



---

**INSTITUTO POLITECNICO NACIONAL**  
**CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN**  
**“CIC”**

---

**“Realidad aumentada  
para el  
control de robots móviles”**

*Que para obtener el Grado de:*

**“Maestría en Ciencias de la Computación”**

Presenta:

*ISC. Navarrete Manzanilla Niels Henrik*

Directores:

Dr. Juan Luis Díaz de León  
Santiago

Dr. Juan Carlos Chimal Eguía



México D.F. a 05 de Junio de 2014



**INSTITUTO POLITÉCNICO NACIONAL  
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

**ACTA DE REVISIÓN DE TESIS**

En la Ciudad de     México, D.F.     siendo las     08:00     horas del día     05     del mes de     junio     de     2014     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:

**“Realidad aumentada para el control de robots móviles”**

Presentada por el alumno:

**NAVARRETE**  
Apellido paterno

**MANZANILLA**  
Apellido materno

**NIELS HENRIK**  
Nombre(s)

Con registro: 

A	1	2	0	4	1	5
---	---	---	---	---	---	---

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**  
Directores de Tesis

Dr. Juan Luis Díaz de León Santiago

Dr. Juan Carlos Chimal Eguía

Dr. Miguel Jesús Torres Ruiz

Dr. José Giovanni Guzmán Lugo

M. en C. Germán Téllez Castillo

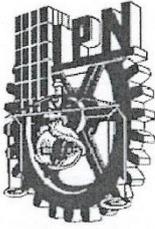
Dr. Amadeo José Argüelles Cruz

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas



INSTITUTO POLITÉCNICO NACIONAL  
SECRETARÍA DE INVESTIGACIÓN  
EN COMPUTACIÓN  
DIRECCIÓN



**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

**CARTA CESIÓN DE DERECHOS**

En la Ciudad de México, D.F. el día 05 del mes de Junio del año 2014, el (la) que suscribe Ing. Niels Henrik Navarrete Manzanilla alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro A120415 adscrito a Centro de Investigación en Computación del IPN, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Juan Luis Díaz de León y Dr. Juan Carlos Chimal Eguía y cede los derechos del trabajo intitulado "Realidad aumentada para el control de robots móviles", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, graficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección ultimatum.of.wisdom@live.com ó fussions evolution@live.com.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

  
NIELS HENRIK NAVARRETE MANZANILLA  
Nombre y Firma.

## **Resumen**

En el presente documento se describe el desarrollo de un sistema de navegación para el control de robots móviles, permitiendo la interacción entre el hombre y la máquina utilizando Realidad Aumentada (AR), el cual tendrá la capacidad de tomar decisiones respecto a las condiciones del entorno que se le presenten. Capaz de guiar y apoyar al conductor en su travesía, permitiendo seguir una ruta de inicio a fin, desplegando información adecuada con la tecnología de AR.

**Palabras clave.**- Algoritmos, Android, Arduino, Control Difuso, Procesamiento de Imágenes, Prototipo, Realidad Aumentada, Sensores ultrasónicos.

## **Abstract**

In this thesis, the development of the navigation system for the control of mobile robots it is described. The system allows a man-machine interaction using augmented reality (AR); it will have the ability to make decisions regarding the conditions of the environment in which it is immersed. It is able to support and guide the driver during his voyage, allowing him to follow a path from start to finish, deploying the appropriate information given by AR technology

**Keywords.**- Algorithms, Android, Arduino, Fuzzy Control, Image Processing, Prototype, Augmented Reality, Ultrasonic Sensors.

## **AGRADECIMIENTOS:**

*Principalmente agradezco a DIOS por haberme dado la oportunidad de realizar una maestría en el Centro de Investigación en Computación CIC-IPN.*

*Permitiéndome realizar el tema de tesis y dándome la inspiración para seguir adelante, cumpliendo con mis objetivos y metas.*

*Agradezco a mi familia por darme su apoyo en cada instante y por la formación que me dieron cada uno.*

*A mi Papá Prudencio Enrique Navarrete R. que continuamente me guía, me brinda su experiencia y hace que me supere cada día.*

*A mi Mamá Herlinda Manzanilla V. que continuamente me apoya y da consejos.*

*A mi Hermano Friedrich Elliott Navarrete M. que me apoya continuamente, me aconseja, me ayuda a crecer y me brinda su tiempo para divertírnos; Sé que será un gran Doctor*

*A mi Tío Oscar Navarrete R. que está en cada momento para apoyarnos*

*A mi Tía Virginia Concepción Navarrete R. y a mi Abuelita Natalia Rodríguez- que en paz descansan. Fueron una guía en mi vida, nunca perdieron la fe y me enseñaron que: “Querer es poder”*

*A la familia Hurtado Ortiz, que brinda continuamente su apoyo y así mismo contribuye con ideas.*

*Agradezco a los directores Dr. Juan Luis Díaz de León y al Dr. Juan Carlos Chimal Eguía, que continuamente procuraron brindarme su experiencia como investigadores y apoyarme para enriquecer la tesis.*

*A los integrantes del Departamento de Prestaciones y Servicios IPN, así como cada uno de los Jefes: Lic. Manuel Salazar, C.P. María de la Luz Magaña y Lic. Maricela Fernández Z.*

*A mis amigos de la Vocacional 11 los “Twisters”, que siempre han tenido el detalle de seguir en contacto.*

*A mis amigos de la Maestría así como a los integrantes del Laboratorio de Simulación y Modelado del CIC-IPN.*

*Dr. Jesús Figueroa Nazuno*

*Eric Ortega Villanueva*

*M. en C. Germán Téllez Castillo*

*Evelia Araceli Santos Camacho*

*Dr. Jesús Martínez Castro*

*Iliac Huerta Trujillo*

*Dr. Miguel Jesús Torres Ruiz*

*Julio César Rangel Reyes*

*Dr. Flavio Arturo Sánchez Garfias*

*Luis Enrique Berthet Dorantes*

*M. en C. Miguel Olvera Aldana*

*Manuel Martínez Álvarez*

*Álvaro Salazar Urbina*

*María Fernanda Chávez*

*Daniel Jiménez Alcántar*

*Rodolfo Navarro Zayas*

*David Jonathan Frías Polanco*

*Thania Mandujano Gallardo*

*Erandi Castillo Montiel*

*Víctor Antonio Ruiz Ibáñez*

*“The Wisdom will save my soul”*

# ÍNDICE:

<b>1. INTRODUCCIÓN.....</b>	<b>2</b>
1.1 <i>Motivación.</i> .....	4
1.2 <i>Objetivo</i> .....	5
1.2.1 <i>Objetivos Específicos</i> .....	5
1.3 <i>Formulación del problema</i> .....	5
1.3.1 <i>Características de la solución</i> .....	6
1.3.2 <i>Alcances</i> .....	6
1.3.3 <i>Limitaciones</i> .....	6
<b>2 ESTADO DEL ARTE .....</b>	<b>8</b>
2.1 <i>AR-Navi</i> .....	10
2.1.1 <i>Funcionamiento</i> .....	11
2.1.2 <i>Problemas Planteados</i> .....	12
2.1.3 <i>Problemas Resueltos</i> .....	12
2.1.4 <i>Conclusiones</i> .....	13
2.2 <i>Interacción y Visualización Aumentada en el Mundo Automotriz.</i> .....	13
2.2.1 <i>Funcionamiento</i> .....	14
2.2.2 <i>Procesamiento de la Información</i> .....	15
2.3 <i>Sistema ARV: En el estudio de la conducción</i> .....	16
2.3.1 <i>Investigación de los Accidentes Automovilísticos</i> .....	17
2.3.2 <i>Funcionamiento de ARV</i> .....	18
2.3.3 <i>Conclusiones</i> .....	21
<b>3 MARCO TEÓRICO.....</b>	<b>24</b>
3.1 <i>Señales de Tránsito</i> .....	24
3.1.1 <i>Señales Verticales</i> .....	25
3.1.2 <i>Señales Horizontales</i> .....	29
3.2 <i>Inicios de la Realidad Aumentada</i> .....	32
3.2.1 <i>Realidad Virtual</i> .....	32
3.2.2 <i>Realidad Aumentada</i> .....	34
3.2.2.1 <i>Funcionamiento</i> .....	36
3.2.2.2 <i>Marcadores AR</i> .....	38
3.2.2.3 <i>Tracking (Rastreo)</i> .....	39

3.3	Sistemas de Realidad Aumentada .....	41
3.3.1	ArToolkit .....	41
3.3.2	Vuforia .....	43
3.3.2.1	Arquitectura .....	45
3.4	Unity 3D .....	47
3.5	Reconocimiento de Patrones a través del Procesamiento de Imágenes .....	48
3.5.1	Extracción de características .....	49
3.5.2	Selección de variables .....	49
3.5.3	Características Basadas en los momentos geométricos .....	50
3.5.4	Interpretación Geométrica de los Momentos Normales .....	51
3.5.5	Momentos Centrales (Invariantes a Traslaciones) .....	52
3.5.6	Invariantes a rotaciones (Los Invariantes de Hu) .....	54
3.6	Arduino.....	56
3.7	Sensores ultrasónicos.....	57
3.7.1	Paquete ultrasónico .....	59
3.7.2	Aplicaciones del sensor ultrasónico. ....	60
3.8	Motores.....	60
3.8.1	Principio de funcionamiento del motor .....	60
3.8.2	Características .....	61
3.9	Servomotor.....	62
3.9.1	Funcionamiento: .....	62
3.10	Motor reductor.....	63
3.10.1	Beneficios al emplear motor reductores.....	63
3.10.2	Características del reductor o motor-reductor .....	63
3.10.3	Características del trabajo a realizar .....	63
3.11	Encoder .....	65
<b>4.1</b>	<b>DISEÑO DEL SISTEMA .....</b>	<b>68</b>
4.1.1	Diagramas UML.....	70
4.1.2	Diagramas de Caso de Uso .....	70
4.1.3	Diagrama de Clase .....	78
4.1.4	Diagramas de Secuencia.....	81
<b>4.2</b>	<b>DESARROLLO Y RESULTADOS.....</b>	<b>83</b>
4.2.1	Realidad Aumentada .....	84

4.2.1.1	NyArToolkit .....	84
4.2.1.2	Processing con NyArToolkit.....	86
4.2.1.3	Vuforia y Android .....	114
4.2.2	<i>Reconocimiento del camino</i> .....	124
4.2.2.1	Movimiento curvilíneo .....	128
4.2.2.2	Radio de curvatura .....	131
4.2.2.3	Orientación.....	133
4.2.2.4	Posición .....	133
4.2.3	<i>Control Móvil</i> .....	136
4.2.3.1	Robot Móvil.....	137
4.2.3.2	Algoritmo de Lógica Difusa. ....	138
5.1	<b>CONCLUSIONES</b> .....	<b>145</b>
5.2	<b>TRABAJO A FUTURO</b> .....	<b>146</b>
6.	<b>REFERENCIAS</b> .....	<b>147</b>

## ÍNDICE DE ALGORITMOS:

Algoritmo 1. ARToolkit .....	85
Algoritmo 2. Configuración de la cámara.....	89
Algoritmo 3. Primera fase de extracción de características del entorno.....	90
Algoritmo 4. Binarización .....	91
Algoritmo 5. Seguidor de contornos y etiquetado simultáneo.....	98
Algoritmo 6. Método SeguidorContorno .....	99
Algoritmo 7. Segunda fase de extracción de características del entorno.....	100
Algoritmo 8. Fases de Desarrollo del Sistema AA-AR.....	118

## ÍNDICE DE DIAGRAMAS:

Diagrama 1. General de sistema AA-AR .....	70
Diagrama 2. Módulo de vinculación del dispositivo móvil.....	72
Diagrama 3. Módulo de Realidad Aumentada .....	74
Diagrama 4. Módulo del Control Móvil.....	76
Diagrama 5. Sistema AA-AR (Clases) .....	79
Diagrama 6. QCAR (Clases).....	80
Diagrama 7. Secuencia del Sistema AA-AR .....	82
Diagrama 8. Componentes de Sistema AA-AR.....	83

## ÍNDICE DE TABLAS:

Tabla 1. Geometría de las Señales de Tránsito .....	28
Tabla 2. Código de colores de las señales de tránsito.....	29
Tabla 3. Dimensiones propuestas para el diseño de un marcador NyArToolkit.....	86
Tabla 4. Descripción de las Bases de Datos Vuforia .....	117
Tabla 5. Entradas y salidas del sistema difuso propuestas .....	141

## ÍNDICE DE FIGURAS:

Figura 1. Experimentos de Sutherland .....	8
Figura 2. Vista AR-Navi .....	11
Figura 3. Resultado AR-Navi .....	13
Figura 4. Ciclo de control convencional .....	14
Figura 5. Sistema ARV.....	17
Figura 6. Componentes de ARV.....	18
Figura 7. Configuración de ARV.....	19
Figura 8. Diagrama de flujo de Datos ARV .....	19
Figura 9. Interacción ARV .....	20
Figura 10. Experimento de manejo con y sin ARV .....	21
Figura 11. Primeras señales de Tránsito.....	24
Figura 12. Señales Multilenguaje .....	24
Figura 13. Señales de Tránsito de Prohibición .....	26
Figura 14. Señales de Tránsito de Restricción.....	26
Figura 15. Señales de Tránsito máximo peligro .....	26
Figura 16. Señales de Tránsito Físicas .....	27
Figura 17. Señales de Tránsito de Información turística y de servicios .....	27
Figura 18. Señales Transitorias.....	28
Figura 19. Líneas segmentadas de separación de circulación opuesta.....	30
Figura 20. Doble Línea continua.....	30
Figura 21. Doble línea mixta.....	30
Figura 22. Zona de no rebasar.....	31
Figura 23. Línea de separación de carriles .....	31
Figura 24. Ejemplos de Realidad Virtual.....	33
Figura 25. Comparación de Realidad Aumentada (AR) vs Realidad Virtual (VR) .....	34
Figura 26. Ejemplos de Realidad Aumentada.....	35
Figura 27. Diagrama de componentes básicos de un Sistema de Realidad Aumentada .....	36
Figura 28. Tipo de Marcadores comunes en AR.....	37
Figura 29. Funcionamiento y Componentes de AR .....	38
Figura 30. Tracking o Seguimiento AR.....	39
Figura 31. Ejemplos de sistemas de seguimiento de tipo encadenamiento.....	40
Figura 32. Vuforia SDK.....	43
Figura 33. Componentes de la plataforma Vuforia de Qualcomm .....	44
Figura 34. Diagrama de flujo de datos del SDK Vuforia en un entorno de aplicación .....	46
Figura 35. SDK Vuforia y Unity 3D .....	47
Figura 36. Placa Arduino Duemilanove .....	57
Figura 37. Comportamiento de los Sensores .....	58
Figura 38. Proyección de los Sensores (Ondas Ultrasónicas).....	59
Figura 39. Ángulo de Propagación.....	59
Figura 40. Servomotor.....	62
Figura 41. Torque de un Motor .....	64
Figura 42. Encoder óptico.....	66

Figura 43. Encoder Magnético .....	66
Figura 44. Marcadores propuestos para el sistema AA-AR.....	87
Figura 45. Señales de tránsito propuestas para el sistema AA-AR.....	88
Figura 46. Imagen Original e Imagen con Umbral.....	91
Figura 47. Vecindad 4 .....	92
Figura 48. Vecindad 8 .....	92
Figura 49. Adyacencia.....	93
Figura 50. Contorno o Frontera.....	94
Figura 51. Imagen con Umbral e Imagen con sus componentes conexas.....	94
Figura 52. Etiquetado de contornos.....	95
Figura 53. Seguidor de contornos y etiquetado.....	97
Figura 54. Extracción de Contornos, bordes y esquinas .....	99
Figura 55. Parámetros intrínsecos.....	102
Figura 56. Distancia focal .....	102
Figura 57. Modelo de transformación de coordenadas.....	103
Figura 58. Componentes conexas y vértices .....	104
Figura 59. Conjunto de máscaras de los marcadores .....	104
Figura 60. Obtención de vértices y componentes conexas.....	105
Figura 61. Extracción de sistema de referencia .....	106
Figura 62. Cálculo de inclinación respecto a los ejes locales .....	106
Figura 63. Triángulo ABC: Lados A, B y C, ángulo $\alpha$ .....	107
Figura 64. Matriz de transformación de la cámara XZ .....	108
Figura 65. Los ejes de la cámara XZ queda, tras la transformación de ejes.....	108
Figura 66. Matriz de transformación de la cámara YZ .....	108
Figura 67. Los ejes de la cámara YZ queda, tras la transformación de ejes.....	109
Figura 68. Sistemas de coordenadas (Marcador, cámara y modelo 2D) .....	112
Figura 69. Sistema de coordenadas OpenGL .....	112
Figura 70. Ejes de proyección del ángulo de la cámara .....	113
Figura 71. Capa intermedia donde se representa objetos virtuales .....	115
Figura 72. Componentes de Vuforia .....	116
Figura 73. Imagen original.....	122
Figura 74. Patrones obtenidos a partir de la imagen original .....	122
Figura 75. Señales de tránsito a escala .....	123
Figura 76. Modelo de color RGB.....	125
Figura 77. Niveles de saturación en el modelo HSV .....	125
Figura 78. Brillo HSV .....	126
Figura 79. Tonalidad HSV .....	127
Figura 80. Modelo HSV .....	127
Figura 81. Vector posición $r$ en un instante $t$ .....	129
Figura 82. Vector velocidad.....	129
Figura 83. Vector aceleración.....	130
Figura 84. Radio de curvatura .....	131
Figura 85. Radio de curvatura $\rho$ de la curva en la posición de abscisa $x$ .....	132
Figura 86. Orientación.....	133
Figura 87. Posición.....	134

Figura 88. Resultado de la detección del camino.....	135
Figura 89. L.A. Zadeh .....	137
Figura 90. Direccionamiento Ackerman .....	137
Figura 91. Estructura general .....	138
Figura 92. Ejemplo de representación de un conjunto difuso .....	140
Figura 93. Función de transferencia para un conjunto difuso triangular.....	141
Figura 94. Función de transferencia para un conjunto difuso trapezoidal .....	141
Figura 95. Control del móvil .....	142
Figura 96. Reglas de Inferencia del Control Móvil .....	142
Figura 97. Base de conocimientos.....	143
Figura 98. Sistema AA-AR y Robot Móvil.....	143

## ÍNDICE DE ECUACIONES:

Ecuación 1. Momentos Geométricos Cartesianos .....	50
Ecuación 2. Momentos Geométricos Cartesianos para una Imagen Digital.....	50
Ecuación 3. Momentos Geométricos Cartesianos para una Imagen Binaria .....	50
Ecuación 4. Función de Momento de Orden Cero (Área) .....	51
Ecuación 5. Función de Momento de Orden Uno (Centro de Masa).....	51
Ecuación 6. Función de Momento de Orden Tres (Sesgo Proyectivo).....	52
Ecuación 7. Momentos Centrales (Invariantes a Traslaciones). .....	52
Ecuación 8. Momentos Centrales $\mu_{PQ}$ .....	52
Ecuación 9. Momentos Centrales x, y .....	52
Ecuación 10. Momentos centrales invariantes a traslaciones (desplazamientos).....	53
Ecuación 11. Combinación .....	53
Ecuación 12. Momentos Centrales utilizando la Combinación.....	53
Ecuación 13. Momentos Centrales de orden 3 .....	53
Ecuación 14 Momentos Centrales de orden 4 y Rotaciones de Hu .....	54
Ecuación 15. Transformación de Coordenadas.....	54
Ecuación 16. Escalamiento de la imagen por cada Eje.....	54
Ecuación 17. Invariantes a Traslaciones, Rotaciones y cambios de escala de Hu.....	55
Ecuación 18. Área de Propagación Ultrasónica.....	59
Ecuación 19. Par Motor o torque .....	62
Ecuación 20. Revoluciones por minuto de un motor .....	64
Ecuación 21. Relación de Engranaje.....	65
Ecuación 22. Diámetro de Paso de un Motor .....	65
Ecuación 23. Valor de pixel .....	103
Ecuación 24. Ángulo de inclinación.....	107
Ecuación 25. Modelo de cámara pinhole.....	110
Ecuación 26. Matriz de la cámara o de parámetros intrínsecos.....	110
Ecuación 27. Modelo de cámara pinhole.....	110
Ecuación 28. Modelo de cámara pinhole.....	110
Ecuación 29. Homografía .....	111
Ecuación 30. Matriz de Rotación y vector de translación .....	111

Ecuación 31. Transformación RGB a HSV .....	126
Ecuación 32. Aceleración Media .....	130
Ecuación 33. Aceleración $a$ en un instante .....	130
Ecuación 34. Movimiento curvilíneo en el plano XY .....	130
Ecuación 35. Ecuación del radio de curvatura $\rho$ .....	131
Ecuación 36. Radio de curvatura.....	132
Ecuación 37. Ecuaciones para Encoder .....	134
Ecuación 38. Orientación del Robot móvil.....	134
Ecuación 39. Conjunto difuso en un universo en discurso .....	140

# **CAPÍTULO I**

## **INTRODUCCION**

## 1. INTRODUCCIÓN.

Durante el siglo XXI, los avances tecnológicos han estado en constante crecimiento y produciendo resultados innovadores en nuestra vida diaria. Permitiendo interactuar con sistemas, capaces de envolver los sentidos y dotar de operatividad a los usuarios.

Esta área de investigación es la Realidad Virtual, que consiste en brindar una experiencia al usuario, dentro de un mundo virtual donde interactúan sus sentidos (*especialmente su visión y su audición*), este ambiente es generado y diseñado por la computadora, separándolo del mundo real; Lo que implica darle un cierto nivel de profundidad dentro de este mundo artificial, para que el usuario pueda percibirlo, como lo hace en su entorno.

Pero el inconveniente de esta tecnología son sus altos costos, tanto para el software como el hardware, ya que deben ser capaces de reproducir este ambiente artificial; los investigadores se dieron la tarea de buscar los medios para ofrecer esta tecnología y difundirla, sin necesidad de crear un software o hardware tan especializado para esta interacción.

Con el tiempo, la tecnológica de las comunicaciones se ha vuelto parte de nuestra vida diaria, con lo que surgió los llamados: dispositivos móviles.

Los dispositivos móviles se pueden categorizar de acuerdo a sus niveles de funcionalidad pero, solamente mencionaremos los más habituales tales como:

- Celulares
- Tablets
- Agendas Personales



Todos han favorecido a la sociedad, como para comunicarnos o brindar funcionalidad multimedia, como lo hizo y lo ha hecho el Internet. Esta revolución ha iniciado, a ofrecer nuevas capacidades como las que nos brinda una computadora:

- Navegación por Internet
- Reproducir contenido Multimedia (Música, Video, etc.)
- Realizar Conferencias
- Enviar mensajes de texto o de correo electrónico
- Agendar citas, eventos, reuniones etc.
- Ejecutar Aplicaciones



Al combinar ambos desarrollos, la ciencia brinda una herramienta para presentar, la interacción del usuario con los elementos virtuales dentro de un mismo ambiente, esta técnica se conoce como Realidad Aumentada (Augmented Reality) o Realidad Inmersiva– que se refiere a la posibilidad de insertar en el mundo real, representaciones tridimensionales de objetos virtuales (artificiales) que pueden contener:

- Textos
- Imágenes
- Videos
- Sonido

La realidad aumentada se basa en una metáfora de interfaz tangible en donde objetos físicos son usados para manipular objetos virtuales de una manera intuitiva, de modo que una persona, que no tenga el conocimiento pueda interactuar con ella, obteniendo una experiencia enriquecida con su uso.



## 1.1 Motivación.

En la historia de la humanidad el hombre siempre procura mejorar sus condiciones de vida y todas estas orientadas a satisfacer sus necesidades. Podemos mencionar la pirámide de Maslow, que realiza la clasificación mediante cinco niveles:

- Auto-realización
- Auto-aceptación.
- Necesidades Sociales
- Necesidades de Seguridad y Protección.
- Necesidades Básicas.

Maslow aborda de otra forma la problemática de lo que es auto-realización, hablando de las necesidades impulsivas esto se traduce, lo que necesita el hombre para ser feliz.

Partiendo de esta premisa, la ciencia se ha vuelto el medio para satisfacer estas necesidades; una de ellas fue la creación de medios de transporte, que facilita el traslado de un lugar a otro con lo que surge la industria automovilística, enfocada en dar seguridad y comodidad durante el viaje a los pasajeros. Aprovechando el constante avance tecnológico, se propuso como meta equipar al usuario con medios que pueda interactuar con el automóvil.

Algunas de estas tendencias que se han desarrollado son:

- **Multimedia:**  
Ofrecen la combinación de: animaciones, video, música, manos libres, etc.  
Cuya finalidad es brindarle al automovilista un ambiente de confort.
- **Sistemas de Geolocalización:**  
Cuyas funciones principales son:
  - Trazar Rutas
  - Mostrar información de calles y avenidas
  - Visualizar mapas
  - Establecer destinos
- **Simuladores de manejo:**  
Diseñados para evaluar a las personas, que tengan el deseo de aprender a manejar
- **Sistemas de evaluación y rendimiento:**  
Enfocados en determinar el estado del automóvil.

Los sistemas mencionados y los existentes solo muestran información, y no son capaces de interactuar con el entorno. Por lo que se plantea la propuesta de crear un sistema de control y navegación autónomo que obtenga información en tiempo real, permitiendo la interacción hombre/máquina de manera natural mediante realidad aumentada, esto beneficiara a los usuarios de automóviles; Proporcionando asistencia vial durante el trayecto, mediante el apoyo de representaciones virtuales dentro del mundo real.

## **1.2 Objetivo**

Diseñar y desarrollar un sistema de navegación para robots móviles basado en realidad aumentada, que permita la interacción entre el hombre y la máquina de una forma natural; mediante la toma de decisiones. El modelo desarrollado se implementara en un ambiente urbano simulado.

### **1.2.1 Objetivos Específicos**

- Desarrollar un modelo de interfaz hombre maquina basado en Realidad Aumentada
- Diseñar un modelo de navegación autónomo basado en Realidad Aumentada
- Diseñar un esquema de integración entre los dos modelos citados anteriormente
- Implementar algoritmos de Reconocimiento de Patrones, para el uso eficiente de modelos de Realidad Aumentada y del análisis del camino
- Implementar sensores que nos brinden información óptima para medir distancias.
- Diseñar un control móvil y una etapa de potencia para el robot móvil
- Diseñar y Desarrollar un prototipo móvil que esquite objetos.

## **1.3 Formulación del problema**

Crear un sistema de control y navegación autónomo que permita una interacción natural hombre/maquina, facilitando la asistencia en todo momento.

Procesar la información adquirida, desde el entorno y además representar información útil y eficiente de acuerdo a la señal de tránsito; mediante modelos de imágenes virtuales representados en un ambiente real, esto favorece agilizar la conducción, disminuir accidentes y reducir distracciones.

### **1.3.1 Características de la solución**

Desarrollar un modelo de interfaz hombre/máquina basado en Realidad Aumentada, esto implica la creación de un sistema de control y navegación autónomo, además del diseño de un prototipo móvil con la capacidad de esquivar objetos que se le presenten, en un entorno propuesto.

Para lograrlo, es necesario:

1. Diseñar un entorno urbano
2. Determinar las señales de tránsito a analizar
3. Realizar un procesamiento del escenario
4. Creación de modelos de realidad aumentada

### **1.3.2 Alcances**

- Diseño y desarrollo de un modelo de interfaz hombre/máquina basado en AR
- Diseño de un modelo de navegación autónomo.
- Integración de los modelos.
- Implementación de algoritmos de Reconocimiento de Patrones
- Diseño y construcción del móvil
- Diseño e implementación de algoritmos, capaces de discernir, las decisiones del usuario y del entorno, para desplazar adecuadamente el robot móvil
- Asistir y guiar constantemente al conductor
- Diseño y desarrollo de un sistema multitarea

### **1.3.3 Limitaciones**

- El tipo y las condiciones de terreno limitarán el acceso del móvil debido a su diseño (superficies planas y no pedregosas).
- Las condiciones ambientales (luz)
- Se consideran algunas de las señales de tránsito (propuestas)
- La distancia de las señales de tránsito con respecto a la cámara deben ser visibles
- La capacidad de almacenamiento y el procesamiento del dispositivo móvil
- La cantidad simultánea de señales de tránsito a procesar estas varían de acuerdo a la biblioteca utilizada

# **CAPÍTULO II**

## **ESTADO DEL ARTE**

## 2 ESTADO DEL ARTE

Para conocer el surgimiento de la realidad aumentada es adecuado retroceder un poco en el tiempo, aproximadamente en 1960, cuando Ivan E. Sutherland (Sutherland I. E.) usó un dispositivo que despliega imágenes tridimensionales, su diseño era parecido a un casco (helmet), para visualizar los gráficos.



Figura 1. Experimentos de Sutherland

Sutherland, realiza una definición sobre la realidad virtual:

*A display connected to a digital computer gives us a chance to gain familiarity with concepts not realizable in the physical world. It is a looking glass into a mathematical wonderland* (Sutherland I. , 1964)

Una pantalla conectada a una computadora que nos da la capacidad de familiarizarnos con conceptos no viables en el mundo físico. Esto es mirar a través de unos lentes un paraíso matemático.

Con el tiempo surgieron acontecimientos importantes tales como:

- 1962: Morton Heilig (Director de fotografía)  
Crea un simulador de moto llamado: “Sensorama”, usaba imágenes, sonido, vibración y olfato.
- 1973: Ivan Sutherland  
Inventa el display de cabeza (HMD) lo que sugiere una ventana a un mundo virtual (Knowlton, 1964).
- 1985: Hurto Myron Krueger  
Crea Videoplace que permite a los usuarios interactuar con objetos virtuales por primera vez.
- 1990: Jaron Lanier  
Acuña el término realidad virtual y crea la primera actividad comercial en torno a los mundos virtuales.

- 1992: Tom Caudell  
Crea el término Realidad Aumentada (Mizell, 1992)  
(Personal guidance system for the visually impaired using GPS, GIS and Vr Technologies, 1993).
- 1994: Steven Feiner, Blair MacIntyre y Doree Seligmann.  
Desarrollan el primer sistema de Realidad Aumentada en un prototipo, “KARMA”, presentado en la conferencia de la interfaz gráfica. Ampliamente citada en la publicación Communications of the ACM al siguiente año.
- 1999: Hirokazu Kato desarrolla ArToolkit en el HitLab y se presenta en SIGGRAPH ese año (Kato H. , Augmented Reality).
- 2000: Bruce H. Thomas  
Desarrolla ARQuake, el primero juego al aire libre con dispositivos móviles de Realidad Aumentada, y se presenta en el International Symposium on Wearable Computers.
- 2008: AR Wikitude  
Es una guía que sale a la venta el 20 de octubre de 2008 con el teléfono Android G1.
- 2009: AR Toolkit.  
Soporta a Adobe Flash (FLArToolkit) por Saqoosha, con lo que la realidad aumentada llega al navegador Web.
- 2009: Logo oficial de la Realidad Aumentada  
Con el fin de estandarizar la identificación de la tecnología aplicada en cualquier soporte o medio por parte del público general.
- 2012: Google  
Diseña lentes que crean la primera realidad aumentada comercializada. Se bautiza a su proyecto como “Project Glass”

Con el tiempo, la realidad aumentada, ha tenido varios enfoques, como:

- Educación
- Animación
- Ventas
- Simuladores
- Juegos
- Ciencia

El enfoque ahora será para la navegación y control de Robots Móviles, por lo que se citaran aquellos sistemas parecidos a lo que se pretende desarrollar.

Estos reflejan la tendencia de dicha tecnología y quizá lo que acontecerá en el futuro.

## 2.1 AR-Navi

### Sistema de Navegación en Vehículos usando Realidad Aumentada

Desde hace unos años se han vendido aproximadamente 24 millones sistemas de navegación de vehículos conocidos como sistemas de geolocalización GPS; producidos en Japón, estos sistemas se han vuelto populares con el paso del tiempo.

Sus funciones básicas de estos sistemas son:

- Mostrar un mapa
- Planificar rutas
- Localización

Lo que fueron lo suficientemente poderosa por un cierto tiempo, ya que las necesidades van cambiando y esto implica generar retos para mejorar este tipo de sistema, con lo que los fabricantes se dieron la tarea de mejorar la navegación en el vehículo respecto a su diseño de la interfaz de usuario. Por ejemplo, incorporar nuevas funciones, tales como pantallas que muestran un paisaje de la ciudad usando gráficos en 3D (CG) o la combinación de imágenes fotografías aéreas y mapas existentes.

Desafortunadamente, la utilidad de estas funciones es limitada debido a los siguientes problemas.

- Los datos y fotografías aéreas 3D utilizadas fueron capturas hace tiempo con lo que generan discrepancias entre los datos y el estado actual.
- Los datos existentes se crearon muestreando el mundo real hasta cierto grado limitado de precisión.

Estos datos se procesan de la manera siguiente la fotografía 3D se le extraen su número de polígonos de acuerdo a su resolución, luego se procesa con lo que puede generar deformaciones, esto puede resultar no representar fielmente lo que el conductor realmente ve.

Debido a estos problemas, el conductor toma decisiones incorrectas, por ejemplo dar vueltas erróneas, si el conductor no es experimentado le ocasiona perder tiempo en encontrar cual es la ruta más adecuada. Ya que solo se basa en la visualización de puras imágenes.

Con esta idea surge AR-Navi, un sistema que utiliza la captura de una cámara para hacer uso de la realidad aumentada.

### 2.1.1 Funcionamiento

AR-Navi es un sistema de navegación para vehículos, que guía al conductor mostrándole la orientación, mediante objetos virtuales capturando lo que hay enfrente de él.



Figura 2. Vista AR-Navi

Donde se notifica el cambio de dirección para anticipar la vuelta y dándole la ubicación a su vez mediante gráficos 3D (CG).

Otras investigaciones basadas en los vehículos que utilizan sistemas de navegación aumentada. Sawada (Sawano, 2004) describen la extracción de la información del tipo de geometría de las carreteras y una manera de poder dibujarlas en tiempo real mediante realidad aumentada. Hu (Zhencheng, 2004) (Japanese Ministry of Land, 2007) (Quddus, 2003) (Guideline for In-vehicle Display Systems Virsion 3.0) describe la extracción de futuros puntos a lo largo de un camino y la manera de ajustar la localización entre la orientación (dirección) dibujando los caminos en tiempo real. Estos estudios han demostrado que la viabilidad de la alta precisión de alineación entre el entorno real y las imagen virtuales se puede realizar.

La tecnología de análisis de imagen que se utiliza en estos sistemas, sin embargo, tiene algunas limitaciones. En primer lugar, el tiempo de complejidad en la tecnología es grande. En segundo lugar, la tecnología no es robusta con respecto a las condiciones ambientales, tales como la lluvia, la oscuridad y luz de fondo. Así, estos sistemas no son utilizados actualmente en la navegación de sistemas.

El objetivo AR-Navi es desarrollar un sistema comercial que sólo necesite una cámara como equipo adicional y que sea robusto con respecto a dichas condiciones ambientales

### 2.1.2 Problemas Planteados

AR-Navi busco de una manera superar los problemas que afectan a los sistemas anteriores:

- **Adecuado Registró:** Los conductores se sienten incómodos cuando encuentran desajustes entre la visualización de la imagen y el entorno real.
- **Control de la Información:** Mucha información de orientación puede llevar al conductor no prestar atención en condiciones de tráfico.
- **Eliminación de imágenes inapropiadas:** Las imágenes capturadas son a veces no adecuadas ya que la cámara no las muestra correctamente.

### 2.1.3 Problemas Resueltos

- **Adecuado Registró:**  
En primer lugar, los sistemas estiman su propia ubicación y dirección combinando los datos del sistema de posicionamiento global (GPS) y datos de navegación, como los sensores de giro y velocidad. En segundo lugar, se utiliza un método de coincidencia de mapa que compara el resultado del primer paso con el camino actual. El sistema corrige la posición del vehículo y cambia los registros de conducción de manera significativa (tal como cuando el vehículo cambia de dirección) o existen diferencias entre el registro del conductor y los caminos. Inmediatamente después de un giro del vehículo, la ubicación y la dirección pueden mostrar una alta precisión pero a veces ocurren errores enormes si el vehículo pierde en un largo tiempo la señal GPS.

AR-Navi propone las siguientes soluciones en relación entre ubicación y dirección:

- Espacio entre Atrás y adelante
  - Espacio entre de lado derecho e izquierdo
  - La vuelta del vehículo
- **Control de la Información:**  
Reducir la cantidad de información y solo mostrar la más representativa para que se tenga conocimiento y poder generar correctas decisiones, Ar-Navi busca llamar la atención del conductor para que visualice la información

- **Eliminación de imágenes inapropiadas:**  
Ajustar la cámara tomando en cuenta las condiciones ambientales.

### 2.1.4 Conclusiones

AR-Navi, es un sistema de navegación, que permite guiar visualmente al conductor, resuelve algunos de los problemas que surgen al mezclar objetos virtuales con el mundo real, propone un reconocimiento de cruces peatonales y filtrado de imágenes basado en valores evaluados, construye un prototipo que es usado para mostrar los métodos mencionados, logrando la exactitud de registro e intersecciones para guiar al conductor.



Figura 3. Resultado AR-Navi

## 2.2 Interacción y Visualización Aumentada en el Mundo Automotriz.

La industria automotriz surgió a principio de 1900. Principalmente se dedicada:

- Diseño
- Desarrollo
- Fabricación
- Ensamblaje
- Comercialización
- Reparación

Durante varios años ha sido la generadora de empleos y hoy en día cada hogar cuenta con al menos un automóvil, volviéndose una necesidad para el ser humano, permitiendo viajar a grandes distancia como: ir al trabajo, visitar a familiares, transportar víveres, etc.

Ha llegado el punto de que la industria busca estrategias enfocadas a la interacción y visualización innovadora. Brindando información del estado de los vehículos para mayor accesibilidad y reducir el trabajo al conductor; este nuevo enfoque se llama: interacción y visualización aumentada.

A principios del siglo XXI, se han propuesto e implementado técnicas para realizar una interacción adecuada con el hombre y el automóvil, surgiendo problemas en la comunicación debido al inadecuado planteamiento de combinar la información visualizada en la pantalla y la obtención de los datos, utilizando dispositivos de entrada

A grandes rasgos se describa las aplicaciones existentes dentro del mundo automovilístico.

### 2.2.1 Funcionamiento

La interacción del conductor respecto al vehículo se describe como un ciclo de control convencional presentado en la siguiente figura.

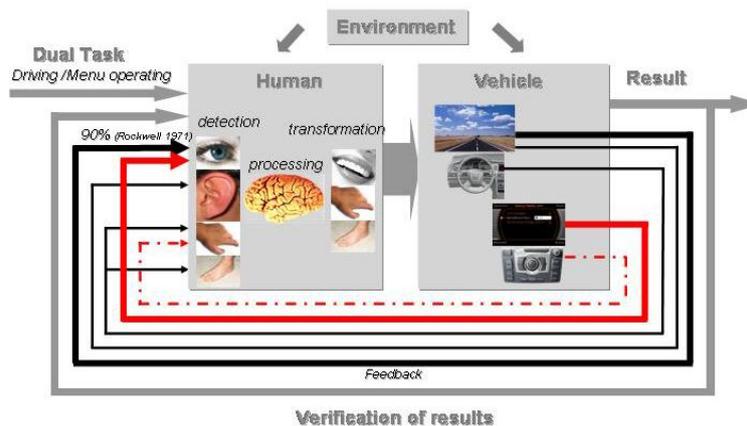


Figura 4. Ciclo de control convencional

De lado izquierdo se visualiza el procedimiento que funge como la entrada del bucle donde sus componentes son: el conductor y el vehículo equipado con tecnología de realidad aumentada.

Las entradas permiten la generación de información y los datos obtenidos siempre están en constante seguimiento para que el usuario pueda ajustarlos si el considera conveniente.

Especialmente en el caso de los sistemas que tienen interacción con un menú se hace ineficiente su uso, debido a dos problemáticas:

- Conducir el automóvil
- Interactuar con el menú

Provocando distracciones, molestias y falta de atención al momento de conducir.

Los sistemas de menú ofrecen tareas las cuales se dividen en:

- **Primarias:**  
Consiste en los requerimientos básicos del manejo.  
Enfocados en la navegación, dirección y estabilidad. Seleccionando la ruta desde un punto de partida y un destino
- **Secundarias:**  
No están enfocadas en el control y seguimiento del carro, solo es utilizada para dar indicaciones de señalamientos.
- **Terciarias:**  
Dirigidas únicamente a los ajustes de temperatura, aire acondicionado, o características de comunicación y entretenimiento.

Sin embargo en un vehículo ocurren situaciones en que el conductor debe interactuar con varios sistemas al mismo tiempo y estos deben tener una interfaz intuitiva sin tantos distractores.

### **2.2.2 Procesamiento de la Información**

Durante el diseño del sistema, es importante identificar los cuellos de botella y la sobrecarga. El humano como operador es una parte central entre la interacción humano/máquina, la corrección de los problemas de carga de trabajo es necesaria para la operación segura y eficiente.

La Teoría de Recursos Múltiples de Wickens (Wickens C. &, 1983) permite predecir cuándo las tareas interfieren unas con otras o poder realizarlas en forma paralela. Si la dificultad de una tarea aumenta una pérdida de rendimiento de otra tarea será el resultado.

Wickens (Wickens, 1984) describe varios recursos de procesamiento. Todas las tareas se pueden dividir de la siguiente forma: codificación, procesamiento central y respuesta al escenario.

Los componentes visuales y auditivos son modalidades de entrada. El componente central de proceso describe el nivel de procesamiento de la información requerida.

El componente de la respuesta consiste en acciones manuales o vocal. Wickens (Wickens., 1988) postula el uso de múltiples canales de información para aumentar la respuesta del conductor, reduciendo distracciones.

## 2.3 Sistema ARV: En el estudio de la conducción

La tecnología AR puede ofrecer un entorno muy realista para incrementar el desempeño del conductor y además enfocado para realizar pruebas a diferentes usuarios que permita estudiar sus habilidades y reacciones en diversos escenarios en el tráfico

Esto se puede lograr mediante la adición de objetos virtuales (personas, vehículos, peligros y otros objetos) a la vista mientras se conduce en un entorno real.

En este trabajo se explora un nuevo sistema de realidad aumentada para vehículo (ARV) que facilite aplicar un nuevo concepto de seleccionar el tipo de tráfico los cuales son:

- Maniobrar a la izquierda/Derecha
- Intersecciones
- Dos vías en diferentes sentidos

ARV (Ghada Moussa, 2012), puede ser instalado en cualquier carro donde el conductor podrá ver a su alrededor a través de HMD (Casco Montable que Muestra Información) objetos virtuales que son generados a través de la computadora y se añaden a la escena.

Esta diversidad de objetos que se presentan, se inicia en una diversidad de escenarios y los resultados de este estudio se sustentan en ser un sistema que brinde apoyo en la realización de pruebas en el aspecto de seguridad y operación del transporte.

Donde un estudio reveló que los participantes (conductores) se sienten más seguros en dar una vuelta con gran espacio dentro de un rango de 4.0-9.0 seg. Esto implica que todos los espacios menores de 4 seg. son rechazados y todo espacio de más de 9 seg. es probablemente aceptado. El valor medio del tiempo de giro a la izquierda era 4,67 seg., que es un poco más alto que los valores reportados en los informes (4,0 a 4,3 s).

Los conductores mayores seleccionan espacios más grandes para darse la vuelta que los conductores jóvenes. Indicando que la capacidad de maniobrar se reduce con la edad; Muchos de los participantes indicaron en las pruebas, que se ofrece un buen nivel de confort y no perciben ningún riesgo mientras manejan con el sistema ARV, ninguno presentó algún tipo de mareo

Una gran parte de la sensación de conducción depende de la interacción del conductor con los componentes del vehículo (volante, frenos y el acelerador) para asegurar el realismo de los resultados.



**Figura 5. Sistema ARV**

### **2.3.1 Investigación de los Accidentes Automovilísticos**

En los Estados Unidos, más del 30% de todos los accidentes de tránsito ocurren en las intersecciones y anualmente son aproximadamente 9000 muertes que son alrededor del 25% de todas las fatalidades. Casi el 70% de las muertes se produjeron por no existir el señalamiento. Generando una preocupación importante en la seguridad dentro de las intersecciones, existen otros accidentes del tipo de dar la vuelta en U de lado izquierdo que representa el 27% y alrededor del 30% de los accidentes la vuelta en U de lado derecho puede ser debido a calcular incorrectamente. El accidente al girar a la izquierda se considera como uno de los accidentes más peligrosos de intersección.

Los estudios han demostrado que las características de los conductores (edad y sexo) son considerados entre los más importantes factores que afectan el comportamiento de dar vuelta, otros factores incluyen distracciones del conductor, la velocidad, el tiempo de espera, la hora del día, y las características de intersección.

Los conductores mayores tienen problemas con la percepción, la cognición y la atención que pueden afectar a su capacidad para detectar adecuadamente, percibir y precisar su evaluación de su seguridad al dar vuelta.

Considerando los conductores de edad avanzada (40-64 años) y a los conductores más jóvenes (18-29 años). En un estudio de simulación de conducción se realizaron pruebas para evaluar el tipo de vueltas a tres grupos de edad (20-30, 31-55 y 56-83), las influencias significativas en edad y género fueron encontrados. Indicaron que el grupo más viejo su tendencia es girar en espacios más grandes que los de los otros dos grupos, en cuestión de género se encontró que las mujeres prefieren espacios más grandes en comparación con la de los hombres.

### **2.3.2 Funcionamiento de ARV**

El Sistema ARV está constituido por tres componentes principales conectados a una computadora, los cuales son:

- Una video Cámara:
- Un sistema de posicionamiento Global (GPS)
- Casco Montable que Muestra Información (HMD)



**Figura 6. Componentes de ARV**

Donde la cámara de vídeo está fijada en el parabrisas del vehículo y el conductor lleva el HMD mientras conduce el vehículo.

A través de la HMD, el conductor ve la realidad aumentada la que se combina con el camino en tiempo real (desde la cámara de vídeo) y virtuales (generadas por ordenador) imágenes de los vehículos, señales de tráfico, intersecciones, edificios, árboles y otros objetos en función del escenario que se esté implementando.

La cámara esta fija alineada a la misma altura de los ojos del conductor, como se muestra en la figura. Para dar una perspectiva al conductor lo que percibe en cada instante. Ofrece al conductor una visión amplia que permite a él / ella manejar fácilmente y con seguridad. Combinando el video real con los objetos virtuales de modo que el conductor no será capaz de diferencia entre lo real y lo virtual



Figura 7. Configuración de ARV

El flujo de los datos del sistema ARV se presenta con el siguiente diagrama

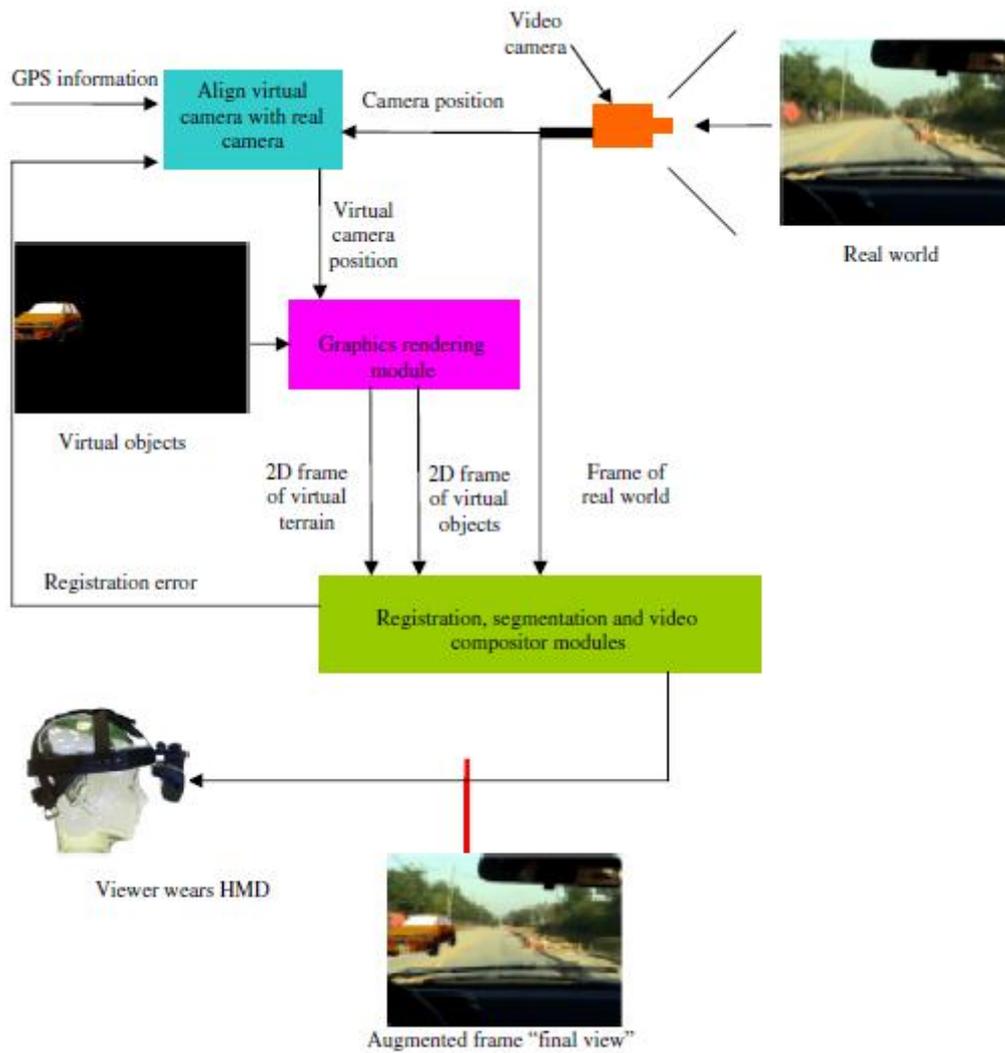


Figura 8. Diagrama de flujo de Datos ARV

La cámara captura en tiempo real, el GPS está conectado al vehículo para proporcionar la ubicación y la dirección mientras se mueve.

Existen parámetros intrínsecos de la cámara (distancia focal, punto principal, coeficiente de asimetría y distorsión de parámetros de coeficiente) y extrínsecos (rotaciones y traslaciones) que deben ser calculados.

Para el cálculo de la cámara de los parámetros intrínsecos, la cámara se calibra desconectándola. Para el cálculo de parámetros extrínsecos, se necesitan dos pasos:

1. Calcular la orientación de la cámara:  
La cámara se calibra en línea utilizando el método checkboard orientación respecto a su plano de referencia
2. La traducción de la cámara se calcula utilizando la información GPS respecto al carro. Ambos parámetros intrínsecos y extrínsecos se utilizan para alinear cámara virtual con la cámara real.

La representación gráfica utiliza la posición de la cámara virtual para hacer fotogramas en 2-D, cada uno contiene los objetos virtuales y el otro contiene el terreno virtual.

Se calcula el error generado en situar el objeto virtual en el entorno mediante una transformación afín. Este error de registro se utiliza para reajustar la cámara virtual. Luego, el módulo de segmentación de los píxeles que pertenecen al vehículo de prueba, el módulo de composición de vídeo combina el marco real y el marco de objetos virtuales, para generar el marco aumentada “vista final”. Por último, la vista final se muestra delante de los ojos del espectador a través del HMD.



**Figura 9. Interacción ARV**

### 2.3.3 Conclusiones

El sistema ARV maneja la inconsistencia visual de la escala, la orientación y/o errores en la alineación de los objetos visuales con objetos reales. Se aprecian visualmente los objetos sin ningún tipo de inconsistencia visual, los objetos virtuales (vehículos y antiniebla) y los objetos reales (por carretera y árboles) están bien alineados indicando la capacidad del sistema para duplicar una escena real, libre de cualquier problema visual.

Los participantes interactuaron con dos escenarios (sin el sistema de ARV, y con el sistema ARV) se les pidió que conducirán a una velocidad de 40 kph y lo más centrado posible en su carril, en tramos rectos como en curvos. Además, se pidió detener el vehículo en la señal de (stop).

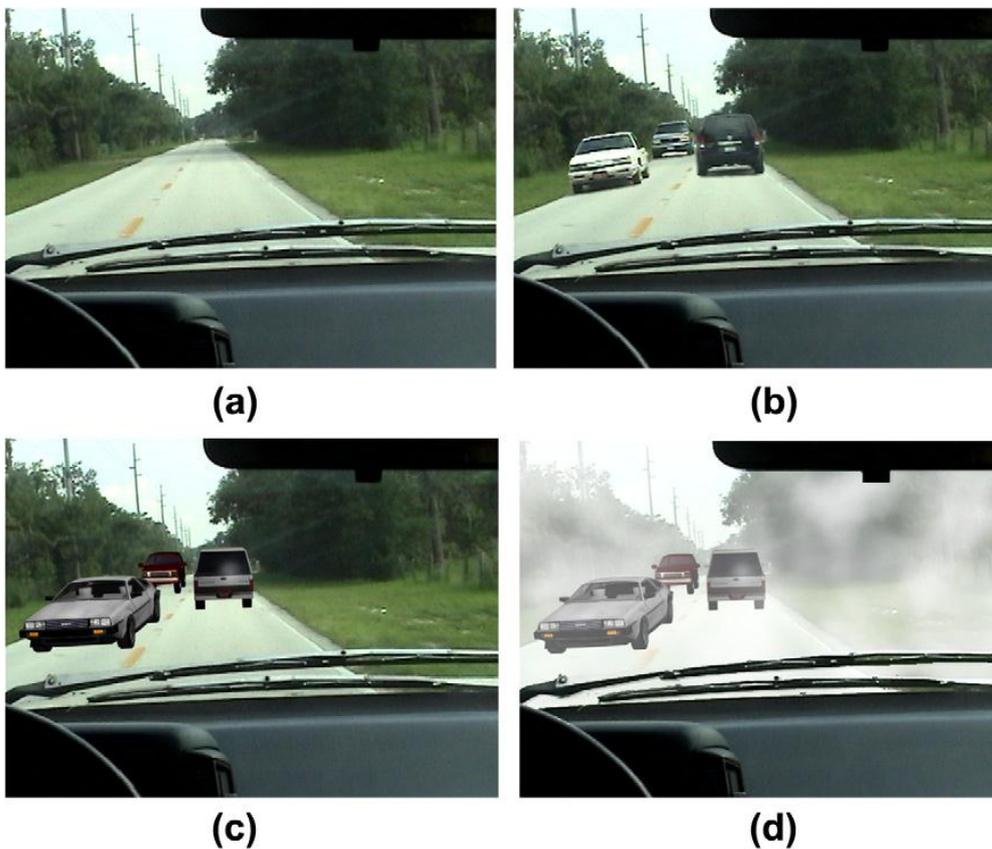


Figura 10. Experimento de manejo con y sin ARV

Las imágenes presentadas son capturadas durante el experimento, donde las dos primeras imágenes ilustran la no utilización del sistema ARV, y las siguientes con el sistema ARV, en una de ellas se establece un escenario con niebla

En el nivel de confianza de 95%, resultados estadísticos indicaron que no hubo diferencias significativas en todos los parámetros evaluados en los dos escenarios cuando se conduce en una recta. Por otra parte, no hubo diferencias significativas en la velocidad media y el desplazamiento desde el centro de la calle entre los dos escenarios cuando se conduce a lo largo de la curva. En base a estos resultados se puede concluir que el juicio de los

conductores en cuestión a distancia no cambió de forma significativa bajo la conducción con el sistema de ARV. En adicional, cambiaba la velocidad de acuerdo a su nivel de comodidad no cambiaba de forma significativa

Esto puede ser explicado que los conductores basan sus decisiones por lo que ven a través de la HMD, en la conducción en curvas se requiere una visión más amplia debido a la naturaleza de la curva y su ancho de giro por los menos 4,57 m. Por lo tanto, cuando los conductores daban la vuelta no contaban con visión amplia ya que la cámara esta fija. Esto condujo a que redujeran su velocidad al conducir con el sistema ARV.

# **CAPÍTULO II**

## **MARCO TEÓRICO**

### 3 MARCO TEÓRICO

#### 3.1 Señales de Tránsito

El hombre constantemente se enfrenta a la necesidad de identificar espacios para guiarse en su entorno, con lo que incorporar símbolos cuyo objetivo es el de comunicar.

En la Edad Media las primeras señales (Enciclografia, s.f.) Se usaron para marcar territorios y direcciones de ciudades estas señales se construían en las intersecciones de vías para señalar rutas y conocer distancias, comúnmente eran señales multidireccionales.



Figura 11. Primeras señales de Tránsito

La primera red internacional de caminos estructurados y organizados fueron las calzadas romanas. Por lo tanto hubo un tiempo en que todas las señales apuntaban a Roma; el primer hito llamado *Milliarium Aureum*, marco el inicio de la extensa red, es precisamente que los romanos son los primeros en establecer un código de señales para orientar al viajero en el tránsito de las calzadas. Estos hitos romanos se llaman: *Millarios* y contaban en millas romanas, la distancia desde ese punto a los diferentes destinos que comunicaba la vía.

La mayoría de las señalizaciones fueron definidas en Roma, el 11 de octubre de 1908, en el primer Congreso Internacional de Tránsito Vial.

La señalización escrita planteaba dificultades para quienes no sabían leer o para personas extranjeras, Por lo que las señales multilinguaje no siempre fue una solución.



Figura 12. Señales Multilinguaje

Para solucionarlo se implementan pictogramas, imágenes de cruces, etc., en sustitución de textos, al principios de 1902 en Francia, 1903 en Italia, 1907 en Alemania.

Para superar la gran cantidad de señales y sobre todo las barreras lingüísticas se desarrollaron sistemas de códigos basados en acuerdos convencionales. Con lo que surge la Convención Internacional de Ginebra de 1909, la cual estandariza a nivel internacional 4 tipos de señales: Giro, Cruce, Cassis y Paso a nivel.

El uso de símbolos reconocidos internacionalmente ha ayudado a mejorar la comprensión de los mensajes. Con el desarrollo de la circulación de automóviles las señales viales se hicieron cada vez más necesarias. En forma general, cualquier señal de tránsito, se conoce como un dispositivo de control de tránsito que debe cumplir los siguientes requisitos:

- Satisfacer efectivamente una necesidad
- Ser visible y llamar la atención
- Contener significado claro y simple
- Inspirar Respeto
- Colocarse de modo que brinde el tiempo adecuado para una respuesta oportuna.

### 3.1.1 Señales Verticales

En la actualidad existe una gran variedad de señales de tránsito, por lo que es conveniente dar a conocer su clasificación (Operadores, s.f.) (Pagaza).

- **Reglamentarias o prescriptivas.**

Estas se subdividen a su vez en:

- a. Prohibición: Determina acción que no puede realizarse
- b. Restricción: Indican los límites de la velocidad, peso y tamaño de los vehículos, uso de estacionamiento y carriles.
- c. Prioridad: Cambian la prioridad respecto a quién corresponde pasar primero en una esquina o tramo del camino



Figura 13. Señales de Tránsito de Prohibición



Figura 14. Señales de Tránsito de Restricción

- **Señales Preventivas.**

Se agrupan en:

- a. Máximo peligro: Indican que hay que conducirse con extrema precaución sobre determinados lugares porque el peligro que hay alrededor es muy grande



Figura 15. Señales de Tránsito máximo peligro

b. Físicas: Simboliza determinadas características de la ruta.



Figura 16. Señales de Tránsito Físicas

- **Señales Informativas**

Se agrupan en:

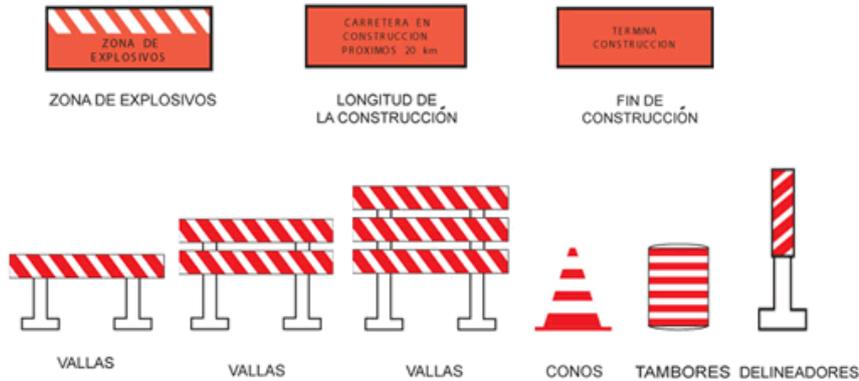
- a) Nomenclatura urbana: Informan el nombre y la altura de las calles o avenidas; identifican la ruta en las zonas rurales, además muestran la distancia que falta para llegar a una localidad o destino.
- b) Características de la vía: Indica características de las rutas o calles.
- c) Información turística y de servicios: Brindan información para los conductores y peatones, como la cercanía de una estación de servicio: teléfono, restaurante, primeros auxilios.



Figura 17. Señales de Tránsito de Información turística y de servicios

- **Señales Transitorias**

Advierten sobre mantenimiento y construcción en la vía.



**Figura 18. Señales Transitorias**

### Diseño de Señales Verticales

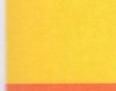
El diseño facilita la identificación por parte del usuario que se encuentra en la vialidad. Por lo que se estandariza el uso de la forma, color y mensaje.

	El octógono se usa exclusivamente para la señal de PARE.
	El triángulo equilátero con un vértice hacia abajo se usa exclusivamente para la señal de CEDA EL PASO.
	El rectángulo con el eje mayor vertical se usa generalmente para señales regulatorias.
	El círculo se usa para señales regulatorias asociadas con la seguridad peatonal, banderines de mano y la señal de posición en los cruces de ferrocarril.
	El rombo se usa para señales preventivas.
	La cruz diagonal blanca se reserva exclusivamente para indicar la posición de un cruce de ferrocarril a nivel.
	El triángulo equilátero con un vértice hacia arriba se usa para ciertas señales preventivas.
	El rectángulo con el eje mayor horizontal se usa para señales de información y guía; señales para obras en las vías y propósitos especiales, así como placas complementarias para señales regulatorias y preventivas.
	El escudo se usa para señalar las rutas.

**Tabla 1. Geometría de las Señales de Tránsito**

## Colores.

Cada señal se distingue con un color el cual debe cumplir con las especificaciones de normas.

	Se usa como fondo en las señales de PARE, en paletas y banderas de PARE, en señales especiales de peligro y señales de entrada a un cruce de ferrocarril; como color de borde en señales de CEDA EL PASO y triángulo preventiva etc.
	Se usa como color de leyenda para las señales que tienen fondo blanco, amarillo, o naranja, en marcas de peligro y en señales de direccionamiento de vías.
	Se usa como color de fondo para la mayoría de señales regulatorias, delineadores de rutas, nomenclatura de calles y señales informativas; y, en las señales que tienen fondo verde, azul, negro, rojo o café, como un color de leyendas, símbolos como flechas y orlas.
	Se usa como color de fondo para señales preventivas, señales complementarias de velocidad, distancias y leyendas, además en señales especiales delineadoras.
	Debe ser fluorescente y se usa como color de fondo para señales de trabajos temporales en las vías y para banderolas en CRUCES DE NIÑOS.
	Tipo I: Se usa como color de fondo para la mayoría de las señales informativas de destino y peajes.
	Tipo II: Se usa como color de fondo para las señales informativas de destino sobre las autopistas.
	Se usa como color de fondo para las señales informativas de servicio y señales de ciclo vías; también, como color de leyenda y orla en señales direccionales de las mismas.
	Se usa como color de fondo para señales informativas turísticas y ambientales.

**Tabla 2. Código de colores de las señales de tránsito**

Se emplean para regular la circulación, advertir o guiar a los usuarios de la vía, por lo que constituyen un elemento indispensable para la seguridad y la gestión de tránsito.

Son blancas, amarillas y azules. Estos colores deben ser uniformes a lo largo de la señalización y bajo los siguientes conceptos básicos:

- **Líneas amarillas:**
  - Separación de tráfico viajando en direcciones opuestas.
  - Restricciones.
  - Descanso
- **Líneas blancas:**
  - Separación de flujos de tráfico en la misma dirección.
  - Zona de estacionamiento
- **Líneas azules:**
  - Zonas tarifadas de estacionamiento con límite de tiempo.

### Tipos de Líneas en la vía (Federal, s.f.).

- **Líneas segmentadas de separación de circulación opuesta.**  
Son de color amarillo y pueden ser cruzadas siempre y cuando haya seguridad



Figura 19. Líneas segmentadas de separación de circulación opuesta

- **Doble Línea continua (línea de barrera).**  
Las líneas de separación de carriles de circulación opuesta continuas dobles, consisten en dos líneas amarillas paralelas



Figura 20. Doble Línea continua

- **Doble Línea Mixta**  
Consisten en dos líneas amarillas paralelas, una continua y otra segmentada. Los vehículos, siempre que exista seguridad, pueden cruzar desde la línea segmentada para realizar rebasamientos.



Figura 21. Doble línea mixta

- **Zona de no rebasar**

Dado que la maniobra de rebasamiento es la de mayor riesgo al conducir, las zonas de no rebasar son definidas cuidadosamente conforme a los criterios especificados a continuación:

Se justifica donde la distancia de visibilidad de rebasar es menor que la distancia de rebasamiento mínimo.

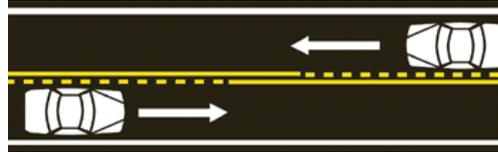


Figura 22. Zona de no rebasar

- **Línea de separación de carriles.**

Ordenan el tráfico y posibilitan un uso más seguro y eficiente de las vías, especialmente en zonas congestionadas.



Figura 23. Línea de separación de carriles

## **3.2 Inicios de la Realidad Aumentada**

La realidad Aumentada es una innovación en el campo de la visión por computadora, permitiendo realizar una combinación adecuada con el mundo real donde se crea un nuevo tipo de dato representado en el frame de la imagen capturado por la cámara.

Esta área de la computación es relativamente reciente y ha sido objeto de consideración en el campo de la investigación e innovación.

Lo novedoso de dicha tecnología es que se integra perfectamente al mundo real, permitiendo naturalmente incrementar la experiencia con la computadora, estamos obligados a plantearnos la siguiente pregunta

¿Cómo surgió la Realidad Aumentada?

Necesitaremos hablar brevemente un poco de su historia, para tener una referencia sobre ella. En primera instancia es recomendable explicar sobre Realidad Virtual (Virtual Reality) puesto que es una extensión de la Realidad Aumentada.

### **3.2.1 Realidad Virtual**

La Realidad Virtual, es un sistema que permite sumergir al usuario en un ambiente tridimensional diseñado por una o varias computadoras, de forma interactiva y autónoma en tiempo real, es decir es un ambiente simulado en el cual el usuario puede usar sus sentidos.

En esta definición se rescatan puntos importantes tales como:

1. El usuario se encuentra inmerso en un mundo virtual (No existe interacción con el mundo físico).
2. Es una instancia de un modelo diseñado mediante software y hardware
3. Existe un cierto nivel de profundidad
4. El usuario con sus sentidos sensoriales puede en algún momento percibirlo como el mundo físico
5. El mundo virtual debe ser sensible a los tipos de respuestas por el usuario

Si alguno de estos elementos no se cumple, el usuario solo percibe que se está reproduciendo una película en la computadora.



Figura 24. Ejemplos de Realidad Virtual

### 3.2.2 Realidad Aumentada

La Realidad Aumentada (Augmented Reality) –también conocida como Realidad Inmersiva– se refiere a la posibilidad de insertar en el mundo real representaciones tridimensionales de objetos, información en textos, imágenes, videos y sonidos

A diferencia de la realidad virtual, las interfaces de realidad aumentada, permiten al usuario mirar al mundo real al mismo tiempo que el contenido virtual generado por la computadora de modo que se “aumenta” la experiencia del mundo real, creando nuevos horizontes de desarrollo en la investigación

Esto nos lleva por encima de los límites que ya estamos acostumbrados, por ejemplo en del desarrollo de aplicaciones de escritorio o de web donde el usuario, introduce datos e interacciona en ella.

Existe una íntima relación entre los objetos virtuales y los objetos reales, estos últimos pueden ser “aumentados” de manera que pueden dar información dinámica en la superficie de los objetos, darle un contexto visual al objeto, generar interacciones físicas. La realidad aumentada se basa en una metáfora de interfaz tangible en donde objetos físicos son usados para manipular objetos virtuales de una manera intuitiva, de modo que una persona sin el conocimiento del manejo de una computadora puede tener una experiencia enriquecida.

La realidad aumentada toma lo que observa de la realidad inherente y lo enriquece con aplicaciones impactantes. No requiere tecnología sofisticada solo un simple impreso en un *flyer*, en una postal o en un póster es suficiente para hacer surgir animaciones tridimensionales en la computadora de cualquier usuario. Lo único que se requiere es una cámara web.

La comparación de Realidad Aumentada con la Realidad Virtual se puede representar gráficamente de esta manera.

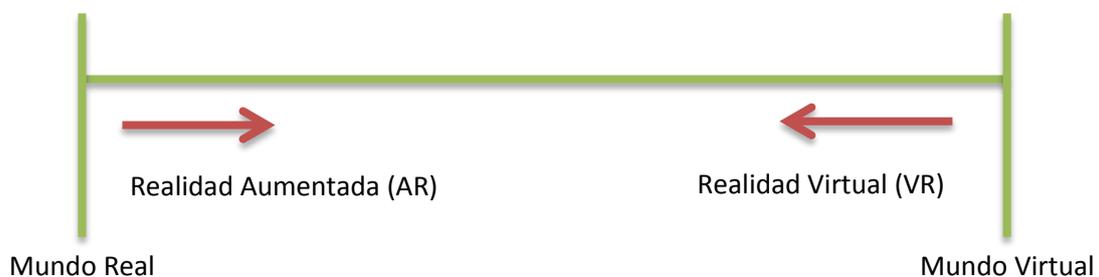


Figura 25. Comparación de Realidad Aumentada (AR) vs Realidad Virtual (VR)



Figura 26. Ejemplos de Realidad Aumentada

### 3.2.2.1 Funcionamiento

Sistema de realidad aumentada básico (IGD) (Vallino), da un enfoque general de los elementos y procedimientos que implican.

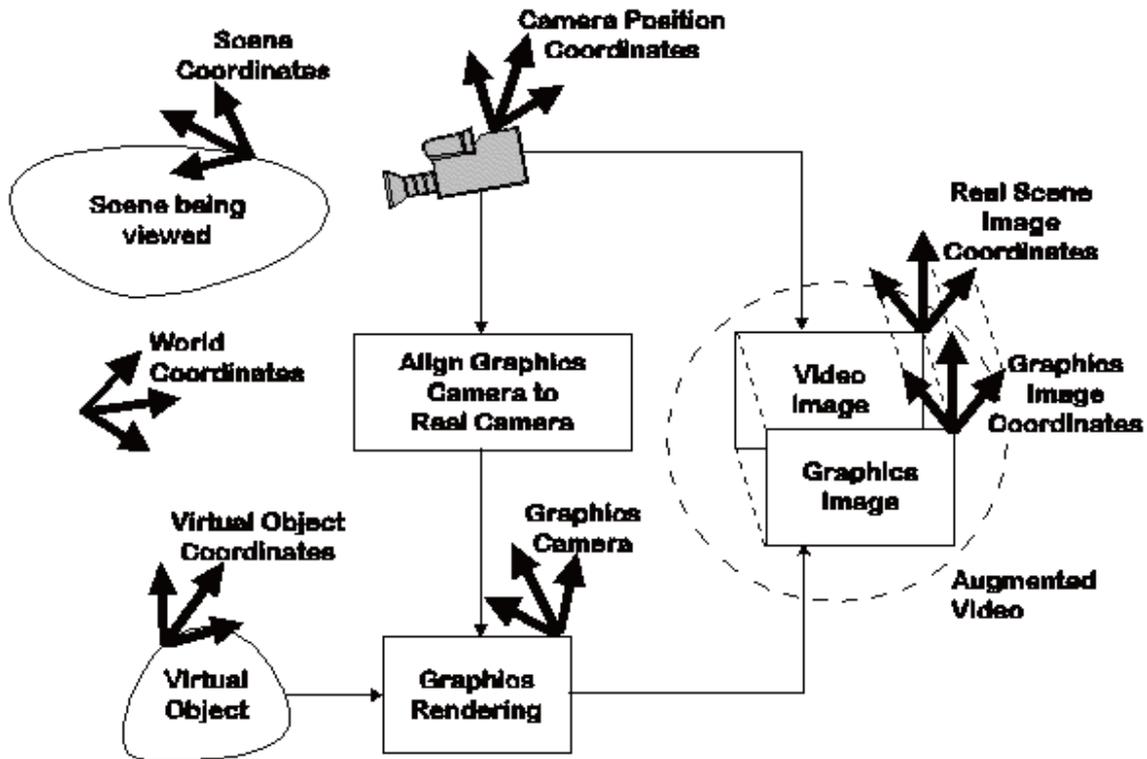


Figura 27. Diagrama de componentes básicos de un Sistema de Realidad Aumentada

El elemento inherente es la cámara que va a dotar de visión, si no existe este elemento tenemos un sistema de realidad virtual.

La cámara tiene una cierta configuración para desempeñar un cierto grado de exactitud, para sobreponer los elementos virtuales en los objetos físicos también involucra la factibilidad de encontrar los marcadores.

La cámara envía información del entorno en tiempo real y el sistema debe de analizar cada frame de la cámara y aplicar por lo menos dos fases:

1. Algoritmos de Filtrado
2. Algoritmos de Extracción de características

Las fases permiten discernir lo que encuentra y lo que se está buscando en el frame actual.

¿Cómo discierne lo que se busca?

Primero debemos saber qué es lo que busca.

## Marcador:

Se puede definir como un símbolo o conjuntos de símbolos plasmados en un medio impreso, sobre algún objeto determinado. Si el diseño es muy simple no posee muchas características, el entrenamiento de la detección de patrones sería muy escasa e induce a la existencia de falsos positivos, por consecuencia no se detecte adecuadamente

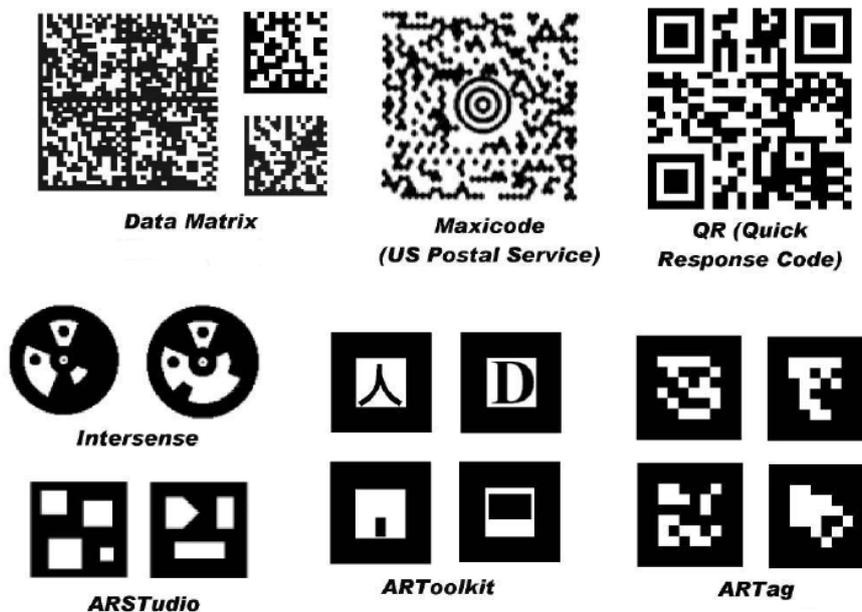


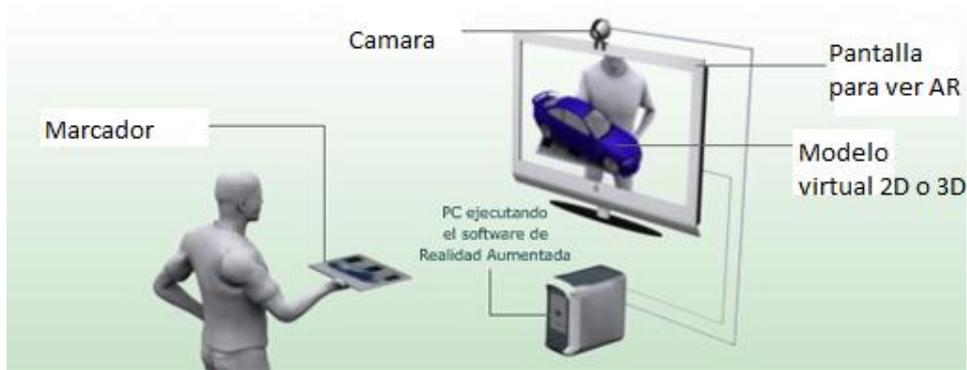
Figura 28. Tipo de Marcadores comunes en AR

En el instante en que se procesa el frame actual, se realizan varias tareas consecutivamente tales como:

1. Hacer una consulta en la base de datos de entrenamiento
2. Calcular las coordenadas de la escena donde se encuentra el marcador
3. Calcular las coordenadas de la posición de la cámara respecto a la escena
4. Aplicar transformadas afines
5. Adquirir los patrones de los marcadores
6. Clasificar los marcadores respecto a la base de entrenamiento

Estos pasos se ejecutarán indefinidamente hasta que la aplicación finalice.

Cuando coincide el patrón con los datos de entrenamiento, ya se pueden representar los elementos virtuales y estos deben identificarse con el marcador encontrado, la calidad del objeto virtual radica en los métodos de renderizado empleados (representar o generar una imagen, con un cierto nivel de detalle).



**Figura 29. Funcionamiento y Componentes de AR**

### 3.2.2.2 Marcadores AR

Destacamos que la visión de la cámara es indispensable para lograr la realidad aumentada, sus funciones principales son:

1. Seguimiento de la escena(Track)
2. Colocación de los objetos virtuales

Muchas aplicaciones, usan una señal o indicador único, que proporcionan información en donde se generara la RA, dentro de la escena.

Los marcadores se pueden definir como:

- Un conjunto de LED's (montados en un casco)
- Un matriz de puntos
- Un conjunto de palabras
- Una imagen.

Por uso práctico y por la sencillez de diseño se utilizan marcadores 2D parecido al QR basados en cuadros blancos y negros, cada uno representa un patrón a clasificar.

No todos los sistemas que implementan realidad aumentada usan marcadores como referencia para la visualización de los objetos virtuales en el mundo real.

### 3.2.2.3 Tracking (Rastreo)

Cuando se crea una aplicación en AR, existen por lo menos tres requisitos importantes:

1. Diseño de los Marcadores
2. Patrones Clasificados
3. Diseño de modelos virtuales

El tracking es el término usado para referirse, al constante seguimiento o monitoreo de la escena, en busca del Target (Marcador).

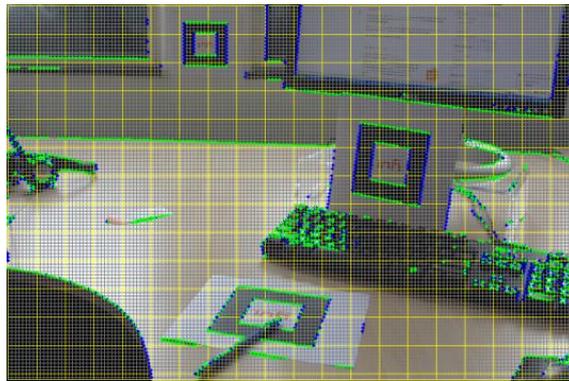


Figura 30. Tracking o Seguimiento AR

Si los objetos virtuales aparecen deformados o en planos equivocados, este efecto se le conoce como: *Localización del rastreo (Location tracking)* o *Simple Rastreo (Simply Tracking)*, el primer caso se produce por la mala transformación de coordenadas del marcador respecto a la cámara y el segundo caso por la mala clasificación de los patrones.

Por lo tanto, los marcadores deben contener información suficiente, para clasificar sus patrones, en la etapa de entrenamiento. La desventaja radica en que se necesita cubrir una cierta área del marcador

Se pueden usar marcadores (landmarks) naturales como:

- Estatuas
- Paisajes
- Caminos etc.

Algunas aplicaciones no usan marcadores específicos, sino imágenes precargadas acorde al escenario, aún siguen en modo experimental.

- **GPS**  
El enfoque de usar el Sistema de Posicionamiento Global en combinación con las aplicaciones AR. Sigue siendo considerada un gran campo de estudio, debido a la exactitud que debe tener los métodos de Tracking para determinar la posición de los marcadores, respecto a la zona geográfica en donde se encuentre.

Sin embargo el uso del GPS, como un sistema de tracking tiene ciertos inconvenientes tales como:

- a. Grado de precisión en entornos cerrados
- b. Perdida de comunicación temporal

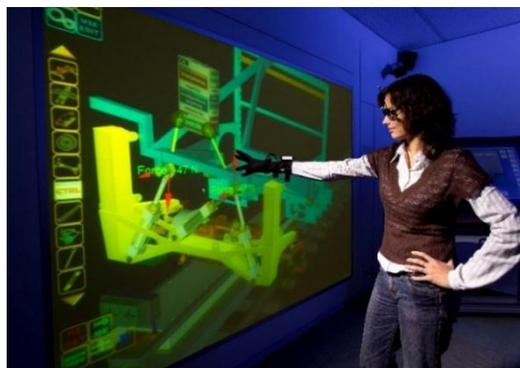
- **SISTEMAS DE SEGUIMIENTO DE TIPO ENCADENAMIENTO.**

Está diseñado para funcionar exclusivamente en áreas cerradas y requiere instalaciones que estén equipadas con sensores. El seguimiento funcionara con: sensores magnéticos, ultrasonido o de forma óptica.

Tales sistemas son muy precisos e inclusive alcanzan rangos de centímetros hasta milímetros.

La gran desventaja de este tipo de sistemas es que se requiere tener instalaciones equipadas con las tecnologías adecuadas antes de poder utilizar AR.

Su área de aplicación es de investigación, por ejemplo si se desea, visualizar el modelo de una partícula o realizar alguna geo-localización o modelar datos a partir de un conjunto de ellos.



**Figura 31. Ejemplos de sistemas de seguimiento de tipo encadenamiento**

### 3.3 Sistemas de Realidad Aumentada

La realidad aumentada surge como una extensión de la realidad virtual, era muy desconocido su funcionamiento debido a su estrecha relación con la misma, solo pocos tenían la herramienta para crearla debido al alto costo computacional y al diseño de hardware requerido.

En 1999 el Dr. Hirokazu Kato (Kato H. , Biography, 2009) crea ArToolkit y su desarrollo sigue siendo apoyado por el laboratorio de Tecnología de Interfaz Humana (HIT Lab) (Kato D. H., ARToolkit) :

- HIT Lab Wa Universidad de Washington.
- HIT Lab Nz Universidad de Canterbury, Nueva Zelanda
- Artoolworks Inc en Seattle.

#### 3.3.1 ArToolkit

Es una biblioteca (Kato D. H., Artoolkit) que brinda funciones esenciales para la creación de la realidad aumentada.

Una de las dificultades del desarrollo de la RA, es el problema del seguimiento de la vista del usuario, es decir, saber en qué punto se va a representar los objetos virtuales.

Se utilizan algoritmos de visión por computadora para resolver el problema y se calcula:

- la posición de la cámara real
- la orientación relativa a los marcadores físicos en tiempo real.

Esta biblioteca es considerada en ser la primera en brindar a los usuarios la capacidad de crear aplicaciones. Pero tiene ciertas limitantes tales como:

- La iluminación
- Algoritmos de visión poco robustos
- Diseño de Marcadores con poca información.

Los sistemas actuales que reproducen RA utilizan variantes de ArToolkit tales como:

- **Atomic Authoring Tool**  
Es un software multi-plataforma, el cual es un Front-end de ArToolkit. Está dirigido para aquellos que no son programadores, permitiendo crear rápidamente, pequeñas y sencillas aplicaciones. Tiene Licencia GNU GPL
- **Atomic Web Authoring Tool**  
Es la extensión de Atomic Authoring tool, para los sitios web, Es un front end para la biblioteca FIARToolkit. Tiene Licencia GNU GPL
- **OSGART**  
Combinación de ArToolkit y OpenSceneGraph

- **ArTag**  
Alternativa de ArToolkit, que usa un procesamiento de imagen y de símbolos digitales, más complejo, para mayor fiabilidad y resistencia a la luz. Con licencia para fines no- comerciales.
- **ArToolkitPlus**  
Versión extendida de ArToolkit, recomendada solo para usuarios experimentados y desarrolladores de software RA. No cuenta con soporte, ya que se discontinuo
- **Studierstube\_Tracker**  
Sucesor de ArToolkitPlus
- **FIArToolkit**  
Es un port en ActionScript 3 de ArToolkit para flash 9 o superior
- **NyArToolkit**  
Biblioteca de clase ArToolkit, enfocada para máquinas virtuales, particularmente aquellas con host Java, C# y Android
- **ARDesktop**  
Biblioteca de clase de ArToolkit que crea interfaces de escritorio con controles y widgets.

Aproximadamente en el año 2000 los dispositivos móviles empezaron a tener un impacto tecnológico, cuando las compañías de telecomunicaciones dejaron de lanzar su propio sistema operativo, esto beneficio a crear estándares de hardware e impulsando a las empresas de hardware diseñar y crear procesadores de alto desempeño que realicen operaciones de punto flotante de 10MFlops y procesadores gráficos dedicados.

Esencialmente cualquier dispositivo móvil, incluye los siguientes elementos:

- Cámara
- Sensor GPS.
- Sensor de Acelerómetro
- Sensor de Campo Magnético

Depende del fabricante pueden existir diferentes sensores (Developers, s.f.), empleando el software para combinar sus características y así construir nuevos sensores.

### 3.3.2 Vuforia

Vuforia es una Plataforma creada y desarrollada por Qualcomm (Qualcomm, Qualcomm, s.f.), enfocada a brindar una nueva dimensión y experiencia en los dispositivos móviles para el uso de la realidad aumentada.

Ofrece una avanzada tecnología y rendimiento a una amplia gama de dispositivos móviles.

Su característica es reconocer una variedad de objetos 2D o 3D (Targets), las plataformas que soporta es:

- IOS
- Android
- Unity 3D

Permitiendo escribir en una sola aplicación todo el procedimiento de la Realidad Aumentada.

Vuforia evita realizar un escaneo de tipo códigos QR, para acceder a la información codificada, en lugar de esto se basa en imágenes, y estas son creadas y diseñadas por el propio usuario a diferencia las de código QR o similares que las genera un software.

Las imágenes están enriquecidas ya que contienen un conjunto de patrones que posteriormente serán analizados y procesados por la cámara del dispositivo móvil.

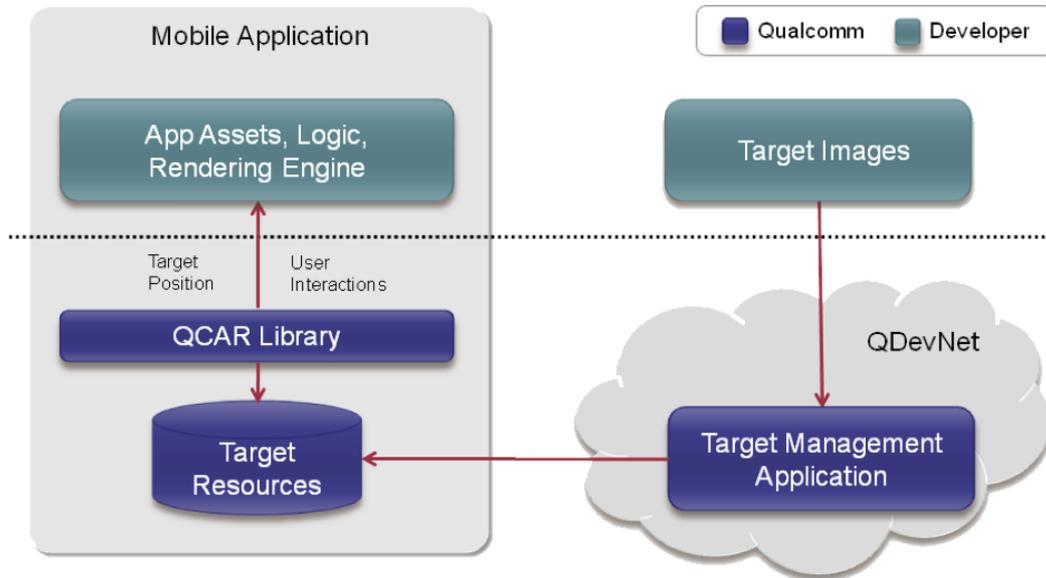
Vuforia requiere una base de datos para almacenar sus imágenes (targets), que se almacenaran localmente en el dispositivo que ejecute el software. Para visualizar los objetos virtuales que van a interactuar con los objetos reales.

En forma general el SDK de Vuforia (Qualcomm Connected Experiences, s.f.), utiliza la pantalla del dispositivo móvil, como una ventana donde coexiste nuestro mundo y el mundo de la realidad virtual, utilizando la cámara, para renderizar el objeto virtual que se va a mostrar o interactuar en el mundo real, estos se superponen en la vista de la cámara y aparecen acoplados en el mundo real.



Figura 32. Vuforia SDK

Se ilustra de manera general el desarrollo con la plataforma Vuforia (Qualcomm, Vuforia Developer, s.f.).



**Figura 33. Componentes de la plataforma Vuforia de Qualcomm**

La plataforma consiste en el SDK Vuforia y el Sistema de Gestión de Target alojada en el portal de desarrolladores QDevNet (Qualcomm, Vuforia Developer, s.f.).

## PROCEDIMIENTO GENERAL

### 1. TARGETS IMAGES

Se elige una imagen de entrada que va ser aquella que nos permitirá representar el objeto virtual dentro del escenario (mundo real)

### 2. TARGET MANAGEMENT APPLICATION

La imagen que es elegida debe ser subida a QDevNet para poder procesarla.

-QDevNet (Qualcomm Developer Network):

Fue creado por los desarrolladores de Qualcomm, para crear una comunidad interesada en desarrollar aplicaciones dentro de los procesadores Qualcomm o utilizar alguna de las API'S. Su enfoque es poder centralizar el acceso a todo lo que se necesita para construir sus aplicaciones, permitiendo descubrir, desarrollar, distribuir y descargar algún recurso o almacenar sus propias creaciones.

### 3. TARGET RESOURCES

Es una pequeña base de datos personal donde se van almacenando cada una de nuestras imágenes asociadas a nuestros objetos virtuales creados, esto nos permite poder acceder en cualquier momento a ellas.

### 4. QCAR LIBRARY

Es una biblioteca de Vuforia que permite enlazar la imagen tracking con las interacciones del usuario y la posición del target. En Android la librería estática es (libQCAR.so) y en iOS (libQCAR.a).

## 3.3.2.1 Arquitectura

La arquitectura de Vuforia SDK (Qualcomm, Vuforia Developer, s.f.) de Realidad Aumentada se compone de los siguientes elementos:

- **CAMARA**  
La cámara se utiliza para asegurar que todo lo que se pre-visualiza se ha capturado, es fundamental que permitiendo ser eficiente el tracking. Los desarrolladores solo tendrán la tarea de decirle a la cámara en que momento empiece a capturar y cuando terminar de capturar. Los frames de la cámara se calibran de acuerdo al formato y el tamaño que maneje el dispositivo móvil.
- **CONVERTIDOR DE IMAGEN**  
Permite convertir los pixeles del objeto virtual, al formato de la imagen de la cámara, para su representación y renderizar la imagen mediante OpenGL ES y para el seguimiento (por ejemplo, al iluminar un objeto)
- **TRACKER**  
Contiene los algoritmos para poder detectar y rastrear los objetos del mundo real mediante los fotogramas (frames) de la cámara. Dichos algoritmos se basan en la imagen de la cámara, donde se detectan los objetos y se realizan las evaluaciones que detecta al pulsar un botón virtual. Los resultados son almacenados en el estado del objeto que es utilizado. Pueden existir multi-imágenes pero solo uno de ellos puede ser activado al mismo tiempo.
- **RENDERIZAR LA VISTA DE LA CAMARA**  
La imagen capturada por la cámara se renderiza, la cual es almacenada en el estado del objeto. Para incrementar y optimizar el procesamiento de la imagen.
- **CÓDIGO DE APLICACIÓN**  
Es la parte donde los desarrolladores deben inicializar todos los componentes y realizar tres pasos importantes en el proceso de la Aplicación. Por cada proceso del frame, el objeto de estado se debe actualizar y la aplicación del código debe realizar una llamada al método de renderizar.

- i. Consultar el estado del Objeto por cada nuevo target, marcador o actualización del estado de estos elementos.
- ii. Actualizar la aplicación lógica con los nuevos datos de entrada.
- iii. Renderizar los gráficos de la realidad aumentada realizando la superposición de gráficos

• **TARGET RESOURCES**

Es el lugar donde se almacenan y gestionan los Targets que se crean por el desarrollador y el acceso de estos datos se obtiene mediante la base de datos que nos brinda QDevNet. Al descargar estos recursos (Targets), dentro de su contenido se encuentra un archivo de configuración XML, que permite realizar ciertos cambios a la configuración básica y un archivo binario que tiene los datos del Tracking todo esto se compila dentro de la plataforma Vuforia SDK.

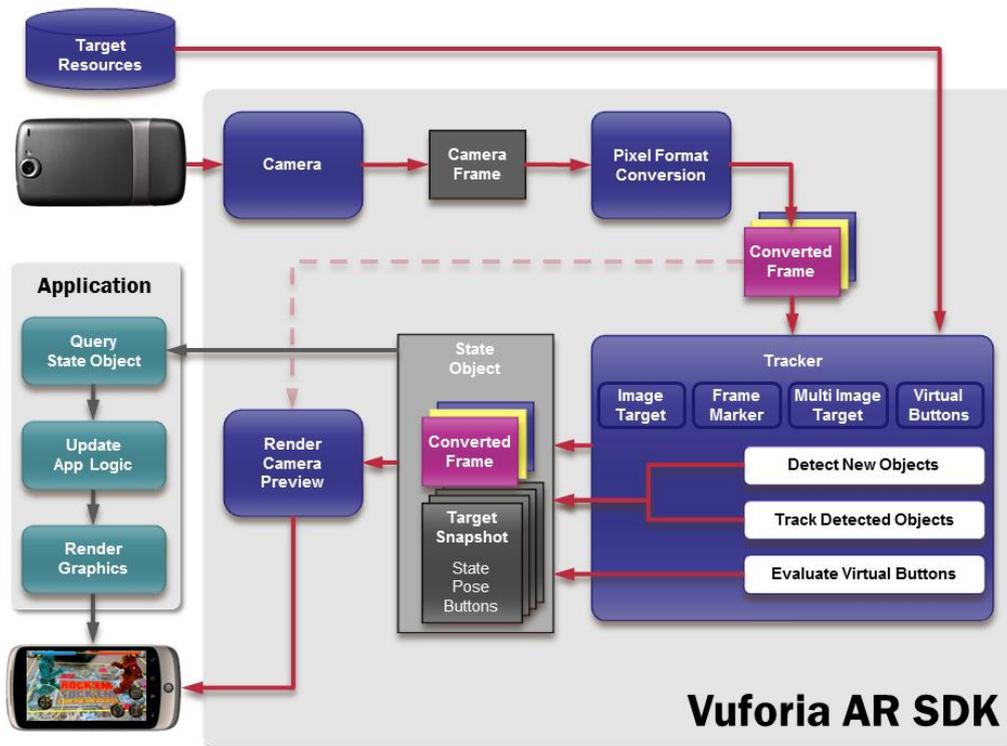


Figura 34. Diagrama de flujo de datos del SDK Vuforia en un entorno de aplicación

### 3.4 Unity 3D

Unity (Technologies, s.f.) es una herramienta creada por Unity Technologies, que nos ayuda a desarrollar videojuegos para diversas plataformas cuenta con un potente motor de renderizado totalmente integrado, un conjunto completo de herramientas intuitivas y flujos de trabajo para crear contenido interactivo en 3D, mediante un editor y scripting para obtener un acabado profesional.

Facilita la creación de objetos virtuales que se representaran en el mundo real, construyendo modelos en 2D o 3D de una manera intuitiva, con soporte para los dispositivos móviles que tenga el S.O.

- Android
- iOS
- Pc's

Además Unity ofrece dos ventajas:

- I. Extensión con Vuforia permitiendo implementar una interfaz de comunicación y exportar nuestros proyectos en Android o iOS
- II. Es multiplataforma (Windows y Mac)

Incluye herramienta de desarrollo MonoDevelop con la que podremos crear scripts en JavaScript, C# y un dialecto de Python llamado “Boo”, extendiendo su funcionalidad del editor utilizando las API que ofrece junto con tutoriales y recursos en su web oficial.

Además de esta herramienta existen en el mercado otras quizás más famosas como:

- UDK de Epic Games
- CryEngine de Crytek.

Habitualmente estas IDE's se utilizan para el desarrollo de videojuegos, se debe tener experiencia sobre la física de los objetos que se desean construir en el modelado y son herramientas parecidas al AutoCAD para generar resultados tangibles en poco tiempo.



Figura 35. SDK Vuforia y Unity 3D

### 3.5 Reconocimiento de Patrones a través del Procesamiento de Imágenes

Consiste en el reconocimiento de patrones de señales. Los patrones se obtienen a partir de los procesos de segmentación, extracción de características y descripción dónde cada objeto queda representado por una colección de descriptores. El sistema de reconocimiento debe asignar a cada objeto su categoría o clase (conjunto de entidades que comparten alguna característica que las diferencia del resto). Para poder reconocer los patrones se siguen los siguientes procesos:

- Adquisición de datos
- Extracción de características
- Toma de decisiones

El punto esencial del reconocimiento de patrones es la clasificación. Donde se extraen sus características.

*Objeto:* Es un concepto con el cual representamos los elementos sujetos a estudio.

*Patrón:* Son los objetos ya clasificados.

*Rasgo:* Propiedad, factor, característica, etc. que se toma en cuenta para estudiar los objetos.

Existen dos tipos:

*Esenciales:* No pueden ser eliminados de la descripción de los objetos sin confundirlos.

*Accidentales:* Pueden ser ignorados en una descripción y los objetos no se confunden.

*Clase:* Es un conjunto de objetos. Al agrupar en clases, se puede hacer de dos formas distintas:

*Por pertenencias duras:* Un objeto pertenece o no a una clase.

*Por pertenencias difusas:* Los objetos pertenecen parcialmente a una clase. Existen clases con intersecciones no vacías.

*Reconocimiento:* Proceso de clasificación de un objeto en una o más clases.

*Filtración:* Consiste en quitar información o datos indeseados de entrada. Dependiendo del uso, el algoritmo o método de filtrado cambia.

### 3.5.1 Extracción de características

Es el proceso de generar características que puedan ser usadas en el proceso de clasificación de los datos. En ocasiones viene precedido por un pre-procesado de la señal, necesario para corregir posibles deficiencias en los datos debido a errores del sensor, o bien para preparar los datos de cara a posteriores procesos en las etapas de extracción de características o clasificación.

Las características elementales están explícitamente presentes en los datos adquiridos y pueden ser pasados directamente a la etapa de clasificación. Las características de alto orden son derivadas de las elementales y son generadas por manipulaciones o transformaciones en los datos.

### 3.5.2 Selección de variables

Consiste en seleccionar cuál es el tipo de características o rasgos más adecuados para describir los objetos. Para ello, se deben localizar los rasgos que inciden en el problema de manera determinante.

Esta etapa también puede ser diseñada dentro de la clasificación.

La selección de variables puede diferenciarse según los objetivos buscados:

- Para la clasificación: la selección de características relevantes, a partir del conjunto total de características que describen a los objetos, se hace con dos motivos fundamentales: mejorar la clasificación o aumentar la velocidad de procesamiento.
- Para la representación: decidir qué características representan mejor a cierto tipo de objetos.

Estrategias de selección de variables:

- Wrapper: La selección de características se hace usando información del mecanismo de clasificación.
- Filter: La selección se hace con un criterio independiente del clasificador.
  - incluye algunos métodos como:
    - Tablas de decisión: Se busca un subconjunto mínimo de variables que no introduzca confusión entre clases.
    - ID3: Se crea un árbol de decisión y se selecciona un conjunto de variables que permita discriminar entre clases.
    - Teoría de testores: Se buscan todos los subconjuntos de variables discriminantes minimales, con estos se evalúa la relevancia de cada variable y se seleccionan aquellas con mayor relevancia.

### 3.5.3 Características Basadas en los momentos geométricos

Los momentos geométricos cartesianos o simplemente momentos han sido ampliamente usados en el análisis de formas, el reconocimiento de patrones y en el análisis de texturas. Los momentos pueden ser calculados para el caso n-dimensional. Para el caso de una función continua 2-dimensional  $f(x, y)$ , la cual se supone ser continua por partes y con soporte compacto, el momento de orden  $(p + q)$  se define como:

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy$$

para  $p, q = 0, 1, 2, \dots$

#### Ecuación 1. Momentos Geométricos Cartesianos

Debido a que  $p$  y  $q$  toman todos los valores enteros no-negativos, se puede demostrar que dichos índices generan un conjunto infinito de momentos y que dicho conjunto determina en forma única cada función  $f(x, y)$ , e inversamente dicho valores quedan determinados unívocamente por  $f(x, y)$  de la ecuación 2.

En el caso de una imagen digital, la doble integral deberá ser reemplazada por una doble suma dando como resultado:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y)$$

para  $p, q = 0, 1, 2, \dots$

#### Ecuación 2. Momentos Geométricos Cartesianos para una Imagen Digital

En el caso de una imagen binaria con valores 0 y 1, la ecuación 2 toma la forma:

$$m_{pq} = \sum_{(x,y) \in R} x^p y^q$$

#### Ecuación 3. Momentos Geométricos Cartesianos para una Imagen Binaria

Donde  $R$  denota la región del objeto.

### 3.5.4 Interpretación Geométrica de los Momentos Normales

Son usados con mayor frecuencia los momentos de hasta orden 3, ya que representan varias propiedades geométricas fundamentales y conocidas de una función de distribución o de un cuerpo. Para visualizar estas propiedades consideremos los valores de los momentos de una función de distribución binaria y continua, como la silueta de un objeto segmentado.

Momento de orden Cero (Área): El momento de orden cero  $m_{00}$  para una función de distribución  $f(x, y)$  dado como:

$$m_{00} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) dx dy$$

#### Ecuación 4. Función de Momento de Orden Cero (Área)

Representa la masa total de función de distribución o imagen dada. Al ser calculada para un objeto segmentado,  $m_{00}$  coincide con el área geométrica del objeto,  $m_{00}$  es el número de píxeles  $(x, y)$  que perteneces a R.

Momentos de orden uno (Centro de Masa): Los momentos de orden uno  $m_{01}$  y  $m_{10}$  son los llamados momentos estáticos en física, siendo usados para localizar el centro de masa (CdM) del objeto.

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

#### Ecuación 5. Función de Momento de Orden Uno (Centro de Masa)

Momento de orden dos: Los momentos de orden dos ( $m_{02}$ ,  $m_{11}$ ,  $m_{20}$ ), conocidos como los momentos de inercia, pueden ser usados para determinar otras características útiles de un objeto como:

- Ejes principales
- Elipse Objeto
- Radio de Giro

Momentos de orden tres (Sesgo proyectivo): Los momentos de orden tres  $\mu_{30}$ ,  $\mu_{03}$ , describen el sesgo de proyecciones del objeto. El sesgo es, como se sabe, una medida estadística clásica del grado de distribución de la desviación de la simetría alrededor de

la media. Los coeficientes de sesgo de las proyecciones de un objeto sobre los ejes “x” y “y” vienen dadas como:

$$Sk_x = \frac{\mu_{30}}{\mu_{20}^{3/2}} \quad Sk_y = \frac{\mu_{03}}{\mu_{02}^{3/2}}$$

**Ecuación 6. Función de Momento de Orden Tres (Sesgo Proyectivo).**

### 3.5.5 Momentos Centrales (Invariantes a Traslaciones)

En el caso de traslaciones quisiéramos que el valor de  $m_{pq}$  se conservase constante ante cambios de posición del objeto sin rotaciones y cambios de escala, en otras palabras quisiéramos que

$$m'_{pq} = m_{pq}, \quad x' = x + a \quad y' = y + b$$

**Ecuación 7. Momentos Centrales (Invariantes a Traslaciones).**

La única manera de lograr esto es que el origen referencial sea movido al centroide del objeto. Esto da origen a los llamados momentos centrales.

Los momentos centrales, denotados como  $\mu_{PQ}$  pueden ser expresados como:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \, dx \, dy$$

**Ecuación 8. Momentos Centrales  $\mu_{PQ}$**

Donde como ya vimos

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad y \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

**Ecuación 9. Momentos Centrales x, y**

Los momentos centrales son equivalentes a los momentos normales de un objeto que ha sido desplazado de manera que el centroide (x, y) del objeto coincide con el origen. Como

resultado, los momentos centrales son invariantes a traslaciones (desplazamientos) de la imagen.

Ahora bien, sabiendo que

$$(a - b)^n = \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k$$

**Ecuación 10. Momentos centrales invariantes a traslaciones (desplazamientos)**

Donde

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

**Ecuación 11. Combinación**

Se tiene que

$$\mu_{pq} = \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} (-1)^{k+l} m_{p-k, q-l} m_{10}^k m_{01}^l m_{00}^{-(k+l)}$$

para  $p + q = 2, 3, \dots$

**Ecuación 12. Momentos Centrales utilizando la Combinación**

En particular, los momentos centrales hasta de orden tres vienen dados como:

$$\begin{array}{ll} \mu_{00} = m_{00} & \mu_{11} = m_{11} - \bar{y}m_{10} \\ \mu_{10} = 0 & \mu_{30} = m_{30} - 3\bar{x}m_{20} + 2m_{10}\bar{x}^2 \\ \mu_{01} = 0 & \mu_{12} = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2m_{10}\bar{y}^2 \\ \mu_{20} = m_{20} - \bar{x}m_{10} & \mu_{21} = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2m_{01}\bar{x}^2 \\ \mu_{02} = m_{02} - \bar{y}m_{01} & \mu_{03} = m_{03} - 3\bar{y}m_{02} + 2m_{01}\bar{y}^2 \end{array}$$

**Ecuación 13. Momentos Centrales de orden 3**

### 3.5.6 Invariantes a rotaciones (Los Invariantes de Hu)

Los primeros cuatro invariantes a traslaciones y rotaciones de Hu vienen dados como:

$$\begin{aligned}\phi_1 &= \mu_{20} + \mu_{02} \\ \phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11} \\ \phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \\ \phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2\end{aligned}$$

#### Ecuación 14 Momentos Centrales de orden 4 y Rotaciones de Hu

Invariantes a cambios de escala (Momentos Centrales Normalizados)

Cambios de escala son causados por transformaciones de coordenadas de la forma:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

#### Ecuación 15. Transformación de Coordenadas.

Invariantes a cambio de escala pueden obtenerse usando momentos al dividir cada momento por un factor de normalización que cancela el efecto de escalamiento.

Si  $f'(x', y')$  es la imagen  $f(x, y)$  después de un escalamiento en cada eje por alfa a través de la aplicación dada por la ecuación, entonces  $f'(x', y') = f(\alpha x, \alpha y)$ , y  $x' = \alpha x$ ,  $y' = \alpha y$ , entonces tenemos:

$$\begin{aligned}m'_{pq} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x'^p y'^q f'(x', y') dx' dy' \\ &= \alpha^{p+q+2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy\end{aligned}$$

Por tanto,  $m'_{pq} = \alpha^{p+q+2} m_{pq}$ , de manera similar  $\mu'_{pq} = \alpha^{p+q+2} \mu_{pq}$ . En particular  $\mu'_{00} = \alpha^2 \mu_{00}$ , de donde podemos ver que

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00} \gamma}, \quad \gamma = \frac{p+q}{2} + 1, \quad p+q = 2, 3, \dots$$

#### Ecuación 16. Escalamiento de la imagen por cada Eje.

Invariantes a traslaciones, rotaciones y cambios de escala

Al reemplazar los momentos centrales en las expresiones para invariantes a rotaciones por los invariantes a escala, resultan los conocidos invariantes a traslaciones, rotaciones y cambios de escala de Hu:

$$\begin{aligned}
 \phi_1 &= \eta_{20} + \eta_{02} \\
 \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11} \\
 \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
 \phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
 \phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[ (\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + \\
 &\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[ 3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] \\
 \phi_6 &= (\eta_{20} - \eta_{02}) \left[ (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + \\
 &\quad 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
 \phi_7 &= (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12}) \left[ (\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + \\
 &\quad (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \left[ 3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right]
 \end{aligned}$$

**Ecuación 17. Invariantes a Traslaciones, Rotaciones y cambios de escala de Hu.**

### 3.6 Arduino

Es una plataforma de hardware libre, para la creación de prototipos basada en software y hardware flexibles.

Se creó para:

- Diseñadores
- Artistas
- Aficionados
- Cualquiera interesado en crear entornos u objetos interactivos.

Tiene la capacidad de controlar luces, sensores, motores, etc. o comunicarse con otros módulos de desarrollo.

El microcontrolador que implementa la placa Arduino, se programa mediante el lenguaje de programación Arduino (basado en **Wiring**) y el entorno de desarrollo Arduino (basado en **Processing**).

Los proyectos realizados en Arduino pueden ejecutarse, sin la necesidad de estar conectado a una computadora, tiene la posibilidad de hacerlo y comunicarse con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP, Java, C++, etc.).

El software puede ser adquirido de forma gratuita. Las bibliotecas (ficheros) de diseño de (CAD) están disponibles bajo licencia abierta, así que se puede adaptarse dependiendo del proyecto.

Las placas de Arduino pueden implementar microcontroladores Atmega 168, Atmega 328, Atmega8, chips sencillos de bajo coste que permiten el desarrollo de múltiples diseños.

**Wiring:** es un entorno de programación de código abierto y de dispositivos electrónicos de E/S, para poder explorar las artes electrónicas, usando medios tangibles, enseñando y aprendiendo la programación de la computadora y los prototipos con electrónica. Se ilustra este concepto de la programación con la electrónica para explorar la interacción física y los aspectos de los medios a utilizar.

**Processing:** es un entorno de programación de código abierto para personas que buscan crear imágenes, animaciones, e interacciones. Inicialmente fue desarrollado para servir como un cuaderno de bocetos de software y para enseñar los fundamentos de la programación dentro de un contexto visual, convirtiéndose en una herramienta para generar trabajos profesionales

Los programas realizados con Arduino se dividen en tres partes principales:

1. Estructura
2. Valores (variables y constantes)
3. Funciones

El lenguaje de programación en Arduino se basa en C/C++ y su interfaz es realizada en Java. (Team, s.f.)



Figura 36. Placa Arduino Duemilanove

### 3.7 Sensores ultrasónicos

Es un transductor que emite “paquetes” de ultrasonido, contiene una serie de ondas sonoras intermitentes.

El paquete se emite en forma canónica, el cual rebota o se refleja en la superficie objetivo, recibiendo su regreso en otro transductor. Se mide la duración del pulso de la onda ultrasónica, el tiempo transcurrido en que tarda en regresar la onda, determina la distancia que se encuentra respecto al objeto.

Varios factores afectan la medición del ultrasonido:

- La naturaleza de la superficie.
- El ángulo del cono y la distancia del objetivo a censar
- Las condiciones ambientales como son:
  - Temperatura
  - Humedad relativa
  - Gases
  - Vapores y la Presión.

Los sensores están calibrados con ajustes manuales o automáticos para compensar la mayoría de estas condiciones cambiantes del entorno.

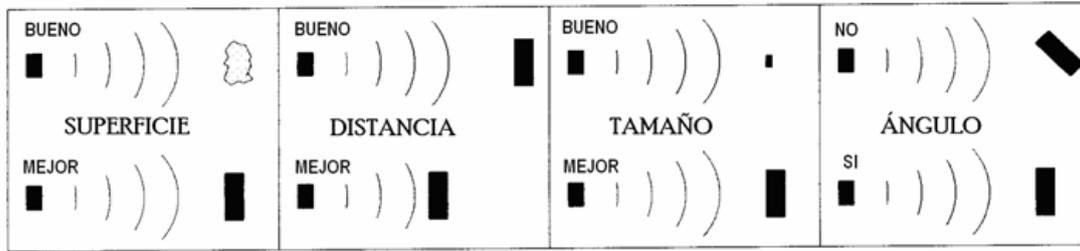


Figura 37. Comportamiento de los Sensores

Las siguientes condiciones aseguran la correcta operación de los sensores ultrasónicos:

✓ **Superficie**

La superficie-objetivo ideal es dura y lisa. Esta reflejara una mayor cantidad de señal que una superficie suave y rugosa.

Un eco débil, resulta ser un objetivo pequeño, o con superficie que absorbe las ondas ultrasónicas, disminuyendo su exactitud.

✓ **Distancia**

Mientras más corta sea la distancia al objetivo, se vuelve más fuerte el eco. De modo que si la distancia aumenta requerimos mejores características reflejantes en la superficie objetivo.

✓ **Tamaño**

Un objetivo grande tendrá mayor superficie para rebotar la señal que un objetivo pequeño. La porción de superficie reconocida como “objetivo” es normalmente la más cercana al sensor.

✓ **Ángulo**

La inclinación de la superficie objetivo afecta la reflectividad. Lo ideal es que la superficie debe estar perpendicular ( $90^\circ$ ) al sensor, para un rebote ideal del eco. Si la superficie, esta inclinada fuertemente la señal será desviada, por lo que causara que nunca llegue al transductor y no exista eco

### 3.7.1 Paquete ultrasónico

El patrón del haz producido por el sensor, se expresa en número de grados, que el haz se separa de la línea central del sensor. El haz se expande con un patrón cónico a partir del transductor y aunque el haz ultrasónico continúa cubriendo más área, el área de detección del sensor empieza a acortarse respecto al rango de operación publicado.

El área de censado se ve afectada por el número de pulsos enviado por el sensor y por el nivel de sensibilidad. A nivel alto de pulsos y sensibilidad, mayor superficie que a niveles bajos. Las imágenes siguientes ilustran estos principios

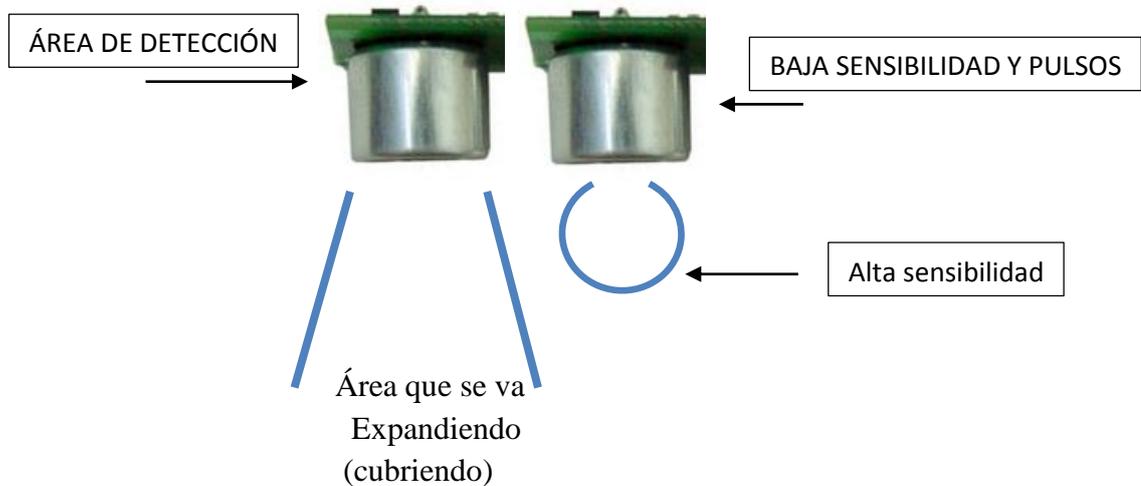


Figura 38. Proyección de los Sensores (Ondas Ultrasónicas)

Calcular el área en que se expande el haz ultrasónico empleando una simple formula de trigonometría.

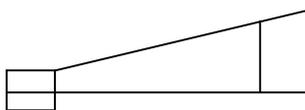
$$a = b \cdot \tan(A)$$

#### Ecuación 18. Área de Propagación Ultrasónica

Donde  $a = b$  desviación del eje del sensor

$b =$  distancia

$A =$  ángulo de separación del haz ultrasónico



Tangente de ángulos usados comúnmente entre los sensores ultrasónicos:

$$\tan(7^\circ) = 0.12278$$

$$\tan(8^\circ) = 0.14054$$

$$\tan(9^\circ) = 0.15838$$

Figura 39. Ángulo de Propagación

Idealmente el área del objetivo, debe exceder el diámetro de haz, expandido a la distancia del rango de censado requerido.

### 3.7.2 Aplicaciones del sensor ultrasónico.

El módulo SRF05 es capaz de generar una señal lógica de salida cuya duración determina la distancia de un objeto. Las aplicaciones son numerosas, citamos unas cuantas a modos de ejemplo:

- Aplicaciones de control, que actúan, en función de la distancia o tamaño de objetos.
- Alarmas, que se activan cuando el intruso se aproxima a una determinada distancia
- Micro robótica en donde es necesario que se actúe en función de la distancia que separa al robot de cualquier otro objeto

## 3.8 Motores

Un motor eléctrico es esencialmente una máquina que transforma energía eléctrica en movimiento o trabajo mecánico, a través de interacciones electromagnéticas, algunos motores eléctricos son reversibles, esto quiere decir que pueden transformar energía mecánica en energía eléctrica funcionando como generadores.

### 3.8.1 Principio de funcionamiento del motor

Se basa en la ley de Faraday que indica que cualquier conductor que se mueve en el seno del campo magnético de un imán, genera una diferencia de potencial (D.D.P) entre su extremo proporcional a la velocidad de desplazamiento.

Normalmente en un motor, emplea un cierto número de espiras devanadas sobre un núcleo magnético de forma apropiada y también en algunas ocasiones, se sustituye el imán permanente, que crea el campo por un electroimán, el cual produce el mismo efecto cuando se le aplica la corriente excitadora. A este último elemento (Imán o electroimán) se le denomina inductor, al conjunto de espiras y núcleos del móvil que constituyen el inducido.

El sentido de la corriente eléctrica que circula por el campo inducido está definido mediante la *Ley de Lenz* que indica que toda variación que se produzca en el campo magnético tiende a crear un efecto en sentido opuesto que compense y anule la causa que la produjo.

Esta ley, indica que la corriente inducida creará un campo magnético, que se oponga al movimiento de la misma lo que obligará a aplicar una determinada energía para mantener el movimiento, dependerá lógicamente de la intensidad de la corriente generada y del valor de la resistencia de carga ( $R_c$ ), se calcula, como el producto de la energía consumida en la carga, por un número que expresará el rendimiento de la conversión.

Ahora bien, todos los fenómenos expresados corresponden al efecto opuesto al de un motor, es decir, que mediante el sistema descrito se genera un corriente eléctrica a partir de un movimiento mecánico, lo que corresponde al principio de funcionamiento de un *dinamo*, sin embargo, al ser dicho efecto reversible, bastará con invertir los papeles; si en lugar de extraer corriente del inducido se le aplica un determinada tensión exterior, se producirá la circulación de una cierta intensidad de corriente por las espiras y éstas comenzarán a girar, completándose así el motor. Es importante considerar que teniendo en cuenta la ley de Lenz mencionada anteriormente, al girar se creará una determinada tensión eléctrica, de sentido contrario al exterior que tenderá a oponerse al paso de la corriente para compensar así las variaciones de flujo magnético producidas, denominada *fuerza contra electromotriz* (FCEM)

### 3.8.2 Características

Normalmente los motores se caracterizan por dos parámetros que expresan directamente sus propiedades. Son los siguientes:

- **Velocidad de Rotación:**  
Indica el número de vueltas por unidad de tiempo que produce el motor y depende por completo de la forma de construcción del mismo, de la tensión de alimentación, así como de la carga mecánica que se acople a su eje. Las unidades empleadas son las revoluciones por minuto (r.p.m.) y las revoluciones por segundo (r.p.s.).
- **Par Motor:**  
Expresa la fuerza de actuación de éste y depende lógicamente de la potencia que sea capaz de desarrollar dicho motor, así como de la velocidad de rotación del mismo. El concepto de par motor es importante a la hora de elegir un modelo para una aplicación determinada; se define como la fuerza que es capaz de vencer el motor multiplicada por el radio de giro. Esto significa que no supone lo mismo mover, por ejemplo, una polea que transmita una fuerza de 10kg., con un radio de 5cm. que con otro radio de 10cm., ya que el par motor será en el segundo caso el doble que el del primero. Las unidades de medida suelen ser el kg x cm o bien, el g x cm.

Además de estos factores también se tienen en cuenta otros como son las condiciones de arranque, la potencia absorbida y el factor de potencia.

Existe una relación matemática que liga ambos parámetros, ya que como se ha explicado no son independientes entre sí; dicha relación se expresa por la siguiente fórmula.

$$M= 0,95 \times P/N$$

#### **Ecuación 19. Par Motor o torque**

Donde M es el par motor expresado en N\*m, P es la potencia absorbida en Watts y N es la velocidad en revoluciones por minuto.

### **3.9 Servomotor**

Es un dispositivo que dispone en su interior un motor con un reductor de velocidad y multiplicador de fuerza, también dispone de un circuito que controla el manejo del motor. El movimiento del eje de salida es de 180° en la mayoría de ellos, pero puede ser fácilmente modificado para tener un recorrido libre de 360° y actuar así como un motor.



**Figura 40. Servomotor**

#### **3.9.1 Funcionamiento:**

El control de posición lo efectúa el servo internamente mediante un potenciómetro que va conectado mecánicamente al eje de salida y controla un pwm (modulador de anchura de pulsos) interno para así compararlo con la entrada pwm externa del servo, mediante un sistema diferencial, y así modificar la posición del eje de salida hasta que los valores se igualen y el servo pare en la posición indicada, en esta posición el motor del servo deja de consumir corriente y tan solo circula una pequeña corriente hasta el circuito interno, si forzamos el servo (moviendo el eje de salida con la mano) en este momento el control diferencial interno lo detecta y envía la corriente necesaria al motor para corregir la posición.

Para controlar un servo tendremos que aplicar un pulso de duración y frecuencia específicas. Todos los servos disponen de tres cables dos para alimentación Vcc y Gnd y otro cable para aplicar el tren de pulsos de control que harán que el circuito de control diferencial interno ponga el servo en la posición indicada por la anchura del pulso.

### **3.10 Motor reductor**

Es un dispositivo que contiene un sistema completo de fuerza motriz, que consta de un motor eléctrico y un tren de engranaje de reducción integrado, dicho engranaje es de rueda dentada, el engranaje reductor y el motor eléctrico están dispuestos axialmente próximos en una carcasa aproximadamente cilíndrica, el motor eléctrico posee un anillo de estator fijo en la carcasa con un rotor que gira en éste alrededor de un eje Z que se extiende centrado en la carcasa.

Los motores reductores permiten que los motores eléctricos funcionen a diferentes velocidades para los que fueron diseñados.

La función de un motor reductor es disminuir la velocidad de los motores y permitir eficientemente su manejo, en cuanto la potencia y fuerza.

#### **3.10.1 Beneficios al emplear motor reductores**

- Tener un excelente equilibrio en la velocidad como en la potencia transmitida.
- Una mayor eficiencia en la transmisión de la potencia suministrada por el motor.
- Control en la velocidad de operación.

#### **3.10.2 Características del reductor o motor-reductor**

- Potencia E/S (HP).
- Velocidad E/S (RPM).
- PAR (o torque), a la salida del mismo, en KG/m.
- Relación de reducción: índice que detalla la relación entre las RPM de entrada y salida.

#### **3.10.3 Características del trabajo a realizar**

- ❖ Tipos de acoplamiento entre máquina motriz, reductor y salida de carga.
- ❖ Carga: uniforme, discontinua, con choque, con embrague, etc.
- ❖ Tipo de Torque y potencia del motor

## Conceptos Fundamentales:

**Revoluciones por minuto (r.p.m) o velocidad angular:** Se define como la cantidad de vueltas completas que da el rotor en el lapso de un minuto; el símbolo de la velocidad angular es  $W$ .

$$W = N = 2\Pi F \qquad F = \frac{1}{t}$$

Donde:  $W=N$  = Revoluciones por minuto o velocidad angular  
 $\Pi$  = Constante [3.14]  
 $F$  = Frecuencia  
 $t$  = Tiempo

### Ecuación 20. Revoluciones por minuto de un motor

**Par o Torque:** Un par de fuerzas es un conjunto de dos fuerzas de magnitudes iguales pero de sentido contrario. El momento del par de fuerza o torque, se representa por un vector perpendicular al plano del par.

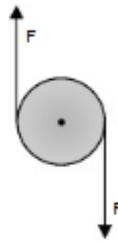


Figura 41. Torque de un Motor

**Par de arranque:** Es el par que va a desarrollar el motor para romper sus condiciones iniciales de inercia y pueda comenzar a operar.

**Eficiencia:** Es un factor que indica el grado de pérdida de energía, trabajo o potencia de cualquier aparato eléctrico o mecánico.

**Engranaje o ruedas dentadas:** Se denomina al mecanismo utilizado para transmitir potencia de un componente a otro dentro de una máquina. Los engranajes están formados por dos ruedas dentadas, de las cuales la mayor se denomina 'corona' y la menor 'piñón'. Un engranaje sirve para transmitir movimiento circular mediante contacto de ruedas dentadas. Si el sistema está compuesto de más de un par de ruedas dentadas, se denomina 'tren'.

## Relación de Engranaje

### -Módulo métrico.

El paso de los engranes en el sistema métrico se basa en esta unidad y se llama modulo (m). El módulo de un engrane solo depende del número de dientes del engrane y el diámetro exterior  $D_e$ .

$$m = \frac{D_e}{N}$$

**Ecuación 21. Relación de Engranaje**

$D_e$ . = Diámetro externo.

N = número de dientes del engrane

### -Diámetro de paso

El diámetro de paso  $D_p$ , o también conocido como círculo de paso es el diámetro en donde existe un punto de contacto tangencial entre dos engranes y con él se puede calcular la distancia entre los ejes de dos engranes rectos.

$$D_p = mN$$

**Ecuación 22. Diámetro de Paso de un Motor**

M=módulo métrico

N= número de dientes

Se define la relación de transmisión  $i$ : 1 como la cantidad de vueltas que debe dar el engranaje del motor para que el engranaje conducido de una vuelta. Por ejemplo la relación 30:1, esto quiere decir, por cada 30 vueltas que dé el piñón se generara una vuelta.

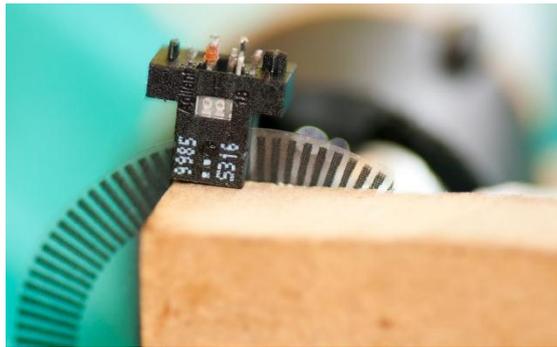
## 3.11 Encoder

En la actualidad hay muchos procesos que emplean piezas móviles, como ejes de motores, máquinas, herramientas, cintas transportadoras, ejes de brazos robóticos, etc., lo que implica tener un control de estos procesos, colocando transductores que se adapten al movimiento y nos faciliten información precisa, como su posición o su distancia a un punto determinado.

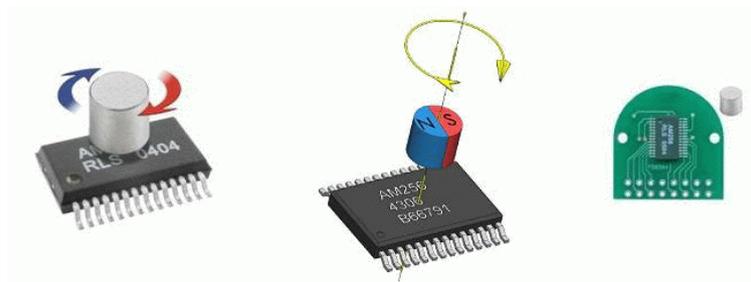
Los encoders son transductores rotativos que transforman un movimiento angular en una serie de impulsos digitales. Estos impulsos generados pueden ser utilizados para controlar los desplazamientos de tipo angular o de tipo lineal, se asocian a cremalleras o huesillos. Las señales eléctricas de rotación pueden ser procesadas por controles numéricos (CNC), controladores lógicos programables (PLC) o sistemas de control, etc.

Los encoders, según su fabricación, pueden ser:

- **Ópticos**  
Proporcionan altas revoluciones y velocidades de operación, se emplean en ambientes industriales por su larga vida útil en ambientes industriales.
- **Magnéticos**  
Utilizados en trabajos pesados como laminadoras de papel y acero, teniendo una buena resistencia al polvo, a la humedad, golpe térmico y mecánico, aunque actualmente están siendo reemplazados por transductores ópticos en todas las aplicaciones.



**Figura 42. Encoder óptico**



**Figura 43. Encoder Magnético**

# **CAPÍTULO IV**

## **DISEÑO**

## 4.1 DISEÑO DEL SISTEMA

Una de las estrategias fundamentales para entender el funcionamiento del Sistema AA-AR, es el uso de la técnica de modelado visual, que tiene la facultada de bosquejar y conocer los distintos aspectos del sistema.

Se mencionaran brevemente algunos conceptos que nos sirven para poder aplicar esta técnica.

¿Qué es un modelo?

Un modelo se puede definir de distintas formas de acuerdo al contexto del que se esté hablando, pero solo se mencionaran los dos más adecuados para este propósito

1. Es la abstracción de un conjunto de mecanismos que permitan crear una representación, de acuerdo al problema que se desea resolver
2. Es un bosquejo que representa un conjunto real con cierto grado de precisión y en la forma más completa posible

¿Por qué el Software necesita modelos?

Necesita modelos para comunicar de manera sencilla una idea abstracta, facilitando la descripción de un producto existente o no.

El modelo más detallado de un producto de software es el código fuente, pero no es de mucha ayuda. Es como decir que el mejor modelo de un edificio es el edificio mismo; esto no nos sirve para concebirlo antes de su construcción ni para entender sus aspectos más ocultos.

Sin embargo, en el software, el modelado es aún más importante por las siguientes razones:

- El software es invisible e intangible: sólo se ve su comportamiento
- El software es mucho más modificable que otros productos
- El software se desarrolla para alguna tarea específica esto implica enfrentarse a nuevos y distintos problemas
- El software es complejo ya que interactúa con diferentes sistemas, que pueden ir cambiando de estados con el tiempo.

No solamente software se puede modelar, sino también el funcionamiento del hardware.

¿Qué es UML?

Surgió en 1995, por iniciativa de Grady Booch, James Rumbaugh e Ivar Jacobson, tres conocidos ingenieros de software, que ya habían creado sus propias notaciones de modelado. Precisamente, UML significa “Lenguaje unificado de Modelo”, ya que surgió como síntesis de los mejores elementos de las notaciones previas (Fontela, 2011)

Se dice que es un Lenguaje, en cuanto a que es una herramienta de comunicación formal, con una serie de construcciones, una sintaxis y una semántica definidas.

Así, los elementos constructivos son diagramas y sus partes, la sintaxis es la descripción de cómo deben realizarse esos diagramas y la semántica define el significado de cada diagrama y elemento de los mismos.

¿Para qué usar UML?

Se puede distinguir dos aspectos fundamentales del para que usarlo

1. Como herramienta de comunicación entre humanos.
2. Como herramienta de desarrollo

En el primer caso, se usa para mejorar el entendimiento de alguno o varios aspectos dentro del desarrollo.

En el segundo caso, se utiliza para colocar aquellos elementos que sean centrales en el sistema, evitando detalles innecesarios, esto permite ser legible y que brinde comprensión.

Teniendo presente, estos conceptos ya nos podemos dar la tarea de modelar el Sistema AA-AR.

#### 4.1.1 Diagramas UML

#### 4.1.2 Diagramas de Caso de Uso

Los Casos de uso están enfocados en explicar la funcionalidad del Sistema AA-AR. Debemos mencionar que el sistema desarrollado engloba tanto la parte del Software como la del Hardware

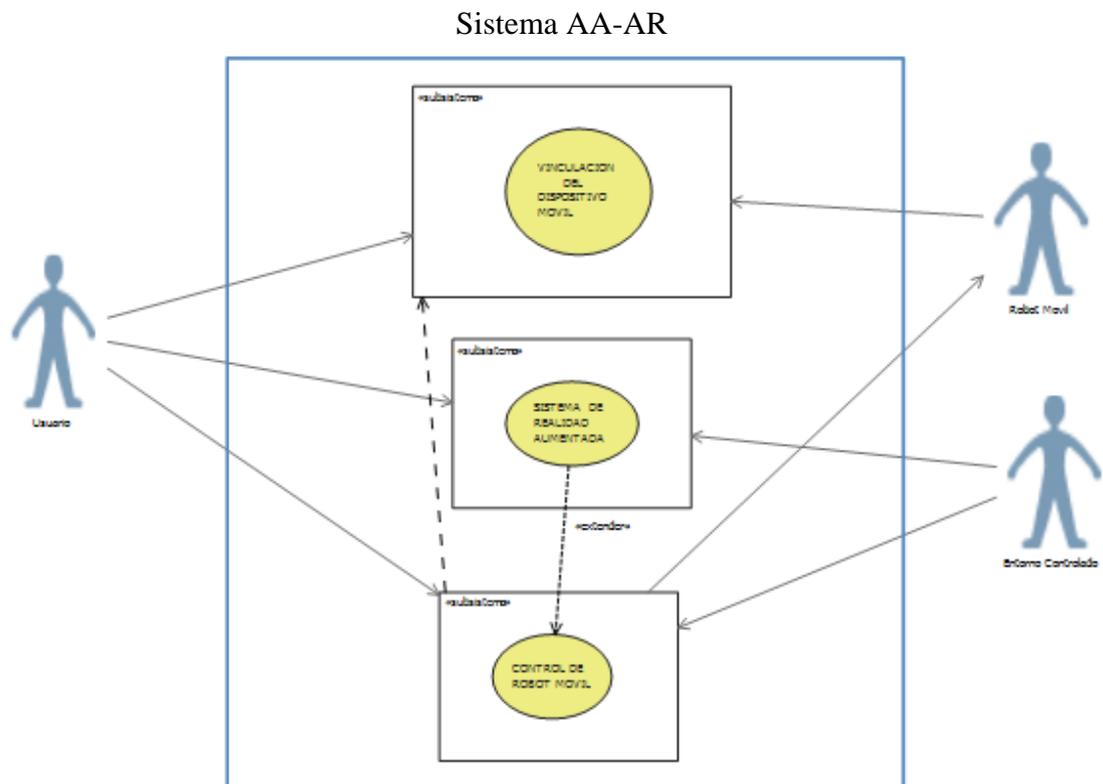


Diagrama 1. General de sistema AA-AR

Sistema AA-AR	
Actor Primario	Conductor (Usuario)
Actor Secundario	El entorno controlado / Robot móvil
Descripción	Interacción entre usuario y el robot móvil para la toma de decisiones
Disparador	Conducción / Control del robot móvil
Pre-condiciones	<ol style="list-style-type: none"> <li>1. Estar encendido el robot móvil</li> <li>2. El Sistema AA-AR debe estar vinculado con el robot móvil</li> </ol>
Post-condiciones	<ol style="list-style-type: none"> <li>1. Se almacena el nombre del dispositivo vinculado</li> <li>2. Se da a conocer oportunamente la información que encuentra dentro del trayectoria</li> <li>3. Se llega al destino establecido</li> </ol>
Flujo Normal	<ol style="list-style-type: none"> <li>1. Se inicia la aplicación (E1)</li> <li>2. Se busca el dispositivo móvil a controlar (S1, S2,E5)</li> <li>3. La cámara y los sensores están censando el entorno (E4)</li> <li>4. Se inicia el recorrido (S3, S4,S7)</li> <li>5. Se encuentra alguna señal de tránsito (E6)</li> <li>6. Se le notifica al usuario mediante AR (S5)</li> <li>7. Termina la trayectoria el robot móvil (E4, E7, S8)</li> </ol>
Flujo Alternativos	<p>S1. Muestra el historial de todos los dispositivos conectados anteriormente.</p> <p>S2. Se Conecta el dispositivo móvil la primera vez</p> <ol style="list-style-type: none"> <li>a) Se manda una petición bluetooth para buscar el dispositivo</li> <li>b) El dispositivo responde a la petición recibida</li> <li>c) Se envía un código de seguridad para establecer la comunicación (E2 a E3)</li> </ol> <p>S3. El usuario toma el control del robot móvil</p> <p>S4. El sistema de navegación continua con la trayectoria</p> <p>S5. El usuario puede tomar las siguientes decisiones de acuerdo a la señal de transito que se le presente</p> <ol style="list-style-type: none"> <li>a) Cancelar la acción que va a realizar el Sistema</li> <li>b) No cancelar la acción que va a realizar</li> </ol> <p>S7. Se busca el camino y continuamente se censa su entorno (E7,E4)</p> <p>S8. No encuentra el camino a seguir por un cierto periodo</p>
Errores	<p>E1. El sistema no inicia por falta de recursos</p> <p>E2. Error en la comunicación con el Dispositivo móvil</p> <p>E3. Error en el código de seguridad</p> <p>E4. Perdida de comunicación</p> <p>E5. El robot móvil se encuentra apagado</p> <p>E6. Interpretar inadecuadamente la señal de Transito</p> <p>E7. Robot sin energía</p>
Prioridad	Alta
Frecuencia de uso	Alta
Reglas de negocio	Debe existir comunicación con el dispositivo móvil
Notas y preguntas	*: es obligatorio conectarse al robot móvil

El sistema se planteó de la siguiente forma dividirlo en tres subsistemas que mutuamente van a comunicarse entre sí, esto permite la interacción entre el hombre y la máquina.

1) Vinculación del Dispositivo Móvil

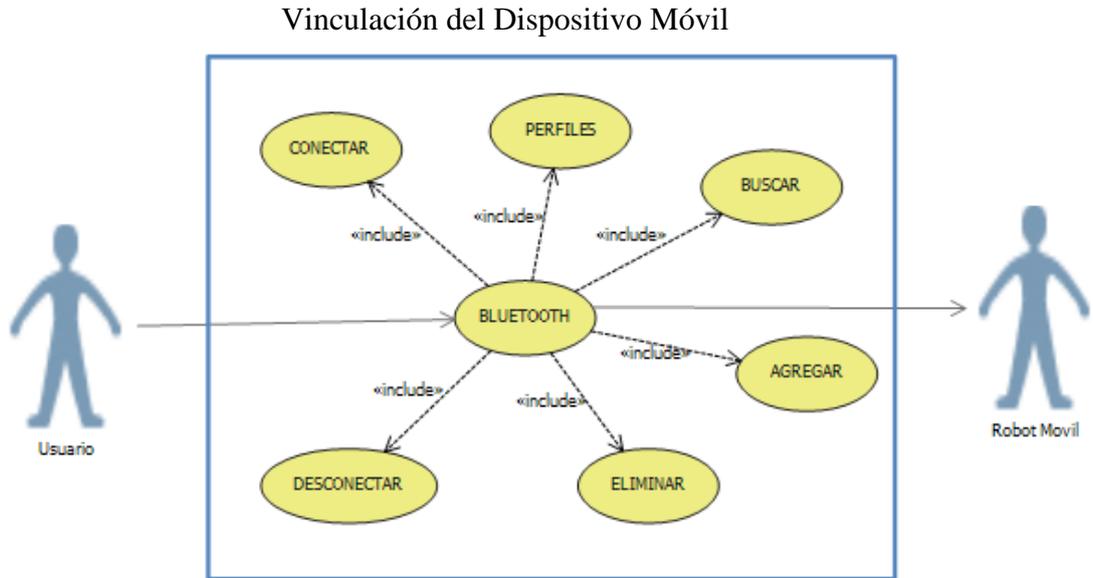


Diagrama 2. Módulo de vinculación del dispositivo móvil

Vinculación del Dispositivo Móvil	
Actor Primario	Conductor (Usuario)
Actor Secundario	Robot móvil
Descripción	Establecimiento de Comunicación por Bluetooth
Disparador	Conducción / Control del robot móvil
Pre-condiciones	1. Contar con Bluetooth tanto el dispositivo móvil como el robot móvil 2. El robot móvil debe estar en modo esclavo
Post-condiciones	1. Se establece la comunicación 2. Se almacena su dirección Mac para futuras conexiones 3. Se termina la comunicación
Flujo Normal	1. Se inicia la aplicación (S1) 2. Petición para activar o cancelar el Bluetooth (S2) 3. Se envía una solicitud (S3) 4. Se elige la dirección Mac "ID" (S4) 5. Se inicia la actividad de descubrimiento (S6) 6. Se crea un puerto serial virtual (S7) 7. Se crea un mecanismo de seguridad (S8, E2) 8. Se realiza la conexión satisfactoriamente (S9)

Flujo Alternativos	<p>S1. Se hace una petición de servicio al Sistema Android para buscar si existe Bluetooth (E1).</p> <p>S2. Dependiendo del estado que se encuentre el bluetooth (Apagado / Prendido)</p> <p>S3. Todo dispositivo bluetooth que este aproximadamente dentro del rango de 10m respecto al emisor, y además que su estado este en modo pasivo, escuchara la petición de solicitud y responderá con su dirección MAC</p> <p>S4. La elección dependerá de la forma en que se haya nombrado el dispositivo móvil. (S5)</p> <p>S5. Se sincroniza con el punto de acceso y se guarda en la paginación del bluetooth (emisor), esto con lleva a sincronizar su reloj y frecuencia con el (receptor).</p> <p>S6. Se estable un enlace con el punto de acceso, para crear un canal de comunicación, mediante el protocolo L2CAP.</p> <p>S7. De acuerdo a las necesidades de la aplicación se agrega un canal adicional denominado RFCOMM que permite el transporte de datos</p> <p>S8. Este mecanismo denominado emparejamiento, brinda restricción de acceso y se establece mediante una clave cifrada (PIN) y se le envía al dispositivo maestro.</p> <p>S9. El dispositivo maestro puede utilizar libremente el canal de comunicación, sin que se interrumpa la señal por otra conexión bluetooth</p>
Errores	<p>E1. No se cuenta con dispositivo bluetooth, por lo tanto se cierra la aplicación</p> <p>E2. No coinciden los código de seguridad, tanto para el emisor como el receptor</p> <p>E3. Perdida de comunicación</p>
Prioridad	Alta
Frecuencia de uso	Alta
Reglas de negocio	Debe existir Bluetooth
Notas y preguntas	*: debe estar dentro del rango del bluetooth

2) Sistema de Realidad Aumentada

Sistema de Realidad Aumentada

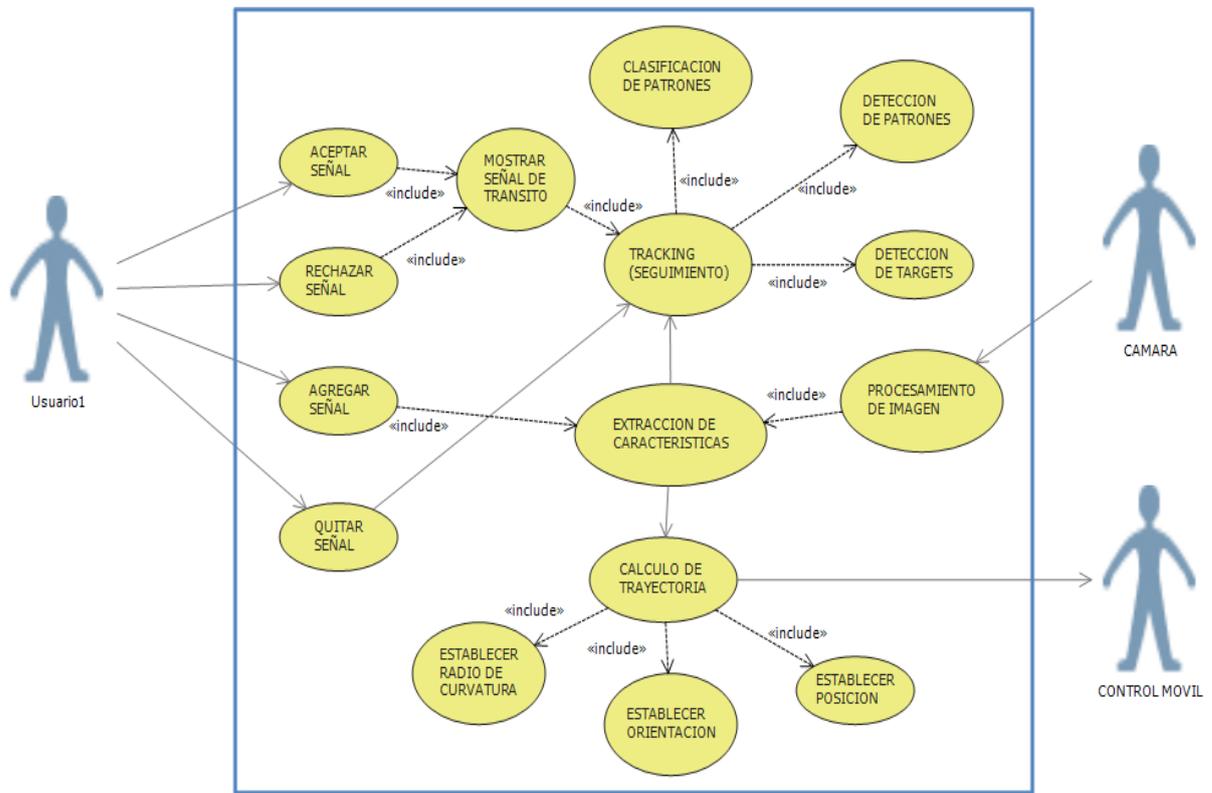


Diagrama 3. Módulo de Realidad Aumentada

Sistema Realidad Aumentada	
Actor Primario	Conductor (Usuario)
Actor Secundario	Cámara/ Sistema de Control móvil
Descripción	Adquisición de Información, Procesamiento e Interacción
Disparador	Inicio de la Trayectoria
Pre-condiciones	<ol style="list-style-type: none"> <li>1. Encendido el control móvil</li> <li>2. Cámara en funcionamiento</li> <li>3. Vinculación con el Control móvil</li> </ol>
Post-condiciones	<ol style="list-style-type: none"> <li>1. Se llega al destino establecido sin ninguna pérdida de información.</li> <li>2. Satisfacción por parte del usuario</li> </ol>
Flujo Normal	<ol style="list-style-type: none"> <li>1. Se inicia la captura de la imagen mediante la cámara (S1, E1)</li> <li>2. Se realiza el procesamiento de imagen (S2)</li> <li>3. Extracción de Características (S3)</li> <li>4. Se realiza el proceso de Tracking (S3, S4,S5,S6)</li> <li>5. Se realiza el cálculo de Trayectoria (S7,E6)</li> </ol>

	<p>6. Se le notifica al usuario mediante AR (S6,E7)</p> <p>7. Agregar nueva señal de Transito (S8,E8)</p> <p>8. Quitar una señal de Transito (S9)</p> <p>9. Se envía al Control móvil, el cálculo de trayectoria y acciones que tomo el usuario (E6, S10)</p>
Flujo Alternativos	<p>S1. Se realiza una petición al S.O Android para el uso de la cámara (E4,E5)</p> <p>S2. Se aplican técnicas adecuadas para el mejoramiento y filtrado de la imagen</p> <p>S3. La extracción debe ser robusta, contener discriminantes y poseer determinadas invariencias</p> <p>S4. Se realiza un seguimiento y búsqueda de los patrones previamente clasificados (señal de transito).</p> <p>S5. El Tracking realiza una comparación entre los patrones detectados (S4)</p> <p>S6. Se muestra la AR en los Targets destino</p> <p>S7. Debe existir al menos una línea para realizar los siguientes cálculos:</p> <ul style="list-style-type: none"> <li>a) Calculo de Posición.</li> <li>b) Calculo de Orientación</li> <li>c) Calculo Radio de Curvatura</li> </ul> <p>S8. El usuario proporciona la nueva señal de Transito a reconocer.</p> <ul style="list-style-type: none"> <li>a) El sistema obtiene las características de la señal de Transito</li> <li>b) Se almacenan las características y se clasifican</li> <li>c) Se asocian las características con el Marcador a encontrar</li> <li>d) Se muestra la AR en el Marcador para brindar información.</li> </ul> <p>S9. El usuario selecciona, la señal de transito que ya no desea reconocer.</p> <p>S10. El control móvil realizara acciones mediante el apoyo de los datos brindados.</p>
Errores/Excepciones	<p>E1. Error de comunicación</p> <p>E2. Error en la comunicación con el Dispositivo móvil</p> <p>E3. Excepción si no se encuentra camino.</p> <p>E4. El dispositivo móvil no dispone de cámara</p> <p>E5. Está en uso por otra aplicación</p> <p>E6. Por un periodo prolongado no se encuentra el camino</p> <p>E7. Interpreta inadecuadamente la señal de Transito</p> <p>E8. Dispositivo no cuenta con suficiente memoria</p>
Prioridad	Alta
Frecuencia de uso	Alta
Reglas de negocio	Tener conocimiento previo de las señales que se van a detectar
Notas y preguntas	*: es obligatorio conectarse al robot móvil

### 3. Control Móvil

Especialmente este caso de uso está enfocado al módulo del hardware y su comportamiento va estar enlazado con la información, que se adquiera por el usuario y/o el subsistema de Realidad Aumentada

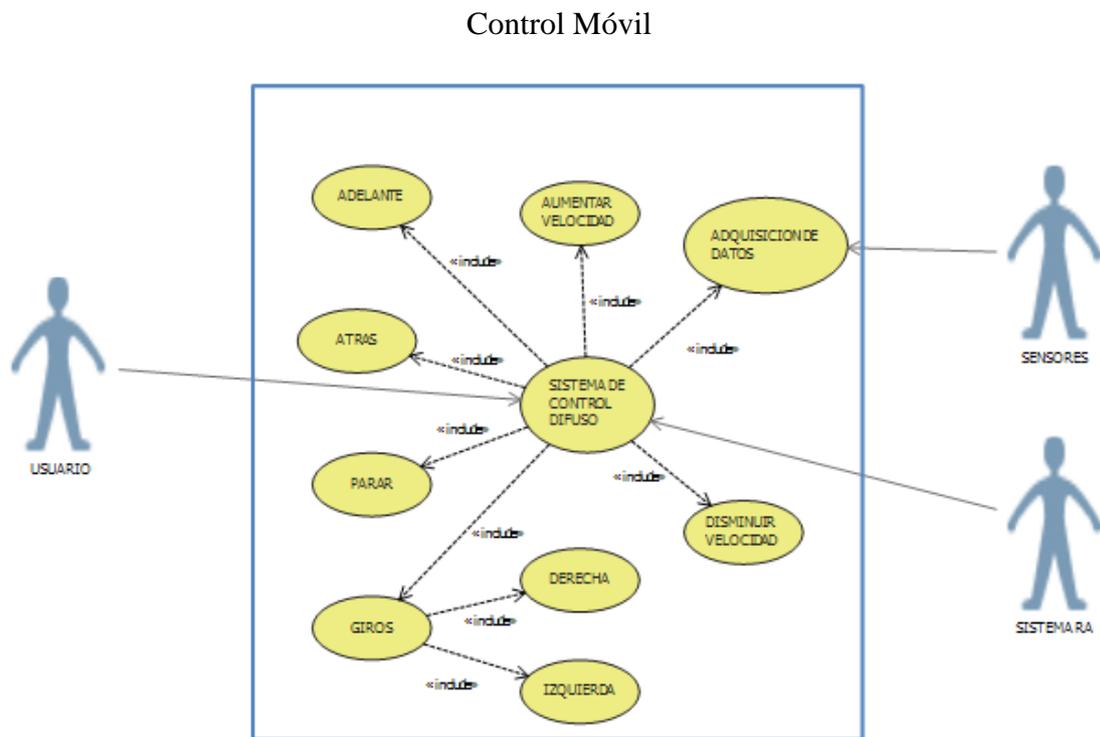


Diagrama 4. Módulo del Control Móvil

Control Móvil	
Actor Primario	Conductor (Usuario)
Actor Secundario	Sensores ultrasónicos/ Sistema de Realidad Aumentada
Descripción	Adquisición de Información, Procesamiento e Interacción
Disparador	Inicio de la Trayectoria
Pre-condiciones	1. Encendido el control móvil 2. Vinculación con el Control móvil
Post-condiciones	1. Se evaden los obstáculos con antelación. 2. Se realizan las acciones de acuerdo a la toma decisiones por parte del usuario y del Sistema AR
Flujo Normal	1. Se adquieren los datos censados (S1,E1) 2. El conductor interactúa con el robot móvil (S2,E2) 3. El Sistema de Control Difuso toma decisiones (S3,S4, E3,E4)
Flujo Alternativos	<p>S1. El sistema determina el entorno que lo rodea S2. El conductor tiene la capacidad de manejar el Dispositivo Móvil de dos formas distintas.</p> <p style="margin-left: 40px;">a) Semi- Automática b) Automática</p> <p><u>La modalidad Semi-Automática:</u> -El usuario conduce manualmente el móvil y se apoya con el Sistema Difuso. (S5)</p> <p><u>La modalidad Automática:</u> -El usuario no interviene en la conducción solo toma decisiones mediante el Sistema de Realidad Aumentada (S5)</p> <p>S3. Se realizan cálculos mediante la información obtenida por el Sistema de Realidad Aumentada y sensores para realizar acciones oportunas (depende de la señal de transito reconocida y los obstáculos que se le presente)</p> <p>S4. Usuario/Sistema Difuso puede realizar las siguientes acciones:</p> <p style="margin-left: 40px;">a) Aumentar Velocidad b) Disminuir Velocidad c) Realizar Giros (E5) d) Avanzar hacia adelante e) Retroceder f) Parar el robot móvil</p>
Errores/Excepciones	<p>E1. Error de comunicación E2. Perdida de comunicación E3. No evade oportunamente los obstáculos (es decir los objetos en movimiento) E4. Obstáculos fuera de rango de visión para los sensores E5. Tiempo estimado para dar la vuelta (depende del entorno)</p>
Prioridad	Alta

Frecuencia de uso	Alta
Reglas de negocio	No debe existir pérdida de enlace de comunicación con el robot móvil
Notas y preguntas	*: es obligatorio conectarse al robot móvil

### 4.1.3 Diagrama de Clase

El propósito de describir el Diagrama de Clase es el de representar los objetos fundamentales del sistema, es decir los que va a percibir el usuario y con los que espera tratar para completar su tarea.

Durante la etapa de análisis, el modelo de dominio es encargado de mostrar el conjunto de clases conceptuales del problema que se va a resolver, funge como base para el diseño de los objetos del software y las relaciones presentes entre sí.

¿Qué es el modelo de Dominio?

Es una representación de las cosas, entidades, ideas, del mundo real, no de los componentes del software



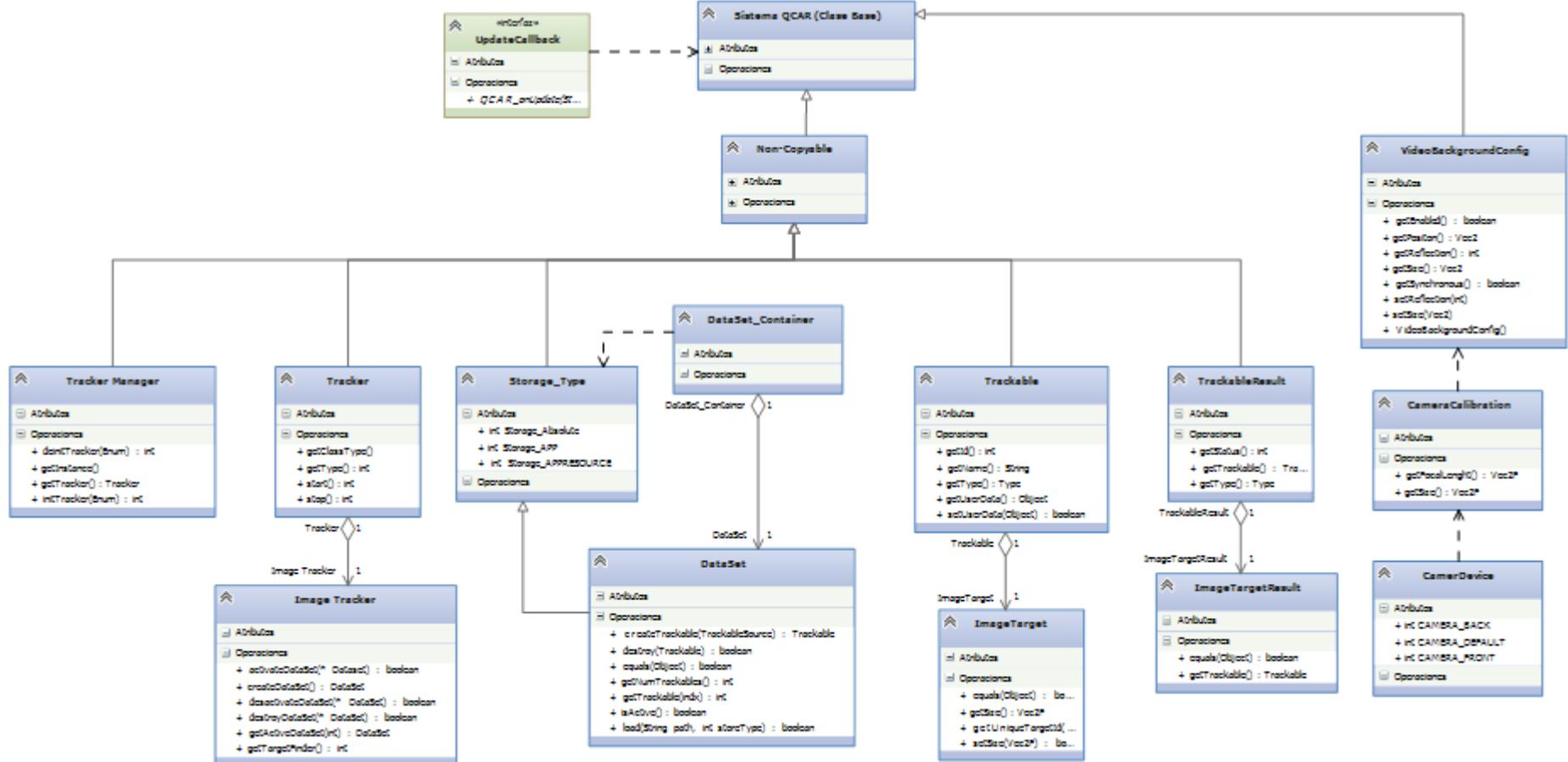


Diagrama 6. QCAR (Clases)

#### **4.1.4 Diagramas de Secuencia**

Se usan para formalizar el comportamiento del sistema, además de poder conocer la comunicación entre objetos, mediante el uso de mensajes y de esta forma los objetos generan acciones e interactúan con otros objetos.

Se debe aclarar que en un diagrama de secuencia, no se describen las situaciones erróneas, puesto que al poner todos los detalles, se generara un diagrama que no se entienda o sea difícil de leer

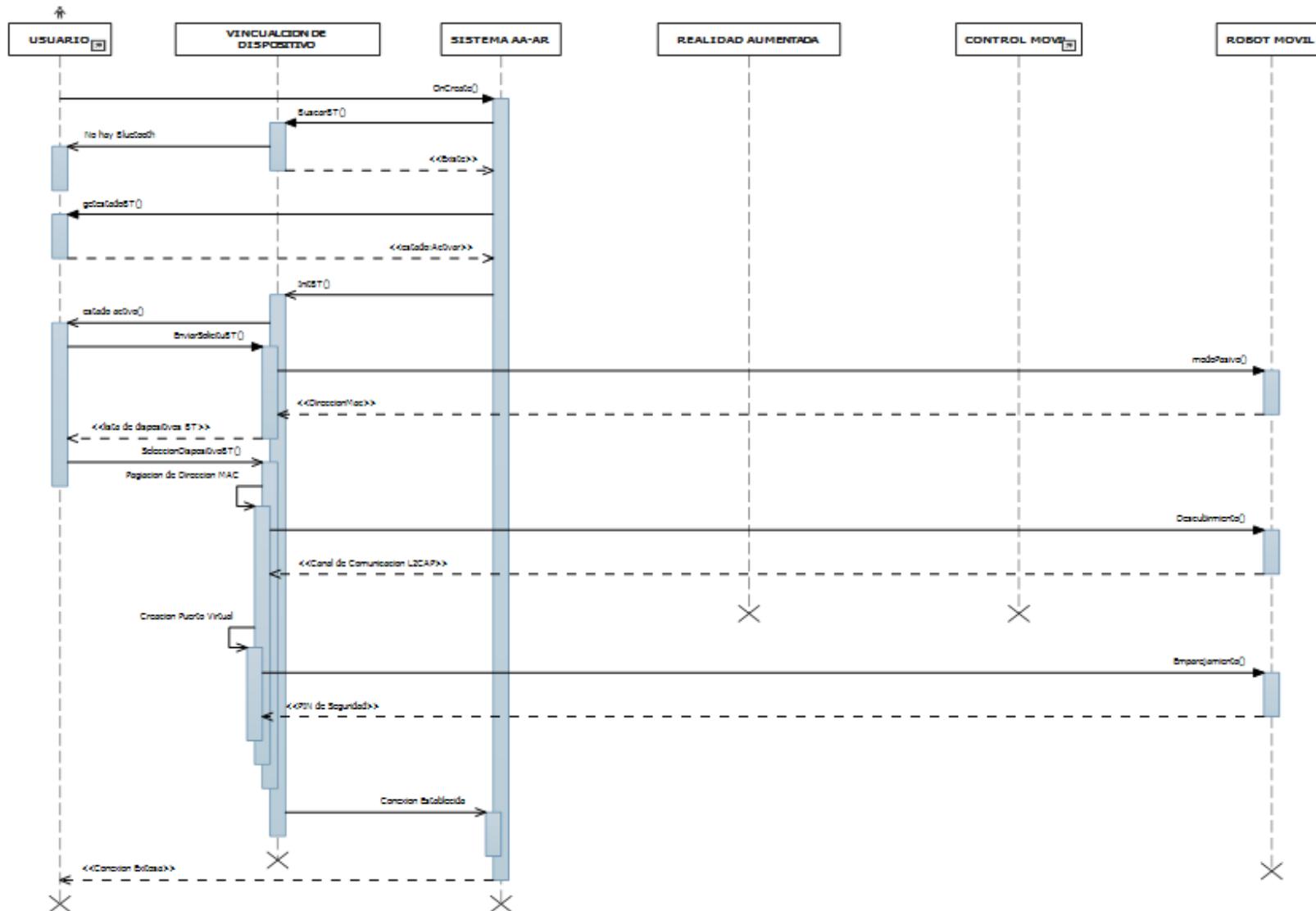


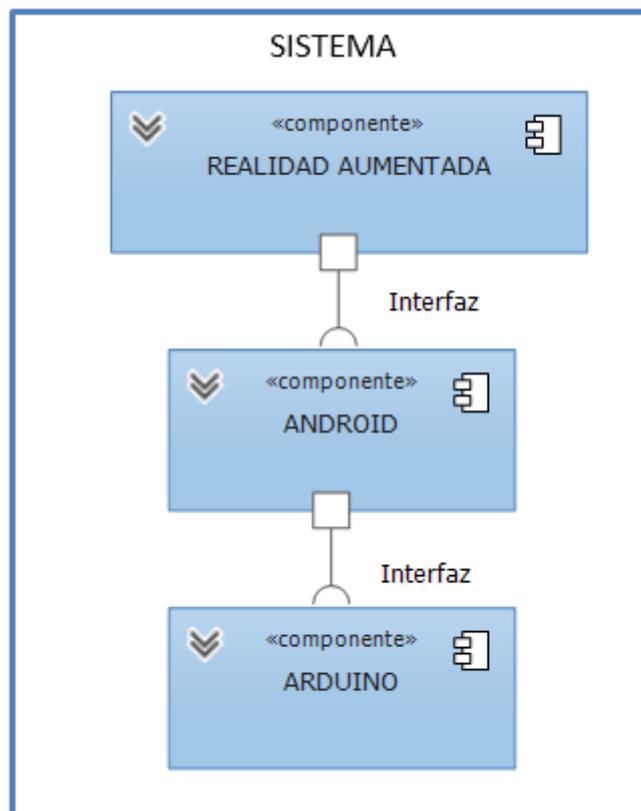
Diagrama 7. Secuencia del Sistema AA-AR

## 4.2 DESARROLLO Y RESULTADOS

El Sistema AA-AR, es creado por la unión de los siguientes elementos:

- Sistema: Es un conjunto de elementos dinámicamente interrelacionados para alcanzar objetivos
- A: Android (Google, s.f.)
- A: Arduino (Team, s.f.)
- AR: Augmented Reality (Realidad Aumentada)

Estos conceptos, forman la estructura del sistema y cada uno de ellos, se puede ver como un componente, capaz de ser modelado y que constituyen la parte física del sistema así como la interacción entre ellas.



**Diagrama 8. Componentes de Sistema AA-AR**

Por lo tanto, se van a describir conforme se desarrolle el documento, permitiendo guiar, adecuadamente desde el inicio hasta su culminación.

### 4.2.1 Realidad Aumentada

Eminente que se ha explicado la realidad aumentada, pero no desde el punto computacional, existen diferentes métodos, cada uno tienen su singularidad de implementar sus algoritmos de seguimiento (tracking) y sus diseño de patrones (marcadores), en donde se va almacenar la información suficiente para representar los elementos virtuales en el mundo real.

Por ejemplo en el servicio de paquetería, se utiliza:

- Maxicode (Ynjiun P. Wang, 2000)
- Matriz de datos (GS1 DataMatrix)
- QR (Quick Response) (Denso-Wave, 2011)

En este trabajo no se pretende hablar sobre la implementación de estos códigos, esta información se puede consultar en el documento publicado de Fiala (Fiala, 2004), donde describe todos aquellos sistemas que lo utilizan.

El Sistema AA-AR tuvo una evolución progresiva de acuerdo a las funcionalidades que se requieren, lo que generó una curva de aprendizaje, que se fue adquiriendo y perfeccionando.

Para empezar cualquier trabajo relacionado a la realidad aumentada es necesario incursionar en ArToolkit (Hirokazu Kato, 2000) que fue desarrollado por el Dr. Hirokazu Kato, pero usaremos la versión desarrollada para java que es NyArToolkit.

#### 4.2.1.1 NyArToolkit

Surge como una extensión de la versión de ArToolkit 2.72.1, su nombre se compone por el principal contribuyente que es Nyatla, tiene las mismas funcionalidades de reconocimiento de visión y proyección de modelos 3D que ArToolkit (Hayet, 2011). Tiene la característica, que sus marcadores, pueden ser de 2 tipos diferentes:

- Marcadores de dos tonalidades (blanco, negro)
- Marcadores de características naturales

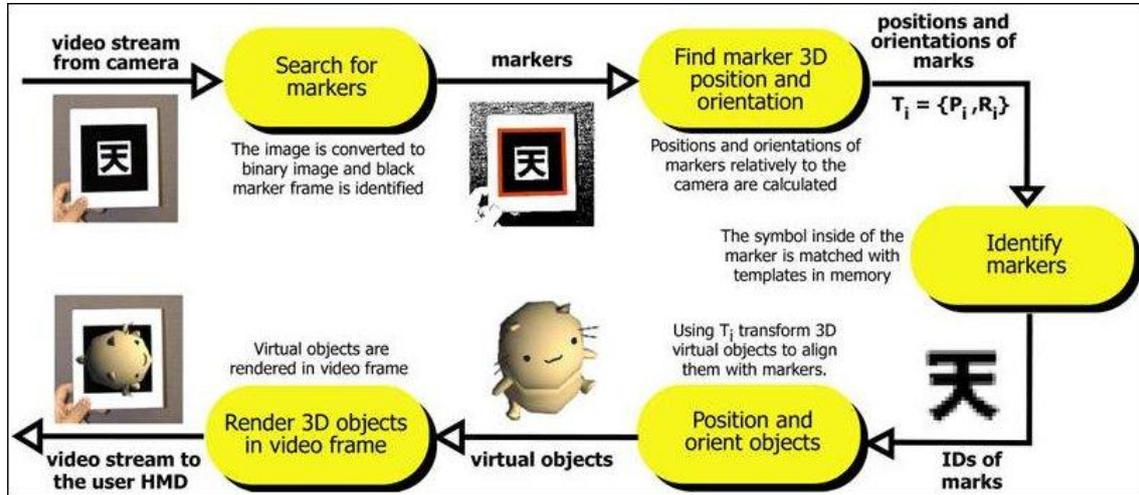
Sin embargo el software ha sido optimizado, para su fácil portabilidad entre diferentes lenguajes de programación, no significa que se mejore su velocidad de ejecución (NyARToolkit Project, s.f.)

Lenguajes de programación donde se puede utilizar:

- Java
- Android (2.3)
- C#
- SilverLight

## Algoritmo

El principio básico de ArToolkit, es el uso de marcadores en forma de cuadros, usando dos tonalidades, donde se aplican algoritmos de seguimiento (tracking).



Algoritmo 1. ARToolkit

El seguimiento funciona de la siguiente forma:

1. La cámara captura la escena del mundo real y lo envía a la computadora.
2. La imagen recibida se binariza, para buscar en cada frame del video, cualquier forma cuadrada.
3. Si el cuadrado es encontrado, el software usa transformaciones matemáticas para calcular la posición relativa de la cámara, hacia marcador (cuadrado)
4. Una vez que la posición de la cámara se conoce, también se extrae el módulo del procesamiento de las imágenes, para poder dibujar los modelos virtuales.
5. El modelo virtual es dibujado sobre el frame del video y se mantiene en el marcador. (no es necesario que este en medio del marcador)
6. El resultado final, se muestra en la pantalla de la computadora, en el instante en que el usuario ve a través de ella, ve que el modelo gráfico aumenta su realidad (hasta que pierda de vista el marcador).

Este algoritmo es la esencia para cualquier otro sistema que implemente RA, la única modificación que puede existir: es en los algoritmos que se utilizan para el procesamiento de la imagen y la extracción de características, para la detección de los marcadores.

#### 4.2.1.2 Processing con NyArToolkit

Se comentó que Processing es un IDE, que proporciona un contexto visual, mediante un lenguaje de programación, para generar trabajos profesionales. Este IDE soporta NyArToolkit, como una biblioteca externa del propio lenguaje, la implementación de la Realidad Aumentada, se hace en una computadora, mediante el seguimiento de los pasos descritos anteriormente.

Hay que resaltar algo muy importante, en los pasos del algoritmo nunca se hace mención, de los entrenamientos, configuraciones y limitaciones que existen. Dependiendo del tipo de software estos van a cambiar, en su debido momento cuando se hable de Vuforia, se notara sus diferencias respecto al NyArToolkit

Los sistemas basados en Realidad Aumentada tienen ciertas limitaciones y es necesario conocerlas, en NyArToolkit:

**I. Procesamiento de la imagen.**

Solo se representa el objeto virtual, cuando se encuentra el marcador, es decir, completamente visible, sin que nada lo esté cubriendo.

**II. Rango.**

Cuanto mayor sea el patrón físico que tenga el marcador, es detectado más lejos. Se traduce como a mayor volumen de patrones mayor probabilidad de rastrearlo. A continuación se representa algunos rangos máximos típicos, para los marcadores de diferentes tamaños, colocados perpendicularmente a la cámara, moviendo la cámara hasta llegar al punto de no reconocerlos.

Tamaño del Patrón (cm)	Dimensión Marcador (cm)
6.98	40.64
8.89	63.50
10.79	86.36
18.72	127.00

**Tabla 3. Dimensiones propuestas para el diseño de un marcador NyArToolkit**

**III. Complejidad del Patrón**

El patrón más simple, contiene grandes regiones en blanco negro (patrones de baja frecuencia) es el más eficaz. Solo desde el punto de vista del procesamiento de imágenes ya que no requiere demasiados cálculos, pero en la práctica es el más susceptible a ruido, por la escasa cantidad de patrones a analizar.

**IV. Orientación del marcador**

Los marcadores respecto a la cámara, se vuelven un poco visibles al estar inclinados, y al estar completamente horizontales, los patrones ya no son visibles y el marcador pasa hacer irreconocible.

**V. Condiciones de Iluminación.**

Es el problema que sufre toda imagen al momento de ser procesada, ya que al existir una gran cantidad de iluminación, aumenta el grado de no encontrar los patrones o inclusive no reconocer ningún marcador, para evitar esto, se recomienda usar un material sin superficies brillosas.

**Desarrollo de Realidad Aumentada.**

**i. Diseño de Marcadores**

Los marcadores, son la parte fundamental para representar los modelos virtuales y estos deben cumplir ciertas características, de acuerdo a los puntos II y III. Se establecen 10 marcadores ya que deseamos representar en cada uno, las señales de tránsito que se encuentran en un entorno urbano.

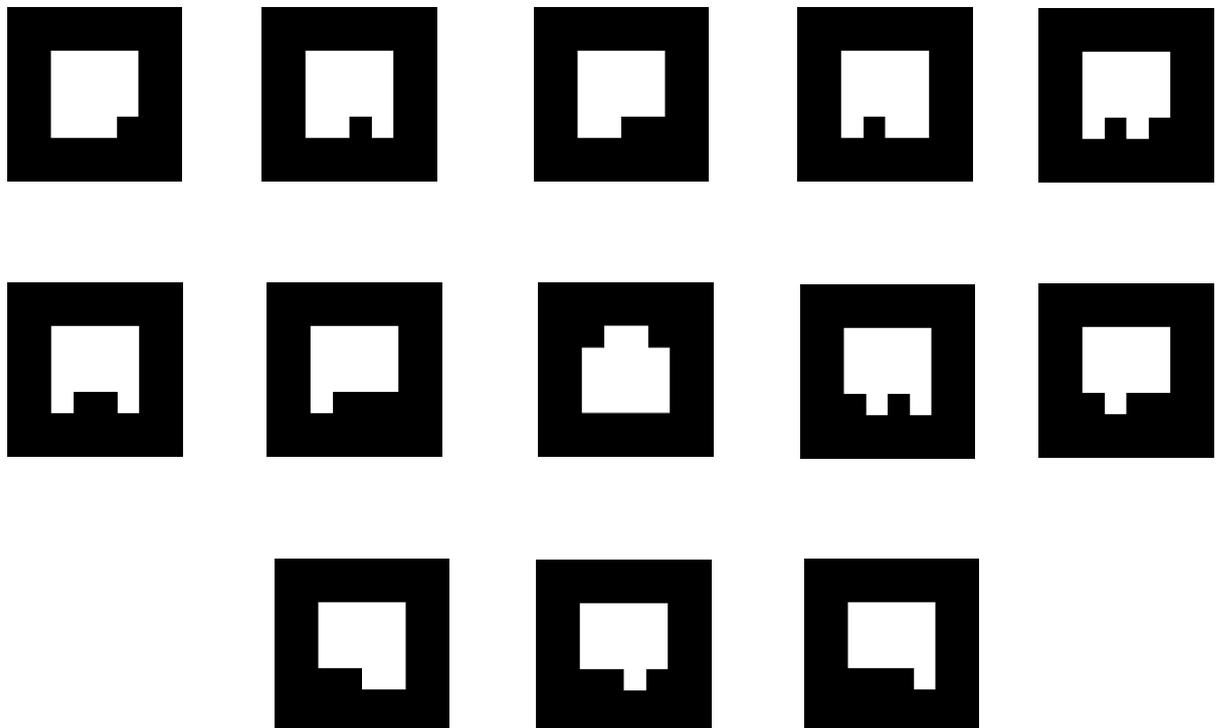


Figura 44. Marcadores propuestos para el sistema AA-AR

**ii. Representación de Modelos Virtuales**

Cada uno de estos marcadores, pueden representar varios modelos al mismo tiempo, pero en este proyecto solo se representara un modelo por cada marcador, puesto que cada señal de tránsito representa una información independiente, no sería conveniente representar dos modelos al mismo tiempo, puesto que el conductor y el sistema AA-AR no identificarían cual es la correcta, esto va cambiar conforme a la situación del entorno en que se encuentre.

Debemos definir que señales de tránsito van aumentar nuestra realidad, para brindarnos información oportuna, al manejar en un entorno urbano.

En la tabla siguiente se presenta las 13 señales de tránsito propuestas, dichas señales, sirven como representación para la interacción con la realidad aumentada

Señales de Tránsito	
Sentido de Circulación	Prohibido sentido de circulación
	
Vuelta a la izquierda permitida	Prohibido vuelta a la izquierda
	
Vuelta a la derecha permitida	Prohibido vuelta a la derecha
	
Hospital o puesto sanitario	Estación de servicio /gasolina
	
Estación del metro	Obligatoriamente hay que parar
	
Límite de Velocidad	Informan el estado del camino (curva peligrosa)
	

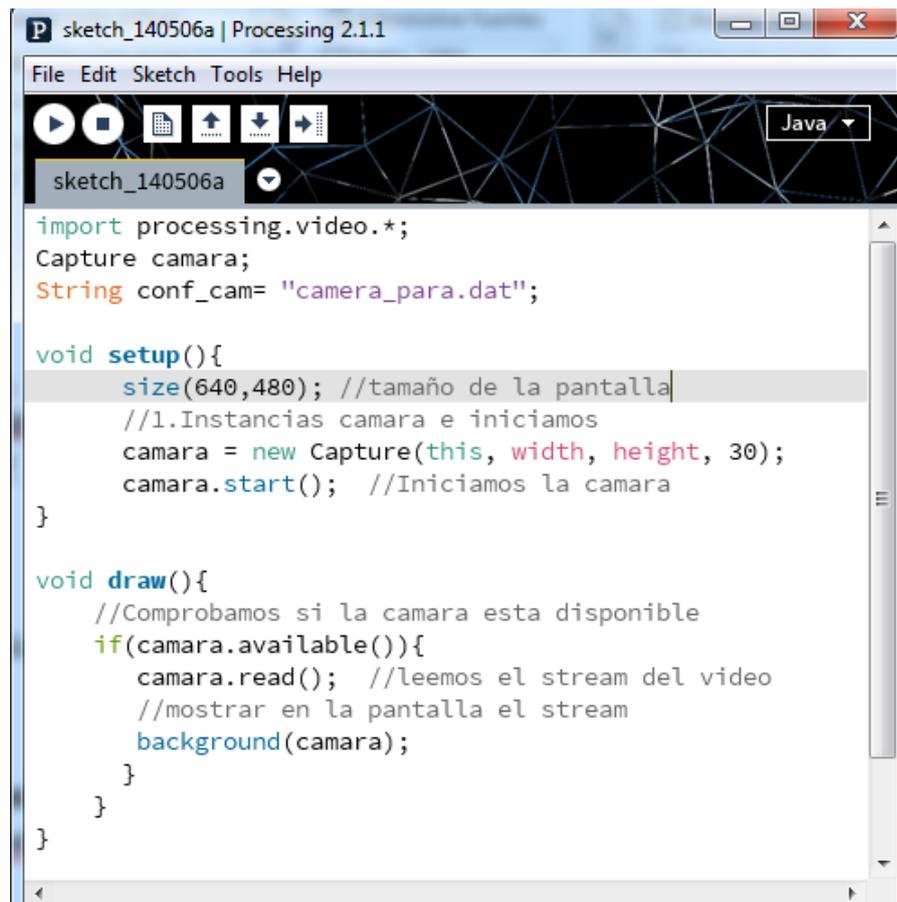
Figura 45. Señales de tránsito propuestas para el sistema AA-AR

## 1. La cámara captura el entorno.

Un sistema que utiliza realidad aumentada, debe disponer esencialmente de un sistema de captura que es la cámara, ya que desempeña la entrada del sistema.

Se crea un objeto tipo Capture que va a representar la cámara del sistema, este mismo pasa dentro de 3 estados los cuales son:

- Crear un instancia de la cámara
- Establecer la configuración (resolución de captura, frames por segundos)
- Iniciar la cámara (Encender)
- Comprobar si la cámara está disponible en caso contrario esperar o ver la existencia de más cámaras
- Leemos el flujo “frames” de la cámara (empezar a capturar)
- Los frames capturados se muestran en la aplicación.

The image shows a screenshot of the Processing IDE window titled 'sketch\_140506a | Processing 2.1.1'. The interface includes a menu bar (File, Edit, Sketch, Tools, Help), a toolbar with icons for play, stop, and file operations, and a dropdown menu for the language (Java). The main area displays the following code:

```
import processing.video.*;
Capture camara;
String conf_cam= "camera_para.dat";

void setup(){
  size(640,480); //tamaño de la pantalla
  //1.Instancias camara e iniciamos
  camara = new Capture(this, width, height, 30);
  camara.start(); //Iniciamos la camara
}

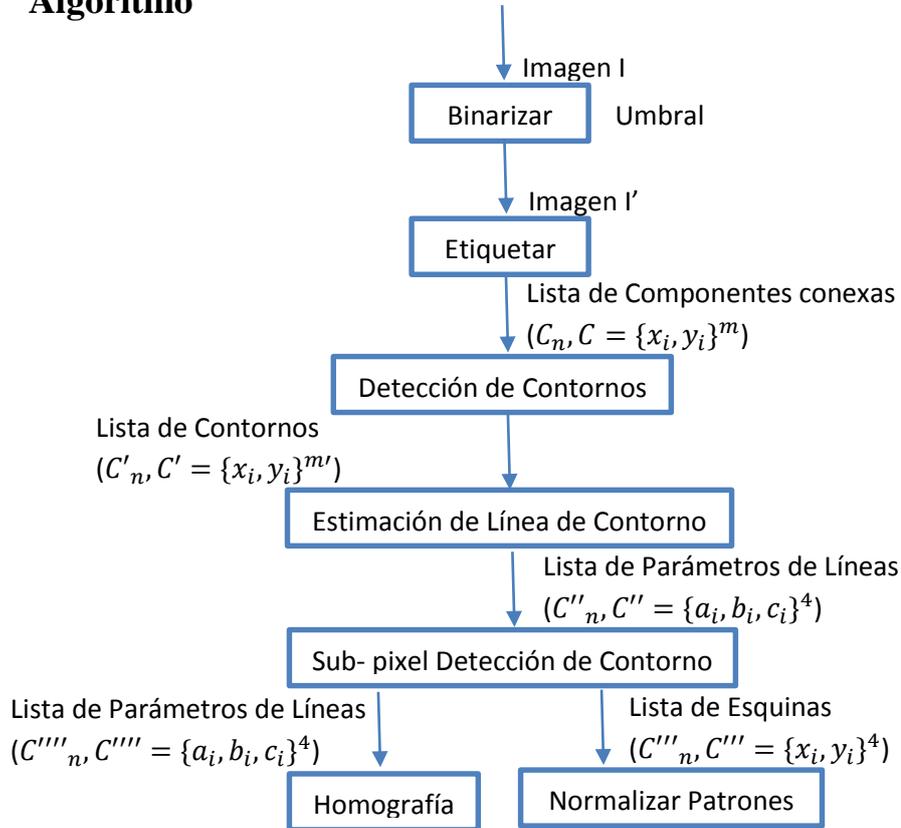
void draw(){
  //Comprobamos si la camara esta disponible
  if(camara.available()){
    camara.read(); //leemos el stream del video
    //mostrar en la pantalla el stream
    background(camara);
  }
}
```

Algoritmo 2. Configuración de la cámara

## 2. Extracción de información del entorno

Se basa en la detección de esquinas con una rápida estimación.

### Algoritmo



Algoritmo 3. Primera fase de extracción de características del entorno

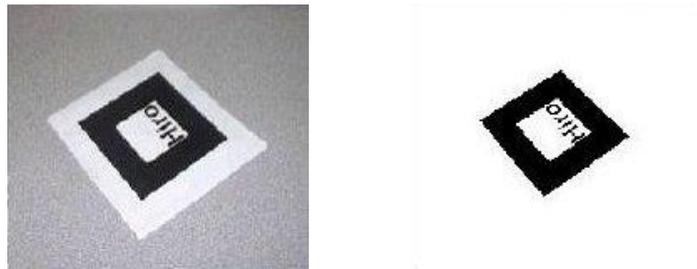
### Funcionamiento:

Cuando se obtiene un frame desde la cámara, esta puede crear una imagen, que contenga una gama de colores. Estas imágenes son discretas, para el uso del procesamiento de la imagen; principalmente cada pixel de la imagen está constituido por 3 canales (Red, Green, Blue), cada uno de ellos puede tener valores de 0 – 255 distintos.

Los marcadores diseñados, solo tiene dos tonalidades blanco y negro, para realizar un procesamiento de imagen más rápido, se aplica el método de binarización global, esto significa que se aplica un umbral (limite) a toda la imagen, el cual permite diferenciar entre lo que se desea analizar y lo que no es de interés.

```
function y = Binarizacion(imagen, umbral)
im=imread(imagen);
imD=double(im);
[f,c]=size(imD);
for i=1:f
    for j=1:c
        if imD(i,j)<=umbral
            nuevaI(i,j) = 0;
        else
            nuevaI(i,j) = 255;
        end
    end
end
imB = uint8(nuevaI);
imshow(imB);
y = nuevaI;
```

**Algoritmo 4. Binarización**



**Figura 46. Imagen Original e Imagen con Umbral**

Trabajar con imágenes binarias resulta muy interesante por 2 motivos:

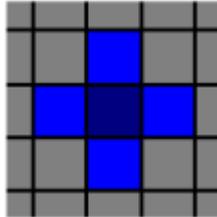
1. Reduce al mínimo el área de estudio y solo filtra la información requerida
2. Porque las propiedades geométricas y topológicas de los objetos presentes en la imagen, se pueden obtener más rápidamente

### **Definiciones Topológicas básicas.**

La extracción de los objetos de interés de una imagen binaria, técnica conocida como segmentación, asume que los puntos pertenecientes a los objetos de interés están cercanos entre sí. Por ello, es necesario introducir algunas definiciones básicas que formalicen este concepto de proximidad espacial. (Véase por ejemplo “Kong, 1989”, para una representación más detallada).

## Vecindad.

Un pixel  $P = (x, y)$  de la imagen tiene 2 vecinos horizontales (este y oeste) y 2 verticales (norte y sur). Las coordenadas de estos pixeles son  $(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$  y  $(x, y-1)$ . Este conjunto de pixeles son los 4-vecinos  $N_4(P)$  de  $(x, y)$

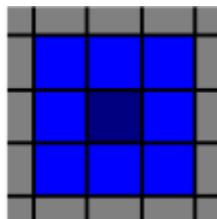


**Figura 47. Vecindad 4**

Píxeles considerados vecinos de  $P(x,y)$  en la 4-vecindad

$$\mathcal{N}_4(P) = \{(x,y-1), (x,y+1), (x-1,y), (x+1,y)\}$$

Los 4 vecinos situados en las diagonales  $N_D(P)$  están en las coordenadas  $(x-1, y-1)$ ,  $(x-1, y+1)$ ,  $(x+1, y-1)$  y  $(x+1, y+1)$ . El conjunto  $N_D(P) \cup N_4(P)$  formado por la unión de los anteriores constituye la 8-vecindad  $N_8(P)$  del pixel  $P(x, y)$



**Figura 48. Vecindad 8**

En la 8-vecindad se incluyen además como vecinos los píxeles en las diagonales

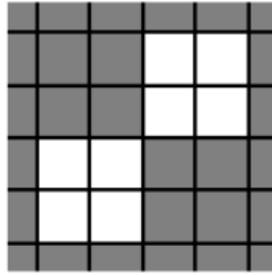
$$\mathcal{N}_8(P) = \mathcal{N}_4(P) \cup \{(x-1,y-1), (x-1,y+1), (x+1,y-1), (x+1,y+1)\}$$

## Adyacencia.

Dos píxeles son adyacentes si:

- Son vecinos (por ejemplo  $N_4(P)$ ,  $N_8(P)$ , .... )
- Sus valores satisfacen algún criterio específico de similitud, típicamente la igualdad.

Notaremos como  $\gamma$  el conjunto de valores de los píxeles usados para definir la similitud: por ejemplo, los píxeles que tras una binarización tienen el valor 1, tiene a 1 como valor de similitud  $\gamma = \{1\}$



**Figura 49. Adyacencia**

En esta porción de imagen se considerarán una o dos componentes conexas dependiendo del tipo de adyacencia que se haya predefinido. Si se considera 8-adyacencia, los píxeles en contacto únicamente por el vértice son vecinos y formarían parte de una misma región de 8 píxeles. Al contrario, si se considera 4-adyacencia pertenecerían a dos regiones distintas, de 4 píxeles cada una.

Consideramos 2 tipos básicos de adyacencia:

- 4-adyacencia: 2 píxeles  $P=(x, y)$  y  $Q=(w, z)$  con  $I(x, y)$  y  $I(w, z) \in \gamma$ , son 4-adyacencia si  $P = N_4(Q)$
- 8-adyacencia: La definición es análoga a la de la 4-adyacencia sustituyendo la 4-vecindad  $N_4(P)$  por la 8-vecindad  $N_8(P)$

## Camino

Un camino de longitud  $n$  desde el píxel  $P_0 = (x_0, y_0)$  al píxel  $P_n = (x_n, y_n)$  es una secuencia de píxeles distintos  $(P_0, P_1, \dots, P_n)$  donde  $P_i$  es adyacente a  $P_{i-1}$  para  $1 \leq i \leq n$ .

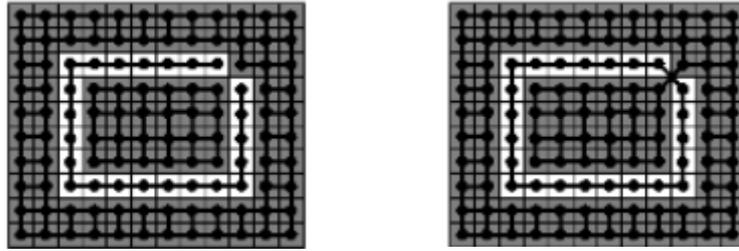
Si  $P_0 = P_n$  el camino es cerrado. Dependiendo del tipo de adyacencia utilizado tendremos 4-caminos, 8-caminos.

Dos píxeles  $P$  y  $Q$  están conectados si existe un camino entre ellos.

## Regiones y contornos.

Para un conjunto  $S$  de píxeles.

- Dado un píxel  $P \in S$ , el subconjunto de todos los píxeles  $Q_i \in S$  conectados a  $P$  forman una componente conexa de  $S$
- Si  $S$  solo tiene una componente conexa,  $S$  es un conjunto conexo.



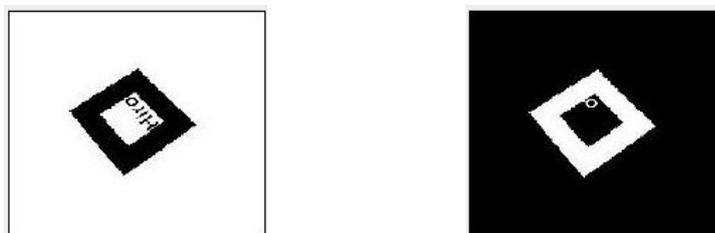
**Figura 50. Contorno o Frontera**

Un subconjunto  $R$  de píxeles de una imagen es una región si  $R$  es un conjunto conexo. Un contorno o frontera de una región  $\mathfrak{R}$  es la curva digital formada por el conjunto de píxeles  $P \in \mathfrak{R}$  que tienen al menos un vecino  $Q$  tal que  $Q \notin \mathfrak{R}$

Un problema importante que debe comentarse al trabajar con curvas es el relativo Teorema de la curva de Jordan (Francisco Garcia Arenas, 1998) y su extensión a imágenes digitales. Según este teorema, una curva cerrada  $C$  divide los puntos de la imagen no pertenecientes a  $C$  en dos conjuntos distintos sin puntos en común (conjunto interior, conjunto exterior) para los cuales esta curva es su frontera común. Es fácil comprobar que si establecemos el mismo tipo de adyacencia para todos los píxeles de la imagen, este teorema no se cumple, lo que tiene serias consecuencias a la hora de desarrollar los algoritmos. Para evitar este problema, basta con establecer un tipo de adyacencia diferente para los píxeles correspondientes al fondo (0) respecto a los objetos (1), de tal forma que si elegimos 4-adyacencia para el fondo, elegiremos 8-adyacencia para los objetos, y viceversa (Rosenfeld, Connectivity in digital pictures., 1970) (Rosenfeld, Digital Topology, 1974)

### **Etiquetado de componentes conexas.**

Entre las regiones presentes en una imagen binaria, es lógico suponer que se encuentre aquellas que corresponden con los objetos de interés en la escena, sin más que atender a un criterio de proximidad espacial.



**Figura 51. Imagen con Umbral e Imagen con sus componentes conexas**

Figura 51. Se van a extraer el área o el centro de gravedad a partir de la simple binarización de la imagen original se procesaran todos los píxeles negros como si constituyen parte del mismo objeto. En la imagen el área total dependerá del tamaño de la imagen a procesar y el centro de

gravedad del conjunto se hallaría en la posición representada en la región interna del cuadrado. Como las aplicaciones generalmente precisan de una extracción individualizada de las características de cada uno de los componentes presentes en la imagen, es preciso previamente establecer de alguna forma cada una de las regiones o componentes conexas que en ella aparecen. Esta operación es lo que se conoce como etiquetado; El etiquetado permite extraer las características individuales de cada objeto estableciendo las áreas individuales en píxeles. Estas áreas individuales ya permiten establecer ciertos criterios para detectar o reconocer determinados objetos. Por ejemplo, se podría establecer que existe un elemento significativamente más grande que el resto y que este tiene su centro de gravedad

Para poder identificar los píxeles que pertenecen a cada una de las regiones y, por tanto, asociar estas regiones a los objetos para extraer sus características, es necesario obtener las *componentes conexas* presentes en la imagen. Es obvio que, si en la imagen aparecen varias regiones, no podrá obtener, por ejemplo, su área o centro de gravedad, hasta que estas no estén identificadas de forma individual. Este proceso se conoce como etiquetado y consiste en definitiva en asignar una misma etiqueta a cada uno de los píxeles que pertenecen a una componente conexa. La etiqueta será un valor numérico que permite identificar todos los píxeles que forman parte de una misma región. De esta forma, la imagen de etiquetas obtenida permitirá la extracción individualizada de características de cada una de las regiones conexas de la imagen.

La definición de un componente conexo dependerá del tipo de adyacencia utilizado. Si utilizamos la 4-adyacencia, las regiones resultantes serán 4-conexas y si utilizamos la 8-adyacencia las regiones serán 8-conexas.

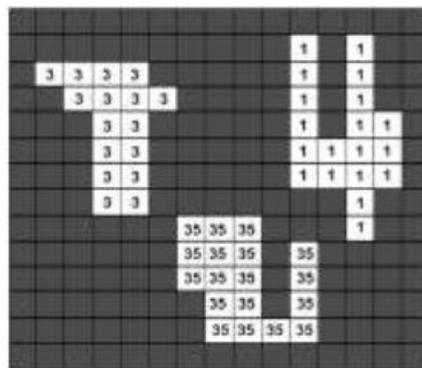


Figura 52. Etiquetado de contornos

## Seguidor de contornos y etiquetado de regiones simultáneos.

El algoritmo propuesto en (Chang 2004) tiene la característica de no solo obtener el contorno exterior y los contornos interiores de una región, sino que simultáneamente realiza el etiquetado de la imagen binaria, y todo ello en un único barrido de la imagen. Actualmente pasar por ser uno de los algoritmos más rápidos de etiquetado, y es el algoritmo de referencia utilizado.

Supondremos que, en la imagen binaria  $B(x, y)$ , las regiones tienen valor 1 y el fondo 0. Para evitar que el algoritmo tenga que comprobar que no se accede a posiciones fuera de los límites de la imagen, se supondrá que no existe ninguna región en contacto con estos límites o se pondrán directamente a 0 los píxeles en los bordes de la imagen.

El algoritmo puede explicarse en función de las diferentes decisiones que se adoptan según las propiedades de los diferentes puntos de una región. Al iniciarse el barrido de una fila desactivamos una bandera,  $etiq\_activa=0$ , que nos servirá para indicar si una región está siendo visitada. Así, durante el barrido de una fila de la imagen binaria  $B(x, y)$ :

1. Encontramos por primera vez un punto  $P_e$  de una región (Figura 63a). En este momento creamos una nueva etiqueta  $netiq$  con la que será etiquetada la región y activamos  $etiq\_activa=netiq$ . El punto  $P_e$  es un punto del contorno exterior de la región  $netiq$ . Mediante una llamada a la función Seguidor de contorno (Figura 64) se creará una lista con los puntos de este contorno y los etiquetará con el valor  $etiq\_activa$  en la imagen de etiqueta  $E$  (Figura 63b). Esta función, con independencia de si el contorno es externo o interno, marca en  $E$  con un valor bandera, por ejemplo -1, todos los 8-vecinos del contorno que no pertenecen a la región. El marcado prevendrá de que un contorno interno se recorra más de una vez.
2. Estando la bandera  $etiq\_activa$  activada,  $etiq\_activa=netiq$ , encontramos un punto  $B(x, y)$  a 1: entonces  $E(x, y) = etiq\_activa$  (Figura 63d)
3. Estando la bandera  $etiq\_activa$  activada,  $etiq\_activa=netiq$ , encontramos un punto  $B(x, y)$  a 0:
  - Si su etiqueta es -1 (Figura 63c) implica que es un punto vecino de los contornos ya recorridos de la región, externo o interno. No se hace nada. La bandera se desactiva,  $etiq\_activa = 0$ .
  - Si su etiqueta es 0 (Figura 63d) significa que es el primer punto encontrado de uno de los posibles agujeros de la región. El punto inmediatamente superior  $P_i=(x-1, y)$  por fuerza es un punto de un contorno interior. Una llamada a Seguidor de contorno (Figura 64) creará una lista con los puntos de este contorno interno y los etiquetará con el valor  $etiq\_activa$  en la imagen de etiquetas  $E$  (Figura 63e)

4. Estando *etiq\_activa* desactivada encontramos un punto  $B(x, y)$  del contorno exterior o interior de la región etiquetado con un valor *netiq*. Por ejemplo, el punto  $Q_i$  en (Figura 63). Obviamente esto implica que el punto anterior  $B(x, y-1)$  está a 0. Activaremos la bandera,  $etiq_activa=E(x, y)$ , que indica que una región está siendo visitada.

La función `SEGUIDORCONTORNO()` realiza varias acciones:

- Obtiene los contornos externo e internos de una región.
- Marca con el valor de etiqueta correspondiente los puntos de contorno
- Marca con una etiqueta bandera -1, los puntos 8-vecinos de los contornos que pertenecen al fondo.

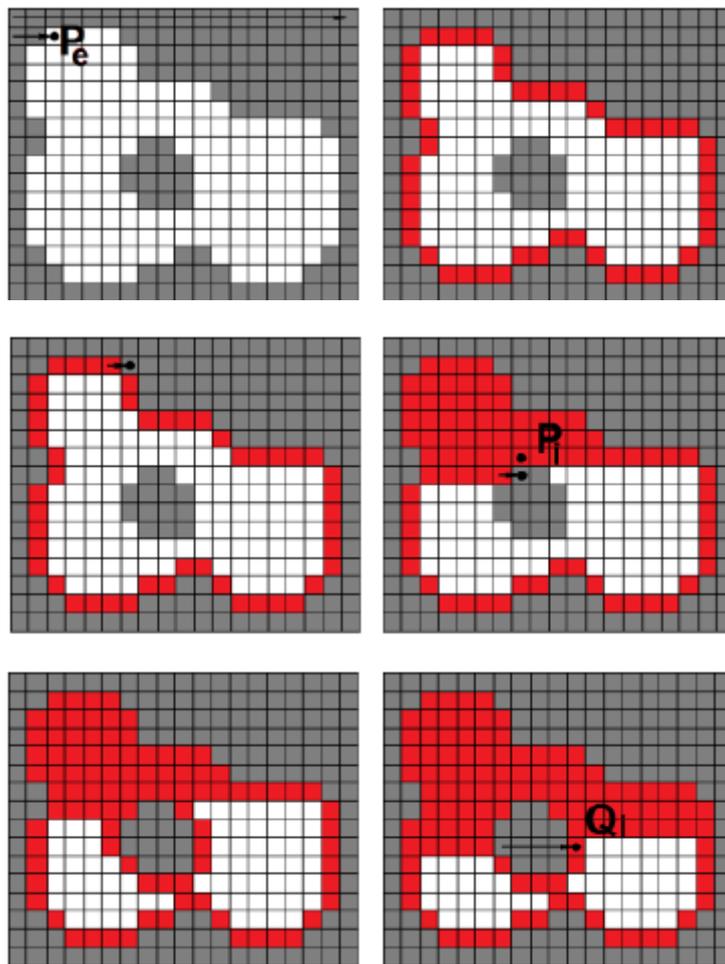


Figura 53. Seguidor de contornos y etiquetado

La función `SEGUIDORCONTORNO()` debe garantizar que no termina de forma prematura al visitar de nuevo el punto de inicio. La condición de parada consiste en salvaguardar: el punto inicial  $P_e$  o  $P_i$  (que guardara en la última posición del contorno, punto  $P_{fin}$ ) y su sucesor  $P_{ini}$  (que se guardara en la primera posición). La parada ocurre cuando en el transcurso del seguimiento el punto actual  $P_{act}$  y su sucesor  $P_{sig}$  coinciden respectivamente con  $P_{fin}$  y  $P_{ini}$ . Además, esta función debe contemplar la presencia de una región monopixel: se detecta fácilmente sin más que comprobar al inicio que  $P_{ini}$  coincide con  $P_{fin}$

El pseudocódigo puede verse en el siguiente Algoritmo:

```

→ B: Imagen binaria de tamaño (filas,columnas).
← E: Imagen etiquetada de tamaño (filas,columnas)
C: Estructura de datos con el contorno exterior y los interiores de cada región
SEGUIDORETIQUETADOSIMULTANEOS(B)
  Inicializar netiq=0, imagen de etiquetas E=0, C=φ
  Poner a 0 B(:,1), B(1,:), B(filas,:) y B(:,columnas) %A 0 los bordes de la imagen
  Para cada x, x=2,...filas-1
    etiq_activa=0
    Para cada y, y=2...columnas-1
      Si B(x,y)==1
        Si etiq_activa≠0 %Etiquetando la región etiq_activa
          E(x,y)=etiq_activa
        en caso contrario
          etiq_activa=E(x,y)
          Si etiq_activa==0 %Nueva región
            netiq=netiq+1
            etiq_activa=netiq
             $P_e=(x,y)$  %Punto inicial del contorno exterior
             $c_e=SEGUIDORCONTORNO(B,E,etiq_activa,P_e,0)$ 
            Añadir  $c_e$  a C(etiq_activa)
            E(x,y)=etiq_activa
          Fin
        Fin
      en caso contrario
        Si etiq_activa≠0 %Acabamos de abandonar la región
          Si E(x,y)==0 %Primer punto encontrado en agujero interno
             $P_i=(x-1,y)$  %Punto inicial del contorno interior
             $c_i=SEGUIDORCONTORNO(B,E,etiq_activa,P_i,3)$ 
            Añadir  $c_i$  a C(etiq_activa)
          Fin
        etiq_activa=0
      Fin
    Fin
  Fin
  Fin
  Devolver E, C

```

Algoritmo 5. Seguidor de contornos y etiquetado simultáneo

La función `SEGUIDORCONTORNO()` debe decidir para cada punto  $P_{act}$  del contorno donde buscar su sucesor  $P_{sig}$ . Salvo en los casos particulares de los puntos de entrada a la función, cualquiera que sea la configuración del punto  $P_{act}$ , el punto sucesor  $P_{sig}$  se hallara buscando en el sentido horario uno de sus 8-vecinos.

```

→ B:      Imagen binaria de tamaño (filas, columnas)
E:      Imagen etiquetada de tamaño (filas, columnas). Se modifica.
etiq:   Etiqueta de la región
Pfin=(x,y): Coordenadas del punto de entrada. Se almacena al final del contorno
codigo:  Código inicial de búsqueda respecto a Pact (0 exterior, 3 interior)
← c:     Vector con las coordenadas del contorno
SEGUIDORCONTORNO(B,E,etiq,Pfin,codigo)
  Inicializar c=0
  contador=1
  (Psig,codigo)=SIGUIENTEPUNTO(B,E,Pfin,codigo)
  c(contador)=Psig %Primer punto del contorno
  Si Psig ≠ Pfin %El punto no es un punto aislado
    seguir=1
    Mientras seguir==1 %Búsqueda en los 7 vecinos
      E(x,y)=etiq %Etiquetamos los puntos de contorno
      codigo=MOD(codigo+6,8)
      Pant=Psig
      (Psig,codigo)=SIGUIENTEPUNTO(B,E,Pant,codigo)
      Si Psig==c(1) AND Pant==Pfin %Se cumple condición de parada
        seguir=0
      en caso contrario
        contador=contador+1
        c(contador)=Psig
    Fin
  Fin
Fin
Devolver c
    
```

Algoritmo 6. Método SeguidorContorno

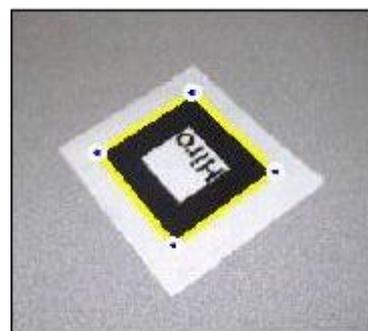
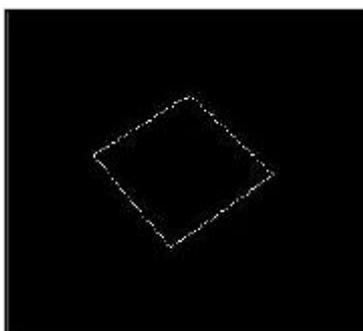


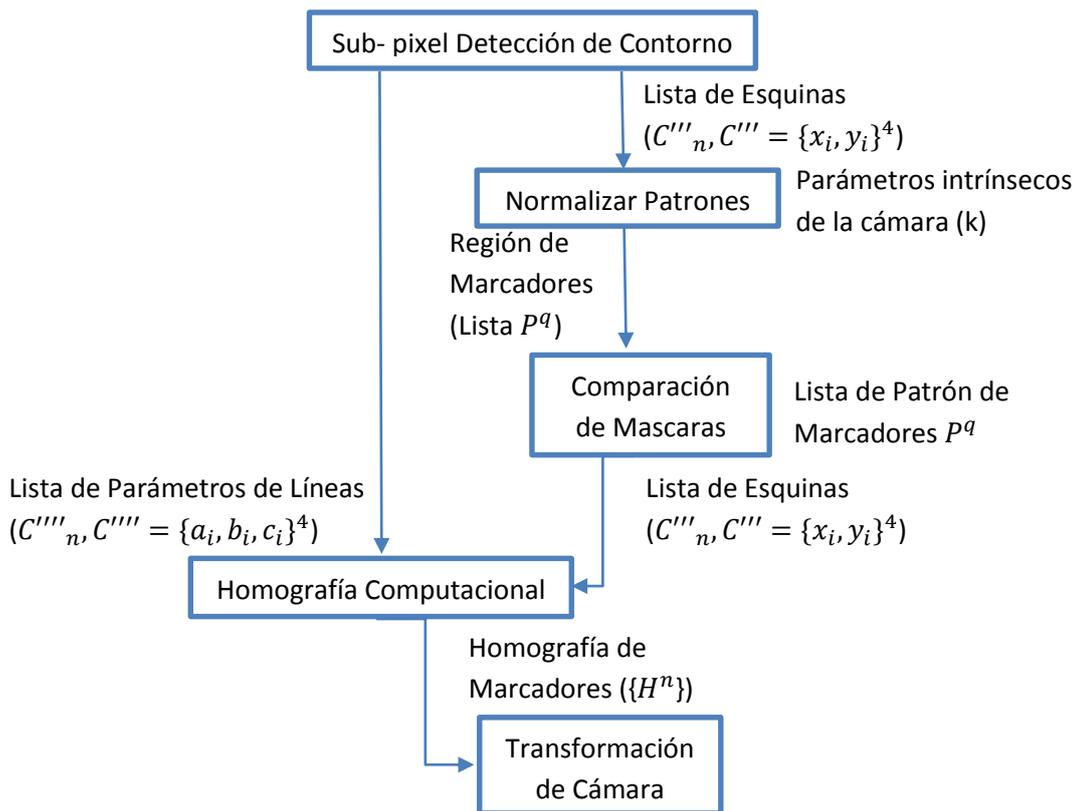
Figura 54. Extracción de Contornos, bordes y esquinas

### 3. Transformaciones matemáticas

Se puede traducir como una función que mapea un conjunto X en otro conjunto o sobre sí mismo. Sin embargo a menudo el conjunto X posee alguna estructura geométrica adicional y el término “transformación” se refiere a una función de X sobre si misma que conserva dicha estructura.

Nos enfocaremos en las transformaciones geométricas: rotaciones y traslaciones. Estas se pueden realizar en el espacio euclidiano, especialmente en 2 y 3 dimensiones. Estas son operaciones que se pueden llevar a cabo utilizando algebra lineal y ser descritas de manera explícita utilizando matrices.

#### Algoritmo



Algoritmo 7. Segunda fase de extracción de características del entorno

## **Funcionamiento.**

En el procedimiento de la obtención de los sub-píxel de la detección de Contorno, se obtienen como salida una lista de esquinas, realmente son los vértices del marcador, usualmente se van a usar como referencia para conocer donde se ubica el marcador, para esto es necesario normalizar los patrones obtenidos.

Una vez obtenidos los datos anteriores, a la imagen se realiza una normalización del carácter. Para esto se centra la imagen en un plano cuyas dimensiones son iguales a  $m \times n$  donde  $m$  es igual al tamaño del lado más largo del carácter a normalizar. Después esta imagen se escala al tamaño del marcador y se le aplicara un filtro para eliminar el ruido que pueda quedar con la imagen para terminar con un suavizado. Es necesario realizar una calibración de la cámara

## **Calibración de la cámara**

Es el proceso que permite estimar los valores de los parámetros intrínsecos y extrínsecos que definen las condiciones de formación de la imagen dentro del campo de la visión. Se puede definir como un paso necesario para conocer como una cámara proyecta un objeto, en el plano de la imagen y extraer información métrica (distancia) a partir de las imágenes (Identificación óptica de la posición y orientación de un vehículo aéreo no tripulado).

## **Parámetros de la cámara:**

- **Parámetros Intrínsecos:**

Son aquellos que definen la geometría interna y la óptica de la cámara. Son constantes en tanto no varíen las características y posiciones relativas entre la óptica y el sensor de la imagen.

### **Distancia Focal:**

La distancia focal de una cámara es la distancia existente entre esta y su foco. Puesto que la lente de la cámara es convergente, la distancia focal es positiva.

En la figura 55 se puede ver el esquema de la cámara, en la que el foco se encuentra a una distancia del plano de la imagen igual a la distancia focal.

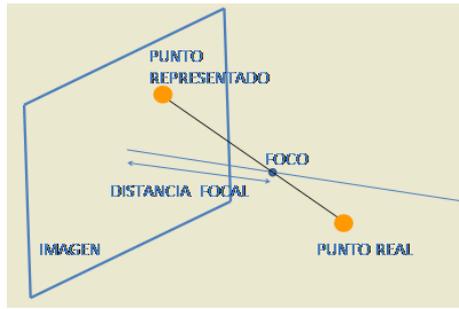


Figura 55. Parámetros intrínsecos

Para el cálculo de la distancia focal de una cámara se van a utilizar dos objetos de referencia y sus representaciones en la imagen de la cámara como se puede observar en la siguiente figura.

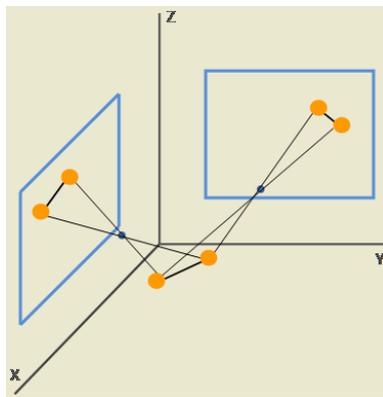


Figura 56. Distancia focal

El foco de una cámara es el punto donde se intersectan de dos o más rectas que vayan desde un objeto hasta su representación en la cámara.

Para empezar, se emplean dos objetos de referencia, se calcula, para cada objeto, la recta que une el objeto en su posición real con su representación en la cámara. Estas rectas pasaran ambas por el foco, así que será en la intersección de ambas donde se obtendrá este.

La intersección de estas rectas, existe imprecisiones en la estima de algún parámetro de la cámara, es probable que no sea exacta, por lo que se calculara el punto situado a la menor distancia posible en dos rectas que se cruzan.

El punto obtenido es el foco de la cámara, así que para calcular la distancia focal de dicha cámara basta con calcular la distancia desde el foco hasta el centro de la cámara. Esta será la distancia focal estimada puesto que, por los errores explicados no pueda ser exacta.

Este error depende de la distancia entre las rectas que se cruzan, la desviación del foco estimado con respecto al eje óptico y la distancia entre el foco estimado y el punto del eje óptico que se encuentra a una distancia igual a la focal de la cámara.

### Factor de conversión pixel-milímetros.

Las coordenadas 2D obtenidas están expresadas en forma de píxeles, así que se ha hallado el número de píxeles por milímetro que usan las cámaras para transformarlas a medidas del Sistema Internacional de Unidades (SI).

Con dos objetos de referencia colocados en un plano paralelo al plano de la cámara se han obtenido resultados de las medidas conseguidas en las imágenes comparándolas con las medidas reales. El sistema empleado queda así.

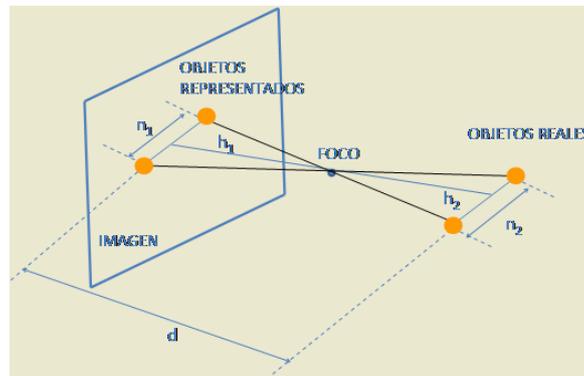


Figura 57. Modelo de transformación de coordenadas

En este caso se busca el valor  $n_1$ , puesto que será la relación  $n_1/n_2$  la que permita conocer el tamaño del pixel.

Los datos conocidos son:

- D: distancia entre el plano de la cámara y el plano de la imagen.
- N2: distancia real entre los objetos de referencia reales.
- X: distancia entre los objetos de referencia representados (en píxeles)
- H1: distancia focal de la cámara indicada por el fabricante, 3.85 mm.

Y mediante las siguientes ecuaciones se puede hallar el valor del pixel:

$$\frac{h_1}{h_2} = \frac{n_1}{n_2} \quad n_1 \times \text{valorPixel} = x$$

**Ecuación 23. Valor de pixel**

- **Parámetros Extrínsecos.**

Relacionan los sistemas de referencia del mundo real y la cámara describiendo la posición y orientación de la cámara en el sistema de coordenadas del mundo real (Identificación óptica de la posición y orientación de un vehículo aéreo no tripulado).

Los parámetros Extrínsecos se van a detallar en el punto 4.

Cuando normalizamos la lista de esquinas (vértices) de los marcadores, mediante los parámetros intrínsecos de la cámara, se envía una lista de los vértices, esto se van a comparar con cada máscara de los marcadores que debe identificar, es decir que después de analizar la imagen del frame del video y al realizar sus operaciones afines se van a comparar cada vértice obtenido por los marcadores que debe encontrar. Gráficamente se puede ilustrar del siguiente modo.



Figura 58. Componentes conexas y vértices

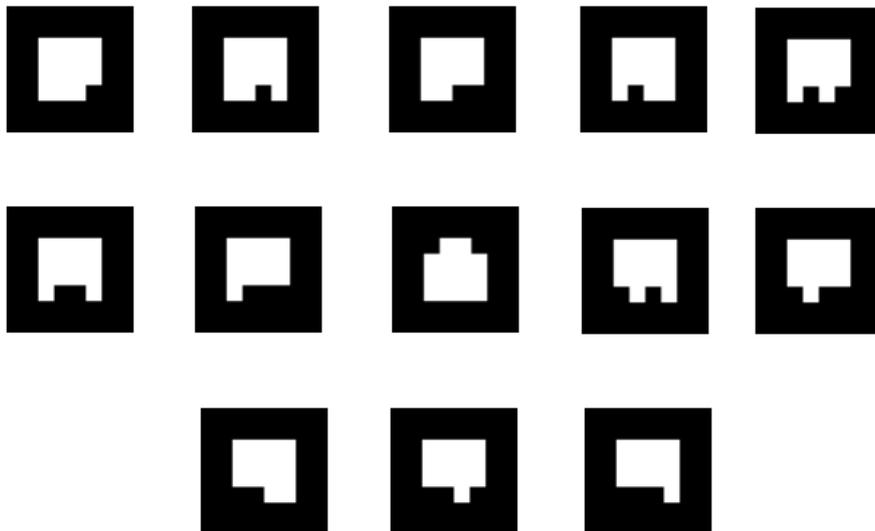


Figura 59. Conjunto de máscaras de los marcadores

La figura 59, representa el conjunto de máscaras que se van a comparar constantemente con cada uno de los marcadores, que encuentre la cámara al momento de estar escaneando su entorno (tracking)

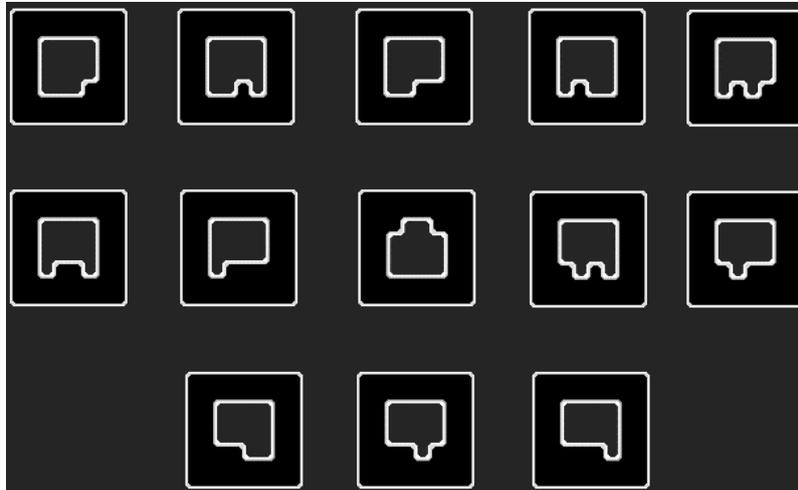


Figura 60. Obtención de vértices y componentes conexas

Visualmente el patrón que se va a comparar con los mascarar no coincide con ninguna, por lo tanto no se logra clasificar, en caso contrario, se obtendrá el tamaño del marcador y el de la mascara  $C''''_n, C'''' = (x, y)$ .

#### 4. Extracción de sistema de referencia

Al clasificar el patrón con las máscaras, es necesario obtener el sistema de referencia de la cámara y el sistema de referencia del marcador encontrado (Beatriz Martin Guadaño).

- **Parámetros Extrínsecos:**

- **Inclinación**

Se considera como inclinación al ángulo que este forma con el eje Z del mundo real.

A pesar de tener la cámara con distintos ejes locales, la inclinación con respecto a las coordenadas son las mismas, por lo que es útil explicarlo en forma gráfica.

Para el cálculo de la inclinación se plantea la siguiente situación:

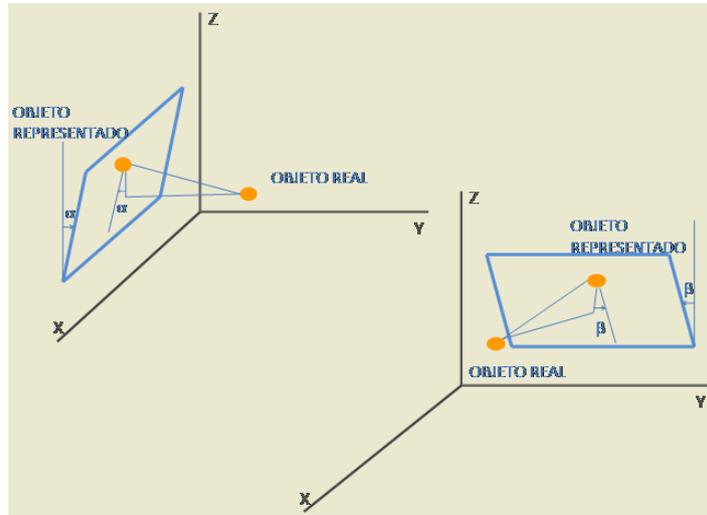


Figura 61. Extracción de sistema de referencia

Analizando de perfil, queda así:

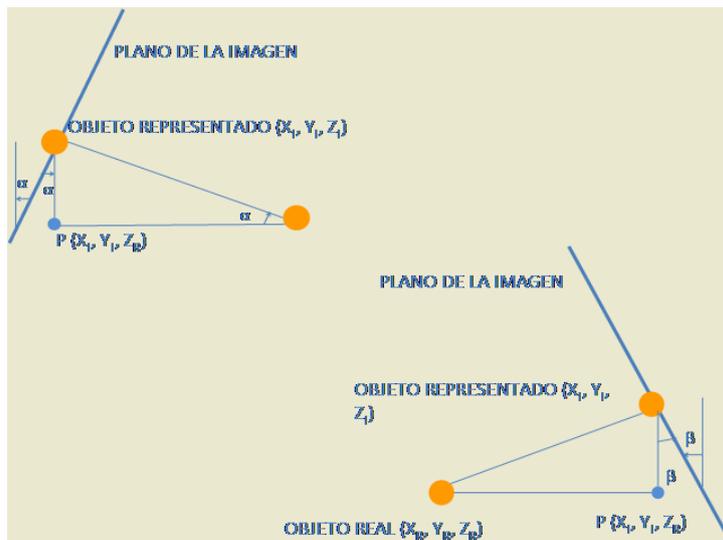
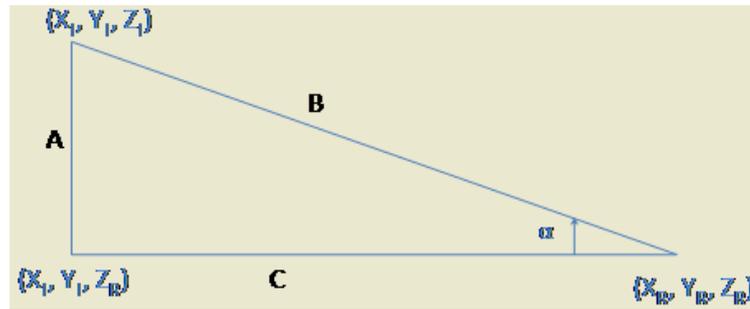


Figura 62. Cálculo de inclinación respecto a los ejes locales

Donde objeto real se conoce y objeto representado se obtiene a partir de la imagen. El punto P está en la recta paralela al plano XZ en el caso de la cámara XZ y al plano YZ en caso de que sea la cámara YZ, que pasa por el objeto representado.

De los esquemas anteriores podemos obtener, en consecuencia:



**Figura 63. Triángulo ABC: Lados A, B y C, ángulo  $\alpha$**

Con lo que se puede calcular el ángulo de inclinación de cualquier cámara  $\alpha$ , así:

$$\text{sen } \alpha = \frac{A}{B}$$

Con: A = distancia  $((X, Y, Z), (X_R, Y_R, Z_R))$

B = distancia  $((X, Y, Z), (X_R, Y_R, Z_R))$

**Ecuación 24. Ángulo de inclinación**

➤ **Distancia de los ejes**

La cámara puede estar desplazada en su plano con respecto a los ejes y por tanto, no estar situada en el origen de coordenadas del mundo real, por ello el usuario debe conocer las distancias de las cámaras a los ejes del mundo real.

La cámara XZ está situada en el plano XZ, pudiendo estar posicionada en cualquier punto de ese plano desplazada a través de los ejes X y Z del origen de coordenadas del mundo real una distancia siempre positiva.

La cámara YZ está situada en el plano YZ. Esta puede estar desplazada del origen de coordenadas a través de los ejes Y y Z una distancia positiva.

➤ **Matriz de Transformación**

Debido a la inclinación y a las distancias de ambas cámaras con respecto a los ejes del mundo real, es necesario hallar sus matrices de transformación para pasar de coordenadas de una cámara a coordenadas del mundo real.

Estas matrices de transformación tienen como parámetros el ángulo de inclinación de las cámaras con respecto al eje Z real ( $\alpha$  para la cámara XZ y  $\beta$  para la cámara YZ) y la distancia de las cámaras a los ejes del mundo

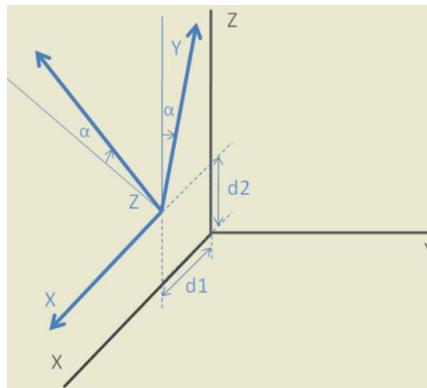
real (en el caso de la cámara XZ, d1 es la distancia en el eje X y d2, en el eje Z; en el caso de la cámara YZ, d1 es la distancia en el eje Y y d2, en el eje Z).

Matriz de transformación de la cámara XZ:

$$\begin{pmatrix} 1 & 0 & 0 & d1 \\ 0 & \text{sen } \alpha & -\text{cos } \alpha & 0 \\ 0 & \text{cos } \alpha & \text{sen } \alpha & d2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**Figura 64. Matriz de transformación de la cámara XZ**

Los ejes de la cámara XZ queda, tras la transformación de ejes así:



**Figura 65. Los ejes de la cámara XZ queda, tras la transformación de ejes**

Matriz de transformación de la cámara YZ:

$$\begin{pmatrix} 0 & \text{sen } \beta & \text{cos } \beta & 0 \\ 1 & 0 & 0 & d1 \\ 0 & \text{cos } \beta & -\text{sen } \beta & d2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

**Figura 66. Matriz de transformación de la cámara YZ**

Los ejes de la cámara YZ queda, tras la transformación de ejes así:

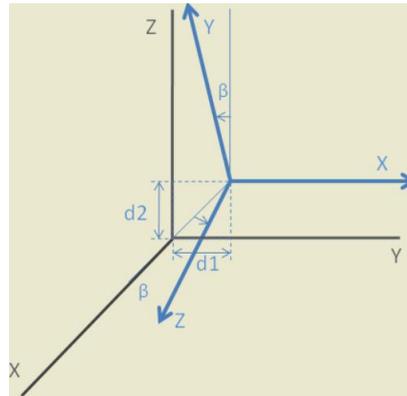


Figura 67. Los ejes de la cámara YZ queda, tras la transformación de ejes

### Método basado en Homografía.

Zhang (Charles Loop, 1999) presenta un algoritmo para estimación de pose basado en homografía o matriz de proyección. Este algoritmo requiere a lo menos 4 puntos coplanares (puntos o líneas que se encuentran en el mismo plano) correspondientes. Antes de describir el algoritmo es necesario conocer el termino *pinhole*.

### Modelo de Cámara Pinhole

Surge de la geometría proyectiva el cual fue fundado por Gérard Descargues (1591 -1661), fundamento matemáticamente los métodos de la perspectiva que habían desarrollado los artistas del renacimiento, y aunque su trabajo fue publicado en 1639, pasó desapercibido durante dos siglos.

Una cámara puede ser explicada de acuerdo a varios modelos, así como la formación de una imagen a partir de la cámara. El más utilizado es el modelo de *pinole*.

Una cámara *pinhole* es una cámara muy simple que no tiene lente y tiene una sola y muy pequeña apertura, de ahí su nombre, *pinhole*, que es la traducción al inglés de la palabra “hoyo de tamaño aguja”. Esta cámara puede ser vista de otra manera como una caja a prueba de luz con un pequeño orificio en uno de sus lados, siendo ese orificio la única entrada de luz que permite la caja. Cuando la luz de una imagen pasa a través de este agujero entonces se forma una imagen invertida en el lado opuesto de la caja (Isern, 2003).

El ojo humano trabaja de esta misma forma y como estamos más familiarizados con la explicación del funcionamiento del ojo es posible generarse una mejor idea del funcionamiento de la cámara pinhole. Mientras más pequeño es el agujero por el que se atraviesa la luz, más nítida y definida es la imagen que se forma en el

interior de la misma. Sin embargo, la imagen se hace también más oscura conforme el orificio disminuye el tamaño. Puede concluirse que mientras más definida es la imagen, ser más opaca.

Teniendo como referencia el modelo *pinhole*, podemos ya describir brevemente el algoritmo de Homografía.

El modelo de cámara pinhole describe las relaciones entre un punto 3D, coordenadas globales  $M = [X, Y, Z, 1]^T$ , y su proyección en el plano imagen  $m = [u, v, t, 1]^T$  Como:

$$sm = A[RT]M$$

**Ecuación 25. Modelo de cámara pinhole**

Donde  $s$  es el factor de escala desconocido,  $R$  la matriz de rotación,  $T$  es el vector de translación y  $A$  es la matriz de la cámara o de parámetros intrínsecos, los cuales están en función de las coordenadas del centro de la imagen  $(u_0, v_0)$ ,  $\alpha$  y  $\beta$  los cuales se derivan del largo focal y tamaño del pixel y  $\gamma$  la asimetría. Así la matriz de cámara es de la forma.

$$A = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

**Ecuación 26. Matriz de la cámara o de parámetros intrínsecos**

Cuando todos los puntos 3D son coplanares en el modelo, se selecciona el sistema de coordenadas globales tal que en el modelo el plano  $Z$  es cero. Por tanto la ecuación 26. Puede ser re-escrita como:

$$sm = A[r_1 \ r_2 \ r_3 \ T][X \ Y \ 0 \ 1]^T$$

$$sm = A[r_1 \ r_2 \ T][X \ Y \ 1]^T$$

**Ecuación 27. Modelo de cámara pinhole**

Se considera  $M' = [X \ Y \ 1]^T$  como un punto en el plano imagen, luego  $m$  y  $M'$  pueden relacionarse por una matriz de homografía  $H$ :

$$sm = HM', \text{ como } H = A[r_1 \ r_2 \ T]$$

**Ecuación 28. Modelo de cámara pinhole**

Si se definen a lo menos 4 puntos no-colineales en el modelo, la homografía H puede ser determinada hasta un factor de escala. Se puede describir

$$H = [h_1 \ h_2 \ h_3] = \lambda A [r_1 \ r_2 \ T]$$

#### **Ecuación 29. Homografía**

Teóricamente se puede descomponer la homografía H para obtener la matriz intrínseca A, en este caso aplicamos el método descrito en Zhang (Zhongfei Zhang)

Luego se puede obtener la matriz de rotación R y el vector de translación T, de la siguiente manera:

$$\begin{aligned} r_1 &= \lambda A^{-1} h_1, \quad r_2 = \lambda A^{-1} h_2 \\ r_3 &= r_1 \times r_2, \quad T = \lambda A^{-1} h_3 \end{aligned}$$

#### **Ecuación 30. Matriz de Rotación y vector de translación**

Donde  $\lambda = I / |A^{-1} h_1| = I / |A^{-1} h_2|$

Debido a que el resultado obtenido en la ecuación 31 no fuerza la ortogonalidad en R, se necesita forzar esta ortogonalidad y se hará mediante el método descrito en Weng et al (J. Weng, 1989). Por minimizar se refina la estimación final de la matriz de calibración y los parámetros R y T.

De los procesos de calibración presentados se requiere conocer puntos 3D y su proyección correspondiente en puntos de imagen 2D, estableciendo un sistema de ecuaciones que permiten determinar los parámetros intrínsecos y extrínsecos por el empleo del método de minimización por mínimos cuadrados. Para más detalles de estos procesos ver (D. Aracena, 2002), (Faugeras, 1993), (Hartley, 1995), (Stockman, 2000), (Verri., 1998), (M. Tuceryan, 1995), (Zhang, 1999)

## **5. El modelo virtual es dibujado**

Se definen diferentes sistemas de coordenadas principalmente el usado para el procesamiento de la imagen y el rendering. Es necesario comprender con precisión las relaciones entre ellas para evitar que la imagen invertida (pinhole), se muestre incorrectamente.

En la figura se definen los principales sistemas de coordenadas, utilizados.

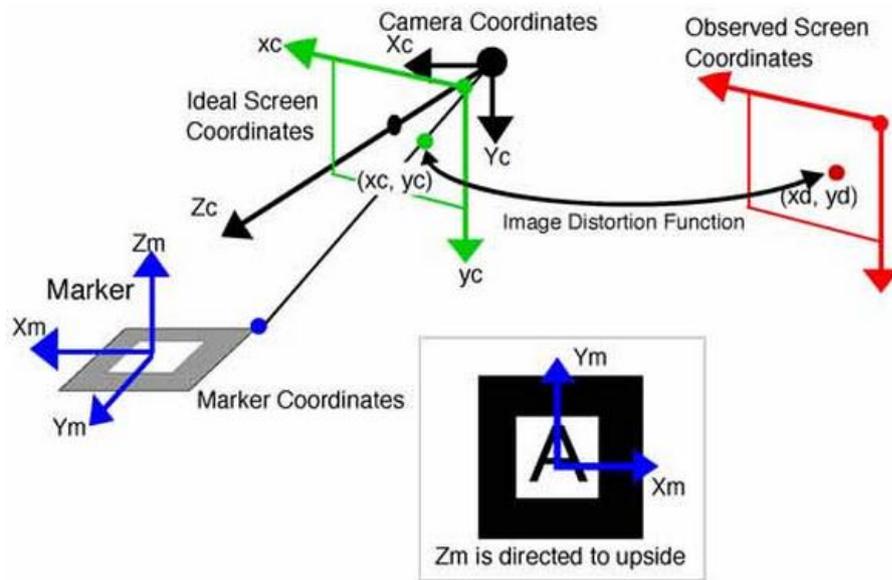


Figura 68. Sistemas de coordenadas (Marcador, cámara y modelo 2D)

Es necesario recordar que la matriz de Transformación, obtiene la posición del marcador respecto al sistema de coordenadas de la cámara (no al revés). Si se desea la posición de la cámara respecto al sistema de coordenadas del marcador se necesita invertir esta transformación (Matriz Inversa)

### Sistema de coordenadas Rendering.

Cuando se utiliza el procesamiento grafico para representar los objetos virtuales, se debe implementar el sistema de coordenadas (es decir, la cámara se encuentra en el eje Z), enfocándonos en OpenGL.

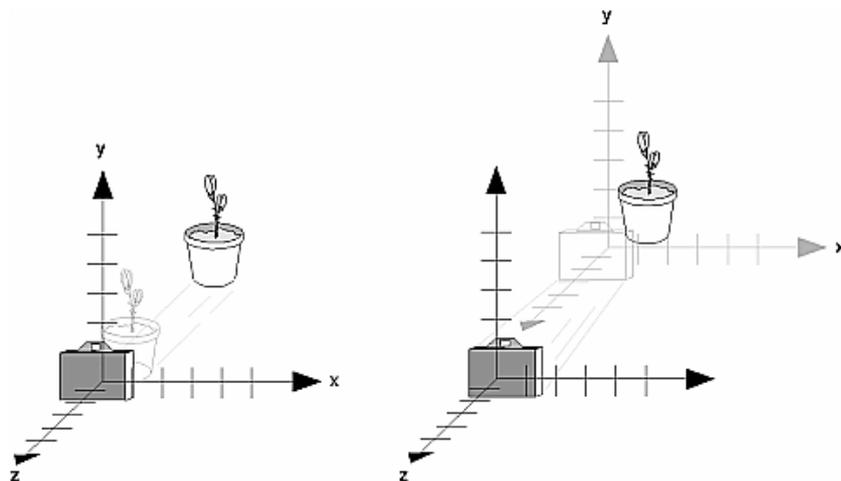


Figura 69. Sistema de coordenadas OpenGL

ArToolkit utiliza una perspectiva de la cámara calibrada que típicamente resulta en una matriz de proyección fuera del eje, cuando se usa OpenGL. Tal proyección no puede ser creada mediante una llamada *gluPerspective*, pero requiere los parámetros adicionales de *glFrustum*, en cambio se carga directamente la matriz de proyección OpenGL estableciendo *glMatrizMode GL\_PROJECTION\_MATRIX*; y llamando *glLoadMatrix*

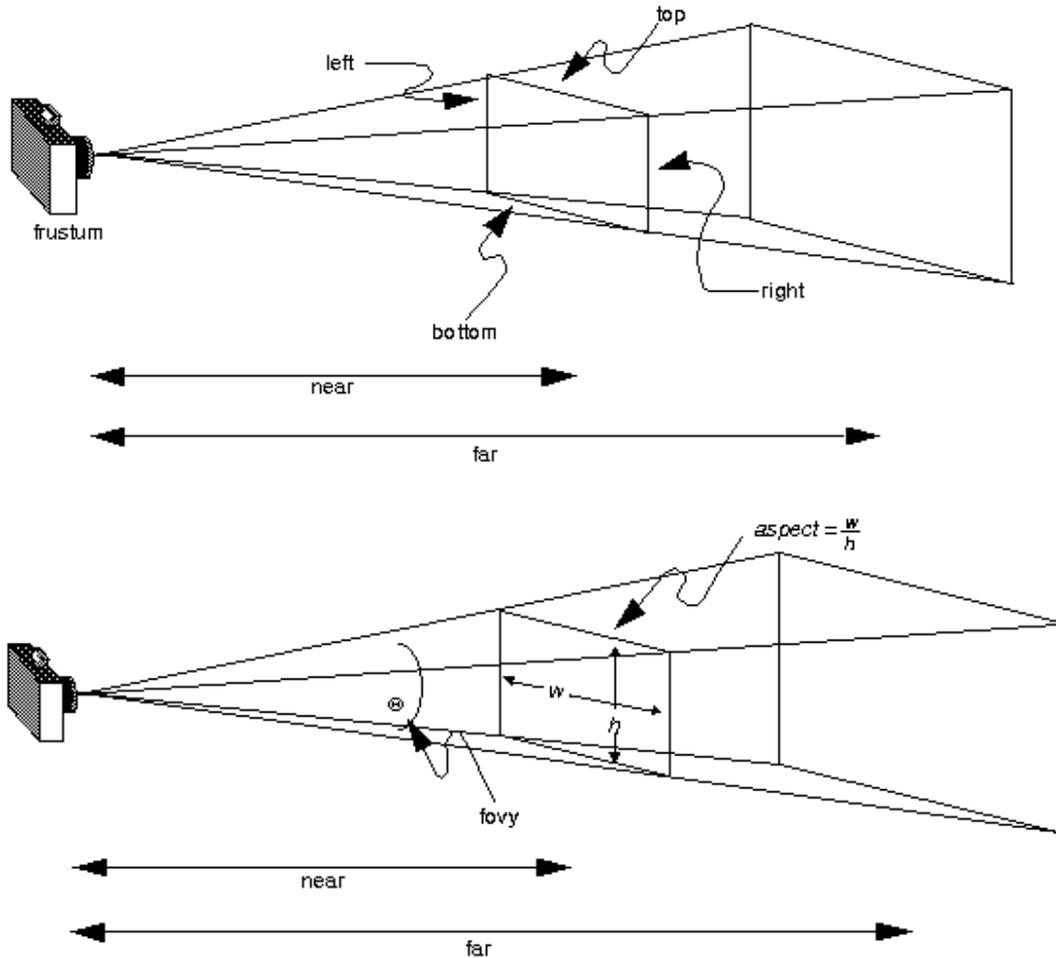


Figura.

**Figura 70. Ejes de proyección del ángulo de la cámara**

Se invita a revisar la documentación de OpenGL, para más información (Group)

### **4.2.1.3 Vuforia y Android**

El Sistema de Realidad Aumentada, al principio uso NyArToolkit, que permitía cumplir con nuestras expectativas, de aumentar el mundo real e interactuar con ella. Surgen inconvenientes, en la práctica:

1) Los marcadores:

Deben cumplir con ciertos requerimientos – Tiene que ser de forma cuadrada y de dos tonalidades. Si en el futuro se desea implementar, en la ciudad usando las señales de tránsito, no va a funcionar.

2) Ruido:

Al tener un patrón simple en el interior del marcador e implemente algoritmos que se basan en componentes conexas mediante el uso de vértices, para clasificar y representar el modelo real.

Estas características son susceptibles al ruido, ya que algún objeto del mundo real, puede cumplir con los patrones de un marcador.

3) Multiplataforma

El Sistema de AA-AR, debe estar al alcance de cualquier usuario, por lo que se desea implementar en dispositivos móviles.

Processing tiene un módulo de desarrollo para Android, el cual esta aun en sus inicios, presentando ciertos problemas. NyArToolkit dispone de la versión para la plataforma Android 2.3 (Gingerbread), todos sus métodos ya están deprecados, entran en conflicto con algunos métodos de Android o no cumplen con el flujo de procesos.

Se toma la decisión de evolucionar el Sistema AA-AR, a la plataforma de Vuforia, para cumplir las expectativas actuales de la tecnología.

## Desarrollando en Vuforia.

En Vuforia se usa la pantalla del dispositivo móvil como un intermedio donde coexiste el mundo real y el mundo virtual. Se procesa los frames de la cámara y los objetos virtuales se superponen en una capa intermedia.



**Figura 71. Capa intermedia donde se representa objetos virtuales**

Una aplicación desarrollada en Vuforia, da al usuario una experiencia más inmersa:

- Detección más rápida de los marcadores (Targets)
- Almacenamiento en la nube de hasta 1 millón de Targets simultáneos
- Se pueden definir en tiempo real los objetos del mundo real que funcionen como Targets nuevos.
- Objetos Cilíndricos – Detección y seguimiento de imágenes sobre una superficie cilíndrica.
- Reconocimiento de Texto – Reconocimiento y seguimiento del texto impreso, incluyendo secuencias alfanuméricas.
- Seguimiento robusto – El objeto virtual, no desaparece al tener el dispositivo móvil en movimiento
- Seguimiento simultáneo hasta 5 objetivos.
- Mejores resultados en condiciones – Bajo la luz, Objeto parcialmente cubierto.
- Capacidad de optimizar los gráficos para representarse más realistas.

Este diagrama describe una visión general del proceso del desarrollo utilizando Vuforia. Básicamente el núcleo de esta biblioteca, ofrece dos servicios para almacenar los Targets (marcadores). El primero de ellos es un Gestor de Targets (Target Management) y el segundo es un Gestor de la nube (Cloud Target Database)

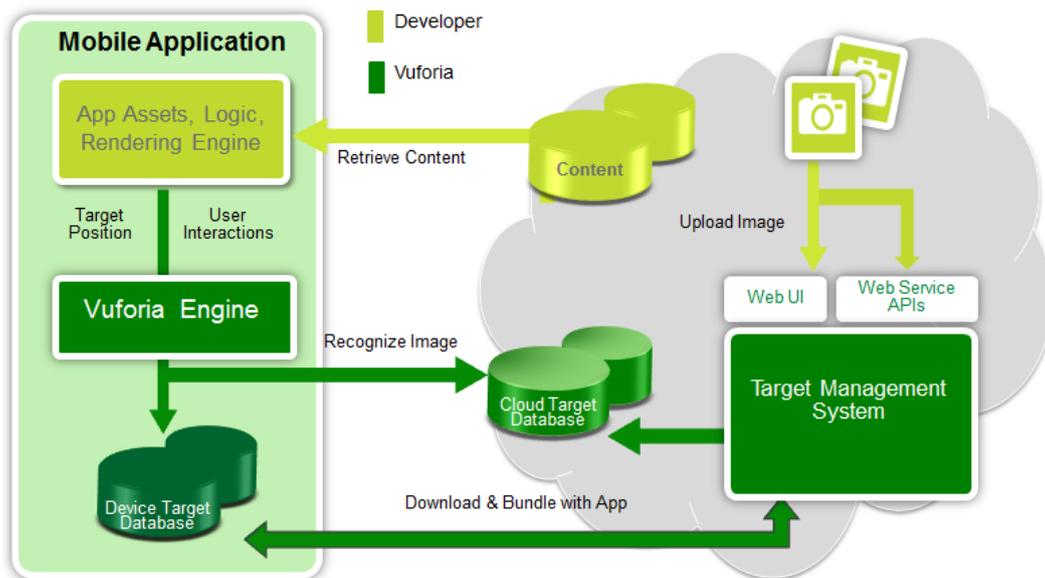


Figura 72. Componentes de Vuforia

La ventaja principal que ofrece Vuforia, es proporcionar, un servicio de base de datos, esto se traduce como un gestor de marcadores (Targets), donde se van almacenar, todos los targets que serán utilizados en diferentes aplicaciones. El usuario final podrá recibir actualizaciones automáticas de nuevos Targets que agregue el desarrollador.

Este mismo se encuentra alojado en la página oficial de Vuforia, donde se debe crear una cuenta y elegir el tipo de base de datos del Manejador de Targets (Vuforia, Qualcomm Vuforia Developer Portal).

- Base de datos del Dispositivo Móvil
- Base de datos de en la Nube.

Al elegir el tipo de base de datos ya no es permitido cambiarse, por lo que es necesario explicar brevemente las características y funcionalidades que ofrecen cada una.

### Base de Datos del Dispositivo Móvil.

Como su nombre indica todos los marcadores se van almacenar localmente en la memoria del dispositivo móvil.

Si uno conoce todos los targets que se van a identificar para representar la realidad aumentada en el dispositivo móvil y son menos de 100 marcadores, la solución correcta es elegir este tipo de base de datos. Puesto que se asegura que se van a identificar más rápido, debido al almacenamiento interno

### Base de Datos en la Nube.

La base de datos no se almacena directamente en el dispositivo móvil, esto ofrece dos ventajas significativas: la primera es que se almacena en la nube es decir solo se tendrá acceso desde el internet y la segunda es que soporta más de un millón de targets.

Este tipo de servicio, es más útil para aquellas aplicaciones que constantemente cambian de targets, es decir si son aplicaciones de catálogos de ropa, promociones, etc.

La desventaja se refleja en el tiempo de reconocimiento (en función de la conexión de internet) pues al no encontrarse en el dispositivo móvil.

Se realiza una tabla comparativa con las características de cada una.

BD Dispositivo Móvil	BD Nube
Límite de 100 Targets por cada dispositivo móvil	Más de 1 millón de Targets
Permite la descarga de Targets en diferentes combinaciones	Una BD con imágenes y metadatos
Objetos descargados solamente para el seguimiento sin soporte de metadatos	Objetos recuperados por el reconocimiento pueden llevar hasta 1MB de metadatos
No es necesaria, la conexión a internet para el reconocimiento	Es necesaria, la conexión a internet para el reconocimiento en la nube
El tiempo de ejecución de respuesta de detección hasta dentro de 2 o 3 frames	Tiempo de respuesta hasta 3 segundos, dependiendo de la conexión
Múltiples base de datos pueden ser activadas (cada una máximo 100 Targets)	Reconocimiento máximo 1 millón de Targets activos en cualquier momento.
Alojamiento Gratis	En función de su uso, puede ser: gratis o de paga

Tabla 4. Descripción de las Bases de Datos Vuforia

Para el Sistema AA-AR se toma la decisión de crear la realidad aumentada, haciendo uso de la base de datos almacenada en el dispositivo móvil, puesto que las señales de tránsito se conocen a priori y a demás no son susceptibles a cambios frecuentes, puesto que se encuentran estandarizadas a nivel internacional.

### Algoritmo.



Algoritmo 8. Fases de Desarrollo del Sistema AA-AR

## Diseñar Targets.

Los Targets son representaciones del mundo real que pueden ser detectados y seguidos, Vuforia ofrece características de acuerdo al target diseñado tales como:

- **Targets de Imagen:**  
Se enfoca a imágenes ordinarias, es decir, imágenes 2D. Por ejemplo: fotos, portadas de revistas o libros, empaques, poster, tarjetas de felicitación.
- **Targets Cilíndricos:**  
Imágenes que se encuentran impresas en objetos cilíndricos o cónicos. Por ejemplo: fotos, logos, botellas, botes, tazas.
- **Targets de Texto:**  
Elementos que representen un conjunto de letras tales como: libros, anuncios, revistas, periódicos o algún medio de difusión.
- **Targets definidos por el usuario:**  
Pueden ser de cualquier tipo de targets, pero estos se crean en tiempo de ejecución, la aplicación debe ser programada para permitirlo.
- **Targets reconocimiento de la nube:**  
Representa cualquier tipo de marcador, pero se obtiene desde la nube.
- **Multi-Targets:**  
Este tipo de marcadores, se enfocan en los empaques, cajas, objetos cuadrados o rectangulares, que se conforman por más de una imagen en sus superficies.
- **Frame markers:**  
Son las imágenes diseñadas con un patrón especial, se suelen encontrar en: los juegos de mesa, códigos QR e inclusive los marcadores que usa NyArToolkit.
- **Virtual Button:**  
Es una característica de multimedia; su objetivo es representar botones virtuales en pequeñas regiones de la imagen, que representa interacción, tal como: reproducir un sonido o alguna animación.

Para más información se puede consultar la documentación de Vuforia (Vuforia, Qualcomm Vuforia Developer Portal)

El tipo de Target que se va definir es de imagen. Generalmente las señales de tránsito se encuentran impresas en superficies planas.

## Definición

Las imágenes con las que puede interactuar Vuforia. Son muy diferentes a los tradicionales marcadores (marcadores con patrón o códigos QR). No hay necesidad de tener dos tonalidades (regiones) o códigos que deben ser reconocidos.

Se puede elegir cualquier tipo de imagen, donde se extraer todas sus características para realizar la detección y el seguimiento adecuado. Esencialmente la imagen se reconoce comprándose con la información almacenada en el dispositivo móvil. Una vez identificada se hace el seguimiento hasta que desaparezca del campo visual de la cámara, es decir, que este oculta en su totalidad; si está parcialmente visible aun realiza el seguimiento, esto no sucede con NyArToolkit.

## Clase Base Trackable (Seguimiento)

Hacemos un pequeño paréntesis para identificar ciertas clase necesarias para el desarrollo de Realidad Aumentada, en Vuforia, el concepto de un Target (marcador) es representado, a través de la definición, de dos clases principales: **Trackable** y **TrackableResult**.

- **Trackable:**

Es usada para identificar todo los objetos del mundo real, permitiendo realizar la detección y seguimiento en el espacio 3D. Se define básicamente las propiedades que se le van a dar al Target (Marcador) como: El nombre, el tipo, ID y los datos definidos por el usuario. Trackable contiene subtipos para cada tipo de Target tales como:

  - **ImageTarget** – Trackable (seguimiento), de puras imágenes.
  - **CylinderTarget** – Trackable, consiste de imágenes representadas en superficies cilíndricas o cónicas.
  - **MultiTarget** – Trackable, imágenes construidas mediante una configuración espacial.
  - **Marker** – Trackable, consiste de un conjunto de marcas o patrones
  - **World** – Trackable, representa elementos de texto,

El subtipo se define, de acuerdo al tipo de imagen. Se analizan los frames que va capturando la cámara, para encontrar el subtipo definido, cuando este coincide un objeto **TrackableResult** es creado y se pasa a la aplicación el **estado** de cada frame.

Una instancia de **TrackableResult** contiene una referencia del Target, un estatus y la matriz pose, que representa la actual posición y orientación en 3D del trackable con respecto a la referencia de la cámara (Vuforia, Qualcomm Vuforia Developer Portal).

Cuando se diseñó el algoritmo para la creación de realidad aumentada. Se hace hincapié, que en cualquier sistema de RA, siempre existe ciertas limitaciones y es necesario conocerlas de antemano para saber hasta donde son sus alcances.

## Diseño de Targets (Marcadores)

Se considera un marcador a cualquier representación gráfica 2D (imágenes), estas se pueden diseñar utilizando ciertas herramientas de diseño gráfico o tomar una ya diseñada.

Pero deben cumplir ciertas características:

- ✓ El formato debe ser JPG o PNG. (RGB o escala de gris, no válido CMYK)
- ✓ Máximo tamaño 2MB
- ✓ Rica en detalles, por ejemplo: la escena de la calle, grupo de personas, collages, mezclas de elementos
- ✓ Buen contraste, es decir, las regiones oscuras y brillantes bien iluminadas.
- ✓ No se permiten patrones repetidos, es decir, un campo de hierba, los granos de arena, la fachada de una casa con ventanas idénticas, un tablero de ajedrez (no se asegura una buena detección).
- ✓ Debe ser de 8 – 24 bits

### EJEMPLO:

La Figura A. representa una imagen, con un contenido diverso, es decir, tiene un buen contraste, formas distintas y diferentes tonalidades, estas características son ideales para ser un marcador, visualmente uno piensa, que es un patrón repetido, por contener el mismo elemento (piedras).

En la Figura B. se visualiza gráficamente los patrones que se van almacenar en la base de datos del dispositivo móvil, computacionalmente es una matriz de patrones, no una imagen.

Estos patrones se van a comparar constantemente con cada objeto del mundo real que capture la cámara (Vuforia, Qualcomm Vuforia Developer Portal).



Figura 73. Imagen original

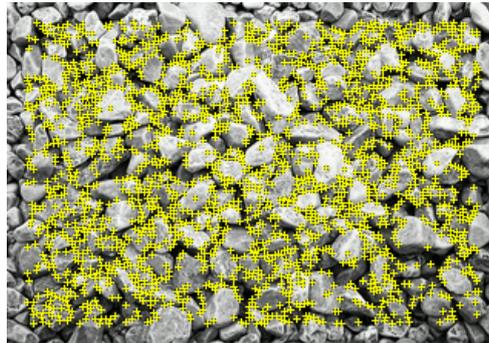


Figura 74. Patrones obtenidos a partir de la imagen original

### Impresión de Targets (marcadores)

- **Tamaño.**

Se recomienda que el marcador, tenga un tamaño mínimo 12cm, esta dimensión dependerá de la cantidad de características que contenga el marcador, por ejemplo, si el marcador contiene pocas características, su dimensión debe ser menor, para que el campo de visión de la cámara lo identifique rápidamente.

Otro factor a considerar es la dimensión de la imagen (marcador), ya que debe estar relacionada con el tamaño de impresión del marcador.

Ejemplo:

Si entrenamos el sistema con un marcador con medidas 10 x 10 = 100 pixeles, se tendría que imprimir un marcador de tamaño aproximado de 2.6cm, puesto el sistema buscara la homografía del marcador.

$$1 \text{ pixel (X)} = 2.645833333333334E-02 \text{ [cm]}$$

$$1 \text{ pixel (X)} = 0.02645833333333334 \text{ [cm]}$$

- **Superficie.**

La calidad del seguimiento puede reducirse significativamente, cuando el marcador no se encuentra en una superficie plana. Se recomienda que se asegure que el marcador no este doblado, enrollado o con deformaciones.

Como Vuforia, nos brinda mayores prestaciones en cuanto al diseño de marcadores y en los algoritmos de seguimiento, se tiene la **bondad de ajuste**<sub>1</sub> es más robusto. Esto ofrece mejor aproximación, cumpliendo con las expectativas del sistema AA-AR, identificar las señales de tránsito, tal cual se representan en el mundo real, por fines de práctica se diseñaron las señales de tránsito, en pequeños modelos de escala.



**Figura 75. Señales de tránsito a escala**

**Bondad de ajuste** es un tipo de prueba que identifica el grado concordancia entre los valores observados ( $F_o$ ) y los valores esperados ( $F_e$ ) en un análisis estadístico

## 4.2.2 Reconocimiento del camino

Como el sistema AA-AR debe interactuar con el entorno que le rodea, para identificar las señales de tránsito, es necesario saber cómo debe desplazarse en el mismo. Es decir, poder seguir la trayectoria que se le presente.

### Algoritmo.

1. Captura de la Imagen.
2. Cambio de dominio de RGB a HSV
3. Transformación de pixeles a Hue
4. Umbralizar
5. Detección de carriles
6. Establecer Radio de Curvatura, Orientación y Posición

### Desarrollo

#### 1. Captura de la Imagen.

- Crear una instancia de la cámara
- Elegir la cámara a utilizar (un dispositivo móvil, puede contar con dos cámaras: frontal y trasera. Habitualmente se usa la trasera su ID = 0)
- Se inicia el flujo de la cámara (Android envía una petición del servicio de la cámara)
- Se obtiene los frames (Un frame: es una imagen particular dentro de una sucesión de imágenes que componen una animación)

#### 2. Transformación de dominio de RGB a HSV

Una imagen RGB, está compuesta por 3 canales (Red, Green, Blue), estos pueden tener una combinación de 0-255 valores, cada uno, estos valores al ser combinados forman un pixel (unidad mínima, en la que se compone una imagen). A veces se encuentra un 4 canal que determina el brillo o intensidad del color (Jorge A. Marcial Basilio, 2012), (RGB to HSV color space conversion, s.f.).

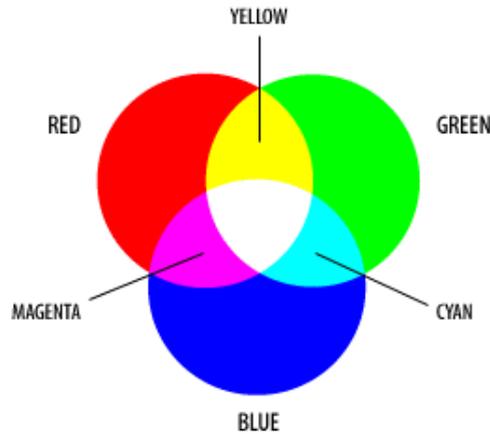


Figura 76. Modelo de color RGB

El modelo de color RGB es conocido como síntesis aditiva, no necesariamente el color obtenido está definido, puede cambiar de aspecto en diferentes dispositivos aunque se utilice el mismo modelo puede variar considerablemente. El RGB es muy utilizado en los dispositivos de hardware (monitores, televisiones, etc.).

Existe una representación más cercana a la forma en que los humanos perciben los colores y sus propiedades, pues se agrupan en tonalidades de color, lo cual es distinto al caso del RGB donde los colores no están necesariamente tan agrupados.

El modelo de color HSV es una transformación no lineal del modelo RGB en coordenadas cilíndricas de manera que cada color viene definido por las siguientes dimensiones:

- **Tonalidad o matiz (Hue):**

Se refiere al estado puro del color, sin el blanco o negro. Se representa como un grado de ángulo cuyos valores van de 0° a 360°

- **Saturación o cantidad de color (Saturation):**

Se representa como la distancia al eje de brillo negro-blanco. Los valores posibles van del 0 al 100%. A este parámetro también se le suele llamar “pureza”. Cuanto menor sea la saturación de un color, mayor tonalidad de gris habrá y más decolorado estará.

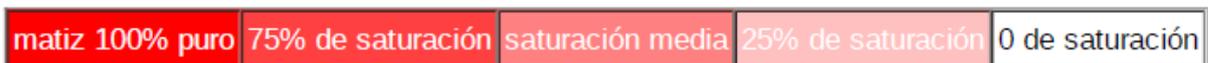


Figura 77. Niveles de saturación en el modelo HSV

- **Brillo (Value):**

Representa la altura en el eje blanco-negro. Los valores posibles van del 0 al 100%. 0 es siempre negro. Dependiendo de la saturación, 100 podría ser blanco o un color más o menos saturado

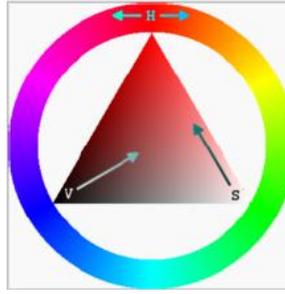


Figura 78. Brillo HSV

### Transformación RGB a HSV

Sea Max el valor máximo de los componentes (R G B) y Min el valor mínimo de esos mismos valores, los componentes del espacio HSV se pueden calcular como:

$$H = \begin{cases} \text{no definido,} & \text{si } MAX = MIN \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 0^\circ, & \text{si } MAX = R \\ & \text{y } G \geq B \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 360^\circ, & \text{si } MAX = R \\ & \text{y } G < B \\ 60^\circ \times \frac{B-R}{MAX-MIN} + 120^\circ, & \text{si } MAX = G \\ 60^\circ \times \frac{R-G}{MAX-MIN} + 240^\circ, & \text{si } MAX = B \end{cases}$$

$$S = \begin{cases} 0, & \text{si } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{en otro caso} \end{cases}$$

$$V = MAX$$

Ecuación 31. Transformación RGB a HSV

## Transformación de píxeles a Hue

La transformación del modelo RGB a HSV, permite manejar la tonalidad de colores puros.



Figura 79. Tonalidad HSV

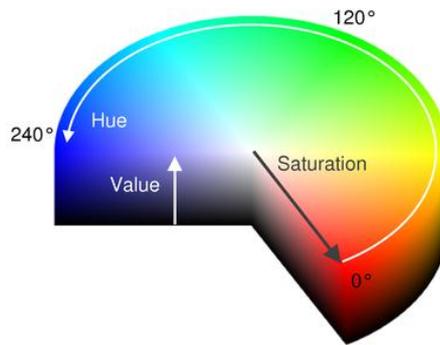


Figura 80. Modelo HSV

Facilitando la detección del color que se desea, por lo tanto se asigna a los 3 canales el Hue (tono), no interesa representar el brillo o saturación del color, estos dos últimos valores solo se usan para discriminar los colores que no cumplan con los parámetros que se establezcan.

### 3. Umbralizar

Es uno de los métodos utilizados de la segmentación (proceso de dividir una imagen en diferentes regiones u objetos). El objetivo es extraer de la imagen el objeto de interés.

## Algoritmo

Se selecciona un umbral  $T$ , que discrimine el fondo con el objeto de interés. Si el valor del pixel es mayor al umbral se establece un valor o si el valor del pixel es menor que el umbral se establece otro valor.

En nuestro caso el valor del umbral  $T$ , se determina de acuerdo al color de las líneas del camino, se elige usar líneas de color amarillo, si el valor del pixel Hue excede al umbral establecido, el valor del pixel se le asigna 0 (ausencia de color) y si el valor del pixel se encuentra dentro del rango del umbral se le asigna al pixel (color negro)

### DetECCIÓN DE CARRILES

La imagen ya se encuentra segmentada pero no se garantiza que lo encontrado, sea un camino. Para determinar que es un camino se propone un ancho mínimo que debe cumplir para que se considere (José Luis Alba y Fernando Martín, 2006), (Algoritmos de segmentación por color), (Woods), (Benito, 2004).

Se plantea crear 3 ventanas temporales para conocer la trayectoria del camino:

- Futuro
- Presente
- Pasado

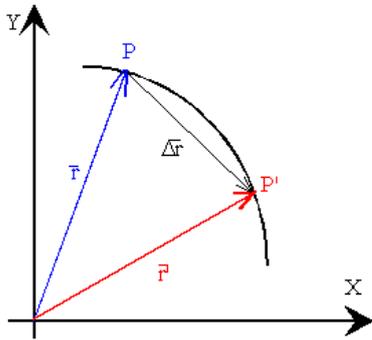
Al ser 3 ventanas, reduce computacionalmente el procesamiento que se debe realizar en la imagen.

## 4. Establecer Radio de Curvatura, Orientación y Posición

### 4.2.2.1 Movimiento curvilíneo

Supongamos que el movimiento tiene lugar en el plano  $XY$ , Situamos un origen, y unos ejes, y representamos la trayectoria del móvil, es decir, el conjunto de puntos por los que pasa el móvil (P., 1972). Las magnitudes que describen un movimiento curvilíneo son:

### Vector posición $\mathbf{r}$ en un instante $t$ .

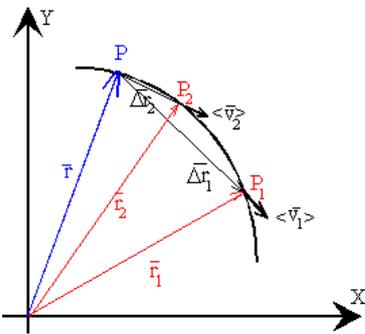


Como la posición del móvil cambia con el tiempo. En el instante  $t$ , el móvil se encuentra en el punto  $P$ , o en otras palabras, su vector posición es  $\mathbf{r}$  y en el instante  $t'$  se encuentra en el punto  $P'$ , su posición viene dada por el vector  $\mathbf{r}'$ .

Diremos que el móvil se ha desplazado  $\Delta\mathbf{r}=\mathbf{r}'-\mathbf{r}$  en el intervalo de tiempo  $\Delta t=t'-t$ . Dicho vector tiene la dirección de la secante que une los puntos  $P$  y  $P'$ .

Figura 81. Vector posición  $\mathbf{r}$  en un instante  $t$

### Vector velocidad

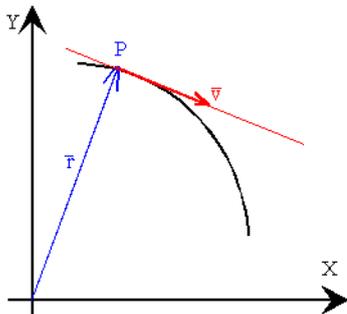


El vector velocidad media, se define como el cociente entre el vector desplazamiento  $\Delta\mathbf{r}$  y el tiempo que ha empleado en desplazarse  $\Delta t$ .

$$\langle \mathbf{v} \rangle = \frac{\mathbf{r}' - \mathbf{r}}{t' - t} = \frac{\Delta\mathbf{r}}{\Delta t}$$

El vector velocidad media tiene la misma dirección que el vector desplazamiento, la secante que une los puntos  $P$  y  $P_1$  cuando se calcula la velocidad media  $\langle \mathbf{v}_1 \rangle$  entre los instantes  $t$  y  $t_1$ .

El vector velocidad en un instante, es el límite del vector velocidad media cuando el intervalo de tiempo tiende a cero.



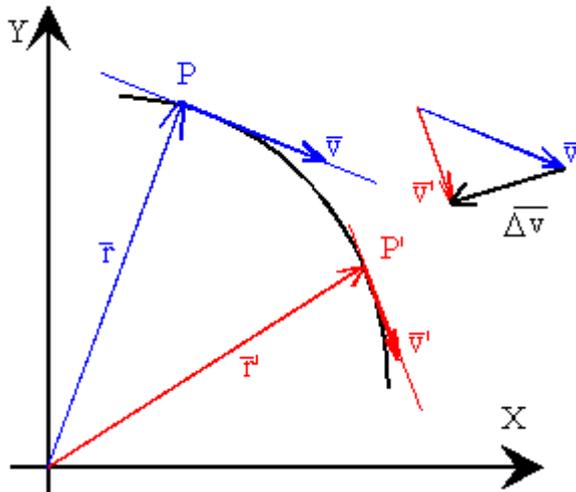
$$\mathbf{v} = \lim_{\Delta t \rightarrow 0} \frac{\Delta\mathbf{r}}{\Delta t} = \frac{d\mathbf{r}}{dt}$$

Como podemos ver en la figura, a medida que hacemos tender el intervalo de tiempo a cero, la dirección del vector velocidad media, la recta secante que une sucesivamente los puntos  $P$ , con los puntos  $P_1, P_2, \dots$ , tiende hacia la tangente a la trayectoria en el punto  $P$ .

En el instante  $t$ , el móvil se encuentra en  $P$  y tiene una velocidad  $\mathbf{v}$  cuya dirección es tangente a la trayectoria en dicho punto.

Figura 82. Vector velocidad

## Vector aceleración



En el instante  $t$  el móvil se encuentra en P y tiene una velocidad  $\mathbf{v}$  cuya dirección es tangente a la trayectoria en dicho punto.

En el instante  $t'$  el móvil se encuentra en el punto P' y tiene una velocidad  $\mathbf{v}'$ .

El móvil ha cambiado, en general, su velocidad tanto en módulo como en dirección, en la cantidad dada por el vector diferencia  $\Delta\mathbf{v}=\mathbf{v}'-\mathbf{v}$ .

Figura 83. Vector aceleración

Se define la aceleración media como el cociente entre el vector cambio de velocidad  $\Delta\mathbf{v}$  y el intervalo de tiempo  $\Delta t=t'-t$ , en el que tiene lugar dicho cambio.

$$\langle \mathbf{a} \rangle = \frac{\mathbf{v}' - \mathbf{v}}{t' - t} = \frac{\Delta\mathbf{v}}{\Delta t}$$

Ecuación 32. Aceleración Media

Y la aceleración  $\mathbf{a}$  en un instante

$$\mathbf{a} = \lim_{\Delta t \rightarrow 0} \frac{\Delta\mathbf{v}}{\Delta t} = \frac{d\mathbf{v}}{dt}$$

Ecuación 33. Aceleración  $\mathbf{a}$  en un instante

Resumiendo, las ecuaciones del movimiento curvilíneo en el plano XY son

$$\begin{array}{lll} x = x(t) & v_x = \frac{dx}{dt} & a_x = \frac{dv_x}{dt} \\ y = y(t) & v_y = \frac{dy}{dt} & a_y = \frac{dv_y}{dt} \end{array}$$

Ecuación 34. Movimiento curvilíneo en el plano XY

La primera fila corresponde, a las ecuaciones de un movimiento rectilíneo a lo largo del eje X, la segunda fila corresponde, a las ecuaciones de un movimiento rectilíneo a lo largo del eje Y, y lo mismo podemos decir respecto del eje Z.

**Por tanto, podemos considerar un movimiento curvilíneo como la composición de movimientos rectilíneos a lo largo de los ejes coordenados.**

**Ejemplo 1:**

Un automóvil describe una curva plana tal que sus coordenadas rectangulares, en función del tiempo están dadas por las expresiones:  $x=2t^3-3t^2$ ,  $y=t^2-2t+1$  m. Calcular:

- Las componentes de la velocidad en cualquier instante.

$$v_x=6t^2-6t \text{ m/s}$$

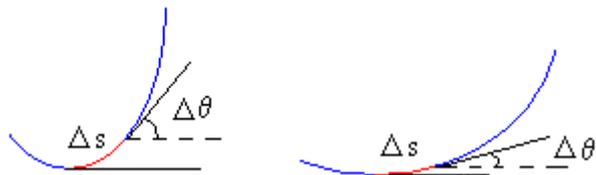
$$v_y=2t-2 \text{ m/s}$$

- Las componentes de la aceleración en cualquier instante.

$$a_x=12t \text{ m/s}^2$$

$$a_y=2 \text{ m/s}^2$$

**4.2.2.2 Radio de curvatura**

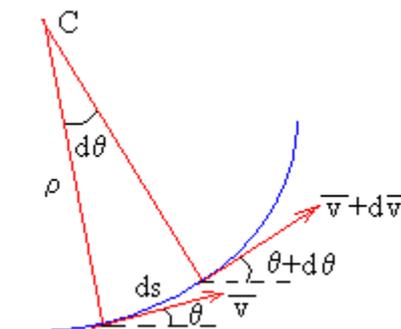


En el lenguaje ordinario, decimos que un pedazo de carretera  $\Delta s$  tiene más curvatura que otro cuando el cambio de dirección  $\Delta\theta$  es mayor al camino recorrido en ambos. Compárese la figura de la izquierda con la de la derecha

El radio  $\rho$  de curvatura medio e instantáneo se definen, respectivamente

$$\langle \rho \rangle = \frac{\Delta s}{\Delta \theta} \quad \rho = \lim_{\Delta s} \frac{\Delta s}{\Delta \theta} = \frac{ds}{d\theta}$$

**Ecuación 35. Ecuación del radio de curvatura  $\rho$**



**Figura 84. Radio de curvatura**

El radio de curvatura  $\rho$  y el centro C de curvatura se determinan del siguiente modo: Se traza la tangente a un punto de la trayectoria y a continuación, se traza la normal. Se toma un punto muy próximo al anterior, se traza la tangente y la normal en dicho punto.

Las normales se cortan en un punto denominado centro de curvatura C, y la distancia de C a uno u otro punto de la trayectoria, infinitamente próximos entre sí, se denomina radio de curvatura  $\rho$ .

Si el ángulo comprendido entre las dos tangentes es  $d\theta$ , este es el ángulo que forman las dos normales. La longitud del arco entre los dos puntos considerados es  $ds = \rho \cdot d\theta$ .

Dada la función  $y=f(x)$ , vamos a determinar la fórmula que nos permite calcular el radio de curvatura  $\rho$  de la curva en la posición de abscisa  $x$ .

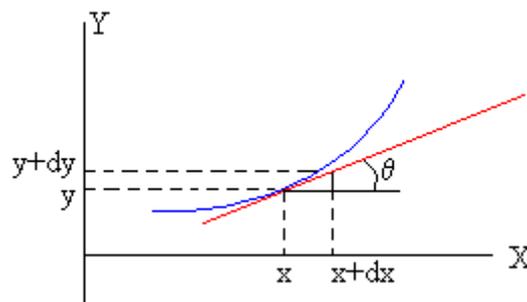


Figura 85. Radio de curvatura  $\rho$  de la curva en la posición de abscisa  $x$

Como vemos en la figura, en el triángulo rectángulo de base  $dx$ , altura  $dy$  e hipotenusa  $ds$ , establecemos las siguientes relaciones

$$\tan \theta = \frac{dy}{dx} \quad ds = \sqrt{dx^2 + dy^2} = \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx$$

La fórmula del radio de curvatura es:

El radio de curvatura es una cantidad positiva

$$\rho = \frac{ds}{d\theta} = \frac{\sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx}{d \arctan \frac{dy}{dx}} = \frac{\sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx}{\frac{\frac{dy^2}{dx^2} dx}{1 + \left(\frac{dy}{dx}\right)^2}} = \frac{\left(1 + \left(\frac{dy}{dx}\right)^2\right)^{3/2}}{\frac{dy^2}{dx^2}}$$

Ecuación 36. Radio de curvatura

### 4.2.2.3 Orientación

La orientación se define como el grado de inclinación que se encuentra respecto al carril. Una orientación de un objeto en el espacio es cada una de las posibles elecciones para colocarlo sin cambiar un punto fijo de referencia. Puesto que el objeto con un punto fijo puede todavía ser rotado alrededor de ese punto fijo, la posición del punto de referencia no especifica por completo la posición, por tanto para especificar completamente la posición necesitamos especificar también la orientación. La orientación puede visualizarse añadiendo una base vectorial ortogonal al punto de referencia del objeto; diferentes bases representarían diferentes orientaciones.

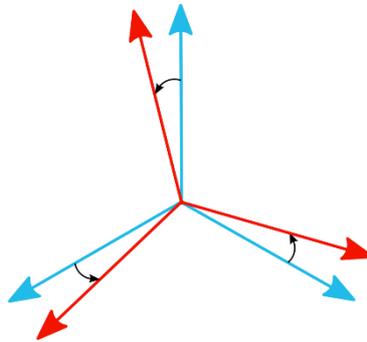
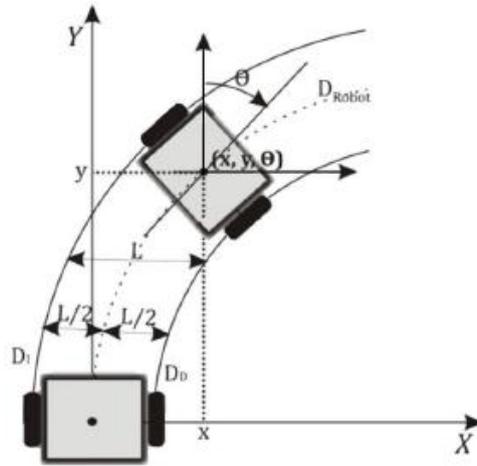


Figura 86. Orientación

### 4.2.2.4 Posición

Se define como el punto del espacio donde se encuentra el robot móvil respecto al carril. Para este tipo de robots existen ecuaciones que se pueden utilizar para estimar la posición relativa del robot. La odometría es uno de los métodos más usados para estimar la posición relativa de un robot móvil respecto a un punto de partida conocido. Se basa en ecuaciones que puede implementarse fácilmente y utilizan la información generada por encoders colocados en los motores o en la descripción del camino. Depende de la configuración del robot estas expresiones pueden variar considerablemente, las ecuaciones que rigen a un robot móvil son las siguientes:



**Figura 87. Posición**

$$D_D = D_I = \pi \times dr \times \text{Número de Revoluciones}$$

$$D_{Robot} = \frac{D_D + D_I}{2}$$

**Ecuación 37. Ecuaciones para Encoder**

Donde  $D_{Robot}$  es el desplazamiento del robot,  $D_D$  es el desplazamiento del motor derecho,  $D_I$  es el desplazamiento del motor izquierdo, DR es el diámetro de las llantas. Tomando como referencia el desplazamiento del robot, el desplazamiento de cada llanta puede expresarse de la siguiente forma:

$$D_I = \theta \times \left( r + \frac{L}{2} \right)$$

$$D_D = \theta \times \left( r - \frac{L}{2} \right)$$

Donde L es la separación entre el punto medio de las llantas. Restado ambas ecuaciones se elimina la variable r que es el radio de cada llanta.

$$D_I - D_D = \theta \times L$$

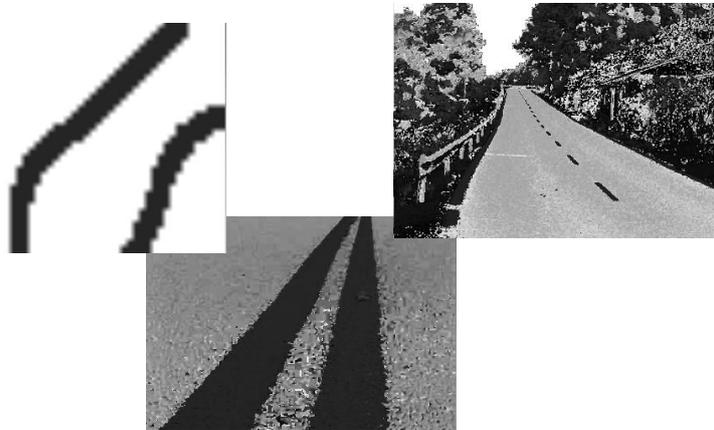
Despejando  $\theta$  podemos conocer la orientación del robot en radianes.

$$\theta = \left( \frac{D_I - D_D}{L} \right)$$

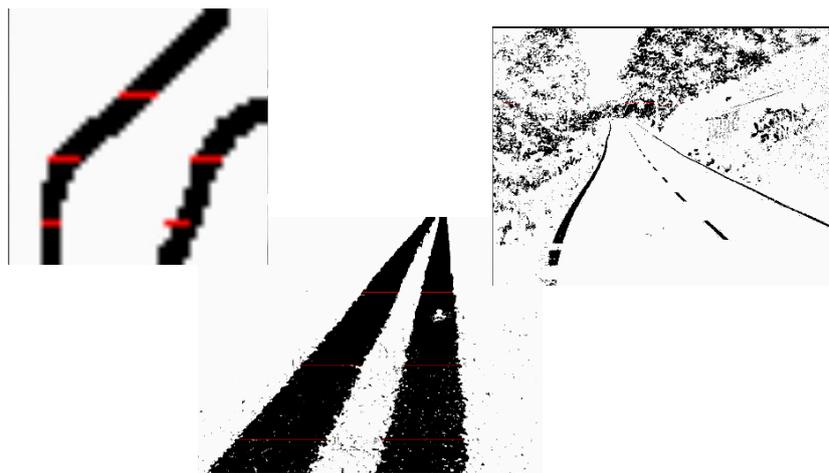
**Ecuación 38. Orientación del Robot móvil**



Imagen original



Transformación de RGB a HSV



Detección de carriles

Figura 88. Resultado de la detección del camino

### 4.2.3 Control Móvil

El proceso de controlar un vehículo autónomo implica aplicar procesos demasiado complejos, cuando no existe un modelo de solución simple o un modelo matemático preciso. Básicamente se necesita usar el conocimiento de un experto, cuando ciertas situaciones son desconocidas y no pueden medirse de forma confiable o cuando el ajuste de una variable puede producir el desajuste de otras. Con lo que se plantea usar lógica difusa.

Lógica difusa: es una técnica de la inteligencia computacional que permite trabajar información con alto grado de imprecisión; esto es opuesto a la lógica convencional que trabaja con información definida y precisa.

Puede ser vista como una metodología cercana al pensamiento humano, que proporciona una manera simple y elegante de obtener un conjunto de soluciones a partir de un conjunto de entradas.

La lógica difusa procura crear aproximaciones matemáticas en la resolución de ciertos tipos de problemas. Pretende producir resultados exactos a partir de datos imprecisos, por lo cual es particularmente útil en aplicaciones de control (hardware)

### Orígenes

La lógica difusa fue investigada, por primera vez, a mediados de los años sesenta en la Universidad de Berkeley (California) por el ingeniero Lotfy A. Zadeh (Department) (Zadeh) cuando se dio cuenta de lo que el llamo principio de incompatibilidad: “Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes”.

Se introdujo el concepto de conjunto difuso (Fuzzy set) bajo el que reside la idea de que los elementos sobre los que se construye el pensamiento humano no son números sino etiquetas lingüísticas. La lógica difusa permite representar el conocimiento común, que es mayoritariamente del tipo lingüístico cualitativo y no necesariamente cuantitativo, en un lenguaje matemático a través de la teoría de conjuntos difusos y funciones características asociadas a ellos. Permite trabajar a la vez con datos numéricos y términos lingüísticos; los términos lingüísticos son inherentemente menos precisos que los datos numéricos pero en muchas ocasiones aportan una información más útil para el razonamiento humano.

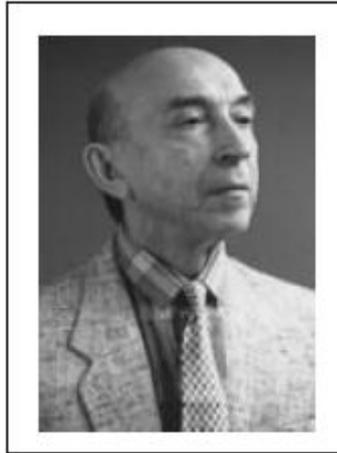


Figura 89. L.A. Zadeh

### 4.2.3.1 Robot Móvil

La configuración que se utiliza en el robot móvil es de direccionamiento Ackerman (Aschman) (Ackermann, s.f.). Se aproxima al sistema de un automóvil, consiste en dos llantas combinadas en la parte trasera que permiten impulsar y dos llantas también combinadas en la parte delantera cuya función es dar orientación al móvil.

El desplazamiento en línea recta no presenta ningún problema dado que las llantas de atrás son impulsadas por motores o uno solo, sin embargo estas presentan deslizamiento en curvas, cuando lleva una cierta velocidad y la mayor limitante que tiene un robot de este tipo es que no puede girar sobre su centro de gravedad.

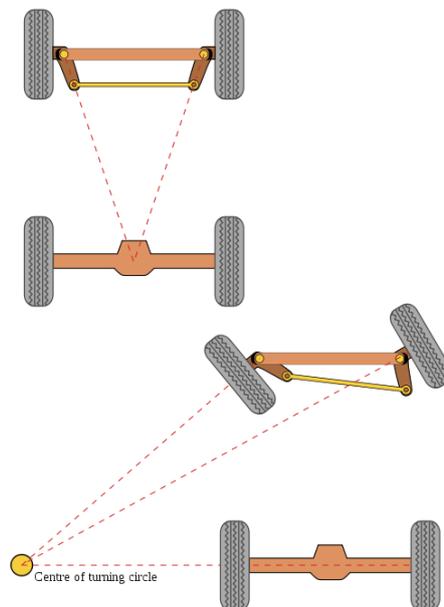


Figura 90. Direccionamiento Ackerman

La lógica difusa permitirá que el sistema de control del robot móvil tome una serie de decisiones para poder llegar a un objetivo o esquivar obstáculos que se le presenten.

#### 4.2.3.2 Algoritmo de Lógica Difusa.

Se debe tener controladas ciertas características de conocimiento que permitan manipular al robot móvil de forma efectiva (Navarrete Manzanilla Niels Henrik, 2011).

Las características de un control difuso son:

- Base de conocimiento (rule base)
- Mecanismo de inferencia (inference mechanism)
- Interfaz de difusificación. (fuzzyfication)
- Interfaz de desdifusificación (defuzzyfication)

En forma general, el algoritmo de la lógica difusa se puede representar en la siguiente figura.

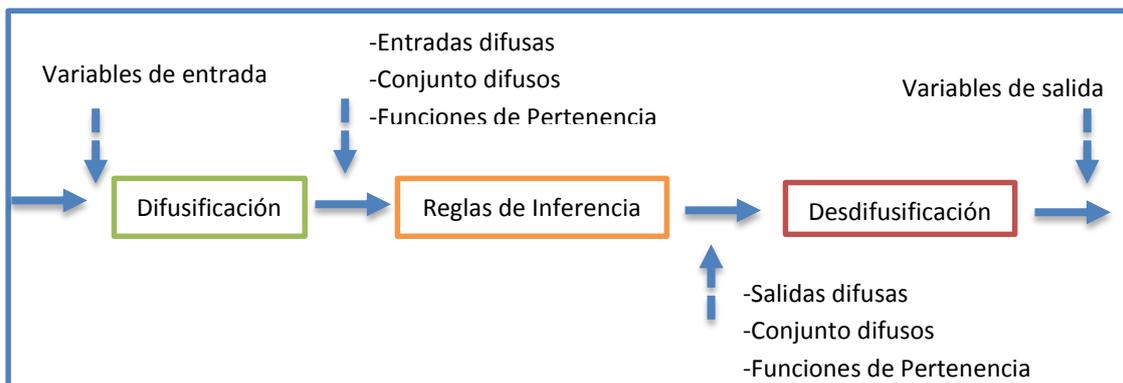


Figura 91. Estructura general

### Control Difuso.

Un sistema de control difuso trabaja de manera muy diferente a los sistemas de control convencionales. Estos usan el conocimiento experto para generar una base de conocimientos que dotara al sistema la capacidad de tomar decisiones sobre ciertas acciones que se presentan en su funcionamiento. Permite describir un conjunto de reglas que utilizaría una persona para controlar un proceso y a partir de estas reglas generar acciones de control.

### Fusificación

Tiene como objetivo convertir valores crisp o valores reales en valores difusos. En donde se asignan grados de pertenencia a cada una de las variables de entrada a los conjuntos difusos previamente definidos utilizando las funciones de pertenencia asociadas a los conjuntos difusos.

## **Base de Conocimiento.**

Contiene el conocimiento asociado con el dominio de la aplicación y los objetivos del control. En esta etapa se deben definir las reglas lingüísticas de control que realizarán la toma de decisiones de la forma en que debe actuar el sistema.

## **Inferencia.**

Relaciona los conjuntos difusos de entrada y salida para representar las reglas que definirán el sistema. En la inferencia se utiliza la información de la base de conocimiento para generar reglas mediante el uso de condiciones, por ejemplo: *si caso 1 y caso2 entonces accion1*

## **Defusificación.**

Realiza el proceso de adecuar los valores difusos generados en la inferencia en valores crisp, que posteriormente se utilizarán en el proceso de control. En la Defusificación se utilizan métodos matemáticos simples como el método del centroide, método del promedio ponderado y método de membresía del medio del máximo.

## **Definición de variables de entrada y salida**

La definición de variables de entrada son aquellas, que van a brindar información continuamente para la toma de decisiones.

Entrada: Sensores ultrasónicos

La salida se refiere al comportamiento que va tomar el control difuso al recibir el conjunto de entradas.

Salida:

- Angulo de dirección
- Velocidad de motores

## **Definición de conjuntos y valores lingüísticos asociados a las variables de entrada y salida.**

Un conjunto difuso se encuentra asociado por un valor lingüístico que está definido por una palabra, etiqueta lingüística o adjetivo. En los conjuntos difusos la función de pertenencia puede tomar valores de intervalo entre 0 y 1, y la transición es gradual y no cambia de manera instantánea como pasa con los conjuntos clásicos (Kosko, 1992), (Demouth H.)

Se definen los conjuntos y valores lingüísticos asociados a cada variable:

Para la variable “Sensores” se definen los siguientes valores lingüísticos:

- Muy Cercano
- Cercano

### Definición de función de pertenencia

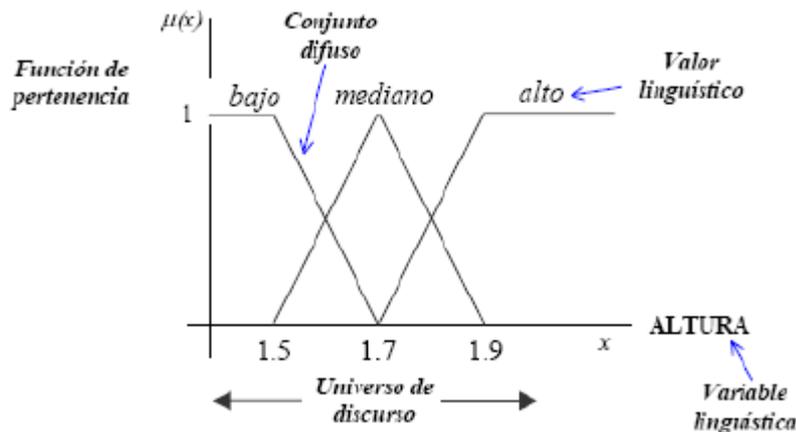
Un conjunto difuso en un universo en discurso puede definirse como lo muestra la ecuación

$$A = \{(x, \mu_A(x)) | x \in U\}$$

**Ecuación 39. Conjunto difuso en un universo en discurso**

Donde  $\mu_A(x)$  es la función de pertenencia de la variable  $x$ , y  $U$  es el universo en discurso. Cuando más cerca este la pertenencia del conjunto  $A$  al valor de 1, mayor será la pertenencia de la variable  $x$  al conjunto  $A$ .

Ejemplo:



**Figura 92. Ejemplo de representación de un conjunto difuso**

Ahora para cada valor lingüístico de la variable “Sensores” se debe definir una función de pertenencia que indique el grado en que una variable “ $x$ ” está incluida en los conceptos representados por las variables lingüísticas.

Cualquier función puede ser válida para definir un conjunto difuso, existen ciertas funciones que son más utilizadas por su simplicidad matemática. La que se implementó es la función tipo triangular también conocida como campana gaussiana, otra función es la trapezoidal.

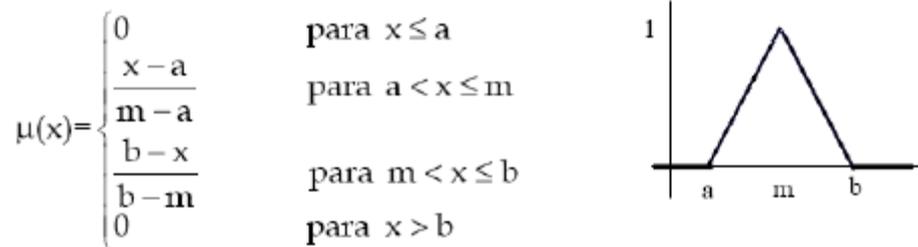


Figura 93. Función de transferencia para un conjunto difuso triangular

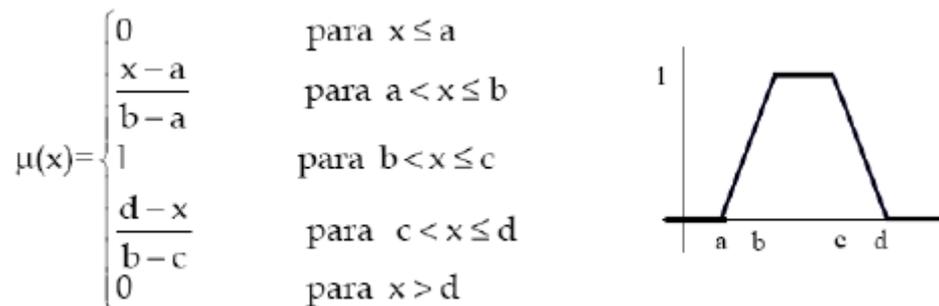


Figura 94. Función de transferencia para un conjunto difuso trapezoidal

Toda la teoría debe ser migrada a datos reales, estos datos son precisamente los que se usaran en el proyecto.

	Tipo	Rango
Entradas	Sensores	$15\text{cm} \leq x \leq 40\text{cm}$
Salida	Angulo de Dirección	$80^\circ \leq 90^\circ \leq 100^\circ$
	Velocidad de Motores	$125\text{bits} \leq 150 \leq 255\text{bits}$

Tabla 5. Entradas y salidas del sistema difuso propuestas

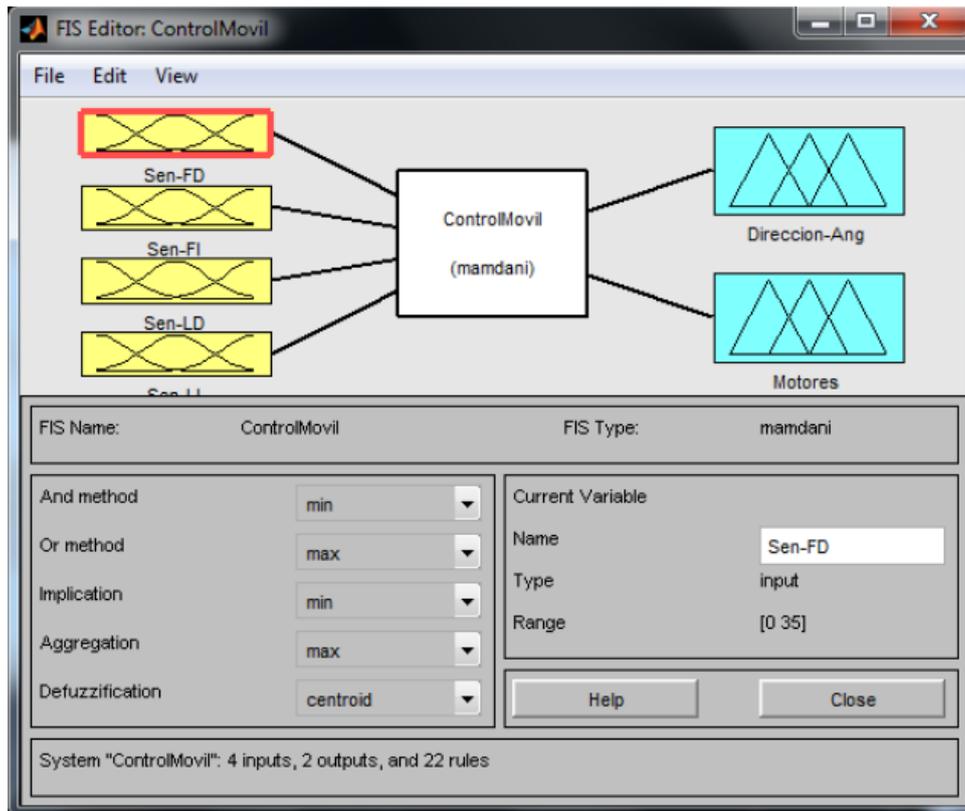


Figura 95. Control del móvil

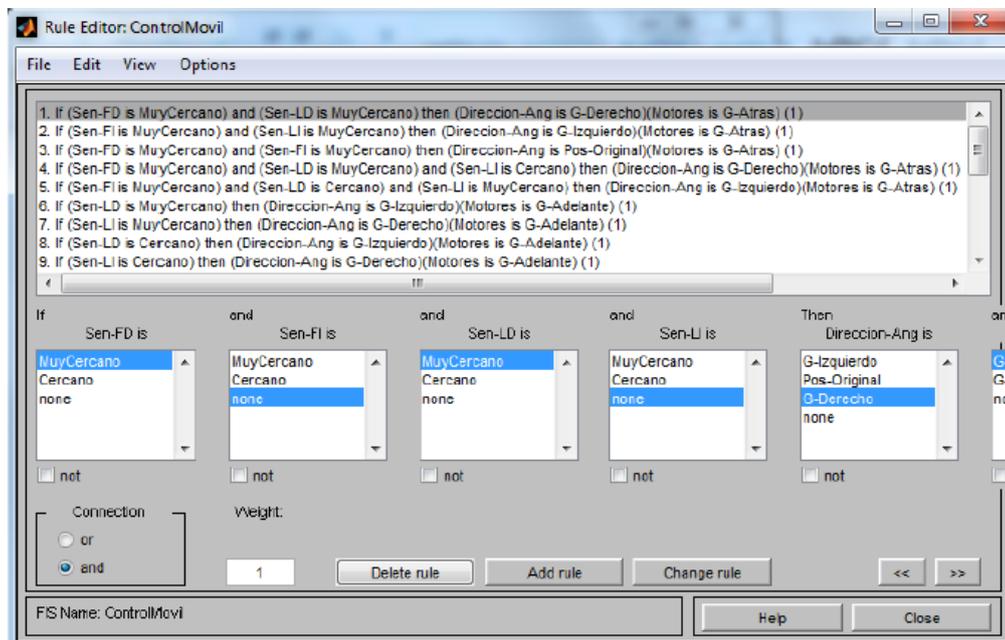
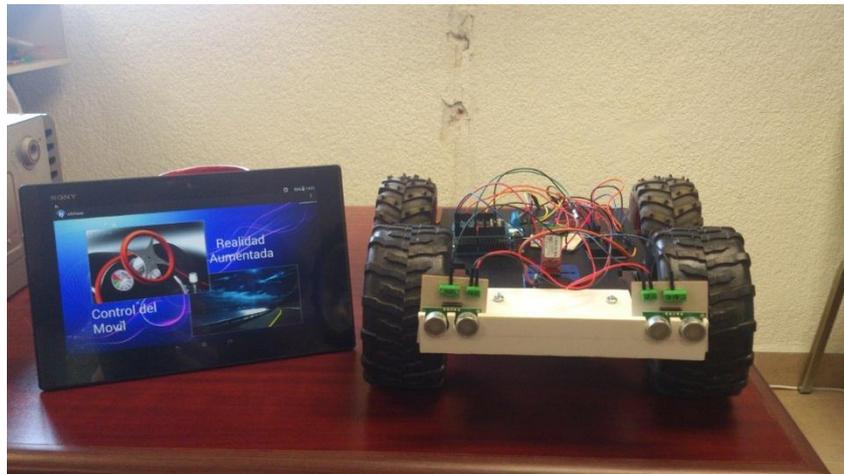


Figura 96. Reglas de Inferencia del Control Móvil

```
9. If (Sen-LI is Cercano) then (Direccion-Ang is G-Derecho)(Motores is G-Adelante) (1)
10. If (Sen-LD is MuyCercano) and (Sen-LI is Cercano) then (Direccion-Ang is G-Izquierdo)(Motores is G-Adelante) (1)
11. If (Sen-LD is Cercano) and (Sen-LI is MuyCercano) then (Direccion-Ang is G-Derecho)(Motores is G-Adelante) (1)
12. If (Sen-FD is Cercano) then (Direccion-Ang is G-Izquierdo)(Motores is G-Adelante) (1)
13. If (Sen-FI is Cercano) then (Direccion-Ang is G-Derecho)(Motores is G-Adelante) (1)
14. If (Sen-FI is Cercano) and (Sen-FI is Cercano) and (Sen-LD is MuyCercano) and (Sen-LI is Cercano) then (Direccion-Ang is G-Derecho)(Motores is G-Adelante) (1)
15. If (Sen-FD is Cercano) and (Sen-FI is Cercano) and (Sen-LD is Cercano) and (Sen-LI is MuyCercano) then (Direccion-Ang is G-Izquierdo)(Motores is G-Adelante) (1)
16. If (Sen-FD is MuyCercano) and (Sen-FI is MuyCercano) and (Sen-LD is MuyCercano) and (Sen-LI is MuyCercano) then (Direccion-Ang is Pos-Original)(Motores is G-Atras) (1)
17. If (Sen-LD is MuyCercano) then (Direccion-Ang is G-Izquierdo)(Motores is G-Adelante) (1)
18. If (Sen-LI is Cercano) then (Direccion-Ang is G-Izquierdo)(Motores is G-Adelante) (1)
19. If (Sen-LI is MuyCercano) then (Direccion-Ang is G-Derecho)(Motores is G-Adelante) (1)
20. If (Sen-LI is Cercano) then (Direccion-Ang is G-Derecho)(Motores is G-Adelante) (1)
21. If (Sen-FD is MuyCercano) and (Sen-FI is MuyCercano) and (Sen-LD is Cercano) and (Sen-LI is MuyCercano) then (Direccion-Ang is G-Derecho)(Motores is G-Atras) (1)
22. If (Sen-FD is MuyCercano) and (Sen-FI is MuyCercano) and (Sen-LD is MuyCercano) and (Sen-LI is Cercano) then (Direccion-Ang is G-Izquierdo)(Motores is G-Atras) (1)
```

Figura 97. Base de conocimientos

### Vista Frontal



### Vista Superior

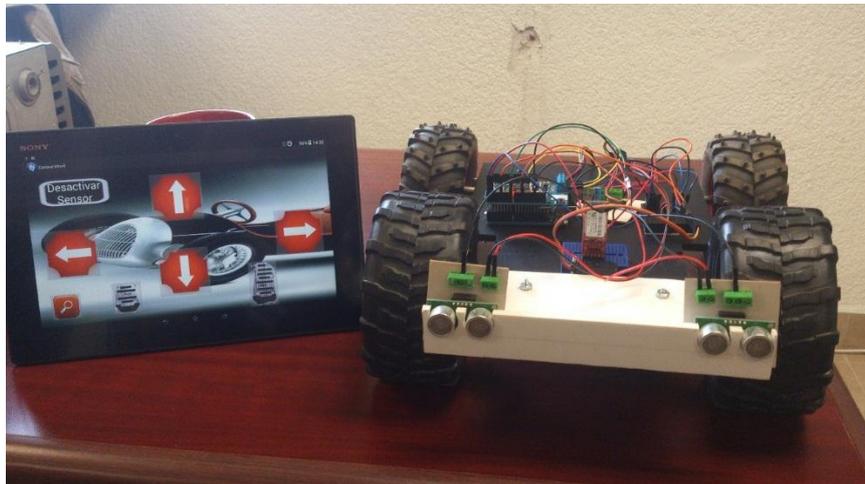


Figura 98. Sistema AA-AR y Robot Móvil

# **CAPÍTULO V**

## **CONCLUSIONES Y TRABAJO A FUTURO**

## 5.1 CONCLUSIONES

Durante el desarrollo del sistema surgieron algunos elementos que no estaban considerados:

- Escalabilidad  
Es la capacidad de un sistema en crecer sin perder la calidad en sus servicios.
- Capacidad de almacenamiento del Dispositivo Móvil  
Se refiere a la asignación de memoria
- Condiciones ambientales  
Enfocado a la cantidad de iluminación
- Prioridad de los procesos  
Asignación de los recursos en el ciclo de vida del S.O. Android.

Los primeros experimentos que se realizaron con realidad aumentada, se utilizó la biblioteca NyArToolkit con Processing, resultando adecuada para el aprendizaje, pero no en la práctica debido a la cantidad de falsos positivos que generaba, esto se debe a sus algoritmos pocos robustos y el diseño de su marcadores que solo permitan formas cuadradas.

Las formas cuadradas son un inconveniente ya que limitan el conjunto de marcadores a diseñar, y si se deseaba usarlo en la vida cotidiana, no era sustentable ya que se tiene que diseñar un marcador por cada señal de tránsito. El solo pensar en diseñar “n” cantidad de marcadores para cada señal de tránsito limitarían al sistema.

Al ser implementado en un dispositivo móvil la biblioteca NyArToolkit no funciona adecuadamente con Android ya que algunos de sus elementos sobre-escriben a los de Android y al no tener un ciclo de vida, llegaba a generarse un cuello de botella.

Por lo que se planteó cambiarse a otra biblioteca llamada Vuforia, dedicada al uso exclusivo de los dispositivos móviles desarrollada por Qualcomm.

Vuforia se caracteriza en implementar un ciclo de vida parecida a la de Android y al mismo tiempo aprovecha adecuadamente el procesador. Los algoritmos utilizados para la realidad aumentada son más robustos, ya que aceptan diferentes tipos de marcadores de acuerdo a las necesidades del usuario o del desarrollador.

Esta biblioteca tiene la capacidad de ser escalable, ya que brinda un servicio de alojamiento, es decir, si el desarrollador desea agregar un nuevo marcador, automáticamente el usuario ya lo tendrá disponible o el mismo usuario puede usar el sistema para agregar un nuevo marcador esta función debe ser activada por el desarrollador.

Sus marcadores no se limitan a formas cuadradas, por lo que es ideal, si se desea utilizar las mismas señales de tránsito como marcadores (targets) para representar la realidad

aumentada y no importa si el marcador está cubierto de manera parcial, este será reconocido, lo que no sucede con NyArToolkit.

Los objetivos planteados en la tesis se lograron, demostrando que la realidad aumentada no solo está dedicada a medios de entretenimiento sino también a medios de toma de decisiones:

- Diseño y desarrollo de un modelo de interfaz hombre/máquina basado en AR
- Diseño de un modelo de navegación autónomo.
- Integración de los modelos.
- Implementación de algoritmos de Reconocimiento de Patrones (Marcadores)
- Diseño y construcción del móvil
- Diseño e implementación de algoritmos, capaces de discernir, las decisiones del usuario y del entorno, para desplazar adecuadamente el robot móvil
- Asistir y guiar constantemente al conductor
- Diseño y desarrollo de un sistema multitarea

## **5.2 TRABAJO A FUTURO**

A partir de los resultados obtenidos por el sistema, se plantean los siguientes tópicos para los investigadores interesados:

- Algoritmos de tracking mediante GPS
- Control de Información para representar Objetos Virtuales
- Homografía de los marcadores
- Condiciones de iluminación de los marcadores
- Creación de sistemas interactivos para encontrar la falla mecánica o eléctrica del automóvil
- Asistencia por voz usando la realidad aumentada
- Cálculo de trayectorias
- Creación de biblioteca de realidad aumentada
- Sistema de seguimiento tipo encadenamiento
- Sistemas de aprendizaje utilizando realidad aumentada
- Modelación de problemas: físicos, químicos, biológicos, etc.

La realidad aumentada está en constante evolución, pero es muy poco conocida por los usuarios finales por lo que el campo de estudio es finito.

## 6. REFERENCIAS

- A., K. (1996). Fuzzy Mathematical Techniques with Applications. In K. A., *Fuzzy Mathematical Techniques with Applications*. Addison-Wesley.
- A.K., J. (n.d.). Fundamentals of Digital Image Processing. Prentice-Hall.
- Ackermann, R. (n.d.). *Ackerman Steering Principle*. Retrieved from Ackerman Steering Principle:  
[http://www.rctek.com/technical/handling/ackerman\\_steering\\_principle.html](http://www.rctek.com/technical/handling/ackerman_steering_principle.html)
- (n.d.). *Algoritmos de segmentación por color*.
- Aschman, T. (n.d.). *The Ackermann Principle of Steering*.
- Beatriz Martin Guadaño, A. M. (n.d.). Identificación Óptica de la posición y orientación de un vehículo aéreo no tripulado. *Sistemas Informáticos Cursos 2008-2009*. España.  
Retrieved from Identificación Óptica de la posición y orientación de un vehículo aéreo no tripulado: <http://147.96.80.209/PaginaWeb2/calibracion.html>
- Benito, D. d. (2004). *Detección automática del color de la piel en imágenes Bidimensionales basado en el análisis de regiones*.
- Charles Loop, Z. Z. (1999). Computing Rectifying Homographies for Stereo Vision. *Microsoft Research - Microsoft Corporation One Microsoft Way, 14*.
- D. Aracena, C. T. (2002). Algoritmo de Seguimiento para aplicaciones en Realidad Aumentada. *Encuentro Chileno de la Computación, Copiapo*. Chile.
- Demouth H., B. M. (n.d.). Manual: "Neural Network Toolbox". In B. M. Demouth H., *Manual: "Neural Network Toolbox"*. MathWorks Inc.
- Denso-Wave. (2011, October 15). QR Code Features. *QR Code Features*.
- Department, E. E. (n.d.). *The Berkeley Initiative in Soft Computing*. Retrieved from University:  
<http://www.cs.berkeley.edu/~zadeh/>
- Developers, G. (n.d.). *Developer Android - Sensor*. Retrieved from Developer Android - Sensor:  
<http://developer.android.com/reference/android/hardware/SensorEvent.html>
- Enciclografia*. (n.d.). Retrieved from Historia y origen de las Señales de Tráfico:  
[http://www.sitographics.com/conceptos/temas/historia/Traffics\\_sign\\_history.html](http://www.sitographics.com/conceptos/temas/historia/Traffics_sign_history.html)
- Faugeras, O. (1993). Tridimensional Computer Vision: A Geometric Viewpoint. *Artificial Intece. MIT Press*.
- Federal, G. d. (n.d.). *Secretaria de Transporte y Vialidad*. Retrieved from Secretaria de Transporte y Vialidad: <http://www.setravi.df.gob.mx/wb/stv/restrictivas>

- Fiala, M. (2004, November). ArTag Revision 1. A Fiducial Marker System Using Digital Techniques. *NRC/ERB-1117*. National Research Council Canada.
- Fontela, C. (2011). UML . In C. Fontela, *UML Modelado de software para profesionales* (p. 165). Alfaomega.
- Francisco Garcia Arenas, M. L. (1998). Jordan's Curve Theorem. *Divulgaciones Matemáticas* , 43-60.
- Ghada Moussa, E. R. (2012, April). Augmented Reality Vehicle System: Left-turn maneuver study. *Transportation Research Part C: Emerging Technologies*. ELSEVIER.
- Google, A. (n.d.). *Android Developers*. Retrieved from Android Developers: <http://developer.android.com/sdk/index.html>
- Group, K. (n.d.). *OpenGL*. Retrieved from OpenGL: <http://www.opengl.org/>
- GS1 DataMatrix. (n.d.). *An Introduction and Technical overview of the most advanced GS1 Application Identifiers compliant symbology*. Belgium.
- Guideline for In-vehicle Display Systems Virsion 3.0. (n.d.). *Japan Automobile Manufactures Association*.
- H, Z. (1996). Fuzzy set Theory and its aplicaciones. In Z. H, *Fuzzy set Theory and its aplicaciones*. Kluwer Academic Publishers.
- Hartley, R. (1995). In Defence of the 8-point Algorithm. *Proceeding of the IEEE International Conference on Computer Vision*.
- Hayet, D. J. (2011, August). nyARToolkit + Processing. *Centro de Investigación en Matemáticas*.
- Hirokazu Kato, M. B. (2000, November). Augmented Reality Applications - Human Interface. *ARToolkit* . Washington.
- Identificación óptica de la posición y orientación de un vehiculo aéreo no tripulado*. (n.d.). Retrieved from Identificación óptica de la posición y orientación de un vehiculo aéreo no tripulado: <http://147.96.80.209/PaginaWeb2/calibracion.html>
- IGD. (n.d.). *Project Group for Augmented Reality* .
- Inc. Ned Gulley, J. R. (n.d.). Manual: "Fuzzy Logic Toolbox". In J. R. Inc. Ned Gulley. MathWorks Inc.
- Isern, G. J. (2003, Julio 8). Estudio experimental de métodos de calibración y autocalibración de cámaras.
- J. Weng, T. H. (1989). Motion and Structure from two Perspective Views: Algorithms, Error Analysis and Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence Vol 11*, (pp. 451-475).

- J.C, L. (1991). Robot Motion Planning. In L. J.C, *Robot Motion Planning*. Academics Publishers, Stanford University USA.
- Japanese Ministry of Land. (2007). *Infrastructure and Transport, Road Bureau*.
- Jorge A. Marcial Basilio, G. A. (2012). *Novel method for image detection using HSV and YCbCr color models*. México.
- José Luis Alba y Fernando Martín. (2006). *Colorimetría y Captura de Imagen*.
- Kato, D. H. (n.d.). Artoolkit. In J. M. Hocine Cherifi, *Digital Information and Communication Technology and Its Applications*. Springer.
- Kato, D. H. (n.d.). ARToolkit. In J. R. Greg Kipper, *Augmented Reality: An Emerging Technologies Guide to AR*.
- Kato, H. (2009). Biography. *IEEE VGTC Virtual Reality Technical Achievement Award 2009*. IEEE VGTC Virtual.
- Kato, H. (n.d.). Augmented Reality. *Computer Graphics and Applications*. IEEE.
- Knowlton, K. (1964). A Computer Technique for Producing Animated Movies. *Proceedings of the Spring Joint Computer Conference*. Washington D.C. : Spartan.
- Kosko, B. (1992). Neural Networks and Fuzzy Systems. In B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs: Prentice Hall.
- Lee, C.-T. L. (1996). Neural Fuzzy Systems. In C.-T. L. Lee, *Neural Fuzzy Systems*. Prentice Hall.
- M. Tuceryan, D. G. (1995). Calibration Requirements and Procedures for a Monitor-Based Augmented Reality System. *IEEE Trans. On Visualization and Computer Graphics* (pp. 255-273). IEEE Trans. On Visualization and Computer Graphics.
- Mizell, T. C. (1992). Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. *Hawaii International Conference on Systems Sciences, 1992*, (pp. 656-669). Hawaii.
- Navarrete Manzanilla Niels Henrik, P. B. (2011). *Sistema de Navegación Inteligente*. México.
- NyARToolkit Proyect. (n.d.). Retrieved from NyARToolkit Proyect - Documentation: <http://nyatla.jp/nyartoolkit/wp/>
- Operadores, D. d. (n.d.). *Secretaria de Transporte Gobierno del Estado de México*. Retrieved from Secretaria de Transporte Gobierno del Estado de México: [http://portal2.edomex.gob.mx/stransporte/licencias/guia\\_examen/index.htm](http://portal2.edomex.gob.mx/stransporte/licencias/guia_examen/index.htm)
- P., P. A. (1972). *Dinámica de la partícula - Radio de Curvatura*. Biblioteca Matematica.
- Pagaza, L. I. (n.d.). *Reglamento de Transito del Estado de México*. México.

- Personal guidance system for the visually impaired using GPS, GIS and Vr Technologies. (1993). *Proceedings of Conference on Virtual Reality and Persons with Disabilities*. Hawai.
- Pertusa Grau, J. F. (2003). *Técnicas de Análisis de Imágenes*. Universidad de Valencia.
- Qualcomm Connected Experiences, I. (n.d.). *Vuforia Developer*. Retrieved from Vuforia Developer: <https://developer.vuforia.com/>
- Qualcomm. (n.d.). *Qualcomm*. Retrieved from Qualcomm: <http://www.qualcomm.com/>
- Qualcomm. (n.d.). *Vuforia Developer*. Retrieved from Vuforia Developer: <https://developer.vuforia.com/resources/dev-guide/getting-started>
- Qualcomm. (n.d.). *Vuforia Developer*. Retrieved from Vuforia Developer: <https://www.vuforia.com/platform>
- Qualcomm. (n.d.). *Vuforia Developer*. Retrieved from Vuforia Developer: <https://developer.vuforia.com/resources/dev-guide/vuforia-ar-architecture>
- Quddus, M. O. (2003). A General Map Matching Algorithm for Transport Telematics Applications. *The GPS Solutions Journal* 7(3).
- Reas, B. F. (n.d.). *Processing 2*. Retrieved from Processing 2: <http://www.processing.org/>
- RGB to HSV color space conversion*. (n.d.). Retrieved from RGB to HSV color space conversion: [http://en.literateprograms.org/RGB\\_to\\_HSV\\_color\\_space\\_conversion\\_%28C%29](http://en.literateprograms.org/RGB_to_HSV_color_space_conversion_%28C%29)
- Rosenfeld, A. (1970). Connectivity in digital pictures. *Journal of ACM*, 146-160.
- Rosenfeld, A. (1974). Digital Topology. *IEEE Trans. on Computers*, 1264-1269.
- Santiago, D. J. (n.d.). *Introducción a la Morfología Matemática de Conjuntos*. Fondo de Cultura Economica.
- Sawano, H. O. (2004, February). Study on An Augmented Reality-based Rendering Method for Car Navigation Utilizin Real-time Video in FCV 2004. *Tenth Kore-Japan Joint Workshop on Frontiers of Computer Vision, Fukuoka Japan*. Japan.
- Stockman, L. S. (2000). Computer Vision. *Computer Vision updated*.
- Stuart Russell, P. N. (n.d.). Inteligencia Artificial Un enfoque moderno. In P. N. Stuart Russell, *Inteligencia Artificial Un enfoque moderno*.
- Sutherland, I. (1964). Sketchpad-A Man-Machine Grapfical Communication System. *Proceedings of the Spoings Joint Computer Conference*. Washington, D.C. Spartan.
- Sutherland, I. E. (n.d.). The Ultimate Display. *Information Processing Techniques*. ARPA, OSD.
- Team, A. (n.d.). *Arduino*. Retrieved from Arduino: <http://arduino.cc/>
- Technologies, U. (n.d.). *Unity*. Retrieved from Unity: <http://unity3d.com/es>

- Vallino, J. (n.d.). *Tesis Doctoral Computer Science for Augmented Reality*. University of Rochester.
- Verri., E. T. (1998). *Introductory to techniques for 3D computer vision*. Prentice Hall.
- Vuforia. (n.d.). *Qualcomm Vuforia Developer Portal*. Retrieved from Qualcomm Vuforia Developer Portal: <https://developer.vuforia.com/target-manager>
- Vuforia. (n.d.). *Qualcomm Vuforia Developer Portal*. Retrieved from Qualcomm Vuforia Developer Portal: <https://developer.vuforia.com/resources/dev-guide/create-targets>
- Vuforia. (n.d.). *Qualcomm Vuforia Developer Portal*. Retrieved from Qualcomm Vuforia Developer Portal: <https://developer.vuforia.com/resources/dev-guide/trackable-base-class>
- Vuforia. (n.d.). *Qualcomm Vuforia Developer Portal*. Retrieved from Qualcomm Vuforia Developer Portal: <https://developer.vuforia.com/resources/dev-guide/natural-features-and-rating>
- Wickens. (1984). *Engineering Psychology and Human Performance*. Columbus.
- Wickens, C. &. (1983). The Dissociation between subjective workload and Performance. *Congreso de la Human Factors Society*. Santa Monica, CA.
- Wickens., Y. Y. (1988). Dissociation of Performance and Subjective Measures of Workload. *Human Factors*.
- Woods, G. &. (n.d.). *Digital Image Processing - Features*.
- Ynjiun P. Wang, A. Y. (2000, Abril 25). US6053407 A. *Maxicode data extraction using spatial domain features*. US.
- Zadeh, L. A. (n.d.). *Electrical Engineering and Computer Sciences*. Retrieved from Electrical Engineering and Computer Sciences: <http://www.eecs.berkeley.edu/Faculty/Homepages/zadeh.html>
- Zhang, A. (1999). *A Flexible New Technique for Camera Calibration*. Retrieved from A Flexible New Technique for Camera Calibration: <http://research.microsoft.com/~zhang>
- Zhencheng, H. K. (2004). Solution of Camera Registration Problem Via 3D-2D Parameterized Model Matching for On-Road Navigation. *Internation Jouranl of Image and Graphics*.
- Zhongfei Zhang, A. R. (n.d.). 3D Reconstruction Based on Homography Mapping. *Department of Computer Science*. New York, Buffalo.

