



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA
COMPUTACIÓN

“SISTEMA DE VISIÓN DE UN ROBOT HUMANOIDE
EN UN AMBIENTE SEMI ESTRUCTURADO”

TESIS

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN CIENCIAS
(COMPUTACIÓN)

P R E S E N T A:

ANDRÉS FERNANDO
ESPÍNOLA AUADA*

TUTOR:

TETYANA BAYDYK

México, D.F.

2011

*Exbecario CONACYT



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A Araceli y Paloma.

Agradecimientos

Primeramente a Dios, quien me ha permitido llegar hasta este punto de mi vida, dándome las fuerzas para recorrer el camino de la investigación.

A mi familia en Paraguay, por su apoyo incondicional en estos siete años y medio fuera de casa. A mi familia en México, Toño, David, Elena y Gabriel, que me acompañaron en mis esfuerzos, desveladas y diversión necesarias para acabar este trabajo. A mi Tutora la Dra. Tetyana Baydyk, quien me guió en este trabajo de tesis y las publicaciones que conseguimos derivados del mismo. A los cinco sinodales de la presente tesis, por sus aportaciones al mejoramiento de la presentación de la misma. A mis compañeros de trabajo desde el 2008, todos los miembros del "Laboratorio de Robótica Móvil y Sistemas Automatizados de la Universidad La Salle". Al Dr. Jacky Baltes, por su invitación a realizar la estancia de investigación con su grupo de trabajo en el "Autonomous Agent Lab", de la Universidad de Manitoba, Canadá. A mis queridísimos amigos, con el cual compartimos increíbles momentos tanto en México así como en otras partes del globo, UrticariaMan y KatanaGirl. A mis compañeros en Canadá, Dr. Chi Tai Cheng, Lau Meng Cheng, Derek Cormier y Andrew Winton, que hicieron muy placentera mi estadía en Winnipeg. "¡No solo no hubiera sido nada sin ustedes, sino con toda la gente que estuvo a mi alrededor desde el comienzo; algunos siguen hasta hoy! ¡Gracias totales!"

El presente trabajo contó con el apoyo de CONACYT bajo el esquema de Beca Nacional de Posgrado para estudios de Maestría, durante el periodo Agosto 2009 - Julio 2011.

Contenido

1. Introducción	1
1.1. Objetivos	2
1.2. Justificación	2
1.3. Aportación	3
1.4. Metodología	3
1.5. Alcances y limitaciones	4
1.6. Organización del trabajo	4
2. Antecedentes y Estado del Arte	5
2.1. RoboCup	5
2.1.1. Cancha de fútbol de la categoría "KidSize"	6
2.1.2. Postes de medio campo	6
2.1.3. Orientación	7
2.2. Robot humanoide	8
2.3. Sistema de visión	10
2.3.1. Visión global y visión local	10
2.3.2. Sistemas de visión local en la RoboCup	11
2.4. Algoritmos de reconocimiento y clasificación	13
2.4.1. Algoritmo Scale Invariant Features Transforms (SIFT)	13
2.4.2. Algoritmo Speeded Up Robust Features (SURF)	15
2.4.3. Redes Neuronales	15
2.4.4. Método de semejanza gráfica	16
2.4.5. Transformación Hough generalizada	16
2.4.6. Semejanza de patrones	16
3. Espacio de Color y Segmentación	17
3.1. Espacio de color	17
3.1.1. Red, Green, Blue (RGB)	17
3.1.2. Hue, Saturation, Value (HSV)	18
3.1.3. Modelo YCbCr	19

3.2. Segmentación por color	19
3.2.1. Cajas delimitadoras de color	20
3.2.2. Segmentador GUI o <i>grafical user interface</i>	21
3.3. Experimentación y comparación	22
3.3.1. Correlación	23
3.3.2. Resultados de la comparación	23
3.3.3. Elección	29
4. Adaptación del sistema de visión a cambios en la iluminación	30
4.1. Introducción	30
4.1.1. Limitante Computacional	30
4.1.2. Trabajos previos	31
4.1.3. Motivación	33
4.2. Calibración activa	33
4.2.1. Cámara IDS uEye	33
4.2.2. Modelado del tiempo de exposición	35
4.2.3. Sistema Fuzzy	36
4.3. Algoritmo general de calibración	39
4.4. Experimentos	41
4.4.1. Experimento 1	41
4.4.2. Experimento 2	41
4.5. Discusión	45
5. Metodología del Procesamiento de Imágenes	46
5.1. Tratamiento de los píxeles	46
5.1.1. Conversión a HSV	46
5.1.2. Clasificación de píxeles y generación de imágenes binarias	48
5.2. Tratamiento de las imágenes binarias	50
5.2.1. Filtros morfológicos	50
5.2.2. Operaciones: dilatación y erosión	50
5.2.3. Filtros: apertura y cerradura	51
5.2.4. Extracción de contornos	51
5.2.5. Separación de objetos	54
6. Reconocimiento y clasificación de objetos clave	56
6.1. Objetos Clave	56
6.2. Reconocimiento de postes de medio campo	57
6.3. Reconocimiento y clasificación de porterías	58
6.3.1. Descriptores geométricos	59
6.3.2. Momentos invariantes de Hu	60

6.3.3. Clases de imágenes de porterías	62
6.3.4. Clasificación	64
6.4. Base de datos	65
6.4.1. Conjunto de datos	66
6.4.2. Optimización de la base de datos	68
6.5. Pruebas preliminares y resultados	69
6.5.1. Prueba 1: postes de medio campo	69
6.5.2. Prueba 2: portería	69
6.6. Clase “no portería”	70
6.6.1. Filtro de área y proporción	70
6.6.2. Nueva clase	71
6.7. Resultados en competencia	71
6.7.1. Singapur 2010	71
6.7.2. Turquía 2011	72
7. Conclusiones y Trabajo Futuro	73

Índice de Tablas

2.1. Dimensiones de la cancha (en centímetros)	7
3.1. Comparación de la correlación y la segmentación del color amarillo en los distintos espacios de color en la escena	25
3.2. Comparación de la correlación y la segmentación del color azul en los distintos espacios de color en la escena	27
3.3. Comparación de la correlación y la segmentación del color naranja en los distintos espacios de color en la escena	27
4.1. Relaciones	34
4.2. Relaciones	35
4.3. Base de conocimiento del sistema Fuzzy	37
4.4. Resultados del experimento 1	41
4.5. Resultados del experimento 2	45
5.1. Valores de segmentación hallados con el programa segmentador GUI para el conjunto de imágenes	48
6.1. Invariantes de Hu para la misma imagen	61
6.2. Invariantes de Hu para la misma imagen	62
6.3. Pruebas de clasificación	69
6.4. Resultado de la clasificación sobre las imágenes con portería	70
6.5. Medidas de la portería vistas a la máxima distancia en la cancha	71
6.6. Resultados de los partidos disputados en el 2010	72
6.7. Resultados de los partidos disputados en el 2011	72

Lista de Figuras

2.1. Cancha en vista aérea [1].	6
2.2. Cancha en perspectiva.	7
2.3. Esquema del robot humanoide [1].	8
2.4. Arquitectura del robot "T1".	9
2.5. Visión monocular montado sobre un sistema de servomotores con dos grados de libertad.	11
2.6. Esquema de visión estéreo.	12
2.7. Esquema de un sistema de visión omnidireccional con dos espejos. 1) Cámara, 2) Espejo superior, 3) Espejo inferior, 4) Punto ciego, 5) campo de visión.	13
3.1. Filtro bayer en el sensor CMOS.	18
3.2. Cajas en el espacio 3D cartesiano representando 6 colores de interés.	20
3.3. Interfaz gráfica del programa segmentador	21
3.4. Interfaz gráfica del programa segmentador mostrando las zonas verdes, amarillas y azules.	22
3.5. Amarillo. Escena base: a) Color sin segmentación, b) Segmentación HSV, c) Segmentación RGB, d) Segmentación YCbCr	24
3.6. Amarillo. Escena menos luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr	24
3.7. Amarillo. Escena más luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr	25
3.8. Azul. Escena base: a) Color sin segmentación, b) Segmentación HSV, c) Segmentación RGB, d) Segmentación YCbCr	26
3.9. Azul. Escena menos luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr	26
3.10. Azul. Escena más luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr	27
3.11. Naranja. Escena base: a) Color sin segmentación, b) Segmentación HSV, c) Segmentación RGB, d) Segmentación YCbCr	28

3.12. Naranja. Escena menos luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr	28
3.13. Naranja. Escena más luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr	29
4.1. a) Esquema de constancia de color, b) Esquema donde no existe constancia de color . . .	31
4.2. Comportamiento del tiempo de exposición: a) Con respecto a la correlación, b) Con respecto a los píxeles correctamente clasificados	35
4.3. Correlación entre histogramas	38
4.4. Píxeles clasificados	38
4.5. Variación del <i>exposure time</i>	39
4.6. Diagrama de la metodología de calibración activa.	40
4.7. Escenas de prueba: desde la cantidad mínima de luz (a), a la cantidad máxima de luz (f)	42
4.8. Escena 1: Segmentación del azul, (a) Imagen base, (b) Condiciones alteradas de iluminación, (c) Histogramas antes del ajuste, (d) Histogramas después del ajuste del <i>exposure time</i>	43
4.9. Escena 2: Segmentación del naranja, (a) Imagen base, (b) Condiciones alteradas de iluminación, (c) Histogramas antes del ajuste, (d) Histogramas después del ajuste del <i>exposure time</i>	44
4.10. Escena 3: Segmentación amarillo, (a) Imagen base, (b) Condiciones alteradas de iluminación, (c) Histogramas antes del ajuste, (d) Histogramas después del ajuste del <i>exposure time</i>	44
5.1. Metodología general del procesamiento de imágenes.	47
5.2. (a), (b) y (c) Imágenes en formato RGB. (d), (e) y (f) Respectivas conversiones al espacio de color HSV pero desplegado en RGB	48
5.3. (a), (b) y (c) Imágenes segmentadas con los valores de amarillo, azul y verde. Todos los demás píxeles no clasificados se pusieron en negro. (d), (e) y (f) Imágenes binarias del color amarillo. (g), (h) e (i) Imágenes binarias del color azul. (j) (k) (l) Imágenes binarias del color verde.	49
5.4. Diferentes formas de elementos estructurantes	50
5.5. Imágenes de la figura 5.3 después de los filtros morfológicos.	52
5.6. Máscaras de escaneo hacia adelante y hacia atrás respectivamente.	53
5.7. (a), (b) y (c) Contornos hallados a las imágenes sin filtrar. (d), (e) y (f) Contornos hallados a las imágenes después de los filtros morfológicos.	54
5.8. Rectángulos delimitadores de los objetos encontrados.	55
6.1. Puntos característicos en círculos blancos de los postes de medio campo y las porterías.	57
6.2. Porterías azul y amarillo antes del 2010 [2].	59

6.3. Imagen: a) Original, b) Rotación de 30° c) Traslación en x , d) Traslación en y , e) Escalamiento en 50 %, f) Todas las transformaciones anteriores	62
6.4. Imagen: (a) Original, (b) Rotación de 30° (c) Traslación en x , (d) Traslación en y , (e) Espejo, (f) Todas las transformaciones anteriores	63
6.5. Clases de la portería: a) Completa, b) Media grande, c) Media pequeña, d) Poste, e) No portería	64
6.6. Algoritmo de clasificación	65
6.7. Montaje de la cancha (vistas global y en acercamiento)	66
6.8. Vista aérea de la cancha . En rojo las observaciones utilizadas, en morado la imagen a 315° excluida por las reglas de la categoría "KidSize"	67

Capítulo 1

Introducción

La robótica en la actualidad cumple un papel fundamental en los procesos de automatización en la industria, haciendo que la producción en serie sea muy efectiva. La visión computacional vino a proporcionar una herramienta fundamental y de bajo costo para dotar de “ojos” a los procesos automatizados, donde ahora se puede diferenciar colores, formas y tamaños para tomar decisiones en estos procesos. Estos robots están diseñados para una tarea muy específica y se les podría llamar fijos, debido a que su movimiento está limitado a un brazo o un par de ellos, con la movilidad necesaria para su tarea. Pero desde hace un par de décadas, surgió una nueva corriente para la robótica donde se quiere lograr una movilidad real de los robots y hacerlos más accesibles para múltiples tareas, surgiendo así la idea de los robots móviles de propósito general. Con esto nace la idea de los robots humanoides, capaces de realizar tareas físicas similares a la de cualquier ser humano.

Muchos grupos de investigación alrededor del mundo se han concentrado en trabajar para lograr que este objetivo se cumpla en el mediano plazo. Este trabajo de tesis forma parte de la línea de investigación en robots humanoides jugadores de fútbol, del equipo “Cyberlords La Salle” [3,4], participante en la RoboCup. Lo que el grupo de investigación trabaja por lograr, es el diseño de *hardware* y *software* para que el robot humanoide tenga la inteligencia para jugar fútbol contra otros robots de manera autónoma. Pero, ¿por qué fútbol? El deporte es de mucha importancia para el ser humano, siendo una actividad física que a la vez en alguno de los casos, constituye eventos de entretenimiento para un determinado público. El fútbol es sin ninguna discusión, el deporte más importante en términos de difusión, público e interés, siendo el torneo de la FIFA (Fédération Internationale de Football Association) el mayor evento deportivo de todo el mundo, superando el doble de la audiencia de los Juegos Olímpicos [5], evento en donde se reúnen la mayoría de los deportes. Con esta misma idea, surge la iniciativa RoboCup (Robot World Cup initiative) [6], creando competencias de robótica en el campo de robots jugadores de fútbol primeramente, para luego expandir a otras categorías como robots de rescate y robots de servicio hogareño. Al tener una competencia, se está delimitando el campo de acción para que la investigación sea más certera, así como para proporcionar un espacio donde todos los grupos puedan medir resultados fuera de sus laboratorios, en ambientes no tan controlados. La visión computacional juega

un papel fundamental en dicha competencia, debido a que la cámara es un sensor extremadamente barato con el que se puede dotar al robot de una herramienta para tener conocimiento de su ambiente, en vez de usar sensores láser, que pueden proporcionar información precisa del área de trabajo, sin embargo el costo de dichos sensores es extremadamente elevado, además de ser muy frágiles.

1.1. Objetivos

El objetivo principal del trabajo de investigación es la creación del módulo del sistema de visión para un robot humanoide, que cumple con los requisitos para jugar fútbol. Este módulo debe de proveer al robot de información de su entorno, que lo haga capaz de orientarse dentro del ambiente de trabajo, así como el reconocimiento de objetos dentro del mismo. Para esto, se crearon dos grandes submódulos dentro del sistema de visión propuesto en este trabajo, cuyos objetivos son los siguientes:

- Módulo de autocalibración ante cambios de iluminación: tiene como objetivo proporcionar al módulo de visión la capacidad de ajustarse a posibles cambios en la iluminación del ambiente, permitiendo así que el robot pueda operar a diferentes horas del día dentro del ambiente de la competencia.
- Módulo de reconocimiento y clasificación: tiene como objetivo el análisis de las imágenes pertenecientes a la cancha de fútbol donde se realiza la competencia, buscando objetos y marcas claves para conocer su posición dentro de la cancha, así como poder orientarse hacia donde tiene que disparar para anotar goles.

1.2. Justificación

Esta línea de investigación del equipo “Cyberlords La Salle”, tiene como objetivo crear y optimizar algoritmos de visión para uso específico, tomando en cuenta la limitante de donde los algoritmos serán implementados y usados, tratando de hacer el menor uso posible de recursos computacionales para lograrlo. Los algoritmos realizados para la tarea de jugar fútbol, serán adaptados para el uso dentro de otras aplicaciones de uso comercial que se están desarrollando en la actualidad en el mismo equipo de trabajo. El desarrollo de los algoritmos de bajo costo computacional fueron de vital importancia debido al escaso poder computacional del modelo de robot para el cual estos se desarrollaron, así como también los sistemas a donde estos algoritmos se migrarán, considerando el alto paralelismo necesario para hacer funcionar sistemas embebidos del mismo estilo.

1.3. Aportación

En términos generales, esta tesis aporta en el área de tratamiento de la iluminación y clasificación de objetos en sistemas embebidos de bajo poder computacional. En términos particulares, proporciona

la capacidad de conocer el ambiente donde se desempeña un robot humanoide jugador de fútbol. Esto abre dos líneas de investigación dentro del grupo de trabajo, ambos centrados en la correcta percepción del robot en el ambiente de trabajo.

1.4. Metodología

Uno de los retos principales de la robótica humanoide es poder obtener información del sistema de visión que sea lo suficientemente robusto para navegar en el ambiente de trabajo, siendo la principal problemática el ruido, que se puede generar en el ambiente de trabajo o ser inherente del sensor de visión. En esta investigación se describen los métodos utilizados para lograr que el sistema entregue de manera certera la información referente a su ambiente de trabajo.

A continuación se detallan los pasos que se llevaron a cabo para realizar y completar el trabajo de investigación.

- Lo primero consistió en analizar la literatura correspondiente a los temas de visión computacional involucrados en el proyecto, particularmente aquellos relacionados con el tratamiento del problema del cambio de iluminación en el ambiente, así como los algoritmos para reconocer y clasificar objetos.
- Se procedió a analizar el *hardware* del robot, particularmente el sensor de visión, que es una cámara digital de uso industrial, para conocer las capacidades y funcionalidades disponibles en su programación. Se analizó el poder de cómputo del procesador del robot, para evaluar el posible rendimiento del mismo.
- Se realizaron experimentos para seleccionar el espacio de color adecuado para la tarea de visión.
- Se eligió el método de segmentación de imágenes acorde a las facilidades que proporciona el ambiente de trabajo.
- Se diseñó e implementó el módulo de autocalibración ante cambios de luz para controlar la cantidad de luz adquirida por la cámara para que no se afecte los umbrales de la segmentación.
- Se diseñó e implementó el reconocimiento y clasificación de los distintos objetos clave y sus puntos específicos de interés dentro del espacio de trabajo.
- Realización de pruebas y análisis de resultados sobre los datos de prueba.
- Se optimizaron los algoritmos y corrigieron los errores encontrados.
- Se probaron los algoritmos en el ambiente real (dos competencias internacionales de RoboCup, en los años 2010 y 2011).

1.5. Alcances y limitaciones

Este proyecto tiene como objetivo proveer al robot del conocimiento de su ambiente, para poder desenvolverse correctamente dentro de él y ser capaz de jugar fútbol. Cabe resaltar que los algoritmos están diseñados para la cancha actual de la competencia, con las marcas que ésta posee. Los algoritmos deberán adaptarse en un futuro si es que se cambian las marcas en la cancha de fútbol. Los algoritmos de visión entregan coordenadas específicas de las imágenes para que el módulo de autolocalización [7] desarrollado por el equipo de investigación, sea capaz de ubicar al robot en una determinada posición dentro de la cancha. La limitante principal consiste en el escaso poder computacional que posee el robot, que consta con una sola unidad de procesamiento para ejecutar todos los módulos del robot en paralelo, así que se tuvo que optimizar lo más posible los algoritmos a ser ejecutados, generando esto mucho trabajo de implementación y pruebas.

1.6. Organización del trabajo

En el capítulo 2, se hace un análisis detallado del ambiente de trabajo, tanto de la competencia RoboCup, así como la cancha de fútbol de la categoría "KidSize", que es el escenario principal donde se desenvuelve el robot. Seguidamente se describe la arquitectura del robot humanoide "T1", plataforma de *hardware* específica para la cual se enfocó este trabajo de investigación. Finalmente se hace una introducción a los sistemas de visión utilizados en la RoboCup en las diferentes categorías, así como los algoritmos de visión comúnmente utilizados en sistemas con mayor poder computacional. El módulo de calibración del sistema para soportar cambios de luz, se contempla en los capítulos 3 y 4. En el capítulo 3 se hace un análisis de los espacios de color a base de experimentaciones, para encontrar el formato que mejor se adecua a la necesidad de ser tolerante a ligeros cambios de iluminación, así como también el método de segmentación elegido para aislar objetos de interés, mientras que en el capítulo 4 se detalla un sistema experto con lógica fuzzy que ajusta parámetros de la cámara para poder hacer tolerante al sistema a cambios de luz más fuertes que no se puede solucionar con el espacio de color elegido. El módulo de reconocimiento y clasificación se contempla en los capítulos 5 y 6. En el capítulo 5 se presenta el procesamiento de imágenes realizado para eliminar el ruido inherente de la cámara así como ruido en el ambiente, hasta la separación total de los objetos y el fondo, facilitando la posibilidad de analizarlos. El capítulo 6 consta del reconocimiento y clasificación de los objetos clave dentro de la cancha, para así poder extraer información acerca de la posición de los objetos, detallando dos clasificadores distintos: uno para los postes de medio campo y otro para las porterías, con un rendimiento muy elevado. Finalmente se muestra algunos resultados de partidos jugados en la RocoCup en dos años consecutivos, validando así el trabajo de la tesis.

Capítulo 2

Antecedentes y Estado del Arte

En este capítulo se presentarán los antecedentes y el estado del arte del trabajo de investigación realizado para esta tesis. Se divide en 3 partes. Primero que nada, se explicará un poco acerca de la competencia de robótica internacional de nombre RoboCup, debido a que este trabajo se enfoca en una de las categorías de esta competencia. En la segunda parte, se hablará acerca del robot humanoide "T1", su arquitectura y limitaciones de *hardware*. En la tercera, se expondrán los tipos de sistema de visión utilizados en la RoboCup y los algoritmos de visión computacional que actualmente utilizan esos sistemas para percibir el ambiente, reconocer objetos, hacer seguimiento de objetos en movimiento, reconocer obstáculos, etc.

2.1. RoboCup

Las competencias de robótica móvil se iniciaron hace ya casi 30 años con el objeto de inspirar a estudiantes e investigadores a enfocar sus esfuerzos en el campo de la robótica, integrando una amplia gama de tecnologías y generando nuevo conocimiento en el camino. Lo que distingue a las competencias robóticas de la investigación tradicional es que proponen problemas estándar al mismo tiempo que proporcionan una medida de desempeño objetiva contra la cual se puede evaluar el resultado de cada una de las distintas soluciones alcanzadas. Además, la presencia de los medios informativos convierte a estas competencias en foros en los que los participantes pueden hacer llegar los resultados de su investigación a una amplia audiencia atrayendo de esta forma el interés de las nuevas generaciones. Entre las competencias más reconocidas a nivel mundial están las de RoboCup, la cual es una iniciativa internacional con la intención de fomentar el avance de la investigación en robótica e inteligencia artificial. La categoría principal con la que nació la RoboCup son los robots jugadores de fútbol. La idea de los robots jugadores de fútbol fue mencionada por primera vez por el Profesor Alan Mackworth (Universidad de British Columbia, Canada) en un artículo llamado "On Seeing Robots" publicado en un libro de Visión Computacional [8] en el año 1993. La propuesta de la Federación Mundial RoboCup, haciendo analogía a la propuesta de que una máquina venza al campeón mundial de ajedrez, cumplida

en 1997, plantea un nuevo reto para la comunidad científica: que los robots humanoides vengzan en un partido de fútbol a la selección campeona de la FIFA antes del 2050. La liga de Humanoides (Humanoid League) es la categoría que busca ese objetivo. Este trabajo de tesis se basa en la subcategoría "KidSize".

2.1.1. Cancha de fútbol de la categoría "KidSize"

El ambiente semi estructurado donde se realiza el juego es una cancha de fútbol como se muestra en la figura 2.1 en vista aérea con las dimensiones especificadas en la tabla 2.1. En su estado actual, el campo de fútbol de la liga RoboCup Humanoid está compuesto de distintas marcas coloridas que facilitan a los robots el uso de técnicas de segmentación de color en su sistema de visión para poder identificar objetos relevantes en la escena. El campo está formado por una alfombra verde uniforme sobre la cual un conjunto de líneas blancas contrastantes delimitan las diferentes secciones del campo, una de las porterías está coloreada de amarillo mientras que la otra es azul haciendo fácil la distinción entre la portería del contrincante y la propia, finalmente existen dos postes de medio campo con código de colores que se encuentran en los extremos de la línea de medio campo. Todas estas marcas tienen posiciones conocidas en tres dimensiones con respecto al marco de referencia del campo, como se definen en las reglas de la liga RoboCup Humanoid [1].



Figura 2.1: Cancha en vista aérea [1].

2.1.2. Postes de medio campo

Mención especial merecen los postes de medio campo, por ser de vital importancia para el reconocimiento de la orientación relativa del robot con respecto a las porterías. A continuación se encuentra una

Tabla 2.1: Dimensiones de la cancha (en centímetros)

A	Longitud de la cancha	600
B	Ancho de la cancha	400
C	Profundidad de la portería	50
D	Ancho de la portería	150
E	Longitud del área de gol	60
F	Ancho del área de gol	300
G	Distancia a la marca del punto penal	180
I	Diámetro del círculo del centro	120
J	Distancia de línea al borde	70
K	Distancia del poste de medio campo a la línea de la cancha	40

breve descripción de los dos postes: poste amarillo y poste azul. En la figura 2.2 se ve la perspectiva de la cancha con los dos postes de medio campo.



Figura 2.2: Cancha en perspectiva.

2.1.2.1. Poste amarillo

Es un poste cilíndrico con 15cm de radio, compuesto por 3 secciones de color, cada una de 15cm de altura, y su código de color es el siguiente: amarillo en las secciones superior e inferior, mientras que la sección intermedia es de color azul.

2.1.2.2. Poste azul

Es un poste cilíndrico con 15cm de radio, compuesto por 3 secciones de color, cada una de 15cm de altura, y su código de color es el siguiente: azul en las secciones superior e inferior, mientras que la sección intermedia es de color amarillo.

2.1.3. Orientación

La información que proporcionan los postes, tiene que ver con cuál de los dos postes el robot está observando dentro de la cancha. Si el robot está viendo el poste amarillo, significa que si gira sobre

su eje a mano derecha, se encontrará viendo la portería amarilla. Si el robot está viendo el poste azul, significa que girando sobre su eje a mano derecha, podrá encontrar la portería azul.

2.2. Robot humanoide

Los robots humanoides de la liga "KidSize", poseen una forma análoga al ser humano, incluyendo las proporciones de cabeza, extensión de extremidades y sensores. Tienen dos brazos, dos piernas, una cabeza unida a un torso, con una restricción de altura total de entre 30 a 60 cm, pudiendo tener uno o dos sensores de visión en la cabeza. El robot debe ser capaz de levantarse, mantenerse de pie, caminar y correr de manera bípeda. Esquema de robot humanoide mostrado en la figura 2.3.

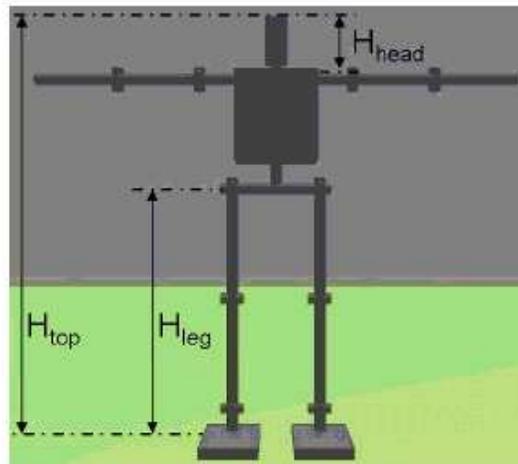


Figura 2.3: Esquema del robot humanoide [1].

Arquitectura general del robot "T1"

Para el concurso de la RoboCup edición 2010, en el equipo "Cyberlords La Salle", se desarrolló una arquitectura para un robot humanoide basado en la arquitectura comercial RoboNova-1 de HITEC [9]. En particular, se agregó un muy necesario grado de libertad vertical en los tobillos para que el robot ahora pudiera realizar movimientos giratorios sobre su eje de una manera mucho más eficiente. También se sustituyeron la mayoría de los servomotores HS-8498HB de las piernas por los HSR-5498SG, de mayor torque. El mecanismo de cabeza de *pan* y *tilt* está accionado por dos servomotores HS-8498HB, que da a la cámara una gama de mayor amplitud y rapidez de movimiento. Este diseño tiene un total de 20 grados de libertad y se representa en la figura 2.4

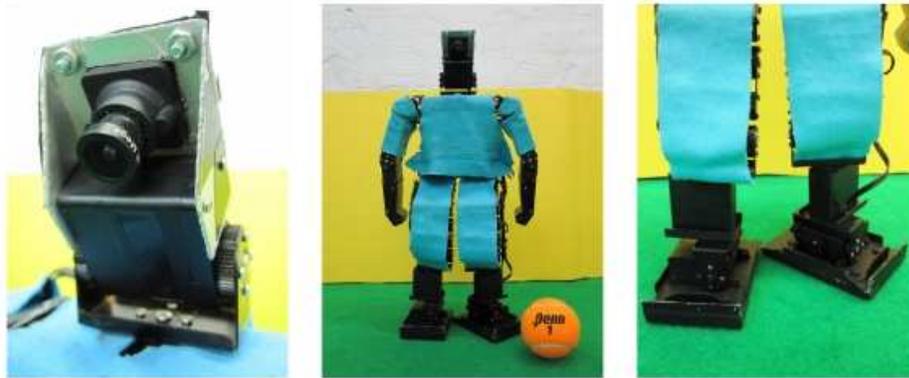


Figura 2.4: Arquitectura del robot "T1".

Sensores

Con el fin de dar al robot humanoide cierto grado de autonomía, se agregó tres tipos de sensores que se conectan al robot:

- Una única cámara de visión monocular uEye IDS-1226LE-C USB 2.0, que se monta en un sistema de *pan-tilt* de dos grados de libertad encima de los hombros del robot.
- Una unidad de seis grados de libertad de medición inercial (IMU), basado en LPR530AL ST y giroscopios LY530ALH más un acelerómetro ADXL335 de triple eje que permite al robot evitar caer dado un cierto grado de inclinación.

Unidad de procesamiento

La RB-100 (Roboard) [10] es la tarjeta controladora con la que el robot cuenta actualmente. Es una fusión entre el tradicional microcontrolador y un computador completamente funcional, todo lo que se necesita es una Micro SD Card (1-2Gb) para cargar el sistema operativo necesita de una alimentación de 6-24 volts de corriente directa. Tiene un tamaño pequeño de solo 96 x 56 mm, siendo de muy bajo consumo de energía. El procesador que trae es un Vortex86DX, una CPU x86 de 32bit corriendo a 1000MHz con 256MB DRAM. Esta capacidad de cómputo pareciera ser buena, inclusive comparable a la capacidad de una NetBook actual, pero su gran desventaja es que la arquitectura interna es i486, mientras que el procesador moderno está actualmente en la arquitectura i686. Esto hace que los algoritmos requieran más tiempo de procesamiento en la RB-100 a diferencia de un procesador de una Netbook. El sistema es compatible con Windows 98/ME, Windows XP, Windows Embedded CE, Windows XP Embedded, Windows Embedded Standard, Linux distribution kernel 2.4.24, 2.4.26. Todos los servomotores y sensores, incluyendo la cámara, están directamente conectados a la RB-100, que es la única unidad de computación programable a bordo del robot. Los servomotores Hitec tienen la interfaz con el RB-100 a través de los puertos PWM. Las salidas IMU se interconectan con la RB-100

con seis de los puertos de entrada del ADC, la cámara uEye IU-1226LE-C utiliza uno de los tres puertos USB 2.0 disponibles en la tarjeta.

2.3. Sistema de visión

Desde sus inicios, la visión por computadora se ha inspirado en el estudio del sistema visual humano. Este último sugiere la existencia de diferentes tipos de tratamiento de la información visual dependiendo de metas u objetivos específicos, es decir, la información visual percibida es procesada de distintas formas según las características particulares de la tarea por realizar. Por eso, la visión por computadora propone varias técnicas que permiten obtener una representación del mundo a partir del análisis de imágenes obtenidas desde cámaras web como modelos de visión (visión artificial).

Los sistemas de visión poseen numerosas aplicaciones en las más diversas disciplinas, pero todas apuntando a lo mismo: el procesamiento de los datos de la escena para la percepción de las máquinas de manera autónoma. Uno de los sensores principales del robot es la cámara que actúa como entrada para el sistema de visión. Este sistema tiene que proporcionar al robot de información útil para navegar sin inconvenientes dentro del área de trabajo. Tiene que ser capaz de reconocer su posición dentro de la cancha, así como conocer la posición de los objetos móviles dentro de la misma, como son la pelota, los robots del mismo equipo y los robots del equipo contrario.

Uno de los principales obstáculos para el sistema de visión es la iluminación dentro del lugar donde se lleva a cabo la competencia [11]. Una de las reglas aclara que el robot tiene que ser capaz de funcionar sin fuentes especiales de iluminación en la cancha, como sí lo tienen otras categorías de la RoboCup, forzando así a que el robot dependa solamente de la iluminación general del lugar de la competencia. Otra problemática es debido a que estos lugares (Centro de convenciones, expos, gimnasios, etc.) por lo general cuentan con ventanas que proporcionan una iluminación extra que va cambiando durante todo el día hasta ser totalmente nula en la noche, haciendo que una calibración de los colores en la mañana sea muy diferente a una realizada en la noche. Los distintos grupos de investigación participantes solucionan este problema realizando una calibración de color para las horas en que van a participar, teniendo que hacer hasta 4 o 5 calibraciones distintas durante un día de competencia, siendo esto muy ineficiente. Una vez que se soluciona esto, en la cancha existen puntos fijos que pueden ser reconocidos, como son las porterías, una de cada color, las marcas de medio campo, las líneas de la cancha, etc.

El sistema de visión tiene que ser lo suficientemente robusto para ser consistente en todo momento y proporcionar la información en tiempo real en un equipo de cómputo muy limitado debido al reducido tamaño disponible en el robot.

2.3.1. Visión global y visión local

Los sistemas de visión global en la RoboCup consisten en aquellos donde el sensor de visión no se encuentra en el robot, sino que es externo a éste. Para ciertas categorías, existen cámaras por

encima de la cancha que funcionan como un servidor de sistema de visión, donde las cámaras tienen una visión completa del área de trabajo. Esto es utilizado en robots pequeños y de gran velocidad, que son incapaces de procesar por sí mismos la información proporcionada por la o las cámaras, siendo una computadora por equipo la encargada de procesar la información y controlar de manera inalámbrica a los robots. Se han presentado varios trabajos referentes a como optimizar este sistema de visión [12], [13] pero este trabajo no se centrará en este tipo de visión debido a que en la "Humanoid League" se utiliza el sistema de visión local. El sistema de visión local, a diferencia del anterior, es aquel donde el sensor de visión es parte de la arquitectura del robot y además el procesamiento de la imagen se lleva a cabo en la computadora a bordo. La gran diferencia, es que la cámara no puede captar todo el ambiente de trabajo en una sola imagen, dificultando así el proceso de reconocimiento de los elementos en la escena. Las cámaras más comunes no superan un ángulo de apertura de 70 grados.

2.3.2. Sistemas de visión local en la RoboCup

Todas las categorías de la RoboCup, salvo las de simulación hacen uso de un sistema de visión para navegar dentro de su ambiente de trabajo. A continuación se presenta una descripción de los tres tipos de sistema de visión local comunes en las diferentes categorías de fútbol.

2.3.2.1. Visión monocular

Es aquella que capta la escena tridimensional con una sola cámara. Por ende el procesamiento se realiza sobre una imagen plana que representa al espacio 3D captado en un instante de tiempo. Por lo general, la única cámara está montada en un sistema de servomotores con uno o dos grados de libertad (PAN y TILT). En la figura 2.5 se observa este esquema del sistema de visión monocular.



Figura 2.5: Visión monocular montado sobre un sistema de servomotores con dos grados de libertad.

Como se describe en [3], la arquitectura para la cual se desarrolló el trabajo de esta tesis está basado en un sistema monocular, debido a las limitaciones de *hardware* para poder procesar más de una imagen de la misma escena a la vez, a causa del elevado costo computacional que esto implica.

2.3.2.2. Visión estéreo

Es el sistema de visión que consta de dos cámaras digitales, captando una imagen cada una de la misma escena. El procesamiento está orientado a reconstruir la escena en 3D a partir de las dos imágenes, izquierda y derecha. Un aspecto que cabe destacar es que las cámaras están desplazadas una cierta distancia una respecto a la otra, tal y como ocurre con los ojos humanos. El trabajo del sistema consiste en identificar en ambas imágenes aquellos píxeles en las dos imágenes que se corresponden con la misma entidad física en la escena 3D, usando para ello algoritmos especializados. La distancia que separa estos píxeles se conoce como disparidad. La medida de la disparidad sirve para obtener la distancia a la que se sitúa físicamente ese objeto en la escena con respecto a las dos cámaras. En la figura 2.6 se puede observar el esquema. Cabe resaltar, que antes del año 2009, en la categoría de humanoides se permitía más de 2 cámaras, pero en adelante se restringieron a un máximo de dos para ser análogo al sistema de visión humano.



Figura 2.6: Esquema de visión estéreo.

2.3.2.3. Visión omnidireccional

Este sistema de visión, es la que cuenta con una cámara con visión omnidireccional, es decir, una cámara con un campo de 360 grados de vista en el plano horizontal, o con un campo visual que cubre (aproximadamente) toda la esfera de visión [14]. Las cámaras omnidireccionales son importantes en las zonas donde se necesita una gran cobertura del campo visual, como en la fotografía panorámica y la robótica. En la literatura, hay un conjunto exhaustivo de las técnicas que se pueden utilizar para crear un sistema de visión omnidireccional. Un método popular es la combinación de una cámara con espejos, siendo este sistema de visión omnidireccional el más comúnmente conocido como un sistema de cámara catadióptrico [15]. En general, estos espejos son o bien planos o curvos. En la figura 2.7 se muestra el esquema de un sistema de visión omnidireccional con dos espejos. Este sistema de visión sigue siendo el más importante fuera de la categoría de humanoides, en la cual se prohibió desde el año 2009.

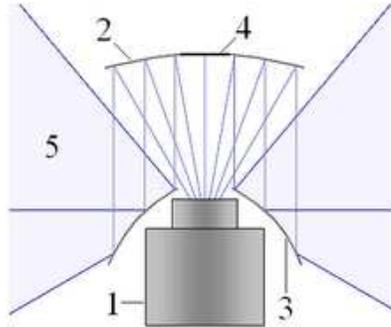


Figura 2.7: Esquema de un sistema de visión omnidireccional con dos espejos. 1) Cámara, 2) Espejo superior, 3) Espejo inferior, 4) Punto ciego, 5) campo de visión.

2.4. Algoritmos de reconocimiento y clasificación

La tarea de reconocimiento y clasificación es parte fundamental en este trabajo de tesis y a continuación se detallarán algunos algoritmos utilizados para tales fines en distintos sistemas de visión y la implicación de usarlos en el sistema monocular de esta investigación.

2.4.1. Algoritmo Scale Invariant Features Transforms (SIFT)

Scale Invariant Features Transforms (SIFT) es uno de los métodos de extracción de puntos invariantes ampliamente utilizado actualmente. Su nombre es debido a que transforma los datos de la imagen en coordenadas invariantes a la escala, rotación y en cierta medida al cambio de iluminación y al punto de vista 3D [16]. Consiste de cuatro pasos significativos:

- Detección de extremos en el espacio de escalas.
- Localización de los puntos clave.
- Asignación de orientación.
- Cálculo del descriptor de los puntos clave.

En las siguientes secciones se explicará de manera breve cada uno de los pasos del método SIFT. Éste fue uno de los principales algoritmos en que se analizó su viabilidad para este proyecto.

Detección de extremos en el espacio de escalas

Utilizando una diferencia de Gaussianas entre las imágenes se pueden encontrar los extremos máximos y mínimos de cada región de éstas. Al realizar una aplicación sucesiva de filtros Gaussianas sobre la imagen original, así como sobre las reducciones de la misma, se obtiene un espacio de escalas. Si se obtienen las diferencias entre imágenes adyacentes del mismo tamaño, se puede conseguir lo que

se conoce como una Diferencia de Gaussianas. Este proceso permite descubrir los puntos importantes en las imágenes, por medio de una búsqueda de máximos y mínimos sobre cada píxel con vecindad 26.

Localización de los puntos clave

Al evaluar cada píxel y comparar su valor con respecto a sus 8 vecinos en la misma imagen, sus 9 vecinos en la imagen superior y los 9 de la imagen inferior se puede determinar que este píxel es un máximo o un mínimo.

Una vez obtenidos los puntos extremos de las diferencias de Gaussianas, se utiliza un proceso de limpieza para eliminar los puntos clave que no aporten información relevante al algoritmo. Para esto se compara la intensidad de los puntos extremos contra un umbral. De esta manera los píxeles con muy baja iluminación son descartados. Un filtro de esquinas es aplicado posteriormente. Se utilizan núcleos de extracción de contornos para identificar aquellos puntos que no sean esquinas y eliminarlos. Los puntos extremos que pertenecen a las esquinas de los objetos son buenos puntos clave.

Asignación de orientación

Sobre cada punto clave encontrado, se obtienen los gradientes en una región alrededor de éste. De cada píxel en la región, se evalúan los cambios con respecto a los píxeles adyacentes en dirección x y y . Al colocar en un histograma de orientaciones y magnitudes los valores encontrados en la región, se asignan una o más orientaciones al punto clave.

Se considera un nuevo punto clave por cada orientación significativa que se encuentre por cada punto clave en una misma posición de la imagen. Todas las operaciones a realizarse sobre las imágenes se hacen con referencia a esta orientación para asegurar la invarianza a rotación.

Cálculo de descriptores

Los descriptores de los puntos clave son una identificación de éstos. Dichos descriptores representan una firma de cada punto clave. Esta firma es invariante a escala, rotación, traslación y cambios de iluminación. Para calcular los descriptores se coloca una mascarilla de dieciséis píxeles sobre cada punto clave. Esta mascarilla está ponderada por una mascarilla de Gauss. Se divide en 16 regiones de 4×4 píxeles y se calculan los gradientes de cada región. Estos gradientes se representan en un histograma de magnitudes y orientaciones de 8 clases. Para una mejor precisión, Lowe sugiere utilizar una expansión de Taylor para encontrar el valor exacto de los gradientes que muy probablemente se puede ubicar en medio de los píxeles y no en un único píxel necesariamente.

Para asegurar la invarianza a rotación, las coordenadas de la mascarilla y los valores de las orientaciones de cada gradiente se toman con referencia a la magnitud principal del punto clave. Una vez obtenidos los 16 histogramas de 8 clases, se construye un vector que contendrá estos 128 datos. Este vector es el descriptor del punto clave y se normaliza para asegurar invarianza a cambios de iluminación.

Con los descriptores calculados para cada punto clave, se puede realizar una base de datos con éstos. Para cada imagen a la que se desee identificar algún objeto se procede a obtener los descriptores de los puntos clave de la nueva imagen y se comparan con una norma Euclidiana con los descriptores en la base de datos. Cabe mencionar que para su experimento, Lowe utilizó una base de datos de 40,000 puntos clave [16].

2.4.2. Algoritmo Speeded Up Robust Features (SURF)

(*Speeded Up Robust Features (SURF)* es otro de los algoritmos más utilizados en la literatura para la extracción de puntos de interés invariantes [17]. La extracción de los puntos la realiza detectando en primer lugar los posibles puntos de interés y su localización dentro de la imagen. Posteriormente se representa la vecindad del punto de interés como un vector de características (por defecto con un tamaño de 64, aunque es posible aumentarlo). El descriptor SURF es relativamente parecido al propuesto en SIFT. El primer paso para obtener el descriptor una vez calculada la escala es el cálculo de la orientación del punto de interés para, posteriormente, calcular el descriptor SURF. Para obtener un punto invariante a la orientación se calcula el Haar-wavelet para las direcciones x e y en una región circular de radio $6s$, donde s es la escala del punto de interés [18]. Una vez calculados para todos los vecinos, se estima la orientación dominante calculando la suma de todos los resultados dentro de una ventana deslizante que cubre un ángulo de $\frac{\pi}{3}$. El cálculo del descriptor se realiza construyendo, primeramente, una región cuadrada centrada en el punto de interés y con un tamaño de $20s$. La región se divide regularmente en 4 sub-regiones, y para cada sub-región se calculan unas pocas características simples. Seguidamente se calculan la Haar-wavelet para x e y luego se suavizan los resultados mediante una Gaussiana, obteniendo dx y dy . Para cada sub-región se suman los resultados dx y dy , además de calcularse su valor absoluto $|dx|$ y $|dy|$. De este modo, cada sub-región proporciona un vector v compuesto por:

$$v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|). \quad (2.1)$$

El descriptor SURF se obtiene mediante la unión de los vectores de las sub-regiones. La característica principal de los puntos de interés SURF es la repetibilidad, si el punto es considerado fiable el detector encontrará el mismo punto bajo distintas perspectivas (diferente escala, orientación, etc). El vector obtenido por el algoritmo deberá ser distintivo y al mismo tiempo robusto al ruido, errores, y deformaciones geométricas y fotométricas.

2.4.3. Redes Neuronales

Varios trabajos han sido presentados en el campo de las redes neuronales artificiales para el reconocimiento de imágenes [19], [20]. Se han desarrollado distintas redes para distintas aplicaciones, pero debido al alto paralelismo de las redes neuronales, su mejor rendimiento se puede dar en sistemas paralelos, y como ya se presentó la arquitectura del robot junto con su unidad de procesamiento i486, se descartó la idea de un sistema de reconocimiento de objetos por redes neuronales.

2.4.4. Método de semejanza gráfica

Este método funciona sobre un modelo del contorno de un objeto en una imagen [21]. La idea es analizar segmentos de recta ortogonales que crucen ciertos puntos del contorno del objeto a reconocer. La búsqueda de características de la imagen se hace sobre de estas líneas. Las características similares son utilizadas para comparar un modelo con una imagen dada para el reconocimiento de objetos. Cualquier característica puede ser implementada con este método (color, brillo, textura o contraste). Este método requiere bastante memoria, sin embargo el tiempo de cómputo es reducido. Con condiciones específicas este método puede trabajar con imágenes de objetos parcialmente ocultas, ampliadas, reducidas o giradas. No funciona con imágenes que contengan varios objetos además que el reconocimiento de objetos pequeños se torna difícil.

2.4.5. Transformación Hough generalizada

En este método los contornos son aproximados mediante un conjunto de puntos tomados del contorno del objeto en la imagen [22]. Cada punto sobre el contorno del objeto se describe en términos de un punto de referencia ubicado dentro del contorno. La imagen de contorno para este método puede ser obtenida aplicando un filtro Laplaciano de Gauss. Con este método pueden reconocerse imágenes de objetos parcialmente ocultos, con diversa escala o girados. Este método no consume demasiada memoria sin embargo el tiempo de cómputo requerido es elevado, además no funciona con objetos pequeños en las imágenes.

2.4.6. Semejanza de patrones

La diferencia de cuadrados entre una imagen I y una imagen plantilla T es sumada respecto a sus píxeles:

$$r(x, y) = \sum_{i \in T} \sum_{j \in T} (I(x + i, y + j) - T(i, j))^2. \quad (2.2)$$

Si la suma resultante es mayor que un cierto umbral obtenido experimentalmente, el objeto plantilla es encontrado en I [23]. Este método consume bastante memoria y tiempo de procesador además que no es capaz de trabajar adecuadamente con cambios de brillo o escalamiento de las imágenes.

Capítulo 3

Espacio de Color y Segmentación

3.1. Espacio de color

La selección del espacio de color para la tarea de visión computacional, especialmente cuando ésta se realiza en un ambiente de iluminación no controlado [3, 24], es de vital importancia para una correcta funcionalidad del sistema. Para efectos de este trabajo, se consideraron 3 espacios de color. El primero es el RGB (red, green, blue) el cual es el más común en el campo del procesamiento digital de imágenes y, en la arquitectura del robot es el espacio nativo con el cual trabaja la cámara uEye [25] utilizada para la fase experimental de este trabajo. El segundo espacio de color es el HSV (Hue, Saturation, Value), el cual es una representación en coordenadas cilíndricas de los puntos en el modelo RGB y se analizará por su comportamiento ante cambios de iluminación. El tercer espacio de color a consideración es el YCbCr, el cual es el espacio nativo en muchas de las cámaras web y con un comportamiento similar al HSV ante los cambios de iluminación [26]. A continuación se detallarán las implicaciones de usar cada uno de estos espacios y se demostrará con una experimentación que el espacio de color adecuado para este trabajo es el HSV.

3.1.1. Red, Green, Blue (RGB)

La descripción RGB para un determinado color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: rojo, verde y el azul. Es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios. El sensor CMOS contenido en muchas cámaras digitales consta de un filtro Bayer como se muestra en la figura 3.1, lo cual implica que al momento en que se solicita una imagen desde la cámara, ésta entrega una imagen en codificación RGB, por medio de una interpolación del driver del dispositivo.

La principal ventaja de trabajar con este espacio de color, es que no se requiere de un procesamiento

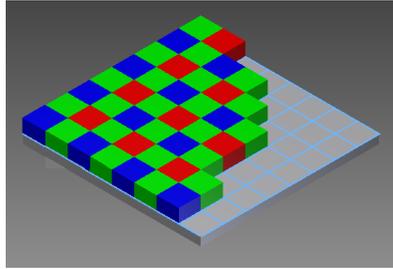


Figura 3.1: Filtro bayer en el sensor CMOS.

posterior de nuestra parte para obtener los valores de los píxeles deseados, lo que implica mayor rapidez en el procesamiento general subsecuente de las imágenes, pero teniendo como gran desventaja, la afectación negativa de los cambios de luz sobre este espacio de color.

3.1.2. Hue, Saturation, Value (HSV)

Los componentes del espacio de color HSV son matiz (hue), saturación (saturation) y valor (value). El matiz es el atributo que describe el color puro, la saturación es una medida del grado de dilución del color con respecto a la luz blanca y el valor corresponde al brillo del color. Los valores para el HSV pueden ser obtenidos a partir de los valores RGB, es decir, se pueden convertir cualquier valor RGB a uno en HSV usando las siguientes ecuaciones geométricas. [27].

$$V = MAX \quad (3.1)$$

$$S = \begin{cases} 0, & \text{if } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{otro caso} \end{cases} \quad (3.2)$$

$$H = \begin{cases} 43 \times (G - B) / (MAX - MIN), & \text{if } MAX = R \\ 85 + 43 \times (B - R) / (MAX - MIN), & \text{if } MAX = G \\ 171 + 43 \times (R - G) / (MAX - MIN), & \text{otro caso} \end{cases} \quad (3.3)$$

donde

- $MAX : \max \{R, G, B\}$,
- $MIN : \min \{R, G, B\}$,

Como se puede ver, para obtener los valores de los píxeles, cada uno de estos sufren una transformación que implica consumo de tiempo, pero en contrapartida, en varias tareas de visión computacional [28], [29], [30] ha demostrado tener un buen funcionamiento en ambientes no controlados de iluminación.

3.1.3. Modelo YCbCr

El espacio de color YCbCr ha sido definido en respuesta a la incrementada demanda de algoritmos digitales para el manejo de video y desde entonces su uso ha sido bastante amplio para el manejo de video digital. Pertenece a la familia de los espacios de color usados para la transmisión televisiva. Las cámaras web actualmente utilizan este espacio de color de forma nativa [31]. El parámetro Y indica la luminancia, los parámetros Cb y Cr indican el tono del color: Cb ubica el color en una escala entre el azul y el amarillo, Cr indica la ubicación del color entre el rojo y el verde. En este trabajo, se tendrá que transformar cada píxel a partir de los valores RGB y para esto se utilizarán las transformaciones correspondientes especificadas en ITU-R BT.709-5 standards of ITU (International Telecommunication Union).

$$Y = K_{ry} \cdot R + K_{gy} \cdot G + K_{by} \cdot B, \quad (3.4)$$

$$Cb = B - Y, \quad (3.5)$$

$$Cr = R - Y, \quad (3.6)$$

$$K_{ry} + K_{gy} + K_{by} = 1, \quad (3.7)$$

$$Y = K_{ry} \cdot R + K_{gy} \cdot G + K_{by} \cdot B, \quad (3.8)$$

$$Cb = K_{ru} \cdot R + K_{gu} \cdot G + K_{bu} \cdot B, \quad (3.9)$$

$$Cr = K_{rv} \cdot R + K_{gv} \cdot G + K_{bv} \cdot B, \quad (3.10)$$

donde

- $K_{ru} = -K_{ry}$,
- $K_{gu} = -K_{gy}$,
- $K_{bu} = 1 - K_{by}$,
- $K_{rv} = 1 - K_{ry}$,
- $K_{gv} = -K_{gy}$,
- $K_{bv} = -K_{by}$,
- $K_{ry} = 0.299$,
- $K_{by} = 0.114$,

3.2. Segmentación por color

La segmentación es el proceso por el cual se aísla una sección de imagen en varias posibles partes o grupos de píxeles u objetos. El objetivo de la segmentación es simplificar y cambiar la representación

de una imagen en otra más significativa y más fácil de analizar. La segmentación se usa tanto para localizar objetos como para encontrar los límites de estos dentro de una imagen. Como ya está descrito, la cancha de fútbol que es el ambiente de trabajo, tiene colores específicos que proporcionan una manera fácil de poder separar los distintos objetos basándose en el color. Una portería azul y amarilla, postes de medio campo con color azul y amarillo en combinación, pelota naranja, piso verde, etc.

3.2.1. Cajas delimitadoras de color

El método de segmentación que se utiliza es el de las cajas delimitadoras o "Bounding Box" [11], el cual consiste en crear cajas (3 ejes en el espacio cartesiano), utilizando valores máximos y mínimos para cada eje. Como los modelos RGB, HSV y YCbCr cada uno posee 3 ejes, se utilizan con valores máximos y mínimos respectivamente, de esta manera se encierra en una caja el color de interés, así todos los valores de píxeles que estén dentro de los límites se clasificarán dentro de cada región que se requiere. La figura 3.2 muestra 6 distintos colores encerrados con valores máximos y mínimos para el espacio de color HSV, siendo los ejes Hue, Saturation, Value.

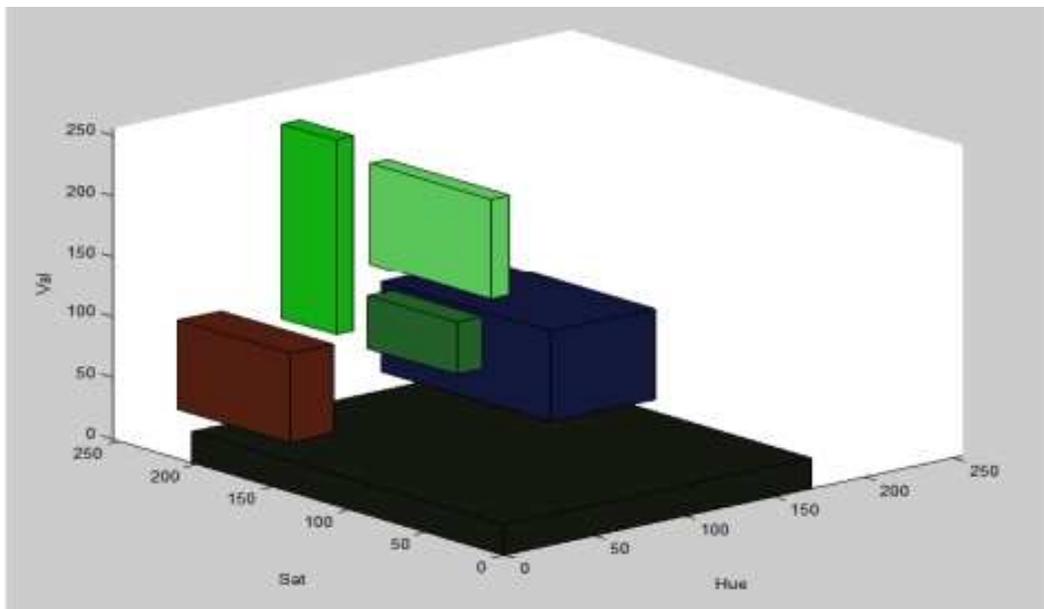


Figura 3.2: Cajas en el espacio 3D cartesiano representando 6 colores de interés.

Los límites máximos y mínimos se determinan fuera de línea y de manera manual con la ayuda de un programa desarrollado para tales fines y que se dará a conocer a más detalle en la siguiente sección.

3.2.2. Segmentador GUI o *grafical user interface*

Se desarrolló un programa de interfaz gráfica como herramienta para encontrar de manera fácil y rápida los valores delimitantes de las cajas de color. El programa tiene como entrada un video guardado en archivo o un video adquirido desde el robot en tiempo real via TCP/IP. La figura 3.3 muestra la interfaz gráfica del programa segmentador. Este programa fue realizado en el sistema operativo GNU-Linux, en el language de programación C++, haciendo uso de las librerías Qt 4.6 [32].

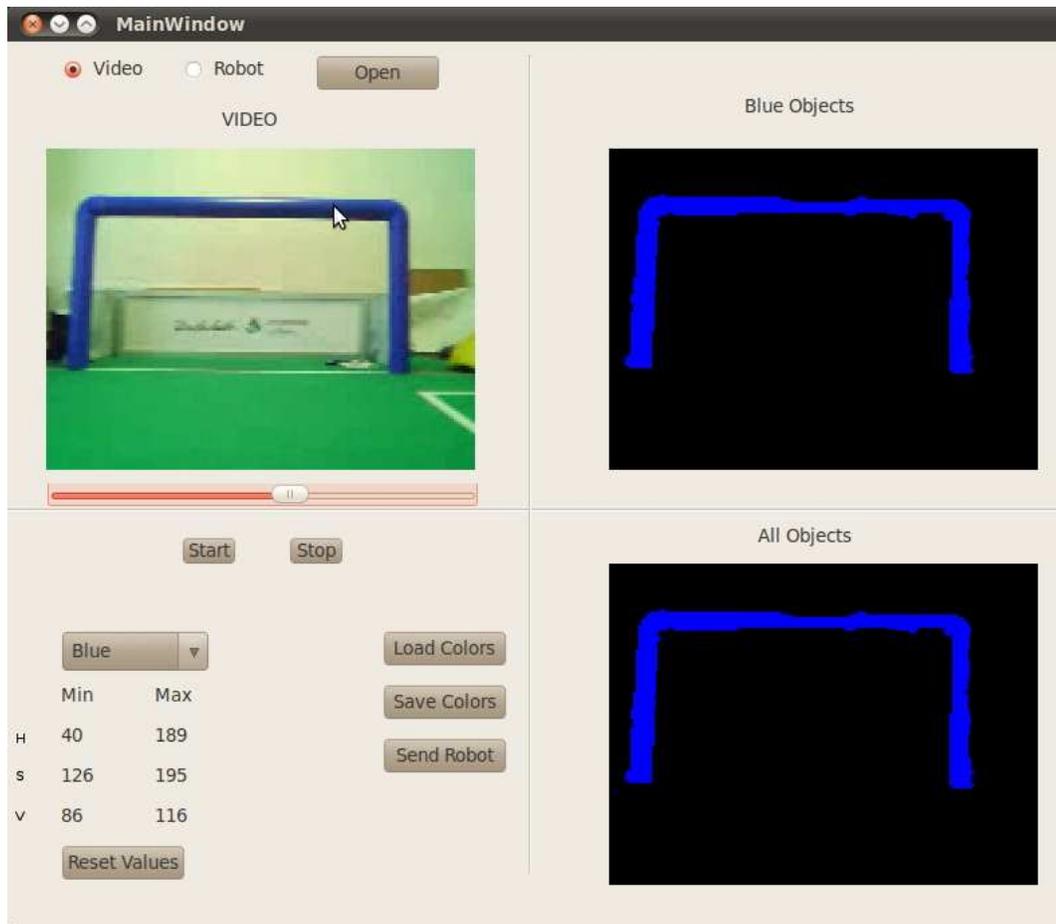


Figura 3.3: Interfaz gráfica del programa segmentador

El programa consta de dos funcionalidades principales, de las cuales se explicará una en esta sección. En el recuadro donde se ve la imagen de color completa, con el título de Video, se observa el video abierto correspondiente a la cancha, y cuya ventana posee una interfaz que espera por eventos del mouse. Al momento de hacer click sobre la imagen, se obtiene el valor del píxel seleccionado, obteniendo su valor de los 3 ejes del espacio de color (por ejemplo, la coordenada 230, 79, 80). Dependiendo de cual sea el color previamente seleccionado, con el algoritmo se van obteniendo los valores máximos

y mínimos con respecto a los píxeles que se van seleccionando uno a otro para el respectivo color. En la figura 3.4 se pueden ver 3 regiones de los colores que se necesitan, objetos azules (fragmento de la portería azul, parte del poste de medio campo), amarillos (parte del poste de medio campo y un objeto que se puede considerar ruido en la escena por no pertenecer a ningún objeto de interés) y verde, que es el caso del suelo de la cancha, además de los valores máximos y mínimos de YCrCb para el amarillo.

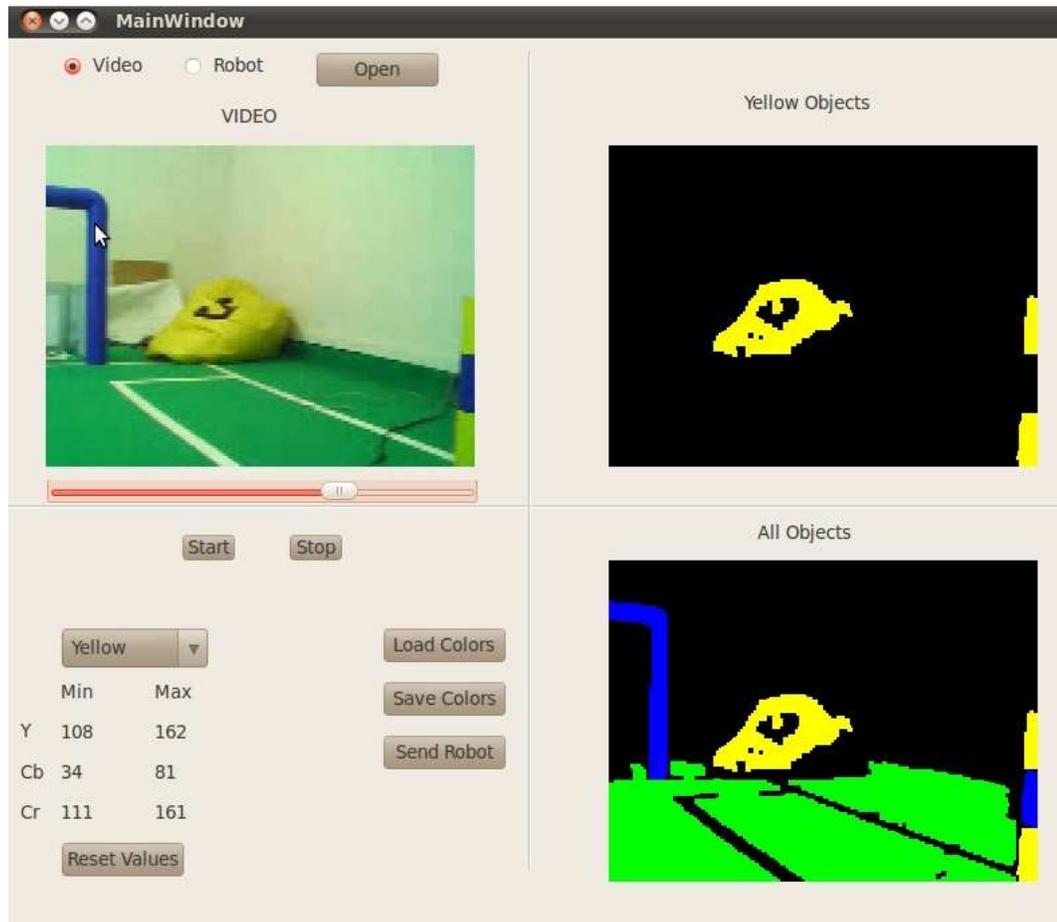


Figura 3.4: Interfaz gráfica del programa segmentador mostrando las zonas verdes, amarillas y azules.

3.3. Experimentación y comparación

El experimento consiste en tomar la misma escena con 3 distintas condiciones de iluminación, realizar la segmentación por medio de las cajas delimitadoras para los colores que se requieren, pero solo en la

escena que se tomará como base de iluminación normal, y esos valores de segmentación se utilizarán sobre la imagen de la misma escena, pero con más luz en una y con menos luz en la otra. Se utilizará la correlación de histogramas como métrica de distancia entre los histogramas de las 3 escenas.

3.3.1. Correlación

La correlación es un coeficiente que mide la relación lineal entre dos distribuciones aleatorias cuantitativas [27]. Esta métrica se utiliza para saber que tan parecido es el histograma de una escena con respecto a la otra, considerando que el solo cambio en la escena es la cantidad de luz. El coeficiente de correlación varía en el rango de $[-1, 1]$, si correlación=1, se tiene una correlación perfecta, entiéndase, la misma escena sin ninguna variación en la iluminación para este caso. Si correlación=0 representa que no existe correlación alguna, y para este caso sería un cambio demasiado grande de iluminación que sería difícil solucionar, además es un caso prácticamente imposible en el ambiente de trabajo. A continuación se muestra la fórmula utilizada para hallar la correlación entre histogramas.

$$distancia(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 (H_2(I) - \bar{H}_2)^2}}, \quad (3.11)$$

donde

$$\bar{H}_k = \frac{1}{N} \sum_J H_k(J), \quad (3.12)$$

y N es el número total de valores del histograma.

3.3.2. Resultados de la comparación

A continuación se mostrará la serie de experimentación de la segmentación en la imagen base y luego aplicando esos umbrales máximos y mínimos a las imágenes con más y menos luz. En la figura 3.5 se mostrará el resultado de aplicar la segmentación en cada uno de los espacios de color antes mencionados y se tomará esta escena como imagen base.

Se puede ver que la separabilidad de los colores es mucho mejor en el espacio HSV y YCbCr. Se tomaron otras dos escenas, una con menos y otra con más luz. Estos experimentos con el color amarillo se muestran en la figura 3.6 y la figura 3.7.

En la tabla 3.1 se puede ver la comparación de la correlación entre los histogramas de las imágenes base, menos y más luz y la diferencia en porcentaje de la cantidad de píxeles que se clasificaron con respecto a la imagen base segmentada en su espacio de color correspondiente (se está comparando RGB contra RGB, HSV contra HSV y YCbCr contra YCbCr, donde lo que varía es la cantidad de luz). Para el color amarillo, se puede ver que el HSV es el espacio de color que sufre menos variación, siendo los resultados muy cercanos a solo 5% de variación para menos y más luz. Siguiéndole se encuentra el

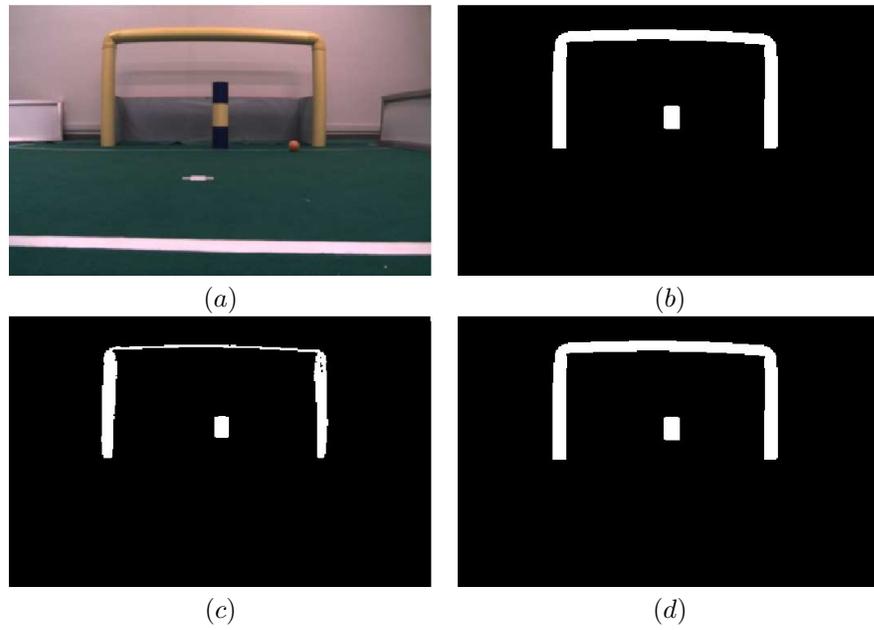


Figura 3.5: Amarillo. Escena base: a) Color sin segmentación, b) Segmentación HSV, c) Segmentación RGB, d) Segmentación YCbCr

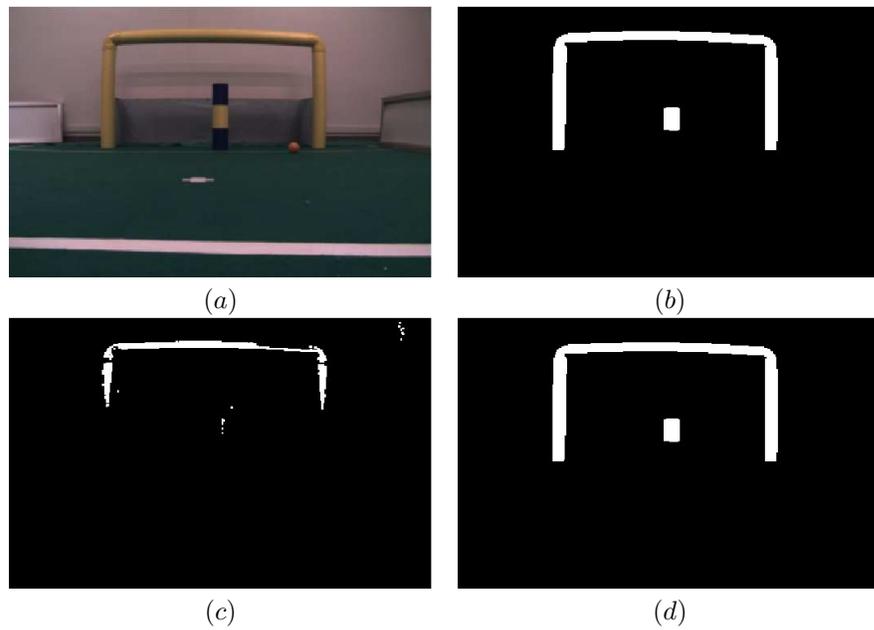


Figura 3.6: Amarillo. Escena menos luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr

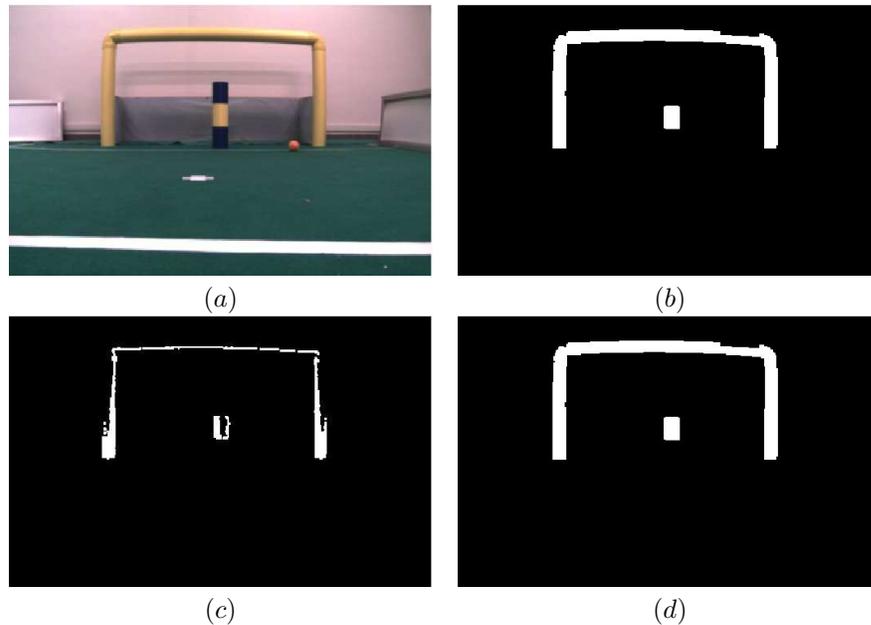


Figura 3.7: Amarillo. Escena más luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr

espacio YCbCr, con variaciones cercanas al 13% y por último el espacio RGB, mostrando variaciones muy grandes cercanas al 45%. Con el resultado de solo este color, se podría dar por inservible al espacio de color RGB para esta tarea, pero de todas maneras se continuará la experimentación con otros dos colores fundamentales en la cancha, como son el azul y el naranja.

Tabla 3.1: Comparación de la correlación y la segmentación del color amarillo en los distintos espacios de color en la escena

Imágenes	Correlación	Dif. en segmentación HSV	Dif. en segmentación RGB	Dif. en segmentación YCbCr
Base y menos luz	0.034	7.587 %	40.713 %	14.867 %
Base y más luz	0.134	4.334 %	53.756 %	12.087 %

En la tabla 3.2 se pueden ver los valores obtenidos para el color azul, una vez más se observa que el HSV es el espacio de color que sufre menos variación, siendo los resultados entre el 5% y 20% de variación para menos y más luz.

Siguiéndole se encuentra el espacio YCbCr, con variaciones entre el 13% y 22% por último el espacio RGB, mostrando variaciones muy grandes entre el 25% y 45%.

En la tabla 3.3 los resultados muestran que en el espacio de color RGB, no se consiguió segmentar con más ni menos luz con los umbrales previamente encontrados, y en contrapartida, el espacio de

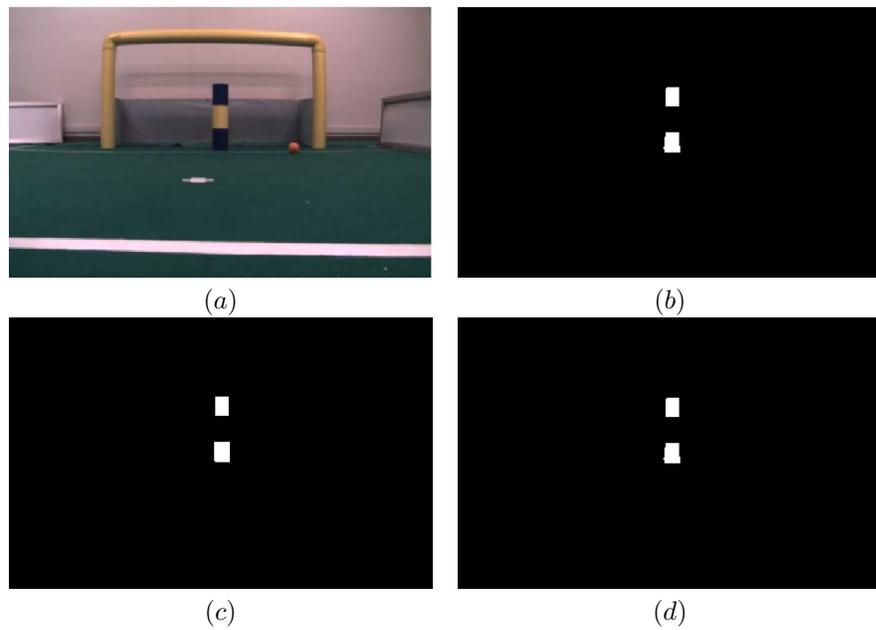


Figura 3.8: Azul. Escena base: a) Color sin segmentación, b) Segmentación HSV, c) Segmentación RGB, d) Segmentación YCbCr

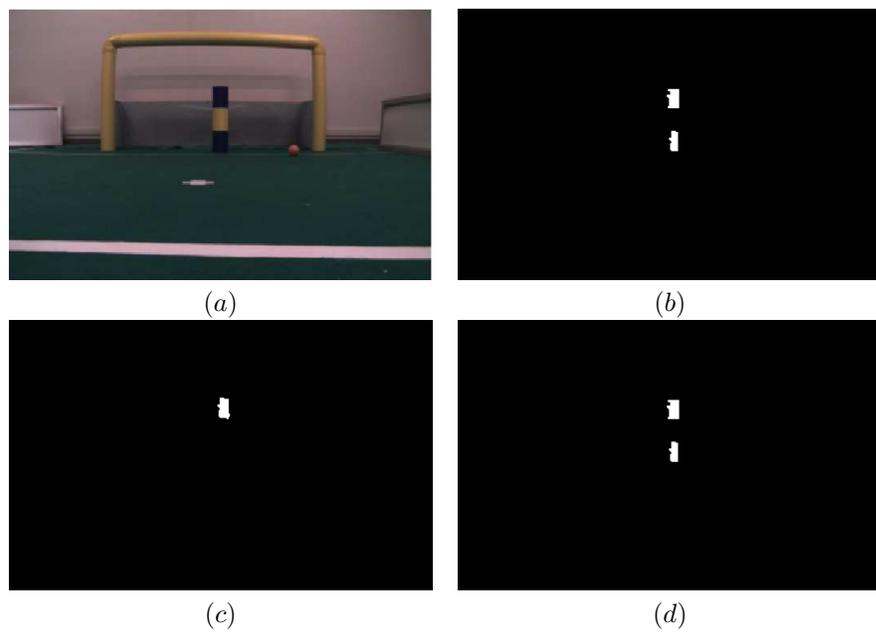


Figura 3.9: Azul. Escena menos luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr

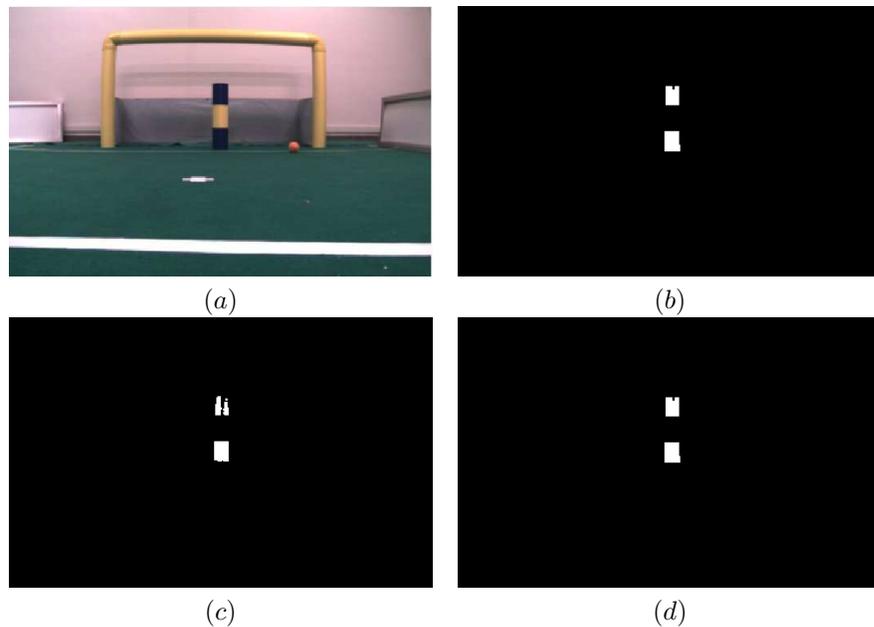


Figura 3.10: Azul. Escena más luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr

Tabla 3.2: Comparación de la correlación y la segmentación del color azul en los distintos espacios de color en la escena

Imágenes	Correlación	Dif. en segmentación HSV	Dif. en segmentación RGB	Dif. en segmentación YCbCr
Base y menos luz	0.034	20.560 %	25.833 %	22.657 %
Base y más luz	0.134	4.672 %	49.1666 %	13.077 %

Tabla 3.3: Comparación de la correlación y la segmentación del color naranja en los distintos espacios de color en la escena

Imágenes	Correlación	Dif. en segmentación HSV	Dif. en segmentación RGB	Dif. en segmentación YCbCr
Base y menos luz	0.034	0 %	100 %	15.876 %
Base y más luz	0.134	0 %	100 %	12.072 %

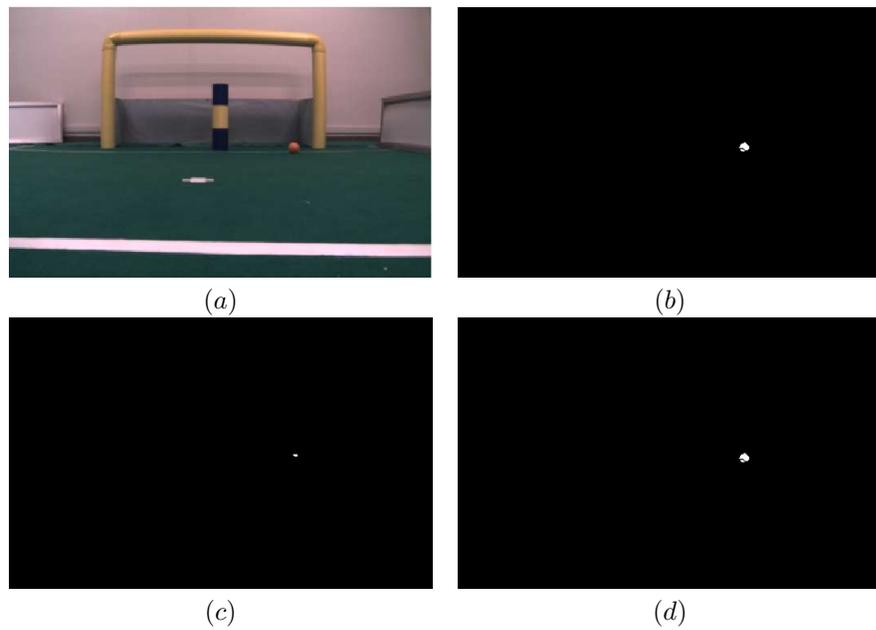


Figura 3.11: Naranja. Escena base: a) Color sin segmentación, b) Segmentación HSV, c) Segmentación RGB, d) Segmentación YCbCr

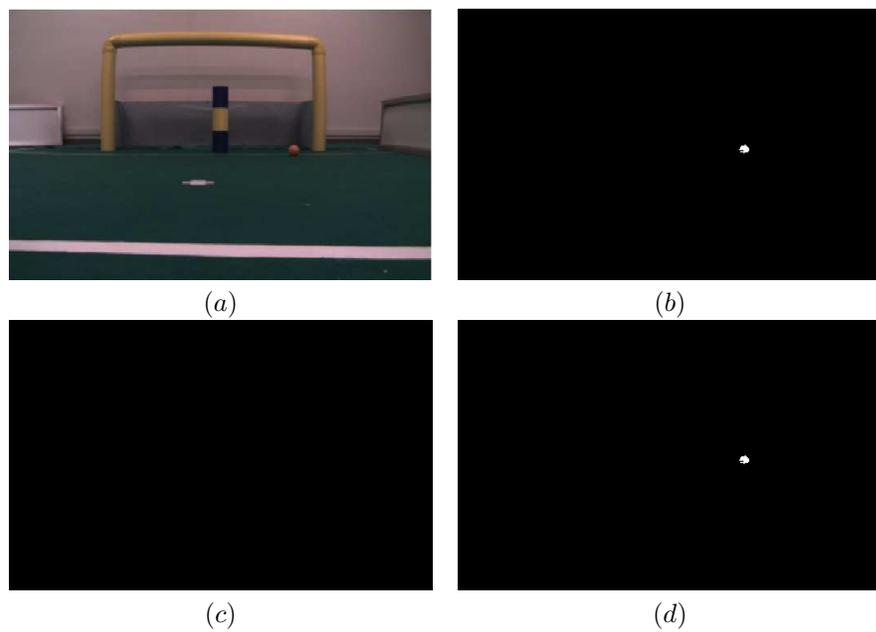


Figura 3.12: Naranja. Escena menos luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr

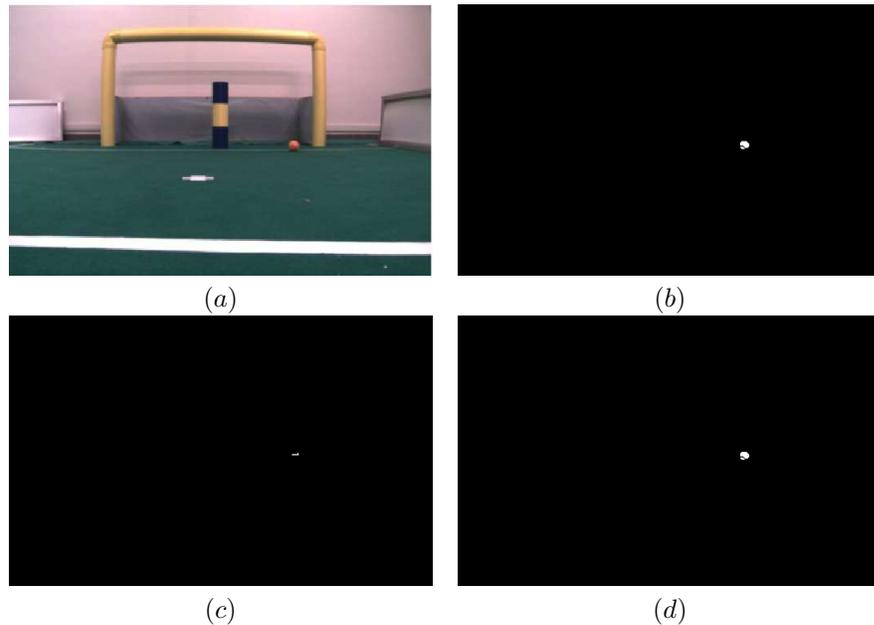


Figura 3.13: Naranja. Escena más luz: (a) Color sin segmentación, (b) Segmentación HSV, (c) Segmentación RGB, (d) Segmentación YCbCr

color HSV mostró una invarianza ante el cambio de iluminación. El espacio de color YCbCr presentó una variación de entre el 12 % y 15 %.

3.3.3. Elección

Dado los resultados presentados, el espacio de color HSV mostró ser lo suficientemente robusto ante cambios de iluminación, quedando en segundo lugar el espacio de color YCbCr. Este último espacio de color es muy utilizado dentro de la RoboCup en algunas categorías debido principalmente a que la gran mayoría de cámaras web de bajo costo entrega la imagen adquirida por el driver en este espacio de color y además, como se ha visto, presenta una variación que por lo general no sobrepasa el 25 %. Para el caso experimental de este trabajo, como la cámara uEye entrega en formato RGB, el costo de convertir cada píxel al valor HSV no es alto en relación a la ganancia de tener un sistema que sea consistente ante posibles cambios ligeros de luz.

Capítulo 4

Adaptación del sistema de visión a cambios en la iluminación

4.1. Introducción

En este capítulo se examinará el problema de la iluminación no controlada en el ambiente de trabajo de la competencia. Años anteriores, la RoboCup garantizaba al menos una iluminación de 500 lux en la cancha de la categoría “KidSize”. Desde el año 2009, esta garantía por parte de los organizadores de la competencia fue removida, haciendo que el efecto de la iluminación no controlada fuera un nuevo reto, además de que se volvió más fácil para los organizadores el preparar el ambiente de trabajo al no tener que lidiar con mediciones de la luz. En la práctica, el nivel de iluminación para las diferentes competencias dependiendo de la distribución del centro de convenciones, puede variar entre 400 y 1500 lux, pudiendo tener variaciones de hasta 500 lux dependiendo de la posición de la cancha dentro de dicho centro. Factores externos pudieran también causar variaciones en la iluminación hasta en un 400 lux durante el día de la competencia [33]. A continuación, se presentarán algunas soluciones proporcionadas para otras categorías, y se verán las limitaciones de *hardware* para poder hacer uso de dichos algoritmos, así como de la metodología desarrollada para hacer que el sistema sea robusto ante los cambios en la intensidad de la iluminación.

4.1.1. Limitante Computacional

Este trabajo fue diseñado para trabajar en sistemas con un poder computacional muy limitado. En este caso, una computadora embebida dentro del robot humanoide de tan solo 45 cm de altura, ejecutando las rutinas para la locomoción de forma bípeda, la inteligencia y toma de decisiones, planeo de ruta y el sistema de visión. Debido a esto, es realmente difícil hacer que todos los módulos se ejecuten de manera correcta al mismo tiempo. Gracias a la metodología desarrollada en este capítulo, es perfectamente extendible a otro ambiente de trabajo donde se cuente con un sistema embebido limitado como

este.

4.1.2. Trabajos previos

En trabajos previos para la RoboCup, se han desarrollado varias técnicas de transformaciones de pre-procesamiento para atacar el problema del cambio en la iluminación [34]. Todos estos trabajos consideran pasos post adquisición. Sería mucho mejor tener una imagen mejorada desde la adquisición en sí misma, antes de considerar un paso después de la adquisición. A continuación se explica algunos de los métodos de pre-procesamiento después de la adquisición.

4.1.2.1. Constancia de color

El fundamento de esta técnica es similar a la que se propone en este trabajo, sin embargo, para poder diferenciarlo de la propuesta, se describe en qué consisten los trabajos en esta área. Constancia de color se refiere a la habilidad de realizar una corrección de color de manera que se pueda adquirir la misma información de una escena con diferentes intensidades en la iluminación.

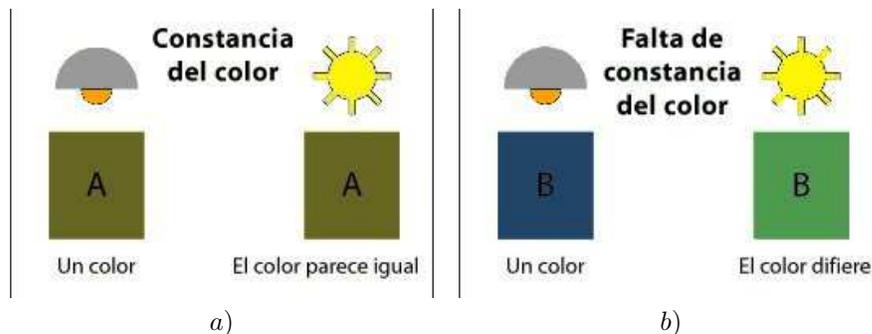


Figura 4.1: a) Esquema de constancia de color, b) Esquema donde no existe constancia de color

Esta corrección cromática puede ser realizada en base a transformaciones sobre los valores de las imágenes haciendo una estimación de la iluminación [34]. Varios métodos han estado usando la estimación de la iluminación en una imagen de entrada de manera a realizar transformaciones de los colores en aquellos previamente conocidos basándose en una iluminación conocida. Algunos de estos métodos son llamados *Métodos estáticos*, que tienen bases estadísticas. Métodos estadísticos simples asumen que el color de la fuente de iluminación puede ser estimado calculando el promedio de colores en la imagen [35]. Tan et. al muestran en [36], un método físico-estadístico la utilización de objetos en la escena empelando información de sus interacciones para poder estimar la fuente de iluminación. Forsyth muestra en [37] un algoritmo de estimación de la iluminación en base a gamas de color. Estos métodos referentes a las gamas de color están basados en la suposición de que hay un número limitado de colores que pueden ser observados en una imagen del mundo real para una cierta iluminación

dada. La idea principal de estos métodos es estimar el tipo de iluminación como una entrada desconocida dado una gama precalculada. Otros enfoques están basados en técnicas de aprendizaje máquina. Como se considera en [34], enfoques de este estilo han utilizado redes neuronales y vectores de soporte para regresión para entregar una constancia de color precisa basado en datos de entrenamiento. Combinando la salida de múltiples métodos, Gijssen y Gevers [38] muestran que seleccionando los algoritmos apropiados dependiendo de la tarea, se puede obtener una constancia de color con una precisión elevada.

4.1.2.2. Adaptación del espacio de color

La aproximación de esta técnica no se centra en transformaciones sobre los píxeles, sino que actúa directamente sobre el modelo de color o sobre el clasificador de colores. Se describirán los métodos encontrados que están más relacionados a un desempeño de tiempo real en el ambiente de la RoboCup pero no necesariamente de la categoría “KidSize” donde la limitante de *hardware* es realmente restrictiva. El primer enfoque presentado en [39] es muy simple, se define un color de base y un cubo de delimitación (similar al presentado en este trabajo en el espacio HSV) en el espacio de color YUV para clasificar el color de base (verde para este trabajo). Otros colores se definen en relación con este cubo. Por ejemplo, el magenta se define como teniendo un valor de V mayor que el límite superior del cubo de referencia y un valor de U menor que el límite inferior del cubo de referencia. Con los cambios de luz, las posiciones de todos los colores se mueven a lo largo de los tres ejes y las intensidades se amplifican o debilitan. Sin embargo, la posición relativa de los colores con respecto a los demás permanece constante. Por lo tanto, sabiendo como cambia el cubo de referencia con el cambio de luz, esto es suficiente para cambiar de posición y las dimensiones de los demás cubos de color en base a los cambios realizados sobre el de referencia.

Otro enfoque interesante es presentado por Anzani et al. en [40], quienes utilizan un método conocido para el modelado de color, basándose en el algoritmo de Expectativa-Maximización (EM). Este algoritmo es un método para encontrar la estimación de máxima verosimilitud o máxima a posteriori (MAP) de los parámetros en los modelos estadísticos, donde el modelo depende de variables latentes no observables. EM es un método iterativo que alterna entre la ejecución de un paso de expectativa (E), y un paso de maximización de (M), que calcula las estimaciones de máxima verosimilitud basándose en las distribuciones de probabilidad generadas en el paso E. Estas estimaciones de parámetros se utilizan para determinar la distribución de las variables latentes en el próximo paso. Este algoritmo lo utilizan para trasladar y rotar las coordenadas del espacio de color HSV conforme a los cambios de iluminación. Adaptaron el algoritmo para que tenga la posibilidad de detener la adaptación en tiempo real cuando fuera necesario. Esta técnica es utilizada en el citado trabajo de Anzani et al. en robots de la categoría “MiddleSize”, donde no son robots humanoides y el espacio físico les proporciona la capacidad de tener entre 3 y 4 computadoras para el proceso de sus algoritmos. Hablar sobre procesamiento de tiempo real es muy diferente al comparar el poder de 4 laptops con una computadora de sistemas embebidos.

Otro ejemplo se presenta en [41], cuyo trabajo no está relacionado con la RoboCup, por lo cual no

tienen una preocupación por la complejidad del algoritmo desarrollado, por lo tanto no posee un análisis de desempeño en cuanto a tiempo. Los autores introducen no una adaptación de un modelo existente, sino un espacio de color nuevo en el que la métrica asociada se aproxima a distancias percibidas y a los desplazamientos del color, captando las relaciones espectrales que son robustas a los cambios en la iluminación.

4.1.3. Motivación

Todos los métodos mencionados hasta el momento, son post-proceso de adquisición y debido a las limitaciones de *hardware* que ya se presentó, no se puede hacer que funcionen correctamente en combinación con todas las tareas actuales en el interior del robot. Debido a esta gran limitante, surgió la idea de tratar de ajustar el parámetro de la cámara *tiempo de exposición* para obtener las imágenes como se necesitan, en vez de hacer un programa de pre-procesamiento de las imágenes. Esta calibración se propone mediante un sistema de lógica difusa con muy buenos resultados los cuales se presentarán en este capítulo. Se tiene que hacer notar que una vez que se alcance el valor del parámetro de calibración con el método propuesto, no hay necesidad de pre-procesamiento de las imágenes para realizar una corrección de color o adaptar el espacio de color que se está usando, así que no hay una pérdida de tiempo de consumo del procesador a causa de un proceso necesario de pre-procesamiento previo a la tarea principal de la visión en el robot.

4.2. Calibración activa

La calibración activa consiste en ajustar los parámetros de la cámara que, para este caso, se lleva a cabo a través de la variación del tiempo de exposición. Esto permitirá obtener la imagen de acuerdo a las necesidades, en comparación con el trabajo previo descrito en la sección 4.1.2, en el que los ajustes de imagen se realizaron después de su adquisición. Dado que el método presentado consiste en el ajuste de los parámetros de cámara comparando los cambios en la iluminación, es necesario tener una segmentación óptima de color a una cierta condición luz para realizar los ajustes en base a los valores de segmentación en respuesta a los cambios de iluminación. En el experimento, se determinó que la correlación necesaria para cumplir las necesidades de este trabajo es de más de 0,6 en el rango de $[-1, 1]$ y el porcentaje de píxeles clasificados correctamente, debe ser al menos el 95 % para que sea considerado satisfactorio el ajuste. Como este método es dependiente del *hardware*, en el caso de la cámara uEye IDS-1226LE-C, se presentará primeramente un análisis de la misma para poder mostrar los tipos de ajustes y rangos que se pueden variar.

4.2.1. Cámara IDS uEye

La UI-1226LE es una cámara ultra-compacta de una platina con un moderno sensor CMOS de Aptina y una resolución máxima Wide VGA (752x480 píxeles). El uso de la tecnología USB 2.0 permite realizar

conexiones con los sistemas más variados sin ningún tipo de problemas. El modelo está adaptado para un uso profesional fuera del sector industrial. Posee una interfaz de programación de aplicaciones (API) muy extensa pero solo se tomarán en cuenta aquellas funciones que están relacionadas con poder variar el tiempo de exposición en la manera que se requiere. Para poder hacer esto, se tienen que analizar 3 conceptos:

- **Pixel clock:** es el parámetro básico de la cámara para controlar el tiempo de captura. Determina la velocidad de reloj a la cual se puede leer desde las celdas el sensor CMOS. Este modelo puede variar dicho parámetro desde 5 MHz hasta 40 MHz. A mayor frecuencia, la imagen es más oscura y permite la mayor cantidad de cuadros por segundo. Cambios en la frecuencia del *pixel clock*, afectan los valores rango de *frame rate* y *exposure time*.
- **Frame rate:** es el rango posible de valores para la velocidad de cuadros, y depende de la frecuencia de píxel seleccionado. Puede seleccionar una velocidad inferior sin cambiar la frecuencia de píxel. Para establecer una mayor velocidad de cuadro, sin embargo, es necesario aumentar la frecuencia de píxel. Se mide en cuadros por segundo, *frames per second* (fps).
- **Exposure time:** es el tiempo de exposición, que depende de la velocidad de cuadros seleccionada. Puede seleccionar un tiempo de exposición más corto, sin cambiar la velocidad de cuadros, pero para establecer un tiempo de exposición mayor, es necesario reducir la velocidad de cuadros. Se mide en milisegundos.

A continuación, en la tabla 4.1 de relaciones entre el *pixel clock* y *frame rate* (fps).

Tabla 4.1: Relaciones

Pixel clock	Rangos frame rate
5 MHz	0.18 - 19.91 fps
10 MHz	0.36 - 39.82 fps
15 MHz	0.54 - 59.73 fps
20 MHz	0.72 - 79.64 fps
25 MHz	0.90 - 99.55 fps
30 MHz	1.08 - 119.46 fps
35 MHz	1.26 - 139.36 fps
40 MHz	1.44 - 159.27 fps

Al momento de comparar las imágenes obtenidas a estos diferentes *pixel clock*, se descartó la utilización de una velocidad muy alta, debido a que algunas imágenes no las capta de manera correcta, o muy baja, debido a que la cantidad de luz que deja entrar es excesiva, por lo cual ahora se hallará la relación entre dos valores de *pixel clock*, dos valores de fps, para hallar el rango de exposure que se genera.

Como se puede observar en la tabla 4.2, los valores resultantes con 15 y 20 Mhz son casi iguales, por lo cual se seleccionará el valor de 20 MHz a 30 fps

Tabla 4.2: Relaciones

Pixel clock	Frame rate	Rango exposure
15 MHz	25 fps	0.21 - 39.93 ms
15 MHz	30 fps	0.21 - 33.24 ms
20 MHz	25 fps	0.16 - 39.95 ms
20 MHz	30 fps	0.16 - 33.24 ms

4.2.2. Modelado del tiempo de exposición

Una vez obtenido el valor de *pixel clock* y *frame rate* que se mantendrá fijo, el modelo de tiempo de exposición se basa en la segmentación de una imagen original (la imagen base) con un tiempo de exposición determinado, que fue de 30 ms. Se ha hecho la calibración pasiva (segmentación) de la imagen de base para clasificar al objeto de interés. Después de eso, se varía el tiempo de exposición en el rango de validez general para obtener la correlación entre los histogramas de brillo y calcular los píxeles clasificados correctamente entre la imagen base y la nueva imagen adquirida con el nuevo tiempo de exposición. La figura 4.2 muestra la relación entre el tiempo de exposición y el coeficiente de correlación (a) y la relación entre la exposición y los píxeles clasificados correctamente (b).

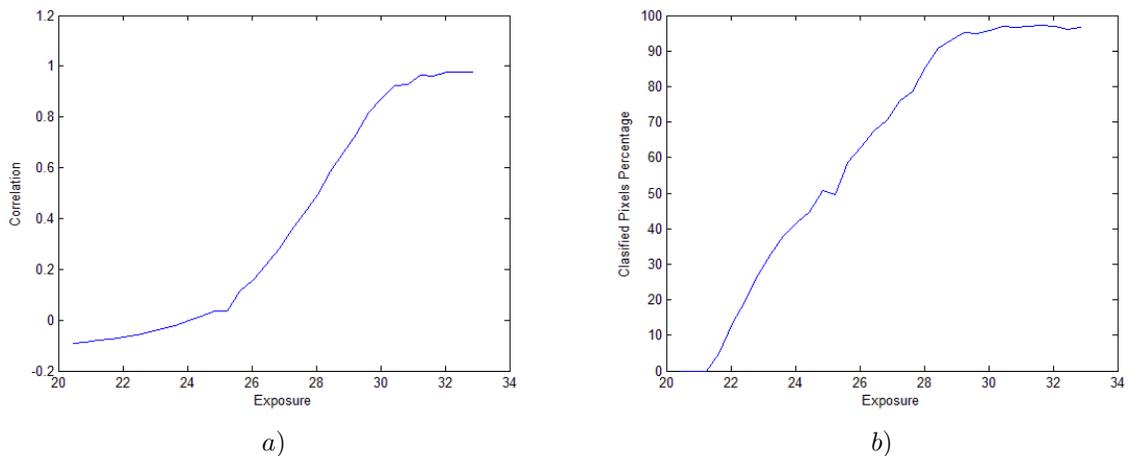


Figura 4.2: Comportamiento del tiempo de exposición: a) Con respecto a la correlación, b) Con respecto a los píxeles correctamente clasificados

De la figura 4.2 ahora se puede ver que a partir de 25 fps, se tiene una correlación positiva, y 50 % de clasificación de píxeles.

4.2.3. Sistema Fuzzy

El sistema Fuzzy será el encargado de proporcionar la variación del tiempo de exposición que se busca para obtener la imagen con la cantidad de luz que se necesita para que no se afecten los colores para la tarea. Para el desarrollo de la programación en esta sección, se utilizó la librería para C++ llamada **Free Fuzzy Logic Library** (FFLL), librería basada en **Fuzzy Control Language** (FCL), que es un estándar para la programación de controladores fuzzy publicado por la **International Electrotechnical Commission** (IEC). Las especificaciones de la sintaxis de la FCL puede ser encontrada en el documento IEC 61131-7. Las funciones de membresía se definen en base al comportamiento que se observó en el modelado del tiempo de exposición y luego algunos ajustes de dichas funciones para hacerlo más eficiente a la hora de hacer el cálculo de la salida. Se definieron dos variables lingüísticas de entrada de tipo triangulares:

- correlación: El rango en el que se hará válida la correlación es [0, 1]

```

FUZZIFY correlacion
TERM bajo := (0, 0) (0.2, 1) (0.4, 0)
TERM medio := (0.3, 0) (0.8, 1) (1, 0)
TERM alto := (0.75, 0) (1, 1) (1, 1)
END_FUZZIFY

```

La figura 4.3 muestra los datos de la función **correlacion** en su representación triangular.

- píxeles clasificados: El rango válido para el porcentaje es [0, 100]

```

FUZZIFY pixeles-clasificados
TERM bajo := (0, 0) (50, 1) (80, 0)
TERM medio := (60, 0) (80, 1) (100, 0)
TERM alto := (80, 0) (100, 1) (100, 1)
END_FUZZIFY

```

La figura 4.4 muestra los datos de la función **pixeles-clasificados** en su representación triangular. Se define la variable lingüística de salida de tipo triangular, que es la variación del tiempo de exposición:

- Variación del tiempo de exposición: el rango válido para la variación del tiempo de exposición está dado por la resta del valor mínimo y máximo de la elección en la sección 4.2.2 (33.24-25 fps), por lo tanto, el rango para la salida es [0, 8.24]

```

FUZZIFY var-exposicion
TERM bajo := (0, 0) (0.2, 1) (0.4, 0)
TERM b_medio := (0.3, 0) (0.8, 1) (1, 0)
TERM a_medio := (0.75, 0) (1, 1) (1, 1)
TERM alto := (0.3, 0) (0.8, 1) (1, 0)

```

```

TERM muy_alto := (0.75, 0) (1, 1) (1, 1)
END_FUZZIFY

```

La figura 4.5 muestra los datos de la función **var-exposicion** en su representación triangular. Una consideración adicional necesaria para definir la base de conocimiento para el sistema, es el signo que se asigna a la correlación. El símbolo de “+” representa que al analizar la imagen captada, ésta posee una iluminación más intensa con respecto a la imagen base, mientras que el signo “-” representa que la imagen captada posee una iluminación menos intensa que la imagen base. Este mismo signo, determina si la variación al tiempo de exposición (variable de salida) tiene que ser positiva (aumentar el tiempo de exposición en la cantidad entregada por la variable de salida) o negativa (disminuir el tiempo de exposición en la cantidad entregada por la variable de salida).

Tabla 4.3: Base de conocimiento del sistema Fuzzy

		Correlación					
		bajo +	bajo -	medio +	medio -	alto +	alto -
Píxeles clasificados	bajo	muy alto	muy alto	alto	alto	-	-
	medio	a_medio	a_medio	b_medio	b_medio	b_medio	b_medio
	alto	b_medio	b_medio	b_medio	b_medio	bajo	bajo

Inferencia

La inferencia es el proceso mediante el cual se genera un mapeo para asignar a una entrada una salida utilizando lógica difusa. El proceso de inferencia provee las bases para la toma de decisiones del sistema. Este proceso involucra la utilización de funciones de pertenencia y las reglas generadas en la base de conocimiento. Se utilizará el método de Mamdani. El método Mamdani utiliza reglas tipo si-entonces (if-else). Una regla de la base de reglas o base de conocimiento tiene dos partes, el antecedente y la conclusión. En un sistema fuzzy tipo Mamdani tanto el antecedente como el consecuente de las reglas están dados por expresiones lingüísticas, que en este caso ya se definieron en formato FCL.

Ahora se definen las reglas en formato FCL a partir de las relaciones mostradas en la tabla de base de conocimiento 4.3 de la correlación y la cantidad de píxeles clasificados.

```

RULE 0: IF pixeles-clasificados.bajo AND correlacion.bajo_pos THEN var-exposicion.muy_alto;
RULE 1: IF pixeles-clasificados.bajo AND correlacion.bajo_neg THEN var-exposicion.muy_alto;
RULE 2: IF pixeles-clasificados.bajo AND correlacion.medio_pos THEN var-exposicion.alto ;
RULE 3: IF pixeles-clasificados.bajo AND correlacion.medio_neg THEN var-exposicion.alto;
RULE 4: IF pixeles-clasificados.medio AND correlacion.bajo_pos THEN var-exposicion.a_medio;
RULE 5: IF pixeles-clasificados.medio AND correlacion.bajo_neg THEN var-exposicion.a_medio;

```

RULE 6: IF pixeles-clasificados.medio AND correlacion.medio_pos THEN var-exposicion.b_medio;
 RULE 7: IF pixeles-clasificados.medio AND correlacion.medio_neg THEN var-exposicion.b_medio;
 RULE 8: IF pixeles-clasificados.medio AND correlacion.alto_pos THEN var-exposicion.b_medio;
 RULE 9: IF pixeles-clasificados.medio AND correlacion.alto_neg THEN var-exposicion.b_medio;
 RULE 10: IF pixeles-clasificados.alto AND correlacion.bajo_pos THEN var-exposicion.b_medio;
 RULE 11: IF pixeles-clasificados.alto AND correlacion.bajo_neg THEN var-exposicion.b_medio;
 RULE 12: IF pixeles-clasificados.alto AND correlacion.medio_pos THEN var-exposicion.b_medio;
 RULE 13: IF pixeles-clasificados.alto AND correlacion.medio_neg THEN var-exposicion.b_medio;
 RULE 14: IF pixeles-clasificados.alto AND correlacion.alto_pos THEN var-exposicion.bajo;
 RULE 15: IF pixeles-clasificados.alto AND correlacion.alto_neg THEN var-exposicion.bajo;

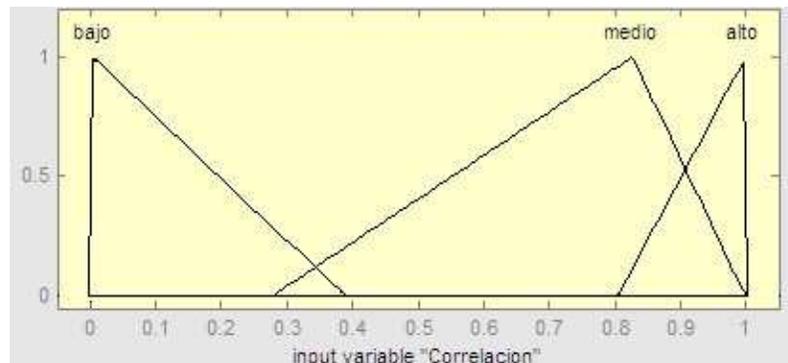


Figura 4.3: Correlación entre histogramas

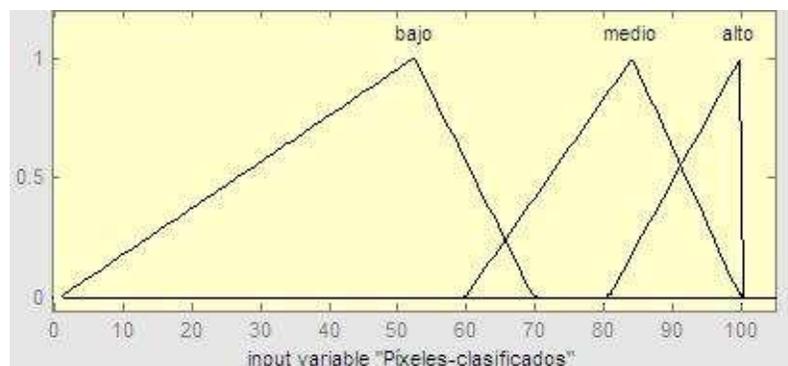


Figura 4.4: Píxeles clasificados

Defuzificación

Después de la inferencia, se tendrá una conclusión difusa, una variable lingüística cuyos valores han sido asignados por grados de pertenencia, sin embargo se necesita un escalar que corresponda a estos

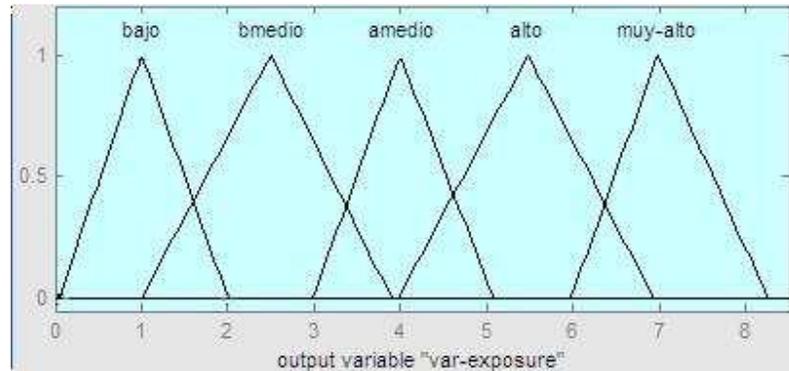


Figura 4.5: Variación del *exposure time*

grados de pertenencia. En la defusificación se utilizan métodos matemáticos simples como el método del centroide, método promedio máximo, método del promedio ponderado y método de membresía del medio del máximo. En este caso, se utilizará el **método de centroide** o “**CoG**” como se designa en el modelo FCL (el más utilizado por los ingenieros de control) donde se calcula el centro de gravedad del polígono que se generó en la inferencia.

4.3. Algoritmo general de calibración

Cabe resaltar, que esta metodología se ejecuta fuera del tiempo de competencia real. La calibración pasiva se lleva a cabo solo una vez durante toda la competencia, seleccionando los umbrales para cada color, mientras que la calibración activa, se realiza momentos antes de empezar el juego. Esto debido a que una vez calibrada la cámara, la iluminación no se afecta considerablemente en el lapso de 10 minutos que dura cada tiempo (dos tiempos por cada partido) de juego de fútbol. Una vez terminado cada tiempo, la calibración activa se puede realizar para hallar el parámetro de la cámara necesario para ese momento.

Los pasos para la calibración pasiva se presentan a continuación:

1. Una imagen es capturada con la cámara en una determinada escena. Se definirá esta escena como la *escena-A* la cual será usada como la imagen base (imagen-A) para el algoritmo.
2. Se realiza la segmentación, con la ayuda de segmentador GUI presentado en la sección 3.2.2, a la *imagen-A* para hallar los umbrales en los ejes H,S,V para las cajas delimitadoras de color, las cuales serán usadas en el robot durante el juego.

Una vez realizada la calibración pasiva, los pasos a seguir para la calibración activa se detallan a continuación:

1. Una imagen es capturada con la cámara en la *escena-A*.

2. Se calcula la correlación de la imagen nueva y la *imagen-A* en base a sus histogramas de brillo.
3. La segmentación de color se realiza de manera automática en la nueva imagen con los umbrales en H,S,V previamente hallados en la *imagen-A*.
4. Se realiza una comparación en la cantidad de píxeles clasificados en la nueva imagen con respecto a los encontrados en la *imagen-A*.
5. Si el resultado de la comparación es al menos 95 %, el algoritmo termina sin necesidad de ningún ajuste.
6. Si no, el sistema Fuzzy empieza teniendo 3 entradas: la distancia de correlación, el número de píxeles clasificados y la dirección del cambio en la iluminación (más oscura o más clara), entregando como salida la variación que se tiene que aplicar al tiempo de exposición (*exposure time*).
7. Se actualiza la cámara con el nuevo tiempo de exposición y el algoritmo continúa al paso 1.

La figura 4.6 muestra el algoritmo usado para realizar la calibración activa.

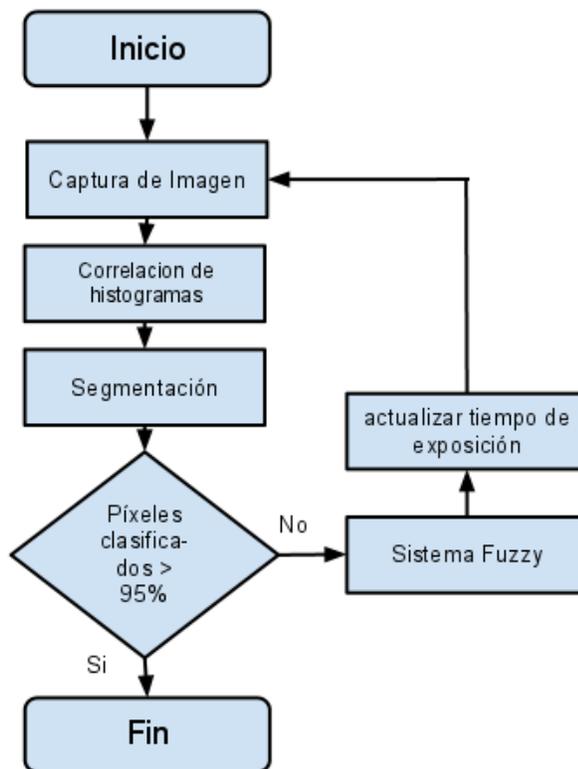


Figura 4.6: Diagrama de la metodología de calibración activa.

4.4. Experimentos

Para probar la metodología propuesta, se diseñaron dos tipos de experimentos que se explicarán a continuación

4.4.1. Experimento 1

En este primer experimento, se quiso probar qué tanta variación se puede tener en el ambiente a ser ajustado. En la figura 4.7 se muestran las 6 imágenes más representativas del cambio de iluminación que se pudo ajustar correctamente. Se puede ver desde la primera imagen a la última, que la variación de iluminación es considerablemente grande, mucho más de lo que se presenta en el ambiente de la RoboCup. Se tomó la escena de la figura 4.7(d), se realizó la calibración pasiva sobre esta imagen, y las demás escenas fueron las probadas una por una. Se inicializó la cámara con un valor de *exposure time* y *gain* (ganancia para simular un ambiente más iluminado) aleatorios, para empezar la calibración. El *exposure time* se calculó para cada una de las diferentes escenas a la respectiva luz del momento. La tabla 4.4 muestra los resultados del sistema de calibración, con el nuevo *exposure time*, y el porcentaje de clasificación de píxeles logrados sobre la imagen calibrada.

Tabla 4.4: Resultados del experimento 1

Figura	Nuevo exposure time (ms)	Píxeles clasificados
Fig. 4.7(a)	32.699	99.693 %
Fig. 4.7(b)	33	99.877 %
Fig. 4.7(c)	30.96	95.088 %
Fig. 4.7(d)	–	–
Fig. 4.7(e)	30.332	97.608 %
Fig. 4.7(f)	25.992	95.994 %

En este experimento, no se tuvo error con el umbral de 95 % de píxeles clasificados.

4.4.2. Experimento 2

Este experimento consistió en tomar la imagen de una escena a cierta condición de luz en el laboratorio, cambiar la iluminación apagando una fuente de luz, prendiéndola, dejando entrar luz por la ventana, luego cerrarla y una combinación de los pasos anteriores. Se detallará con imágenes 3 alteraciones de la iluminación en las escenas y el respectivo ajuste que se realiza. En la figura 4.8 se puede ver la **escena 1**, donde la imagen base para realizar la calibración pasiva es la figura 4.8(a), mientras que la figura 4.8(b) es la imagen de la misma escena pero con una iluminación diferente. En la figura 4.8(c) y 4.8(d) se puede ver la representación de histogramas que se detallarán a continuación. El histograma de brillo se halló a partir las imágenes convertidas a escala de grises, donde cada píxel está representado por 8 bits, entonces, se puede tener 255 posibles valores en la escala en el eje horizontal. El eje vertical está dado por el máximo valor encontrado dentro del rango de los 255 posibles valores de

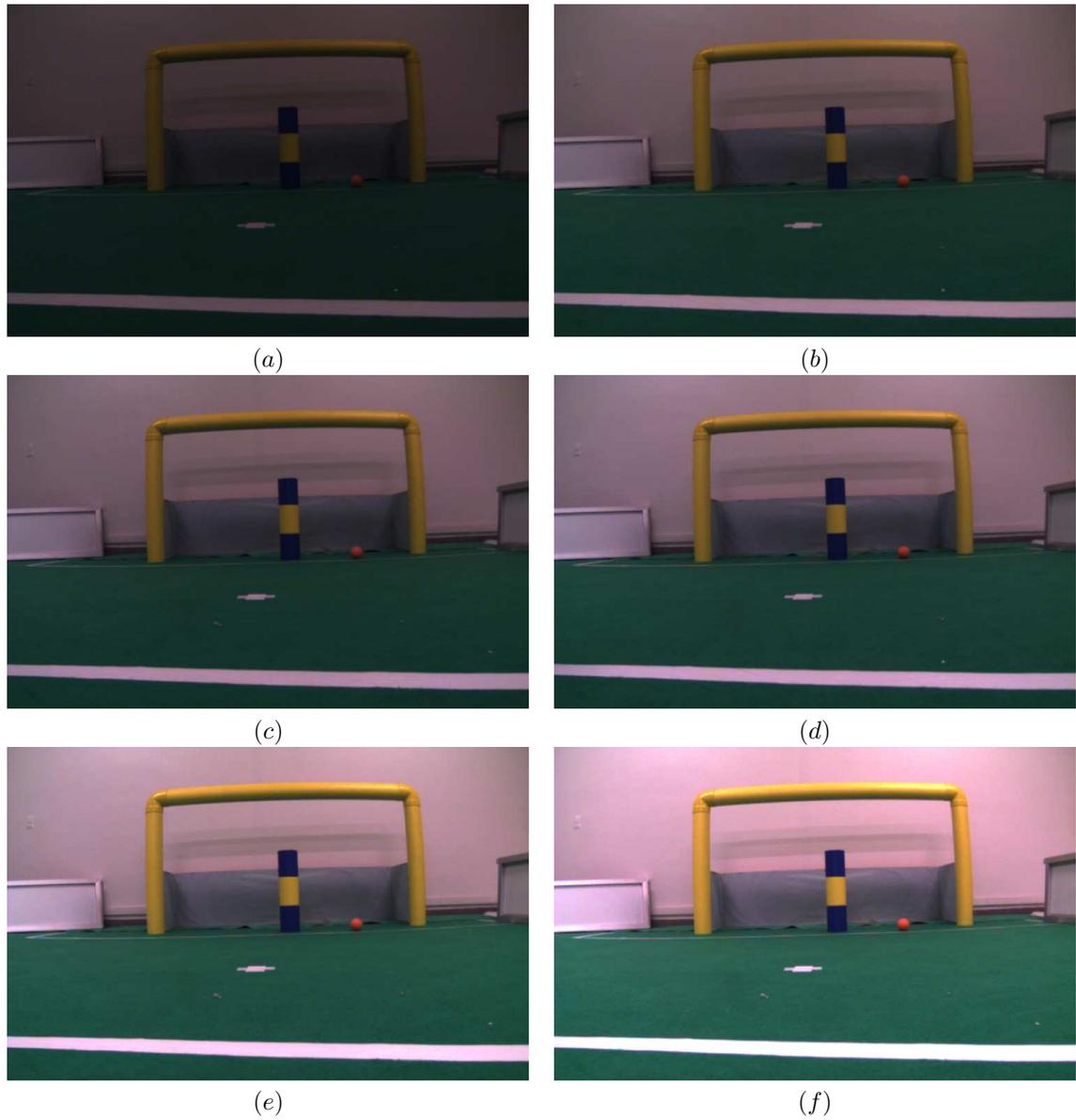


Figura 4.7: Escenas de prueba: desde la cantidad mínima de luz (a), a la cantidad máxima de luz (f)

brillo.

En la figura 4.8(c) se muestra en barras verdes, el histograma de brillo de la imagen base y en barras azules, el histograma de brillo de la nueva imagen adquirida con el cambio de iluminación. La intersección de estos dos histogramas se representa en cian. Después del ajuste, ambos histogramas deberían de estar casi o totalmente traslapados. Seguido de esto, empieza el algoritmo de calibración activa, calculando la correlación entre los histogramas y el porcentaje de píxeles clasificados. Cuando el ajuste proporciona un exposure time que logra una clasificación superior al 95 %, se da por calibrada la cámara. En la figura 4.8(d) se muestra los histogramas de la imagen base y la última imagen donde se logró clasificar correctamente los píxeles. Como se puede ver, ambos histogramas están traslapados en un gran porcentaje (barras cian). Con esto se puede demostrar, que se está adquiriendo la imagen como se necesita, por ende, no se necesitará realizar una nueva segmentación de color para los ejes H,S,V de las cajas delimitadoras.

La **escena 2** se muestra en la figura 4.9 y la **escena 3** en la figura 4.10. Se aplicó la misma metodología que en la **escena 1** para la experimentación.

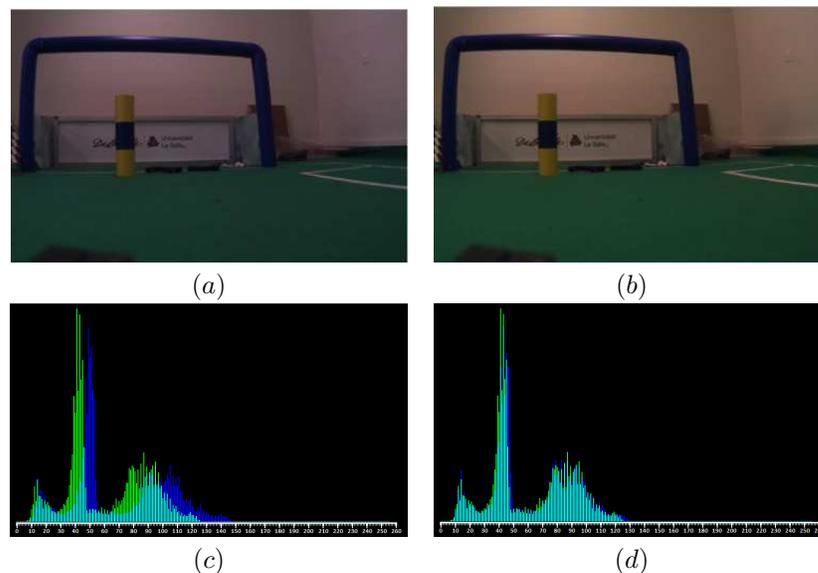


Figura 4.8: Escena 1: Segmentación del azul, (a) Imagen base, (b) Condiciones alteradas de iluminación, (c) Histogramas antes del ajuste, (d) Histogramas después del ajuste del exposure time

La tabla 4.5 muestra los resultados de la experimentación en 10 escenas distintas del experimento 2, donde las 3 primeras escenas son las que fueron detalladas con anterioridad. Se muestra las correlaciones de los histogramas de brillo antes y después del ajuste, así como la cantidad de píxeles clasificados superior al 95 %.

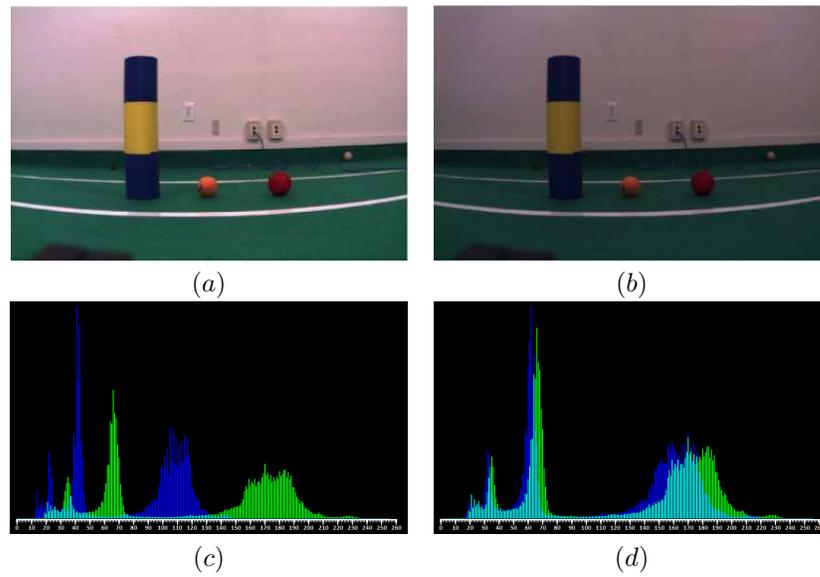


Figura 4.9: Escena 2: Segmentación del naranja, (a) Imagen base, (b) Condiciones alteradas de iluminación, (c) Histogramas antes del ajuste, (d) Histogramas después del ajuste del exposure time

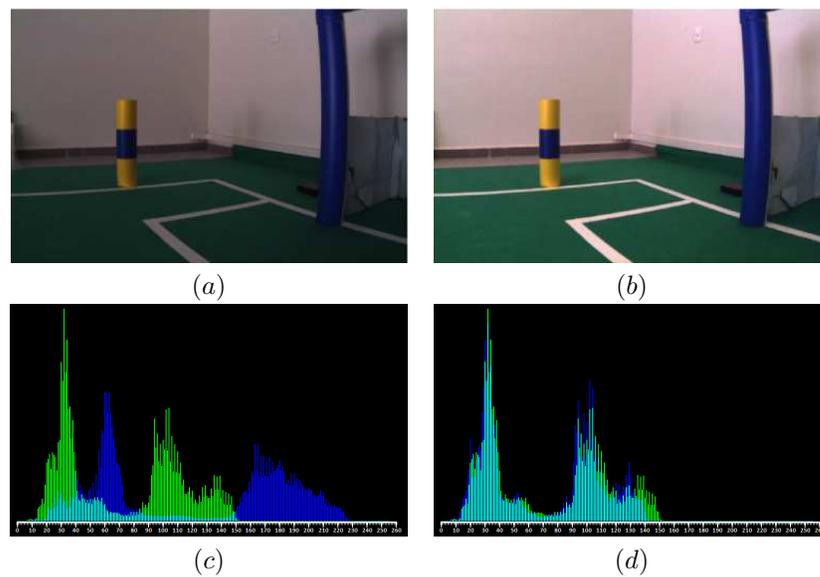


Figura 4.10: Escena 3: Segmentación amarillo, (a) Imagen base, (b) Condiciones alteradas de iluminación, (c) Histogramas antes del ajuste, (d) Histogramas después del ajuste del exposure time

Tabla 4.5: Resultados del experimento 2

Escena	Correlación inicial	Correlación final	Píxeles clasificados
1	0.330	0.809	97.22
2	0.03	0.624	98.31
3	0.01	0.951	99.30
4	0.370	0.778	96.52
5	0.103	0.871	98.12
6	0.45	0.991	99.80
7	0.01	0.859	97.82
8	0.231	0.824	98.31
9	0.41	0.781	95.98
10	-0.11	0.863	95.68

4.5. Discusión

La metodología presentada constituye un aporte importante para trabajos donde la iluminación es variable de forma similar a la presentada previamente. En ningún momento se está demeritando los trabajos previos presentados donde los ajustes son post-adquisición, sino que se tiene la certeza de que una combinación del método presentado con los demás analizados, podría ser una solución óptima a problemas más complejos de iluminación. Se hace énfasis en que este método se consideró óptimo para este trabajo, debido a que no se consumió recursos del robot una vez que éste haya empezado su trabajo. La elección correcta del espacio de color HSV para esta tarea y el *hardware* de la cámara, ayuda a realizar el ajuste de una manera efectiva, rápida y no tan compleja.

Capítulo 5

Metodología del Procesamiento de Imágenes

Una vez que ya se tiene un sistema robusto a cambios en la iluminación del ambiente en la competencia, se está en condición de realizar el proceso de detección y clasificación de objetos en la escena, así como la extracción de los puntos que se consideran de interés. En este capítulo, se explicará toda la metodología del procesamiento de imágenes necesario para el proceso de detección y clasificación de objetos. Se puede ver el esquema general de la metodología que se aplicó para dejar las imágenes listas para ser reconocidas en la figura 5.1. Cabe resaltar, que los algoritmos desarrollados para las siguientes secciones de este trabajo de tesis, son independientes de la plataforma de adquisición. Los algoritmos fueron probados en la cámara uEye y en la Play Station Camera, así como en otras dos cámaras web comerciales (Logitech y Minoru).

5.1. Tratamiento de los píxeles

La primera etapa, tiene como entrada la imagen adquirida desde la cámara, que se encuentra en formato RGB, con 8 bits para representar cada uno de los tres canales. El primer paso, fue encontrar una implementación adecuada para convertir a la imagen en formato HSV, que es el primer paso y se detalla a continuación.

5.1.1. Conversión a HSV

La conversión a HSV es un proceso costoso en cuanto a recursos y tiempo, porque se está realizando sobre cada imagen que se adquiere desde la cámara, y en las primeras pruebas, se pudo ver que el robot no era capaz de caminar correctamente porque el hilo de la visión consumía mucho tiempo de proceso y no dejaba tiempo de procesamiento al algoritmo que controla el proceso de locomoción, así que se tuvo que ver qué alternativa existía para usar este espacio de color. Se probaron varias

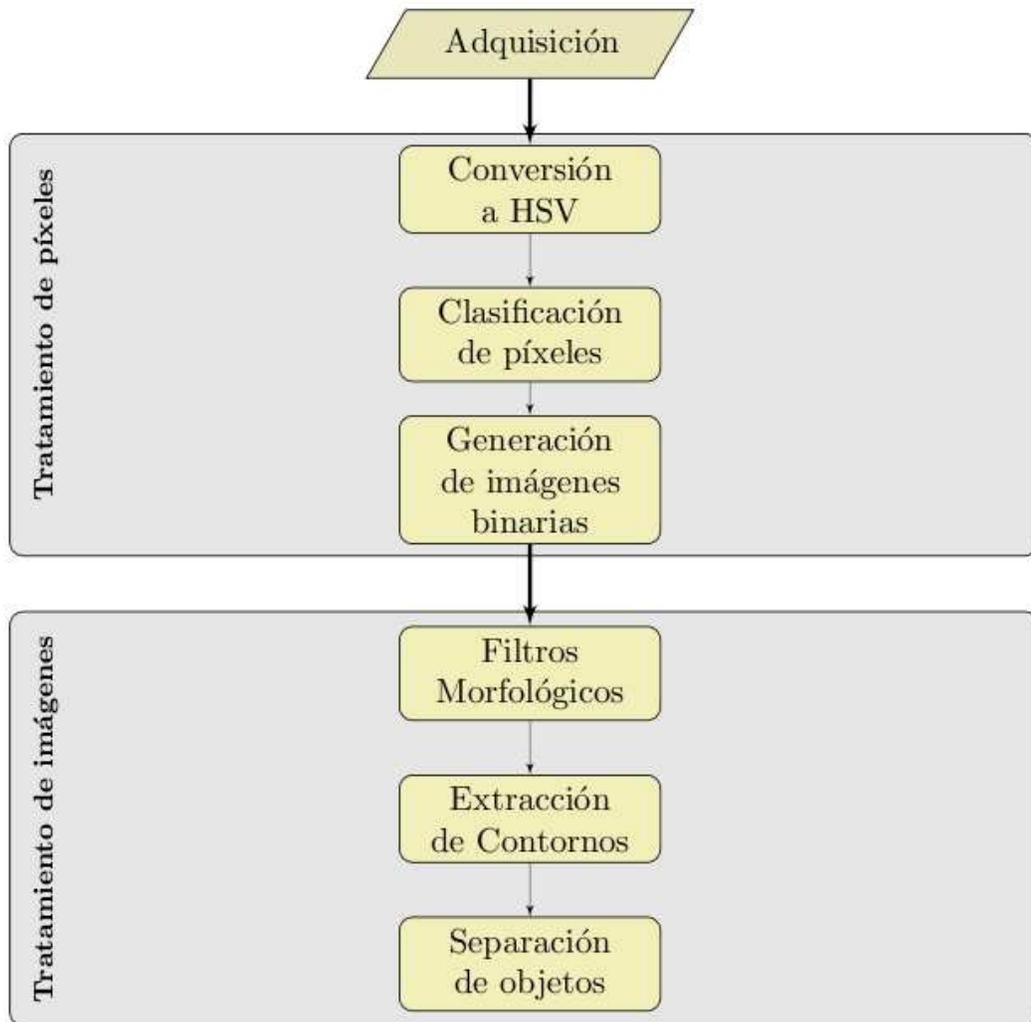


Figura 5.1: Metodología general del procesamiento de imágenes.

implementaciones hasta que se encontró la razón por la cual no se daba con el problema: todas las implementaciones utilizan operaciones de punto flotante, siendo esto lento en una arquitectura i486, además que se hacen varias operaciones sobre cada píxel para encontrar sus valores en HSV. La solución fue utilizar una implementación sin la utilización de la unidad de punto flotante del procesador, y esto se logró utilizando variables enteras sin signo (`uint8_t` en el language C/C++). Se implementaron las ecuaciones 3.3, 3.2 y 3.1, dando un buen tiempo de respuesta. La figura 5.2 muestra la imagen original y la imagen con sus píxeles convertidos al espacio de color HSV.

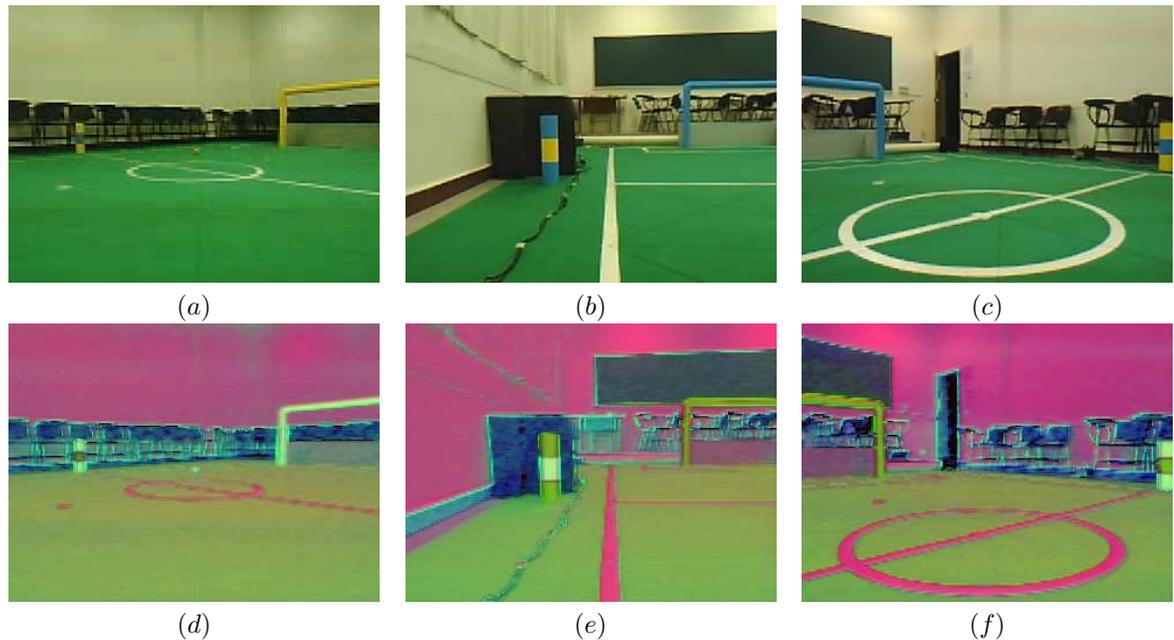


Figura 5.2: (a), (b) y (c) Imágenes en formato RGB. (d), (e) y (f) Respectivas conversiones al espacio de color HSV pero desplegado en RGB

5.1.2. Clasificación de píxeles y generación de imágenes binarias

La metodología para clasificar cada píxel fue detallada en la sección 3.2. Para ejemplificar y demostrar el proceso, se hallaron los valores mínimos y máximos en los ejes H,S,V para clasificar cada píxel que esté en estos umbrales. Los valores se puede ver en la tabla 5.1

Tabla 5.1: Valores de segmentación hallados con el programa segmentador GUI para el conjunto de imágenes

Eje	amarillo min	amarillo max	azul min	azul max	verde min	verde max
H	92	138	25	57	71	92
S	138	255	75	207	142	190
V	115	255	74	186	93	255

El resultado de clasificar los píxeles con los valores de la tabla son las imágenes mostradas en la figura 5.3

Poseer la imagen binaria de cada color, es necesario para ciertos tipos de filtros, particularmente los filtros morfológicos y para los algoritmos de separación de objetos por contornos. Debido a ésto, se separó cada imagen convertida a HSV en una imagen binaria por color. Se pueden ver las imágenes binarias para cada imagen del conjunto en la figura 5.5. Nótese que las imágenes binarias fueron

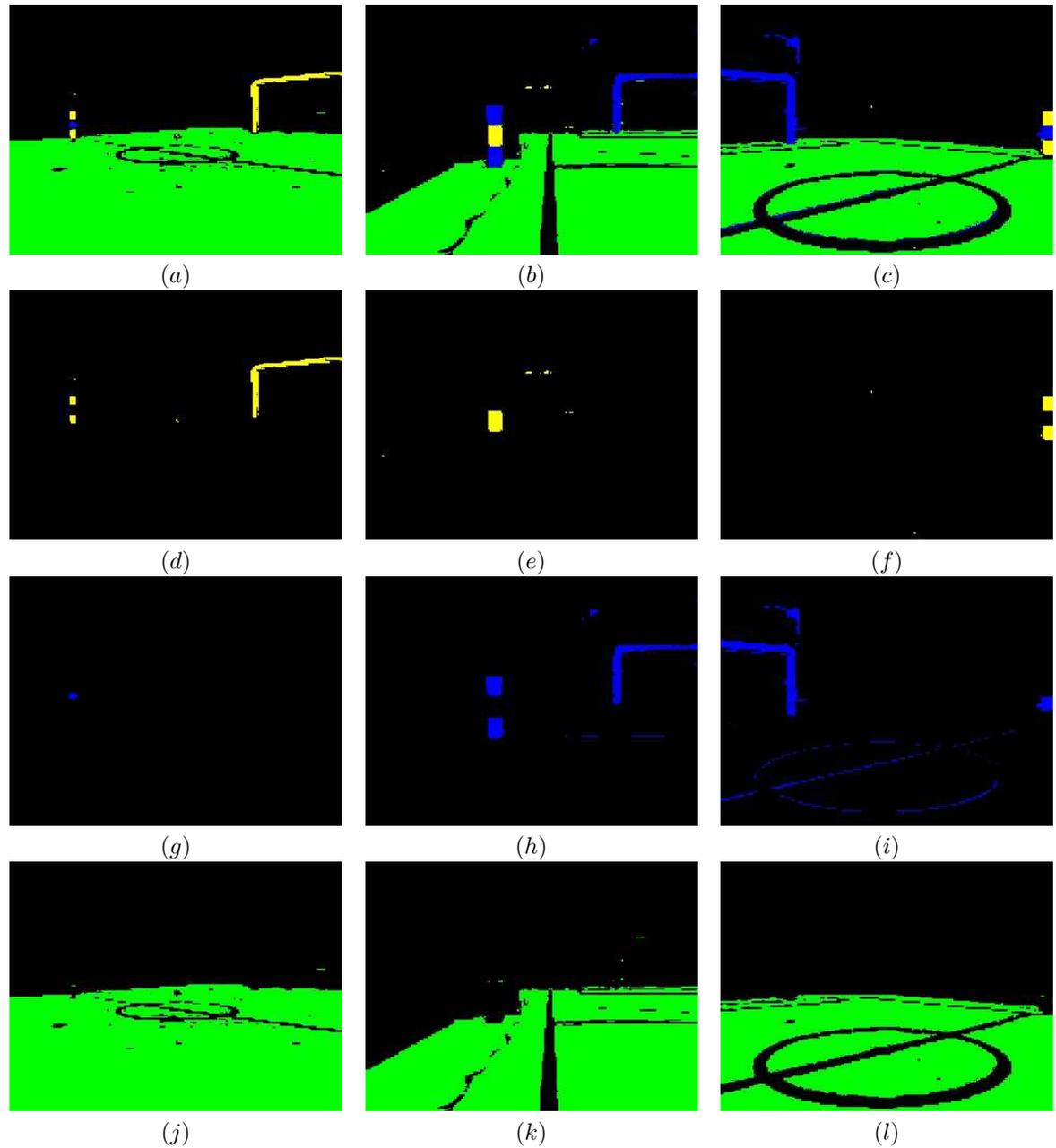


Figura 5.3: (a), (b) y (c) Imágenes segmentadas con los valores de amarillo, azul y verde. Todos los demás píxeles no clasificados se pusieron en negro. (d), (e) y (f) Imágenes binarias del color amarillo. (g), (h) e (i) Imágenes binarias del color azul. (j) (k) (l) Imágenes binarias del color verde.

coloreadas con fines ilustrativos.

5.2. Tratamiento de las imágenes binarias

Una vez que ya se obtuvieron las imágenes binarias correspondientes a los colores que se necesitan, se procede al tratamiento de cada una de éstas, primero para filtrar ruido inherente del algoritmo de interpolación de la cámara y ruido pequeño del ambiente que se pudiera confundir con objetos del mismo color que son de interés. El primer paso, se realizó con los bien conocidos filtros morfológicos.

5.2.1. Filtros morfológicos

La descripción básica de la Morfología Matemática descansa en la "teoría de conjuntos" cuyos primeros trabajos se deben a Minkowski [42]. Tiene diversas aplicaciones en el procesamiento de imágenes y representa un ejemplo del procesamiento no lineal. El ámbito y alcance de las transformaciones morfológicas es tan amplio como el propio procesamiento de imágenes. Se pueden encontrar aplicaciones tales como la segmentación [43], percepción remota [44], detección de bordes, aumento de contraste, análisis de texturas [45], etc. Las transformaciones morfológicas tienen como objetivo principal la extracción de estructuras geométricas en los conjuntos sobre los que se opera, mediante la utilización de otro conjunto de forma conocida denominado elemento estructurante. El tamaño y forma de este elemento se escoge de antemano de acuerdo a la morfología del conjunto sobre el que se va a interaccionar y de acuerdo a las formas que se desean obtener. En la figura 5.4 se presenta ejemplos de estructuras.

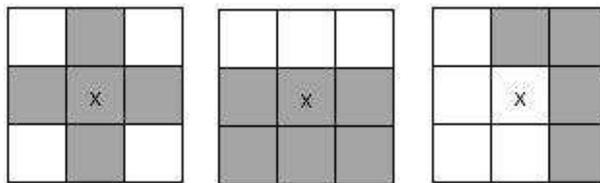


Figura 5.4: Diferentes formas de elementos estructurantes

5.2.2. Operaciones: dilatación y erosión

La dilatación se basa en aumentar el nivel de los valores de los píxeles en el entorno de los objetos presentes en la imagen. La erosión se basa en reducir el nivel de los píxeles del entorno de un objeto. El número de píxeles a los que se aumenta o reduce el nivel depende del tamaño y forma del elemento estructurante usado para procesar la imagen. La dilatación y la erosión expande y contrae la imagen. La dilatación de X por B se define como el conjunto de píxeles X tales que Bx (elemento estructurante desplazado al punto x) intersecta a X [46]

$$\delta_B = X \oplus B \equiv \{x | Bx \cap X \neq 0\} \quad (5.1)$$

La erosión [46] de X por Bx se define como el conjunto de píxeles X tales que Bx está contenido en X de la siguiente manera:

$$\epsilon_B = X \ominus B \equiv \{x | Bx \subseteq X\} \quad (5.2)$$

5.2.3. Filtros: apertura y cerradura

Las operaciones de dilatación y erosión se combinan para formar diferentes métodos de procesar la imagen. Por ejemplo, uno de ellos es la apertura de una imagen, que es la realización de una erosión seguida de una dilatación utilizando el mismo elemento estructural en ambas operaciones. Este método se aplica cuando se desea eliminar los pequeños objetos y mantener el tamaño en los grandes (eliminar ruido). La operación de cerradura de una imagen se realiza cuando se aplica la dilatación y posteriormente la erosión (contrario a la apertura). Este proceso se caracteriza por rellenar huecos y conectar objetos que están próximos entre sí. La apertura [46] de X con respecto al elemento estructural B se define como

$$X \circ B \equiv X_B = (X \ominus B) \oplus B \quad (5.3)$$

La cerradura [46] de X con respecto al elemento estructural B se define como

$$X \cdot B \equiv X^B = (X \oplus B) \ominus B \quad (5.4)$$

El principal objetivo de usar estos filtros morfológicos es el de eliminar cualquier ruido inherente de la interpolación de la cámara o el ruido que se genera al segmentar la imagen, clasificando algunos píxeles en los bordes de los objetos que realmente no son parte de él. El filtro de apertura elimina de una manera muy buena estos picos en los objetos, además de eliminar el ruido aislado separado de los objetos. Estos dos filtros fueron probados con distintos tipos de elementos estructurantes para esta tarea, así como de distintos tamaños, y con la debida experimentación, se pudo notar que no convenía un elemento estructurante mayor a un cuadro de 3x3 (9 píxeles) porque eliminaba datos importantes dentro de la cancha, además de que aumenta el tiempo de ejecución del algoritmo. El resultado final de la selección fue el cuadrado de 3x3 píxeles, por ser el elemento estructurante que mejor respetó la forma de los contornos además de ayudar a suavizarlos. Para ver una comparación de las imágenes antes y después de la utilización de filtros morfológicos, se pueden comparar la figura 5.3 con la figura 5.5 las cuales ya pasaron por los filtros de apertura y cerradura.

5.2.4. Extracción de contornos

Una vez que a las imágenes binarias le han sido aplicados los filtros morfológicos y todo ruido de la cámara, debido a la interpolación de colores y objetos pequeños, ha sido eliminado; cada imagen

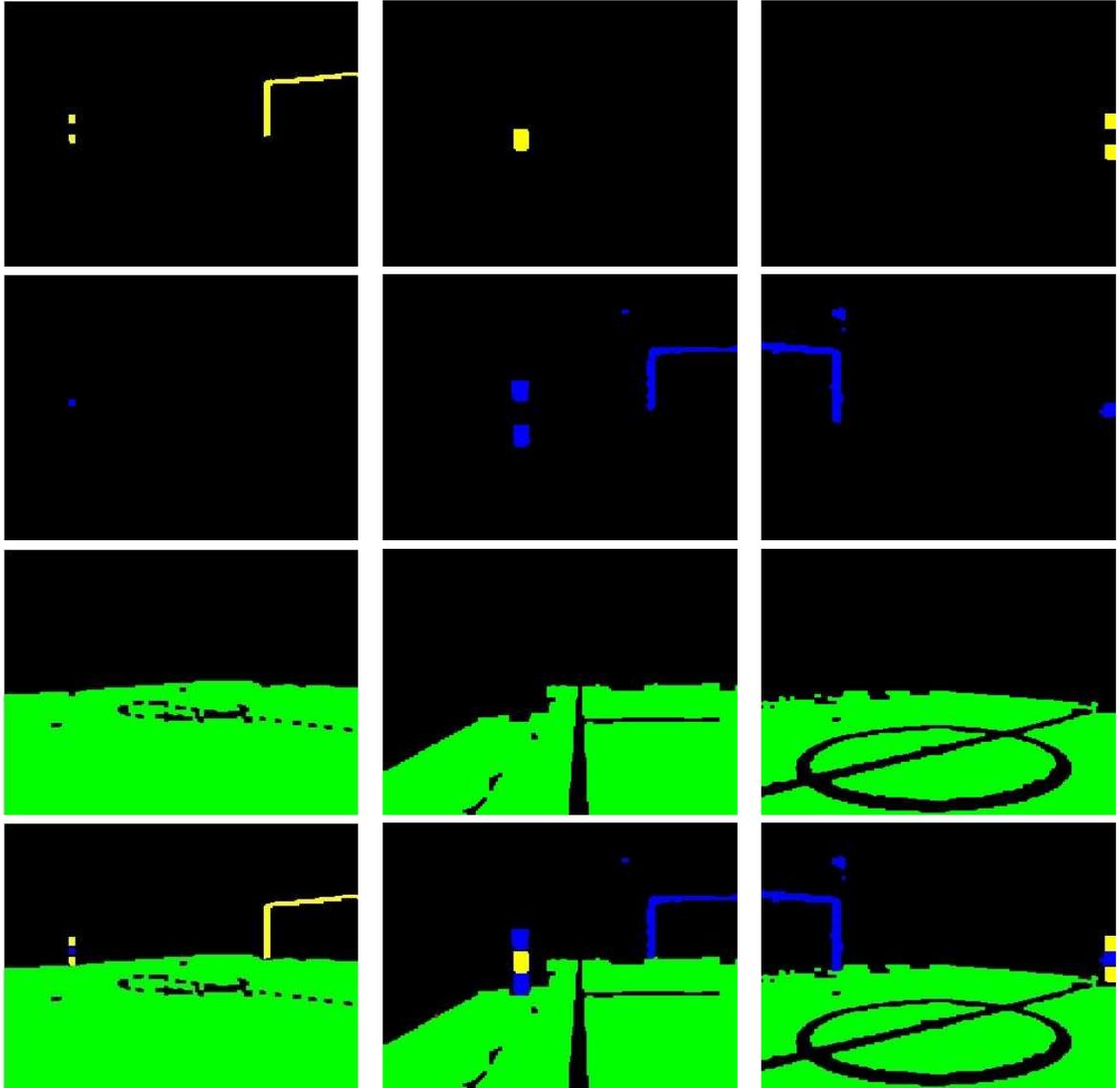


Figura 5.5: Imágenes de la figura 5.3 después de los filtros morfológicos.

binaria respectiva de cada color es procesada con el algoritmo de Suzuki [47].



Figura 5.6: Máscaras de escaneo hacia adelante y hacia atrás respectivamente.

El algoritmo realiza repetidas alternaciones en direcciones hacia atrás y adelante. La imagen binaria $b(x, y)$ consiste en valores de píxeles O y F indicando objeto y fondo respectivamente donde $O > F$. Como en el paso de escaneado simple, se tiene acceso a los píxeles de forma secuencial en el mapa de bits para escanear con la máscara de exploración que se muestra en la figura 5.6.

Al mismo tiempo, también actualiza una tabla de una dimensión llamada tabla de conexión T para memorizar información de equivalencia de las etiquetas. Se realizan operaciones similares locales en el orden de escaneo hacia atrás utilizando la máscara de exploración que se muestra en la figura 5.6. Una característica clave del algoritmo de Suzuki es el uso de la tabla de conexiones de etiqueta para reducir el número de exploraciones hacia adelante y hacia atrás. La asignación de etiquetas provisionales no solo se propaga en los componentes conectados, sino también en la tabla de conexión. En la etapa de escaneo, el algoritmo Suzuki determina la etiqueta provisional de un píxel en la posición (x, y) como sigue:

$$g(x, y) = \begin{cases} F & \text{if } b(x, y) = F, \\ m, (m = m + 1) & \text{if } \forall (i, j) \in M_s, g(x - i, y - j) = F, \\ T_{min}(x, y) & \text{otro caso, donde} \\ & T_{min}(x, y) = \min\{T[g(x - i, y - j)] | (i, j) \in M_s\} \end{cases} \quad (5.5)$$

Donde m es inicializado en 1, $(m=m+1)$ incrementa la etiqueta m y M_s denota la región de la máscara. La tabla de conexión es actualizada, simultáneamente con la asignación de las etiquetas provisionales como sigue:

$$\begin{cases} \text{no-operacion} & \text{if } b(x, y) = F, \\ T[m] = m & \text{if } \forall (i, j) \in M_s, g(x - i, y - j) = F, \\ T[g(x - i, y - j)] = T_{min}(x, y) & (i, j) \in M_s, g(x - i, y - j) = O \end{cases} \quad (5.6)$$

Las operaciones de escaneo hacia atrás son definidas de manera similar con las modificaciones apropiadas a las fórmulas. La figura 5.7 muestra 3 escenas distintas donde se hallaron los contornos, primero con la imagen de la escena sin pasar por los filtros morfológicos y luego después de estos. Una vez que se tienen las listas de los contornos de una manera rápida y eficiente, se procede a separar los objetos, con el algoritmo que se detalla en la siguiente sección.

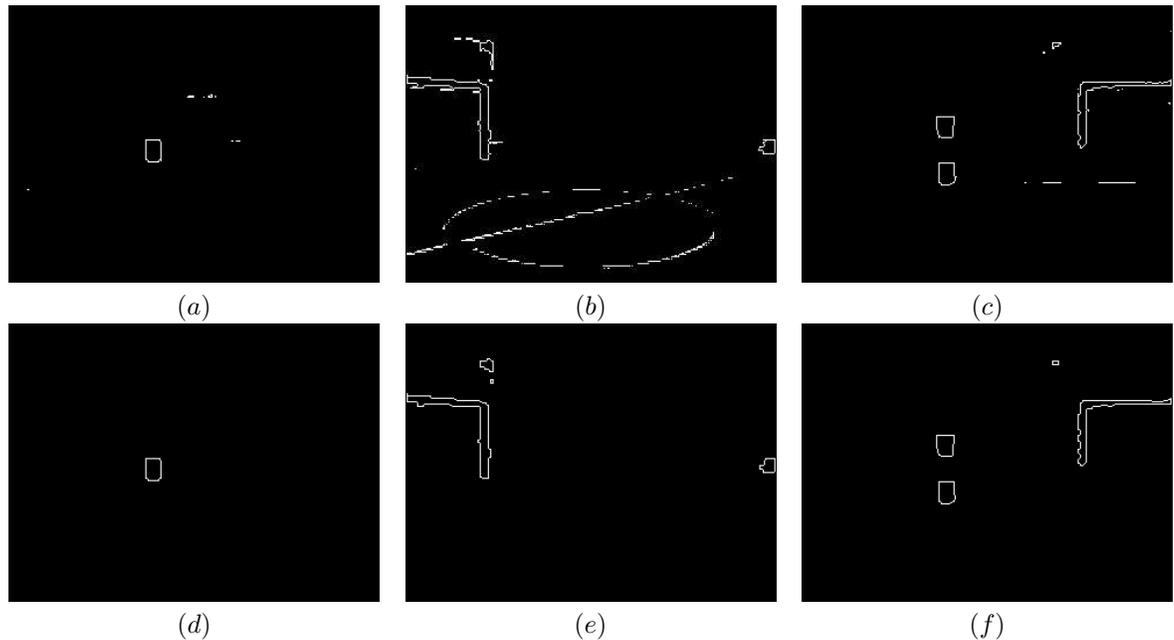


Figura 5.7: (a), (b) y (c) Contornos hallados a las imágenes sin filtrar. (d), (e) y (f) Contornos hallados a las imágenes después de los filtros morfológicos.

5.2.5. Separación de objetos

Hasta el momento, lo que se tiene es una lista con todos los contornos encontrados en las imágenes binarias, y para el proceso de clasificación que se verá posteriormente, lo que se necesita son las coordenadas de la imagen que encierran al objeto en un rectángulo o cuadrado. Aquí se creará una lista en donde se guardará los cuatro vértices con las posiciones x,y respecto al origen de la imagen binaria. Para esto, simplemente se hace un recorrido de las coordenadas de cada contorno, y se va buscando mínimos y máximos en x,y . Al tener x mínima, x máxima, y mínima e y máxima, se está encerrando en un rectángulo al objeto, sin tener en cuenta la complejidad del contorno o cuantas veces éste ha cambiado de dirección (dirección del contorno).

En la figura 5.8 se pueden ver los objetos encerrados en los rectángulos, pero no solo se encuentran objetos de interés: como se puede ver, hay 4 objetos encerrados pero solo 3 son los que se quieren analizar, por lo tanto es necesario realizar un filtro de área mínima. Para esto se necesita saber de cuánto es el área mínima del objeto más pequeño que se quiere reconocer, que en este caso, es una sección del poste de medio campo y a la mayor distancia posible, que es la diagonal que cruza la cancha desde un poste de medio campo hasta la esquina opuesta de la cancha. Esta área encontrada fue de 49 píxeles, entonces, al encontrar el rectángulo que contiene al objeto, solo se agregará a la lista aquellos que tengan un área de al menos 42 píxeles. Estos valores que ahora se tienen en las

listas, representan coordenadas dentro de la imagen. Luego cada una de esas secciones de imagen serán las entradas a los clasificadores que se verán en el siguiente capítulo.

La salida del separador de objetos para la imagen 5.8 es el siguiente: Objeto 1 {236,65,319,125 }, Objeto 2 {115,95,133,115}, Objeto 3 {115,135,133,157}

Como se puede ver, solo los objetos que se necesitan forman parte de la lista final de objetos, dejando de lado al objeto superior en la imagen, por no superar con el umbral de área mínima.

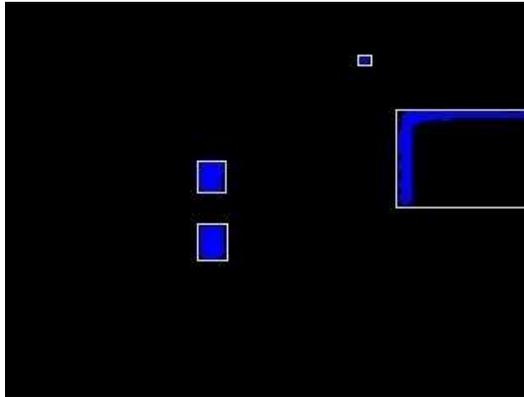


Figura 5.8: Rectángulos delimitadores de los objetos encontrados.

Capítulo 6

Reconocimiento y clasificación de objetos clave

Una vez que se ha realizado el procesamiento de las imágenes, lo que resta es clasificar los objetos obtenidos después de las operaciones vistas en el capítulo anterior.

6.1. Objetos Clave

Primero que nada, se definirán los objetos a clasificar y del porqué son considerados objetos clave dentro de la cancha de fútbol. Como se vio en 2.1.1, se tienen dos tipos de objetos que proporcionan información para la localización del robot con respecto a la cancha: las porterías de cada color, así como los postes de medio campo (amarillo y azul). Reconocer estos objetos es necesario para que el robot pueda saber hacia dónde buscar la portería del equipo contrario. Como se ha presentado hasta ahora, lo que se busca es encontrar los algoritmos con el menor costo computacional posible, que pueda solucionar los problemas de visión. En este capítulo, se presentarán dos tipos de clasificadores que se ajustan a las necesidades de bajo consumo de recursos en el robot. Para los algoritmos que se van a presentar, se hará uso de las listas que se poseen de objetos azules y objetos amarillos, entregado por la etapa de procesamiento de imágenes.

Puntos característicos

Los puntos característicos de los objetos clave se definieron en relación a la necesidad de trasladar las coordenadas 2D de los objetos clave en sus coordenadas reales en 3D para que el robot sea capaz de autolocalizarse gracias a las observaciones hechas por la cámara. El algoritmo de autolocalización se ejecuta en el robot en el módulo de autolocalización en paralelo con el módulo de visión y el resto de los módulos que controlan al robot. Este algoritmo se presentó en [7] y es el trabajo de tesis de maestría

del Ingeniero Alberto Isay Romay Tovar, miembro del equipo “Cyberlords La Salle”. Los puntos que el algoritmo necesita son los siguientes:

- Los 3 centroides de cada sección de los postes de medio campo.
- Las esquinas de la portería: 4 esquinas cuando la vista de la portería es completa y solo dos puntos cuando la portería está vista parcialmente. La figura 6.1 muestra los puntos característicos necesarios para el algoritmo de autocalización que entrega el sistema de visión.

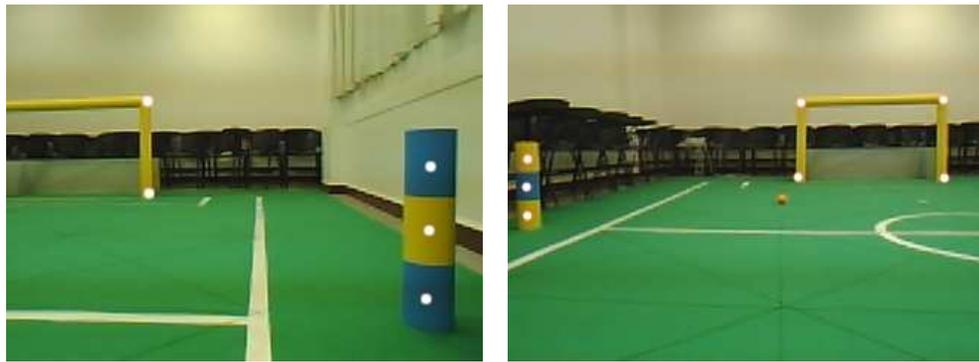


Figura 6.1: Puntos característicos en círculos blancos de los postes de medio campo y las porterías.

6.2. Reconocimiento de postes de medio campo

Para reconocer el poste de medio campo, se parte con la búsqueda del poste amarillo (su código de color es el amarillo–azul–amarillo). Para realizar esto, se toma la lista de objetos azules, y se analiza el primer objeto como sigue:

- El objeto de la lista entrega las 4 coordenadas de las esquinas del rectángulo que contiene un objeto del color encontrado, en este caso el primer objeto azul. Se calcula la altura del objeto y se guarda en una variable de *distancia*, así como la posición del centroide de dicho objeto.
- Se halla una posición $p(x, y)$ calculada con la coordenada x del centroide y en la coordenada y la resta de la y del centroide más la *distancia*.
- Se busca en la imagen binaria amarilla, la posición p hallada en el punto anterior. Si el píxel tiene el valor de 1:
 - Se busca en la lista de objetos amarillos, cuál de los objetos es el que contiene a la coordenada p y se marca tal objeto como posible sección superior del poste.
 - Se halla una posición $q(x, y)$ calculada con la coordenada x del centroide y en la coordenada y la suma de la y del centroide más la *distancia*.

- Se busca en la imagen binaria amarilla, la posición q hallada en el punto anterior. Si el píxel tiene el valor de 1:
 - Se busca en la lista de objetos amarillos, cuál de los objetos es el que contiene a la coordena p y se marca tal objeto como sección inferior del poste.
 - Se da como reconocido el poste amarillo, guardando las coordenadas de los centroides de las secciones que lo conforman y luego eliminando el objeto azul y los dos objetos amarillos de sus respectivas listas a fin de que no vuelvan a ser analizados al momento de buscar portería.
 - Si el píxel tiene valor de 0, entonces la sección azul no corresponde a la del poste de medio campo, y se sigue analizando el siguiente objeto azul.
- Si el píxel tiene valor de 0, entonces la sección azul no corresponde a la del poste de medio campo, y se sigue analizando el siguiente objeto azul.

Este proceso se repite tal cual para buscar el poste azul, en donde se empezó a analizar los objetos amarillos, buscando secciones azules arriba y abajo para así conjuntar el poste azul de medio campo. La facilidad del método está basado en el resultado entregado por el método de procesamiento de la imagen que entregó las listas de objetos.

6.3. Reconocimiento y clasificación de porterías

Para hacer un análisis de los trabajos previos en la RoboCup respecto a las porterías de la liga “KidSize” se tiene una limitante fundamental: desde el 2010, la portería tiene la forma actual presentada en 2.1.1. Con anterioridad poseía una forma más simple como se explica en las reglas del 2009 . [2] La portería anterior contenía todo el color correspondiente por dentro, formando un área muy grande del color muy fácil de identificar, sin necesidad de tomar en cuenta la forma de la portería. Las porterías anteriores se muestran en la figura 6.2. Hasta el momento solo se ha presentado un trabajo en relación al reconocimiento de las nuevas porterías [48], y fue publicado por miembros de un equipo de la categoría “Standard Platform League”, cuya portería es igual al de la cancha de “KidSize”. En ese trabajo describen como reconocen los postes verticales y horizontales de manera independiente por medio de la transformada de Hough, para luego concatenar las líneas encontradas con respecto a la forma de la portería, buscando las líneas próximas entre sí. Como en el caso de este trabajo, utilizan el espacio de color HSV y segmentación basándose en el color. En el presente trabajo se hará uso de los descriptores geométricos para resolver el problema de la clasificación de las porterías, en 4 posibles clases, que hacen referencia a las porciones que el robot puede observar desde dentro de la cancha.

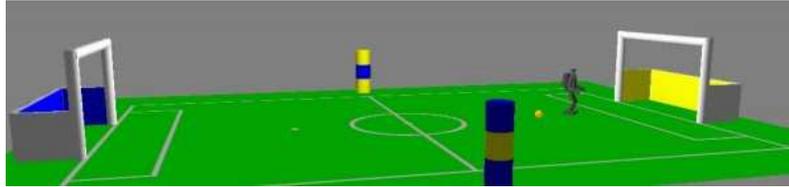


Figura 6.2: Porterías azul y amarillo antes del 2010 [2].

6.3.1. Descriptores geométricos

Área

El área de un objeto en una imagen binaria, donde $I(x, y) = 1$ si el píxel forma parte del objeto, $I(x, y) = 0$ si el píxel forma parte del fondo, está definida por la siguiente fórmula:

$$A = \sum_{x=1}^N \sum_{y=1}^M I(x, y) \quad (6.1)$$

Momentos

La geometría de una región plana se basa en el tamaño, la posición, la orientación y la forma. Todas estas medidas están relacionadas con una familia de parámetros llamada momentos [49]. Los momentos de orden $p + q$ de una imagen $I(x, y)$ son

$$u_{p,q} = \sum_{x=1}^N \sum_{y=1}^M x^p y^q I(x, y) \quad (6.2)$$

Centro de gravedad

El centro de gravedad de un cuerpo es el punto respecto al cual las fuerzas que la gravedad ejerce sobre los diferentes puntos materiales que constituyen el cuerpo producen un momento resultante nulo [50]. El centro de gravedad de un cuerpo no corresponde necesariamente a un punto material del cuerpo. Para una imagen binaria, el centro de gravedad \bar{x}, \bar{y} de un cuerpo se puede expresar respecto al área (ecuaciones 6.3 y 6.4) o respecto a sus momentos (ecuaciones 6.5 y 6.6).

$$\bar{x} = \frac{\sum_{x=1}^N \sum_{y=1}^M x I(x, y)}{A} \quad (6.3)$$

$$\bar{y} = \frac{\sum_{x=1}^N \sum_{y=1}^M yI(x, y)}{A} \quad (6.4)$$

$$\bar{x} = \frac{u_{1,0}}{u_{0,0}} \quad (6.5)$$

$$\bar{y} = \frac{u_{0,1}}{u_{0,0}} \quad (6.6)$$

6.3.2. Momentos invariantes de Hu

El uso de los momentos como una invariante para la representación de formas fue propuesta por Hu en 1961 [51]. Hu utilizó esta técnica para clasificar caracteres escritos a mano. Esta técnica ha sido utilizada ampliamente para distintos tipos de trabajos como reconocimiento de rostros [52], reconocimiento de objetos 3D [53], reconocimiento de expresiones faciales [54], entre otros, debido a que los momentos de Hu son invariantes a tamaño, posición y rotación del objeto. El trabajo de Hu además ha sido extendido en [55] y [56] para hallar invariantes a convoluciones y cambios de contraste.

Para poder hacer una descripción independiente de la posición del objeto, los momentos pueden calcularse respecto al centro de gravedad. Esto debido a que el cálculo es en función de la distancia entre los píxeles del objeto y el origen, se trasladó el origen al centro de gravedad \bar{x}, \bar{y} . De esta manera se está removiendo variaciones respecto a traslaciones en los ejes. Estos momentos son llamados momentos centrales y pueden ser hallados con la ecuación 6.7

$$u_{p,q} = \sum_{x=1}^N \sum_{y=1}^M (x - \bar{x})^p (y - \bar{y})^q I(x, y) \quad (6.7)$$

Como se tiene que $u_{0,0}$ es el área del objeto, se normalizan los momentos para tener una descripción independiente del tamaño. Los momentos centrales normalizados están dados por:

$$\eta_{p,q} = \frac{u_{p,q}}{u_{0,0}^\alpha} \quad \alpha = \frac{p+q}{2} + 1 \quad (6.8)$$

Los momentos individuales no poseen el poder descriptivo único para representar formas arbitrarias, así como tampoco poseen las características invariantes requeridas, pero se puede derivar un conjunto de estos momentos que sí lo poseen. Hu derivó un conjunto de siete momentos invariantes rotacionales con los que se puede representar la forma del objeto. Estos momentos invariantes son independientes de la posición, tamaño y rotación del objeto. Los momentos invariantes se representa a continuación:

$$h_1 = \eta_{2,0} + \eta_{0,2} \quad (6.9)$$

$$h_2 = (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{1,1}^2 \quad (6.10)$$

$$h_3 = (\eta_{3,0} - \eta_{1,2})^2 + (\eta_{2,1} - \eta_{0,3})^2 \quad (6.11)$$

$$h_4 = (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{0,3} + \eta_{2,1})^2 \quad (6.12)$$

$$h_5 = (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{0,3} + \eta_{2,1})^2] + \\ (3\eta_{2,1} - \eta_{0,3})(\eta_{0,3} + \eta_{1,2})[3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{0,3} + \eta_{2,1})^2] \quad (6.13)$$

$$h_6 = (\eta_{2,0} - \eta_{0,2})[(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{0,3} + \eta_{2,1})^2] + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{0,3} + \eta_{2,1}) \quad (6.14)$$

$$h_7 = (3\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{0,3} + \eta_{2,1})^2] - \\ (\eta_{3,0} - 3\eta_{1,2})(\eta_{0,3} + \eta_{2,1})[3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{0,3} + \eta_{2,1})^2] \quad (6.15)$$

Estos siete coeficientes forman el vector descriptor del objeto. Para comprobar qué tanto se puede confiar en la invarianza de este vector descriptor, se presenta una prueba a manera de ejemplo, que consiste en tomar dos imágenes de la portería, una en vista completa y otra donde no se ve uno de los postes. A dichas imágenes se aplicarán ciertas transformaciones a las cuales el vector descriptor no debería de presentar cambios significativos. Para la primera imagen, presentada en la figura 6.3, se aplicó una rotación de 30°, traslación en el eje x , traslación en el eje y , escalamiento a un 50 %, y luego la combinación de todas las transformaciones anteriores. En la tabla 6.1 se puede observar el resultado de encontrar los momentos invariantes de Hu para cada caso. Es posible observar que el error es mínimo entre una y otra transformación con respecto a la imagen original.

Tabla 6.1: Invariantes de Hu para la misma imagen

Imagen	h_1	h_2	h_3	h_4	h_5	h_6	h_7
Original	1.4720	1.0925	0.37815	0.10322	-0.016953	-0.095467	-0.011338
Rotación 30°	1.4890	1.1319	0.35012	0.12299	-0.017550	-0.109220	-0.016810
Traslación en x	1.4721	1.0926	0.37815	0.10322	-0.016953	-0.095467	-0.011338
Traslación en y	1.4721	1.0926	0.37815	0.10322	-0.016953	-0.095467	-0.011338
Escalamiento en 50 %	1.4737	1.1613	0.36522	0.11400	-0.018086	-0.103574	-0.009911
Todas las anteriores	1.4695	1.0955	0.37322	0.10145	-0.016360	-0.093367	-0.011049

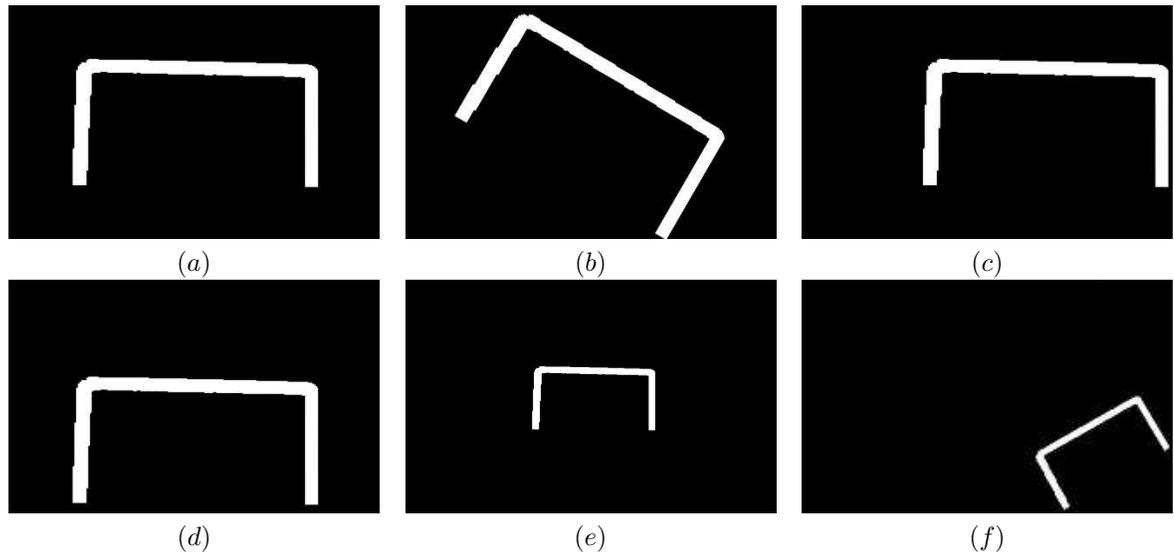


Figura 6.3: Imagen: a) Original, b) Rotación de 30° c) Traslación en x , d) Traslación en y , e) Escalamiento en 50 %, f) Todas las transformaciones anteriores

La segunda imagen, presentada en la figura 6.4 es probablemente la sección de portería que más ocurrencias tiene en cuanto a las observaciones del robot. A esta imagen se le aplicó las transformaciones de translación en x , translación en y , rotación de -45° , imagen espejo y finalmente la combinación de todas las anteriores. La tabla 6.2 presenta los resultados mostrando los cambios mínimos entre las distintas transformaciones y la imagen original.

Tabla 6.2: Invariantes de Hu para la misma imagen

Imagen	h_1	h_2	h_3	h_4	h_5	h_6	h_7
Original	0.82099	0.28945	0.47970	0.055827	-0.0082260	-0.022413	0.0136041
Rotación 45°	0.81107	0.27551	0.38667	0.051282	-0.0033102	-0.012576	0.0064110
Traslación en x	0.80643	0.27458	0.38041	0.052820	-0.0020308	-0.008768	0.0072028
Traslación en y	0.81140	0.28637	0.36582	0.049546	-0.0023144	-0.009926	0.0062561
Espejo	0.81057	0.27110	0.38480	0.049888	-0.0029158	-0.012048	0.0062672
Todas las anteriores	0.77833	0.25210	0.34104	0.045300	-0.0023664	-0.009869	0.0051091

6.3.3. Clases de imágenes de porterías

Una vez demostrado lo viable del método, se describen a continuación las 5 clases que se requieren clasificar las imágenes captadas de las porterías. Esto debido a que no siempre se va a ver la portería completa o desde el mismo ángulo, pero cada clase es importante que se clasifique sin importar su

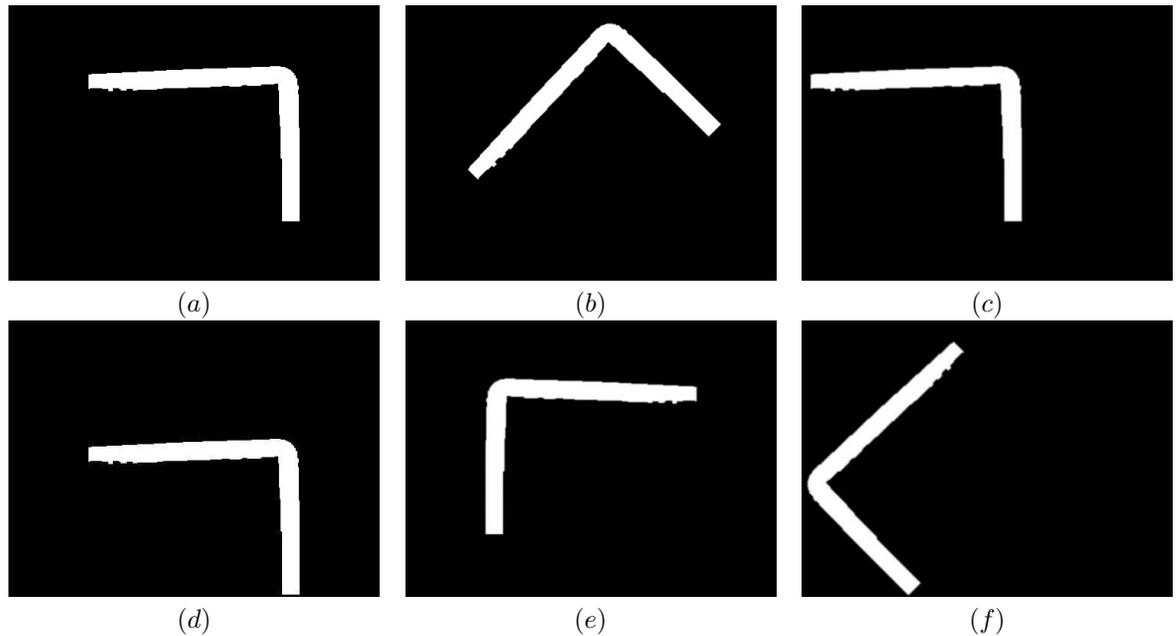


Figura 6.4: Imagen: (a) Original, (b) Rotación de 30° (c) Traslación en x , (d) Traslación en y , (e) Espejo, (f) Todas las transformaciones anteriores

posición, tamaño o rotación. Se definieron las siguientes clases:

- Portería completa
- Portería media grande
- Portería media pequeña
- Poste
- No portería

Ejemplos se muestran en la figura 6.5.

Estas clases representan las diferentes imágenes captadas de la portería o de un objeto lo suficientemente grande para ser considerado portería pero que no lo es. Las imágenes que el robot adquiere pueden estar con cierto grado de rotación, traslación y definitivamente con tamaño diferente, esto debido a la distancia en que se podría encontrar el robot de la portería, además no siempre se va a ver a la portería desde el frente, sino que podría verse con una cierta perspectiva, que se considerará al momento de realizar la base de datos de los momentos de Hu para clasificar las imágenes.

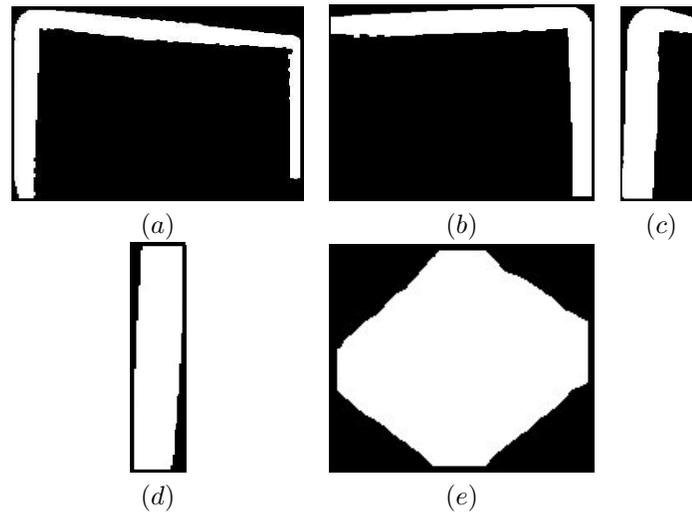


Figura 6.5: Clases de la portería: a) Completa, b) Media grande, c) Media pequeña, d) Poste, e) No portería

6.3.4. Clasificación

Los métodos de clasificación por medio de aprendizaje máquina son siempre el cuello de botella a la hora de separar clases en aplicaciones de tiempo real en sistemas embebidos, respecto a la velocidad del proceso. Por eso se optó por probar el clasificador conocido como distancia mínima o error mínimo. Para esto, se ha preparado una base de datos previamente calculada, con una serie de momentos de Hu de imágenes correspondientes a cada clase. Más adelante se detallará más acerca de la construcción de la base de datos. La distancia entre dos vectores se calcula de la siguiente manera:

$$\sum_{i=0}^{N-1} |X_i - M_i|, \quad (6.16)$$

donde X representa el vector de momentos de Hu para una imagen en la base de datos y M representa el vector de momentos de Hu de la imagen adquirida a clasificar. Con la ecuación 6.16 se está hallando la distancia L_1 (también conocida como distancia Manhattan) entre los vectores de momentos de Hu X y M . El algoritmo de clasificación realiza la misma operación pero sobre todo el conjunto de imágenes dentro de la base de datos, para encontrar cuál de esas imágenes es la más cercana a la que se está tratando de clasificar. Al encontrar el error mínimo, simplemente se clasifica la imagen M como si fuera de la misma clase de la imagen que es más cercana. El algoritmo se ejecuta como sigue:

- Hallar momentos invariantes de Hu de la imagen M binaria a clasificar.
- Desde inicio de base de datos hasta el fin de la base de datos hacer:
 - Hallar la distancia L_1 entre el vector X_i perteneciente a la base de datos y el vector M .

- Si la distancia actual es menor que la anterior, actualizar con el valor actual.
- Clasificar respecto a la clase que tenga la distancia mínima.

En la figura 6.6 se puede observar el algoritmo para encontrar la clase a la cual pertenece la imagen adquirida.

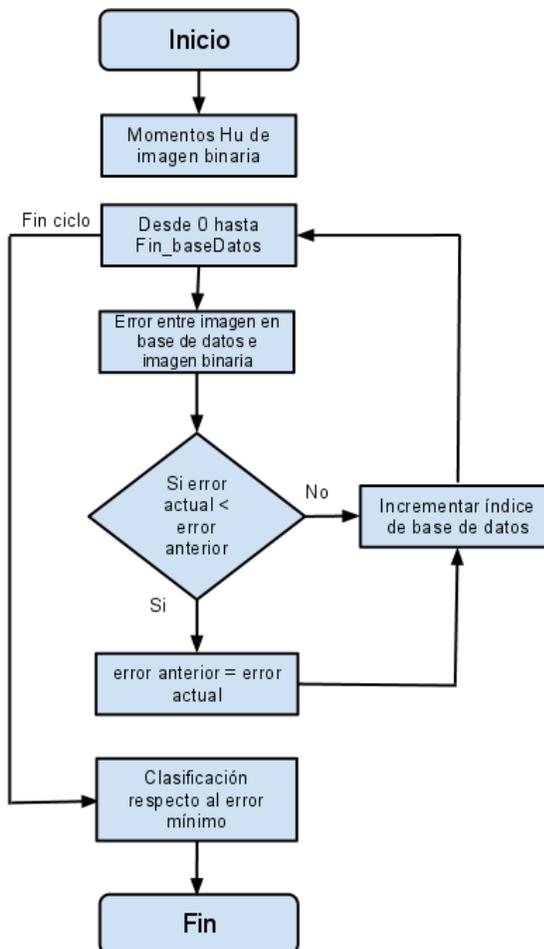


Figura 6.6: Algoritmo de clasificación

6.4. Base de datos

Dada la premisa de los algoritmos de aprendizaje máquina, donde se necesita entrenar al sistema con una cantidad muy grande de información, al principio de este clasificador se utilizó la misma idea.

Por ejemplo, las redes neuronales utilizan generalmente más del 50 % de los datos disponibles para entrenar. Para explicar a detalle, primero se necesita hacer mención al conjunto de datos utilizados para probar esta metodología de clasificación.

6.4.1. Conjunto de datos

Se detallará a continuación la creación y características de los 3 tipos de conjuntos utilizados para probar los algoritmos. Cabe resaltar que el conjunto de datos se generó para probar los dos clasificadores, el de porterías y el de postes de medio campo. En la sección de pruebas y resultados, se presentan los resultados de ambos clasificadores.

Conjunto 1

Este conjunto fue el único generado en una cancha completa reglamentaria de la liga “KidSize”, que se pudo realizar gracias al permiso de la Facultad de Ingeniería de la Universidad La Salle, permitiendo al *Laboratorio de Robótica Móvil y Sistemas Automatizados*, utilizar un salón completo para instalar la cancha por un corto periodo. Para probar la efectividad de la propuesta metodológica se ideó un experimento que comprende el uso del campo de fútbol de RoboCup del laboratorio, el robot humanoide “T1” y una cámara PlayStation Eye montada sobre el mecanismo de paneo e inclinación del robot con una resolución de 320×240 píxeles. Se marcó el campo con una rejilla hecha de hilos negros a intervalos de 100cm. Cada uno de los recuadros resultantes se cruzó de manera diagonal en ambas direcciones. El patrón resultante se puede observar en las figuras 6.7 y 6.8 .



Figura 6.7: Montaje de la cancha (vistas global y en acercamiento)

El robot se programó para mantener una pose estándar y se utilizó una fuente de poder externa para asegurar que esta pose permaneciera constante a lo largo de todo el experimento. El robot se colocó manualmente en cada una de las esquinas de la rejilla y se tomó una imagen para cada una de las ocho diferentes orientaciones indicadas por los cruces de los hilos. Se colocaron marcas en los pies del robot para servir como guía al momento de colocar al robot en las diferentes orientaciones. Se tomaron un total de 244 imágenes, en dos subconjuntos:

- NO_ZOOM: 127 imágenes con la cámara en su modo de operación normal.

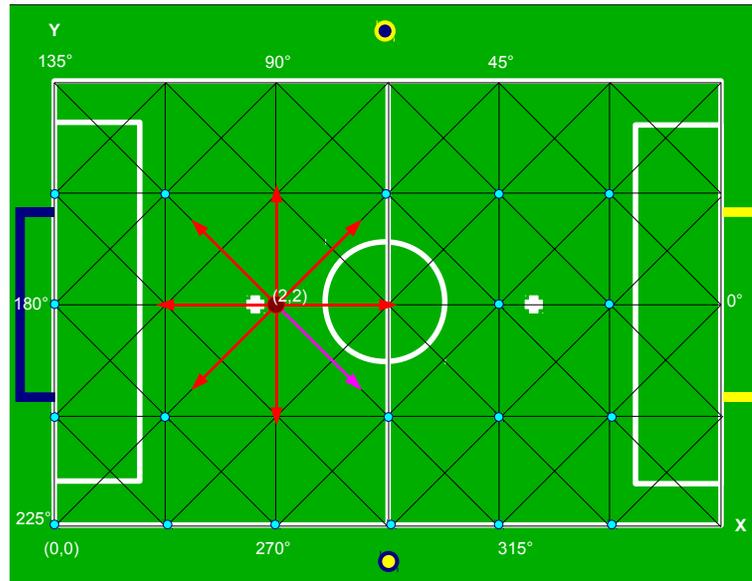


Figura 6.8: Vista aérea de la cancha . En rojo las observaciones utilizadas, en morado la imagen a 315° excluida por las reglas de la categoría “KidSize”

- Si_ZOOM: 117 imágenes con la cámara en con el zoom óptico habilitado.

Cada imagen se etiquetó con toda la información referente a la posición (x, y) y orientación θ de la cámara del robot con respecto a la cancha. Como se puede observar, no todas las imágenes contienen información útil al momento de utilizarlas para reconocer los objetos, debido a que dichos objetos simplemente no se pueden observar dado la posición x, y del robot y el ángulo de orientación.

Conjunto 2

Para facilitar la creación de la base de datos de las porterías, se llevó a cabo una segunda captura de imágenes, pero ahora solo teniendo en cuenta las porterías. El punto importante aquí fue capturar ambas porterías a distintas distancias, diferente perspectiva y diferentes porciones de la misma, previamente identificadas como las clases que son de interés. Para este conjunto se capturó la suma total de 250 imágenes exclusivamente de las porterías. Esta sesión se realizó con la cámara web Logitech a una resolución de 320×240 píxeles.

Conjunto 3

Para efecto de pruebas, a fin de que el proceso pueda ser más dinámico se capturaron una serie de videos con un promedio de cuatro minutos cada uno, que si se expresan como una sucesión de imágenes, 30 de éstas por segundo, equivale a tener más de 7000 imágenes de prueba en cada video.

- Video 1: video de la portería amarilla vista completamente desde diferentes distancias y ubicaciones en la cancha, con algunas inclinaciones de la cámara generando de esta manera imágenes rotadas y trasladadas dentro del video.
- Video 2: video de la portería azul vista siempre como se definió la portería media grande, cuidando siempre no ver la portería completa o una sección más pequeña de lo que se definió para la clase.
- Video 3: video de la portería amarilla vista siempre como se definió la portería media pequeña, cuidando de solo cambiar la distancia y perspectiva de la sección de la portería vista.
- Video 4: video de la portería azul con la vista de un solo poste, cuidando solo cambiar la distancia y perspectiva del poste visto.
- Video 5: video de la portería amarilla donde se combina todas las vistas con diferentes perspectivas y distancias para comprobar la clasificación completa.
- Video 6: video del poste amarillo de medio campo, desde diferentes puntos de la cancha, cuidando de que siempre se encuentre visible.
- Video 7: video del poste azul de medio campo, desde diferentes puntos de la cancha, cuidando de que siempre se encuentre visible.

Considerando que las pruebas se realizaron sobre todo el conjunto, se contó con más de 42000 imágenes de prueba.

6.4.2. Optimización de la base de datos

Toda la experimentación para la realización de la base de datos fue hecha sobre el conjunto 2 de datos, explicados en la sección anterior. Primeramente, del conjunto total de 250 imágenes, se utilizaron 200 imágenes para la base de datos (homológamente, a la fase de entrenamiento en algoritmos de aprendizaje máquina), separando 50 imágenes para cada clase distinta, considerando solamente las clases pertenecientes estrictamente a la portería. La clase “no portería” todavía no se consideró en esta parte del trabajo. Al ver que se obtuvo un 100 % de clasificación, para lograr una optimización de la cantidad de comparaciones que debe de realizar el clasificador visto en la sección 6.3.4, se procedió a disminuir la cantidad de imágenes en la base de datos para así acelerar el tiempo de ejecución del algoritmo. Los resultados de las pruebas de clasificación se muestran en la tabla 6.3

Como se puede observar, se redujo el tamaño de la base de datos a un porcentaje de tan solo el 10 % de la cantidad original, manteniendo un porcentaje de clasificación del 100 %, mostrando así la eficiencia de los momentos invariantes de Hu para representar las clases de portería. Teniendo la base de datos optimizada solo con 20 imágenes, se tuvo listo el sistema para ser probado con los demás conjuntos de datos.

Tabla 6.3: Pruebas de clasificación

Base de datos (imágenes)	Imágenes por clase	Imágenes prueba	Porcentaje de clasificación
200	50	50	100 %
100	25	150	100 %
60	15	190	100 %
40	10	210	100 %
20	5	230	100 %
8	2	242	55.371 %

6.5. Pruebas preliminares y resultados

Las pruebas preliminares y resultados se presentarán en dos secciones, la primera para la prueba de los postes de medio campo y la segunda para las porterías.

6.5.1. Prueba 1: postes de medio campo

Para probar el algoritmo de reconocimiento de postes de medio campo, se utilizó el conjunto de datos 1 y parte del conjunto de datos 3, en particular los videos 6 y 7.

Las pruebas sobre el conjunto 1 dio como resultado lo siguiente:

- 127 imágenes del subconjunto NO_ZOOM: 100 % de reconocimiento de postes amarillos y 100 % de reconocimiento de postes azules.
- 117 imágenes del subconjunto SI_ZOOM: 100 % de reconocimiento de postes amarillos y 100 % de reconocimiento de postes azules.
- Video 6 con 7000 cuadros, descrito en la sección 6.4.1, con un porcentaje de reconocimiento del 100 % de postes amarillos.
- Video 7 con 7000 cuadros, con un porcentaje de reconocimiento del 100 % de postes azules.

6.5.2. Prueba 2: portería

Para probar el reconocimiento y clasificación de la portería en sus distintas clases, se utilizó el conjunto 2 y parte del conjunto 3, específicamente los videos 1, 2, 3, 4 y 5. Como ya se ha presentado en la sección 6.4.2, el clasificador tuvo un resultado de 100 % de reconocimiento y clasificación sobre el conjunto 2, con un total 230 imágenes de prueba, donde las 20 restantes son las pertenecientes a la base de datos optimizada.

La prueba certera para conocer el rendimiento del algoritmo de clasificación se realizó sobre los videos debido a la cantidad de imágenes donde se pudo probar. En la tabla 6.4 se puede ver el resultado de la clasificación.

Tabla 6.4: Resultado de la clasificación sobre las imágenes con portería

Video	Núm. de imágenes	Clasificación correcta	Error	Falsos positivos
1	7019	100 %	0 %	0 %
2	6986	100 %	0 %	0 %
3	7035	100 %	0 %	0 %
4	6995	100 %	0 %	0 %
5'	7010	100 %*	23.195 %	23.195 %

Con el video 5, se puede ver algo muy particular. Los resultados dieron un porcentaje de clasificación mayor al número de cantidades de objetos vistos en el video; para ser exactos, se tuvo 1626 imágenes clasificadas de más sobre el conjunto de 7010, teniendo un total de 8636 porterías clasificadas. Se analizó el video para ver qué particularidad posee el video para que proporcionara estos falsos positivos. El error consistió básicamente en la técnica del clasificador: como se ha visto en la sección 6.3.4, el clasificador solo distingue entre las 4 posibles clases de portería, siendo así cualquier objeto del color buscado que no se haya eliminado de la lista de objetos va a caer en una de éstas 4 clases obligatoriamente. El video 5 es el único de entre el conjunto de datos donde en el ambiente existe objetos del color que no son parte de la cancha (mochila amarilla y un puf amarillo, ambos de la misma tonalidad del amarillo de la portería), lo cual generó un nuevo problema a resolver: la clase “no portería”.

6.6. Clase “no portería”

La primera aproximación es buscar un margen de error entre clases, pero al analizar los valores de la base de datos, se pudo observar que las clases no son linealmente separables, por lo cual se efectuó a aplicar otras alternativas. La primera y muy efectiva fue el filtrar la lista de objetos.

6.6.1. Filtro de área y proporción

La primera medida aplicada fue la de crear un filtro para eliminar de la lista de objetos, todos aquellos cuya área y proporción no corresponda a una posible vista de la portería. Como se vió en la sección 6.2, al terminar la búsqueda de los postes de medio campo, si fueron encontrados estos son eliminados de las listas de objetos azules y amarillos. Una vez terminado este proceso, se eliminarán todos los objetos azules y amarillos que no poseen un tamaño y proporción adecuado para ser una sección de portería. Para realizar esto, primero se tuvo que medir, dado la resolución a la que se está trabajando (320×240), desde la distancia máxima de 6.5 metros (el robot no puede estar en un punto más lejano dentro de la cancha). En la tabla 6.5 se pueden ver las medidas mínimas que pueden tener los objetos que pueden ser secciones de la portería.

Con estas medidas, se pueden derivar dos reglas generales para discriminar objetos antes de que estos pasen a la etapa de clasificación:

Tabla 6.5: Medidas de la portería vistas a la máxima distancia en la cancha

Clase	Base	Altura	Área	Proporción altura / base
Completa	105	55	5775	0.524
Media grande	55	55	3025	1
Media pequeña	15	55	825	3.667
Poste	7	55	385	7.857

- El área del rectángulo que contiene al objeto debe de poseer un área mayor a 350.
- Si el área es menor a 2500 y mayor a 350, se debe comprobar que la altura del rectángulo sea al menos de 52 píxeles.

Cualquier objeto que no cumpla estas restricciones se elimina automáticamente de las listas de objetos.

6.6.2. Nueva clase

Una vez que se tiene en la lista solamente objetos que pueden tener el tamaño necesario para ser considerado parte de la portería, se realizó pruebas para la creación de la clase “no portería”, principalmente para agregar objetos particulares que se pudieran encontrar en el ambiente de la competencia (carteles publicitarios, columnas, etc.) que sean de color amarillo o azul, que el equipo considere sea un peligro de falso positivo al momento de clasificar las porterías. En el laboratorio se probó con objetos grandes de color amarillo y azul, como fueron mochilas, playeras, pedazos de cartón de los colores de interés, entre otros. Después de la creación de la nueva clase, así como la aplicación del filtro de área y proporción, en el video 5 se pudo eliminar sin ningún inconveniente los falsos positivos encontrados.

6.7. Resultados en competencia

La competencia de la RoboCup proporciona el ambiente real para la tarea, y a diferencia del laboratorio donde el ambiente puede ser fácilmente controlado, es el ambiente donde todos los grupos de investigación participantes tienen un espacio en igualdad de condiciones para probar a sus robots. En esta sección se muestran los resultados finales de los partidos en los cuales se ha participado en los años 2010 y 2011, correspondiente a los años de desarrollo del trabajo de esta tesis. Con esto se quiere comprobar que el trabajo presentado no es solo válido en laboratorio, sino que en situación real el robot hace uso del sistema de visión como principal sensor para reconocer su ambiente y jugar fútbol.

6.7.1. Singapur 2010

La edición del año 2010 se llevó a cabo en Singapur y el equipo “Cyberlords La Salle” participó probando así la primera versión del sistema de visión capaz de proporcionar información de orientación

dentro de la cancha con el reconocimiento de los postes de medio campo y una versión inestable de reconocimientos de portería que se descartó después de la competencia. La tabla 6.6 muestra el resultado de los partidos jugados por el equipo, mostrando el equipo contrario, su país de origen, goles a favor y goles en contra.

Tabla 6.6: Resultados de los partidos disputados en el 2010

Equipo	País	GF	GC
Darmstadt Dribblers	Alemania	0	10
NYP Lions	Singapur	1	0
Robo Erectus	Singapur	4	2
Bogobots-TecMTY	México	0	0
Darwin	USA	1	5
Team KMUTT	Tailandia	0	1

6.7.2. Turquía 2011

La Edición del año 2011 se llevó a cabo en Turquía y el equipo “Cyberlords La Salle” participó probando así la versión final del sistema de visión capaz de proporcionar información de orientación dentro de la cancha con el reconocimiento de los postes de medio campo y el reconocimiento y clasificación de la portería, pudiendo así ser más certero al momento de hacer los tiros a gol. La tabla 6.7 muestra el resultado de los partidos jugados por el equipo, mostrando el equipo contrario, su país de origen, goles a favor y goles en contra.

Tabla 6.7: Resultados de los partidos disputados en el 2011

Equipo	País	GF	GC
Darmstadt Dribblers	Alemania	0	10
JEAP	Japón	0	0
FalconBots	México	2	0
Pioneros Mexico	México	4	0
TKU	Taiwán	1	1
FUmanoids	Alemania	0	1

Capítulo 7

Conclusiones y Trabajo Futuro

La metodología presentada para que el sistema de visión se pueda calibrar para soportar cambios de iluminación brinda una ventaja muy grande debido a que se están modificando parámetros de la cámara para adquirir las imágenes acorde al requerimiento presentado. Los algoritmos de constancia de color y de adaptación de los espacios de color o clasificadores presentan un costo computacional elevado para sistemas embebidos de un solo procesador. Para la tarea de la RoboCup, así como para otras tareas en sistemas embebidos, la solución aquí presentada es eficaz y eficiente en cuanto a costo computacional. Este trabajo se puede extender a sistemas de mayor poder de cómputo, creando una combinación del trabajo presentado con las demás soluciones netamente de *software*.

Los algoritmos de reconocimiento y clasificación usados para la tarea específica, entregaron resultados satisfactorios del 100 % sobre el conjunto de datos a un bajo costo computacional. La única desventaja de los algoritmos es la sensibilidad presentada frente a una mala segmentación de las imágenes, pero que en este caso, al menos el cambio de luz no afecta la calidad de la segmentación. La herramienta principal para la clasificación en la propuesta de este trabajo son los descriptores geométricos invariantes de Hu, teniendo como desventaja que el objeto de interés debe estar bien separado de los objetos de fondo.

El campo para trabajo futuro es bastante amplio, la cancha de fútbol de la RoboCup posee más marcas a ser identificadas, así como en unos años, algunas de estas marcas dejarán de existir, como por ejemplo, las porterías ya no tendrán color, las dimensiones de la cancha y objetos dentro de ella irán aumentando de tamaño. Los algoritmos se deberán ir adaptando acorde con los cambios dentro del ambiente de trabajo. En un ámbito más cercano, los módulos desarrollados para esta tesis, están siendo adaptados para el uso de un producto de video vigilancia automatizada, donde se requiere solucionar problemas de iluminación, así como la detección y clasificación de objetos, personas así como movimiento en general.

Bibliografía

- [1] RoboCup Humanoid League. *RoboCup Soccer Humanoid League Rules and Setup for the 2010 competition in Singapore*. <http://www.tzi.de/humanoid>, March 2010.
- [2] RoboCup Humanoid League. *RoboCup Soccer Humanoid League Rules and Setup for the 2009 competition in Graz, Austria*. <http://www.tzi.de/humanoid/bin/view/Website/Downloads>, March 2009.
- [3] Luis F Lupian, Alberto Romay, Andres Espinola, Rafael Cisneros, Juan Manuel Ibarra, Daniel Gutierrez, Manuel Hunter, Christian del Valle, and Karla de la Loza. Cyberlords RoboCup 2010 Humanoid KidSize Team Description paper. In *Proceedings of the RoboCup Symposium*, 2010.
- [4] Luis F. Lupián, Alberto Romay, Andrés Espínola, and Erick Ramírez. Cyberlords RoboCup 2011 Humanoid KidSize team description paper. In *RoboCup World Championship*, Istanbul, Turkey, July 2011. RoboCup Federation.
- [5] Wikipedia, the free encyclopedia. Wikipedia website. World Wide Web electronic publication: <http://es.wikipedia.org/wiki/Futbol> Retrieved 2011/07/10.
- [6] The Robot World Cup Initiative. RoboCup Website. World Wide Web electronic publication: <http://www.robocup.org> Retrieved 2011/07/10.
- [7] Luis F Lupián, Alberto Romay, and Andrés Espínola. Vision based localization of humanoid robots by inverse pose-estimation using a small set of fixed landmark features. In *Robotic Symposium, IEEE Latin American*, Bogotá, Colombia, 2011.
- [8] Alan K. Mackworth. On seeing robots. In A. Basu and X. Li, editors, *Computer Vision: Systems, Theory and Applications*, pages 1–13. World Scientific Press, Singapore, 1993. Reprinted in P. Thagard (ed.), *Mind Readings*, MIT Press, 1998.
- [9] HITEC Robotics. HITEC website. World Wide Web electronic publication: <http://www.robonova.de>.
- [10] DMP Electronics Inc. Webpage for roboard. <http://www.roboard.org> Retrieved 2011/08/18.
- [11] Andrés Espínola, Tetyana Baydyk, Ernst Kussul, and Felipe Lara. Calibration of camera settings to not affect the thresholds of segmentation in a task performed in an environment of changing

- illumination. In *International Congress on Instrumentation and Applied Sciences 2010. Conference proceedings (ISBN 978-607-02-0840-9)*, 2010.
- [12] David Ball, Gordon Wyeth, and Stephen Nuske. A global vision system for a robot soccer team. In Nick Barnes and David Austin, editors, *Australasian Conference on Robotics and Automation*, Canberra, 2004. Australian Robotics and Automation Association Inc.
- [13] Gourab Sen Gupta, C. H. Messom, and S. Demidenko. Real-time identification and predictive control of fast mobile robots using global vision sensing. *IEEE Transactions on Instrumentation and Measurement*, 54:200–214, 2005.
- [14] Z. L. Cao, S. J. Oh, and E. L. Hall. Dynamic omnidirectional vision for mobile robots. *Journal of Robotic Systems*, 3:5–17, 1986.
- [15] N. Winters, J. Gaspar, G. Lacey, and Santos J. Victor. Omni-directional vision for robot navigation. In *Proceedings of the IEEE Workshop on Omnidirectional Vision*, pages 21–28, 2000.
- [16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.
- [17] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. *ECCV*, pages 404–417, 2006.
- [18] Miguel Cazorla and Anna Romero. Comparativa de detectores de características visuales y su aplicación al SLAM. *Workshop of Physical Agents*, 2009.
- [19] D.A. Mitzias and B.G. Mertzios. A neural multiclassifier system for object recognition in robotic vision applications. *Measurement, Elsevier*, 36:315–330, 2004.
- [20] T. Baydyk. Application of flat image recognition technique for automation of micro device production. In *Proceedings of the International Conference on Advanced Intelligent Mechatronics AIM01, Italy, July*, pages 48–494, 2001.
- [21] E. Kefalea, O. Rehse, and v.d. Malsburg. Object classification based on contours with elastic graph matching. In *Proceedings of the 3rd International Workshop on Visual Form, Singapore*, 1997.
- [22] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [23] G.K. Toledo Ramírez. *Investigación y desarrollo de sistemas de control mediante visión técnica para micromáquinas, herramientas y micromanipuladores*. PhD thesis, Facultad de Ingeniería, Universidad Nacional Autónoma de México, 2007.
- [24] Tarokh M., Merloti P., Duddy J., and Lee M. Vision based robotic person following under lighting variations. In *Third International Conference on Sensing Technology, Tainan, Taiwan*, 2008.

- [25] IDS Imaging Development Systems GmbH. *Manual for uEye Cameras. Version 3.50.00*, October 2009.
- [26] Sanjay Kr. Singh, D. S. Chauhan, Mayank Vatsa, and Richa Singh. A robust skin color based face detection algorithm. *Tamkang Journal of Science and Engineering*, 6:227–234, 2003.
- [27] R. Gonzalez and P. Woods. *Digital Image Processing*. Addison-Wesley, 2002.
- [28] E.L. van den Broek, P. M. F. Kisters, and L.G. Vuurpijl. chapter The utilization of human color categorization for content-based imageretrieval, pages 351–362. Bellingham, Wash., USA : Society of Photo-optical Instrumentation Engineers, 2004.
- [29] S. Sural, Gang Qian, and S. Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *International Conference on Image Processing*, volume 2, pages II–589 – II–592, 2002.
- [30] Hasan Fleyeh. Traffic signs color detection and segmentation in poor light conditions. In *Conference on Machine Vision Applications, Tsukuba Science City, Japan*, pages 8–20, 2005.
- [31] Michael H Schimek, Bill Dirks, Hans Verkuil, and Martin Rubli. *Video for Linux Two API Specification: Revision 0.24*, March 2008.
- [32] Nokia. Qt cross-platform. <http://qt.nokia.com>.
- [33] Gordon Wyeth and Ben Brown. Robust adaptive vision for robot soccer. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2000)*, pages 41–48. Press, 2000.
- [34] A. Gijsenij, T. Gevers, and J. van de Weijer. Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, (in press), 2011.
- [35] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of The Franklin Institute-engineering and Applied Mathematics*, 310:1–26, 1980.
- [36] Robby T. Tan, Ko Nishino, and Katsushi Ikeuchi. Color constancy through inverse-intensity chromaticity space. *J. Optical Society of America A*, 21:2004, 2004.
- [37] D.A. Forsyth. A novel algorithm for color constancy. *IJCV*, 5(1):5–36, August 1990.
- [38] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *PAMI*, 33(4):687–698, April 2011.
- [39] Matthias Jünger, Jan Hoffmann, and Martin Löttsch. A real-time auto-adjusting vision system for robotic soccer. In *In 7th International Workshop on RoboCup 2003 (Robot World Cup Soccer Games and Conferences), Lecture Notes in Artificial Intelligence*, pages 214–225. Springer, 2004.

- [40] Anzani Federico, Bosisio Daniele, Matteucci Matteo, and Sorrenti Domenico Giorgio. On-line color calibration in non-stationary environments. In Itsuki Noda Ansgar Bredendfeld, Adam Jacoff and Yasutake Takahashi, editors, *RoboCup 2005: Robot Soccer World Cup IX*, volume LNCS 4020/2006, pages 396–407. Springer Berlin / Heidelberg, June 2006.
- [41] Hamilton Y Chong, Steven J Gortler, and Todd Zickler. A perception-based color space for illumination-invariant image processing. *ACM Transactions on Graphics*, 27(3), 2008.
- [42] Gonzal Pajares Martinsanz and Jesús M. de la Cruz Garcia. *Visión por computador: Imágenes digitales y aplicaciones*. RA-MA Editorial, second edition, 2008.
- [43] Jonathan Weber, Sébastien Lefèvre, and Pierre Gançarski. Spatio-temporal quasi-flat zones for morphological video segmentation. In *ISMM*, pages 178–189, 2011.
- [44] Mauro Dalla Mura, Jon Atli Benediktsson, and Lorenzo Bruzzone. Self-dual attribute profiles for the analysis of remote sensing images. In *ISMM*, pages 320–330, 2011.
- [45] Johan Debayle and Jean-Charles Pinoli. General adaptive neighborhood viscous mathematical morphology. In *ISMM*, pages 224–235, 2011.
- [46] S Jarayaman, S Esakkirajan, and T VeerakumarV. *Digital Image Processing*. Tata McGraw Hill, 2009.
- [47] S. Suzuki and K. Be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, April 1985.
- [48] José Cañas Plaza, Eduardo Perdices, Tomás González, and Domenec Puig. Recognition of Standard Platform RoboCup Goals. *Journal of Physical Agents*, 4(1), 2010.
- [49] R Mukundan and K R Ramakrishnan. *Moments funtions in image analysis. Theory and Applications*. World Scientific Publishing Co, 1998.
- [50] Wikipedia. Centro de gravedad. http://es.wikipedia.org/wiki/centro_de_gravedad Retrieved 2011/08/18.
- [51] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, February 1962.
- [52] Jean-Christophe Terrillon, Martin David, and higeru Akamatsu. Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments. In *In Proc. of the Third International Conference on Automatic Face and Gesture Recognition*, pages 112–117, 1998.
- [53] Muharrem Mercimek, Kayhan Gulez, and Tarik Veli Mumcu. Real object recognition using moment invariants. *Sadhana*, 30(6):765–775, 2005.

-
- [54] Y. Zhu, L. C. De Silva, and C. C. Ko. Using moment invariants and HMM in facial expression recognition. *Pattern Recogn. Lett.*, 23:83–91, January 2002.
- [55] Jan Flusser and Barbara Zitová. Combined invariants to convolution and rotation and their application to image registration. In *ICAPR'01*, pages 359–368, 2001.
- [56] S Maitra. Moment invariants. In *Proceedings of th IEEE*, volume 67, pages 697–699, 1979.