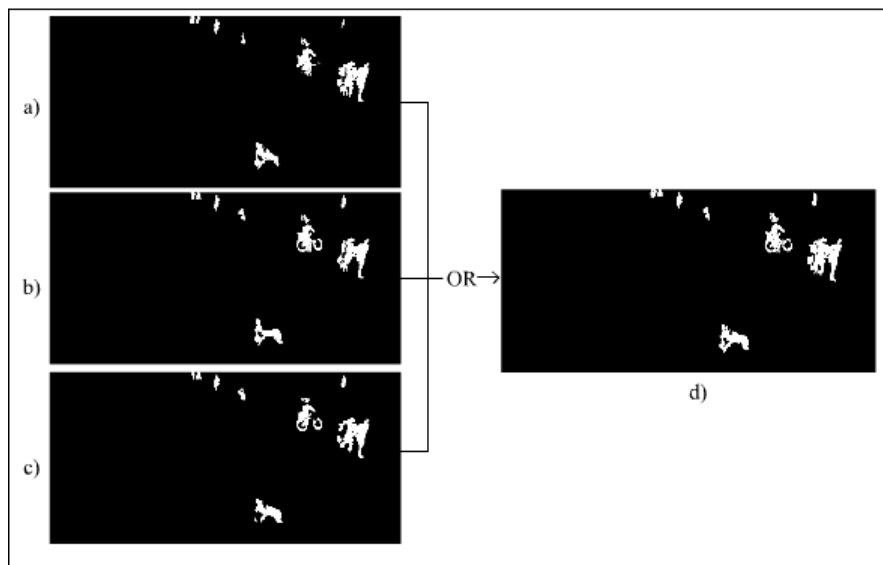


Figura 4-7. Recomposición de la imagen. En las imágenes a, b y c, se muestran las imágenes binarias correspondientes a las capas roja, verde y azul respectivamente, en d) se muestra la imagen resultante de combinar estas tres capas.

4.5.5. Etiquetado

El etiquetado se realiza sobre la imagen binaria obtenida del paso anterior y consiste en asignar un número identificador a cada grupo de píxeles que constituye un objeto; de esta manera se identifican todos los objetos detectados en la imagen.

También conocido como "etiquetado de componentes" además de otras maneras, esta, es una aplicación algorítmica de la teoría de grafos, donde subconjuntos de componentes conectados son marcados con etiquetas únicas. Se basa en una heurística dada el etiquetado de componentes conectados, no se debe confundir con la segmentación misma de las imágenes.

Este etiquetado de componentes, se utiliza en visión por computador para detectar regiones conectadas en imágenes digitales binarias, aunque las imágenes en color y datos de mayor dimensionalidad también pueden ser procesadas, en la figura 4-8, se muestra un ejemplo del etiquetado sobre una imagen.

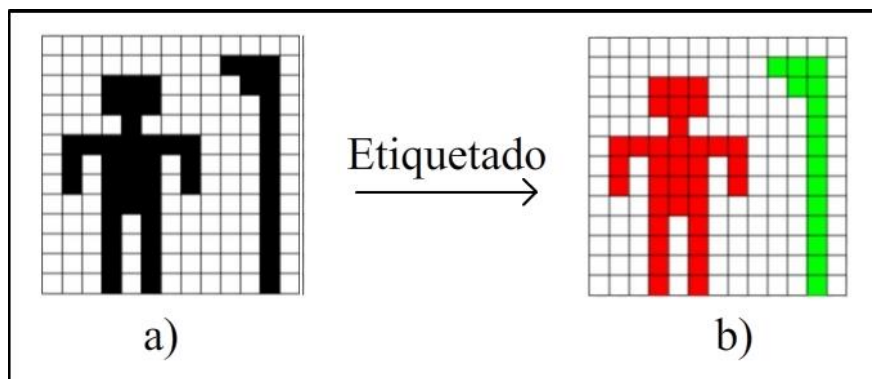


Figura 4-8. Ejemplo de etiquetado. En a) se muestra una imagen binaria con dos objetos, después del proceso de etiquetado, se puede ver la misma imagen pero con los elementos etiquetados utilizando diferentes colores para ver mejor resaltados los dos objetos.

Es importante aclarar que para el cerebro humano, el proceso de etiquetado es muy intuitivo, fácil de comprender, incluso pareciera carecer de importancia, pero en la computadora no lo es tanto, por lo que se tiene que proceder a realizar alguna programación siguiendo los pasos de algún algoritmo para conseguir este fin. Algunos algoritmos como el de "un componente a la vez" o el de "dos pasos" son comúnmente utilizados para el etiquetado de píxeles en imágenes binarias.

4.5.6. Generación de las imágenes de salida

Después de clasificar los píxeles que pertenecen a los objetos en movimiento y los que pertenecen al fondo de la imagen, en sencillo generar las imágenes de salida, estas se muestran en la ventana de salida que se podemos apreciar en la figura 4-6. La generación de estas imágenes, y el proceso de mostrarlas igual que una secuencia de video, nos da una manera muy visual de ver la calidad de los resultados obtenidos.

En los documentos analizados, tan solo se muestran algunas pocas imágenes estáticas, y algún resultado obtenido de estas en muy baja resolución, estos resultados no muestran en toda su amplitud los resultados obtenidos con las diferentes técnicas; Por tanto se complica comparar que tan buenos o malos son los resultados obtenidos.

En la interfaz gráfica, se muestran dos secuencias imágenes; en la primer secuencia se muestran los objetos en movimiento en pixeles rojos, y el fondo en color negro, esta imagen es el resultado obtenido después de realizar la composición de la imagen, y se pueden apreciar perfectamente los objetos en movimiento, además se muestra en color cian el centroide calculado de cada objeto ; Para la segunda secuencia de imágenes, se crea una matriz basándose en la imagen de la secuencia original, de esta primeramente se elimina la capa roja, y después se agregan a la misma capa, los pixeles que fueron clasificados como pertenecientes a objetos en movimiento, lo cual nos da una secuencia de imágenes en la cual se pueden apreciar fácilmente los objetos segmentados en movimiento con un retardo de menos de una décima de segundo, con lo cual se da la impresión de que ¡se detectan al mismo tiempo que se mueven!

4.6.Escritura de datos

El último paso del proceso de comparación de imágenes, es la generación del archivo de texto con la información del archivo de los objetos detectados. Es gracias a este paso que se puede generar un reporte acerca de todos los elementos detectados. Exactamente cuándo se tiene la imagen binaria la cual es resultado de la operación OR entra las tres matrices binarias resultantes de FD y umbralado, se pueden obtener los datos de los elementos detectados. Para cada elemento encontrado en el etiquetado, se realizan las operaciones algebraicas necesarias para encontrar la posición del centroide y el área de cada objeto, además del identificador único con respecto a esa imagen que se le asignó a cada elemento, y el identificador único con respecto a toda la secuencia de imágenes, también se utiliza el número de imagen que está siendo procesada y el número de elementos detectados en esa imagen.

El primer paso es crear un archivo de texto que servirá como almacén de toda la información generada, después, por cada imagen procesada, se genera un conjunto de información que se va almacenando en el archivo de texto, esto se realiza mientras se está procesando el video, una vez terminado el proceso de análisis de las imágenes, el proceso de almacenado también es finalizado.

Por cada elemento detectado, se genera una línea de información que se va almacenando en el archivo de texto, esta línea admite el identificador único del elemento además de la información explicada anteriormente, en la imagen 4-16 se muestra un ejemplo de este archivo generado.

La figura 4-9 nos muestra el proceso de comparación de imágenes explicado en los pasos anteriores.

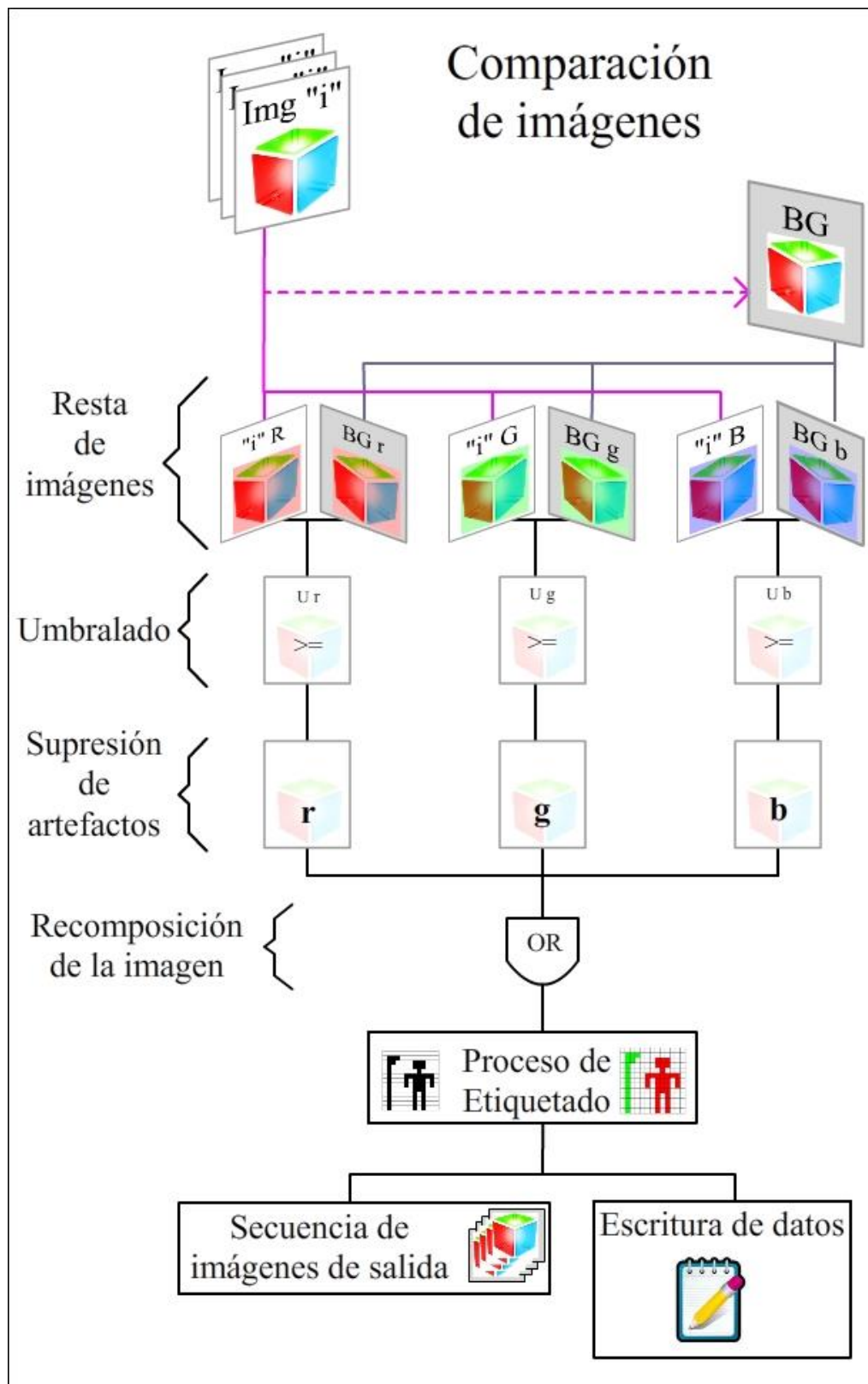


Figura 4-9. Pasos seguidos durante la comparación de imágenes en el algoritmo propuesto.

4.7. Correlación de los objetos detectados

La tercera y última parte se encarga de analizar los resultados generados en el proceso anterior. Todos estos datos han sido guardados en un archivo de texto con el nombre “*objsLog.txt*” para ser analizados de la siguiente manera:

Primero se detectan todos los objetos que pertenecen a la primera imagen, con su posición en “y”, posición en “x” y el área de cada uno, después se realiza la misma operación con la segunda imagen; el siguiente paso consiste en comparar cada uno de los objetos de la primera imagen con cada uno de los objetos de la segunda imagen, para detectar el equivalente en cada imagen; se toma en cuenta la posición anterior y actual, con ello se calcula la distancia euclidiana entre todos los elementos detectados en la primera imagen y cada uno de los elementos que se analizan en la segunda imagen, también se analiza el parecido en el área entre ambos objetos. Para este fin se utilizan dos umbrales, el primero es el umbral de desplazamiento del objeto y se utiliza para decidir si la distancia euclidiana (en píxeles) se considera correcta para relacionar los dos objetos, el segundo es el umbral de cambio en por ciento de área. Ambos umbrales dependerán del video que se está analizando para darles los pesos y parámetros correctos, pues el desplazamiento puede variar dependiendo de lo que se esté moviendo, por ejemplo: una persona caminando, un automóvil, personas en bicicleta, aves etc.; El umbral de por ciento en el cambio del área, puede variar mucho dependiendo de la iluminación y el movimiento aleatorio de los objetos, un ejemplo de esto es: una persona que se analiza de frente con las extremidades extendidas, puede ocupar el doble de píxeles si se compara con la misma persona de perfil en donde unas extremidades quedan ocultas detrás de otras, o si la misma persona está en cuclillas; se analizan otros ejemplos similares a este en el capítulo cinco.

Al realizar la comparación de los objetos de dos imágenes sucesivas, digamos imagen A e imagen B, se pueden generar algunos sucesos que se ejemplifican y explican a continuación, en algunos de los ejemplos solo se consideran dos objetos, pero puede aumentar indefinidamente este número, por ejemplo se muestra la fusión de dos objetos en uno, pero pudieran ser más. Los posibles sucesos son:

Suceso 1.- “x” objeto de la imagen A “se convirtió en” “y” objeto de la imagen B, donde $x \in$ imagen A, e $y \in$ imagen B. En la figura 4-10 se muestra gráficamente este suceso.

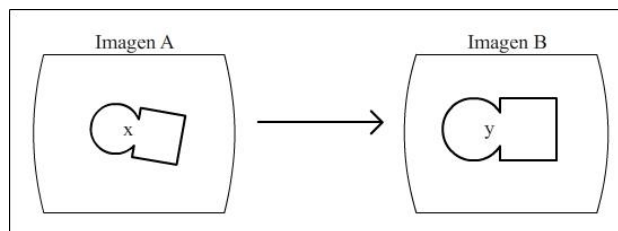


Figura 4-10. Ejemplo del suceso “se convirtió en” de la “imagen A” a la “imagen B”.

Suceso 2.- “x” objeto de la imagen A “Se fusionó con” “y” objeto de la imagen A “y formó” el objeto “z” de la imagen B. donde “x” e “y” \in imagen A, y “z” \in imagen B. En la figura 4-11 se muestra gráficamente este suceso.

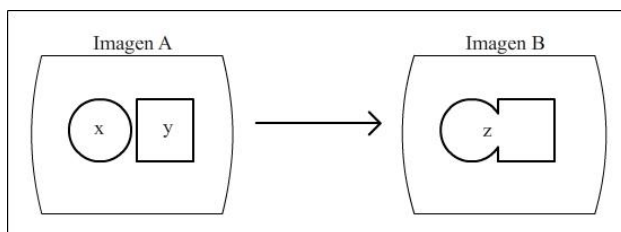


Figura 4-11. Ejemplo del suceso “se fusionó con” de la “imagen A” a la “imagen B”.

Suceso 3.- “x” y “y” objetos de imagen B “son producto de la separación” del objeto “z” de imagen A, donde “x” y “y” \in imagen B e “y” \in imagen A. En la figura 4-12 se muestra gráficamente este suceso.

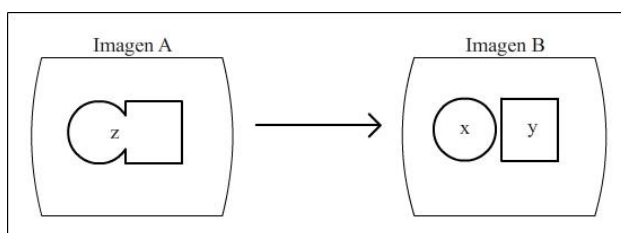


Figura 4-12. Ejemplo del suceso “son producto de la separación” de la “imagen A” a la “imagen B”.

Suceso 4.- “x” objeto de la imagen B “Apareció”. Esto sucede cuando no se encontró un objeto similar o alguna separación con algún objetos similar en la imagen anterior. En la figura 4-13 se muestra gráficamente este suceso.

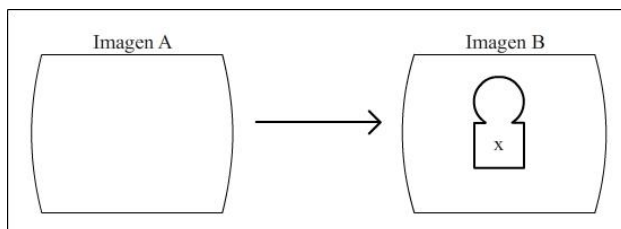


Figura 4-13. Ejemplo del suceso “apareció” de la “imagen A” a la “Imagen B”.

Suceso 5.- “x” objeto en la imagen A “Desapareció”. Esto sucede cuando no se encuentra un objeto similar en la imagen B ni una separación con algún objeto similar. En la figura 4-14 se muestra gráficamente este suceso.

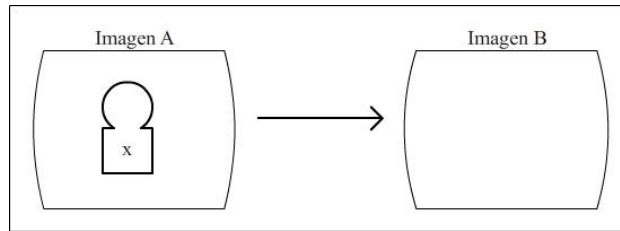


Figura 4-14. Ejemplo del suceso “desapareció” de la “imagen A” a la “imagen B”.

En el análisis de la primera imagen de la secuencia, todos los objetos detectados se marcan como objetos que aparecieron. En ejemplo de este análisis y el archivo de salida que se ha nombrado “*observer.txt*”, lo podemos ver en la figura 4-15.

También en la figura 4-15 se resalta en la imagen la información correlacionada que resulta de ejecutar el análisis del archivo de texto.

```

1 Aparecio
2 Aparecio
3 Aparecio
4 Aparecio
5 Aparecio
6 Aparecio
7 Aparecio
8 Aparecio
9 Aparecio
10 1 Se convirtió en 10
11 2 Se convirtió en 11
12 3 Se convirtió en 12
13 4 Se convirtió en 14
14 7 Se convirtió en 16
15 8 Se convirtió en 15
16 9 Se convirtió en 17
17 4 Se fusionó con 15 y formó 13
18 6 Se fusionó con 14 y formó 13
19 13 Se fusionó con 15
20 10 Se convirtió en 18
21 12 Se convirtió en 19
22 13 Se convirtió en 20
23 13 Se convirtió en 21
24 14 Se convirtió en 23
25 15 Se convirtió en 25
26 16 Se convirtió en 24
27 17 Se convirtió en 26
28 23 Aparecio
29 18 Se convirtió en 27
30 19 Se convirtió en 28
31 20 Se convirtió en 29
  
```

Figura 4-15. Ejemplo del archivo de salida *observer.txt*

Para generar el reporte final que nos arrojan los datos procesados y mostrados en el archivo “observer.txt”, se puede proceder de dos maneras: la primera consiste en que se genere el reporte básico del seguimiento a un objeto en específico, para lo que es necesario llamar a la función “caracterizaX” que recibe como argumento el identificador único del objeto. Algunos resultados se muestran en el capítulo cinco.

La otra forma de proceder para realizar el reporte final, es ejecutar el comando “caracteriza”, a partir del cual se generará el reporte individual de todos y cada uno de los objetos detectados, en toda la secuencia analizada; el archivo de salida se llamará “FinalRep.txt”.

Este programa puede ser modificado para agregarle más funcionalidad, más detalle en la salida de la información, más controles en la ventana de interfaz para hacer que las nuevas funciones sean más amigables, o puede ser modificarlo con algún otro fin, por ejemplo, detectar la cantidad de flujo de gente o vehículos. Una modificación que fue implementada consistió en que el algoritmo detectara movimiento en secuencias de video, y que la salida fuera la lista de cuadros en donde se detectó el movimiento; también se capturan y se guardan únicamente los cuadros donde se detectaron cambios para poder referirnos rápidamente a ellos y así poder visualizar información que pudiera ser importante; al final el algoritmo muestra una gráfica donde se marca en una línea la secuencia de video con los cuadros importantes marcados; esta implementación permite analizar videos de cualquier duración automáticamente, así, en vez de que este tedioso trabajo tuviera que ser realizado por alguna persona, podría ser llevado a cabo por una computadora en un tiempo menor y con una gran cantidad de detalle.

4.8. Interfaz gráfica

La interfaz gráfica consta de 2 ventanas: la primera es la consola de salida y la segunda es la ventana de figuras.

La figura 4-1 nos muestra un ejemplo de la consola de salida; en ella se muestran los datos resultantes del algoritmo en modo texto separados en columnas; cada columna está separada por el carácter barra vertical “|”. Solo la primera línea es diferente a las demás en cuanto a su estructura; en ella se pueden leer las dimensiones del vector de las imágenes procesadas: alto, ancho y número de capas de color respectivamente; todas las imágenes procesadas deben tener las mismas dimensiones. En la figura 4-15 podemos ver cómo a partir de la segunda línea, se detalla la siguiente información:

Columna 1.- Identificador único del suceso detectado: Puede ser un objeto, un conjunto de objetos o alguna anomalía.

Columna 2.- Total de objetos detectados en esa imagen.

Columna 3.- Identificador de la imagen procesada.

Columna 4.- Identificador del objeto en esa imagen.

Columna 5.- Coordenada vertical del centroide.

Columna 6.- Coordenada horizontal del centroide.

Columna 7.- Área en pixeles del suceso detectado.

En la figura 4-16 se muestra un ejemplo de la consola de salida.

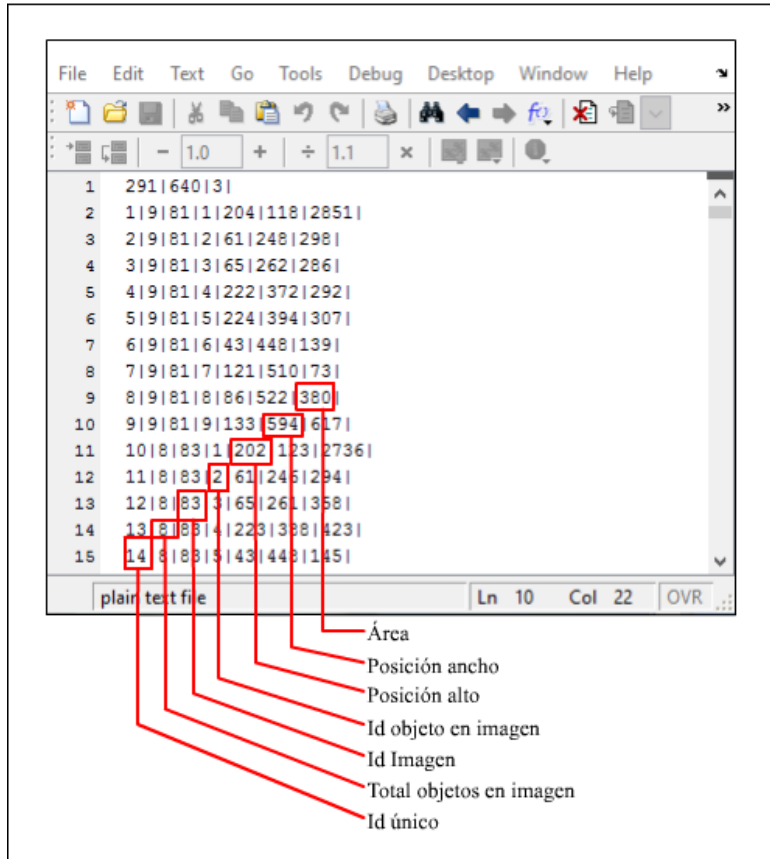


Figura 4-16. Ejemplo de la ventana de salida del algoritmo desarrollado.

La figura 4-17 nos muestra la segunda ventana de la interfaz, la cual está compuesta por una imagen dividida en cuatro cuadrantes, los botones de control, una barra de desplazamiento y un cuadro de texto que nos indica el número de la imagen procesada.

En la imagen 4-16, donde podemos ver la ventana de figuras, también podemos ver dentro de ella, una imagen que consta de cuatro partes. En la parte superior izquierda se muestra la imagen en colores que está siendo procesada. En la parte inferior izquierda se muestran en color rojo sobre un fondo negro los objetos en movimiento que fueron detectados, y en color ciano un pequeño cuadrado que indica el centroide del objeto detectado. En la parte inferior derecha se muestra la imagen de fondo. Finalmente, en la parte superior derecha se muestra una composición del resultado obtenido, resaltando en color rojo los objetos en movimiento. Esta figura mostrada sirve como referencia para comparar y analizar los resultados de manera visual.

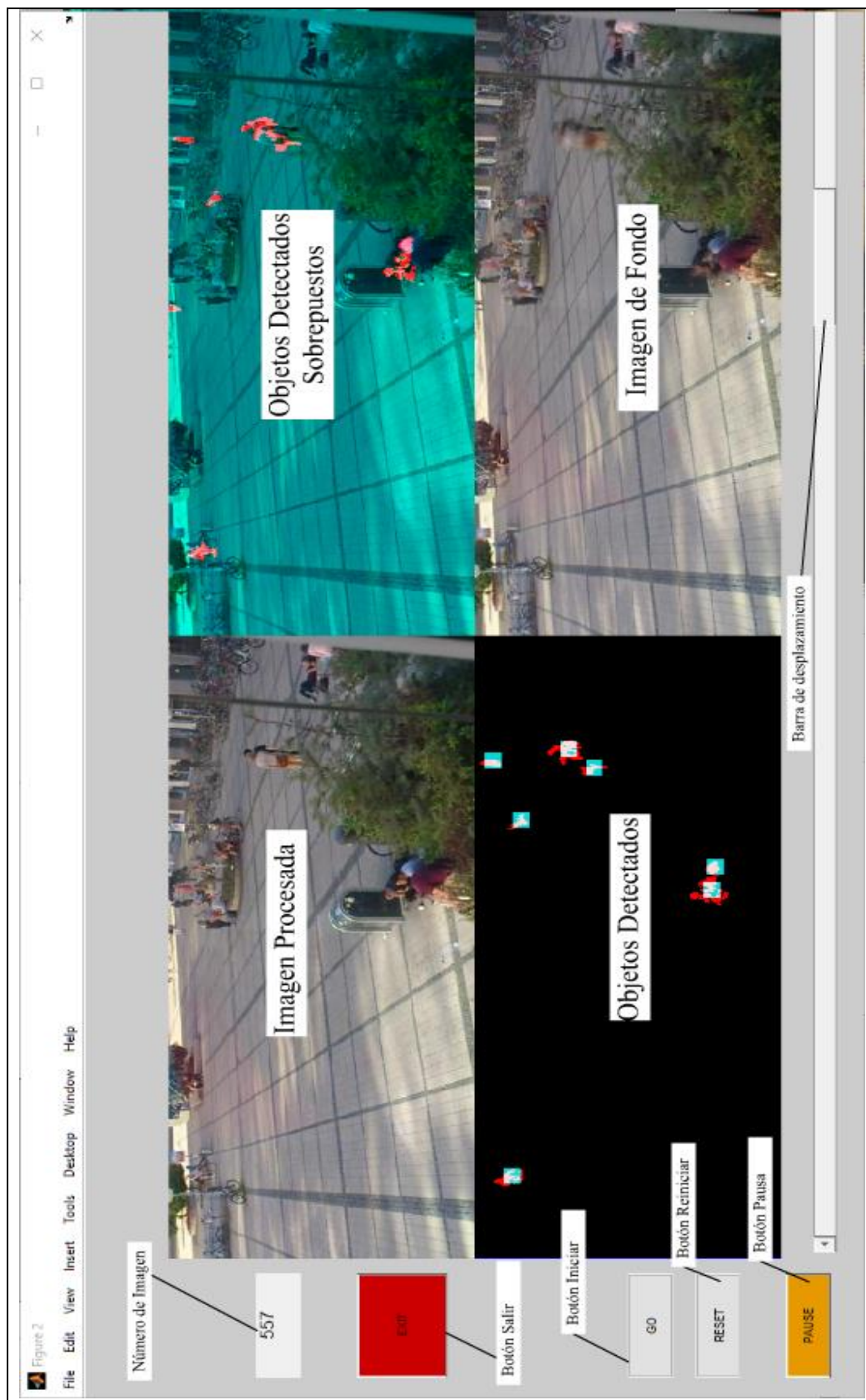


Figura 4-17. Ventana de figuras.

4.9.Pseudocódigo del algoritmo

Hasta aquí, se ha descrito el procedimiento general que sigue el algoritmo de segmentación del movimiento y se mostró el diagrama de flujo del procedimiento general; con esto en mente, ya es posible mostrar el pseudocódigo para tener una mejor comprensión de la forma en que funciona el algoritmo.

```
// Se inicia obteniendo la imagen de Fondo //
Si (existe la imagen de fondo) Entonces
    Ir al bucle principal;
    Comenzar a trabajar a partir de la primera imagen;
Si no entonces
    // Obtener imagen de fondo mediante promediado de N imágenes //
    Realizar promediado de las primeras N imágenes y asignar resultado como imagen de fondo;
    Comenzar a trabajar a partir de la siguiente imagen;
Fin si

// ** BUCLE PRINCIPAL ** //
PROCEDIMIENTO SEGMENTAR_MOVIMIENTO ();
    Mientras (exista imagen consecutiva) Hacer
        // Se realiza la resta pixel a pixel en cada capa de color //
        Resultado_Capa_Roja ← abs ( Imagen_Actual_Rojo - Imagen_De_Fondo_Rojo );
        Resultado_Capa_Verde ← abs ( Imagen_Actual_Verde - Imagen_De_Fondo_Verde );
        Resultado_Capa_Azul ← abs ( Imagen_Actual_Azul - Imagen_De_Fondo_Azul );

        // Binarizar cada capa de color según un umbral//
        Para cada (pixel ∈ Resultado_Capa_Roja) Hacer
            Si (valor_pixel > umbral_Rojo)
                Valor_pixel ← 1;
            Si no entonces
                Valor_pixel ← 0;
            Fin Si
        Fin para cada

        Para cada (pixel ∈ Resultado_Capa_Verde) Hacer
            Si (valor_pixel > umbral_Verde)
                Valor_pixel ← 1;
            Si no entonces
                Valor_pixel ← 0;
            Fin Si
        Fin para cada

        Para cada (pixel ∈ Resultado_Capa_Azul) Hacer
            Si (valor_pixel > umbral_Azul)
                Valor_pixel ← 1;
            Si no entonces
                Valor_pixel ← 0;
            Fin Si
        Fin para cada
```

```

// Supresión de artefactos //
Para cada imagen_binarizada Hacer
    Calcular área de los objetos detectados;
    Si (área de objeto < Area_Mínima_Permitida) Hacer
        Eliminar objeto;
    Fin Si
Fin Para cada

// Combinar Capas Resultantes //
resultado ← Resultado_Capa_Roja OR Resultado_Capa_Verde OR Resultado_Capa_Azul;

// Etiquetado //
Detectar todos los grupos de pixeles conectados en resultado;
Asignar a cada grupo un identificador único;
Calcular centroides de cada objeto;
Calcular el área de cada objeto;
Escribir en variable Datos_Tmp los centroides, identificadores, áreas, y total
de objetos;

Obtener imagen consecutiva;
Si (Imagen_De_Fondo es promediado) Entonces
    Actualizar Imagen de Fondo;
Fin de Si
Fin Mientras // Bucle Principal //
Escribir en archivo de texto plano la variable Datos_Tmp;
FIN PROCEDIMIENTO

```

4.10. Conclusiones del capítulo

En este capítulo se ha explicado la forma en que funciona esta propuesta para realizar la segmentación del movimiento. Se mostró tanto el diagrama de flujo y su descripción, y el pseudocódigo. Con esto es posible ver que por su funcionamiento se puede modificar fácilmente para realizar otras tareas parecidas a la segmentación y rastreo de objetos. Cumple con una de las metas principales: que se pueda implementar en tiempo real en computadoras de casa u oficinas pequeñas con los estándares actuales de videovigilancia. También se desarrolló la interfaz del usuario que es sencilla y fácil de usar.

En el tema de la segmentación del movimiento y en general en la visión por computadora, la investigación y búsqueda de nuevos algoritmos o mezcla de algunos ya existentes siguen de manera efervescente.

En el siguiente capítulo se analizan algunos experimentos, las limitaciones del algoritmo propuesto, así como los errores y/o fallas del mismo.

Capítulo 5. Evaluación de los resultados

En un trabajo como el que el lector tiene en sus manos, es común presentar resultados del algoritmo propuesto y su respectiva comparación con algoritmos similares. A pesar de que la segmentación de movimiento es un tema muy común en la literatura científica computacional actual, al buscar en el estado del arte métodos de comparación con este algoritmo, no fue posible encontrar videos procesados cuyos resultados se pudieran tomar como referencia para comparar los resultados de este algoritmo. Se analizaron más de 30 documentos científicos relacionados al tema y ninguno menciona su implementación en tiempo real, ni el procesamiento por separado de las tres capas de color, o el análisis de los objetos detectados.

Los resultados obtenidos de los algoritmos procesados en los documentos, normalmente vienen ejemplificados con imágenes en escala de grises, y en la mayoría de las veces solo son algunas imágenes procesadas y dos o tres resultados en blanco y negro o en escala de grises. No fue posible encontrar en alguna página web, resultados de videos procesados que fueran reconocidos como *ground truth*, mucho menos una secuencia de más de 100 imágenes procesadas.

Tal vez, desarrollar un método de comparación, podría ser parte de otro tema de tesis.

Así las cosas, se realizaron algunos experimentos. Se especifica que la implementación del algoritmo propuesto, así como las pruebas, fueron realizadas en una computadora personal con un procesador Core-i3, con 4GB en memoria RAM. Esto supone una computadora de características “normales” dentro de la diversidad de las computadoras actuales, es decir, no tiene una gran capacidad computacional como la tendría un servidor, y tampoco es tan limitada como lo sería una computadora con recursos demasiado básicos.

Los procesos morfológicos normalmente son exigentes computacionalmente, pero a pesar de utilizar algunos de ellos, el costo computacional no es muy alto, porque son aplicados a imágenes binarias lo cual los agiliza enormemente.

En ocasiones, por la misma naturaleza de las imágenes o por la electrónica de los dispositivos utilizados para capturar las imágenes, alguna de las capas de color puede afectar de manera negativa el resultado; en estos casos, con este proceso se le pueden asignar diferentes ponderaciones a cada capa de color, y con esto mejorar los resultados.

Es posible llevar a cabo la escritura de los datos obtenidos en tiempo real en un archivo de texto, los que después de ser almacenados, se analizan para brindar un reporte acerca de los datos calculados sobre los objetos en movimiento encontrados en los videos analizados.

5.1.Experimentos

Para entender cómo obtener mejores resultados se realizaron algunos experimentos. A continuación se muestran los resultados de algunos de ellos; los experimentos consistieron, por ejemplo, si el desplazamiento es a profundidad, o cómo afecta si el objeto detectado cambia de posición o simplemente varían los parámetros que utiliza el algoritmo.

5.1.1. Variación del umbral de diferencia entre pixeles

Para clasificar los pixeles que pertenecen a objetos en movimiento, se establece un umbral de diferencia que debe existir entre el valor de estos y los que pertenecen a la imagen de fondo. Se realizó el ejercicio de cambiar este umbral en una misma imagen con buena iluminación en el medio ambiente, y con un umbral de apertura por área de cincuenta pixeles. En la siguiente lista se muestran los valores que se utilizaron para la diferencia de imágenes, y el número de objetos detectados. El archivo de video es "mensa.avi" y el fotograma analizado es el número 138.

Tabla del experimento 5.1.1	
Valor del umbral FD	Número de objetos detectados
5	44
8	30
11	19
13	16
15	17
17	16
19	16
21	14
23	9
25	9
27	9
30	9
33	8
36	8
39	9
42	7
45	5
47	5

Tabla 5-1. Relación del umbral de diferencia y objetos detectados.

Al aumentar el umbral de diferencia entre pixeles, se obtiene un menor número de objetos detectados, debido a que si hay poca diferencia entre los pixeles, algunos de los objetos se confundirán con el fondo y pasarán a formar parte de éste; si por el contrario, el umbral de diferencia es muy pequeño, muchos de los pixeles que son parte del fondo, se detectarán como falsos positivos y formarán grupos de pixeles que aparentarán ser objetos independientes, aunque en realidad no lo sean.

En algunas imágenes, la segmentación es deficiente por las condiciones de poca luz, o por la similitud entre los colores del objeto en primer plano y el fondo de la imagen.

En la figura 5-1 se muestran las imágenes correspondientes a este experimento; el área en color rojo muestra los grupos de pixeles que han sido detectados como objetos en movimiento.

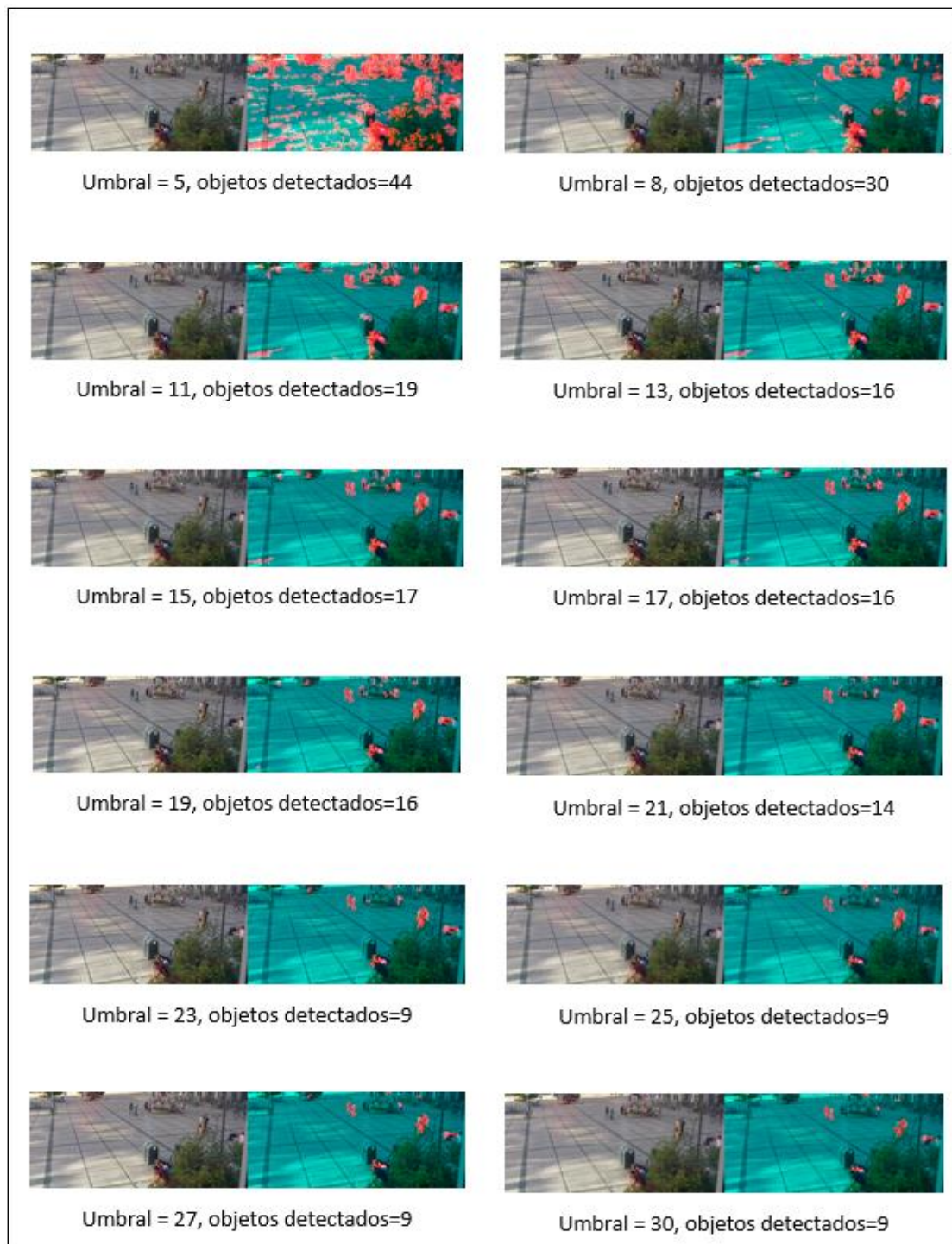


Figura 5-1. Muestra de la variación del umbral FD en la misma imagen. El área roja muestra los pixeles que fueron detectados como objetos en movimiento.

En la siguiente gráfica de la figura 5.2, se muestra la relación entre el umbral establecido y el número de objetos detectados.

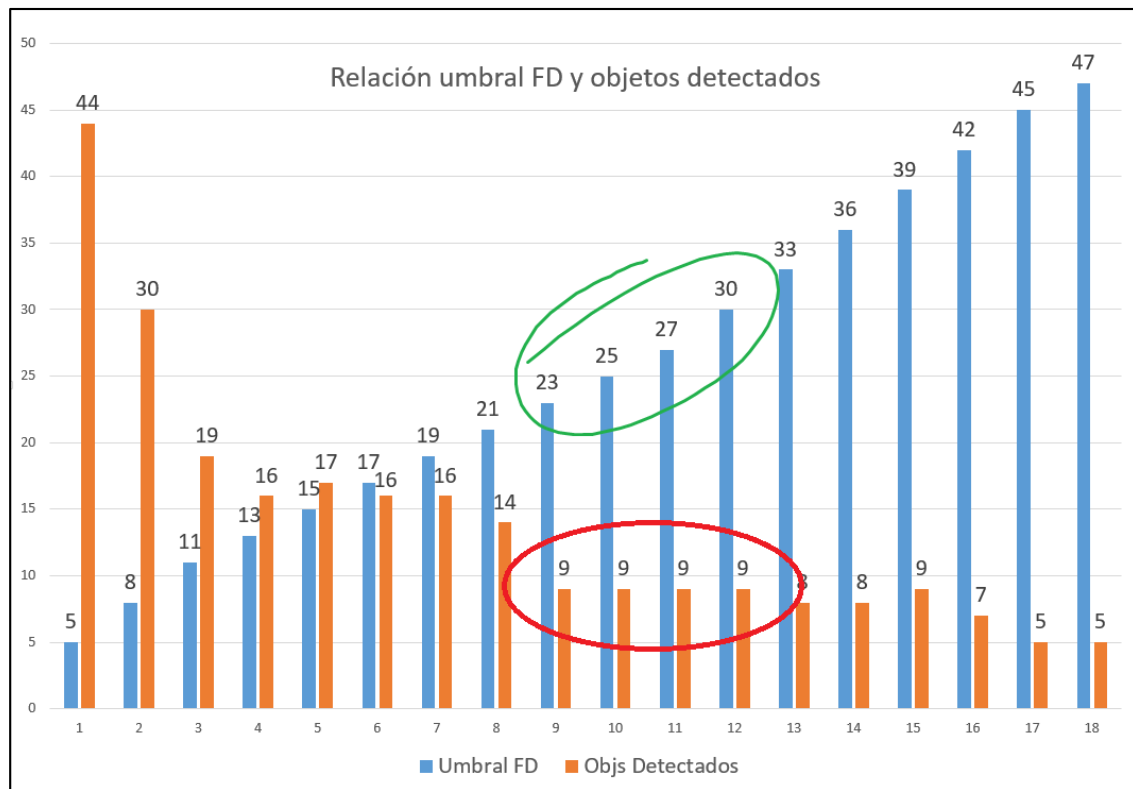


Figura 5-2. Relación entre el umbral FD y los objetos detectados en el experimento anterior.

En el fotograma que se utilizó, los objetos detectados en realidad son nueve; en la gráfica de la figura 5-2 se puede apreciar que del umbral 23 al 30 que se muestran encerrados en un círculo verde, se detectan correctamente los positivos verdaderos, los cuales se muestran dentro de una elipse roja. Esto nos indica que nuestro sistema se desenvuelve bien con un umbral de FD=26, para imágenes en colores con profundidad de ocho bits y en los casos en que la iluminación es buena; no se puede decir que éste sea el mejor umbral, pues esto dependerá de las condiciones en las que se utilice este algoritmo; cuando las condiciones sean más adversas, se necesitará buscar un umbral que se adapte mejor a esas condiciones realizando otros experimentos.

5.1.2. Cambio en el umbral de apertura por área

El umbral de apertura por área que utiliza este algoritmo, se utiliza para eliminar el ruido y/o artefactos que se generan en el proceso de resta de imágenes; sin embargo, puede utilizarse también con otros fines, por ejemplo, para filtrar objetos cuya área se encuentre entre un umbral mínimo y uno máximo, es decir, en una secuencia de video donde estén mezclados automóviles y peatones, si se desea sólo contar las personas que aparecen en el video, podemos quitar el ruido con un umbral pequeño, y utilizar un umbral mayor para eliminar los autos (que son considerablemente más grandes que los humanos), poniendo como regla que si el área del objeto detectado es mayor a cierto umbral, no se tome en cuenta.

En otro experimento se tomó el fotograma número 2 de una secuencia de video, se analizó tan solo variando el umbral de apertura por área; en la siguiente tabla, se muestran los resultados obtenidos variando el umbral de apertura. El archivo utilizado fue “jena.avi.

Tabla del experimento 5.1.2	
Valor umbral apertura por área	Número de objetos detectados
0	564
3	192
6	41
9	40
12	23
15	20
18	19
21	17
24	16
27	14
30	14
33	12
36	11
39	11
42	11
45	10
48	10
51	10
55	9
60	7
65	7
70	7
80	6
200	4

Tabla 5-2. Relación del umbral de apertura por área y el número de objetos detectados.

En la información mostrada, se aprecia que si el umbral es demasiado pequeño se detectan demasiados objetos (que son falsos positivos), y a medida que se incrementa el umbral, el número de objetos detectados va disminuyendo. Dependiendo de los resultados que deseamos obtener, así se decidirá en valor del umbral que usará nuestro algoritmo. Si detectamos que un objeto está francamente en movimiento y el algoritmo no lo detecta, no podemos declarar que es un falso negativo, como se mencionó anteriormente, pues puede ser que estemos discriminando algunos objetos por su tamaño.

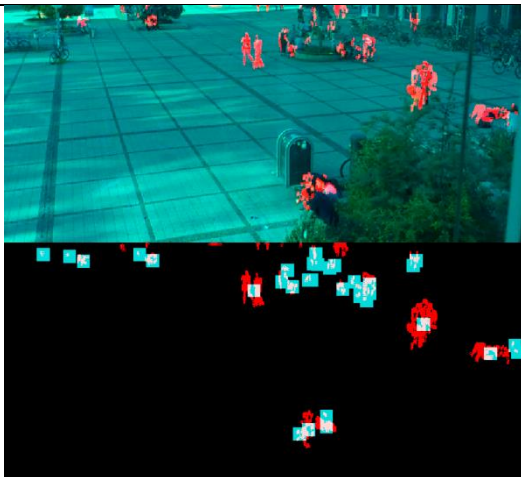
En la figura 5-3 se muestran algunas imágenes con los resultados obtenidos.



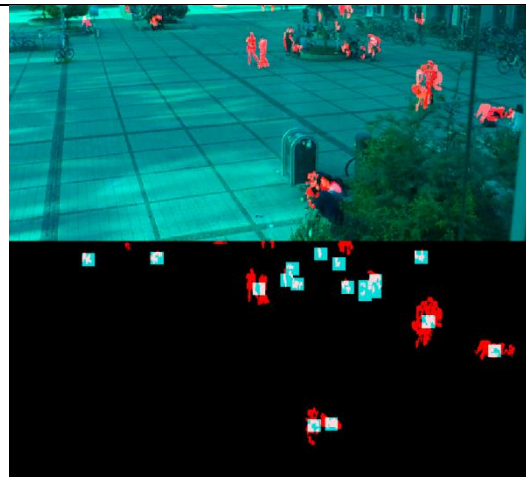
Umbral=0, objetos detectados=564.



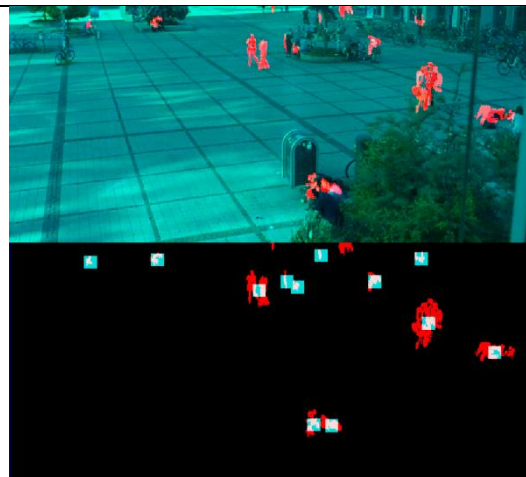
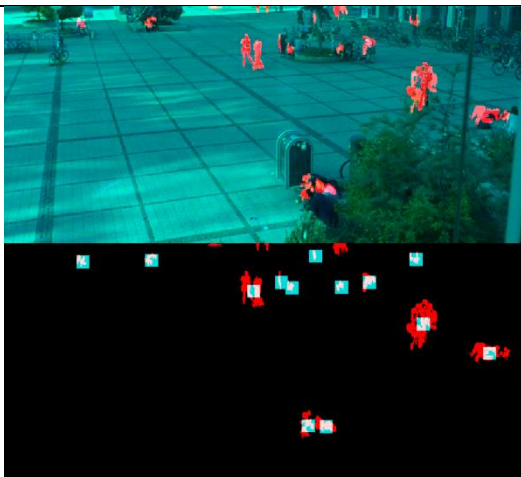
Umbral=3, objetos detectados=192.



Umbral=6, objetos detectados=41.



Umbral=12, objetos detectados=23.



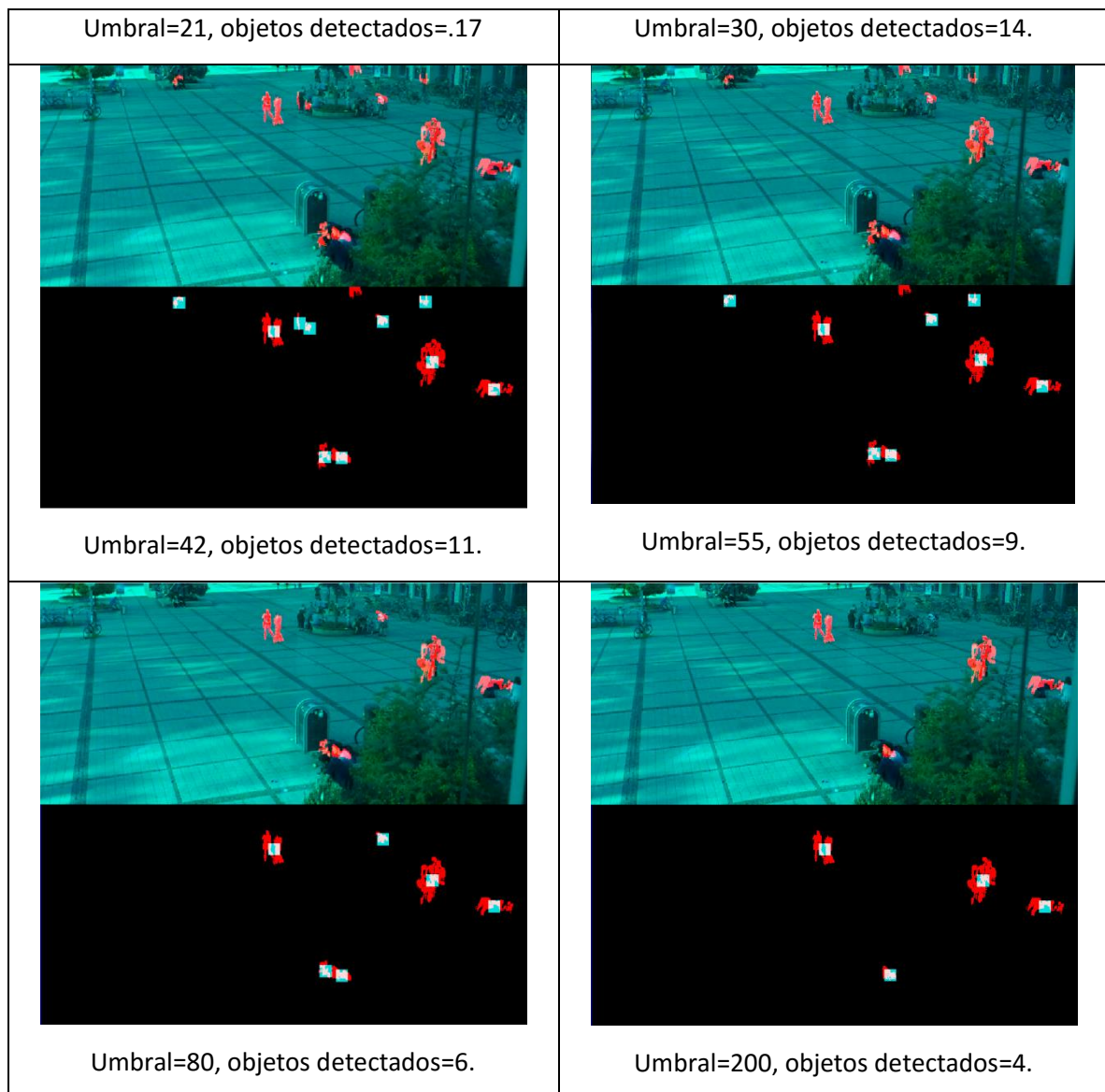


Figura 5-3. Imágenes obtenidas al variar el umbral de la apertura por área. En cada imagen se muestran en la parte superior los objetos detectados sobrepuestos a la imagen original; en la parte inferior la imagen nos muestra cada objeto con su centroide en un rectángulo color ciano. En la secuencia de imágenes se nota cómo al aumentar el umbral disminuye el número de objetos detectados.

En la última imagen de la figura 5-3, solo se detectaron 4 objetos; las áreas de esos objetos son 195, 514, 652, y 1116 píxeles; con estas cifras podemos saber cuánto deberíamos aumentar el umbral para desaparecer uno a uno los objetos detectados.

En la figura 5-4 se muestra la relación entre los objetos detectados y el cambio en el umbral de la apertura por área.

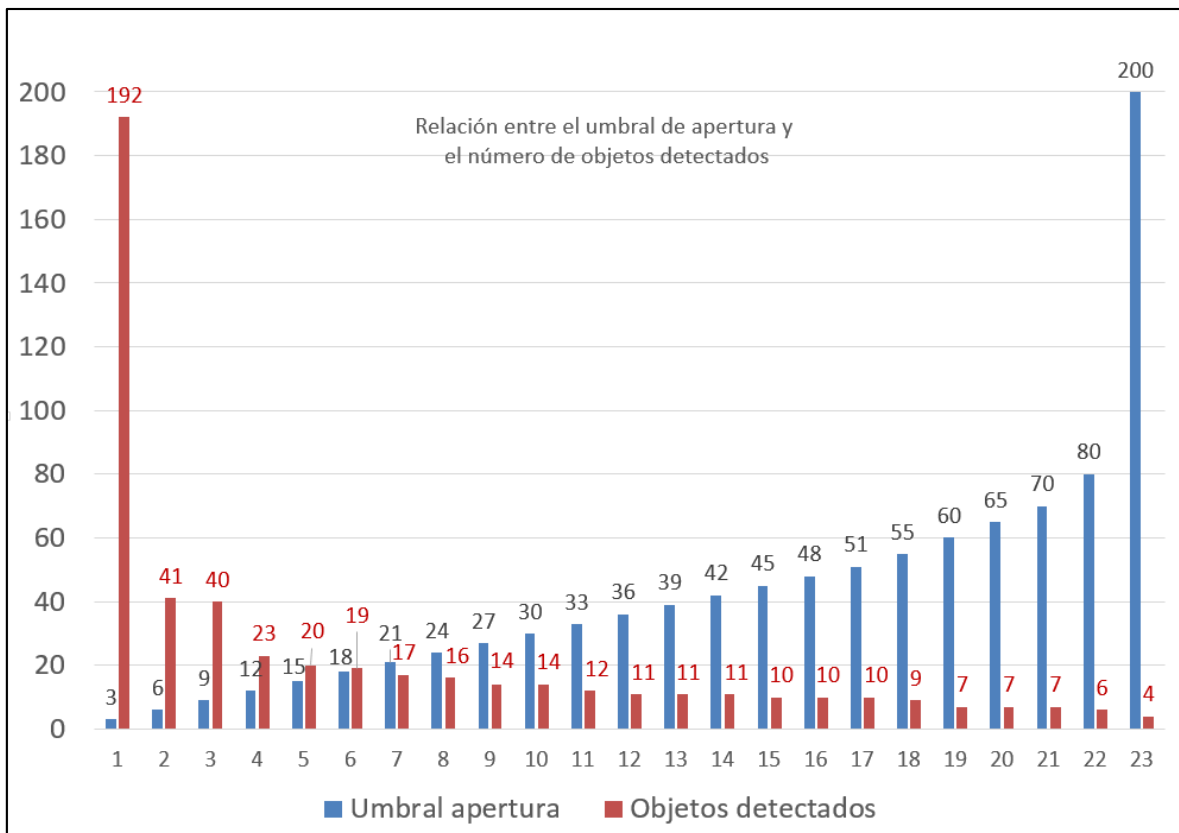


Figura 5-4. Relación entre el umbral de apertura y el número de objetos detectados.

En la interfaz de usuario que se creó para este algoritmo, solo se pueden cambiar los parámetros de apertura por área y el umbral de FD. La selección de valores para obtener buenos resultados se realizará de manera manual y dependerá de los objetivos que se quieran alcanzar y de las condiciones de la iluminación en el video procesado.

5.1.3. Desplazamiento alejándose de la cámara

Es intuitivo pensar que si se realiza el análisis de un objeto que se aleja de la cámara, disminuirá su área en píxeles, pero no siempre es así. En el siguiente experimento se realiza el análisis de una persona que se aleja de la cámara, pero que en un cierto fotograma, a pesar de alejarse, se detecta un crecimiento en el área del sujeto.

En la figura 5.5 se muestran cinco imágenes que resultaron de la segmentación de un sujeto en movimiento; en la figura se detalla la posición del centroide en "y" y en "x"; también se especifica el área del objeto detectado y se muestra su centroide con un pequeño cuadro de color ciano. El archivo de video es "Emi07.avi".


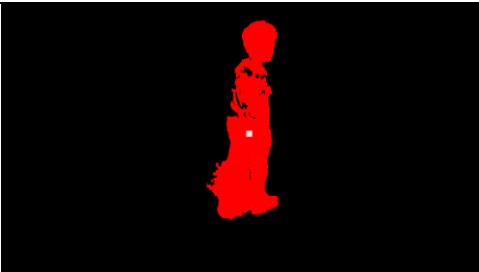
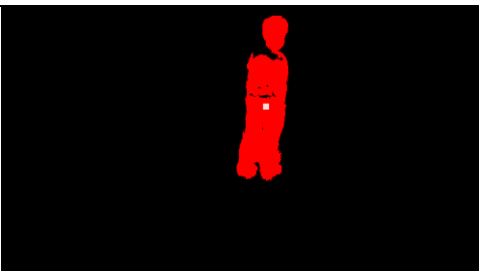
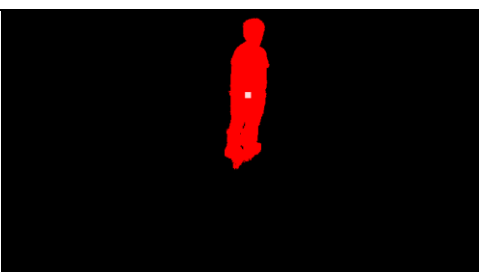
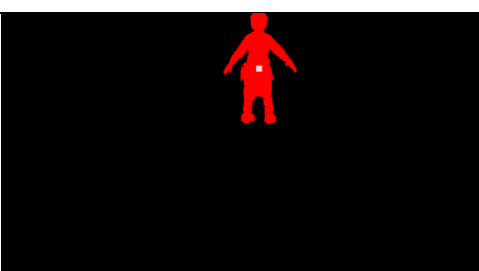
	<p>Número de fotograma:604</p> <p>Posición en Y: 370</p> <p>Posición en X:620</p> <p>Área:54000</p>
	<p>Número de fotograma:650</p> <p>Posición en Y: 352</p> <p>Posición en X:643</p> <p>Área:60082</p>
	<p>Número de fotograma:706</p> <p>Posición en Y: 265</p> <p>Posición en X:710</p> <p>Área:38508</p>
	<p>Número de fotograma:744</p> <p>Posición en Y: 224</p> <p>Posición en X:662</p> <p>Área:30902</p>
	<p>Número de fotograma:890</p> <p>Posición en Y: 147</p> <p>Posición en X:689</p> <p>Área:21523</p>

Figura 5-5. Análisis del movimiento de una persona alejándose de la cámara.

En la figura 5-6 se puede apreciar una gráfica que muestra el aumento del área en el objeto detectado.

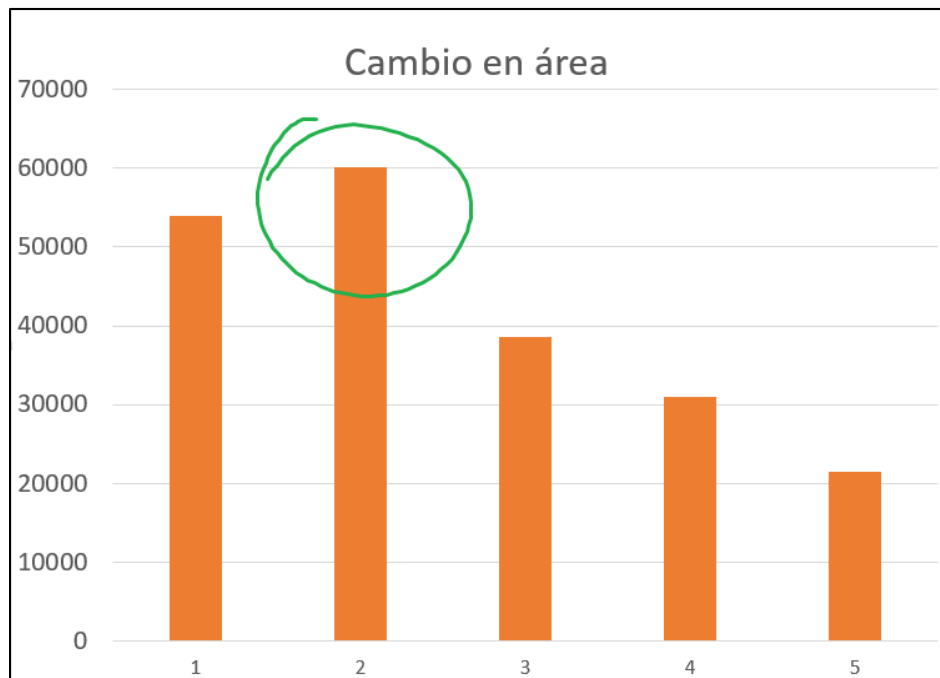


Figura 5-6. Análisis de un objeto que se aleja de la cámara. En el círculo verde se nota un aumento en el área del objeto, a pesar de que éste se aleja de la cámara.

Los resultados del experimento anterior nos muestran que, al intentar describir el movimiento de un objeto que se acerca a la cámara tan solo por el cambio del área en píxeles, existen grandes posibilidades de que obtengamos errores. No porque exista un aumento en el área significa que el objeto se acerca, sino pueden ser tan solo movimientos naturales del cuerpo los que aparentan un acercamiento. También existe la otra cara de la moneda para este error, el cual se analiza a continuación. De todos modos, el resultado global de un objeto que se aleja de la cámara, aparecerá indudablemente de menor tamaño en fotogramas sucesivos.

5.1.4. Desplazamiento acercándose hacia la cámara

En este experimento se analiza una persona que se acerca hacia la cámara, y al igual que en el experimento anterior, se lleva un registro del área en píxeles del objeto. En la figura 5-7 se muestran las imágenes analizadas y sus estadísticas.

Se puede ver cómo a pesar de que el sujeto se acerca hacia la cámara, en dos de los fotogramas se detecta una disminución en el área, lo cual nos indica que no siempre existirá un aumento en el área detectada aunque el movimiento sea de aproximación. Por lo tanto, tampoco se puede afirmar que si existe un incremento en el área, implica que el sujeto se acerca al dispositivo de grabación de video.

El archivo analizado es "Emi08.avi"; en la secuencia de figuras se puede ver el centroide del objeto como un pequeño rectángulo en color ciano.






	<p>Número de fotograma:260</p> <p>Posición en Y: 141</p> <p>Posición en X:697</p> <p>Área:21931</p>
	<p>Número de fotograma:336</p> <p>Posición en Y: 201</p> <p>Posición en X:682</p> <p>Área:28721</p>
	<p>Número de fotograma:402</p> <p>Posición en Y: 264</p> <p>Posición en X:660</p> <p>Área:38366</p>
	<p>Número de fotograma:422</p> <p>Posición en Y: 341</p> <p>Posición en X:615</p> <p>Área:37256</p>
	<p>Número de fotograma:290</p> <p>Posición en Y: 297</p> <p>Posición en X:660</p> <p>Área:33198</p>

Figura 5-7. Secuencia de imágenes de sujeto acercándose a la cámara.

En la figura 5-8 se muestra gráficamente la variación del área, así como que hay una disminución incluso en dos de los fotogramas, debido a que los movimientos del sujeto influyen en la cantidad de píxeles detectados; esto también se analiza en el siguiente experimento.

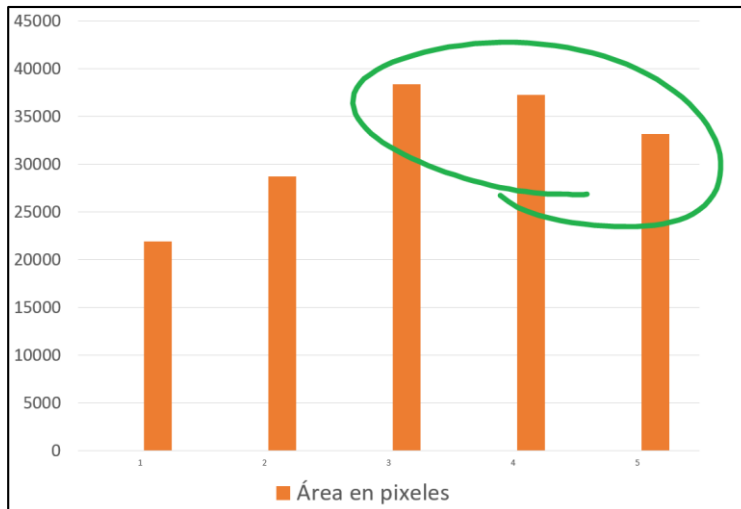


Figura 5-8. Gráfica de la variación de área del sujeto que se acerca a la cámara, donde se puede apreciar en la marca verde que hay dos decrementos sucesivos en el valor del área.

Los experimentos de acercarse y alejarse nos muestran que no siempre existirá una relación entre movimiento a profundidad y el cambio en el área de los objetos. Por tanto, no es recomendable basarse en estos eventos para llevar una estadística relacionada al desplazamiento de los objetos. El cambio en el área puede verse afectada por diferentes motivos, por ejemplo cambios en la posición del sujeto, o tal vez si el sujeto genera sombras solo en ciertas ocasiones durante la secuencia de imágenes, o por ejemplo, un automóvil visto de frente, podría ocupar menos espacio que de perfil, aunque estuviera más alejado; también la generación de reflejos y sombras afecta al área detectada, así como la iluminación.

5.1.5. Variación del área del sujeto a la misma distancia

En este sencillo experimento, se puede ver cómo a pesar de que el sujeto se encuentra a la misma distancia, tan solo con cambiar su posición varía enormemente el espacio que ocupa en la imagen.

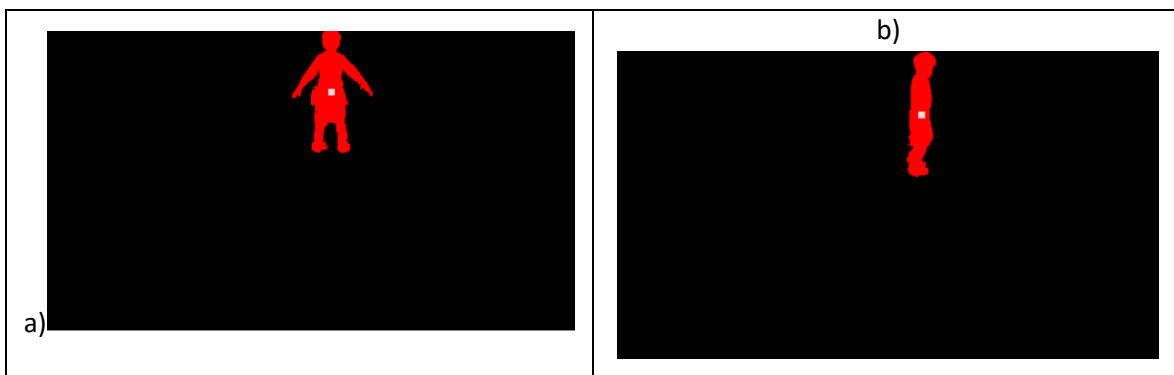


Figura 5-9. Comparación del mismo sujeto a la misma distancia en diferente posición. En a) se ocupa un área de 21,523 píxeles, en b) ocupa un área de 13119 píxeles.

En la figura 5-9 se observan dos imágenes: en la primera el cuerpo detectado ocupa un área de 21,523 pixeles, mientras que en la segunda se detectó un área de 13,119 pixeles. La lejanía del sujeto es la misma en ambas imágenes, pero el área en la segunda imagen es de tan solo el 61% aproximadamente con respecto a la de la primera imagen.

Esto también genera problemas cuando si quiere realizar la correlación de los objetos para generar el reporte, pues el algoritmo intenta buscar el mismo objeto en diferentes fotogramas no utilizando la posición, sino también el área, lo cual puede generar resultados inesperados al existir tanta diferencia en el mismo objeto.

5.1.6. Generación del reporte

El algoritmo diseñado, es un segmentador de movimiento y caracterizador de los objetos detectados. La parte final de la comparación de las imágenes, nos genera un archivo de texto a partir del cual podemos generar este reporte de caracterización. El módulo generador del reporte, puede generar el reporte de un solo objeto o de todos los objetos detectados como se explicó en el capítulo cuatro. El comando para generar la información de únicamente un objeto, se ejecuta con el comando “caracterizaX” en la consola de comandos, indicando también el id del objeto a ser analizado; por ejemplo, si queremos el reporte del objeto con el identificador mil, se tendría que escribir en la ventana de comandos la instrucción “caracterizaX(1000);”. Al ejecutar este comando, el reporte mostró que en la secuencia analizada este objeto: apareció, se separó en dos objetos, se fusionó con otro objeto, volvió a separarse y desapareció, todo en solo siete cuadros de la secuencia de imágenes, para este ejemplo se utilizó el archivo jena.avi. Aquí se muestra la ejecución del comando y la salida en consola del reporte generado de este objeto:

>> caracterizaX(1000);	Instrucción en consola.
Salida:	
<pre> INICIO de:1000 ***** 1000 ->1008 1008 ->1016 1016 Se separo en :1025 y :1026 1025 Se fusiono con :1026 y formo :1033 1033 ->1042 1042 Se separo en :1050 y :1051 1050 Se desaparecio. </pre>	<pre> Encabezado con id del objeto Esta línea indica que el objeto 1000 se convirtió en el objeto 1008. Esta línea indica que el objeto 1008 se convirtió en el objeto 1016. Esta línea indica que el objeto 1016 se dividió y resultaron los objetos 1025 y 1026. Esta línea indica que los objetos 1025 y 1026 se fusionaron y formaron el objeto 1033. Esta línea indica que el objeto 1033 se convirtió en el objeto 1042. Esta línea indica que el objeto 1042 se dividió y resultaron los objetos 1050 y 1051. Esta línea indica que el objeto 1050 desapareció. </pre>

Esta función es llamada consecutivamente para generar el reporte completo, este reporte completo se puede ejecutar con el comando caracteriza, y no lleva ningún parámetro, tan solo se tiene que ejecutar en el mismo directorio en donde se encuentre el archivo con los datos de salida del proceso anterior, el archivo de texto con el nombre “analizado.txt”.

Un fragmento de un ejemplo del reporte final se muestra a continuación:

```
Aparecio objeto con id=456, en y=224, y x=384 con area de:525pixeles, en el cuadro:104
(456) se convirtio en (465) en cuadro:104, se movio:4 pixeles hacia el:Noreste
(465) se convirtio en (475) en cuadro:106, se movio:6 pixeles hacia el:Sureste
(475) se convirtio en (486) en cuadro:108, se movio:1 pixeles hacia el:Oeste
(486) se convirtio en (498) en cuadro:110, se movio:1 pixeles hacia el:Este
(498) se convirtio en (509) en cuadro:112 No se desplazo.
(509) se convirtio en (521) en cuadro:114, se movio:2 pixeles hacia el:Oeste
(521) se convirtio en (532) en cuadro:116 No se desplazo.
(532) se convirtio en (545) en cuadro:118, se movio:1 pixeles hacia el:Este
(545) se convirtio en (556) en cuadro:120 No se desplazo.
(556) se convirtio en (565) en cuadro:122, se movio:1 pixeles hacia el:Este
(565) se convirtio en (574) en cuadro:124, se movio:1 pixeles hacia el:Sur
(574) se convirtio en (585) en cuadro:126, se movio:1 pixeles hacia el:Noroeste
(585) Se desaparecio en cuadro:128.
***** Inicio reporte de objeto con id:460 *****
Aparecio objeto con id=460, en y=134, y x=583 con area de:295pixeles, en el cuadro:104
(460) se convirtio en (469) en cuadro:104, se movio:1 pixeles hacia el:Oeste
(469) Se fusiono con (470) y formo (479) en cuadro:108
(479) se convirtio en (491) en cuadro:108 No se desplazo.
(491) Se separo en (503) y (504) en cuadro:112
(503) se convirtio en (514) en cuadro:112, se movio:1 pixeles hacia el:Norte
(514) se convirtio en (526) en cuadro:114, se movio:1 pixeles hacia el:Sur
(526) se convirtio en (536) en cuadro:116, se movio:1 pixeles hacia el:Noreste
(536) Se fusiono con (537) y formo (548) en cuadro:120
(548) se convirtio en (559) en cuadro:120, se movio:2 pixeles hacia el:Suroeste
(559) se convirtio en (568) en cuadro:122, se movio:5 pixeles hacia el:Este
(568) se convirtio en (578) en cuadro:124, se movio:4 pixeles hacia el:Oeste
(578) Se desaparecio en cuadro:126.
***** Inicio reporte de objeto con id:461 *****
Aparecio objeto con id=461, en y=133, y x=606 con area de:196pixeles, en el cuadro:104
(461) se convirtio en (470) en cuadro:104, se movio:5 pixeles hacia el:Noroeste
(470) Se fusiono con (469) y formo (479) en cuadro:108
(479) se convirtio en (491) en cuadro:108 No se desplazo.
(491) Se separo en (503) y (504) en cuadro:112
(503) se convirtio en (514) en cuadro:112, se movio:1 pixeles hacia el:Norte
(514) se convirtio en (526) en cuadro:114, se movio:1 pixeles hacia el:Sur
(526) se convirtio en (536) en cuadro:116, se movio:1 pixeles hacia el:Noreste
(536) Se fusiono con (537) y formo (548) en cuadro:120
(548) se convirtio en (559) en cuadro:120, se movio:2 pixeles hacia el:Suroeste
(559) se convirtio en (568) en cuadro:122, se movio:5 pixeles hacia el:Este
(568) se convirtio en (578) en cuadro:124, se movio:4 pixeles hacia el:Oeste
```

El archivo de texto generado lleva por nombre “finalRep.txt”.

Se analizaron 30 segundos de la secuencia de video "jena.avi", el archivo de texto generado del reporte final de este video, tiene 4,303 líneas, y tardó en generarse poco más de 11 horas. En promedio, en este video se detectaron 11 objetos por fotograma.

El tiempo de obtención de los resultados obtenidos, dependerá en gran parte, del número de objetos detectados por cada fotograma. En el tiempo de análisis, influyen en gran medida, los objetos que desaparecieron sin motivo aparente, porque se tiene que llevar un historial de ellos, y posteriormente estarlos comparando constantemente con cada objeto que aparece, pues es la única forma de saber si han reaparecido o no; esto nos va generando un cantidad considerable de objetos que se están comparando repetidamente. Normalmente, la lista de objetos en espera de ser reencontrados, incrementará.

El tiempo de procesamiento de esta información es demasiado grande, por lo tanto no, fue posible ejecutarlo ni mostrar los resultados en tiempo real.

Los datos obtenidos en este reporte, solo encuentran la correlación de la información dependiendo de la posición y el área de los objetos detectados, los resultados completamente dependerán de la información obtenida de los pasos anteriores.

Un fruto de este reporte es que, a partir del análisis espacio-temporal del movimiento de los objetos presentes en las imágenes, se obtiene su caracterización

5.2.Limitaciones del sistema

En el mundo de la visión por computadora, es sabido que cuando se quiere segmentar un objeto que tiene un color similar a la imagen de fondo, las técnicas de sustracción de fondo o de diferencia de imágenes tienen un desempeño muy pobre; esta es la principal limitación de un sistema basado en diferencia de imágenes. Por lo tanto, se tiene que optar por otras técnicas que normalmente implican el uso de hardware más especializado para lograr los objetivos, por ejemplo, cámaras térmicas o infrarrojas, el dispositivo Kinect, detectores de movimiento colocados en puntos estratégicos, etc.; estas limitaciones también afectan negativamente al desarrollo que se presenta en este documento.

5.2.1. Influencia de las sombras en el área de los objetos

En la figura 5-10 se pueden apreciar 2 fotogramas consecutivos en las que se resalta la misma persona con una marca verde; en la primera podemos ver que la persona que se detectó junto con su propia sombra da un área de 1,721 píxeles; en la segunda imagen la sombra se ha separado, y el área de la misma persona se ha reducido a 1,252 píxeles.

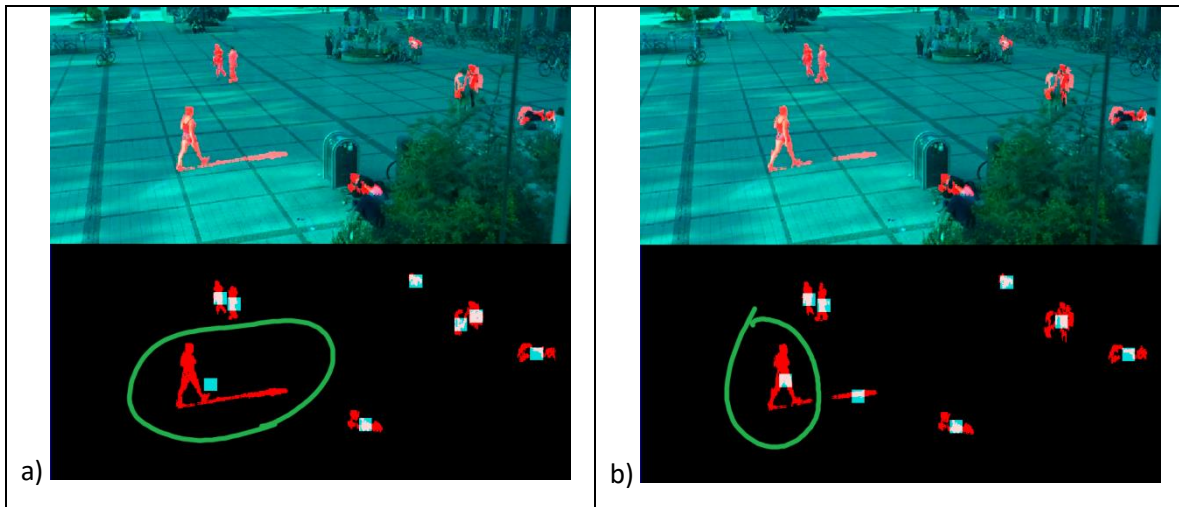


Figura 5-10. Comparación del área del mismo sujeto con sombra y sin sombra. En a) el sujeto con sombra es resaltado con una marca verde, en b) el mismo sujeto, ya sin sombra, también resaltado con una marca verde.

Este ejercicio nos permite entender de mejor manera que las sombras son un problema para llevar a cabo un buen análisis de los objetos detectados. Es difícil encontrar un motivo para querer detectar las sombras generadas por los objetos detectados.

La figura 5-10 b) también muestra que la sombra es detectada como otro objeto en movimiento, y por tanto se genera un registro de un objeto nuevo encontrado; es muy probable que al generar el reporte de los objetos detectados, este tipo de eventos genere resultados no deseados. Este fenómeno que se puede calificar como “indeseable”, es un problema común tanto en interiores como en exteriores.

En un principio, se intentó deducir si el objeto detectado era una persona o no, tan solo por la relación de aspecto, pero dado que las personas podemos tomar muchas formas y la relación de aspecto varía demasiado, esto no fue posible. En el ejemplo anterior, también se nota cómo las sombras influyen de manera negativa para lograr este fin.

5.2.2. Problemas de iluminación

En condiciones de baja iluminación, el algoritmo tiene un mal desempeño; al intentar modificar los umbrales, aparecen artefactos imposibles de detectar como tales y por tanto se detectan como falsos positivos, o disminuye la detección de positivos verdaderos.

No es posible calcular un valor fijo para la diferencia de imágenes pixel a pixel, pues en un mismo video se pueden tener tanto condiciones de poca luz como condiciones de luz adecuada para el proceso de las imágenes, y se podría requerir de dos o más umbrales dependiendo de la región de la imagen que se esté analizando.

Además de las condiciones de luz, el contraste de colores entre el fondo y los objetos que se quieren detectar, juega un papel muy importante en la detección y segmentación del movimiento. Por ejemplo en la figura 5-11, se puede ver una imagen con regiones bien iluminadas y otras con

muy poca luz, en este caso es recomendable utilizar diferentes umbrales dependiendo de la región que se esté procesando.



Figura 5-11. Imagen con iluminación muy desigual.

El procesamiento de las imágenes y la escritura de datos, se realiza de manera correcta a una tasa de 12 imágenes por segundo, procesando imágenes con una profundidad de color de 24 bits por pixel y con una resolución de 800 x 600 pixeles de ancho por alto, respectivamente.

Cuando las imágenes son de alta definición completa (FHD – *Full High Definition*), el tiempo de procesamiento aumenta demasiado, y en los mejores casos solo es posible procesar 2 imágenes por segundo o menos. Esto conlleva a tomar alguna de las siguientes decisiones:

- Se puede optar por utilizar una computadora más potente. En caso de ser necesario, este algoritmo es fácilmente transportable a otra plataforma, con hardware que tenga un mejor desempeño.
- Utilizar el algoritmo en tiempo real con una tasa ligeramente menor a un cuadro por segundo. Si las necesidades no requieren procesar una tasa de 12 cuadros por segundo, como cuando sólo se quiere saber a qué hora hubo cambios importantes en la escena, que no impliquen un escaneo mayor que un cuadro por segundo, se puede utilizar este algoritmo.
- Guardar el video en algún medio de almacenamiento masivo y procesarlo posteriormente, esto es, si no es imprescindible realizar el análisis del video en el instante de la captura de éste, se puede procesar posteriormente, aunque lleve más tiempo que la duración original del video.
- por ejemplo, al generar el reporte individual del objeto con id = 1000, el reporte mostró que en la secuencia analizada este objeto apareció, se separó en dos objetos, se fusionó con otro objeto, volvió a separarse y desapareció, todo en solo siete cuadros de la secuencia de imágenes. Aquí se muestra la ejecución del comando y la salida en consola del reporte de este objeto:

```
>> caracterizaX(1000);
```

```
Instrucción en consola.
```

- Salida:

INICIO de:1000 *****	Encabezado con id del objeto
1000 ->1008	Esta línea indica que el objeto 1000 se convirtió en el objeto 1008.
1008 ->1016	Esta línea indica que el objeto 1008 se convirtió en el objeto 1016.
1016 Se separo en :1025 y :1026	Esta línea indica que el objeto 1016 se dividió y resultaron los objetos 1025 y 1026.
1025 Se fusiono con :1026 y formo :1033	Esta línea indica que los objetos 1025 y 1026 se fusionaron y formaron el objeto 1033.
1033 ->1042	Esta línea indica que el objeto 1033 se convirtió en el objeto 1042.
1042 Se separo en :1050 y :1051	Esta línea indica que el objeto 1042 se dividió y resultaron los objetos 1050 y 1051.
1050 Se desaparecio.	Esta línea indica que el objeto 1050 desapareció.

-

5.3.Conclusiones del capítulo

Con los resultados obtenidos, se puede decir que el algoritmo propuesto funciona bien si se quiere utilizar con fines de videovigilancia o si se quiere usar tan solo como un detector de movimiento que guarde las imágenes o fragmentos de video donde se detectó movimiento.

Se pueden mejorar los resultados sacrificando el costo computacional, y con esto sacrificando también su implementación en tiempo real. Considero que es un método de fácil implementación, sencillo de utilizar y adaptar, con el cual se consiguió la meta de implementarse en tiempo real.

Desde un inicio, en los trabajos actuales se especificó que este algoritmo es altamente sensible a condiciones de luz adversas, y debe existir un buen contraste entre el fondo y los objetos para tener buenos resultados; se pudo ver que en condiciones de poca luz, es mejor optar por otra solución.

Es claro que se pueden realizar más experimentos; dependiendo de los resultados que se quieran alcanzar, se han de establecer umbrales que se ajusten a los objetivos con los que se analiza el video.

Capítulo 6. Conclusiones y trabajos futuros

Después del análisis del estado del arte, de haber hecho pruebas siguiendo diferentes caminos con el objetivo de entender la manera en que se ha atacado el problema de la segmentación de movimiento por otros autores, y de mezclar diferentes técnicas con este fin, a continuación se presentan algunas de las conclusiones de este trabajo.

Las técnicas analizadas nos muestran metodologías muy diferentes unas de otras, algunas muy intuitivas o muy ingeniosas y otras muy rebuscadas. En ocasiones, al tratar de fusionar 2 de estas técnicas con los datos obtenidos, se detectó que hay una gran incompatibilidad entre estas técnicas, por lo que no nos es posible realizar una combinación entre ellas; esto nos dice las grandes diferencias entre los métodos desarrollados para segmentar el movimiento. Es muy alentador entender cómo los investigadores han buscado por diferentes medios, métodos y estrategias la forma de conseguir este fin, lo que nos da pie a continuar intentando mejorar estas técnicas propuestas y buscar nuevos métodos que nos permitan obtener mejores resultados.

6.1.Conclusiones

En este trabajo de tesis se propone un nuevo método para obtener la segmentación del movimiento; el algoritmo en principio sigue el camino que se describe a continuación:

- Obtener una imagen de fondo, si no se puede obtener de manera directa, se puede conseguir por promediado o mediante una composición de la imagen.
- Realizar una resta de imágenes entre la imagen de fondo y el cuadro que se esté analizando, procesando cada capa de color por separado.
- Aplicar un proceso de umbralado por cada capa de color, obteniendo así tres matrices binarias, cada una correspondiente a cada capa de color.
- Eliminar los artefactos o ruido mediante un proceso morfológico de apertura por área en cada capa; en caso de que sólo se quiera analizar objetos de cierto tamaño, se pueden agregar umbrales de decisión, para eliminar los objetos que no cumplen con un mínimo y/o un máximo de tamaño.
- Realizar una operación OR con las tres matrices obtenidas para obtener los pixeles que han sido clasificados como “objetos en movimiento”.
- El análisis posterior de estos objetos en movimiento, dependerá del fin que se quiera alcanzar, el cual puede ser muy variado. En este trabajo el análisis se ha hecho con fines de videovigilancia, y se entrega un reporte acerca de los objetos detectados.

La implementación en tiempo real fue posible. Estos algoritmos se pueden implementar para su funcionamiento en tiempo real si se utilizan computadoras personales con resoluciones y tasas de transferencia normales, es decir, unos 600 x 800 pixeles, 24 bits de profundidad de color por pixel, y a 12 cuadros por segundo, en computadoras caseras o de oficinas pequeñas, o con computadoras más potentes.

Se desarrolló una interfaz sencilla de usar y los datos generados se han guardado siempre en algún medio de almacenamiento.

La implementación de técnicas que consumen muchas operaciones de cálculo matemático, aproximaciones, iteraciones en la misma imagen, etc., normalmente utiliza más tiempo de ejecución, lo cual conlleva a no poder ser implementadas en tiempo real.

También fue posible guardar los datos obtenidos en archivos de texto plano en tiempo real. El análisis de los resultados se realiza posteriormente y se entrega un reporte de ello.

Dependiendo de lo que se quiera obtener, se puede modificar el programa desarrollado, por ejemplo, para detectar movimiento, analizar la dirección del desplazamiento, analizar la cantidad de objetos en movimiento, verificar la cercanía de los objetos con algún punto de interés, etc.

6.2.Recomendaciones

Con el mejoramiento de los procesos de cómputo utilizando microchips más potentes, se pueden mejorar los resultados agregando procesos iterativos, como el algoritmo “Corte de grafos” para delinear de mejor manera los objetos detectados en movimiento.

Además de este tipo de algoritmos, existen otros que pueden mejorar los resultados si las necesidades así lo exigen, siempre tomando en cuenta que al agregar procesos pesados, aumenta el costo computacional, y podría verse reflejado en el rendimiento del algoritmo, lo que haría necesario el uso de computadoras más potentes o dedicadas exclusivamente a esta tarea.

Realizar diferentes tipos de pruebas y experimentos con diferentes umbrales, nos amplía el panorama de funcionamiento de estos algoritmos. Aunque en ocasiones las condiciones para la ejecución de los programas son propicias, recibimos mucha más retroalimentación de las ocasiones en que los retos son más difíciles de superar. Al haber trabajado desde un principio con videos reales, y al no ocasionar las condiciones propicias para obtener mejores resultados, la retroalimentación ha sido muy provechosa, de tal manera que al utilizar este mismo algoritmo ya pulido con videos más sencillos, ya no ha sido necesario modificar mucho la programación del algoritmo.

Para obtener mejores resultados, se recomienda la combinación de diferentes técnicas como las citadas en el estado del arte. Se ha visto que al fusionarlas de manera correcta muestran mejoras en las imágenes obtenidas. Combinar diferentes técnicas, de diferentes maneras, nos permite analizar resultados variados, y junto con esto, entender cómo mejorar en los algoritmos implementados.

Mientras más grande sea el análisis del estado del arte, más grande es la retroalimentación, y también más serán las ideas que surgen para el análisis del problema, pueden ser más ingeniosas y/o multidisciplinarias, lo cual culmina en una mejor solución del problema que se quiere resolver.

6.3.Trabajos futuros

Con el fin de mejorar los resultados, hay muchas implementaciones que se pueden realizar en este trabajo, a continuación se muestran algunas de ellas.

6.3.1. Agregar Histograma en el reconocimiento de objetos

Al buscar el equivalente de un objeto de una imagen en la siguiente imagen, se analizan la posición y el área en píxeles de los objetos detectados; esto se realiza entre cada par de imágenes consecutivas; puede ocurrir que se fusionen dos o más objetos de una imagen a la siguiente, por ejemplo cuando se saludan dos personas; también puede suceder que un objeto se convierta en dos o más, por ejemplo, cuando van dos personas caminando juntas y se separan; también puede suceder que algunos objetos desaparezcan, lo que puede suceder por diferentes razones.

Un mejor análisis para realizar el seguimiento de estos sucesos sería analizar el histograma de cada objeto en la imagen actual y en la imagen siguiente; esto es, analizar según la posición, el tamaño y los colores, y su correspondencia con el objeto de la siguiente imagen. Muchos trabajos que realizan el rastreo de objetos han utilizado esta técnica con muy buenos resultados.

No hay duda de que mejorarían los resultados acerca del seguimiento con la implementación de este método.

6.3.2. Agregar “cortes de grafos” para delinear mejor objetos detectados

Existe un algoritmo conocido como “Corte de Grafos” que se analizó en la sección 2.3.7, con el cual se puede separar algún objeto de una imagen, de su fondo. Una modificación al proceso analizado, consiste en seleccionar solo algunos píxeles que pertenecen al objeto que se desea segmentar; después se pueden seleccionar o no, algunos píxeles que pertenecen al fondo de la imagen.

La selección de los píxeles pertenecientes al objeto se puede hacer con una línea sobre el objeto, ya sea recta o simplemente un garabato sobre el objeto. De la misma forma, se pueden seleccionar o no, píxeles pertenecientes al fondo. Un ejemplo de esto se muestra en la figura 6.1

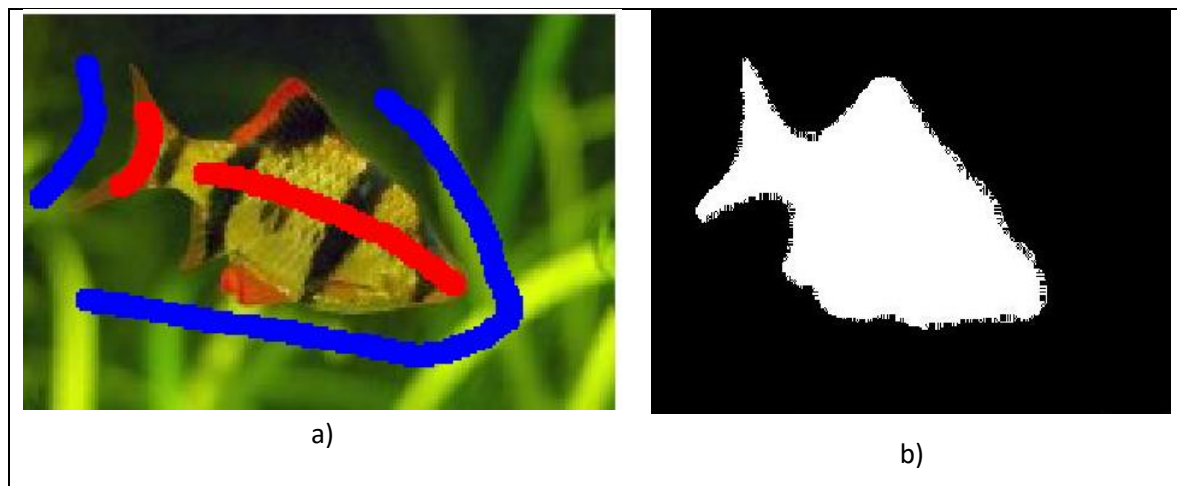


Figura 6-1. Ejemplo de algoritmo de corte de grafos. En la imagen a), se hacen dos garabatos rojos sobre el objeto que se quiere segmentar, y se dibujaron otros garabatos azules sobre el fondo de la imagen, b) es la imagen binaria obtenida después de aplicar el algoritmo de corte de grafos.

Si no se hubiera hecho ningún garabato sobre el fondo de la imagen, se da por hecho que todos los píxeles que no se seleccionan de ninguna manera, pertenecen al fondo.

Este algoritmo se puede implementar de manera que se ejecute iterativamente, cada vez acercándose más a la línea donde se define el borde entre el objeto segmentado y el fondo, para ejecutarlo nuevamente con los parámetros obtenidos en el resultado anterior, con mayor cantidad de píxeles que los de la línea y mejorar la frontera entre el objeto y el fondo. En la figura 6-2 en lugar de los garabatos rojos se toman los píxeles rojos que habían resultado del proceso anterior, y como fondo serán los píxeles marcados en color azul.



Figura 6-2. Método de corte de grafos aplicado recursivamente. En a) se muestran en color rojo los píxeles que pertenecen al objeto y en color azul los pertenecientes al fondo, en b) es nuevamente la imagen original, ahora con el borde más definido.

Se pueden seguir muchos caminos para obtener una mejor segmentación. Pero es necesario tomar en cuenta que al agregar más procesos para mejorar los resultados, aumentan los costos computacionales, y seguramente disminuirá el rendimiento en la ejecución, con lo que la meta de obtener los resultados en tiempo real difícilmente sería alcanzada.

6.4.Conclusión final

Es muy diferente la manera en cómo funciona el cerebro humano en cuanto a la visión, y la manera en cómo las computadoras procesan la información de las imágenes. Se han intentado desarrollar técnicas que copian el funcionamiento del cerebro humano como el uso de las redes neuronales para el reconocimiento de objetos, o el flujo óptico para diferenciar el fondo de los objetos, pero aunque estas técnicas han sido muy trabajadas por años, los resultados no han sido muy alentadores. Sin embargo, en algunas ocasiones cabe mencionar que han tenido resultados sorprendentes.

Es muy importante la elección de un tema que nos apasione, pues no es lo mismo desarrollar un tema que nos atrae y con el cual nos sentimos a gusto trabajando, que la lección de temas que no despierten interés profesional para su investigación y desarrollo. En general en la vida hay que buscar tópicos que nos apasionen y no tópicos que no nos despierten el interés de investigarlos y resolverlos.

Se considera un resultado significativo el que, a partir del análisis espacio-temporal del movimiento de los objetos presentes en las imágenes, se haya podido lograr su caracterización.

Existe un amplio campo de estudio e investigación en el procesamiento de imágenes digitales y la visión por computadora, a simple vista no se detecta algún método computacional, que pudiera empezar a compararse con la habilidad que tiene el cerebro humano para procesar imágenes y obtener información de estas. Hay mucho trabajo por hacer en este campo, la búsqueda efervescente con este fin, sigue en todas direcciones por todo el mundo.

Referencias

- [Ayer y Sawhney, 1995] Serge Ayer y Harpreet S. Sawhney, 1995. "Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding", En IEEE, Conference of computer vision, paginas 777 a 784. IEEE.
- [Canny, 1986] John Canny, 1986. A computational Approach to Edge Detection. Massachusetts Institute of Technology, EE. UU.
- [David Gray, 2001] David Gray, 2001. Introduction to the formal design of real-time systems. Springer, EE. UU.
- [Dengsheng, 2001] Dengsheng Zhang y Guojun Lu, 2001. Segmentation of moving objects in Image Sequence: A Review, Monash University, Australia.
- [Gonzalez y Woods, 2008] Gonzalez y Woods, 2008. *Digital Image Processing 3rd Ed.* Editorial Pearson, USA.
- [James Martin, 1980] James Martin, 1980. Design real-time computer systems, Prentice Hall.
- [Jianobo y Jitendra, 2006] Jianobo Shi y Jitendra Malik, 2006. "Motion Segmentation and Tracking Using Normalized Cuts". University of California, EE.UU.
- [Jong y Hang, 2001] Jong Bae Kim, Hang Joon Kim, 2011. Efficient region-based motion segmentation for a video monitoring system. Elsevier, Korea.
- [Kekre, 2010] Dr. H. B. Kekre, 2010. Image segmentation using extended edge operator for mammographic images. International Journal on Computer Science and Engineering, India.
- [Lee y Krishna, 2010] Yang Lee y C.M. Krishna, 2010. Readings in realtime systems, IEEE Computer Society Press. EE. UU.
- [Liu, 2000] W.S.Liu, 2000. Real-time systems, Prentice Hall. USA.
- [Potter, 1975] Jerry L. Potter, 1975. Velocity as a Cue to Segmentation. IEEE, USA.
- [Raanan et al., 2001] Raanan Fattal, Dani Lischinski, Michael Werman, 2001. Gradient Domain High Dynamic Range Compression. The Hebrew University of Jerusalem. Israel.
- [Roberts, 1965].- L. Roberts, 1965 Machine Perception of 3-D Solids. Optical and Electro-optical Information Processing, MIT Press 1965.
- [Shi y Malik, 1997] J. Shi y J. Malik, 2006. "Normalized Cuts and image segmentation". IEEE Conference Computer Vision and Pattern Recognition páginas 731-737, 1997.

[Soille, 2013] Libro *Morphological Image Analysis: Principles and Applications*, Author: Pierre Soille, 2013, Editorial Springer.

[Spagnolo et al, 2005] P. Spagnolo, T.D' Orazio, M. Leo, A. Distanto, 2005. "Moving object segmentation by background subtraction and temporal analysis", Istituto di studi sui sistemi Intelligenti per l'Automazione, Italia.

[Stalin y Amitabh, 2014] Stalin Alex, Amitabh, 2014. Background subtraction frame difference algorithm for moving object detection and extraction. Journal of Theoretical and Applied Information Technology, International refereed research publishing journal.

[Wen, 2013] Wen Jun-Qin, 2013. Human detection based on Canny edge extraction and frame difference. Zhejiang Technical Institute of Economics, China.

[Yu-Jin Zhang, 2006] Yu-Jing Zhang, 2006. Image Segmentation in the Last 40 Years. Tsinghua University, Beijing, China.

[Zapella et al, 2008] Luca Zapella, Xavier Lladó y Joaquim Salvi, 2008. "Motion segmentation: a review", Instituto de informática y aplicaciones, España.