



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**LOCALIZACIÓN EN ROBOTS HUMANOIDES MEDIANTE
REGRESIÓN SOBRE VISTAS SINTÉTICAS**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS (COMPUTACIÓN)
PRESENTA:

BARUCH DEMIAN GAXIOLA VALLES

DIRECTOR DE TESIS:
DR. FERNANDO ARÁMBULA COSÍO
CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO, UNAM

MÉXICO D.F. MARZO 2013



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.

Baruch Demian Gaxiola Valles: *Localización en Robots Humanoides Mediante Regresión Sobre Vistas Sintéticas*, © MARZO 2013

La ciencia es un mecanismo, una forma de mejorar nuestro conocimiento de la naturaleza. Es un sistema para comparar nuestros pensamientos con el universo, y ver si se corresponden.

— Isaac Asimov

Dedicado a aquellas personas que paciente y cariñosamente me han acompañado en mi ausencia...

D.

*Una máquina puede realizar el trabajo
de cien hombres ordinarios,
pero no hay máquina que pueda hacer el trabajo
de un hombre extraordinario.*

— Ebert Hubbard

AGRADECIMIENTOS

Una vez más encabezan la lista mis padres, más allá de por su apoyo incondicional, porque son tan fuertes cada uno a su manera, que para mí no ha habido otro camino que seguir adelante y con la frente en alto. Soy con orgullo, pero muy a mi manera, un reflejo de los dos. A mis hermanos también les agradezco, pues me han ayudado más de lo que creen. Una por aguantarme tal y como soy, tal y como es. El otro por que al ser mi eterno rival, nunca me ha dejado dar un paso atrás.

A Liz, por haberme acompañado desde el principio de este proceso, y haberme apoyado como nadie, hasta el final; porque verla seguir su sueño, me ayuda a seguir con el mío.

A tantos amigos y familiares que han estado ahí, de cerca, de lejos, cumpliéndome mis caprichos cafeteros de cada semana, como válvula de escape del estrés cotidiano. No se que hubiera hecho sin ellos. Gracias por escucharme, distraerme, acompañarme al aeropuerto y recogerme de regreso. Ya me tocará agradecerles personalmente.

También quiero agradecer a aquellas personas que me ayudaron directamente a crecer durante esta etapa. A mi tutor de tesis, el Dr. Fernando Arámbula, así como al Dr. Walterio Mayol, quien fue mi asesor durante la estancia en Bristol; les agradezco por confiar en mí de la forma en que lo hicieron, y por haberme apoyado tanto para vivir la experiencia de desempañarme en un país diferente al mío, como para aprender a hacer investigación científica, que es el camino que he decidido para mi vida. Al Dr. Adalberto Llarena, quién me acogió bajo su experiencia y conocimiento, como bajo su propio techo; este trabajo no habría existido de no ser por él.

Por último, pero no menos importante, agradezco todo el apoyo que me brindaron las siguientes instituciones: La UNAM, mi *alma máter*, que ha sido mi segundo hogar desde hace más años de los que me gustaría recordar; el IIMAS, por darme la oportunidad de seguir creciendo y encontrar el camino que me satisface; el CONACYT, por brindarme la tranquilidad para dedicarme a mis estudios sin pensar en lo económico, y también por abrirme las puertas a una experiencia increíble en el extranjero; a la Universidad de Bristol por acogerme durante mi estancia de investigación; y a la Coordinación de Estudios de Posgrado de la UNAM, por el apoyo que me dieron para mi viaje.

RESUMEN

El presente trabajo introduce un método de localización para robots móviles que cuenten con una cámara y que se desenvuelvan en un ambiente de dimensiones y proporciones conocidas de antemano. El método lleva a cabo una regresión de una imagen tomada por la cámara, sobre vistas sintéticas generadas mediante un modelo tridimensional del entorno. Las vistas sintéticas se encuentran almacenadas en una base de datos que contiene las representaciones paramétricas de dichas vistas, de tal forma que sólo se emplea un conjunto reducido de puntos de control para describir cada una.

En este trabajo se definió una métrica que permitiera comparar una imagen tomada de la cámara del robot, con una vista sintética preprocesada a partir de un modelo tridimensional del ambiente conocido de antemano, para así asignar valores de semejanza entre la imagen y las vistas sintéticas de una base de datos que representan las poses en que se puede encontrar un robot dentro de la cancha. Se desarrollaron también filtros que permitieron que, después de analizar su entorno percibido desde su pose actual, el robot obtenga un número reducido de poses candidatas en las cuales pudiera encontrarse. Además, se analizaron dos casos en los que se aplicó una búsqueda de locación informada, pasándole al método una aproximación de la pose donde puede estar el robot, y una probabilidad de que se encuentre ahí, consiguiendo resultados sobresalientes, con lo que finalmente, el método se puede utilizar no sólo para aproximar la posición real del robot, sino como una buena herramienta para realizar un rastreo en tiempo real de sus movimientos mediante métodos probabilísticos.

CONTENIDO

1	INTRODUCCIÓN	1
1.1	Localización Robótica	3
1.2	Nuestro Método	4
1.3	Objetivos	6
2	MARCO TEÓRICO	9
2.1	Antecedentes Histórico - Contextuales	9
2.1.1	Robótica	9
2.1.2	Robots Humanoides	12
2.1.3	RoboCup	12
2.2	Antecedentes Conceptuales y Metodológicos	16
2.2.1	Inteligencia Artificial	16
2.2.2	Procesamiento Digital de Imágenes	22
2.2.3	Visión Computacional	28
2.2.4	Ambientes Virtuales	38
3	ESTRATEGIA METODOLÓGICA	41
3.1	Plataforma robótica y herramientas externas	41
3.1.1	OpenCV	41
3.1.2	Blender	42
3.1.3	Darwin-OP	42
3.2	Herramientas extras	43
3.2.1	Modelo de Referencia	44
3.2.2	La Base de Datos	44
3.2.3	Simulación y Ambiente Virtual	48
3.3	Método de Localización	50
3.3.1	Consideraciones Extras	60
3.4	Sistema	61
3.5	Experimentos y Validación	62
3.5.1	Validación de la métrica Φ	62
3.5.2	Validación de los filtros	62
3.5.3	Experimentos con localización por zonas	63
3.5.4	Experimentos con localización informada	63
4	RESULTADOS Y DISCUSIÓN	67
5	CONCLUSIONES	75
	REFERENCIAS	79

LISTA DE FIGURAS

Figura 1	Algoritmo de Hough: Cada pixel vota por una familia de líneas que pasan por ese punto, y corresponden a una senoide. Los puntos donde se registran más cruces corresponden a las líneas con mayor número de votos. 27
Figura 2	Poses desde las que se generaron las vistas sintéticas 44
Figura 3	45
Figura 4	Traslape de los segmentos del algoritmo de Hough Probabilístico 46
Figura 5	Para determinar si el segmento azul corresponde a la línea roja (que a su vez corresponde a un segmento analizado previamente), se extiende y se buscan dónde cruza con los bordes de la imagen. Se consideran la misma línea si las distancias δ_1 y δ_2 son menores a un valor preestablecido. 47
Figura 6	Ambiente virtual 49
Figura 7	Elementos de la cancha virtual: piso de alfombra verde, línea de cinta plástica blanca, y porterías de PVC. 50
Figura 8	(a - c) Diferentes modelos de vallas utilizados. (d) Suzanne 51
Figura 9	Ejemplo de fondos utilizados para probar el cambio de intensidad de la luz y problemática de líneas espurias. Nótese que no es relevante para los fines de el cambio de fondo, que las perspectivas coincidan. 52
Figura 10	Diagrama de la métrica Φ . Se sobreponen los puntos de control sobre la imagen binarizada, y se busca la distancia que hay entre cada uno y el pixel del borde más cercano, sin sobrepasar el tamaño de δ . Φ Es la razón que hay entre el complemento de la suma de las distancias entre el máximo alcanzable. 54
Figura 11	55
Figura 12	64
Figura 13	Diagrama de Bloques 65

- Figura 14 Las gráficas de ROC muestran la eficacia de un clasificador, al relacionar la razón que hay entre los *Falsos Positivos* ($1 - \text{Especificidad}$) y los *Verdaderos Positivos* (*Sensibilidad*), al ir variando los valores del umbral de aceptación. El *área bajo la curva* es una medida de precisión para el clasificador, mientras más cercano a 1, mejor su desempeño. 69
- Figura 15 72
- Figura 16 La figura (a) muestra la imagen de la cámara del robot. Al realizar la comparación con las vistas sintéticas, la métrica se confunde con las líneas de las vallas, asignando un valor mayor a la vista de la imagen (b) que a la de la imagen (c), que es la que corresponde realmente. 73
- Figura 17 Las figuras (a) y (b) muestran el tipo de oclusiones contra el que se enfrentó el algoritmo, siendo que la imagen (b) muestra una oclusión muy prominente, pero son pocos casos en que pasa eso. Las figuras (c) y (d) muestran los cambios de intensidad que se pueden presentar en las imágenes generadas por la simulación, al cambiar el fondo y por tanto la iluminación ambiental. 76

LISTA DE TABLAS

Tabla 1	Resultados de las pruebas de la métrica Φ . FP son los <i>Falsos Positivos</i> , VP son los <i>Verdaderos Positivos</i> . 67
Tabla 2	Resultados de las combinaciones de filtros. El lugar indica en que lugar de la lista ordenada quedaron en promedio las poses correspondientes; Δd es la distancia promedio entre la posición real y la encontrada; $\Delta\theta$ es la diferencia entre la orientación real y la orientación de la pose encontrada; los positivos indican la tasa de aciertos que tuvo el sistema. F: Filtro Frontal; L: Filtro Lateral; I: Filtro Inverso. 68
Tabla 3	Resultados de la búsqueda por zonas. El lugar indica en que lugar de la lista ordenada quedaron en promedio las poses correspondientes; Δd es la distancia promedio entre la posición real y la encontrada; $\Delta\theta$ es la diferencia entre la orientación real y la orientación de la pose encontrada; los positivos indican la tasa de aciertos que tuvo el sistema 70
Tabla 4	Resultados de la búsqueda informada. El radio corresponde al área al rededor de la pose candidata, de donde se toman las vistas a comparar; el lugar indica en que lugar de la lista ordenada quedaron en promedio las poses correspondientes; Δd es la distancia promedio entre la posición real y la encontrada; $\Delta\theta$ es la diferencia entre la orientación real y la orientación de la pose encontrada; los positivos indican la tasa de aciertos que tuvo el sistema. 71
Tabla 5	Resultados de las pruebas sin oclusiones en las imágenes de muestra. Los valores son muy similares a los obtenidos con la muestra original. El lugar indica en que lugar de la lista ordenada quedaron en promedio las poses correspondientes; Δd es la distancia promedio entre la posición real y la encontrada; $\Delta\theta$ es la diferencia entre la orientación real y la orientación de la pose encontrada; los positivos indican la tasa de aciertos que tuvo el sistema. F: Filtro Frontal; L: Filtro Lateral; I: Filtro Inverso. 71

INTRODUCCIÓN

Desde hace muchos años se ha jugado con la idea de crear entes que realicen tareas que para los seres humanos pudieran parecer muy monótonas, tediosas o hasta peligrosas. Estos entes originalmente se crearon bajo la idea de tener una especie de esclavos para los humanos, idea que surgió de la ciencia ficción, dígase obras de teatro y novelas, que han inspirado a los científicos a generar mucha investigación alrededor del tema. No obstante, este tipo de entes no son aquellos que nos vienen a la mente cuando pensamos en el término **ROBOT**, sino más bien máquinas disformes que únicamente repiten una y otra vez la misma tarea, tendiendo más bien a ser autómatas, como aquellos creados desde hace muchos años para trabajar en fábricas.

ROBOT

La idea que se ha perseguido durante mucho tiempo es la de tener robots que sean capaces de reaccionar ante cualquier estímulo que encuentren en su entorno, tal que sea posible ponerlos en un ambiente no controlado y que reaccionen de una manera similar a la que reaccionaría un ser humano, de tal forma que se les pueda confiar una tarea y olvidarse de ellos hasta que la terminen. Aquí es donde se involucra el campo de la inteligencia artificial, que es el que le brinda la toma de decisiones a los robots. Así, más allá de tener máquinas capaces de repetir una acción una y otra vez, estamos hablando de dotarle a estas máquinas la capacidad de reaccionar ante su entorno, y aún más, se busca que los robots sean capaces de tomar en cuenta los sucesos por los que han pasado, los datos que han recolectado y la información que van obteniendo durante el tiempo para así procesarlos y utilizarlos en un futuro. Podríamos decir así que el fin último de la robótica es el de crear máquinas que puedan reaccionar como lo haría un ente vivo racional. De hecho, muchas de las investigaciones y los avances que se han tenido en el área se han inspirado en procesos biológicos, simulándolos en las computadoras para obtener comportamientos como los que estamos acostumbrados a observar y llevar a cabo los seres humanos.

Dentro del campo de la robótica existe una iniciativa llamada **ROBOCUP**® (véase sección 2.1.3), que tiene 17 años llevándose a cabo y tiene como meta principal desarrollar las diferentes ramas de la robótica, pero de manera lúdica y con un toque de rivalidad, aunque con el objetivo de desarrollar el conocimiento en comunión. El evento principal de dicha competencia es el de soccer (**ROBOCUP SOCCER**®), que tiene varias categorías que dependen del tamaño y del tipo de habilidades que se desea desarrollar, pero el punto importante es que

ROBOCUP

se tienen robots que juegan fútbol, lo que implica tener un entorno conocido, un objetivo muy específico y bien entendido y reglas que cumplir, lo que permite desarrollar muchos de los aspectos de la robótica. En este caso, el fútbol es sólo un pretexto para utilizar un juego que permite la sana competencia y que le agrada a la gente; la ROBOCUP intenta ir más allá del simple deporte, al brindar los medios para juntar diversos grupos de investigación para lograr un objetivo común. En este sentido, al impulsar por ejemplo el caminado de los robots que juegan fútbol, es posible llevar los resultados obtenidos de esta competencia a muchas otras situaciones a las que un robot móvil cualquiera se podría enfrentar. Gracias a esto, nos acercamos cada vez más a la meta de crear los mencionados entes humanoides que lleven a cabo las tareas que les sean encomendadas para facilitar la vida de los seres humanos.

Cabe mencionar que la ROBOCUP busca dotarle a sus robots de características lo más cercanas a las humanas, y no más allá de ellas. Esto como consecuencia de que el objetivo principal de esta competencia es que para el año 2050 se haya generado un equipo de robot humanoides que puedan jugar contra el campeón mundial de ese entonces y se pueda dar una competencia pareja.

El Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas (IIMAS) de la UNAM, en conjunto con el Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADDET), cuenta con un equipo de robótica llamado DOTMX que participó en la ROBOCUP 2012 y en el Torneo Mexicano de Robótica 2012. En estas competencias se pudieron observar los puntos fuertes y débiles del equipo, así como los métodos que se utilizan tanto a nivel nacional como internacionalmente para controlar los diversos mecanismos de los robots, como localización, caminado, orientación, pateo, etc. Se pudo observar que uno de los puntos más débiles del DOTMX es que utiliza estrategias de orientación basadas en la ubicación de elementos sobresalientes de la cancha apoyándose en segmentación de colores pero no cuenta con un método de auto-localización. Éste trabajo se llevó a cabo con la intención de desarrollar un método de localización que pueda ser utilizado por el DOTMX, en los robots que compiten en la categoría Kid Size de la ROBOCUP Soccer, en particular para robots DARWIN-OP (véase la sección 3.1), y aun cuando se busca que se pueda emplear en cualquier otro tipo de robots móviles que se encuentren en circunstancias similares, sí se buscó que dadas las características y limitantes de ese modelo de robot, el método consumiera pocos recursos tanto de memoria como de procesamiento, que tuviera resultados precisos y que fuera rápido, teniendo en mente la posibilidad de ajustarlo para ser utilizado en tiempo real, aunque en principio está pensado para que la localización se lleve a cabo únicamente en momentos clave de una confrontación y que se integre sin obstaculizar los demás procesos (por ejemplo la localización de la pelota y las porterías).

El contar con un método de localización preciso es necesario para poder tomar decisiones adecuadas o crear estrategias avanzadas en cualquier robot móvil, ya sea que tenga un comportamiento preprogramado o que se trabaje con técnicas de aprendizaje de máquinas donde el comportamiento se va adecuando de acuerdo con la información que va siendo recolectada. En particular dentro de la competencia de ROBOCUP existen penalizaciones cuando los elementos de un equipo no están lo suficientemente automatizados, en el sentido de que cuando empieza el juego, haya cambio de cancha o alguien meta gol, los robots deben regresar a su lado de la cancha en zonas específicas, y aunque todavía se permite que una persona acomode a los robots manualmente, se aplican penalizaciones que pueden ser de tiempo o que obligan a situar al robot en una posición de desventaja.

Éste trabajo se desarrolló en su etapa inicial en el Laboratorio de Visión Computacional de la Universidad de Bristol, bajo la supervisión del Dr. Walterio Mayol Cuevas, como parte de una estancia de investigación acordada entre el IIMAS, el CONACYT y la Universidad de Bristol. En dicho laboratorio tienen gran experiencia en métodos visuales para resolver problemas como localización, mapeo, rastreo de cámara, segmentación, caracterización de la imagen, entre otras cosas. Además es ahí donde se desarrollaron varios de los métodos y artículos a los que se hace referencia en este trabajo. lo cual fue de gran ayuda a la hora de darle forma a la problemática y encontrar la solución aquí propuesta.

1.1 LOCALIZACIÓN ROBÓTICA

La localización en el campo de la robótica es un problema popular al que se le conoce como SLAM por sus siglas en inglés que significan *Localización y Mapeo Simultáneos*. El SLAM, en pocas palabras, consiste en que un robot estime su posición relativa a su entorno al mismo tiempo que genera un mapa de éste. Un robot podría llegar a mapear con gran precisión su entorno aprovechando los sensores que tiene a su disposición, y en función de esto es que puede llegar a tomar decisiones procedimentales. En el presente trabajo, no se hará mucha referencia al mapeo del entorno; está enfocado en la localización del robot, bajo el supuesto de que ya se conoce el entorno (la cancha de fútbol) y se cuenta con un mapa o modelo preciso del mismo, y utilizaremos esa información para ubicar en dónde se encuentra el robot en un momento dado. A esto se le conoce como el *problema de la relocalización*, que se concentra en la habilidad del robot de localizarse desde el momento en que aparece en el entorno o en los momentos en que pierda el rastro de dónde se encuentra. De aquí se deriva otro de los grandes problemas de la robótica que es el llamado problema del secuestro, en el cual un robot que se encuentra bien localizado, es transportado a otro lugar del entorno sin ser avisado, y debe poder

SLAM

relocalizarse inmediatamente. La diferencia entre éste y el problema global de la localización es que en este caso, el robot debe tener la certidumbre de encontrarse en una posición determinada antes de ser secuestrado.

1.2 NUESTRO MÉTODO

La mayoría de los métodos de localización se ayudan de otros métodos de reconocimiento de elementos del entorno. Por ejemplo, algunos como el de Betke [10], comparan lo que el robot está percibiendo con sus sensores, y utilizan modelos matemáticos para llevar a cabo una triangulación que les permite obtener la posición desde donde se tomó cada observación. Otros métodos pueden llevar a cabo una segmentación de los elementos de la imagen basada en color y así reconocer elementos como las porterías, la pelota o hasta el área de la cancha [8]. Muchos de los métodos de segmentación e indexado que se usan en la ROBOCUP (y otras situaciones parecidas) para localización sufren problemas con los cambios de luz, pues de entrada, las diferentes condiciones entre los escenarios de las competencias provocan que sea necesario calibrar los sistemas de visión antes de cada competencia y aún más, antes de cada partido; sin embargo, aún cuando la calibración es adecuada, se realiza cuando no hay espectadores alrededor de la cancha, y en cuanto inicia el partido, la presencia de éstos provoca cambios significativos en la iluminación percibida por los sensores de los robots. La otra situación que presenta problemas a estos algoritmos de localización es que durante el juego, los demás robots en la cancha generan oclusiones a la visión del robot, por lo que las imágenes que percibe son distintas a las que se esperan cuando la cancha está vacía. Nuestro método se desarrolló con la idea de hacerlo robusto ante cambios de luz y oclusiones.

El procedimiento que se siguió para desarrollar este método comenzó por observar cómo es que los seres humanos se localizan en un entorno, particularmente en el caso de un jugador que se localiza en una cancha de fútbol, pensando que una vez que entra en el campo, le basta con observar y crear un modelo abstracto mental, y así en un dado momento que quiera saber donde está, es suficiente observar en alguna dirección distinguible del campo y con eso logra recordar dónde se encuentran su portería y la contraria, o los elementos que tiene a sus lados o detrás de él, esto en función de lo que está viendo y sin necesidad de realizar un reconocimiento total cada vez.

Teorizando un poco cómo pareciera funcionar el cerebro humano a la hora de localizarse, el método está apoyado en una de las grandes áreas de las ciencias de la computación, que es la visión computacional, pues todo indica que la localización humana se basa principalmente en el ámbito visual. Además de esto, el método se basa en el uso de ambientes virtuales, lo que permite generar diferentes escena-

rios con los elementos adecuados para el campo de interés. Como el modelo está enfocado en robots humanoides que jueguen fútbol, se puede limitar el ambiente a una cancha virtual de fútbol en la que se pueda controlar variables como la intensidad de la luz, o lo que se encuentra en el fondo del escenario; particularmente se hace uso de los elementos invariables del ambiente, como lo son las líneas de la cancha, dado que se sabe que no van a modificar sus dimensiones ni su disposición sin importar dónde se lleve a cabo un encuentro. También se toman en cuenta ciertas reglas de la competencia para mejorar el método, aprovechando ciertas características del juego que se conocen de antemano, que dictan dónde puede o no puede encontrarse un robot dentro de la cancha, o que cuando se va a iniciar el juego cada jugador debe estar en una zona específica dependiendo de su posición en el equipo y si ataca o defiende.

Una vez ideado el método, se desarrolló un sistema que permitiera probarlo, y se implementó con la intención de que se pueda integrar al propio sistema del robot lo más directamente posible, empleando en el sistema una arquitectura y una configuración similares, además de que se buscó que fuera compatible con las estrategias con las que se contaba anteriormente.

Debido a circunstancias imprevistas, durante el año 2013 y particularmente durante el desarrollo de este trabajo, no se contó con robots disponibles para participar en las competencias que hubo a lo largo del año, ni tampoco para realizar las pruebas del sistema. Sin embargo, esta situación se solucionó desarrollando una metodología que no hiciera uso de un robot real o de una cancha reglamentaria, pero que pudiera ser integrado sin complicaciones al sistema de los robots una vez que estuvieran listos. Otra ventaja de desarrollar una simulación es que de esta manera se evita el desgaste que sufren los robots causado por el uso común.

Se observó que era necesario un entorno que se pudiera controlar en cualquier aspecto para emular las diversas situaciones a las que podría enfrentarse el método, tales como el número de fuentes de luz, la intensidad de las mismas, elementos que generaran oclusiones, cambios en los elementos que rodean la cancha y que no están definidos en el reglamento, como las vallas de contención, los elementos del techo o las paredes, o los mismos espectadores, y en general cosas que pudieran afectar la visión del robot y por consecuencia al método. Dicha simulación aprovecha los avances de la computación gráfica que permiten generar escenas casi indistinguibles de un entorno real. Conociendo bien las características de ambiente, es posible generar *vistas sintéticas* de una cancha genérica, para así tener una idea de lo que espera ver el robot desde una posición dada. Para esto utilizamos otra simulación del ambiente cuyos elementos son únicamente los elementos invariantes de la cancha.

En lo que resta de esta tesis, hablaremos de *vistas sintéticas* como

aquellas imágenes tomadas de la simulación que sólo contiene los contornos de los *elementos invariantes* de la cancha de fútbol, que son tanto la líneas del campo, como las porterías, y se denominan invariantes porque sin importar en qué lugar se lleve a cabo un partido de fútbol robótico o bajo qué condiciones, siempre y cuando la cancha sea reglamentaria, estos elementos no cambian ni de posición ni de forma.

1.3 OBJETIVOS

El objeto de estudio de este trabajo son los robots, y en particular el problema de la localización de los mismos, que abarca casi cualquier tipo de robot móvil, en especial los robots humanoides. Para esto primero se analizarán algoritmos y estrategias de la visión computacional y otras disciplinas que forman la base de esta rama de estudio. Este trabajo está fundamentado en algunos métodos de localización basada en modelos, y por tal motivo es que se construyó el modelo de una cancha de fútbol robótico. También se desarrolló una simulación del sistema visual del robot, la cual además de tener la capacidad de emular el tipo de cámaras que utilizan los robots, intenta emular un ambiente real, por lo cual se tomó como base el modelo de la cancha y se le agregaron escenarios, luces, materiales, texturas y otros elementos que se presentan en situaciones reales de juego, con lo que al final se cuenta con dos herramientas para generar imágenes sintéticas, tanto de la cancha con un modelo de únicamente elementos invariantes, como de la cancha en una situación suficientemente cercana a la realidad, de tal forma que una imagen renderizada con esta herramienta se asemeja lo suficiente a la imagen tomada por la cámara del robot.

El objetivo de este trabajo es desarrollar un método de localización para robots humanoides que sea eficaz, pero al mismo tiempo tenga un bajo consumo de recursos puesto que los robots a los que está enfocado no cuentan con un procesador poderoso ni mucha memoria, y que además debe poder aplicarse a robots que únicamente cuenten con sensores que emulen el cuerpo humano, aunque en este caso en particular el trabajo está limitado a la visión. Dicho método presenta una alternativa a otros métodos de localización que cuentan con conocimiento previo del entorno, así como aquellos que se apoyan en técnicas de regresión sobre alguna base de datos.

También se quieren encontrar resultados extensibles, en el sentido de que este método de localización busca no sólo ubicar un robot en una cancha de fútbol, sino localizar un robot cualquiera que se encuentre en cualquier entorno del que se cuenta con un modelo o mapa conocido de antemano.

Lo que resta de este trabajo está organizado de la siguiente forma: en el **Capítulo 2** se realiza una reseña de como de cómo es que surge la problemática a la que nos enfrentamos, así como una descripción

de otros trabajos del estado del arte actual. En el **Capítulo 3** se señala cuál fue la metodología que se empleó para enfrentar este problema, tanto para desarrollar el método como para el desarrollo del sistema que sirvió para probarlo. En el **Capítulo 4** se presentan los resultados obtenidos de los experimentos realizados por el sistema, y se lleva a cabo una discusión de los mismos. Por último, en el **Capítulo 5** se presentan las conclusiones del proyecto.

MARCO TEÓRICO

2.1 ANTECEDENTES HISTÓRICO - CONTEXTUALES

En esta sección vamos a hablar del área del conocimiento en donde se puede ubicar el problema, así como de la rama específica a la que pertenece, y de esta forma explicaremos tanto su origen como las circunstancias que lo condicionan. La finalidad es esclarecer la importancia que tiene el objeto de estudio de este trabajo y por tanto la resolución al problema que le corresponde.

2.1.1 *Robótica*

Dado que el objeto de estudio de este trabajo son los robots, es importante remarcar la principal área del conocimiento que le corresponde, que en este caso es la robótica, que es la ciencia y la tecnología que está detrás de los robots, y para esto lo primero que hay que comprender es el término robot. La palabra robot viene de una obra de teatro que escribió un escritor checo de nombre Karel Capek, quien escribió una crítica de la sociedad en la que describía unos seres creados por los humanos que sólo se dedicaban a realizar una y otra vez tareas impuestas por sus amos. Aun cuando los robots de Capek eran creados químicamente, la idea fue la que trascendió más allá de la obra. El término ROBOT surge de las palabras checas *robota*, que significa trabajo, y *robotnick* que significa sirviente. Cuando Capek pensaba en qué nombre les podía dar a sus seres, pensó en la palabra *autónomo*, que significa autónomo, pero le parecía que era una palabra muy intelectual, así que pidiéndole consejo a su hermano Josef, éste le sugirió la palabra Robot [60]. Finalmente fue el término que utilizó para su obra *Rossum's Universal Robots (R.U.R.)*, la cual trataba de personas creadas artificialmente en una fábrica, para que trabajaran para los humanos, aunque como muchas de las historias que tienen que ver con robots, estos se rebelan al final contra sus amos, pues al serles conferida la capacidad de razonar, deciden que los conflictos bélicos han sido la principal causa del progreso de la humanidad[54].

A partir de la puesta en escena de esta obra, el término robot se empezó a utilizar despectivamente para describir a aquellas personas que realizaban trabajos repetitivos [39]. Pero al mismo tiempo se empezó a desarrollar la idea de los robots de una forma más técnica, aún cuando los medios para crear una máquina electrónica capaz de realizar tareas, no existieron durante muchos años. Sin embargo la idea ya recorría la mente de los científicos y fanáticos de la ciencia ficción,

RUR

ISAAC ASIMOV y destaca el caso de *Isaac Asimov*, un científico y escritor ruso que desde sus años en la universidad empezó a escribir historias cortas de ciencia ficción y en una de éstas llamada *Runaround*, fue que nació el término *Robótica*. Un compendio de estas historias fue publicado como libro, *'I, Robot'*, uno de los muchos de ciencia ficción que escribió durante su vida, además de libros de historia y textos divulgativos. En esta serie, Asimov describe las tres leyes de la robótica, que son

ROBÓTICA

LEY UNO Un robot no puede lastimar a un ser humano o, mediante su falta de acción, permitir que un ser humano sufra algún daño, a menos que esto viole una ley de mayor orden.

LEY DOS Un robot debe obedecer las órdenes dadas por un ser humano, salvo que dichas ordenes estén en conflicto con alguna ley de mayor orden.

LEY TRES Un robot debe proteger su propia existencia, siempre y cuando dicha protección no entre en conflicto con ninguna ley de mayor orden.

A estas leyes más tarde les agregé la ley cero, la cual tiene el orden más grande:

LEY CERO Un robot no puede perjudicar a la humanidad o, mediante su falta de acción, permitir que ésta sufra algún daño.

Los escritos de Asimov son trabajos muy interesantes de lo que se puede llamar *la psicología robótica*, que desataron una muy trascendente discusión acerca de lo que podría pasar a partir del momento en que tengamos robots capaces de pensar de forma independiente.

Cuando se habla de los robots modernos, se piensa en realidad en las máquinas que se crearon en los años sesenta, las cuales podían repetir una y otra vez una tarea sin necesidad de supervisión. Inicialmente fue en la industria en la que se utilizó este tipo de máquinas, pero poco a poco se ha extendido su uso a otras áreas, como el ámbito doméstico, médico o educativo, en dónde el uso de los robots permite mejorar el desempeño general del trabajo en dichas áreas, sin contar todas las aplicaciones desarrolladas en el ámbito militar. La definición de robot según la RAE¹, es

Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas.

Sin embargo, esa es una definición que no toma en cuenta aspectos como razonamiento, inteligencia, resolución de problemas, percepción de su ambiente o aprendizaje. Una definición más adecuada podría ser la que da Ron Arkin, que lo define como

¹ Real Academia Española

Una máquina capaz de extraer información de su ambiente y utilizar el conocimiento acerca de su mundo para actuar con seguridad de forma significativa y con propósito.

Podemos pues definir la robótica como la ciencia que estudia estas máquinas.

Los primeros avances en esta área se dieron a finales de la Segunda Guerra Mundial, cuando se buscaba encontrar una manera de manipular materiales radiactivos de una forma más segura, con lo cual se desarrollaron manipuladores a control remoto que intentaban reproducir la forma y el movimiento de un brazo humano. Posteriormente se buscó dotarlos con un sistema de retroalimentación que le permitiera al operador sentir las fuerzas y resistencias a las que se enfrentaba la máquina. De ahí se empezó a trabajar en máquinas autónomas que pudieran realizar operaciones repetitivas, y fue en los años 50 que se crearon los primeros dispositivos articulados programables. Esto en conjunción con la idea anterior llevó al desarrollo de máquinas programables capaces de 'sentir' su entorno y usar esa información para realizar tareas automáticas sin la necesidad de un operador. Muchos han sido los avances que se han dado a partir de ese momento y hoy la robótica involucra muchas áreas de investigación como lo pueden ser la cinemática, la dinámica, sistemas de planeación, control, y sensores [23]. Además involucra muchas otras grandes disciplinas como lo son las Ciencias de la Computación, la Inteligencia Artificial, la Física, las Matemáticas, entre muchas otras.

Un robot está compuesto básicamente por una estructura de soporte, sensores, efectores, actuadores, y unidad de control. La diferencia entre los efectores y los actuadores es que los primeros tienen un impacto en el ambiente del robot, como lo pueden ser las piernas, los brazos o las llantas, y los actuadores que son los que le permiten al robot realizar diversas acciones, como son los motores.

Los robots se pueden dividir en aquellos que son teleoperados, y aquellos que son completamente autónomos, pero además se pueden clasificar de acuerdo a cómo son construidos, de la siguiente forma:

- Robots con Ruedas
- Vehículos Autónomos, que comprenden vehículos no tripulados terrestres, aéreos y marítimos
- Robots Antropomórficos
- Robots Animaloides
- Robots Humanoides

siendo estos últimos los de mayor interés en el presente trabajo.

2.1.2 *Robots Humanoides*

Dada la naturaleza de este trabajo, se enfocará en los robots humanoides, sin embargo se busca que el método de localización desarrollado se pueda extender sin mucha complicación a la mayoría de los robots móviles. Los robots humanoides son aquellos que asemejan el cuerpo humano; además, se busca que sean robots que perciban el ambiente, procesen la información y actúen de la forma en que actuaría un ser humano. Surgen de la idea de estudiar ciertos aspectos del ser humano, por lo que comprenden un área de investigación que apoya y es apoyada por diversas disciplinas. Para que pueda existir un robot humanoide se necesita conocer bien cómo es que se mueve un ser humano, cómo es que piensa, escucha o ve, entre otros aspectos para los que es necesario desarrollar mucha investigación que luego pueda aplicarse a estos robots; pero de manera similar, tener una herramienta como un robot humanoide permite a todas esas áreas probar ciertas hipótesis o trabajar en ciertas teorías y ampliar las fronteras del conocimiento.

Una de las ventajas de contar con robots humanoides es que el entorno al que estamos acostumbrados es un entorno que está diseñado para los seres humanos, así como las herramientas y objetos que utilizamos están diseñadas ergonómicamente; por lo tanto, la manera más sencilla de que un robot se pueda integrar a este ambiente, es haciendo que estos artefactos tengan dimensiones, motricidad y dinamicidad muy similares a las de los seres humanos. Así, podrían realizar trabajos y utilizar herramientas que están pensados para un ser humano, con lo cual se puede generar una convivencia *humano-robótica* en un lugar de trabajo, o más aún, confiarle a los robots la mayoría de las tareas tediosas, peligrosas o dañinas para los humanos.

2.1.3 *RoboCup*

Una de las preguntas que surgen es ¿por qué crear un método para localizar un robot que juega fútbol específicamente? Y para poder responderla es necesario entender los orígenes y la intención real que hay detrás de la RoboCup. Desde hace muchos años se han utilizado plataformas de investigación para impulsar diversas áreas del conocimiento. La AAI² ha lanzado por muchos años competencias para robots autónomos, cambiando las reglas y metas cada año, aunque con la desventaja de que es para robots individuales. En el año 1992 se llevó a cabo un congreso en Tokio, Japón, en donde se discutieron cuáles eran las opciones de *Grand Challenges Projects* que debían ser abordados por la comunidad de investigadores, haciendo referencia a aquellos problemas cuya solución trae consigo un gran impacto social

² Association for the Advancement of Artificial Intelligence

[30]; ahí *Kitano et. al.* [31] propusieron como uno de estos grandes retos, el desarrollar robots capaces de jugar fútbol, argumentando que éste era al mismo tiempo un *Problema Estándar*, o sea aquél que permite a los investigadores comparar técnicas computacionales, identificar problemas y examinar propuestas, así como evaluar teorías o algoritmos desde diferentes perspectivas, pero resolviendo la misma problemática. Un ejemplo de este tipo de problemas es el del ajedrez, en donde se buscaba crear una programa computacional que pudiera vencer a un humano experto en el juego. Aquí la diferencia es que un juego de fútbol es mucho más dinámico, los robots deben reaccionar de forma inmediata y continua a los cambios no determinísticos que se les impongan; tampoco cuentan con la información completa de cómo está la configuración del juego ni de nadie excepto ellos mismos y para lograr desarrollar un equipo de robots que jueguen fútbol hace falta desarrollar algoritmos de cómputo distribuido.

PROBLEMA
ESTÁNDAR

Además de un problema estándar, la RoboCup es catalogada como un *Landmark Project*, que son problemas que se plantean de forma abierta, fijando una meta muy complicada, pero que se pueda alcanzar. En este caso la meta particular está planteada de la siguiente manera:

LANDMARK
PROJECT

Para mediados del siglo 21, un equipo de robots humanoides autónomos deberá vencer al equipo humano campeón de la COPA MUNDIAL DE FUTBOL bajo las regulaciones oficiales de la FIFA [4].

Para que un proyecto sea catalogado en esta categoría debe ser público, de tal forma que los avances y logros que se tengan estén abiertos y que sean de interés; que sea lo suficientemente complicado para que se necesiten muchas innovaciones para que se pueda llevar a cabo; que sea factible y que se pueda definir un paso inicial en el proyecto; y por último, una vez que se logre la meta, el tipo de tecnologías desarrolladas fundamenten a la industria de la siguiente generación. Un ejemplo es el programa Apollo, lanzado en EEUU, en donde la meta era aterrizar un hombre en la luna y lograr traerlo de regreso a la tierra sano y salvo [31]. Una de las ventajas de este tipo de proyectos es que logra la colaboración de muchos grupos de investigación en cuanto a que periódicamente se presentan los resultados que se van consiguiendo y además se comparan los diferentes métodos unos contra otros en un ambiente de competencia.

La propuesta original de la RoboCup era crear una competencia que se dividiera en tres ligas: La *Real Robot League*, en la que robots físicos se enfrentan en competencias que ponen a prueba los diseños y el control que se tiene sobre ellos; la *Software Robot League*, la cual pone a prueba los algoritmos que se emplean en los robots sin darle importancia ni dependencia al hardware; y la *Expert Robot Competition* en la que se llevaban a cabo pequeñas competencias de habilidades

muy específicas por separado, como el caminado o el pateo, y que una vez comprobada su eficacia se pudieran integrar en un robot.

Bajo la idea de crear un equipo de robots que jueguen fútbol propiamente, la RoboCup permite desarrollar investigación en muchas áreas como la fusión de los sensores en tiempo real, el comportamiento reactivo, la adquisición de estrategias, el aprendizaje de máquina, planeamiento en tiempo real, sistemas multiagentes, reconocimiento de contextos, visión computacional, toma de decisiones estratégicas, control motriz, control de robots inteligentes y, por supuesto, localización robótica. Sin embargo, había ciertas áreas que no se tomaban en cuenta dadas las limitantes que supone el juego de fútbol, y es por eso que surgieron las categorías que hay actualmente, que complementan a la RoboCup Soccer, que son [31]:

- RoboCup Rescue son competencias para robots que se ponen en ambientes que emulan zonas de desastre, que el robot no conoce de entrada y que enfrenta a situaciones imprevistas.
- RoboCup Home competencias para robots dedicados a la convivencia con los humanos, realizando pruebas dentro de un ambiente controlado como lo es un hogar, y la idea es desarrollar robots que apoyan a la gente.
- RoboCup Junior cuyo objetivo es desarrollar en la nuevas generaciones el gusto por la robótica, y que consiste en competencias simples en escalas pequeñas que permitan a jóvenes entusiastas iniciarse en el mundo de la robótica.

Otra de las ventajas que presenta la iniciativa de la RoboCup es que presenta un problema de robótica multiagente. Antes de la RoboCup se intentaron proponer esquemas de planeación cooperativa reactiva en espacios de problemas dinámicos utilizando juegos, como [9, 40], cuyo objetivo resultó muy simple o bien el ambiente en que se desarrollan se ajusta mejor al estudio de agentes individuales. Por su lado, la RoboCup presenta un escenario en que se tiene una meta global que consiste en conseguir que se lleve a cabo un partido de fútbol, y al mismo tiempo cada equipo tiene una meta propia que consiste en anotar la mayoría de los goles y ganar cada partido, y aún más, cada jugador tiene una submeta que depende del rol que juega cada jugador dentro de un equipo.

Los retos importantes que presenta la RoboCup es que presenta un ambiente dinámico, que cambia tanto por los robots del mismo equipo, como por los del equipo contrario, además de que la pelota va a estar moviéndose todo el tiempo sin que haya manera de predecir su posición; por otro lado está la percepción limitada de los robots, pues éstos solo conocen lo que logran percibir mediante sus propios sensores, que son aquellos que emulan los sensores humanos, como las cámaras y los micrófonos; otro reto que se debe superar es que

cada robot tiene un rol diferente que jugar, por lo que se deben desarrollar diferentes estrategias dependiendo de estos roles; por último, los robots que compiten en la RoboCup no tienen permitido comunicarse con una computadora central o con otros robots mediante una red inalámbrica o algo por el estilo, sino que la comunicación entre los robots sólo puede darse de forma similar a la que se comunicarían dos personas, ya sea mediante señas o sonidos, además de con el referee.

En este aspecto, en las primeras competencias el papel de referee era asignado únicamente a un ser humano, pero poco a poco se busca eliminar la figura del referee humano y cambiarla por una computadora centralizada llamada RefereeBox que se encargue de todo. En la última competencia llevada a cabo en junio de 2013, el referee era un humano que simultáneamente manejaba el RefereeBox para enviarle las señales necesarias a los jugadores, como cuando empieza el juego, se hace una pausa, o alguien mete gol. Se incentiva el uso del RefereeBox en lugar de un acomodador humano penalizando a los robots que no interactúen con éste, quitándoles tiempo de acción o situándolos en posiciones de desventaja.

Para poder participar en la RoboCup, un equipo debe enfrentarse con ciertos desafíos clave. El primero es que se necesita tener el diseño y el control de los robots que va a utilizar; es posible utilizar, como en nuestro caso, diseños o plataformas de acceso libre, siempre y cuando se presenten avances anuales sobre el diseño y el control. Por otro lado los robots deben tener comportamientos individuales dependientes de la posición que jueguen en el equipo, así como los comportamientos generales que necesita cada robot, pero más allá de eso, es necesario desarrollar para los robots de cada equipo un comportamiento social, pues para poder llevar a cabo su meta, es necesario que se comuniquen e interactúen de una forma novedosa. Otro reto que se presenta es la fusión de los sensores, puesto que un robot que se encuentre en otro tipo de situaciones hace uso de diferentes tipos de sensores además de la visión, pero el tiempo y tipo de procesamiento que toma utilizarlos es demasiado si se quieren incorporar en un ambiente reactivo como lo es un juego de fútbol, por lo que es necesario fusionar los sensores de tal forma que se utilicen óptimamente, tal cual si fueran un sólo sistema; por último, y el más importante en nuestro caso, está el reto de la visión, pues es necesario manejar problemas de visión computacional grandes, como lo son el reconocimiento de objetos, el mapeo, el reconocimiento del ambiente, la localización, etc., y actualmente, la cámara es el sensor en que más se apoyan estos robots.

Más allá de los retos iniciales a los que se debe enfrentar cualquier equipo que ingrese a la RoboCup, la reglamentación está ideada para que cambie progresivamente de forma que poco a poco todos los participantes sean capaces de seguir los avances que se van alcan-

zando. Aunque como objetivo final se tiene el que se juegue con las reglas oficiales de la FIFA, de inicio las reglas tuvieron que limitarse para poder adaptarlas a las tecnologías a las que se tenía acceso. Así, las dimensiones de la cancha y el número de jugadores fueron reducidos; poco a poco el tamaño, la forma y el peso de los robots deben asemejar más a la forma y proporción del cuerpo humano, al igual que el movimiento. Cada año, después de la competencia principal, se reúnen los comités técnicos de cada liga con investigadores y algunos de los representantes de los equipos más destacados, para discutir y revisar el reglamento, modificando aquellas reglas cuyas limitantes se vayan resolviendo, y se anuncia con anticipación de 1 o 2 años, aquellos cambios que se van a hacer en el reglamento oficial, de tal forma que todos los equipos tengan el tiempo suficiente para resolver los nuevos retos. Además, se traza el rumbo que tomará la competencia en los próximos 5 a 10 años.

2.2 ANTECEDENTES CONCEPTUALES Y METODOLÓGICOS

En esta sección vamos a hablar de la teoría que se encuentra alrededor de nuestro método, así como de los enfoques del estado del arte que han permitido resolver los problemas anteriores al nuestro. Abarcamos las grandes ramas de la computación que dan fundamento a nuestra hipótesis y los métodos y herramientas de que hacemos uso para desarrollarla.

2.2.1 *Inteligencia Artificial*

La Inteligencia Artificial es una disciplina cuya meta final es crear entidades inteligentes y comprenderlas, que no necesariamente artefactos sino sistemas inteligentes que nos permitan comprender cómo es que pensamos y cómo es que podríamos reproducir la forma en que pensamos, preguntas que se han hecho los seres humanos desde hace muchos años. Ésta es una disciplina que interactúa fácilmente con otras, puesto que la investigación en esos otros campos está encauzada desde el punto de vista de la comprensión humana. A lo largo de los años la Inteligencia Artificial ha tenido varias veces el foco de atención al dar soluciones y enfoques creativos a problemas abiertos de la ciencia.

Sin embargo, aun cuando es un área muy universal, es muy complicado asignarle una definición concreta, puesto que el mismo concepto de *inteligencia* es difícil de definir. Muchas son las definiciones que se le han dado a la inteligencia artificial y en su mayoría han sido ampliamente investigadas. Aún así, es posible separarlas en dos categorías, las que creen que la inteligencia artificial tiene que ver estrictamente con los procesos del pensamiento y el razonamiento, y las que piensan que más bien está definida por el comportamiento.

Al mismo tiempo, se puede realizar una división independiente de aquellas definiciones que se enfocan en el desempeño y actuar humano, y las que ven la inteligencia como un concepto independiente. Así, tenemos finalmente cuatro grandes categorías en que se pueden agrupar las definiciones de inteligencia artificial [44]:

- IA como comportamiento similar al humano: Se basan en el enfoque que propuso Alan Turing al definir un sistema inteligente mediante una prueba conocida como *El Test de Turing*, el cual consistía en poner a un humano en una sala a conversar con el sistema de forma remota, tal que sólo se pudieran comunicar de forma escrita. Si al final de la conversación la persona no podía decidir si su interlocutor era una máquina o una persona, entonces podía considerarse al sistema como *inteligente*.
- IA como proceso de pensamiento humano: Es un enfoque de *modelado cognitivo*, cuyo objeto de estudio es la mente humana y tiene dos grandes corrientes de trabajo, la que se basa en la introspección que estudia la mente propia y genera teorías y ensayos acerca de cómo es que funciona, y la otra es mediante experimentos psicológicos y en base a éstos genera teorías. De hecho, existe una rama muy grande del conocimiento que se llama *Ciencia Cognitiva*, la cual es una ciencia transdisciplinaria que tiene como objetivo estudiar los procesos de aprendizaje de sistemas pensantes, ya sean naturales o artificiales y las principales disciplinas que la conforman son la Psicología, la Inteligencia Artificial, las Neurociencias, la Filosofía y la Lingüística.
- IA como proceso del pensamiento racional: Este enfoque se conoce como *Las Leyes del Pensamiento*, y se centra en la *Lógica Matemática*, pues es una herramienta precisa para poder definir pensamientos, eventos y acciones del mundo real en forma de enunciados, y además brinda un conjunto de reglas que sirven para relacionarlos. Esta disciplina inició desde la antigua Grecia con la lógica aristotélica, más bien como parte de la Filosofía. Con el surgimiento de las lógicas formales y posteriormente la invención de las computadoras, grandes esfuerzos se han llevado a cabo para implementar las leyes del pensamiento en programas capaces de razonar a través de ellas. Sin embargo, aún cuando es fácil definir enunciados simples, surge el problema de modelar de la misma manera el conocimiento informal, pues al no ser estructurado, se ha encontrado que es muy difícil convertir cualquier idea en términos formales; además, la lógica supone la sencillez de llevar a la práctica las soluciones que encuentra, mas en el mundo real no sucede de esa forma.
- IA como comportamiento racional: Bajo este enfoque, se ve al actuar racional como alcanzar una meta siguiendo las creencias

que se tienen. Así, un agente se considera algo que puede percibir su entorno de alguna forma, y que pueda actuar en consecuencia. Las definiciones de esta categoría ven a la inteligencia artificial como el estudio y la construcción de *agentes racionales*. A diferencia del enfoque de las leyes del pensamiento en que se busca bajo todo medio generar inferencias correctas, un agente racional sólo se apoya en un fundamento lógico detrás de cada acción que toma; sin embargo frecuentemente no existe una acción correcta que tomar, pero siempre se necesita tomar una decisión; por lo tanto, actuar racionalmente no siempre involucra realizar inferencias estrictamente correctas. La ventaja que tiene este enfoque sobre los demás, es que es más sensible a emplearse en un ámbito científico, pues aquellos enfoques que tienen como meta alcanzar las capacidades del ser humano, aún cuando este es producto de un proceso evolutivo muy avanzado, no se pueden definir como el estado definitivo y perfecto de la inteligencia. Por otro lado, el intentar imitar el pensar humano implica una demanda de recursos computacionales muy alta como para ser satisfecha hoy en día. Es de hecho éste el enfoque más utilizado para el desarrollo de aplicaciones de inteligencia artificial y especialmente en robótica, y también es el enfoque con el que desarrollamos este trabajo, y el tipo de agentes que nos competen son los robots.

Aún cuando no es posible dar una única definición de lo que es la IA, lo que sí es posible es caracterizar un comportamiento que se pueda considerar inteligente, como aquel que tiene la capacidad de extraer información importante de un entorno que está lleno de detalles e información irrelevante. También debe tener la capacidad de aprender de ejemplos y práctica, y generalizar el conocimiento adquirido para aplicarlo al enfrentarse a situaciones bajo nuevas circunstancias. Al mismo tiempo debe poder inferir hechos importantes aun presentándosele información incompleta. Y el último paso para considerar un comportamiento realmente inteligente es que debe poder generar sus propias metas y de la misma forma ser capaz de diseñar planes que lo lleven a alcanzarlas. Alcanzar estas capacidades es la meta de la investigación en Inteligencia Artificial.

La Inteligencia Artificial tiene contribuciones de muchas otras áreas del conocimiento. Su base original se encuentra en los orígenes de la filosofía, cuando algunos se preguntaban cómo funcionaba la mente humana; ya desde entonces se catalogó como una máquina que operaba sobre el conocimiento codificado de una forma que sólo ella conocía, y se vio que el pensamiento es el medio para tomar las decisiones correctas. Por otro lado, los matemáticos fueron los que desarrollaron las herramientas para manipular los enunciados lógicos de los que se tiene certidumbre, pero también para trabajar en situaciones donde está presente la incertidumbre, utilizando métodos probabilísticos

para estudiar dichas situaciones. También fue desde las matemáticas que se desarrollaron los fundamentos del razonamiento algorítmico, de donde surgieron las Ciencias de la Computación. Aún los psicólogos contribuyen a esta disciplina, pues se han dedicado a estudiar el pensamiento humano, y muestran cómo se puede ver a los humanos y otros animales como máquinas procesadoras de información. Por su parte, ha sido la Ingeniería de la Computación la encargada de diseñar y desarrollar las herramientas y artefactos que hacen posibles las aplicaciones de la Inteligencia Artificial.

2.2.1.1 *Toma de decisiones*

Un problema básico en robótica es la planeación de movimientos para resolver una tarea específica, y después controlar el robot para que ejecute los comandos necesarios para llevar a cabo esas acciones. En este contexto, planeación significa decidir sobre el curso de acción antes de actuar. Un plan es una representación del curso de acción para alcanzar una meta. Un robot planificador intentará encontrar un camino de un escenario inicial a uno final, en que la meta que busca alcanzar esté resuelta, donde camino es una secuencia de operaciones consideradas primitivas por el sistema. Una solución al problema es la base de la secuencia correspondiente de acciones físicas en el mundo físico. Planear es una función inteligente de un robot; provee inteligencia y capacidad de resolución de problemas, pero es un área activa que necesita algoritmos de planeación eficientes y poderosos, capaces de funcionar en tiempo real. [23]

Un robot necesita de la Inteligencia Artificial, especialmente para tomar decisiones que le permitan llevar a cabo una tarea o alcanzar una meta. Para tomar estas decisiones es posible hacer uso de otras herramientas. La teoría de la probabilidad nos permite generar sistemas que razonen bajo situaciones de incertidumbre, aunque en estos casos, las acciones que puede llevar a cabo un robot (o un agente en general) no se sabe si son necesarias para alcanzar la meta ni si el robot será capaz de llevarlas a cabo. Por este motivo, un robot necesita ponderar la importancia de una meta particular y calcular qué tantas probabilidades tiene de alcanzarla, para lo cual se utiliza la teoría de la utilidad, y al combinarla con la teoría de la probabilidad se constituye la teoría de la decisión, la cual nos permite crear agentes racionales para mundos inciertos. [44]

Los robots carecen de la información completa acerca de su entorno, puesto que normalmente no se desempeñan en un ambiente ideal y el mundo real tiene una complejidad muy grande. Es necesario entonces que actúen tomando en cuenta la incertidumbre, pues es imposible evitarla en situaciones reales, lo que implica que las simplificaciones que eran posibles durante las inferencias deductivas de la lógica, ya no son válidas. Así, una decisión racional depende tanto de la importancia relativa de las submetas que componen su labor

como de la probabilidad que existe de conseguir cada una de ellas. En la mayoría de los casos no se sabe de antemano cual es la probabilidad de conseguir cada una de esas metas, por lo que se utiliza una herramienta de la teoría de la probabilidad que permite utilizar el resultado de un evento conocido para predecir la secuencia de eventos que llevaron a dicho resultado. Esta herramienta se llama Regla de Bayes y es la siguiente:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (1)$$

donde a $P(B|A)$ se le conoce como probabilidad *a posteriori*, y se calcula tomando las probabilidades conocidas de antemano (*a priori*) y la información que se va adquiriendo con el tiempo.

La ventaja principal de utilizar un razonamiento probabilístico en lugar de un razonamiento puramente lógico para decidir qué acción debe tomar un agente, es que cuando se deben tomar decisiones racionales, no es posible que el robot no realice ninguna acción aún cuando no exista suficiente información para predecir si el realizarla va a llevar al agente a alcanzar eventualmente su meta. Ponderar la importancia de cada submeta y calcular las probabilidades que tiene de realizarlas, le permite tomar la mejor decisión con que cuenta a la mano dadas sus creencias. La asignación de importancia hacia las metas se realiza utilizando la *Teoría de la Utilidad*, en la cual la preferencia de un agente por encontrarse en un estado u otro del mundo, se captura en una *función de utilidad*, la cual regresa un valor numérico a la deseabilidad de encontrarse en un estado dado. Este valor, llamado *utilidad*, se combina con las probabilidades que tienen los posibles resultados de efectuar una acción, y así asignar una *utilidad esperada* para cada acción [44].

TEORÍA DE LA
UTILIDAD

Con la unión de la Teoría de la Probabilidad, que describe qué es lo que debe creer un agente dada la evidencia que tiene y la Teoría de la Utilidad, que describe qué es lo que quiere hacer el agente, se genera la Teoría de la Decisión que permite definir qué es lo que debería hacer un agente, cuya idea principal recae en el llamado Principio de la Máxima Utilidad Esperada, que dice que un agente sólo se puede considerar racional si toma la acción que le brinde la utilidad esperada más grande, tomada de entre todos los posibles resultados de la acción.

La Teoría de la Decisión se utiliza para que un robot pueda tomar decisiones en situaciones en las que la incertidumbre y las metas conflictivas llevarían a un agente puramente lógico a no poder decidir nada. Se puede utilizar la Teoría de la Decisión, para crear un agente que tome decisiones considerando todas la acciones que puede tomar y escogiendo aquella que lo conduzca al mejor resultado. Es en este momento en que podemos considerar que tenemos un agente racional.

2.2.1.2 *Automatización e Independencia*

El diseñar un mecanismo inteligente e implementarlo, para darle autonomía a una máquina, es el propósito general en el campo de la investigación en automatización, donde el objetivo principal es dotar a las máquinas de la capacidad de generar comportamientos y respuestas a ciertas situaciones, utilizando conjuntos de reglas que los humanos preparan con anticipación dependiendo de dichas situaciones.

Se trabaja aquí bajo el supuesto de que tener una máquina que se opere automáticamente, es mejor que tener una máquina que necesite ser operada manualmente. La *automatización* de una máquina se puede ver como un mecanismo que sigue ciertas reglas que corresponden con el ambiente que la rodea, para generar respuestas propias a las situaciones que se le presenten en dicho ambiente. Sin embargo, si las acciones de un agente se basan completamente en conocimiento que se le implanta desde el inicio y no toma en cuenta su experiencia, dicho agente carece de *autonomía*. Para que un agente se pueda considerar autónomo, su comportamiento debe basarse tanto en el conocimiento del que se le provea desde su construcción (el cual debe establecerse en función del ambiente en que va a operar), como en su propia experiencia [38, 32]. En nuestro caso particular tenemos que es importante brindarle a los robots que juegan fútbol el conocimiento inicial que necesitan, que correspondería a las reglas del juego, una estrategia de juego personal, una estrategia de juego en equipo, etc. Sin embargo es igual de importante darle la habilidad de aprender, lo cual le brindaría autonomía y flexibilidad, y para esto necesitamos primero tener los medios para que el robot pueda percibir correctamente su entorno y que pueda ubicarse correctamente dentro de él. Una vez contando con éste y otros métodos que le permitan comprender su situación correctamente, necesitará una estrategia de aprendizaje, para lo cual existe toda una disciplina llamada Aprendizaje de Máquina.

AUTOMATIZACIÓN

AUTONOMÍA

2.2.1.3 *Aprendizaje de Máquina*

Una de las opciones que se tienen para que un robot realice una tarea es proveerlo de todas las reglas de comportamiento desde su construcción. Esto se hace mediante modelos matemáticos de las actividades que se quiera que realice, ya sea caminar, nadar, buscar objetos o rutas, o en nuestro caso ganar un partido de fútbol. Se podrían realizar modelos de dichas actividades, sin embargo, no existe una teoría perfecta de cómo es que se debe realizar una acción como caminar o correr, por lo que usualmente se usan modelos que aproximan un comportamiento, los cuales pueden llegar a ser muy buenos, pero se quedan muy cortos comparados con lo que puede realizar un ser humano u otro tipo de animal. Es por esto que se ha busca-

do lograr brindarle a una máquina los medios para que ella misma aprenda una tarea en función de prueba y error, aprendiendo de su experiencia.

La idea detrás del aprendizaje está en que las herramientas con las que cuenta un robot, no las utilice únicamente para definir su actuación, sino también para mejorar a futuro la habilidad que tiene para actuar. El aprendizaje surge como resultado de la interacción entre un agente y el mundo, pero también de la observación que realiza de su mismo proceso de toma de decisiones, y puede variar desde una memorización trivial de sus experiencias, hasta la capacidad de desarrollar sus propias teorías, aunque la investigación en el campo todavía se encuentre lejos de esta meta.

La esencia del aprendizaje es generar conocimiento a través de ensayo y error, y reaccionar utilizando dicho conocimiento [28]. En el marco del aprendizaje de máquinas, un humano inicialmente define las tareas que le quiere dar al robot para realizar, y después debe diseñar una forma de evaluación que permita valorar qué tan bien la respuesta de la máquina satisface el deseo del humano. Las funciones de evaluación normalmente se conforman de fórmulas numéricas y lo importante de estas es que sus evaluaciones estén relacionadas con el contexto lo más estrechamente posible. Es posible entonces que una respuesta particular de un robot, obtenga diferentes valores de evaluación dependiendo de la función que se utilice para evaluarla, puesto que la definición de una función depende en gran medida del punto de vista de los investigadores que la plantean. Una máquina únicamente puede trabajar con el conocimiento adquirido mediante un proceso de aprendizaje, puesto que éste va a estar enfocado a la realización de una tarea específica, y le sería muy difícil aplicar ese conocimiento a otro tipo de tarea.

Formalmente el área del Aprendizaje de Máquinas se encarga del diseño y desarrollo de algoritmos y técnicas que le permitan a una máquina aprender, en el sentido antes mencionado. El objetivo de esta disciplina es desarrollar la habilidad de una máquina de mejorar sus comportamientos automáticamente a partir de la experiencia.

2.2.2 *Procesamiento Digital de Imágenes*

Una vez establecida la importancia de la inteligencia artificial, podemos pasar a otra de las áreas de investigación que nos sirvieron para desarrollar el método, que es el procesamiento digital de imágenes. Dado que nuestro método se basa en el sistema de visión del robot, antes de ver lo que es la visión como un sistema completo, hay que entender cómo es que se procesan las imágenes dentro del sistema del robot para poder hacer uso de algoritmos básicos que nos permitan generar un método más avanzado y complejo. Aún cuando la visión es un problema tridimensional, los algoritmos de visión

robótica se basan en manejar imágenes bidimensionales que representan imágenes en 3D, por tal razón es importante comprender cómo es que se representan y procesan las imágenes.

Las operaciones más simples que se pueden obtener del procesamiento de imágenes se llaman operadores puntuales, y son aquellas cuya salida depende únicamente del valor de cada uno de los píxeles que integran una imagen. Dentro de estos operadores están los operadores de brillo y contraste, así como los de colorización y umbralización. Más allá de los operadores puntuales se encuentran los filtros u operadores locales o de vecindad, llamados así porque utilizan un conjunto de valores de píxeles que se encuentran en la vecindad de un píxel dado para determinar la imagen resultante de la aplicación del filtro. Además de utilizarse para ajustes de tono locales, los filtros se utilizan para suavizar una imagen, disminuir el ruido o enfocar una imagen, entre otras aplicaciones.

Dentro de los tipos de filtros están los lineales, que son aquellos cuya salida está en función de combinaciones ponderadas de píxeles en pequeñas vecindades. Se llaman lineales puesto que el resultado de aplicar un filtro a la suma de dos señales (imágenes en este caso), es igual a sumar el resultado de la aplicación del filtro a cada una de las señales por separado. La ventaja que tienen estos filtros es que son fáciles de componer, y además están sujetos a la teoría de análisis de respuesta de frecuencias, puesto que una imagen se puede ver como una señal y es así como se han desarrollado muchos de los filtros y métodos dentro del campo del procesamiento de imágenes.

Un filtro lineal que es de nuestro particular interés es el operador extractor de Bordes de Sobel, el cual es una combinación separable de un filtrado horizontal de diferencias centrales, y de un filtro de suavizado vertical. Sin embargo, existen extractores de bordes que entregan mejores resultados. La extracción de bordes ha sido parte esencial de muchos sistemas de visión computacional, y nuestro método no es la excepción. Su principal ventaja es que sirve para simplificar el análisis de las imágenes reduciendo drásticamente la cantidad de información que se necesita procesar, y al mismo tiempo se mantiene la información estructural útil que se presenta en una imagen, como lo son los límites de un objeto o sus partes [50].

El trabajo de John Canny ha sido uno de los más importantes en lo que a extracción de bordes se refiere, usado por [2][41][37] entre muchos otros. Cuando él inició sus trabajos en esta área, ya había mucha investigación hecha en el campo y ya se habían establecido dos criterios indispensables que debía cumplir un buen detector de bordes, a los que él agregó un tercero y son los siguientes [14]:

1. UNA BUENA DETECCIÓN: Éste implica que un detector debe tener una baja probabilidad de fallar al marcar un borde real en una imagen, al igual que debe ser baja la probabilidad de marcar como borde algo que no es.

2. UNA BUENA LOCALIZACIÓN: Un extractor debe marcar cuáles de los píxeles de una imagen corresponden a un borde. Este criterio implica que aquellos píxeles marcados como bordes, deben encontrarse lo más cercanos al borde real en la imagen.
3. UNA RESPUESTA ÚNICA: Éste implica que un operador debe asignar un solo estado a un píxel, i.e. o lo marca como borde o como no-borde.

El tercero fue el criterio que agregó Canny, y aun cuando en teoría ya está implicado en el primer criterio, según Canny la formulación matemática que se tenía para este último, no manejaba explícitamente las respuestas múltiples y era esencial hacerlo, hecho que demostró con su método de detección de bordes, que resultó uno de los más importantes que existen, pues aún cuando lo desarrolló hace varios años, se sigue usando hasta la fecha (con algunas variaciones en la mayoría de los casos). Su enfoque se centraba en el análisis de los criterios de detección de bordes desde una perspectiva matemática. Ahí se dio cuenta que los bordes ocurren en los límites entre regiones definidas ya sea por sus diferencias de intensidad, textura, color. Sin embargo la segmentación precisa de una imagen es bastante complicada, por lo que decidió buscar la forma de detectar bordes utilizando únicamente información local de vecindades de píxeles, como puede serlo la localización o el cambio de intensidad.

En este sentido, es posible ver una imagen de tal forma que los valores del color le asignaran alturas a una malla, y ahí, los bordes se pueden ver como aquellos lugares donde se presentan escalones o donde haya cambios significativos de alturas. Modelando este escenario matemáticamente, se puede ver como un campo vectorial, donde los cambios de altura corresponden a las variaciones en el gradiente de las superficies en un punto dado. Es utilizando los gradientes que, cuando se presenta un cambio de altura correspondiente a un borde en la imagen, se puede conocer su magnitud y su dirección, utilizando la ecuación

GRADIENTE

$$J(x) = \nabla I(x) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)(x) \quad (2)$$

donde I es la imagen y $x = (x, y)$ son las coordenadas de cada píxel. La dirección del vector gradiente J apunta en dirección perpendicular al borde correspondiente, que es lo mismo que la dirección del ascenso en el cambio de altura y la magnitud indica la fuerza de variación de la pendiente.

Desafortunadamente, al tomar las derivadas parciales de una imagen se acentúan las frecuencias altas, que es donde la proporción de ruido en una señal es más grande y por tanto se amplifica el ruido presente en la imagen. Para hacer frente a este problema, es necesari-

rio suavizar (desenfocar) la imagen utilizando un *filtro de paso bajo*³ antes de calcular el gradiente, y así no generar tanto ruido. Como lo que se quiere es un detector cuya respuesta sea independiente de la orientación de una imagen, se necesita un filtro de suavizado que sea simétrico circularmente. El filtro Gaussiano es el más utilizado en los algoritmos de detección, pues es un filtro simétrico circular y además es separable, lo cual aprovecha las propiedades de las operaciones lineales y por tanto es conmutable con la derivación. El gradiente de una imagen suavizada puede ser escrito como

$$J_{\sigma}(x) = \nabla[G_{\sigma}(x) * I(x)] = [\nabla G_{\sigma}](x) * I(x) \quad (3)$$

en donde se realiza una convolución de la imagen con las derivadas horizontal y vertical del kernel Gaussiano, y el parámetro σ indica el ancho de la función.

En muchas aplicaciones se busca obtener bordes lo más delgados posible. Para esto se buscan los valores máximos en la magnitud del gradiente en dirección correspondiente. A esto se le conoce como la *Supresión de los No-Máximos*, y es una técnica de adelgazamiento de bordes. Consiste en tomar la derivada direccional de la magnitud del gradiente del campo de fuerza en dirección del gradiente, que equivale al producto punto entre el segundo operador de gradiente y el resultado del primero, y buscar *cruces cero*, i.e. los puntos donde la magnitud del gradiente asume el máximo local en dirección al gradiente. Para esto es necesario definir un conjunto de orientaciones discretas, pues no se conoce la dirección del gradiente localmente y posteriormente se recorre un filtro de 3×3 sobre cada pixel y en caso en que el pixel central no presente la mayor magnitud, se elimina como parte del borde, preservando sólo los puntos máximos.

SUPRESIÓN DE LOS
NO-MÁXIMOS

CRUCES CERO

Los gradientes de mayor intensidad tienen una mayor posibilidad de corresponder a bordes en la imagen que los que tienen menor intensidad. En la mayoría de los casos, es imposible definir un umbral con el cual todos los gradientes de mayor intensidad correspondan únicamente a los bordes, pues cuando se tienen varios objetos en una imagen, cada objeto puede tener un valor diferente de intensidad en el borde y además, al calcular las derivadas de una imagen suavizada, es posible encontrar puntos donde exista una intensidad relativamente alta, pero que no correspondan a un borde. Para enfrentar esta situación, se utiliza una técnica llamada *histéresis*, en la cual se requiere de dos umbrales, uno alto y uno bajo. Al utilizar el umbral alto, sabemos con mucha probabilidad que aquellos puntos cuya intensidad lo sobrepase forman parte de un borde en la imagen. Posteriormente, bajo el supuesto de que los bordes importantes son por lo regular curvas continuas, aquellos bordes que ya se definieron con el primer umbral se recorren ligando los pixeles que los conforman siguiendo la dirección marcada por sus gradientes, que es el

³ Un filtro de paso bajo es un filtro que deja pasar las señales de baja frecuencia y atenúa las frecuencias altas en función de un umbral establecido.

camino trazado por el borde. Cuando se llega al final de un borde, se continúa recorriendo la imagen en esa dirección y se aplica el umbral bajo, y si los píxeles que están en el camino sobrepasan el umbral, se marcan también como bordes definitivos. El resultado de la histéresis es el resultado final del extractor de bordes de Canny.

Mientras que los bordes y las curvas en general sirven para describir los contornos naturales de los objetos, podemos observar que el mundo que nos rodea está conformado en gran parte por líneas rectas, y en este hecho nos apoyamos para nuestro método, especialmente cuando estamos tratando con un ambiente delimitado por líneas rectas tan marcadas como lo son las líneas de la cancha y de las porterías. La detección de líneas en una imagen puede ser muy útil para ésta y muchas otras aplicaciones, como lo son el modelado arquitectónico o el análisis de documentos digitalizados. Uno de los filtros de detección de líneas más popular y robusto es la *Transformada de Hough*, un método desarrollado por Paul Hough en [26], y que no sólo sirve para detectar líneas rectas, sino que sirve para detectar básicamente cualquier curva que se pueda parametrizar. Éste es un método que se utiliza en mucha investigación de visión computacional, e involucra áreas como lo son la parametrización, el diseño de acumuladores, algoritmos de votación, detección de picos en señales, entre otras.

TRANSFORMADA DE
HOUGH

La Transformada de Hough utiliza la información adquirida previamente por un extractor de bordes, para localizar líneas plausibles en una imagen. Cada pixel correspondiente a un borde vota por todas las líneas que pasen por ese punto, y esos votos se guardan en un acumulador para cada línea; aquellas que corresponden a los acumuladores más grandes, son las que se toman como líneas de la imagen. Sin embargo, para poder realizar una votación para escoger la líneas más adecuadas, es necesario primero definir una forma de parametrizarlas. La parametrización más utilizada en el algoritmo de Hough es la de la normal y la distancia a una línea (ρ, θ) , en la cual una línea se representa mediante la ecuación

$$\rho(\theta) = x \cos \theta + y \sin(\theta) \quad (4)$$

donde θ es el ángulo que va del origen al punto más cercano de la línea, y ρ es la distancia que hay entre dicho punto y el origen. En la figura 1 se puede observar cómo es que se representa esta parametrización, y se muestra también el *Espacio de Hough*, donde se puede visualizar como funciona el algoritmo gráficamente.

Este algoritmo puede ser generalizado para encontrar casi cualquier forma, ya sean figuras geométricas regulares, o bien figuras no regulares pero que se puedan parametrizar de alguna forma, tal como lo hacen en [57][7]. En estos casos, en lugar de usar una ecuación paramétrica de una curva, se utiliza una tabla de búsqueda para definir las relaciones entre las posiciones de los límites y las orientaciones, y los segmentos que conforman la curva.

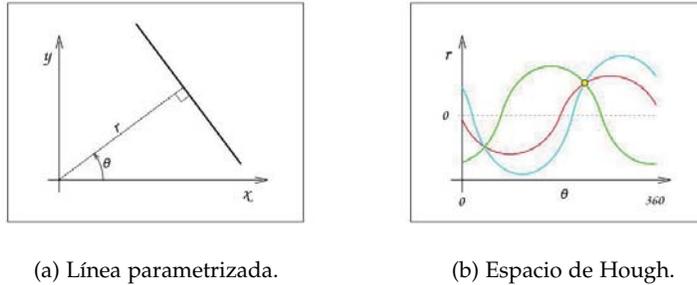


Figura 1: Algoritmo de Hough: Cada pixel vota por una familia de líneas que pasan por ese punto, y corresponden a una senoide. Los puntos donde se registran más cruces corresponden a las líneas con mayor número de votos.

Sin embargo hay ocasiones en que conviene, más que obtener líneas, obtener únicamente los segmentos de recta que aparecen en la imagen. Para esto se emplea una versión probabilística del método de Hough, cuya idea principal es que, cuando se tienen líneas que tienen un soporte fuerte, i.e. que son líneas largas en relación con el tamaño de la imagen, únicamente una fracción pequeña de los puntos que comprenden la línea necesita votar para que el acumulador correspondiente alcance un valor suficiente para que sea considerada una línea de la imagen. Para líneas más pequeñas, una porción mayoritaria es la que necesita votar para que sean tomadas en cuenta. También se puede dar el caso en que el ruido de la imagen vote a favor de líneas espurias. Tomando en cuenta todos esos casos, el algoritmo de Hough Probabilístico empieza con todos los puntos de los bordes como la población total de posibles votantes. Luego se escoge aleatoriamente un punto para que realice sus votos. Después de cada voto en los correspondientes acumuladores, se verifica si el total de votos de un acumulador logró exceder un umbral establecido que define el máximo de votos que podría tener un acumulador en caso que tenga puros votos debidos al ruido. Si ese es el caso, entonces se toma como línea la correspondiente al acumulador, se eliminan los votos que se tenían acumulados ahí, y se remueven de la población total los puntos que votaron por dicha línea. Con esto, el umbral debe cambiar cada vez que se escoge una línea, pues el número de puntos que pueden provenir de ruido aleatorio está en función del número de puntos que hay en la población. Finalmente el algoritmo queda así:

Finalmente, el algoritmo de Hough probabilístico regresa los segmentos de línea que forman una imagen, parametrizados mediante las coordenadas de sus extremos.

ALGORITMO DE HOUGH PROBABILÍSTICO

1. Revisar la imagen de entrada; si está vacía, termina.
2. Actualizar el acumulador con un pixel individual escogido aleatoriamente de la imagen de entrada.
3. Eliminar el pixel seleccionado de la imagen de entrada.
4. Revisar si el pico más grande en el acumulador que fue modificado por el nuevo pixel es mayor al umbral $\text{thr}(N)$. Si no, ir a 1.
5. Buscar a lo largo del corredor especificado por el pico en el acumulador, y encontrar el segmento más largo que sea continuo o bien presente un espacio no mayor al umbral dado.
6. Eliminar los pixeles que forman parte del segmento, de la imagen de entrada.
7. Eliminar los votos del acumulador de todos los pixeles provenientes de la línea por la que se votó previamente.
8. Si el segmento de línea es más largo que el tamaño mínimo, agregarla a la lista de salida.
9. Ir a 1.

2.2.3 *Visión Computacional*

Llegamos pues a la disciplina que se encarga emular los sistemas visuales mediante el uso de las computadoras. Ésta surge de la curiosidad de los psicólogos perceptuales que se han dedicado a comprender cómo es que funciona el sistema visual del ser humano, y definieron principios bajo los cuales pensaban que funcionaba, aun cuando se toparon con varias ilusiones ópticas que conformaban la excepción a esas reglas. Por su parte, los científicos de la computación que investigan la visión, han desarrollado paralelamente modelos matemáticos que sirven para representar formas tridimensionales y para proveer de una apariencia realista a objetos imaginarios. Sin embargo, aún después de todos los avances que se han tenido en el área, no se ha logrado desarrollar un sistema que funcione a la par del sistema de visión humano, ni aún en las primeras etapas de su vida.

Los enfoques que se han utilizado para hacer frente a esta situación, implican la utilización de modelos probabilísticos y modelos físicos para generar soluciones potenciales. Estos modelos se utilizan en conjunción con las capacidades de la computación gráfica para generar los algoritmos de la visión computacional. Tanto la computación gráfica como la física modelan cómo es que los objetos se mueven y se animan, o como la luz se refleja en las superficies o se dispersa en la atmósfera, o bien cómo se refracta a través de lentes de cámaras y finalmente se proyecta en una superficie plana. En el área de la visión computacional, se intenta hacer lo contrario, i.e. en lugar de modelar cómo es que funcionan todos esos fenómenos, se realiza una descripción del mundo que se puede observar en una o varias imágenes, y en

función de esto se busca reconstruir sus propiedades como la forma, la iluminación o la distribución de colores.

Curiosamente, en los inicios de la Inteligencia Artificial, se pensaba que la parte cognitiva de la inteligencia era mucho más complicada que la parte perceptual, de hecho se pensaba que esta última se habría de solucionar en muy poco tiempo y que eventualmente sólo tendrían que preocuparse del aprendizaje. Muchos de los que se encontraban en el campo del procesamiento digital de imágenes, extendieron su investigación con el deseo de recuperar estructuras tridimensionales del mundo a partir de imágenes, y utilizar esto como el primer escalón hacia un entendimiento completo de una escena.

En lo que a nosotros respecta, las capacidades visuales le brindan al robot un mecanismo sensorial para responder al ambiente de una manera inteligente y flexible. Con el paso del tiempo, estos sistemas visuales han mejorado de tal forma que han flexibilizado a la robótica en general. De hecho se puede considerar a la visión como el medio de percepción más importante de que hace uso un robot. El proceso general de la visión robótica, consiste en extraer las imágenes del mundo real, caracterizarlas e interpretar la información presente en las imágenes, para que el robot pueda actuar en función de dicha interpretación.

Las etapas que comprende un sistema de visión robótica son la extracción de imágenes mediante los sensores, el preprocesamiento de las imágenes, la segmentación de las imágenes, la descripción de los objetos de la escena, el reconocimiento de los objetos, y la interpretación de imágenes en función de los objetos presentes en ellas. Existe una clasificación general de estas etapas, que las separa en bajo, mediano y alto nivel. Las de bajo nivel son aquellas que emulan las reacciones automáticas de un sistema, que corresponden al uso de los sensores para adquirir las imágenes. Las de nivel medio implican la segmentación, caracterización y etiquetación de los objetos. Por último, las etapas de nivel alto corresponden a la cognición y son el reconocimiento y la interpretación de las imágenes.

2.2.3.1 Reconocimiento de características

Una de las bases principales de la visión computacional está en el reconocimiento de características de la imagen, que sirven para asignar una relación entre diferentes imágenes que tienen elementos en común. La detección de estas características y el posterior emparejamiento, son componentes principales en muchos sistemas de visión. Uno de los usos más comunes es correlacionar imágenes para unir las en un mosaico compuesto, o bien construir modelos tridimensionales de un objeto, o crear imágenes intermedias entre dos imágenes separadas. Para poder aprovechar las ventajas de la detección de características, se debe primero definir qué tipo de características se van a utilizar. Por lo regular se usan locaciones específicas de la imagen,

conocidas como *puntos de interés*, y se describen mediante parches de píxeles que rodean una locación en la imagen donde existe la mayor discordancia en una vecindad, para que puedan ser fácilmente distinguibles unos de otros.

Otro tipo de características importantes son los bordes, que pueden ser emparejados mediante su orientación y su apariencia focal. Además de ser buenos indicadores de los límites de un objeto, también ayudan a decidir cuándo existen oclusiones en una imagen o una secuencia de video. Los bordes pueden ser agrupados en líneas rectas que compartan los puntos de fuga desde donde se tomaron. Los puntos de interés pueden ser utilizados para encontrar un conjunto disperso de locaciones correspondientes en diferentes imágenes. Esto se ha utilizado para calcular la pose de la cámara o calcular correspondencias al realizar emparejamientos estereográficos y, más recientemente, para realizar instanciación y reconocimiento de objetos, fusión de imágenes y modelado tridimensional automático. Una ventaja de utilizar puntos de interés es que permiten realizar un emparejamiento aún en la presencia de oclusiones y de cambios de escala y orientación significativas.

Existen dos grandes enfoques para encontrar los puntos característicos y sus correspondencias en diferentes imágenes. El primero tiene que ver con encontrar características que puedan ser rastreadas de manera precisa utilizando técnicas de búsqueda local; local con respecto a las posiciones en la imagen original. Este enfoque es ventajoso cuando se trata con imágenes tomadas desde puntos de vista muy cercanos, o bien cuando se trabaja con secuencias de imágenes o videos. El segundo enfoque busca las características independientemente en una imagen y las almacena en una base de datos, y luego busca en otras imágenes mediante una búsqueda generalizada. Este enfoque se utiliza cuando se espera que la apariencia de una imagen o el punto de vista desde donde fue tomada, cambie mucho con respecto a las demás.

La extracción de características puntuales en una imagen consiste en un proceso muy específico que empieza por la fase de detección o extracción de las características, en la que se buscan aquellas que por su locación se puedan distinguir con mayor facilidad, caso que se da cuando las variaciones en la vecindad de un píxel son muy grandes. Después está la fase de descripción de características, en donde cada región de los puntos detectados se convierte en un parche o descriptor compacto e invariante, que pueda ser comparado con otros descriptores, por lo que todos deben ser diferentes entre sí. Posteriormente está la fase de emparejamiento de características entre imágenes, que busca los candidatos de descriptores probables presentes en otras imágenes. Finalmente está la fase de rastreo de características, que en realidad es una alternativa a la fase anterior, pues en lugar de únicamente buscar los descriptores de unas imágenes en otras, toma

en cuenta la localización de los descriptores de la imagen original y busca los mismos descriptores en las imágenes correspondientes al siguiente tiempo únicamente en la vecindad de dichas locaciones. Uno de los algoritmos más utilizados para realizar este tipo de emparejamientos o rastreos, es el algoritmo de SIFT de David Lowe [35], en el cual se explican con mayor detalle estas fases.

Por su parte, los bordes en una imagen conforman un tipo de características que se encuentran con mayor frecuencia, y además contienen asociaciones semánticas mucho más importantes que los descriptores puntuales, pues una vez que se encuentran los límites de un objeto, es más sencillo definir qué representa ese objeto. Como ya se explicó anteriormente, para la extracción de los bordes es posible utilizar varios métodos como los filtros de Sobel o Canny, aunque también se han utilizado métodos que intentan aproximar los bordes mediante polilíneas o *splines*. Teniendo los bordes extraídos, la aplicación de la transformada de Hough o algún otro extractor de líneas, permite realizar una correlación entre éstas y un ambiente tridimensional, calculando los puntos de fuga presentes en las imágenes y agrupando las líneas en función de estos. Esto se puede hacer porque en una escena, líneas estructuralmente importantes tienen el mismo punto de fuga dado que son paralelas en el mundo real. De hecho, como se verá en el apartado 2.2.4, estos elementos nos permiten realizar las calibraciones extrínsecas e intrínsecas de una cámara, lo cual permite calcular la posición relativa de la cámara con respecto a la imagen.

2.2.3.2 Reconocimiento de objetos

Otro de los objetos de estudio de la visión computacional es el reconocimiento de objetos. Sin embargo, ésta es una tarea bastante complicada, pues el mundo real está conformado por objetos muy diversos, que en una imagen se ocluyen unos a otros y pueden presentarse en una infinidad de poses con respecto a la cámara. Se ha intentado realizar búsquedas exhaustivas de ejemplares en una base de datos, pero este enfoque tiene el problema de la variabilidad intrínseca aún dentro de un mismo tipo de objetos, la cual se debe a que por lo regular tienen una articulación no rígida sino más bien compleja y a las variaciones extremas de forma y apariencia, por lo que sólo se puede realizar imponiendo grandes restricciones sobre la búsqueda.

Se puede ver al reconocimiento de objetos como el proceso de (teniendo una imagen segmentada previamente) identificación de los objetos presentes y el etiquetado de los mismos. La mayoría de los métodos de reconocimiento asumen que los objetos en la imagen han sido segmentados correctamente y que se conoce la geometría de la vista desde donde se tomó la imagen, que por lo regular tiende a ser perpendicular al eje de visión de la cámara.

Los enfoques de reconocimiento se dividen en dos categorías: Los métodos Teóricos de Decisión y los métodos Estructurales [23]. Los primeros se basan en descripciones cuantitativas de los objetos, utilizan funciones de decisión y vectores de patrones para definir un objeto, de tal forma que a un vector de patrones se le aplican las funciones de decisión, y aquella función que regrese el valor más alto es la que designa a qué clase corresponde dicho objeto. Los enfoques estructurales se fundamentan en las relaciones geométricas inherentes a la forma de un objeto, i.e. se descomponen los objetos en patrones primitivos conformados por descripciones simbólicas y sus relaciones, se almacenan en una base de datos, y luego se utilizan esas descripciones para compararlas con patrones hallados en la imagen que se está analizando. Es común en este tipo de enfoque utilizar algoritmos de votación similares a la transformada de Hough, o también búsquedas indexadas, como lo hacen en [25].

2.2.3.3 Localización

El problema de la localización consiste en la determinación de la pose del robot a partir de la información obtenida de sus sensores. Existen diferentes problemas de localización [53], que van desde el *rastreo de posición*, en el cual el robot conoce su posición inicial y sólo busca compensar los errores incrementales presentes en los sistemas de odometría robótica. Por otro lado está el *problema global de localización*, en el cual el robot debe localizarse sin tener información *a priori* de su posición. También está el *problema del secuestro*, en el cual un robot que ya conoce su localización, se teletransporta a un lugar diferente sin aviso alguno y debe volver a localizarse sin problemas.

Es común que los enfoques que se le dan a este problema primero establezcan una correspondencia entre las mediciones de los sensores del robot y las características sobresalientes del ambiente que lo rodea, como pueden ser líneas, esquinas, puntos de interés, o hasta objetos fácilmente reconocibles en el campo. Posteriormente se determine la localización del robot utilizando un marco de coordenadas relativo a dichas características.

Por lo regular los problemas de localización se presentan en ambientes no controlados y se utilizan sensores que entregan información contaminada. Por lo que una buena solución al problema de la localización tiene que cumplir con tres condiciones [6]:

- Debe ser tolerante a errores en las mediciones.
- Debe tolerar reconocimientos falsos de características del entorno.
- Debe ser lo suficientemente simple como para aplicarlo de forma rápida y eficaz.

Para poder abordar bien éste problema, es necesario contar con un buen modelo de movimiento del sistema, o del robot en este caso. Por lo regular los modelos de movimiento son de dos tipos, los basados en odometría y los basados en velocidad. El primer caso se da cuando un robot se mueve por medio de ruedas, y tiene tres grados de libertad, la primera rotación, la traslación y la segunda rotación, por lo que cada movimiento se parametriza mediante un vector $(\delta_{rot1}, \delta_{trans}, \delta_{rot2})$. El segundo caso se da cuando el robot no tiene ruedas, por lo que su movimiento es más impredecible; tiene 2 grados de libertad, correspondientes a la velocidad traslacional y a la velocidad rotacional, y se parametrizan con un vector (v, w) .

Cuando se busca realizar la localización de un robot, es posible resolverlo en dos modalidades distintas. Cuando se resuelve de forma estática, el sistema recibe la información recolectada por los sensores y en función de ésta, debe calcular su pose con respecto al marco de coordenadas del mundo real. Por su parte, la localización dinámica asume que el sistema ya resolvió los problemas de la localización estática, por lo que sólo se necesita aplicar una ligera perturbación en el estado anterior. En este trabajo se busca resolver la localización estática, para dar pie a resolver el problema en su forma dinámica, la cual se complica pues el entorno también es dinámico, especialmente al utilizar características locales.

Frecuentemente, la localización va de la mano con el problema del mapeo, y en este caso el robot desconoce no solo su posición sino que no conoce nada del ambiente, por lo que debe realizar una construcción incremental de un modelo de su entorno, y al mismo tiempo debe encontrar su posición dentro del mapa recién creado. A esto se le conoce como el problema del *SLAM*, y es uno de los más importantes que hay dentro de la comunidad de robots móviles, pues es necesario resolverlo para lograr que un robot sea realmente autónomo. Éste es un problema que se ha abordado desde muchas perspectivas y se podría considerar como resuelto en la teoría, sin embargo en la práctica sólo se han encontrado soluciones que funcionan para casos particulares y al enfrentarlos con ambientes grandes y variados llegan a tener problemas.

En la competencia de RoboCup se utilizan muchos métodos diferentes de localización, desde los que se basan en detección de bordes como en [43][12], hasta los que detectan detección de líneas del campo y otros elementos invariantes de la cancha, como en [49][8][51][45].

EL SLAM se originó cuando se empezaron a introducir métodos probabilísticos en el área de visión e inteligencia artificial, en las áreas de mapeo y localización. Basándose en muchos de los trabajos que se publicaron en ese momento, Smith *et al.* [48] realizaron un análisis en el que mostraron que si se situaba a un robot móvil en un ambiente desconocido y se ponía a explorarlo tomando mediciones de las características de su entorno, las estimaciones de dichas características

iban a estar correlacionadas conforme iba moviéndose, pues el error en las estimaciones es común a todas ellas y está en función del movimiento. Por tanto, en ese tipo de problemas en que la localización y el mapeo deben realizarse de forma simultánea, es necesario que el registro que lleva el robot de su estado contenga la pose estimada del vehículo, pero también la posición de las características del ambiente que va registrando [21]. Una de las dificultades más grandes que se encuentran al intentar resolver este problema es que conforme se van registrando nuevas características, la incertidumbre crece ilimitadamente, especialmente cuando aumenta la distancia entre los sensores y las características del entorno; además, tiende a surgir una divergencia en los registros, pues suele suceder que se realicen malas asociaciones entre las características que se observan y las que están registradas ya en el mapa, y muchas veces se tiende a confundir características conocidas por desconocidas pero parecidas y viceversa.

Sin embargo en situaciones como la que se presenta en este trabajo, la parte del mapeo se tiene resuelta de antemano, pues es un ambiente bien conocido, aún cuando se tiene que resolver la dinamicidad del ambiente, pues aunque éste no cambia *per se*, existen muchos elementos que generan variaciones grandes en las mediciones. Cuando se intenta resolver un problema de localización, existen dos grandes paradigmas: los Filtros de Kalman y los Filtros de Partículas, que son los mismos usados para SLAM, pero no se necesita ir modelando el mapa.

Es posible utilizar el teorema de Bayes para realizar una estimación de la distribución de probabilidad $P(x|z, u)$ de un estado x de un sistema, condicionado a los datos obtenidos tanto de los sensores (z), como de los controles (u). Una creencia es calcular esta probabilidad en un tiempo dado

$$bel(t) = p(x_t | z_{1:t}, u_{1:t}), \quad (5)$$

donde x_t es el estado en que se encuentra el sistema (en este caso el robot) en el tiempo t , $z_{1:t}$ es el conjunto de mediciones realizadas desde el tiempo 1 hasta el tiempo t , y $u_{1:t}$ es el conjunto de acciones realizadas desde el tiempo 1 hasta el tiempo t . Un *Filtro Bayesiano* es aquel en el que se actualizan recursivamente las creencias basándose en la acción y medidas del tiempo anterior, y con éste es posible saber la probabilidad de encontrarse en cualquier estado de forma certera[52].

FILTRO BAYESIANO

FILTRO DE
KALMAN

Un *Filtro de Kalman*, descrito por primera vez en [29], es un filtro de Bayes el cual asume que

- Cada variable de estado está valuada en los reales y distribuida de forma Gaussiana.
- Cada sensor genera ruido Gaussiano no sesgado (de media 0).

- Cada acción se puede describir mediante un vector de números reales, uno por cada variable de estado.
- EL nuevo estado es una función lineal del estado anterior y de la acción llevada a cabo.

De cumplirse todo esto, es posible realizar las predicciones y estimaciones implementando únicamente una serie de cálculos sobre matrices, aún si se trata de un número grande de variables de estado[44].

Los filtros de Kalman procesan todas las mediciones disponibles para estimar el valor de las variables de interés, mediante el uso del conocimiento del sistema, de la dinámica de los sensores y de la información inicial de las condiciones de las variables de interés, así como de la descripción estadística del ruido del sistema, errores en el sistema, y las incertidumbres del modelo de dinámica [36].

La forma en que el filtro calcula el estado del sistema es mediante dos ecuaciones lineales

$$x_t = A_t x_{t-1} + B_t u_t + \epsilon_t \quad (6)$$

$$z_t = C_t x_t + \delta_t \quad (7)$$

donde la matriz A es la matriz de transición y describe la evolución del estado en el tiempo $t - 1$ al estado en el tiempo t , sin tomar en cuenta el control ni el ruido; la matriz B describe cómo cambia la acción u_t del tiempo $t - 1$ al tiempo t ; por su parte la matriz C representa el mapeo que se hace del modelo que se tiene en el estado x_t al que se obtiene al agregarle la observación z_t . ϵ_t y δ_t son los factores de ruido del control y de las observaciones respectivamente, y son independientes uno del otro.

Este filtro es recursivo en el sentido de que no requiere mantener registro de todos los datos recolectados, sino que re-procesa cada vez que se realiza una nueva observación. Es un ciclo que va de la *actualización en el tiempo* (predicción) a la *actualización de las mediciones* (corrección). La actualización en el tiempo es responsable de proyectar a futuro el estado actual y los errores de covarianza estimados, para obtener las estimaciones *a priori* que se usarán en el siguiente tiempo. La actualización de las medidas es responsable de incorporar las nuevas observaciones en las estimaciones *a priori* para obtener un estimado *a posteriori* mejorado.

Las ecuaciones de la actualización en el tiempo se conocen como ecuaciones *predictoras* y son

$$\hat{x}_t^- = A \hat{x}_{t-1} + B u_t \quad (8)$$

$$P_t^- = A P_{t-1} A^T + Q \quad (9)$$

donde P_t^- es la estimación *a priori*, P_t es la estimación *a posteriori*, A y B son las mismas que en (6) y Q es la covarianza del ruido de las acciones.

Las ecuaciones de *la actualización de las mediciones* se conocen como ecuaciones correctoras y son

$$K_t = P_t^- H^T (H P_t^- H^T + R)^{-1} \quad (10)$$

$$\hat{x}_t = \hat{x}_t^- + K_t (z_t - H \hat{x}_t^-) \quad (11)$$

$$P_t = (I - K_t H) P_t^- \quad (12)$$

Lo primero que se hace para la actualización de las medidas es calcular la ganancia de Kalman, K_t , donde H es la matriz de predicción de las medidas. Después se realizan las medidas y se almacenan en z_t para poder generar la estimación *a posteriori* \hat{x}_t del estado. Finalmente se obtiene el error de covarianza *a posteriori* mediante la ecuación (12). Después de cada iteración, el proceso se repite utilizando los estimados *a posteriori* como los nuevos estimados *a priori* [55].

Los resultados que arroja este método son muy buenos si se cumplen sus requisitos, como lo muestran en [11][3]. El problema que presenta el filtro de Kalman es que la mayoría de los sistemas de interés son no lineales en algún aspecto, por lo que no es posible emplearlo. Aún así es un método muy empleado, en especial cuando se aplica una linearización local al sistema y de esta forma es posible usarlo en sistemas no lineales. Para esto se utilizan por lo regular una de dos variantes.

Filtro de Kalman
EXTENDIDO

El Filtro de Kalman Extendido utiliza series de Taylor para llevar a cabo la linearización, que en términos prácticos implica utilizar matrices Jacobianas en lugar de las transformaciones lineales originales. Esta variante ha dado buenos resultados en cuanto a localización y mapeo, como en [59] que realiza el mapeo usando propiedades estructurales del ambiente, [33] que utiliza guías geométricas para usar de características del ambiente, y otros como [15][17]b. Por otro lado, el Filtro de Kalman 'Unscented' estima las predicciones de estado y de observación, así como la covarianza cruzada entre ambas, muestreando el espacio de estados alrededor de la media y la covarianza actuales, aplicando la transformación no lineal a los puntos sigma, y recuperando la media y covarianza transformadas [51].

UNSCENTED
KALMAN FILTER

Un filtro de partículas es otra forma de calcular el estado de un sistema utilizando una relación existente entre este y el conjunto de mediciones que se adquieren durante el tiempo. Originalmente se desarrolló para rastrear una variable de interés conforme evoluciona en el tiempo, usando una distribución de densidad de probabilidad regularmente no Gaussiana.

Básicamente lo que hace es generar una muestra de hipótesis sobre el estado en que se puede encontrar la variable, asignadas de manera aleatoria entre los posibles estados. A estas hipótesis se les llama partículas. A cada una se le asigna un valor que está en función de la probabilidad que tiene dicha partícula de tener el mismo estado que la variable de interés. A ese valor se le llama peso de la partícula, y

cada partícula tiene asignado un peso distinto. Para estimar el valor de la variable de interés, se toma la suma ponderada de los estados de la partículas.

El algoritmo del filtro de partículas es recursivo y consta de dos fases, la de predicción y la de actualización [18].

FASE DE PREDICCIÓN Después de realizarse una acción cada partícula se modifica de acuerdo al modelo de acciones y de movimiento, incluyendo un valor de ruido aleatorio que simula el ruido de la variable de interés. El modelo simula el efecto que tiene un movimiento sobre el conjunto de partículas.

FASE DE ACTUALIZACIÓN Los pesos de las partículas se actualizan de acuerdo a la información obtenida en una nueva medición. Después se realiza un re-muestreo en el cual se genera un nuevo conjunto de partículas, pero en lugar de asignarles el estado de forma aleatoria, se hace función del peso de las partículas del conjunto anterior, de tal forma que es posible que varias partículas compartan el mismo estado, y que los estados con menos probabilidad no obtengan ninguna partícula que los represente.

Los filtros de partículas son aproximadores universales de densidad, que enfocan los recursos computacionales en las áreas relevantes gracias al re-muestreo, y son fáciles de implementar, lo que los convierte en un paradigma atractivo de localización, utilizado por [13][45][27] entre otros.

Sin embargo, el tamaño de la muestra debe mantenerse grande, o un robot que ya esté bien localizado puede perder su posición debido a que ninguna nueva partícula se genere en el estado correcto. Lo cual no es un problema cuando el mapa es pequeño, pero conforme crece, el número de partículas necesarias puede ser demasiado.

Tampoco es el mejor método para enfrentar el problema del secuestro, pues las partículas difícilmente se situarán cerca de la nueva posición [53].

Cuando se realiza la localización usando sensores visuales, es muy común que se haga mediante un sistema de visión estereográfica, tal como lo emplean [1]. Pero existe un enfoque muy grande llamado *Localización Monocular*, que es cuando sólo se cuenta con una cámara de video como sensor del robot para realizar la localización, tal como lo hacen en [20][22]. Últimamente es muy común realizar esta tarea utilizando cámaras RGBD, que además de imagen capturan la distancia que hay de la cámara al objeto que observan, como en [24][19]. Sin embargo, como es el caso presente, hay veces en que no es posible usar cámaras de este tipo y hay que calcular las distancias de otra forma. Hay quienes han utilizado la intensidad de la imagen como valor de cambio de distancias. También se ha considerado usar puntos de interés en la escena, pero aunque son fáciles de detectar, son proclives

a oclusiones y para emparejarlos de manera confiable se necesita un gran conjunto de puntos de visión.

2.2.4 *Ambientes Virtuales*

En años recientes, el auge de las tecnologías de *Realidad Virtual* ha ido acrecentándose. Cada vez es más fácil tener acceso a los medios que permiten generar simulaciones e interactuar con ellas, lo cual ha permitido mejorar los sistemas de información y además permite realizar investigaciones en ambientes controlados e ideales en disciplinas tan variadas como la ingeniería, la robótica, la medicina y la arquitectura, entre otras. Por tal motivo, se hace un esfuerzo cada vez mayor por desarrollar escenarios virtuales cada vez más realistas y sofisticados.

Son las áreas de la Computación y de la Visualización las que se encargan de los avances más importantes en este campo, puesto que la Realidad Virtual es una representación tridimensional de un ambiente no real, generada mediante una simulación computarizada. Además, la Visualización en tres dimensiones cada vez se convierte más en una necesidad, dado que permite a los seres humanos observar patrones, relaciones y tendencias que de otra forma sería difícil comprender.

En el campo de la robótica se han hecho avances grandes en la creación de modelos de robots cada vez más hábiles y poderosos. Simplemente en el campo de la medicina y en particular en rehabilitación, se han desarrollado robots que tienen mayor grado de libertad o que son capaces de levantar con mayor facilidad pacientes que necesitan ser movidos o ayudados. Sin embargo, no sólo el desarrollo técnico es lo que importa y para poder entrenar estos robots es necesario hacer muchas pruebas que necesitan la interacción con personas, lo cual puede ser muy cansado o hasta peligroso para los sujetos de prueba. No sólo eso, siendo tecnologías tan nuevas, se implica también que su costo es muy elevado, por lo que hacer pruebas de entrenamiento una y otra vez tiende a desgastar las partes del robot, con lo que la inversión necesaria puede aumentar mucho [56]

Es por tal motivo que la interacción de la robótica con el campo de la Realidad Virtual y la Simulación es tan importante, pues permite realizar simulaciones tanto de los robots como de los ambientes, cada vez más realistas y por lo tanto más útiles para el desarrollo y la investigación.

La Realidad Virtual y el desarrollo de los simuladores van muy de la mano con el desarrollo de la aviación. Los primeros simuladores se crearon para tener un método más seguro para el entrenamiento y las pruebas de los aeroplanos en la Segunda Guerra Mundial, hecho que deja ver dos lecciones muy importantes: la primera es que la Realidad Virtual y la necesidad de ambientes inmersivos de entrenamiento son

más antiguos que la revolución de las computadoras. La segunda es que la Realidad Virtual como simulación de la realidad ha sido un factor indispensable en el desarrollo de las tecnologías que simula.

Fue hasta los años 90 que la tecnología computacional se encargó del desarrollo de los simuladores, y fue ahí que la necesidad de simulaciones realistas empujó el avance en el desarrollo de las tecnologías gráficas, especialmente para la visualización de detalles geográficos en tiempo real. De ahí que ha habido toda una carrera por desarrollar las tarjetas de video y demás elementos que permitan mejorar la graficación por computadora, que ha ido de la mano con el desarrollo de las tecnologías de la computación gráfica y la visión computacional [34]

Pero más allá del ámbito visual, las simulaciones de Realidad Virtual se han utilizado para desarrollar muchas de las investigaciones en robótica. Por ejemplo, se han utilizado simuladores de realidad virtual para simular el caminado adaptativo en terrenos no estructurados para robots con múltiples patas, lo cual permitió generar un método rápido de validación de diversas características de los modelos matemáticos usados para el caminado, identificando errores potenciales en fases tempranas del desarrollo de los robots. En estos casos, es posible identificar y remover errores modificando el modelo de la simulación, lo cual permite ahorrar en tiempo y en recursos. [5].

Un punto importante en nuestro método es la capacidad de simular cualquier cámara que utilice un robot, de tal forma que no sea dependiente de una cámara en particular. Para poder hacer esto en necesario calcular los parámetros de calibración de la cámara, o parámetros intrínsecos.

Normalmente se utilizan patrones de calibración para estimar los parámetros intrínsecos de la cámara, que tienden a ser muy parecidos al patrón de un tablero de ajedrez. En robótica es necesario usar marcos de calibración pequeños en comparación con los usados en fotogrametría, por lo que conviene situarlos de tal forma que abarquen todo el espacio posible, puesto que este tipo de marcadores tienden a fallar al predecir los componentes de la pose que quedan afuera del plano.

Una forma simple de realizar la calibración de una cámara es ondear un patrón de calibración en frente de la cámara y tomar una secuencia de imágenes. La pose del patrón debe ser calculada en conjunción con los parámetros intrínsecos. Cada imagen se utiliza para calcular una homografía \hat{H} que mapee los puntos del plano de calibración $(X_i, Y_i, 0)$ a coordenadas de la imagen (x_i, y_i) ,

$$x_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \sim \mathbf{K} \begin{bmatrix} r_0 & r_1 & t \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \sim \hat{\mathbf{H}} p_i, \quad (13)$$

donde r_i son las dos columnas de la matriz rotacional R de la cámara y \sim indica igualdad a escala. En [58] se muestra como recuperar de aquí la matriz K de calibración que contiene los valores intrínsecos.

Una vez teniendo los parámetros de calibración de la cámara, es posible utilizarlos en una simulación para obtener imágenes similares a las que se obtienen por la cámara con la que se realizaron las pruebas. Aún más, con éstos valores es posible modificar las imágenes tomadas con la cámara real para eliminar la distorsión radial de su lente.

ESTRATEGIA METODOLÓGICA

Una vez habiendo establecido las motivaciones y la importancia de realizar este trabajo, podemos centrarnos en el problema a resolver: dotar a un robot que juega fútbol de un método que le permita localizarse a sí mismo cuando está dentro de la cancha, utilizando únicamente los sensores que se tienen para emular los sentidos humanos. Como ya se mencionó con anterioridad, durante las últimas competencias en que participó el equipo de DotMX, no se contó con ningún medio de localización para los robots; éstos dependían de la ubicación de la pelota y las porterías mediante métodos de segmentación de la imagen, pero esto sólo les permitía tener una relativa orientación de hacia donde estaban mirando y por lo tanto sus estrategias eran muy básicas.

En este capítulo vamos a hablar acerca del método que propusimos para resolver este problema, así como de las herramientas utilizadas y construidas para probar el método e implementarlo en un sistema compatible con el sistema operativo de los robots con los que se cuenta.

3.1 PLATAFORMA ROBÓTICA Y HERRAMIENTAS EXTERNAS

Aquí se va a hablar de cuáles fueron las herramientas utilizadas para implementar el método de localización -herramientas que se usan para desarrollar algoritmos de la visión computacional, ambientes virtuales y simulaciones-, así como de la plataforma robótica que se utilizó como base para implementar el método.

3.1.1 *OpenCV*

Cuando se hace investigación en el área de la visión computacional, se necesita una librería básica que contenga los elementos necesarios para llevar a cabo el procesamiento digital de las imágenes, que sea capaz de manejar los diferentes tipos de formatos en que se necesita trabajar una imagen, que permita manipular la imagen de forma matricial, dando acceso a los valores de cada pixel para poder implementar filtros y demás algoritmos. Aunque es posible hacer esto manualmente, se ha optado por utilizar una librería de acceso abierto, que cumple con los requisitos anteriores y además trae implementados muchos algoritmos básicos, lo que permite que el enfoque esté en el método y no en los algoritmos modulares.

La librería que usamos se llama OpenCV, y es una librería de visión computacional de código abierto, lo que permite no sólo utilizar muchos métodos y algoritmos preprogramados, sino también editar el código de los mismos para adaptarlo a las necesidades propias.

3.1.2 *Blender*

Se necesitaba un programa que permitiera generar un ambiente virtual que tuviera los elementos invariables de la cancha de fútbol, para generar con éste las vistas sintéticas de la base de datos.

También se usó porque, a falta de un robot con el que hacer las pruebas, se decidió generar un segundo ambiente virtual, que tuviera todos los elementos necesarios para producir imágenes que simularan apropiadamente las imágenes generadas por la cámara de un robot. Este ambiente cuenta con modelos de la canchas, de las porterías, de elementos que rodean la cancha y objetos que pudieran generar oclusiones, todo con colores, materiales, texturas y luces acordes a los objetos simulados.

Por este motivo fue que se utilizó Blender, un programa de animación y modelación 3D, de código abierto y licencia GPL que es una licencia libre que permite el uso del software, así como su modificación y distribución. Se trata de un programa de uso profesional, cuyo proyecto se encuentra mantenido por una comunidad de desarrollo que se sostiene a base de donaciones, pero además otorga a cualquier usuario la posibilidad de generar cambios o programar subrutinas y enviarlas a la comunidad para ser evaluadas y, en el mejor de los casos, agregadas en una versión posterior. De hecho, está programado de tal forma que resulta posible programar los comportamientos necesarios mediante scripts de Python, lo que permite integrarlos sin necesidad de recompilar el código. Para éste trabajo se programaron varios scripts que permitieran, una vez teniendo el modelo de la cancha y demás elementos, configurar la cámara de tal forma que emulara el sistema visual del robot y conseguir las tomas necesarias para probar el método desarrollado.

Como mucho otras softwares de creación y manipulación de ambientes tridimensionales, Blender tiene la capacidad de generar imágenes foto-realistas y modelos precisos tridimensionales, utilizando muchos algoritmos de la visión computacional.

3.1.3 *Darwin-OP*

Además de una herramienta para la generación de los modelos y ambientes virtuales, era necesario tener un robot que sirviera como referencia, a fin de poder usar sus dimensiones y demás atributos para asignarle valores a la simulación y tener un punto de comparación. Por tanto se decidió hacer uso del Darwin-OP, un robot que

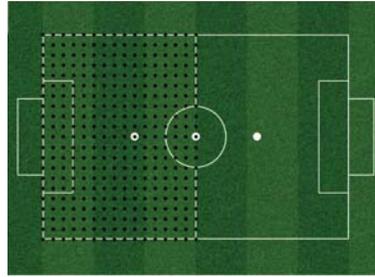
fue utilizado durante las competencias del año 2012 y que, durante el desarrollo de este trabajo, el equipo DotMX trabajaba en construir y conseguir más robots de este tipo para utilizarlos en lo sucesivo. El sistema para implementar el método de localización desarrollado se programó de acuerdo con las especificaciones de esta plataforma, sin embargo, el método es fácilmente extensible a robots de otros modelos de los cuales se cuente con la información necesaria.

El Darwin-OP es una plataforma abierta diseñada por la Universidad de Pensilvania, la Purdue University y el Virginia Tech, en conjunción con la empresa ROBOTIS, la cual se especializa en la manufactura de robots, lo que implica que el diseño, especificación y manuales están accesibles para cualquiera en la página de internet del proyecto. Es con este modelo con el que el equipo de dicha universidad, el Team Darwin, que ha ganado las últimas 2 copas de la categoría KIDSIZE, y fue a partir de la primera copa ganada, que se abrió el proyecto para que otros equipos de investigación pudieran usar su plataforma como base implementando sus propias mejoras. Lo anterior con la idea de avanzar abiertamente para alcanzar la meta que se tiene en común.

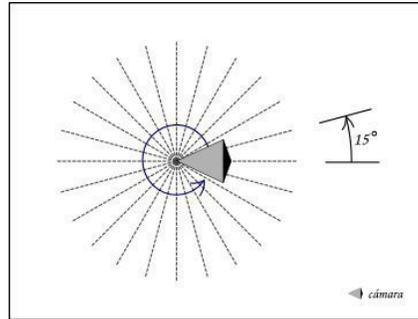
Cabe mencionar que existe un simulador llamado *webots* que sirve para realizar simulaciones de robots, y de hecho también existe en línea una simulación del robot Darwin-OP que es libre. Sin embargo, aunque se trata de un buen simulador de comportamientos y de física, no se enfoca en los aspectos visuales, razón por la cual se consideró que era necesario desarrollar una simulación propia enfocada en el sistema visual del robot y no en su sistema motriz, pues es con el primero con el que trabaja el método aquí desarrollado.

3.2 HERRAMIENTAS EXTRAS

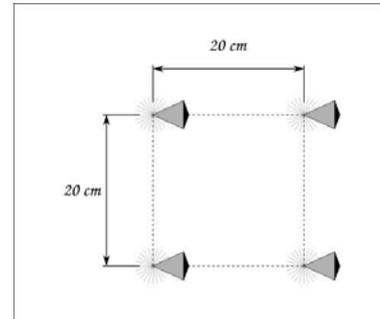
En esta sección se describirán las herramientas que se tuvieron que generar para poder programar el sistema posteriormente haciendo uso de ellas. Dado que el método desarrollado se sirve de una base de datos que contiene la información de las vistas sintéticas y sus respectivas poses, había que generar un modelo tridimensional de la cancha que únicamente tuviera los elementos invariantes de la misma. Las vistas sintéticas se generaron distribuyendo homogéneamente las poses de la cámara, i.e. se estableció una posición cada 20 cm en la dirección del eje X, y cada 20 cm en dirección del eje Y, y desde cada posición se generó una vista sintética cada 15° . Sin embargo, sólo se tomaron vistas en una mitad de la cancha (contando la línea de media cancha), dada la simetría de la cancha y el hecho de que las porterías son indistinguibles, por lo que al final la base de datos quedó conformada por 8064 vistas sintéticas correspondientes a poses en la cancha. En la figura 2 se muestra un diagrama de cómo se tomaron las poses de la base de datos.



(a) Poses distribuidas



(b) Punto de giro



(c) Distancia entre puntos de giro

Figure 2: Poses desde las que se generaron las vistas sintéticas

3.2.1 Modelo de Referencia

El modelo generado en Blender es un modelo tridimensional de los elementos invariantes de la cancha con las medidas oficiales que aparecen en el reglamento de la RoboCup de la categoría KidSize [42]. En éste, se especifica que la cancha debe medir 400×600 cm, que el ancho de las líneas de la cancha es de 5 cm y que las porterías deben medir 80×150 cm con un diámetro de 10 cm. Por su parte, la cámara de la simulación se ajustó con un ancho focal de 35 mm y el tamaño del sensor a 32 mm, aunque estos son sólo valores de referencia que se deben ajustar dependiendo de la cámara que se desee emular.

3.2.2 La Base de Datos

Una vez teniendo el modelo, se le aplicó un filtro que únicamente renderiza sus bordes, generando así imágenes binarizadas de la cancha. En la figura 3 se muestra el modelo tridimensional de la cancha utilizado en este trabajo, así como una de las imágenes obtenidas con el mismo. Todas las imágenes de la base se generaron en formato PNG¹, formato de compresión sin pérdida con una resolución de 640×480 píxeles.

¹ Portable Network Graphics

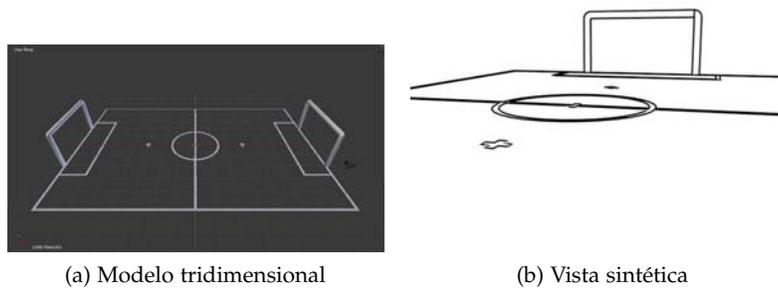
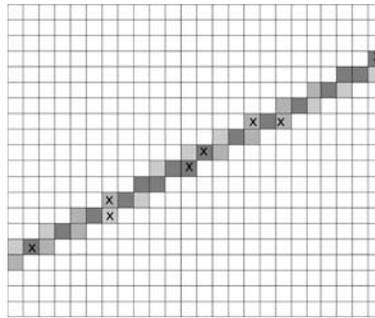


Figura 3

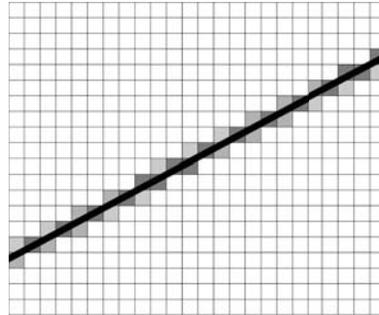
Sin embargo, con la intención de optimizar el acceso a la base de datos, se pensó que en lugar de utilizar imágenes de referencia en las que se tendrían que tomar en cuenta 307200 píxeles por cada una de las 8064 imágenes, era posible extraer puntos de referencia que fueran representativos de cada pose. Para esto se aplicó a cada imagen el algoritmo de Hough Probabilístico, el cual extrae los segmentos de recta que conforman cada imagen. Nótese que la utilización de imágenes generadas con un modelo y renderizadas con únicamente los bordes ayuda a la aplicación del algoritmo de *Hough Probabilístico*, el cual exige tener una imagen binarizada, que tenga únicamente los bordes para poder aplicar los algoritmos de votación. Pero este paso se evitó al haber renderizado las imágenes del modelo con únicamente los bordes, lo cual también permitió obtener mejores resultados que cualquier método de extracción de bordes, pues a diferencia de lo que ocurre con aquellos, los bordes generados de este modo son precisos y no se produce ningún tipo de ruido.

Una vez obtenidos los segmentos de rectas que forman las imágenes de la base, surgió el problema de que, aplicando el algoritmo de Hough Probabilístico a una imagen que está conformada por píxeles con coordenadas discretas y no continuas, cuando se aplica el algoritmo de votación no es posible encontrar un segmento de recta que contenga todos los puntos, por lo que lo que regresa segmentos de rectas traslapados, a los que –aunque teóricamente pertenecen a la misma línea– se les asignan segmentos con pendientes ligeramente diferentes. Obsérvese la figura 4, en la que se puede ver que en la imagen de bordes 4a, todos los píxeles grises corresponden a una misma línea, y es la línea negra de la figura 4b la recta que idealmente representaría a esos puntos. Si se diera el caso en que durante la votación se impusieran los píxeles marcados con una X, entonces se podría observar que no hay una línea que los contenga a todos, por lo que se generarían segmentos sobrepuestos como los que se ven en la figura 4c.

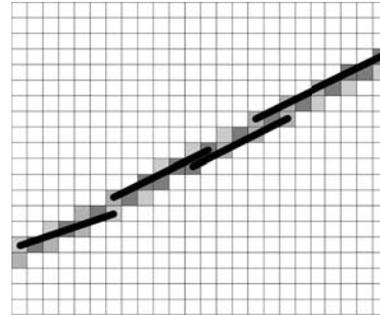
Para solucionar este problema, se desarrolló un procedimiento heurístico en el cual se van tomando cada uno de los segmentos que en-



(a) Imagen de bordes



(b) Línea ideal



(c) Segmentos traslapados

Figura 4: Traslape de los segmentos del algoritmo de Hough Probabilístico

traga el algoritmo de Hough Probabilístico y se recorren de uno de sus extremos al otro: se toma el primer extremo como el primer punto de control y luego se recorre a lo largo del segmento una distancia específica, luego se toma el siguiente punto, repitiendo hasta llegar al otro extremo. Sin embargo, se puede dar el caso en que dos segmentos formen parte de la misma recta, así que para determinar si éste es el caso, se toma en cuenta que las imágenes son discretas y por lo tanto es posible asignar valores de tolerancia, esto es, se sabe que a cada segmento le corresponde la ecuación de una recta y por tanto se puede saber en qué puntos cruza esa recta con los bordes de la imagen. Así, es posible comparar qué tan cerca cruzan las rectas correspondientes a los segmentos que se están verificando, con los bordes de la imagen. Entonces se deben tomar en cuenta tres factores: qué tan cerca cruzan las rectas con el borde derecho, qué tanto respecto al lado izquierdo, y qué tanto difieren los ángulos de las pendientes. Si estas diferencias son menores respectivamente a umbrales previamente establecidos, se puede considerar que los segmentos forman parte de la misma recta.

Para este desarrollo, se seleccionaron dichos umbrales utilizando una muestra de 100 imágenes que únicamente contenían una línea con las mismas características que las líneas generadas por el modelo, y después de aplicar el algoritmo de Hough probabilístico, se midieron las distancias entre los puntos de cruce, así como las diferencias

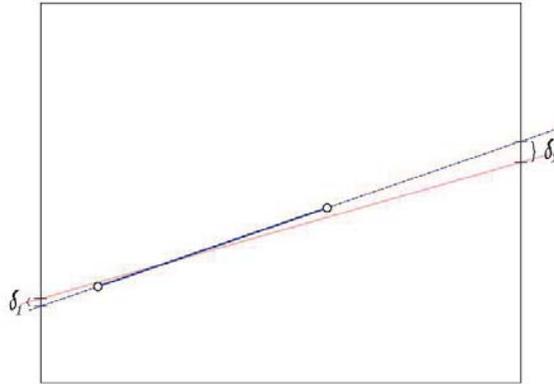


Figura 5: Para determinar si el segmento azul corresponde a la línea roja (que a su vez corresponde a un segmento analizado previamente), se extiende y se buscan dónde cruza con los bordes de la imagen. Se consideran la misma línea si las distancias δ_1 y δ_2 son menores a un valor preestablecido.

entre las pendientes, tomándose los valores por encima de la media. En la figura se muestra un diagrama de cómo se obtienen estos valores.

Habiendo determinado si dos segmentos forman parte de la misma recta, es sencillo saber si se traslapan, pues basta verificar si ambos extremos de un segmento son menores o iguales (o mayores o iguales) que ambos extremos del otro segmento. En caso de que los segmentos se traslapan, se toman los puntos control del primero de los segmentos, y del segundo sólo se toman los puntos de las partes del segmento que no se sobreponen al anterior. De esta forma se obtienen los puntos de control de una imagen, que son puntos que van a caracterizar el conjunto de líneas que conforman los bordes de los elementos invariantes de la cancha dependiendo del punto desde donde se tome la imagen. Cuando se habla de puntos de control, se hace referencia a puntos que parametrizan la imagen, están formados por coordenadas X y Y, y además tienen asignada la dirección normal a la pendiente del segmento de recta de donde fueron obtenidos.

Finalmente, teniendo las imágenes parametrizadas mediante sus puntos de control, en lugar de tener que tomar en cuenta 307200 píxeles en las vistas sintéticas, solamente se toman en cuenta los puntos de control; en el presente caso, la imagen con mayor número de puntos de control tiene 244 puntos de control, y mientras que el valor promedio es de 70.

La base de datos se configura de tal forma que cada una de las vistas sintéticas contiene un apuntador a la vista correspondiente, a un giro de 30° hacia la izquierda y otro a la derecha. Estos apuntadores

se utilizan durante la fase de filtrado lateral explicado en la sección 3.5.

3.2.3 *Simulación y Ambiente Virtual*

Como se explicó anteriormente, al no contarse con un robot físico, y considerando que se buscaba desarrollar un método generalizado, se determinó que la mejor opción era utilizar un simulador. En este caso se desarrolló una segunda simulación del ambiente virtual en la que se desempeña el robot, que además permitiera manipular los elementos y características del sistema de visión de cualquier robot cuyo elemento principal es una cámara.

Para esto se creó una simulación en Blender que pudiera generar imágenes cuya apariencia fuese lo más realista posible. Esta simulación fue diseñada utilizando dimensiones proporcionales a las de los objetos del mundo real que en ella aparecen. Cada elemento se modeló utilizando las texturas y los colores correspondientes, además de que se les asignaron las propiedades de los materiales que los componen, de tal forma que interacción con la luz fuera lo más semejante a las imágenes obtenidas por un robot real, pues una vez que se renderiza una imagen bajo estas condiciones, podemos observar los mismos comportamientos que en una cancha real. Así, al emular los comportamientos de los reflejos de la luz en los materiales y la forma que se proyectan las sombras, nos permite tener imágenes más similares a las que va a tomar un robot, pues genera el mismo ruido en la imagen que un ambiente real. Particularmente dado que el método desarrollado tiene como base la localización correcta de los elementos invariantes en el campo utilizando un extractor de bordes, es importante tomar en cuenta las luces y sombras que puedan dificultar el desempeño del método al presentarse bordes espurios.

Los elementos que componen la simulación son:

- El piso de la cancha que, según las reglas oficiales, cuenta también con una zona externa a la línea del contorno, que es dónde se sitúan los árbitros y acomodadores, y también es por donde entran los robots cuando el partido está transcurriendo. Tiene el color y la textura del pasto sintético utilizado en las competencias oficiales, y utiliza un material difuso sin brillos ni reflejos.
- Las líneas de la cancha son similares a las utilizadas en la primera simulación, pero tienen el color y la textura de las cintas que se utilizan para marcar las líneas de la cancha en las competencias oficiales, y el material del que están hechas es difuso y sólo genera brillos bajo luz reflejada directamente hacia la cámara.
- Las porterías se modelaron con base en modelos utilizados en competencias anteriores. Ambas son amarillas según el nuevo reglamento y el material que utilizan es difuso, pero tiene un



Figura 6: Ambiente virtual

toque de lustroso, lo que proyecta reflejos de luz desde diversas partes de la cancha, que generan ruido en las imágenes. Véase Figura 7.

- Se agregaron vallas contenedoras como las que se utilizan normalmente en las competencias, pues aun cuando éstas no están reglamentadas, se usan comúnmente para delimitar el espacio de juego y proteger a los robots. Como estos elementos no tienen formas ni medidas específicas, se generaron 3 diferentes tipos de vallas con distintos materiales, texturas y estampados que permitieran simular diferentes problemas a los que podría enfrentarse el método.
- Para probar la robustez del método con respecto a las oclusiones, se utilizó un modelo de referencia que se emplea frecuentemente al realizar pruebas con Blender, pues es un modelo poligonal con un número significativo de caras, lo que permite probar rápidamente comportamientos de colores, texturas, luces y sombras, entre otras cosas; tiene la forma de la cabeza de una chimpancé y se llama Suzanne. En éste caso se empleó como si se tratara de otro robot en la cancha, para ver cómo se comportaba el método al tener dentro del campo visual un objeto que ocluyera los elementos invariantes de la cancha. El material asignado a este modelo fue el correspondiente al material que utiliza el DarwinOP en su carcasa. Véase Figura 8.
- La iluminación se generó simulando lámparas de área originadas en diversos puntos del techo, pero también se empleó ilu-

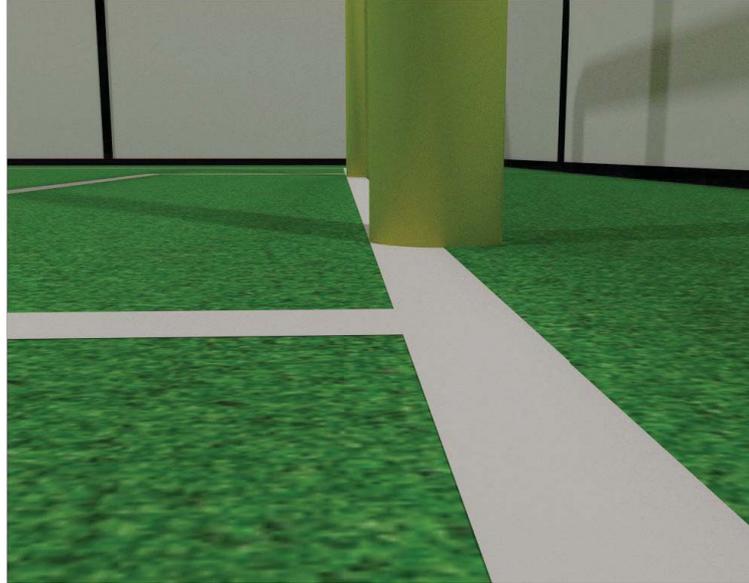


Figura 7: Elementos de la cancha virtual: piso de alfombra verde, línea de cinta plástica blanca, y porterías de PVC.

minación ambiental, que depende del fondo de la escena para calcular la intensidad y tono de la luz al renderizar una imagen. Esto permitió confrontar al método con condiciones semejantes a las que se presentan en el mundo real con respecto a los cambios de iluminación e intensidad provocados por fenómenos naturales o artificiales. Para esto se utilizaron 53 diferentes imágenes de fondo tomadas de salas o bodegas similares a las que se usan para realizar las competencias.

Gracias a esta simulación, se pudo contar con una herramienta con la que generar imágenes con una calidad y semejanza suficiente para emular imágenes tomadas por la cámara de un robot real.

3.3 MÉTODO DE LOCALIZACIÓN

El método de localización aquí propuesto tiene dos requisitos. Uno es que el robot debe contar con una cámara, ya sea a color o en blanco y negro. El otro es que el entorno en que se vaya a desempeñar el robot, sea un entorno conocido, como en este caso es la cancha de fútbol, cuyas dimensiones y proporciones se conocen de antemano permitiendo la construcción de un modelo tridimensional con el que generar una base de datos como la que se explicó inicialmente.

Cabe mencionar que otros métodos [19, 16, 46] de localización usan algoritmos de extracción de características puntuales como SIFT o FAST, que han demostrado un buen desempeño frente al problema del SLAM y del rastreo de la posición de un robot. Sin embargo, en este trabajo se buscó desarrollar un método alternativo pensando en los

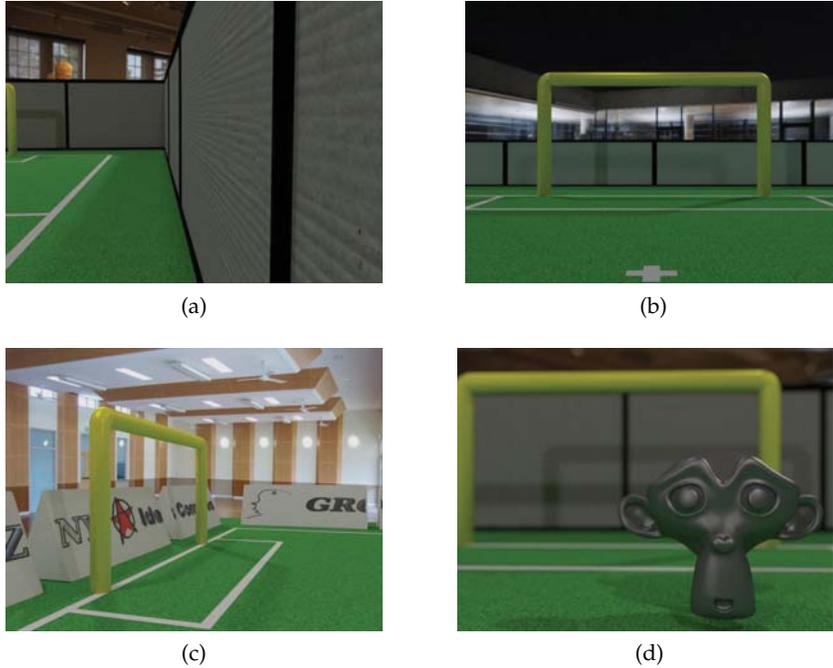


Figura 8: (a - c) Diferentes modelos de vallas utilizados. (d) Suzanne

siguientes aspectos. Este tipo de extractores, obtienen características puntuales, las cuales pueden describir pequeñas regiones de objetos, pero carecen de la habilidad de describir propiedades como las estructura de los objetos que representan, y puesto que se está tratando con un ambiente conocido – la cancha de futbol robótico – se buscó desarrollar un método que permitiera aprovechar esa información. Por otro lado, dadas las variaciones que se dan entre dos canchas cualesquiera, es importante tener un método que sea capaz de realizar un preprocesamiento de los elementos que caracterizan cada una de las vistas en la base de datos, de tal forma que no sea necesario realizar un gran esfuerzo para generar una base de datos adecuada a cada cancha en que se juega, o con cada variación importante que se presente, como un cambio de luz como consecuencia de la aglomeración de gente que se reúne al inicio de cada partido. Por último, aún cuando los extractores de características reducen significativamente el número de comparaciones que se harían en una búsqueda de fuerza bruta para comparar una vista con una imagen, de todas formas necesitan de un gran número de descriptores – regularmente parches – para poder realizar un emparejamiento óptimo, por lo cual se buscó que el método aquí desarrollado hiciera uso del menor número de descriptores posible, haciendo que el procesamiento sea rápido, pero que a su vez presentara resultados aceptables.

El método aquí presentado está inspirado en el trabajo de Gee & Mayol-Cuevas [24], en el que presentan un método de relocalización

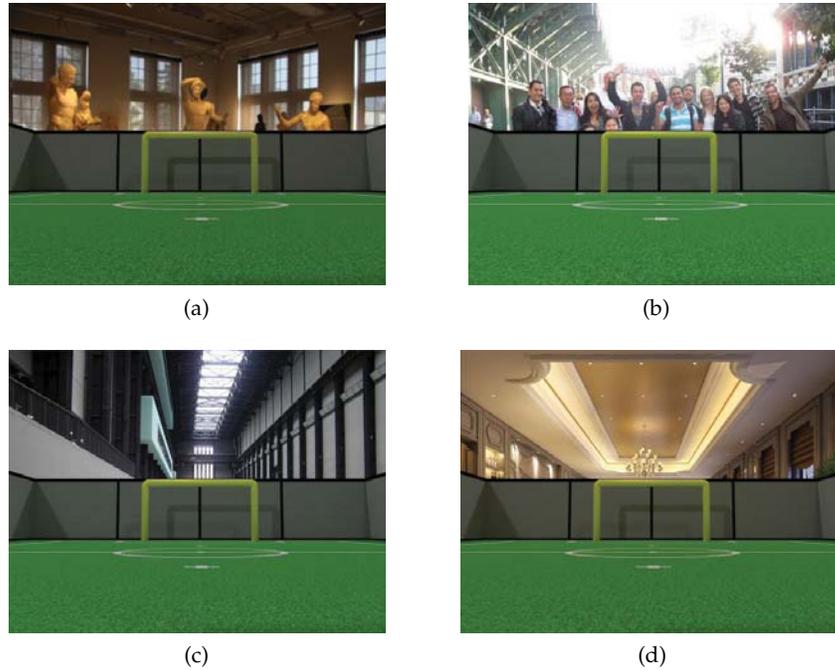


Figura 9: Ejemplo de fondos utilizados para probar el cambio de intensidad de la luz y problemática de líneas espurias. Nótese que no es relevante para los fines de el cambio de fondo, que las perspectivas coincidan.

6D para cámaras RGBD empleando una regresión general sobre un conjunto de vistas generadas sintéticamente. En dicho trabajo, las vistas sintéticas son generadas mediante interpolación de cuadros tomados de la trayectoria de mapeo, tomando en cuenta que son poses con gran probabilidad de que cerca de ellas se aplique la relocalización. En el caso del presente trabajo, no existen trayectorias con percepción definida para el robot, por lo que era necesario generar un modelo que permitiera generar las vistas sintéticas desde cualquier punto dentro de la cancha, y viendo en cualquier dirección. Sin embargo, esto implica que las vistas sintéticas tendrán una naturaleza distinta a las imágenes capturadas por el robot, puesto que las primeras son renderizaciones de los bordes de los elementos invariantes del modelo, y además, una vez preprocesadas, las vistas se parametrizan en puntos de control, por lo que era necesario ajustar el tipo de regresión que se debe aplicar entre las imágenes y las vistas.

La idea principal del método es que, en un momento dado, el robot está en la cancha y necesita saber dónde se encuentra. Con este fin, toma una imagen con su cámara y la procesa de tal forma que pueda compararla con las vistas sintéticas que se tienen en la base de datos. Para esto es necesario definir una métrica que permita establecer una medida de semejanza entre una vista sintética y una imagen tomada

por la cámara del robot. Así, es posible asignar un valor de semejanza a cada vista con respecto a la imagen de la cámara y posteriormente ordenarlas para entonces tomar aquellas cuya semejanza sea mayor, extrayendo de ahí la vista sintética más parecida a la imagen de la cámara para determinar la pose correspondiente al lugar desde donde se encuentra el robot al momento de tomar la imagen.

Para definir la métrica que compara una vista sintética con una imagen de la cámara, se toma la imagen, se le sobreponen los puntos de control de la vista sintética, y se busca cuantificar las coincidencias entre los puntos de control y los elementos invariantes de la cancha presentes en la imagen.

Para lo anterior, es necesario aislar de alguna forma dichos elementos, lo cual comúnmente se hace a través de algún tipo de segmentación por colores, pero esto conlleva muchas dificultades cuando se trata de un ambiente tan dinámico con respecto al cambio de intensidad de las luces. Por tal motivo se decidió utilizar un extractor de bordes de Canny, con lo que se logró sobreponer los puntos de control directamente a la imagen de bordes y establecer una área de búsqueda alrededor de los puntos de control para ver si dentro de ésta se encuentran bordes en la imagen binarizada. De esta forma, conforme mayor es la correspondencia entre la imagen y la vista sintética es perfecta, los puntos de control caen de manera regular muy cercanos a los bordes y se puede establecer entonces un valor de semejanza grande.

La métrica que se propone aquí consiste en que una vez que se tiene la imagen binarizada de bordes, y los puntos de control se sobreponen a ella, para cada punto de control se analizan las líneas de escaneo respectivos, i.e. cada punto de control tiene registro de la dirección normal al borde a que corresponde, por lo que podemos usarla para trazar una línea perpendicular al borde que va del punto de control, hasta una distancia establecida de antemano δ , en ambas direcciones. Por cada línea de escaneo, se revisa en dirección a la normal, cuál es el primer pixel que corresponda a un borde de la imagen y se registra la distancia complementaria desde el punto de control hasta dicho pixel, es decir, δ menos la distancia encontrada, o en caso de no haber coincidencias, se asigna un valor de cero. Después se repite el proceso en dirección contraria, y finalmente se comparan ambas direcciones, y se selecciona el valor mayor, el cual se suma a un acumulador. Después de revisar cada punto de control, el acumulador tendrá la suma total de las distancias complementarias encontradas. Este valor lo dividimos entre delta multiplicada por el número de puntos de control que conforman la vista sintética. Así, si existe una correspondencia perfecta entre la imagen y la vista, obten-

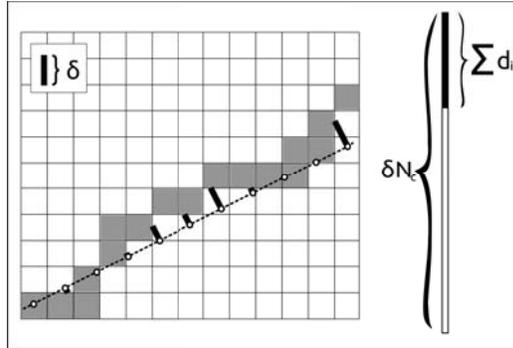


Figura 10: Diagrama de la métrica Φ . Se sobreponen los puntos de control sobre la imagen binarizada, y se busca la distancia que hay entre cada uno y el pixel del borde más cercano, sin sobrepasar el tamaño de δ . Φ Es la razón que hay entre el complemento de la suma de las distancias entre el máximo alcanzable.

dremos un valor final de 1, en lo que resulta de la fórmula final de nuestra métrica, la cual hemos denotado con Φ :

$$\Phi = \frac{\sum_{i=1}^n (\delta - \hat{d}_i)}{\delta N_c} \quad (14)$$

donde \hat{d}_i es la distancia complementaria del i -ésimo punto de control, y N_c el número total de puntos de control de la vista sintética. Véase la Figura 10.

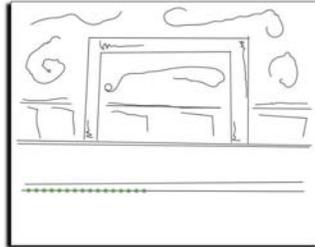
En este punto, nuestra métrica asigna valores entre 0, en el caso en que la vista no tenga nada que ver con la imagen, y 1, cuando la correspondencia es perfecta. Al utilizar esta métrica y analizando los resultados que arrojaba, observamos que la métrica se enfrentaba a dos grandes dificultades.

PROBLEMA 1 DE Φ : El primer problema al que nos enfrentamos es que los puntos de control correspondientes a un segmento de recta, pueden regresar un valor cercano a 1 si coinciden regularmente con un borde, aún cuando el tamaño de éste exceda por mucho el tamaño del segmento comparado. Véase figura 11²

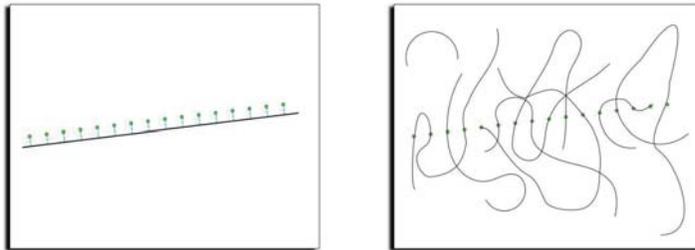
PROBLEMA 2 DE Φ : El segundo problema se da cuando se asignan buenos valores al encontrar distancias complementarias grandes entre puntos de control y una imagen con muchos bordes que no corresponden a los segmentos con los que se comparan. Véase la figura 11 donde se observa que en la figura de la izquierda se presenta una coincidencia regular además de que

² Con *coincidencia regular* nos referimos al caso en que al comparar los puntos de control de una vista con un borde, las distancias complementarias son similares en cada punto, implicando un cierto nivel de paralelismo entre el borde y el segmento correspondiente a los puntos de control.

hay coincidencias totales, por lo que el valor de su métrica debe ser mayor que la imagen de la derecha, que aunque también presenta coincidencias totales y la distancia complementaria es mayor, está compuesta por bordes que no corresponden al segmento.



- (a) Problema 1 de Φ : Los puntos verdes sobre la imagen de bordes representan una vista sintética. Al aplicar Φ se obtiene un valor alto dada la coincidencia regular, sin embargo la vista no corresponde a la imagen.



- (b) Problema 2 de Φ : La imagen de la izquierda obtiene un valor menor de semejanza que la de la derecha, aún cuando la correspondencia entre la imagen y la vista es grande. La imagen de la derecha obtiene un valor muy grande debido a que todos los puntos de control de la vista sintética coinciden con bordes en la imagen, sin embargo no hay relación entre la vista y la imagen.

Figura 11

Para enfrentar los problemas arriba descritos, buscamos otros factores que pudieran combinarse con el valor de Φ , y que mejoraran su desempeño. Encontramos que cuando la imagen binarizada corresponde a la vista, además de que la distancia que hay entre ellas debe ser pequeña, deben también ser pocos los casos en que una línea de escaneo no encuentre una coincidencia con algún pixel. Puede darse el caso en que una vista corresponda a la imagen con que se está comparando, pero dado que las vistas están sujetas a poses discretas en la cancha, aún la vista más parecida a la imagen puede estar desplazada

una distancia significativa debido a la traslación del punto de vista de la cámara con respecto a la pose de referencia de la vista sintética, con lo cual se obtendría un valor pequeño de Φ , pero si tomamos en cuenta que cada punto de control encontraría una coincidencia con un borde, la métrica debe asignar un valor grande en casos así. Por tal motivo se tomó en cuenta la proporción de puntos de control que coinciden con la imagen de bordes, declaramos un valor α que definimos como

$$\alpha = \frac{c}{N_c} \quad (15)$$

donde c es el número de coincidencias entre algún punto de control y un borde, y N_c es el número de puntos de control en la vista sintética. Este valor mejora los resultados de la métrica al enfrentarse a situaciones que tienen que ver con el primer problema de Φ .

Para hacer frente al segundo problema utilizamos un segundo valor que está en función de los cambios de tendencia de las distancias complementarias que se dan al comparar un segmento con un borde. *Decimos que hay un cambio de tendencia cuando al ir registrando las distancias complementarias se da un cambio en el signo de las mismas, lo que implica que hay un cruce entre el segmento de la vista sintética y un borde de la imagen.* Mientras más cruces existan, aumenta la probabilidad de que se trate de bordes que no corresponden a líneas del campo o de las porterías, o que no se trate de una sola línea. Definimos este valor como

$$\beta = \frac{s}{N_c - 1} \quad (16)$$

donde s es el número de cambios de tendencia, y $0 \leq s \leq N_c - 1$.

Para probar el desempeño de estos dos valores y la importancia de combinarlos con la métrica se llevaron a cabo las pruebas que se describen en la sección 3.5, donde se puede observar en la tabla 1 que al combinar todos los elementos antes descritos, obtenemos valores de semejanza bajos para aquellas imágenes que no corresponden a las vistas, lo cual es una ventaja aún cuando los valores de semejanza asignados a las correspondencias perfectas entre vistas e imágenes sean menores.

Esta situación se da porque aún cuando una imagen sea tomada desde la pose exacta de donde se genere una vista sintética y se comparen con cualquier métrica, la imagen de cámara contiene elementos impredecibles, igual que las condiciones del entorno, de tal forma que dos imágenes captadas desde el mismo lugar y tomadas en tiempos distintos siempre van a variar aunque sea mínimamente, por lo que sus imágenes de bordes correspondientes también serán distintas, y por consecuencia, su valor de semejanza al compararse con una misma vista.

Otra situación que imposibilita el poder conseguir un resultado único y sin posibilidad de fallas al realizar una regresión de este tipo es que no es posible evitar el llamado problema del barbero [47],

```

% *****
% ALGORITMO DEL FILTRO FRONTAL
% *****

1. Revisar la imagen de entrada; si está vacía, termina.
2. Extraer los bordes de la imagen.
3. Para cada vista de la base de datos:
4.     Extraer lo puntos de control de la vista.
5.     Calcular y almacenar la semejanza entre la vista y la
        imagen, usando la métrica Phi.
6. Ordenar las vistas de acuerdo a los valores de semejanza
        recién calculados.
7. Regresa la lista de vistas candidatas y termina.

```

también conocido como *El Problema de la Apertura*, que se presenta, puesto que es posible generar dos vistas sintéticas cuya representación visual sea muy similar o hasta idéntica en algunos casos, pero las poses desde donde se generan tales vistas sean diferentes. Como estos problemas no se resuelven modulando más la métrica, lo que hicimos fue emplear la idea central de los filtros de partículas, pero adaptándola a nuestro caso. Para esto se diseñaron tres fases de filtrado.

La primera fase se denominó *Filtro Frontal*, y consiste en tomar una imagen de la cámara del robot alineando la vista de la cámara con la orientación del cuerpo del robot. Con la métrica Φ podemos asignar un valor de semejanza a cada una de las vistas de la base de datos con respecto a la imagen de la cámara, pero no es posible asegurar que la vista que obtenga el valor más alto sea la que corresponda a la pose desde donde se tomó la imagen. Sin embargo, el primer filtro permite eliminar del conjunto de búsqueda aquellas imágenes cuya semejanza sea muy baja. Así, tomamos sólo aquellos candidatos que tengan una buena semejanza con la imagen, que son aquellos que pasen un primer umbral y máximo se escoge al 10% de la población, y es sobre ellos que se realiza el segundo filtro.

El segundo filtro, o *Filtro Lateral*, utiliza imágenes tomadas desde la misma posición que la imagen frontal, pero toma una imagen volteando 30° a la izquierda y otra imagen volteando 30° a la derecha con respecto a la imagen frontal. Aquí es donde vamos a hacer uso de los apuntadores que tienen cada una de las vistas sintéticas hacia sus vistas laterales, tal y como se explicó en la sección 3.2. En esta fase, tenemos un subconjunto de la base de datos a las que llamamos *poses candidatas*, producto del primer filtrado. Se toma la imagen lateral izquierda y se utiliza la métrica para asignar un valor de semejanza entre ésta y las vistas laterales izquierdas de las poses candidatas. De la misma forma, se toma la imagen derecha y se compara con las vistas laterales izquierdas del conjunto de candidatas. Así, cada vista candidata tendrá asignado un valor de semejanza para su vis-

```

% *****
% ALGORITMO DEL FILTRO LATERAL
% *****

1. Revisar la imagen de entrada, tanto la izquierda como la
   derecha; si alguna está vacía, termina.
2. Extraer los bordes de la imagen izquierda.
3. Extraer los bordes de la imagen derecha.
4. Para cada vista de la lista de candidatas:
5.     Extraer lo puntos de control de la vista izquierda de la
       vista candidata.
6.     Calcular y almacenar la semejanza entre los puntos de
       control de la vista izquierda y la imagen de bordes izquierda
       , usando la métrica Phi.
7.     Extraer lo puntos de control de la vista derecha de la
       vista candidata.
8.     Calcular y almacenar la semejanza entre los puntos de
       control de la vista derecha y la imagen de bordes derecha,
       usando la métrica Phi.
9.     Multiplicar ambos valores de semejanza, y luego
       multiplicarlos por la semejanza original de la vista
       candidata, asignandole el resultado como el nuevo valor de
       semejanza.
10. Ordenar las vistas de acuerdo a los nuevos valores de
     semejanza.
11. Regresar la lista de vistas candidatas y terminar.

```

ta izquierda y otro para su vista derecha. Dado que nuestra métrica regresa valores entre 0 y 1, es posible multiplicar ambos valores y seguir teniendo un resultado dentro del mismo rango. Éste resultado se multiplica por el resultado de la primera fase y los valores obtenidos se vuelven a ordenar, de tal forma que se pueda volver a extraer un nuevo conjunto de poses candidatas para la tercera fase.

Hasta este punto el método obtiene buenos resultados, pero sigue teniendo dificultades al enfrentarse a los problemas de Φ descritos anteriormente, puesto que sólo se esta considerando la relación que va de los puntos de control de una vista a los bordes de una imagen, pero no hemos considerado la relación inversa, pues recorrer cada uno de los pixeles de una imagen de bordes, y compararlos con los puntos de control de una vista, puede ser una operación bastante costosa computacionalmente hablando. Sin embargo decidimos implementar un tercer filtro, al que llamamos *Filtro Inverso*, en el cual se analizara esta relación mediante el uso de una segunda métrica. En esta fase se recorre la imagen de bordes (de la imagen frontal), y se toma uno de cada m pixeles, que es el valor que define la densidad de los pixeles seleccionados, lo que nos deja un conjunto de *puntos de referencia* suficientemente representativo de la imagen de bordes, pero nos permite realizar una comparación con pocos puntos entre una imagen y los

```

% *****
% ALGORITMO DEL FILTRO INVERSO
% *****

1. Utilizar la imagen de bordes del filtro frontal.
3. Para cada vista de la lista de candidatas:
4.     Extraer lo puntos de control de la vista.
5.     Extraer lo puntos de referencia de la imagen de bordes.
6.     Calcular y almacenar la semejanza entre la vista y la
        imagen, usando la métrica Psi.
7. Ordenar las vistas de acuerdo a los valores de semejanza
        recién calculados.
8. Regresa la lista de vistas candidatas y termina.

```

puntos de control. Cabe mencionar que se observó que si se toma m como un número primo (mayor que 2), los puntos de referencia aparecen en posiciones uniformemente distanciadas, mientras que en caso contrario, se producen aglomeraciones cuando la imagen de bordes contiene líneas verticales u horizontales o bien, líneas paralelas.

Para calcular la segunda métrica se toma el conjunto de puntos de referencia y por cada uno de ellos se calcula la distancia que hay entre éste y cada uno de los puntos de control, y se toma la distancia entre el punto de referencia y el punto de control más cercano. Si la distancia es menor a un umbral preestablecido, se toma como que existe una coincidencia entre los dos, y se registra en un acumulador de coincidencias, aunque dado que un punto de control puede corresponder al punto más cercano de varios puntos de referencia, se maneja un acumulador para los puntos de referencia coincidentes, y otro para los puntos de control. Finalmente, la segunda métrica se define como

$$\Psi = \left(\frac{k_r}{N_r} + \frac{k_c}{N_c} \right) \quad (17)$$

donde N_c es el número de puntos de control de la vista, N_r es el número de puntos de referencia de la imagen de bordes, y k_r y k_c son los acumuladores de coincidencias respectivos.

Esta segunda métrica se utiliza en filtrado inverso, gracias al cual se mejoran los resultados obtenidos en situaciones donde se presentan los problemas de la primera métrica. Sin embargo, este filtro consume muchos recursos y tiempo de procesador, y esta es la razón del por qué sólo se aplica a un conjunto de poses candidatas producto de los filtrados anteriores y por tanto muy reducido en tamaño.

Finalmente presentamos el algoritmo del método de *Localización mediante Regresión sobre Vistas Sintéticas*. Es necesario contar ya con la base de datos de las vistas sintéticas preprocesadas, y también con las imágenes de la cámara, de las cuales la imagen frontal se toma poniendo al robot en la misma posición con la que se generaron las vistas sintéticas, que en nuestro caso es estando en modo juego, con

```

% *****
% MÉTODO DE LOCALIZACIÓN MEDIANTE REGRESIÓN SOBRE VISTAS
% SINTÉTICAS
% *****

1. Tomar con la cámara del robot la imagen frontal.
2. Girar la cabeza 30\degree a la izquierda y tomar con la
   cámara del robot la imagen izquierda.
3. Girar la cabeza 60\degree a la derecha y tomar con la cámara
   del robot la imagen derecha.
4. Usar la imagen frontal para aplicar el filtro frontal y
   almacenar las vistas candidatas.
5. Seleccionar de las vistas candidatas al 10 % con mayor valor
   de semejanza y desechar las otras.
6. Usar la imagen izquierda y la imagen derecha junto con la
   lista de candidatas, y aplicar el filtro lateral,
   almacenando las vistas candidatas que regrese.
7. Seleccionar de las vistas candidatas al 10 % con mayor valor
   de semejanza y desechar las otras.
8. Usar la imagen frontal para aplicar el filtro inverso y
   almacenar las vistas candidatas.
9. De la lista final, tomar la primera vista y extraer la pose
   correspondiente a ella.
10. Regresar la pose como aquella desde la cual se tomaron las
    imágenes de la cámara.

```

las piernas y el tronco alineados, y con los motores de *roll* y *yaw* de la cabeza ajustados en su origen, alineados también con el tronco, y con el motor de *pitch* ajustado a -10° con respecto a la horizontal. Las imágenes laterales se obtienen girando 30° el motor de guiñada tanto a la izquierda como a la derecha de la posición frontal. Ya con estos elementos se procede de la siguiente manera:

3.3.1 Consideraciones Extras

3.3.1.1 Búsquedas heurísticas

Aún cuando el método en esta fase no comprende la inclusión de un filtro de partículas o algún otro tipo de localización probabilística, se probó el método frente a dos situaciones observadas.

La primera situación es que, según el reglamento, los robots inician el juego desde una de 4 zonas:

1. Portero: Debe estar dentro de su área al iniciar un juego o durante un *kick-off*³.

³ Un *Kick-off* es una manera de empezar o reiniciar el partido, y se da al inicio del juego, después de que se anota un gol, al inicio de la segunda mitad del partido, o al inicio de un periodo extra.

2. Defensa o Ataque: Si el robot utiliza el RefereeBox, puede situarse detrás de su media cancha en cualquier punto fuera del círculo central. Si no lo utiliza, lo puede posicionar un humano, pero debe iniciar no más lejos que la línea del área de su portería.
3. Centro: Un robot del equipo atacante debe situarse dentro del círculo central, ya sea autónomamente o ayudado por un humano.
4. Fuera: Cuando un robot quiere ingresar mientras el juego está transcurriendo, puede hacerlo desde la línea de fuera a la altura de la mitad de la cancha.

Con estas indicaciones, es posible hacer que el método de localización realice una búsqueda local en lugar de revisar todas las poses de la base de datos, lo que permite obtener resultados más precisos y reduce en gran medida el problema de la apertura. Para probar esta hipótesis se realizaron pruebas utilizando imágenes tomadas y categorizadas dentro de estas zonas, tal y como se explica en la sección 3.5.

La segunda situación es que, en una segunda fase del presente trabajo, será necesario adaptar el método a los robots del equipo. Pero más allá de eso, es posible mejorar los resultados obtenidos si se aplica un filtro de partículas o algún otro tipo de localización probabilística que utilice la información adquirida por este método, y pueda reducir el rango de búsqueda en localizaciones posteriores a la primera, realizando una búsqueda informada, que tome en cuenta la posición anterior y el tiempo que pase desde el momento en tomar la medición anterior. También se realizaron pruebas para ver como se desempeñaría una búsqueda informada, tomando una pose probable y asignando un radio de búsqueda.

3.4 SISTEMA

El sistema que se implementó para probar y desarrollar el método, será utilizado por los robots del equipo DotMX para localizarse. Éste se implementó como un módulo similar a los que utiliza la plataforma del DarwinOP para poder integrarlo fácilmente. El código se programó en C++, utilizando la librería de OpenCV, por lo que basta con incluir ambos en el sistema del robot para que se puedan utilizar las funciones de localización. En la figura 12 se observa cómo se ve el sistema en funcionamiento, y en la figura se ve el diagrama de bloques del sistema.

3.5 EXPERIMENTOS Y VALIDACIÓN

En esta sección se explica cuáles fueron los experimentos realizados para validar el funcionamiento del método, así como los experimentos que sirvieron para definir y justificar tanto la métrica Φ como el orden en que se ejecutan los filtros de las vistas.

3.5.1 Validación de la métrica Φ

Las pruebas que se realizaron para validar nuestra métrica y comparar las diferentes combinaciones de los valores α y β con la propuesta original, consistieron en lo siguiente.

- Se tomaron 460 imágenes de cámara desde poses aleatorias pero correspondientes a poses de la base de datos, y cada una de estas imágenes se comparó contra las vistas sintéticas de la base de datos, por lo que para cada imagen estaba presente la vista sintética tomada desde la misma pose.
- Utilizando cada una de las métricas candidatas se asignó un valor de semejanza entre cada una de las 460 imágenes y cada una de las 8064 vistas.
- Dado que para cada una de las imágenes de prueba se conoce la vista correspondiente, se tomaron todos los valores de semejanza de las 460 imágenes, se calculó el promedio y se estableció como el valor de los *Valores Correspondientes*.
- De manera similar, para todas las comparaciones entre imágenes y vistas no correspondientes, se tomó el promedio total de los valores de semejanza y se estableció como el valor de los *Falsos Positivos*.

3.5.2 Validación de los filtros

Dado que se contaba con 3 filtros diferentes (frontal, lateral y inverso), existen 15 combinaciones posibles para realizar la localización. Para probar cuál era la mejor combinación, se realizaron pruebas con 600 conjuntos de imágenes (frontal, izquierda y derecha), generadas aleatoriamente sin limitantes en la pose, por lo que no corresponden a poses exactas en la base de datos. Sobre éstas se corrió el algoritmo de localización. En cada prueba se sabe la pose real de la imagen, por lo que es posible registrar en qué lugar quedó de la lista final ordenada de poses candidatas, donde la pose en el primer lugar es la respuesta del algoritmo. También se registra la distancia y la diferencia de orientación entre la pose real y la pose correspondiente a la vista sintética que da como respuesta final el método. Con estos datos podemos ver

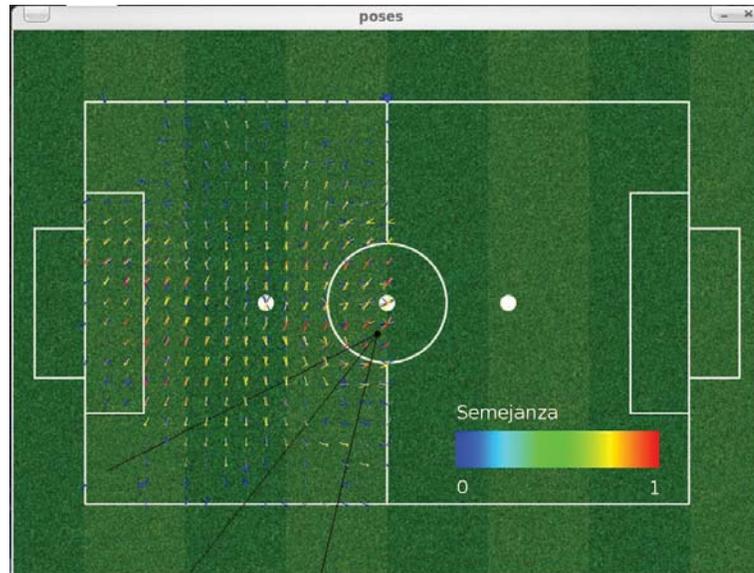
para cada combinación de filtros cuál es el promedio de la mejor posición alcanzada por las vistas correspondientes a las poses reales, así como el promedio de las distancias y diferencias de orientación entre las soluciones y las poses reales. Para poder registrar el número de respuestas positivas para cada combinación se asignaron dos valores de aceptación relativos, que marcan subjetivamente que tan correcta es una localización cuando la pose real no se ajusta a una de las poses de la malla. Se consideró pues, que si la distancia entre la pose real y el resultado final era menor a 30 cm y la diferencia de orientación era menor a 30° , la localización es correcta.

3.5.3 *Experimentos con localización por zonas*

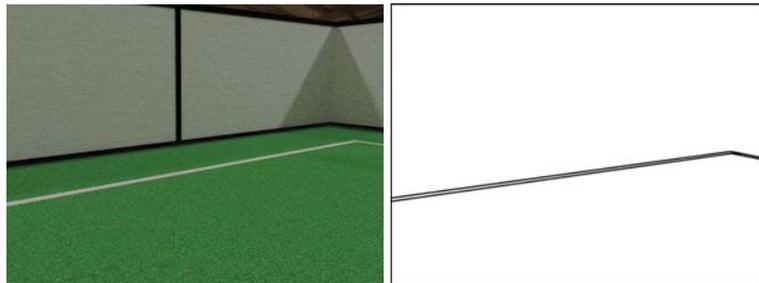
Para las pruebas de localización por zonas, se generaron 250 imágenes con sus respectivas imágenes laterales, pero cuyas poses correspondían aleatoriamente a las zonas desde dónde se inicia el juego. Éstas son zonas ajustadas heurísticamente, de tal forma que el robot tenga pistas más firmes de dónde podría encontrarse. Las pruebas se realizaron bajo el mismo procedimiento que se utilizó para validar los filtros, sólo con un conjunto más reducido, pues el número de poses desde donde se puede encontrar el robot al realizar este tipo de búsqueda es mucho menor.

3.5.4 *Experimentos con localización informada*

Durante las pruebas realizadas para ver como se desempeñaba la localización informada, se maneja una pose en el campo correspondiente a la posición donde se espera que se encuentre el robot, que puede ser calculada de varias formas, entre ellas utilizando rastreo por odometría. Además se utilizó una variable a la que se le asignaba el radio de búsqueda alrededor de la pose esperada, lo cual sirve para poder reducir el área de búsqueda, dependiendo de la probabilidad que se tenga de que el robot se encuentre en una pose dada. Las pruebas se realizaron tomando variaciones del radio de búsqueda, con radios de 120, 80 y 40 centímetros alrededor de la pose, y así, las únicas poses consideradas eran aquellas que estuvieran dentro del radio de búsqueda.



(a) Imagen que muestra las poses candidatas después de realizar un filtrado frontal. Los colores de los vectores correspondientes a cada pose reflejan el valor de semejanza obtenido, que corresponde a la probabilidad de que el robot se encuentre en esa pose. El círculo negro marca la pose real.



(b) A la izquierda se muestra la imagen de cámara del robot, y en el lado derecho se muestra la vista sintética encontrada por el método, correspondiente a la pose candidata con mayor valor de semejanza.

Figura 12

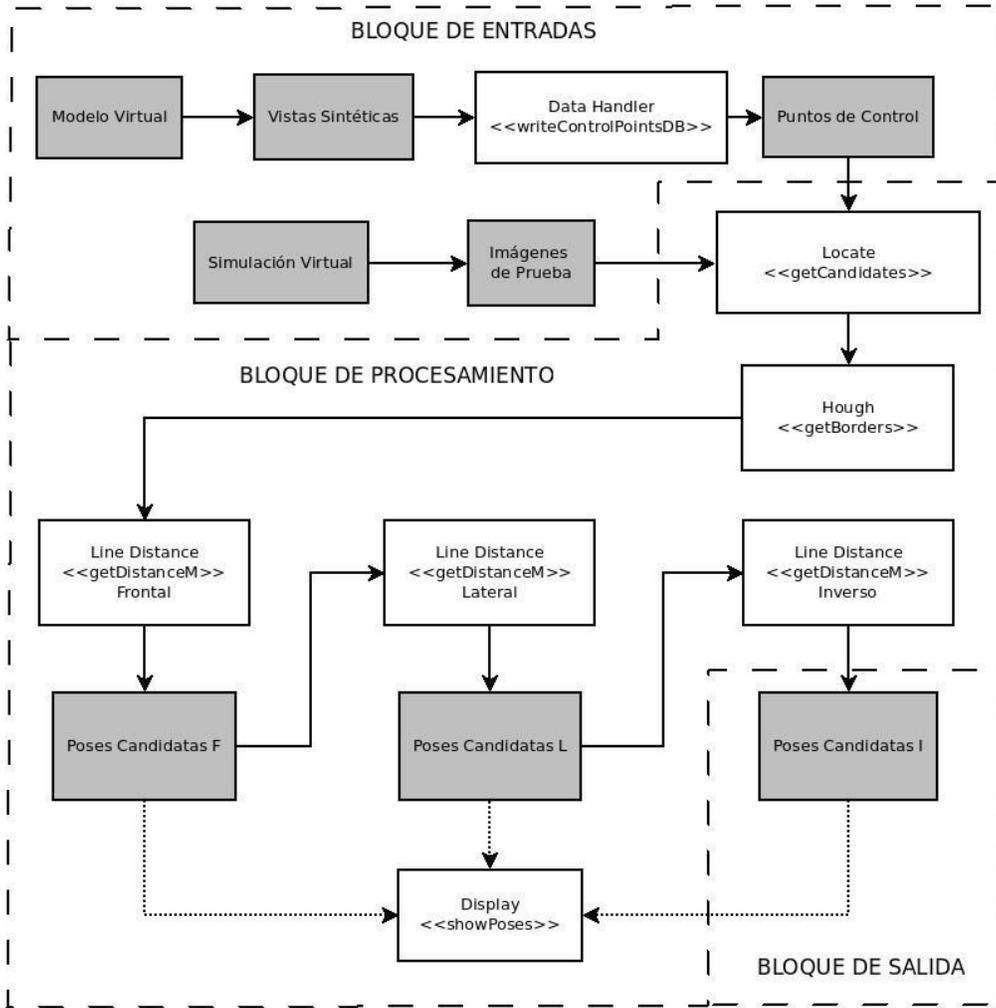


Figura 13: Diagrama de Bloques

RESULTADOS Y DISCUSIÓN

En la tabla 1 se ven los resultados de las pruebas realizadas con la métrica Φ (eq. (14)). Podemos destacar dos situaciones que se observan en la tabla. La primera es que al utilizar la propuesta original, que sólo toma en cuenta las distancias y el número de Puntos de Control, se obtiene el valor de semejanza más grande cuando se trata de una correspondencia entre una imagen y una vista sintética. Sin embargo también es cuando se presentan los valores de semejanza más grandes entre imágenes y vistas que no son correspondientes. Puesto que tal como se estipuló anteriormente, es imposible conseguir correspondencias del 100 %, nos interesa más que nuestra métrica presente valores pequeños en los falsos positivos, y aún más, que la diferencia entre los positivos falsos y los positivos verdaderos se maximice. La segunda situación es que, aún cuando la combinación de la métrica original y el factor α regresaron el valor de semejanza más pequeño en cuanto a positivos falsos, es hasta que se combinan con el factor β que se maximiza la diferencia entre verdaderos y falsos, lo que nos permite discernir de manera más acertada la correspondencia entre una imagen y una vista. En la figura 14 se muestran las curvas ROC¹ de las combinaciones usadas, y sus valores de *Área Bajo la Curva* muestran que cualquiera de las 4 son buenos clasificadores, por tener valores entre 0.8 y 0.9, pero al mismo tiempo, se comprueba que la combinación de los 3 factores es la que presenta un mejor desempeño como clasificador.

Los resultados de la validación de los filtros se enlistan en la tabla 2. Lo primero que podemos observar es que aquellas combinaciones

¹ Receiver Operating Characteristic

COMBINACIÓN	FP	VP	DIFERENCIA
Φ	0.2606	0.7745	0.5143
$\Phi \cdot \alpha$	0.1514	0.7142	0.5628
$\Phi \cdot \beta$	0.22	0.7411	0.5211
$\Phi \cdot \alpha \cdot \beta$	0.1733	0.7671	0.5938

Tabla 1: Resultados de las pruebas de la métrica Φ . FP son los *Falsos Positivos*, VP son los *Verdaderos Positivos*.

COMBINACIÓN	LUGAR	Δd	$\Delta\theta$	POSITIVOS	TIEMPO
[F, -, -]	111	141.367	72.625	0.124792	00:04:56
[F, L, -]	37	63.797	28.282	0.650082	00:05:52
[F, I, -]	95	112.435	57.941	0.287853	00:12:20
[F, L, I]	33	60.737	27.025	0.671641	00:06:58
[F, I, L]	75	65.202	29.033	0.643449	00:16:32
[L, -, -]	97	129.321	66.359	0.079866	00:11:54
[L, F, -]	91	153.1	71.983	0.049916	00:10:55
[L, I, -]	64	118.115	54.87	0.211314	00:21:20
[L, F, I]	89	142.102	66.805	0.128119	00:13:26
[L, I, F]	64	123.396	55.82	0.179700	00:22:13
[I, -, -]	65	115.85	54.113	0.282861	02:05:14
[I, F, -]	61	178.93	75.653	0.043261	02:03:15
[I, L, -]	26	78.258	33.384	0.657237	02:02:00
[I, F, L]	25	80.08	33.645	0.637271	02:05:13
[I, L, F]	57	165.654	71.986	0.148086	02:34:45

Tabla 2: Resultados de las combinaciones de filtros. El lugar indica en que lugar de la lista ordenada quedaron en promedio las poses correspondientes; Δd es la distancia promedio entre la posición real y la encontrada; $\Delta\theta$ es la diferencia entre la orientación real y la orientación de la pose encontrada; los positivos indican la tasa de aciertos que tuvo el sistema. F: Filtro Frontal; L: Filtro Lateral; I: Filtro Inverso.

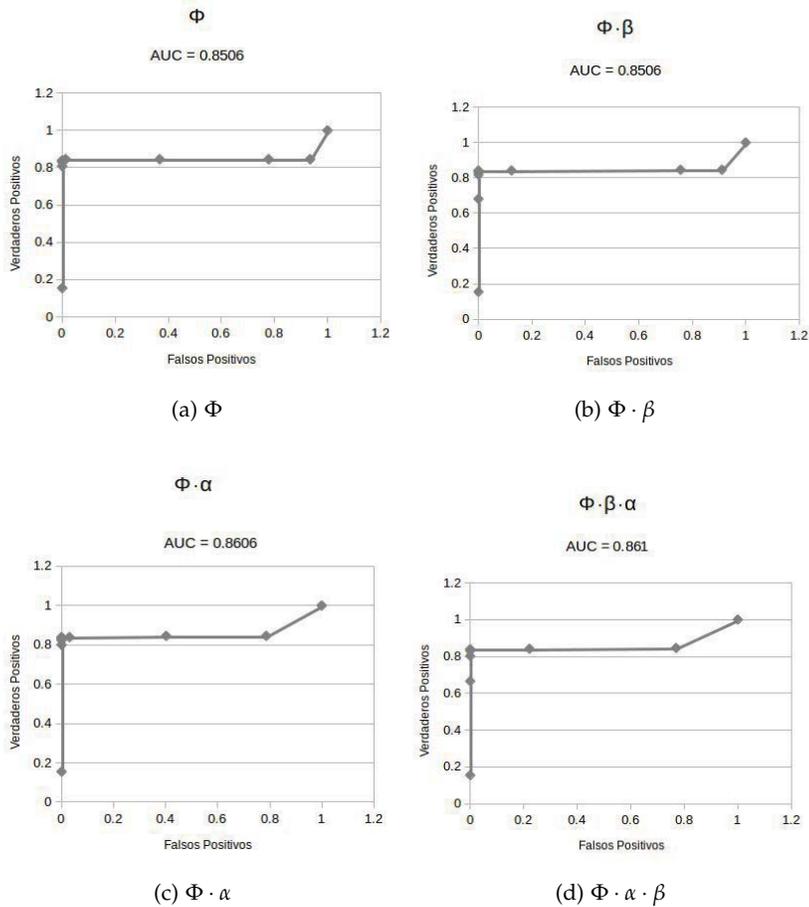


Figura 14: Las gráficas de ROC muestran la eficacia de un clasificador, al relacionar la razón que hay entré los *Falsos Positivos* ($1 - \text{Especificidad}$) y los *Verdaderos Positivos* (*Sensibilidad*), al ir variando los valores del umbral de aceptación. El *área bajo la curva* es una medida de precisión para el clasificador, mientras más cercano a 1, mejor su desempeño.

que primero realizan el filtro inverso, aunque cuando hay combinaciones que dan muy buenos resultados, tardan significativamente más tiempo en obtenerlos, debido a que la métrica Ψ (eq. 17) tiene que realizar muchas más operaciones para llegar a una conclusión, y cuando el conjunto de búsqueda son todas las poses de la base de datos, el requerimiento de recursos es inmenso.

Por otro lado, se observa que aquellas combinaciones que realizan primero el filtrado lateral presentan resultados relativamente bajos, y por el contrario, cuando se empieza con el filtro frontal y se combina de alguna forma con el filtro lateral, se obtienen los mejores resultados, aún más, la mejor combinación es la que empieza con el filtro frontal, luego hace el lateral y termina con el inverso. Por tal motivo es que se tomó la decisión de realizar los filtrados en dicho orden,

COMBINACIÓN	LUGAR	Δd	$\Delta\theta$	POSITIVOS
[F, L, -]	11	20.1434	4.06349	0.896
[F, L, I]	10	18.67	3.81	0.92
[F, I, L]	19	19.1731	4.30952	0.916

Tabla 3: Resultados de la búsqueda por zonas. El lugar indica en que lugar de la lista ordenada quedaron en promedio las poses correspondientes; Δd es la distancia promedio entre la posición real y la encontrada; $\Delta\theta$ es la diferencia entre la orientación real y la orientación de la pose encontrada; los positivos indican la tasa de aciertos que tuvo el sistema

pues además de los buenos resultados, estos se consiguen en un tiempo muy corto de procesamiento.

Tomando las tres mejores combinaciones de filtros, se realizaron las pruebas de la localización por zonas, cuyos resultados se muestran en la tabla 3. Los resultados obtenidos muestran que teniendo idea de las zonas donde puede encontrarse el robot y una idea aproximada de la orientación –en este caso, la portería contraria, o el centro de la cancha cuando el robot ingresa o antes de un *kick off* – la precisión del método es muy grande, lo que le permite al robot tener una muy buena aproximación de su posición inicial, y mejorar sus posibilidades de realizar un rastreo de su posición.

En cuanto a la búsqueda informada, además de las combinaciones que obtuvieron los mejores valores, también se probó un filtrado en que sólo se utiliza la imagen frontal por cada intento de localización, de tal forma que sólo el filtro frontal y el filtro inverso se aplican (FI). La tabla 4 muestra que conforme el radio de búsqueda disminuye, la precisión del método aumenta significativamente, pero además, los resultados de la combinación FI son lo suficientemente buenos como para poder considerar la utilización del método en tiempo real sin necesidad de adquirir las imágenes laterales cada vez que se necesite localizar el robot.

Para mostrar la robustez del método con respecto a las oclusiones, se tomaron algunas de las combinaciones con mejores resultados y se realizaron pruebas similares, pero con una muestra de imágenes generadas sin la figura de Suzanne, por lo que ninguna imagen presentaba oclusiones. Se puede observar en la tabla 5 que los resultados obtenidos en esta prueba son muy similares a los obtenidos al introducir oclusiones en las imágenes de la muestra, lo cual pone en evidencia que el método no se ve afectado en gran medida por la presencia de elementos oclusivos, siempre y cuando se logren percibir con cierta claridad suficientes elementos invariantes de la cancha.

De acuerdo con los resultados de las pruebas realizadas, podemos hacer algunas observaciones, empezando porque la proporción de

COMBINACIÓN	RADIO	LUGAR	Δd	$\Delta\theta$	POSITIVOS
[F, L, -]	120	10	30.164	5.96	0.775374376
[F, L, I]	120	7	28.106	5.78	0.7903494176
[F, I, L]	120	18	30.11	6.241	0.772046589
[F, I, -]	120	20	48.095	8.589	0.5224625624
[F, L, -]	80	6	24.366	5.65	0.8136439268
[F, L, I]	80	6	23.765	5.123	0.8252911814
[F, I, L]	80	9	24.732	6.442	0.7986688852
[F, I, -]	80	10	36.686	8.442	0.6139767055
[F, L, -]	40	5	16.48	5.46	0.9816971714
[F, L, I]	40	2	16.382	5.119	0.9850249584
[F, I, L]	40	6	16.712	5.683	0.9750415973
[F, I, -]	40	3	19.568	8.607	0.9384359401

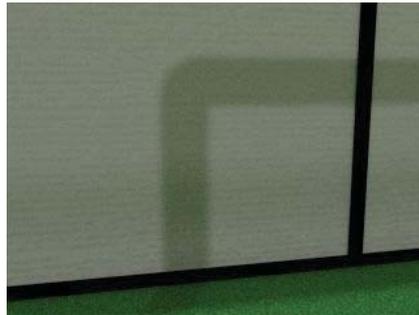
Tabla 4: Resultados de la búsqueda informada. El radio corresponde al área al rededor de la pose candidata, de donde se toman las vistas a comparar; el lugar indica en que lugar de la lista ordenada quedaron en promedio las poses correspondientes; Δd es la distancia promedio entre la posición real y la encontrada; $\Delta\theta$ es la diferencia entre la orientación real y la orientación de la pose encontrada; los positivos indican la tasa de aciertos que tuvo el sistema.

COMBINACIÓN	LUGAR	Δd	$\Delta\theta$	POSITIVOS	TIEMPO
[F, L, -]	36	73.775	27.778	0.669788	00:05:49
[F, L, I]	32	59.167	32.085	0.697662	00:07:05
[F, I, L]	79	71.612	26.462	0.653991	00:14:08
[I, L, -]	24	73.217	30.120	0.671438	01:57:36
[I, F, L]	24	63.881	33.469	0.649816	02:14:37

Tabla 5: Resultados de las pruebas sin oclusiones en las imágenes de muestra. Los valores son muy similares a los obtenidos con la muestra original. El lugar indica en que lugar de la lista ordenada quedaron en promedio las poses correspondientes; Δd es la distancia promedio entre la posición real y la encontrada; $\Delta\theta$ es la diferencia entre la orientación real y la orientación de la pose encontrada; los positivos indican la tasa de aciertos que tuvo el sistema. F: Filtro Frontal; L: Filtro Lateral; I: Filtro Inverso.

localizaciones positivas cuando no se tiene ninguna información a priori podrá parecer baja, pero habría que tomar en cuenta ciertos factores antes de llegar a esa conclusión. De entrada, es un método que utiliza como único sensor la cámara del robot y depende enteramente de la apariencia, por lo que al comparar una imagen de la cámara con una vista sintética, aún cuando exista una semejanza casi perfecta entre la imagen y la vista, puede no existir correspondencia alguna entre las dos. Los principales problemas que encontramos fueron los siguientes:

- Cuando el robot se encuentra cerca de una de las líneas que delimitan el campo y está orientado hacia afuera del mismo, dentro de su campo de visión no se encuentran elementos invariantes, o al menos no con suficiente representatividad, aún cuando se realiza el filtrado lateral. Mientras más marcada sea esta situación, es más difícil definir en qué pose se encuentra. Sin embargo, hay que destacar que estas mismas posiciones, pero orientadas hacia el centro de la cancha o hacia alguna de las porterías, son aquellas que presentan el mayor índice de precisión para el método. Véase la figura 15a
- El robot siempre tiene dentro de su campo de visión una parte que no corresponde al campo de juego. Por lo regular corresponde a las vallas contenedoras y en una pequeña proporción, al público o lo que sea que haya en el fondo. Cuando las vallas son bajas, al ruido del fondo aumenta, lo cual afecta especialmente el desempeño del filtrado inverso. Véase la figura 15b



(a) El robot se encuentra muy cerca de la línea de fuera, por lo que no logra ver bien ninguno de los elementos invariantes de la cancha

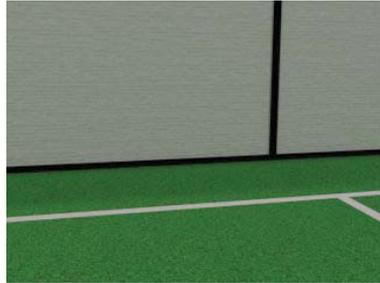


(b) Los elementos externos genera mucho ruido en el campo de visión del robot, lo que lleva a confusiones para el método

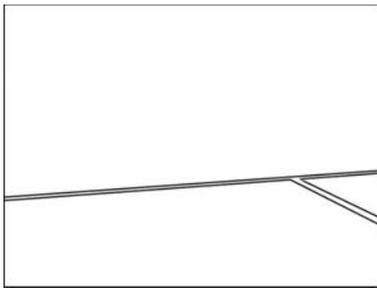
Figura 15

- Independientemente del tamaño, las vallas se encuentran por lo regular paralelas a las líneas que delimitan el campo, por lo cual es fácil confundirlas con éstas. Esto implica que en muchas ocasiones el robot encuentra bordes más significativos en las vallas

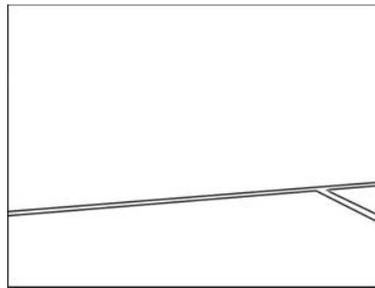
que en las mismas líneas del campo, pues conforme más se acerca a las vallas, su campo de visión abarcará menos las líneas del campo y más las líneas de las vallas. Esto genera una confusión para el método, que aún cuando regresa una pose muy cercana a la pose real, con orientaciones prácticamente iguales, la distancia entre ambas va a variar en función de la distancia que hay entre las líneas y las vallas. Véase la figura 15b



(a)



(b)



(c)

Figura 16: La figura (a) muestra la imagen de la cámara del robot. Al realizar la comparación con las vistas sintéticas, la métrica se confunde con las líneas de las vallas, asignando un valor mayor a la vista de la imagen (b) que a la de la imagen (c), que es la que corresponde realmente.

El último punto se puede generalizar, pues el método depende mucho de la perspectiva de la imagen, y dado que el entorno está conformado por líneas perpendiculares entre sí, la gran mayoría de las ocasiones en que el método llega a equivocarse, la orientación que obtiene varía con respecto a la orientación real en múltiplos de 90° .

Aún cuando estas situaciones son inevitables, podemos considerar que los resultados obtenidos mediante éste método son muy aceptables, especialmente si se ve como preámbulo para una estrategia de localización probabilística, como por ejemplo un filtro de partículas, que por lo regular hace uso de sensores que miden la distancia que va del robot hacia elementos que lo rodean. En éste caso, se puede utilizar nuestro método para proveer al filtro de partículas del conjunto de posiciones candidatas que toma en cuenta en iteraciones posteriores. Sin embargo, esto sale de los alcances de este trabajo.

Por otro lado, aún utilizando únicamente éste método como forma de localización al robot, la distancia promedio encontrada con respecto a la pose real, así como la diferencia entre las orientaciones, le brinda una idea suficientemente buena al robot de en qué posición se encuentra y permite generar estrategias de juego más avanzadas.

CONCLUSIONES

En este trabajo se desarrolló un método de localización para robots que juegan fútbol y que sólo cuentan con una cámara como sensor, el cual utiliza los principios de la visión computacional. Se aprovecharon los conocimientos que se tienen del entorno para crear un modelo tridimensional que permite generar vistas sintéticas de la cancha de fútbol, para utilizarlas como los elementos de una base de datos que contiene poses distribuidas por dentro de la cancha.

Las vistas sintéticas se parametrizaron utilizando puntos de control correspondientes a los bordes de los elementos invariantes de las canchas, que son las líneas y las porterías. De esta forma fue posible tener una base de datos con 8064 poses diferentes, pero utilizando un conjunto muy reducido de puntos para representar cada una de las vistas correspondientes.

La base de datos se utilizó para realizar regresiones de poses correspondientes a imágenes tomadas por la cámara del robot. Para esto se definieron dos métricas distintas que miden la semejanza entre una imagen de cámara y una vista sintética, y con éstas métricas se realizaron filtros de las poses candidatas para finalmente obtener un conjunto reducido de poses cuyas vistas sintéticas correspondientes se asemejen más a la imagen tomada por la cámara del robot, considerando como la respuesta aquella con el valor más alto.

Para el desarrollo de este método, se implementó una simulación del sistema de visión del robot, así como del entorno en que se desenvuelve. La simulación emplea elementos que le brindan a las imágenes apariencias foto-realistas, gracias a los algoritmos y métodos de la computación gráfica.

Se realizaron pruebas bajo condiciones realistas, con cambios aleatorios tanto de los elementos que rodean la cancha, oclusiones, y cambios en la intensidad de la luz, y se logró encontrar la pose correcta en el 65 % de los casos; simultáneamente se encontró que el error promedio fue de 60.73 cm y 27.02° con respecto a la pose real, la cual quedó situada en promedio dentro de las 33 poses con los mayores valores de semejanza. Por otro lado, también se realizaron pruebas para búsquedas informadas, donde el método recibía información extra acerca de la pose donde podría encontrarse el robot y su probabilidad, o bien donde sólo se tomaban zonas de la cancha donde el robot podría estar en una posición inicial relativa. En ambos casos los resultados obtenidos mejoraban significativamente con respecto a las búsquedas no informadas, obteniendo resultados positivos de hasta el 98 % de los casos.

Estos resultados muestran que el método se puede utilizar directamente para obtener la localización del robot en los momentos clave del partido, como cuando inicia, o cuando ubica la pelota, o bien cuando se posiciona directamente frente a ella, pero además, es posible usarlo como preámbulo para un método de localización probabilístico que se pueda emplear para el rastreo del robot en tiempo real.

Nuestro método es robusto ante oclusiones, pues fue probado en presencia de objetos que llegaban a estorbar su campo de visión. Además es robusto en cuanto a cambios de luz, pues fue probado en varios tipos de iluminación y entorno. Ambas situaciones están presentes y son muy comunes en el ámbito de la competencia de RoboCup y demás competencias de fútbol robótico, que es donde se desea poner a prueba el método. Véase Figura 17.

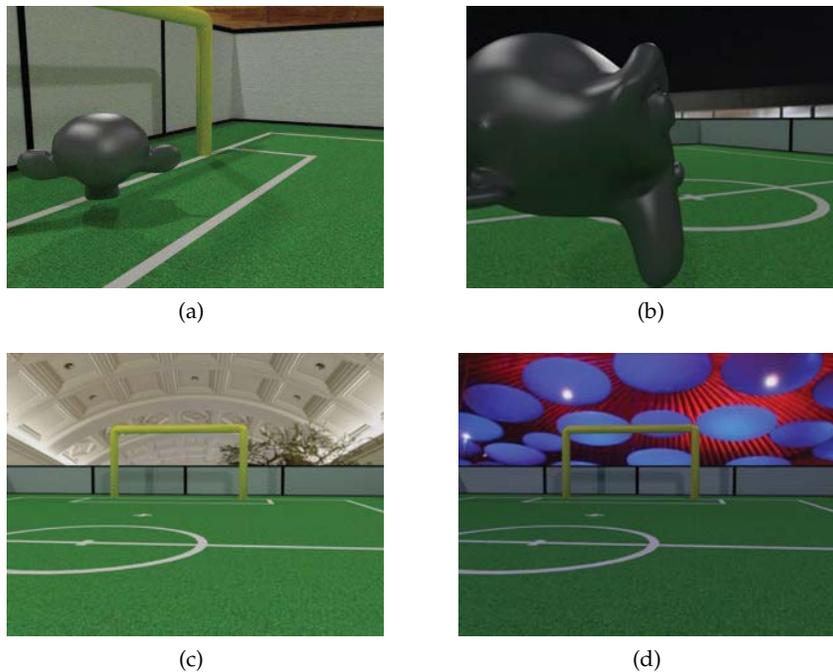


Figura 17: Las figuras (a) y (b) muestran el tipo de oclusiones contra el que se enfrentó el algoritmo, siendo que la imagen (b) muestra una oclusión muy prominente, pero son pocos casos en que pasa eso. Las figuras (c) y (d) muestran los cambios de intensidad que se pueden presentar en las imágenes generadas por la simulación, al cambiar el fondo y por tanto la iluminación ambiental.

Aún cuando fue desarrollado bajo las especificaciones de un robot Darwin-OP y de las dimensiones y características reglamentarias de las canchas de fútbol robótico, es un método que puede ser utilizado por cualquier robot móvil que cuente con una cámara, y se puede

emplear en cualquier entorno conocido de antemano, en el cual el problema del mapeo se encuentre resuelto.

REFERENCIAS

- [1] Agrawal, Motilal y Kurt Konolige: *Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS*. (Cita en página 37.)
- [2] Aider, Omar Ait, Philippe Hoppenot y Etienne Colle: *A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences*. *Rob. Auton. Syst.*, 52(2-3):229–246, Agosto 2005, ISSN 09218890. <http://linkinghub.elsevier.com/retrieve/pii/S0921889005000412>. (Cita en página 23.)
- [3] Armstrong, Martin y Andrew Zisserman: *Robust Object Tracking*. (Cita en página 36.)
- [4] Asada, Minoru, Hiroaki Kitano, Itsuki Noda y Manuela Veloso: *RoboCup: Today and Tomorrow What we have learned*. (Cita en página 13.)
- [5] Asif, U. y J. Iqbal: *Rapid Prototyping of a Gait Generation Method using Real Time Hardware in Loop Simulation*. *Int. J. Model. Simul. Sci. Comput.*, 2(4):393–411, 2011. (Cita en página 39.)
- [6] Atiya, Sami y GD Hager: *Real-time vision-based robot localization*. *Robot. Autom. IEEE ...*, 1993. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=265922 (Cita en página 32.)
- [7] Ballard, DH: *Generalizing the Hough transform to detect arbitrary shapes*. *Pattern Recognit.*, 1981. <http://www.sciencedirect.com/science/article/pii/0031320381900091> (Cita en página 26.)
- [8] Bandlow, Thorsten, Michael Klupsch, Robert Hanek y Thorsten Schmitt: *Fast image segmentation, object recognition and localization in a robocup scenario*. *Rob. Robot Soccer ...*, 2000. http://link.springer.com/content/pdf/10.1007/3-540-45327-X_13.pdf (Citas en páginas 4 y 33.)
- [9] Benda, M., V. Jagannathan y R. Dodhiawalla: *On Optimal Cooperation of Knowledge Sources*. Informe técnico, Boeing AI Center, 1985 (Cita en página 14.)
- [10] Betke, Margrit y Leonid Gurvits: *Mobile Robot Localization using Landmarks*. 1995 (Cita en página 4.)
- [11] Borges, Geovany Araujo y Marie José Aldon: *Line Extraction in 2D Range Images for Mobile Robotics*. páginas 267–297, 2004 . (Cita en página 36.)

- [12] Borisov, Alexander, Alireza Ferdowsizadeh, Christian Mohr, Heinrich Mellmann, Martin Martius, Thomas Krause, Tobias Hermann, Oliver Welter y Yuan Xu: *The RoboCup NAO Team of Humboldt Universitat zu Berlin*. 2009. (Cita en página 33.)
- [13] Burchardt, Armin, Tim Laue y R Thomas: *Optimizing Particle Filter Parameters for Self-localization*. páginas 145–156, 2011 . (Cita en página 37.)
- [14] Canny, John F: *A Computational Approach to Edge Detection*. IEEE Trans. Pattern Anal. Mach. Intell., PAMI-8(6):679–698, 1986. (Cita en página 23.)
- [15] Chekhlov, D., W.W. Mayol-Cuevas y A.D. Calway: *Appearance Based Indexing for Relocalisation in Real-Time Visual SLAM*. Proceedings Br. Mach. Vis. Conf. 2008, páginas 37.1–37.10, 2008. <http://www.bmva.org/bmvc/2008/papers/183.html> . (Cita en página 36 .)
- [16] Chekhlov, D., M. Pupilli, W. Mayol-Cuevas y A. Calway: *Real-time and robust monocular slam using predictive multi-resolution descriptors*. Proc Int. Symp. on Visual Computing, 2006. (Cita en página 50)
- [17] Comport, A I, E Malis y P Rives: *Real-time Quadrifocal Visual Odometry*. 1, 2008. (Cita en página 36.)
- [18] Cuevas, Erik J., Daniel Zaldivar y Raul Rojas: *PARTICLE FILTER IN VISION TRACKING*. e-Gnosis, páginas 1–11, 2007. (Cita en página 37 .)
- [19] Damen, Dima, Andrew Gee, Walterio Mayol-cuevas y Andrew Calway: *Egocentric Real-time Workspace Monitoring using an RGB-D Camera* (Citas en páginas 37 y 50.)
- [20] Davison, Andrew J, Ian D Reid, Nicholas D Molton y Olivier Stasse: *MonoSLAM: real-time single camera SLAM*. IEEE Trans. Pattern Anal. Mach. Intell., 29(6):1052–67, Junio 2007, ISSN 0162-8828. <http://www.ncbi.nlm.nih.gov/pubmed/17431302>. (Cita en página 37.)
- [21] Durrant-Whyte, Hugh y Tim Bailey: *Simultaneous Localization and Mapping : Part I*. 2006. (Cita en página 34.)
- [22] Eade, Ethan y Tom Drummond: *Edge landmarks in monocular SLAM*. Image Vis. Comput., 27(5):588–596, 2009, ISSN 0262-8856. <http://dx.doi.org/10.1016/j.imavis.2008.04.012>. (Cita en página 37 .)
- [23] Fu, K. S., R. C. Gonzalez y C. S. G. Lee: *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill. (Cita en páginas 11, 19 y 32.)

- [24] Gee, AP y W Mayol-Cuevas: *6D Relocalisation for RGBD Cameras Using Synthetic View Regression*. [bmva.org](http://www.bmva.org/bmvc/2012/BMVC/paper113/paper113.pdf), páginas 1–11, 2012. <http://www.bmva.org/bmvc/2012/BMVC/paper113/paper113.pdf>. (Cita en páginas 37 y 51.)
- [25] Horaud, Radu y Humberto Sossa: *Polyhedral Object Recognition by Indexing*. 28(12):1855–1870, 1995. (Cita en página 32.)
- [26] Hough, P. V.: *Method and means for recognizing complex patterns*. 1962. (Cita en página 26.)
- [27] Inam, W.: *Particle filter based self-localization using visual landmarks and image database*. (Cita en página 37.)
- [28] Jordan, I. M.: *Forward models: Supervised learning with a distal teacher*. 16(Cognitive Science):307–354, 1992. (Cita en página 22.)
- [29] Kalman, RE: *A new approach to linear filtering and prediction problems*. J. basic Eng., 82(Series D):35–45, 1960. <http://160.78.24.2/Public/Kalman/Kalman1960.pdf>. (Cita en página 34.)
- [30] Kitano, H., WV Hahn, L. Hunter, R. Oka, B. Wah y T. Yokoi: *Grand challenge AI applications*. IJCAI, 1993. <http://wah.cse.cuhk.edu.hk/wah/Wah/papers/C093/C93.pdf>. (Cita en página 13.)
- [31] Kitano, Hiroaki, Minoru Asada y Yasuo Kuniyoshi: *RoboCup A Challenge Problem for AI*. 18(1):73–85, 1997. (Cita en páginas 13 y 14.)
- [32] Kurashige, Kentarou, Yukiko Onoue y Toshio Fukuda: *From Automation To Autonomy*. En *Mach. Learn.*, páginas 39–52. (Cita en página 21.)
- [33] Leonard, John J. y Hugh Durrant-whyte: *Mobile Robot Localization by Tracking Geometric Beacons*. (Cita en página 36.)
- [34] Levy, Richard M.: *The Virtual Reality Revolution: The Vision and the Reality*. En *Virtual Real. Hum. Comput. Interact.*, páginas 21–40. (Cita en página 39.)
- [35] Lowe, D.G.: *Object recognition from local scale-invariant features*. páginas 1150–1157 vol.2, 1999. (Cita en página 31.)
- [36] Maybeck, Peter S.: *Stochastic models, estimation, an control.*, volumen 1. ACADEMIC PRESS, 1979, ISBN 0124807011. (Cita en página 35.)
- [37] Mayol-Cuevas, Walterio y Andrew P. Gee: *Real-Time Model-Based SLAM Using Line Segments*. (Cita en página 23.)

- [38] Mellouk, Abdelhamid y Abdennecer Chebira: *Machine Learning*. 2011, ISBN 9783902613561. (Cita en página 21.)
- [39] Nocks, Lisa: *The robot: the life story of a technology*. Greenwood Publishing Group, 2007. (Cita en página 9.)
- [40] Pollack, ME y M Ringuette: *Introducing the Tileworld: Experimentally evaluating agent architectures*. 1990. <http://www.aaai.org/Papers/AAAI/1990/AAAI90-028.pdf>. (Cita en página 14.)
- [41] Princen, John, John Illingworth y Josef Kittler: *A hierarchical approach to line extraction based on the Hough transform*. *Comput. Vision, Graph. Image Process.*, 52(1):57–77, Octubre 1990, ISSN 0734189X. <http://linkinghub.elsevier.com/retrieve/pii/0734189X9090123D>. (Cita en página 23.)
- [42] RoboCup-Team: *RoboCup Soccer Humanoid League Rules and Setup*. 2013. (Cita en página 44.)
- [43] Röfer, Thomas y Matthias Jünger: *Fast and Robust Edge-Based Localization in the Sony Four-Legged Robot League*. 2003, 2004. (Cita en página 33.)
- [44] Russell, Stuart J, Peter Norvig, John F Canny, Jitendra M Malik, Douglas D Edwards y Stuart J Stuart Jonathan: *Artificial Intelligence: A Modern Approach*. 1995, ISBN 0131038052. (Citas en páginas 17, 19, 20 y 35.)
- [45] Schulz, Hannes, Weichao Liu, Jörg Stückler y Sven Behnke: *Utilizing the Structure of Field Lines for Efficient Soccer Robot Localization*. *Proc. Rob. International Symp.*, (June):1–12, 2010. (Citas en páginas 33 y 37.)
- [46] Se, S., D. Lowe y J. Little: *Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks*. *International Journal of Robotic Research*, (21):735–758, 2002. (Cita en página 50.)
- [47] Smagt, MJ Van der y GR Stoner: *Occlusion and the solution to visual motion ambiguity: Looking beyond the aperture problem*. *J. Vis.*, 8:1–12, 2008. <http://www.journalofvision.org/content/8/2/4.short>. (Cita en página 56.)
- [48] Smith, R., M. Self y P. Cheeseman: *Estimating uncertain spatial relationships in robotics*. En Cox, I.J. y Wilfon G.T. (editores): *Auton. Robot Veh.*, páginas pp. 167–193. Springer-Verlag, New York, NY, USA, 1990. (Cita en página 33.)
- [49] Strasdat, Hauke, Maren Bennewitz y Sven Behnke: *Multi-Cue Localization for Soccer Playing Humanoid Robots*. páginas 1–12. (Cita en página 33.)

- [50] Szeliski, Richard: *Computer Vision : Algorithms and Applications*. Springer, 2010. (Cita en página 23.)
- [51] Tasse, Stefan, Matthias Hofmann y Oliver Urbann: *On Sensor Model Design Choices for Humanoid Robot Localization*. Rob. 2012 Robot Soccer World Cup XVI, páginas 380–390, 2012. (Cita en páginas 33 y 36.)
- [52] Thrun, Sebastian: *Probabilistic Robotics*. 2002. (Cita en página 34.)
- [53] Thrun, Sebastian, Dieter Fox, Wolfram Burgard y Frank Dellaert: *Robust Monte Carlo localization for mobile robots*. Artif. Intell., 128(1-2):99–141, Mayo 2001, ISSN 00043702. <http://linkinghub.elsevier.com/retrieve/pii/S0004370201000698>. (Cita en páginas 32 y 37.)
- [54] Čapek, Karel: *RUR (Rossum's Universal Robots)*. 1920. (Cita en página 9.)
- [55] Welch, G y Gary Bishop: *An introduction to the Kalman filter*. 1995. <http://clubs.ens-cachan.fr/krobot/old/data/positionnement/kalman.pdf>. (Cita en página 36.)
- [56] Ying Jin, ShouKun Wang, Naifu Jiang y Yaping Dai: *A Novel Rehabilitation Evaluation Method on Virtual Reality Based on RRR-I*. En *Virtual Real. Hum. Comput. Interact.*, capítulo 4, página 61. (Cita en página 38.)
- [57] Yuen, H. K., J. Princen, J. Dlingworth y J. Kittler: *A Comparative Study of Hough Transform Methods for Circle Finding*. Proceedings Alvey Vis. Conf. 1989, páginas 29.1–29.6, 1989. <http://www.bmva.org/bmvc/1989/avc-89-029.html>. (Cita en página 26.)
- [58] Zhang, Z.: *A flexible new technique for camera calibration*. IEEE Trans. Pattern Anal. Mach. Intell., 22(11):1330–1334, 2000. (Cita en página 40.)
- [59] Zunino, Guido: *Simultaneous Localization and Mapping for Navigation in Realistic Environments*. Tesis de Doctorado, 2002, ISBN 9172832460. http://nyx-www.informatik.uni-bremen.de/792/1/zunino_thesis_02.pdf. (Cita en página 36.)
- [60] Zunt, Dominik: *Who did actually invent the word robot, and what does it mean?* <http://capek.misto.cz/english/robot.html>. (Cita en página 9.)

DECLARACIÓN

Yo declaro aquí que esta tesis titulada Localización en Robots Humanoides Mediante Regresión Sobre Vistas Sintéticas, es resultado de mi propia investigación, con excepción de aquello citado en las referencias.

México, MARZO 2013



Baruch Demian Gaxiola
Valles