

Tesis de Máster



Universidad del País Vasco Euskal Herriko Unibertsitatea

Algoritmo de reconocimiento de forma y color para una plataforma robótica

Jon Aristondo Etxeberria

Donostia - San Sebastián, septiembre 2010

Universidad del País Vasco / Euskal Herriko Unibertsitatea
Departamento de Ciencia de la Computación
e Inteligencia Artificial

Directores: Elena Lazkano Ortaga
Basilio Sierra Araujo

www.ccia-kzaa.ehu.es

Índice:

1- Introducción.	1
2- Estado del arte.	1
2.1 Color.	1
2.1.1 RGB	2
2.1.2 YUV	3
2.1.3 HSL	3
2.2 Forma.	4
2.3 Estimación de la posición del robot.	5
2.3.1 Estimadores explícitos	5
2.3.1.1 Sistema odométrico.	6
2.3.1.2 Sistema de navegación inercial.	7
2.3.1.3 Estimación basada en estaciones de transmisión.	8
2.3.2 Estimadores basados en la percepción del entorno	9
2.3.2.1 Estimación mediante marcas o balizas.	10
2.3.2.2 Posicionamiento basado en mapas.	11
3- Especificaciones del problema.	12
4- Metodología.	14
5- Color tracking.	14
5.1 RGB.	14
5.2 YUV.	17
5.3 HSL.	18
6- Reconocimiento de formas	19
7- Arquitectura de control.	20
8- Experimentación y resultados.	23
9- Conclusiones.	26
10- Referencias.	27

Índice de figuras:

2.1- Tetraedro de color RGB.	2
2.2- Modelo de color HSL.	4
2.3- Esquema estimadores explícitos.	5
2.4- Dibujo de un robot.	6
2.5- Órbita de los satélites empleados para el GPS.	9
2.6- Esquema de los estimadores basados en la percepción.	7
3.1- Robot recoge pelotas.	8
4.1- Metodología empleada.	10
5.1- Sistema de coordenadas cartesianas de RGB.	11
5.2- Imagen capturada de la cámara.	12
5.3- Imágenes filtradas para los distintos colores (amarillo, azul y rojo)	12
5.4- Imagen capturada de la cámara.	13
5.5- Imágenes filtradas para los distintos colores (amarillo, azul y rojo)	14
5.6- Imagen capturada de la cámara.	14
5.7- Imágenes filtradas para los distintos colores (amarillo, azul y rojo)	15
6.1- Imagen filtrada con los contornos marcados.	15
6.2- Imagen con los resultados de las formas circulares detectadas.	16
7.1- Diagrama de estados.	17
7.1- Diagrama de uso del filtro de Kalman.	18
8.1- Tabla de resultados de la experimentación iluminación adecuada.	19
8.2- Tabla de resultados de la experimentación iluminación inadecuada.	20
8.3- Eficacia de los distintos modelos de color.	20
8.4- Eficacia según el color de la pelota.	21
8.5- Eficacia según la luminosidad.	21
8.6- Eficacia luminosidad adecuada.	22

1. Introducción

La robótica se puede definir como la ciencia encaminada a diseñar y construir aparatos y sistemas capaces de realizar tareas propias de los seres humanos. Los robots son empleados en diversidad de aplicaciones, desde robots que limpian el salón de nuestras casas, robots soldadores en la industria de la automoción y hasta brazos teleoperados en el transbordador espacial.

El noventa por ciento de los robots trabajan en fábricas y más de la mitad hacen automóviles. Las fábricas de coches están tan altamente automatizadas que la mayoría de los humanos supervisan o mantienen los robots y otras maquinas. Aunque la mayoría de los robots se utilicen en el mundo de la automoción, existen múltiples tipos de robots, cuya complejidad va desde aquellos que se utilizan en tareas industriales hasta llegar a los que presentan aspecto humano

Existen robots que necesitan tener la capacidad de percepción del entorno en el que se sitúan para trabajar autónomamente. En estos casos se necesitan técnicas para recibir información del entorno mediante sensores montados en los propios robots. Una de las técnicas más empleadas es la colocación de cámaras en los robots y el empleo de la visión artificial para tener una percepción del entorno. Gracias a esta técnica es posible detectar determinados objetos en los que el robot deba centrar su atención.

Existen infinidad de trabajos en los que se pueden detectar distintos objetos bien por la forma del objeto o bien por el color que tenga. Como se muestran en los siguientes trabajos [9][14] es posible detectar y seguir la trayectoria de una pelota de tenis basándose en la forma de la pelota, gracias a las imágenes capturadas de una cámara. Ren et al [10][11] también presentan técnicas para detección y seguimiento de un balón de fútbol basándose en la forma de los objetos.

Por otro lado [6][3][13] presentan sus técnicas de detección de objetos, pero en este caso se basan en el color del objeto en vez de la forma del mismo. En otros casos como en [7] la detección de objetos se basa tanto en la forma como en el color del objeto que se quiere reconocer. Aunque también es posible detectar objetos mediante las diferencias entre las imágenes capturadas en momentos diferentes y centrándose en los objetos que han cambiado de posición, como se hace en [8][15].

En este trabajo se describe el sistema de control de un robot que recoge de forma autónoma una pelota de color amarillo, rojo o azul en un recinto abierto sin determinar. Permite además oclusiones temporales de la pelota en el proceso de recogida de la pelota gracias a la utilización de filtros de Kalman.

El documento está estructurado de la siguiente manera: la sección 2 muestra el estado del arte del proyecto. En la sección 3 se muestran los requisitos y las especificaciones del problema. La sección 4 muestra la metodología empleada. Las secciones 5 y 6 explican el proceso de detectar la pelota y en la sección 7 se detalla la arquitectura del control del sistema. Por último se muestran la experimentación y los resultados en la sección 8 y las conclusiones en la sección 9.

2. Estado del arte

Las técnicas que se han empleado en la realización de este proyecto han sido el reconocimiento de colores y la detección de formas. También nos centraremos en los diferentes métodos para la estimación de la posición del robot.

2.1 Color

Una opción interesante es utilizar las características de color para el seguimiento visual, debido a su simplicidad y robustez ante oclusiones parciales y cambios de escala. Sin embargo existen varios problemas a tratar, entre los que destaca la constancia de color [4] ya que le afectan las sombras, cambios de iluminación y características de la cámara. Si se considera el seguimiento de color como un método apropiado para aplicaciones de tiempo real (como es el caso del robot móvil recoge pelotas), se necesita una presentación robusta y eficaz de los objetos de color para que el procesamiento sea preciso y veloz. El modelo de color juega un papel importante en la precisión computacional y la robustez del sistema. A continuación se describen algunos de los modelos de color empleados en este proyecto.

2.1.1 El modelo RGB

Éste es un modelo de color basado en lo que se conoce como síntesis aditiva, con lo que es posible representar un color por la mezcla por adición de los tres colores primarios con los que se forma: rojo, verde y azul. Le asignamos un valor a cada uno de los colores primarios para indicar con qué proporción se mezcla cada color. Así, por ejemplo, el valor 0 significa que no interviene en la mezcla, y en la medida que ese valor aumenta, aportará más intensidad a la mezcla, hasta llegar al valor 255 o 1 si se normaliza. El valor de cada color primario se representa con un byte por lo que su valor puede llegar hasta 255.

Este modelo está basado en el sistema de coordenadas cartesianas. El subespacio de color de interés es el tetraedro mostrado en la figura 2.1, en el cual los valores RGB están en tres vértices; cyan, magenta y amarillo se sitúan en otros tres vértices, el negro corresponde al origen y el blanco en el vértice más alejado del origen.

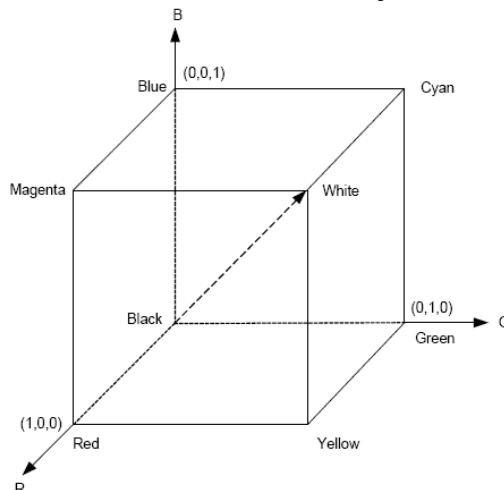


Figura 2.1: Tetraedro de color RGB

En este modelo la escala de grises se extiende desde el negro al blanco a lo largo de la diagonal que une esos dos puntos, y los colores son puntos dentro del tetraedro, definidos por vectores desde el origen.

2.1.2 El modelo YUV

El modelo YUV define un espacio de color en términos de una componente de luminancia (Y) y dos componentes de crominancia (U,V). Codifica una imagen en color teniendo en cuenta la percepción humana, de este modo hace que sea más robusto ante cambios de iluminación. Este modelo es el utilizado en los sistemas de difusión de televisión PAL y NTSC, los cuales son los estándares en la mayoría del mundo.

Es posible obtener los valores YUV de un píxel partiendo de los valores RGB del mismo mediante la utilización de la ecuación (1):

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

Se asume que R, G y B están en el rango 0 a 1, con 0 representando la intensidad mínima y 1 la máxima. Y está en el rango 0 a 1, U está en el rango -0.436 a 0.436 y V está en el rango -0.615 a 0.615.

Éste modelo de color es el empleado en la función recogepelotas desarrollado por Tekniker previamente a este proyecto. Entonces solo se tenía en cuenta el color para discriminar la pelota amarilla en un entorno cerrado de color negro.

2.1.3 El modelo HSL

El modelo HSL es un modelo de representaciones considerado "natural", ya que se acerca bastante a la percepción fisiológica del color que tiene el ojo humano. En efecto, el modelo RGB (rojo, verde, azul) puede resultar adecuado para la representación de colores en el equipo o para su presentación en dispositivos de visualización, pero no permite seleccionar los colores con facilidad.

El modelo HSL define el espacio de color con tres características: tono (*Hue*), saturación (*Saturation*) y luminosidad (*Luminance*). Se denomina longitud de onda dominante a la luz que se ve y que corresponde a una determinada cantidad de tono.

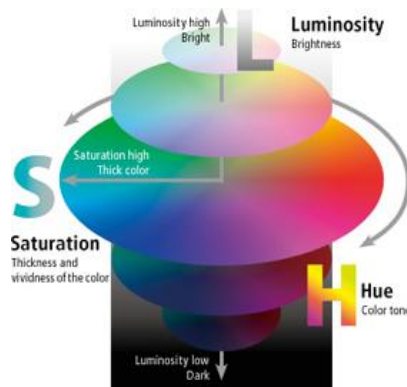


Figura 2.2 Modelo de color HSL

La saturación corresponde a la pureza del color. Corresponde a la proporción de la luz pura de longitud de onda dominante y la luz blanca necesaria para definir un color. Un color 100% puro tiene una saturación del 100% y no contiene luz blanca. Colores que se alejen del color puro contienen una mezcla entre la luz blanca y el color puro comprendida entre el 0% y 100%.

La luminosidad es la cantidad de luz que tiene un color determinado. Cualquier valor de tono con una luminosidad del 100% será el color blanco. En cambio si presenta un valor de 0% de luminosidad se tratará del color negro.

2.2 Forma

Si bien es relativamente sencillo que una máquina capte lo que le rodea, no lo es tanto el conseguir que sea capaz de reconocerlo. En el área de procesamiento de imágenes, la detección de los bordes de una imagen es de suma importancia y utilidad, pues facilita muchas tareas, entre ellas, el reconocimiento de objetos, la segmentación de regiones, entre otras.

Los bordes de una imagen digital se pueden definir como transiciones entre dos niveles de gris significativamente distintos. Suministran una valiosa información de fronteras de los objetos y pueden ser utilizadas para el reconocimiento de objetos.

Para el reconocimiento de formas se ha empleado un reconocedor de formas geométricas el cual emplea internamente la detección de bordes. Existen multitud de técnicas para la detección de bordes y a continuación se presentan algunos de ellos.

Algunas técnicas de reconocimiento de bordes como Sobel o Prewitt [9] emplean una máscara 3x3 para detectar los bordes. También existen otros filtros como el Laplaciano del Gaussiano, los filtros basados en el momento, los operadores de Shen y Castan [12], los operadores de Canny [1] y Deriche [2]. Algunos problemas comunes de todos estos son la sensibilidad al ruido, el alto costo computacional y la no isotropía, es decir, que su comportamiento cambia según en qué dirección se encuentre el borde del objeto.

Los beneficios de emplear un clasificador difuso para la detección de bordes que requieren un bajo costo computacional, tienen baja sensibilidad al ruido, son isotrópicos, es decir que su comportamiento es el mismo sin importar en qué dirección se encuentre el borde, y de fácil modelado.

2.3 Estimación de la posición del robot

El sistema de percepción de un robot móvil permite que éste se mueva libremente en un entorno desconocido de una forma autónoma. Para determinar la posición, la percepción del entorno que tiene el robot debe de ser alta por lo que el robot debe ir equipado de un sistema sensorial que le suministre información abundante y lo más precisa posible.

A continuación se presentan los distintos métodos existentes para la estimación de la posición de un robot móvil. En primer lugar hay que diferenciar entre dos tipos de estimadores: por un lado los que no requieren adquisición e interpretación de la información del entorno del robot y por otro los estimadores basados en la percepción del entorno. En el primer caso destacan los métodos basados en la odometría, sistemas de navegación inercial, estimación absoluta de la posición (GPS) y la estimación de la posición mediante triangulación con respecto a marcas situadas en el entorno. Entre los estimadores basados en la percepción del entorno se distinguen entre los sensores activos (laser, sonar) y los sensores pasivos (cámaras de video).

2.3.1 Estimadores explícitos

En esta categoría se encuentran aquellos sistemas que son capaces de estimar la posición del vehículo sin necesidad de capturar información del entorno. Entre este tipo de estimadores se distinguen dos grupos: los basados en medidas internas y los basados en estaciones de transmisión.

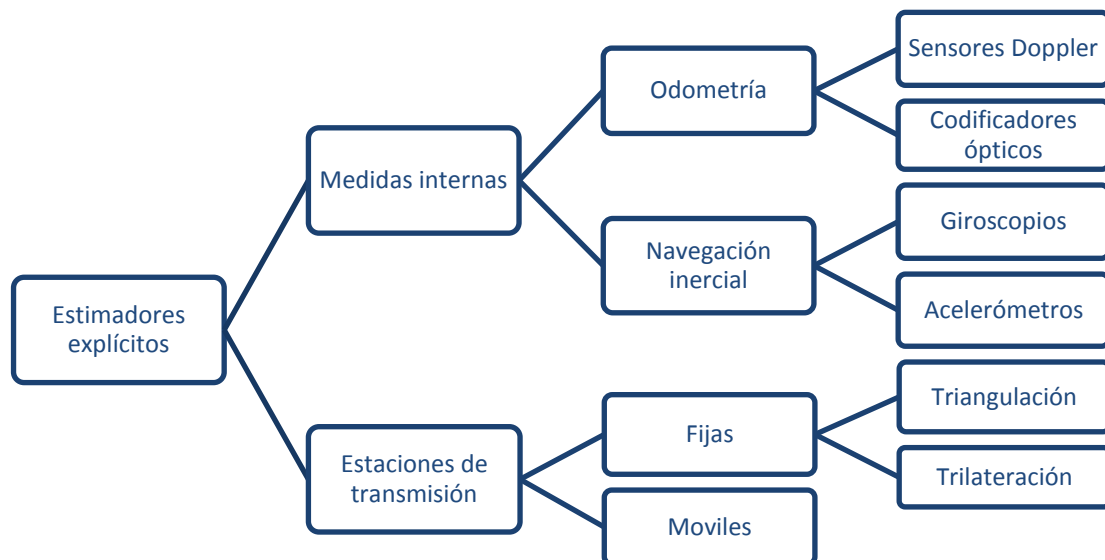


Figura 2.3: Esquema estimadores explícitos

Los primeros trabajan sin ningún tipo de información exterior, con sensores integrados en el vehículo como codificadores, giroscopios, brújulas, acelerómetros, tacómetros, etc. Los segundos, en cambio están configurados en base a dos unidades bien diferenciadas, una montada sobre el vehículo y la otra u otras colocadas en posiciones conocidas del entorno. La unidad montada en el robot actúa como sensor receptor, mientras que las externas actúan como emisores o señales de referencia.

Obsérvese que, en esta clasificación, las unidades emplazadas en el medio no tienen por qué ser activas. Así, por ejemplo, un sistema basado en la localización mediante cámaras de vídeo y de marcas artificiales colocadas a tal fin en el entorno (código de barras, señales, etc.), será englobado dentro de los estimadores explícitos, puesto que, aunque puede entenderse que este proceso conlleva la percepción del entorno, la posición no es estimada a partir del análisis ó interpretación del entorno percibido, sino que es determinado de una forma más o menos directa por triangulación.

Para estimar la posición y la orientación de un robot móvil, la manera más simple es saber la trayectoria recorrida por éste empleando una serie de medidas: vueltas dadas por las ruedas, velocidades, aceleraciones, cambios de orientaciones, etc. Dependiendo de la información que se utilice, se distinguen dos grupos: sistemas odométricos y sistemas de navegación inercial.

2.3.1.1 Sistema odométrico

La odometría es una técnica que tiene como objetivo estimar la posición y orientación de un robot en la navegación, contando el número de vueltas que dan las ruedas al desplazarse. Actualmente para llevar la cuenta del número de vueltas y fracciones de éstas que realizan las ruedas, se utilizan codificadores ópticos (*encoders*).

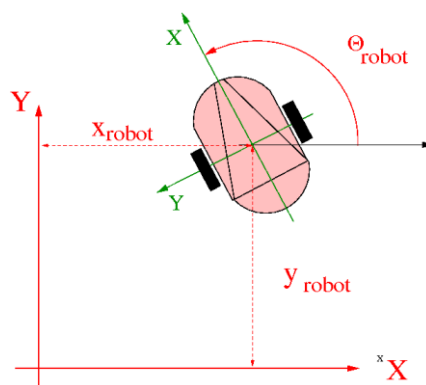


Figura 2.4: Dibujo de un robot

La gran ventaja de la odometría es la simplicidad y su bajo coste pero su punto flaco es que el error se va acumulando con el movimiento del robot. En concreto la acumulación de errores de orientación causa grandes errores de estimación de la posición. Estos errores aumentan proporcionalmente con la distancia recorrida del robot.

Los errores se pueden clasificar en dos grupos (sistemáticos y no sistemáticos) según cuál sea el origen de este.

Entre los errores sistemáticos se encuentran los siguientes:

- Distintos diámetros de las ruedas
- Diámetros nominales de las ruedas distintos de los diámetros reales
- Desalineación de las ruedas
- Incerteza acerca del valor real de la distancia entre ruedas (debido a que no se sabe donde apoya la rueda en realidad)
- Una baja resolución del encoder
- Baja tasa de captura del valor del encoder

Por otro lado están los errores no sistemáticos entre los que se encuentran:

- Desplazamiento del robot por superficies irregulares
- Desplazamiento sobre objetos inesperados
- Ruedas que patinan debido a
 - Superficie resbaladizo
 - Exceso de aceleración
 - Giros excesivamente rápidos
 - Que alguna de las ruedas no toque el suelo
 - Fuerzas externas o internas que modifiquen el curso

Los errores sistemáticos pueden llegar a ser corregidos o bien tenidos en cuenta para disminuir el error de odometría pero los no sistemáticos se pueden intentar reducir pero nunca podrán ser eliminados.

2.3.1.2 Sistema de navegación inercial

Un sistema de navegación inercial (INS) es un sistema de ayuda a la navegación que estima la posición y la orientación del robot empleando las medidas de movimiento del acelerómetro y la de rotación del giroscopio.

Los acelerómetros suelen estar basados en sistemas pendulares. La precisión del acelerómetro resulta crítica, ya que, la alineación, errores de factor escala y errores de compresión son cruciales, porque pequeños errores cometidos repercuten notablemente en la posición estimada.

Como sucede en los sistemas odometricos, a los sistemas de navegación inercial no les afectan los problemas causados en la interacción del robot con el suelo y las regularidades causadas por las irregularidades del entorno. Esta es la principal razón por la que los sistemas de navegación inercial son más fiables y precisos que los sistemas odometricos. En cambio, tienen un coste mayor que los sistemas odometricos y son más frágiles.

Aunque las dos técnicas sean fáciles de implementar tienen un grave inconveniente como estimadores de la posición absoluta: los errores se van acumulando por lo que la

incertidumbre aumenta proporcionalmente al espacio recorrido por el robot. Por este motivo estos sistemas no suelen utilizarse en solitario, sino que son complementados con otros sistemas de posicionamiento que reducen la incertidumbre cada determinado tiempo.

2.3.1.3 Estimación basada en estaciones de transmisión

Este sistema de posicionamiento absoluto hace décadas que se utiliza en aplicaciones marítimas y aeronáuticas. Con la aparición de robots móviles que se mueven en lugares abiertos estos sistemas han cobrado un especial interés. La principal ventaja que tiene este sistema es que proporciona la localización absoluta de un vehículo en un área, sin tener ninguna información del entorno. Estas características hacen de este sistema un sistema apropiado cuando el vehículo se emplea en espacios grandes y al aire libre. Estos sistemas están formados por un receptor a bordo del vehículo y un conjunto de estaciones transmisoras que pueden ser fijas o móviles.

Las estaciones de posicionamiento mediante estaciones fijas pueden ser básicamente de dos tipos:

- **Triangulación:** Este método también es conocido como método de navegación hiperbólica. El sistema compara los tiempos que tarda en recibir la misma señal desde dos puestos de transmisión situados en un lugar conocido. De este modo consigue estimar la distancia que hay entre el robot y las distintas estaciones. Para evitar errores a la hora de estimar la posición, son necesarias como mínimo tres estaciones de transmisión.
- **Trilateración:** A estos sistemas se les conoce como sistemas activos y determinan la posición midiendo el retardo que tiene la señal que manda las estaciones emisoras al robot. Pueden recibir información desde transceptores situados hasta a 100 kilómetros de distancia, con un error que no supera los dos metros. El funcionamiento de la técnica se resume así: el robot móvil emite una señal que es captada por varios transceptores que devuelven la señal de nuevo al robot. El robot mide el retardo que tiene la señal que manda a los transceptores para determinar la distancia que existe entre ellos.

Por otro lado se sitúan los sistemas de estaciones móviles que son los que mayor interés tienen para robots móviles. En este apartado el sistema más conocido y utilizado en la actualidad es el GPS (Global Positioning System). Los sistemas móviles reciben la información que emiten 24 satélites móviles que se mueven alrededor de la tierra en 6 orbitas diferentes. El receptor emplea por lo menos tres satélites y calcula la altitud, latitud y la velocidad del vehículo por triangulación.

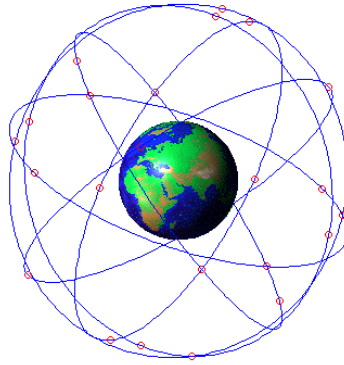


Figura 2.5: Órbita de los satélites empleados para el GPS

La precisión del GPS depende de distintos aspectos entre los que destacan la alteración de la velocidad de propagación de la señal, errores en el posicionamiento orbital, etc. La estimación que ofrece este sistema no es exacta puesto que el sistema siempre ha estado restringido al uso militar, aunque con la corrección de errores se puede incrementar la precisión hasta un par de metros.

2.3.2 Estimadores basados en la precepción del entorno

Estos estimadores son los que dotan a los robots móviles de un sistema sensorial capaz de proporcionar información del entorno para que éste pueda de forma autónoma saber su posición. Para dotar a los robots de ese sistema sensorial existen dos tipos de estimadores, los sensores activos y los pasivos.

Los sensores activos son aquellos que emiten algún tipo de energía al entorno. Entre los más conocidos se encuentran la luz infrarroja, laser, ultrasonidos, ondas de radio, etc. Estos sensores proporcionan directamente medidas de distancias al entorno, lo que les dota de unas prestaciones muy acordes con los requerimientos necesarios para la estimación de la posición de un robot móvil.

El funcionamiento más empleado es el denominado “tiempo de vuelo”, el tiempo que tarda el eco de la señal que emite el emisor en volver a ser recibido por el receptor. A partir de esta medición y teniendo en cuenta la velocidad de propagación de la señal, es posible estimar la distancia existente entre el emisor y el objeto. Esto presenta un problema cuando se encuentran varios robots cerca ya que pueden producirse interferencias entre sus señales y dar estimaciones incorrectas. Otra característica importante de los sensores activos es que necesitan más energía y más procesamiento de los datos obtenidos que los sensores pasivos.

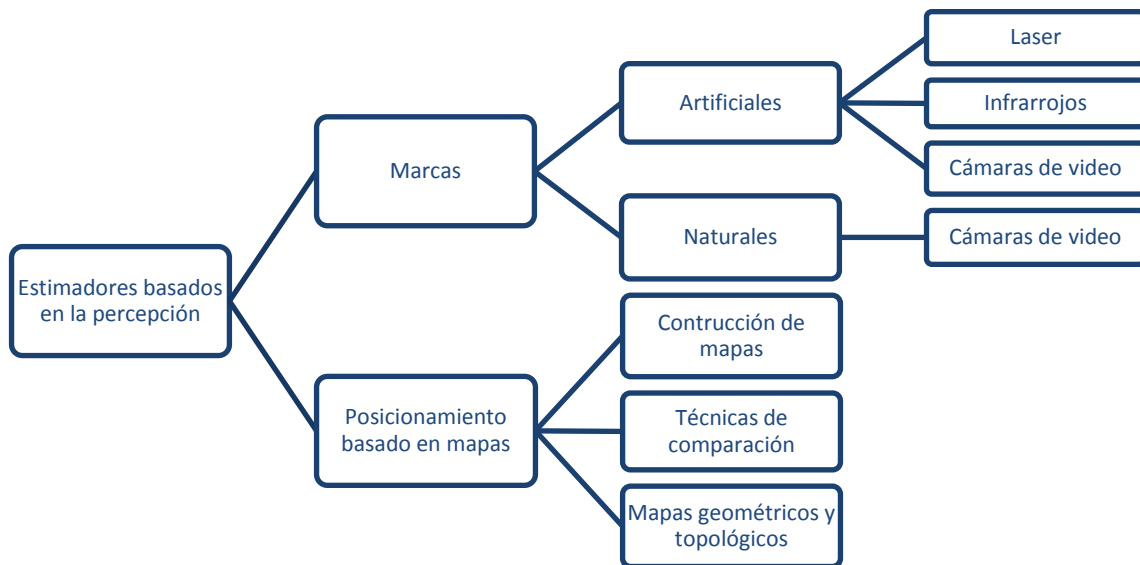


Figura 2.6: Esquema de los estimadores basados en la percepción

Los dos métodos más utilizados en la robótica son el anillo de sonares y el dispositivo de rastreo radial. Sus principales ventajas su bajo coste y su simplicidad pero son muy vulnerables a reflexiones especulares y al medio. Los sensores laser, aunque de elevado precio, cada vez tienen más cabida en el campo de los robots móviles. La razón por la que se utilizan es porque presentan una gran precisión y fiabilidad. Aunque también presentan algunos inconvenientes relacionados con las partes mecánicas que incorporan, hoy por hoy, son los sensores que ofrecen mayores prestaciones para aplicaciones en robots móviles.

Por otro lado, se sitúan los sensores pasivos, donde entre los más utilizados destacan las cámaras de video. Entre las ventajas que presentan las cámaras destacan la alta precisión y velocidad al adquirir la información, el bajo consumo, la independencia sobre los dispositivos mecánicos y son relativamente baratos. En cambio existe una fuerte dependencia con las condiciones ambientales de iluminación y una excesiva complejidad computacional a la hora de recoger información tridimensional.

Existen varias técnicas para estimar la posición del vehículo que pueden ser presentadas en dos categorías. La primera técnica es la de obtener la distancia de los objetos de la imagen respecto a la cámara. La segunda en cambio la estimación de la posición se hace emparejando imágenes extraídas de la imagen con el modelo conocido.

2.3.2.1 Estimación mediante marcas o balizas

Las marcas o balizas son características del entorno que se emplean para localizar el robot. Pueden ser naturales o ser introducidas artificialmente en el entorno. Esta técnica se sitúa entre las que interpretan el entorno aunque la estimación de la posición en realidad no se hace dependiendo del entorno sino que por las marcas detectadas.

Las marcas naturales son objetos o características del entorno y no tienen la funcionalidad de posicionar el robot aunque se empleen para eso. En la navegación con marcas naturales, una de las tareas más difíciles es la detección de las marcas. Por lo general, en la búsqueda de marcas naturales se emplean las cámaras de video y como marcas se trata de detectar bordes verticales de gran dimensión que pueden corresponder por ejemplo a marcos de puertas o paredes. En estos casos la dificultad está en posicionar la marca en el entorno para posteriormente posicionar el robot en ese entorno.

Por el otro lado se encuentran las marcas artificiales que pueden ser diferentes y variados. Por lo general se emplean formas geométricas o colores pero también existe la posibilidad de emplear distintos códigos de barras o incluso sistemas de infrarrojos o laser más avanzados y precisos. La característica más importante que las diferencia de las marcas naturales es que en este caso la posición de la marca en el entorno es conocida por lo que facilita la navegación del robot móvil. Una vez detectadas las marcas necesarias la posición del robot se estima empleando el método de triangulación. Este método determina la posición empleando bien distancias, ángulos o las dos.

2.3.2.2 Posicionamiento basado en mapas

El posicionamiento basado en mapas es una técnica en la que el robot utiliza sus sensores para crear un mapa de su entorno. El mapa local se compara con un mapa global previamente almacenado en la memoria. Si existe una relación es posible estimar la posición y la orientación del robot en el entorno.

Las principales ventajas que tiene el posicionamiento basado en mapas son las siguientes:

- Este método ofrece información de posición sin modificar el entorno en el que se mueve el robot.
- Se pueden utilizar para tener un mapa actualizado del entorno. Los mapas son importantes para otras tareas de la robótica móvil como son la planificación de rutas y la evitación de obstáculos.
- El posicionamiento basado en mapas permite al robot aprender nuevos entornos y mejorar la precisión de la localización mediante la exploración del entorno.

En cambio, las desventajas que tiene esta técnica de posicionamiento son básicamente los requerimientos que tiene para satisfacer la navegación:

- Debe haber suficientes características fácilmente detectables para emparejar.
- El mapa de sensores debe ser lo suficientemente preciso para que sea útil.

Existen dos puntos de partida fundamentales en el posicionamiento basado en mapas: hay un mapa existente (comparación de mapas) o el robot debe crear uno desde cero (construcción de mapas).

3. Especificaciones del problema

El robot recogepelotas es un robot fabricado por Tekniker cuyas diferentes partes son componentes comerciales que se puede obtener en el mercado. La base de este robot es la plataforma móvil Create de la compañía iRobot que tiene las siguientes características:

- Programable. La plataforma se comercializa totalmente montada e incorporando más de 30 sensores, incluyendo sensores de contacto.
- Plataforma robusta y probada. Es la base de la aspiradora robótica Roomba, una aspiradora comercial que ha vendido varios millones de unidades en el mundo.
- Tamaño reducido. Su tamaño reducido y su tracción diferencial lo hacen idóneo para desempeñar la función de recogepelotas.
- Precio. Su precio es reducido comparando con otras plataformas robóticas.

En la figura 3.1 se puede observar el robot recogepelotas. Para poder detectar la pelota se ha dotado al robot de una cámara AXIS 207W. Esta cámara permite capturar 30 fotogramas por segundo a una resolución de VGA (640x480) pero debido al coste computacional las imágenes utilizadas durante todo el proyecto han sido de una resolución de 320x240 píxeles. La comunicación con la cámara se establece de manera inalámbrica a través de Wi-Fi.



Figura 3.1: Robot recogepelotas

Para que el robot pueda coger la pelota, la plataforma móvil se ha dotado de un brazo robótico Lynxmotion SES. La capacidad de carga de este brazo es de 200 gramos aproximadamente, por lo que no tiene ningún impedimento para levantar la pelota que pesa unos 150 gramos. Además, las dimensiones y el peso, 38 centímetros completamente abierto y 600 gramos respectivamente, lo hacen idóneo para colocarlo en el robot Create.

El microcontrolador de 8 bits Atmel AVR Atmega168 que incorpora el robot no ofrece suficiente capacidad de procesamiento como para tratar las imágenes de la cámara y ejecutar el algoritmo de control, por lo que la ejecución del sistema se realiza en un ordenador externo. El ordenador utiliza una conexión Wi-Fi para comunicarse con la cámara y una conexión Bluetooth para comunicarse con el robot. También existe la posibilidad de comandar el robot remotamente utilizando un mando a distancia, mediante la voz, un mando de la consola Wii o por gestos gracias a un guante diseñado para ello.

El software que se ha empleado para comandar el robot ha sido Microsoft Robotics Studio (Microsoft Robotics Developer Studio). En cambio para el tratamiento de las imágenes se han empleado las librerías Matrox Imaging Library (MIL).

Microsoft Robotics Studio es un entorno basado en Windows para el control y la simulación de procesos robóticos. Se trata de una herramienta de programación visual, que permite que el usuario escriba las rutinas de software que controlarán el comportamiento de su robot. Se puede emplear alguno de los lenguajes de programación soportados por la plataforma .NET, como Visual Basic, C o C++. El desarrollador también puede interactuar con sus robots mediante interfaces basadas en Web.

Matrox Imaging Library (MIL) son unas librerías comerciales con un conjunto de funciones de alto nivel para adquisición, transferencia y presentación de imágenes, así como para procesamiento y análisis de las mismas. Son unas librerías exclusivas para plataformas Linux y Windows. Las aplicaciones MIL para Windows están desarrolladas utilizando Visual C++ 6.0 y Visual C++.NET 2003. Las implementadas para Linux han sido desarrolladas usando GNU.

Entre los principales recursos de las librerías MIL se encuentra las descritas a continuación, que son las que se han empleado para tratar las imágenes capturadas de la cámara durante este proyecto:

- **Edge Finder:** Es una herramienta que posibilita la extracción de los bordes o contornos de los elementos de una imagen. El *Edge Finder* se puede emplear para detección de defectos, análisis y reconocimiento de formas. Permite la utilización de distintos algoritmos para la extracción de bordes.
- **Geometric Model Finder:** Este instrumento permite detectar diferentes formas geométricas en una imagen mediante la utilización de los contornos de los elementos de la misma. Es posible que busque estas formas geométricas en los resultados obtenidos utilizando el anteriormente nombrado *Edge Finder*.

4. Metodología

En el tratamiento de la imagen capturada de la cámara el objetivo principal es identificar la pelota y determinar cuáles son las coordenadas del punto medio de la misma. Para este fin se sigue el proceso que se muestra en la figura 4.1.

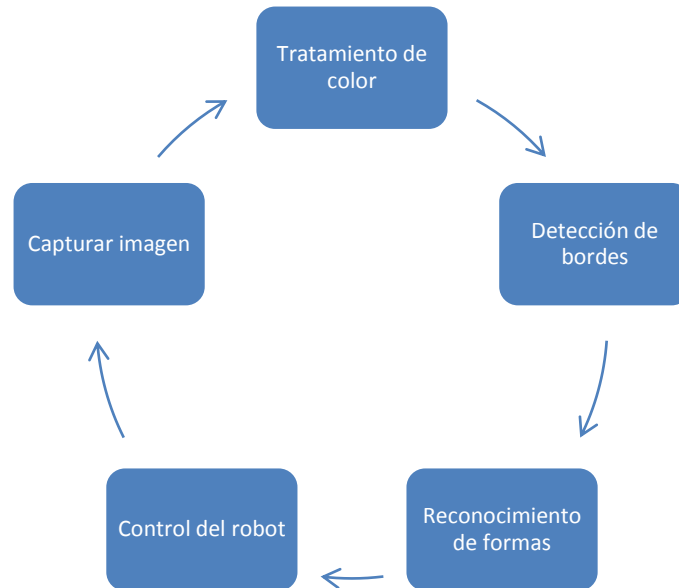


Figura 4.1 Metodología empleada

En primer lugar se captura una imagen de la cámara montada en el robot y se transmite al ordenador encargado de todo el proceso mediante Wi-Fi. A continuación se le aplica el filtro de color correspondiente al color de la pelota que se quiere recoger. Luego se detectan los bordes en la imagen filtrada empleando el módulo *Edge Finder* de las librerías MIL para después detectar formas circulares sobre esos bordes con el módulo *Model Finder*. Finalmente tras tratar los resultados obtenidos por el *Model Finder* se consigue la posición de la pelota en la imagen y se comanda el robot para que se acerque hacia ella mientras se vuelve a repetir todo el proceso con la siguiente imagen que se recoge de la cámara.

5. Color tracking

Como ya se ha comentado en el capítulo anterior los métodos empleados serán RGB, YUV y HSL. A continuación se profundizará en cada uno de ellos para la realización de un filtro para discriminar el color amarillo de la imagen capturada por la cámara.

5.1 RGB

Durante el control de color existe la necesidad de saber si un color es el mismo que el de la pelota que se quiere recoger. Para representar un color se utiliza el siguiente sistema (R,G,B), siendo R, G y B el valor representado en un byte de cada color

primario. Por lo tanto, el rojo se obtiene con (255,0,0), el verde con (0,255,0) y el azul con (0,0,255), obteniendo, en cada caso un color resultante monocromático. La ausencia de color, lo que se conoce como negro, se obtiene cuando las tres componentes son 0, (0,0,0).

En el caso de este proyecto la pelota que se quiere buscar puede ser amarilla (255,255,0), roja (255,0,0) o azul (0,0,255). Y la pregunta es: ¿cómo saber que el color que se tiene es el deseado?

Como se puede ver en la figura 5.1 se pueden tratar los componentes RGB como un sistema de coordenadas cartesianas y calcular la distancia entre dos colores como el módulo del vector que los une. La distancia que nos interesa será la que hay al color deseado puro.

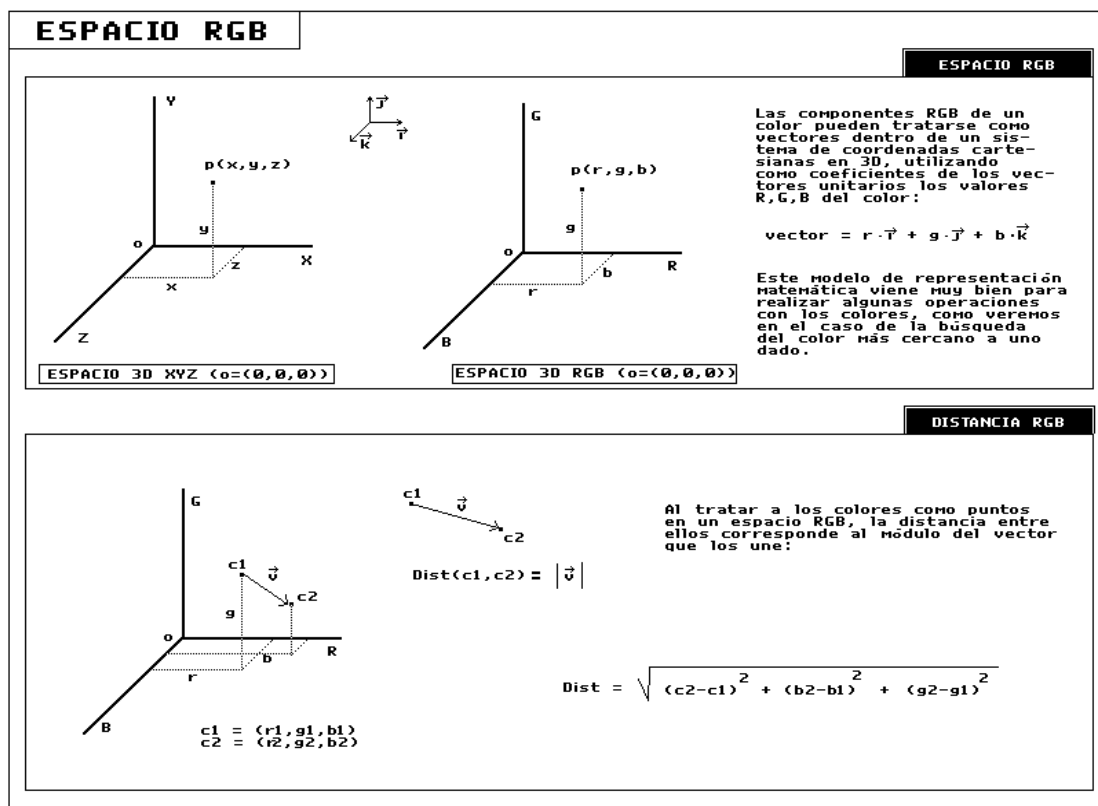


Figura 5.1 Sistema de coordenadas cartesianas de RGB

El problema de este método es que hay colores que no interesaría considerarlos como deseados y tienen menor distancia respecto al color objetivo que otros colores que se considerarían como color deseado. Éste es el motivo por el que este método de clasificar colores fue descartado.

Sabiendo que el color amarillo puro es (255,255,0) y que en general toda la gama de amarillos tienen los valores de R y G parecidos y muy altos respecto al B se hace un filtro para decidir que colores serán considerados “amarillos” y cuáles no. Se consideraran amarillos todos los colores que cumplan la función (2):

$$|R - G| < 40 \text{ y } (R + G)/2 > (B + 100) \quad (2)$$

Con el color rojo sucede algo parecido sabiendo que toda la gama de rojos tienen valores de componente R altos y los valores de G y B parecidos. Por lo que se considerarán “rojos” todos aquellos los colores que cumplan las condiciones mostradas en (3):

$$|G - B| < 80, R > 120 \text{ y } (R + B)/2 < (R - 70) \quad (3)$$

El color azul es el más difícil de discriminar porque los valores RGB de los colores que se consideran “azules” son los que más se extiende en el modelo de color con el cambio de iluminación. Así que se consideraran “azules” los colores que cumplan las condiciones (4) o (5):

$$|G - R| < 80, B > 120 \text{ y } (R + G)/2 < (R - 60) \quad (4)$$

o

$$|G - R| < 40, B > 90 \text{ y } (R + G)/2 < (R - 30) \quad (5)$$

En las imágenes 5.2 y 5.3 se presentan la imagen original adquirida de la cámara y las obtenidas después de aplicarles los distintos filtros de color:



Figura 5.2: Imagen capturada de la cámara.



Figuras 5.3: Imágenes filtradas para los distintos colores (amarillo, azul y rojo).

5.2 YUV

Dado que la iluminación del entorno de pruebas del robot varía entre las distintas pruebas, se ha optado por utilizar en modelo de color YUV para crear un filtro para el color de la pelota que se quiere recoger en cada momento. Para discriminar el color deseado de los demás se emplearán las componentes U y V de manera independiente o conjunta.

La imagen es recorrida píxel a píxel midiendo la similitud del valor de las componentes U y V del píxel con el valor de referencia, como indica la función (6):

$$\begin{aligned} \text{similitud}_u &= 1 - (U_i - U_{deseado}) / 0.872 \\ \text{similitud}_v &= 1 - (V_i - V_{deseado}) / 1.23 \end{aligned} \quad (6)$$

donde U_i y V_i son los valores de las componentes U y V del píxel actual y $U_{deseado}$ y $V_{deseado}$ los valores de referencia de los valores U y V respectivamente.

Para discriminar los colores amarillos se emplean las dos componentes U y V con los valores de similitud y deseabilidad presentados en (7):

$$\begin{aligned} V_{deseado} &= 0,049989 & V_{similitud} &= 0,95 \\ U_{deseado} &= -0,14 & U_{similitud} &= 0,95 \end{aligned} \quad (7)$$

En el caso del color rojo basta con el componente V para discriminar toda la gama de rojos:

$$V_{deseado} = 0,17166 \quad V_{similitud} = 0,95 \quad (8)$$

Por último para diferenciar los colores azules también es necesaria la utilización de las dos componentes con los valores de (9):

$$\begin{aligned} V_{deseado} &= -0,13882 & V_{similitud} &= 0,9 \\ U_{deseado} &= 0,108717 & U_{similitud} &= 0,9 \end{aligned} \quad (9)$$

En las imágenes 5.4 y 5.5 se observan imágenes resultantes de los diversos filtros a partir de la imagen capturada de la cámara.



Figura 5.4: Imagen capturada de la cámara.



Figuras 5.5: Imágenes filtradas para los distintos colores (amarillo, azul y rojo).

5.3 HSL

Al igual que el modelo de color YUV, el HSL se emplea por su robustez ante cambios de iluminación. Con este método al igual que con los anteriores, se recorren todos los píxeles de la imagen capturada de la cámara y se decide si tal píxel es del color que se está buscando o no.

Para el color amarillo se considerarán píxeles amarillos todos aquellos que cumplan las características especificadas en (10):

$$H \in (31, 53) \text{ y } S > 128 \quad (10)$$

En cambio con el color rojo los píxeles deben cumplir las condiciones de (11):

$$H < 11 \text{ o } H > 244 \text{ y } S > 70 \quad (11)$$

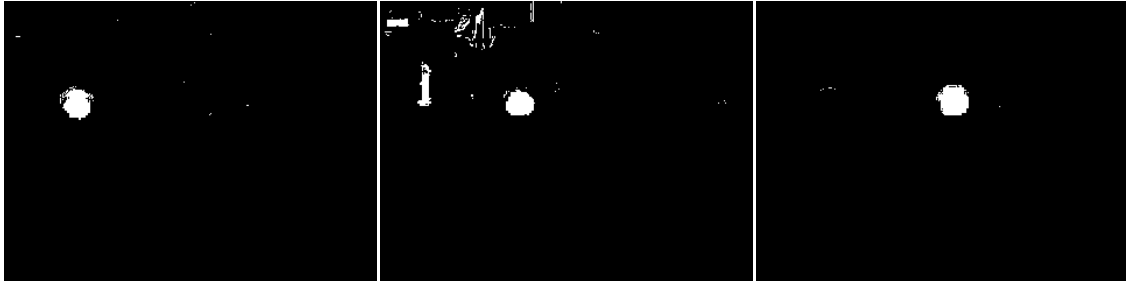
Y por último los píxeles que serán considerados de color azul tendrán que cumplir las características mostradas en (12):

$$H \in (130, 177), S > 90 \text{ y } L \in (20, 175) \quad (12)$$

Se puede observar en las imágenes 5.6 y 5.7 como quedan las imágenes después de aplicarle el filtro de color correspondiente.



Figura 5.6: Imagen capturada de la cámara.



Figuras 5.7: Imágenes filtradas para los distintos colores (amarillo, azul y rojo).

6. Reconocimiento de formas

La primera decisión que hay que tomar en este apartado es con que librerías de tratamiento de imágenes se va a trabajar. En este caso se ha utilizado Matrox Imaging Library (MIL), ya que es una de las librerías empleadas en el ámbito de la visión artificial en el Centro Tecnológico Tekniker. Esta herramienta posee numerosos módulos para el tratamiento de imágenes de las cuales en este proyecto se han empleado la detección de bordes y el reconocimiento de formas geométricas.

El reconocimiento de la pelota se realiza a partir de la imagen en blanco y negro que se obtiene gracias al filtro de color aplicado a la imagen original. Para detectar la pelota en la imagen se sigue un proceso de dos pasos. En primer lugar se detectan todos los bordes que existan en la imagen y posteriormente se pasa a la búsqueda de formas circulares entre los resultados obtenidos en el apartado anterior. A continuación se explica más detalladamente estos dos pasos que se dan.

Las librerías MIL poseen un módulo que permite la búsqueda de bordes en las imágenes con la opción de escoger el algoritmo que más se adecue a las necesidades del usuario. En el caso de este proyecto se ha empleado el algoritmo Shen-Castan para reconocer dichos bordes. En la imagen 6.1 se observa como los contornos quedan marcados sobre la imagen ya filtrada.

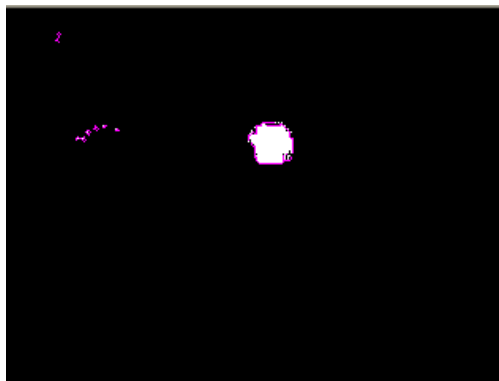


Figura 6.1: Imagen filtrada con los contornos marcados.

Una vez que se obtienen los bordes de la imagen en blanco y negro se procede a la búsqueda de formas circulares entre dichos resultados. Para esta búsqueda se emplean tres reconocedores de formas, una para cada rango de distancia en la que pueda estar la pelota. En la imagen 6.2 se observan las formas circulares detectadas en los contornos obtenidos por el paso anterior mediante el reconocedor de formas circulares.

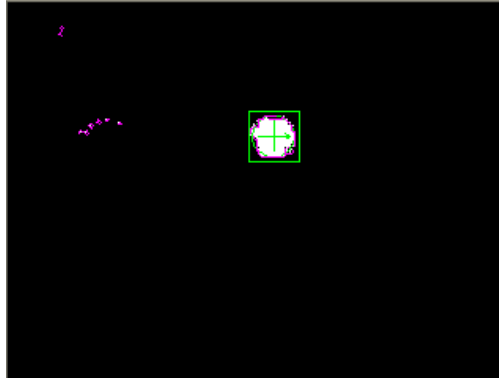


Figura 6.2: Imagen con los resultados de las formas circulares detectadas

Aunque en la figura 6.2 se observe una forma circular detectada es posible que se detecten más de un círculo. Siendo esto así es necesario diferenciar el círculo que pertenece a la pelota de los demás círculos. Para tomar la decisión de que círculo es la correspondiente a la pelota se observan los parámetros que devuelve el detector de formas circulares.

7. Arquitectura de control

El control del robot se ejecuta mediante el empleo de la herramienta Microsoft Robotics Studio y para implementar esta aplicación se ha definido una maquina de estados. La figura 7.1 muestra el diagrama de los estados identificados y las transiciones entre ellos.

El proceso de coger la pelota lo componen cuatro estados. El robot comienza en el estado *buscando pelota*. Bajo este estado el robot gira sobre su eje y analiza las imágenes que obtiene en busca de la pelota. Si la detecta, el estado lógico del robot pasa a ser el de *pelota localizada*. En dicho estado se calculan las potencias de los motores con el objetivo de acercarse a la pelota. Estos cálculos se realizan en base a dos factores: la distancia a la que se encuentra el robot de la pelota y el desvío del centro de gravedad de la pelota con respecto al centro de la imagen. El objetivo es, por tanto, mantener la pelota centrada en la imagen, disminuyendo la velocidad del robot según se vaya acercando.

El robot entra en el estado de *pelota cercana* cuando se encuentra lo suficientemente cerca de la pelota. En este estado se limita la potencia máxima del robot con el objetivo de aumentar el tiempo de reacción y maniobrar de forma más precisa.

Cuando el robot se encuentra frente a la pelota, éste pasa al estado *recogiendo pelota*. El brazo robótico realiza un movimiento prefijado para recoger la pelota. Una vez recogida la pelota, se vuelve a pasar al estado *detenido* y a la espera de una nueva orden para buscar de nuevo la pelota.

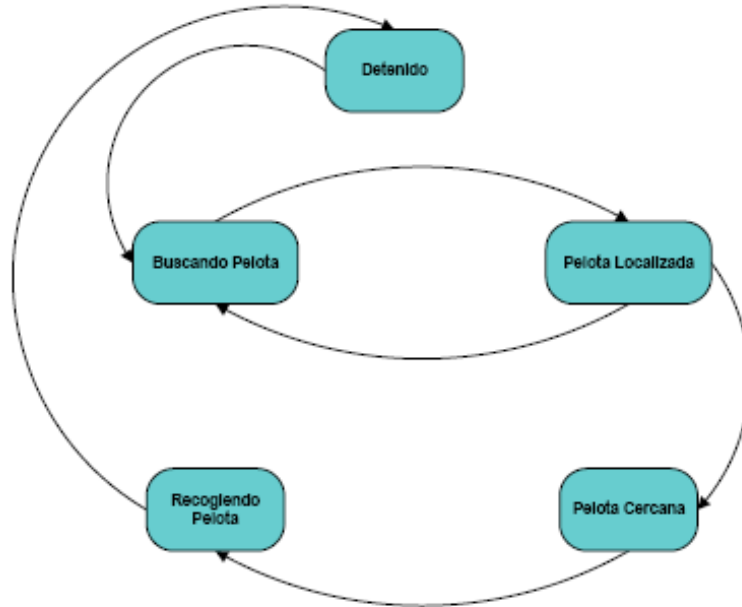


Figura 7.1: Diagrama de estados

Uno de los comportamientos del robot observados era que cuando en algún momento del acercamiento a la pelota si el robot perdía de vista la pelota éste se quedaba quieto. Esto sucede cuando alguna persona pasa entre el robot y la pelota cuando el robot se está acercando hacia la pelota.

Para evitar este comportamiento se ha empleado el filtro de Kalman[6] para estimar la posición de la pelota cuando existen oclusiones de la misma para que el robot no se quede quieto y avance hacia la pelota.

El filtro de Kalman es un conjunto de ecuaciones matemáticas que proveen una solución recursiva eficiente del método de mínimos cuadrados. Esta solución permite un estimador lineal, insesgado y óptimo del estado de un proceso en cada momento del tiempo con base en la información disponible en el momento $t-1$, y actualizar, con la información adicional en el momento t , dichas estimaciones. Este filtro es el principal algoritmo para estimar sistemas dinámicos especificados en la forma de estados-espacio.

En la figura 7.2 se presenta el diagrama de casos del filtro de Kalman y posteriormente se explica su funcionamiento:

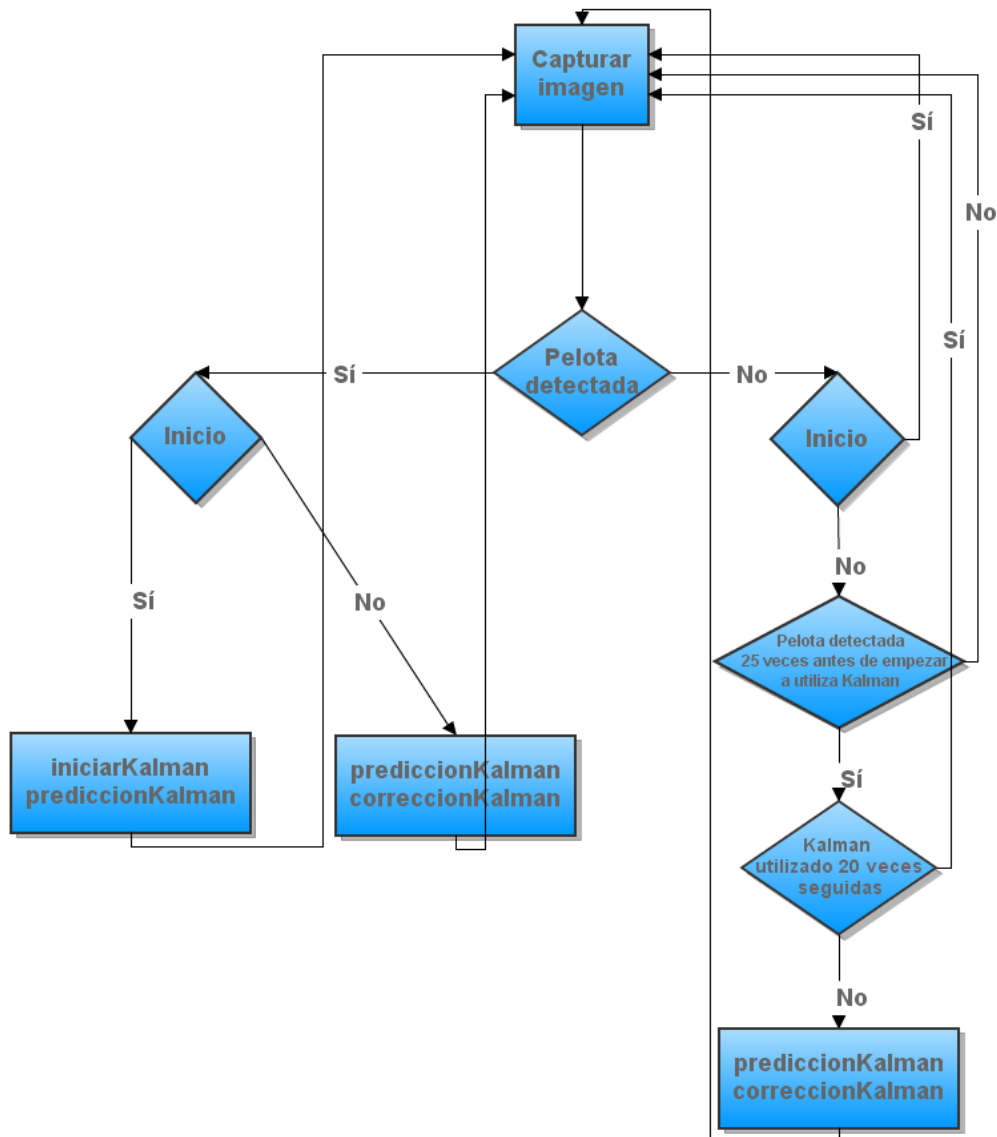


Figura 7.2: Diagrama de uso del filtro de Kalman

En primer lugar se captura la imagen de la cámara y se observa si se ha detectado la pelota o no. Si se ha detectado la pelota no es necesaria la utilización del filtro de Kalman para la ubicación de la misma pero hay que prepararlo para futuras imágenes.

En caso de que sea la primera imagen en la que se detecta la pelota habrá que iniciar el proceso de Kalman y calcular la primera predicción con las funciones *iniciarKalman* y *prediccionKalman*.

En cambio si no es la primera vez que se detecta la pelota, se procede a calcular la predicción del filtro de Kalman y a su posterior corrección mediante las funciones *prediccionKalman* y *correccionKalman*. En estos dos casos se estima la predicción de Kalman pero los datos de la posición de la pelota que se emplean son los que se obtienen con el algoritmo de reconocimiento.

Por otro lado se encuentra la situación en la que realmente se emplea la posición estimada por el filtro de Kalman como la verdadera posición de la pelota en la imagen, cuando no se detecta la pelota en ella. Pero para que se puedan utilizar estos datos se tienen que cumplir las siguientes exigencias:

1. No tiene que ser el inicio del proceso, es decir, se ha tenido que detectar la pelota con anterioridad.
2. Se ha tenido que detectar la pelota como mínimo un número determinado de veces, en este caso 25, antes de empezar a utilizar los datos obtenidos del filtro de Kalman.
3. No se ha sobrepasado el número máximo de veces que se puede utilizar el filtro de Kalman consecutivamente, en este caso 20 iteraciones.

Estas exigencias son necesarias porque el empleo del filtro de Kalman cuando no se cumplen, puede dar resultados incorrectos y no dar el posicionamiento correcto de la pelota en la imagen.

8. Experimentación y resultados

En esta sección se analizan los resultados más importantes obtenidos en las pruebas realizadas para comprobar la eficiencia del sistema. Para ello se han realizado pruebas para ver si el robot recoge las pelotas de distintos lugares en diferentes condiciones de iluminación y para cada modelo de color diferente. En la figura 8.1 y 8.2 se presentan las tablas de resultados obtenidos en la experimentación:

COLOR	LUMINOSIDAD	MODELO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Amarillo	Adecuada	RGB	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Amarillo	Adecuada	YUV	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Amarillo	Adecuada	HSL	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Azul	Adecuada	RGB	No	No	Sí	Sí	Sí	Sí	No	Sí	No	No	Sí	Sí	Sí	No	Sí
Azul	Adecuada	YUV	Sí	Sí	Sí	No	No	Sí	Sí	Sí	No	Sí	No	No	Sí	Sí	Sí
Azul	Adecuada	HSL	No	Sí	Sí	Sí	Sí	No	No	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí
Rojo	Adecuada	RGB	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Rojo	Adecuada	YUV	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Rojo	Adecuada	HSL	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí

Figura 8.1: Tabla de resultados de la experimentación con luminosidad adecuada

COLOR	LUMINOSIDAD	MODELO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Amarillo	Inadecuada	RGB	Sí	Sí	No	No	Sí	Sí	Sí	Sí	No	Sí	No	Sí	Sí	Sí	No
Amarillo	Inadecuada	YUV	Sí	Sí	Sí	Sí	Sí	No	No	Sí	No	Sí	Sí	Sí	Sí	No	Sí
Amarillo	Inadecuada	HSL	Sí	Sí	Sí	No	Sí	No	No	Sí	Sí	No	Sí	No	Sí	Sí	Sí
Azul	Inadecuada	RGB	No	No	No	Sí	No	No	No	Sí	No	No	No	No	No	Sí	No
Azul	Inadecuada	YUV	Sí	No	No	No	No	Sí	Sí	No	No	No	No	No	No	No	Sí
Azul	Inadecuada	HSL	No	No	No	Sí	Sí	No	Sí	No	No	No	Sí	Sí	Sí	No	Sí
Rojo	Inadecuada	RGB	Sí	No	Sí	No	No	Sí	Sí	Sí	Sí	Sí	No	No	Sí	No	Sí
Rojo	Inadecuada	YUV	No	No	Sí	No	No	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí
Rojo	Inadecuada	HSL	Sí	Sí	No	No	No	Sí	No	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí

Figura 8.2: Tabla de resultados de la experimentación con luminosidad inadecuada

Viendo los resultados obtenidos en la experimentación se pueden sacar conclusiones atendiendo a distintos aspectos como el modelo de color empleado, la iluminación en la que se ha realizado la prueba y el color de la pelota que se quiere recoger.

Centrándose en el modelo de color empleado para el filtro de color se han obtenido los resultados mostrados en la figura 8.3:

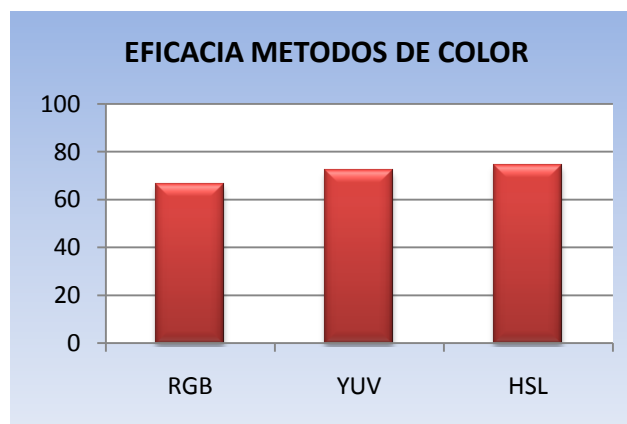


Figura 8.3: Eficacia de los distintos modelos de color

Como se puede observar en el grafico mostrado en la figura 8.3 la eficiencia obtenida con los filtros realizados con los distintos modelos de color es bastante parecida. Con el modelo RGB se obtiene una eficacia del 66.7%, con YUV el 72.2% y con HSL 74.4%.

Por otro lado se presentan los resultados dependiendo el color de la pelota que se ha querido recoger.

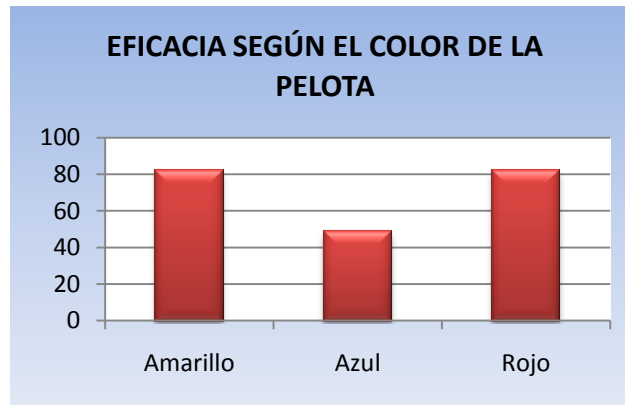


Figura 8.4: Eficacia según el color de la pelota

En este caso se puede ver que con el color amarillo (82.2%) y el rojo (82.2%) se consiguen mayores tasas de acierto que con el color azul (48.9). Esto se debe a que el color azul es más difícil de discriminar ya que el rango del color que se puede considerar como azul es más amplio. Por lo que existen colores que se consideran azules cuando no se deberían considerar. Sería posible limitar más el rango de color que se consideraría como azul pero en ocasiones descartaríamos colores que no interesaría descartar.

Por último en el gráfico que se puede ver en la figura 8.5 se presentan los resultados dependiendo de la luminosidad con la que se han realizado las pruebas.

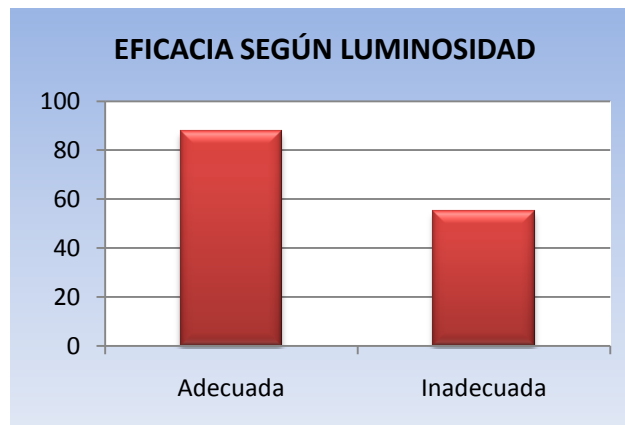


Figura 8.5: Eficacia según la luminosidad

Se observa que la efectividad del sistema baja considerablemente cuando la luz del entorno en la que se ha realizado la prueba se considera inadecuada. Con una luminosidad adecuada se consigue una eficacia del 87.4% y en cambio cuando la luminosidad es considerada inadecuada la eficiencia cae hasta el 54.8%. En general la efectividad del sistema es de un 71.1%.

Si en cambio se tienen en cuenta solo las pruebas realizadas con una iluminación adecuada, que es cuando realmente se empleará el robot recogepelotas, se consiguen los resultados mostrados en la figura 8.6.

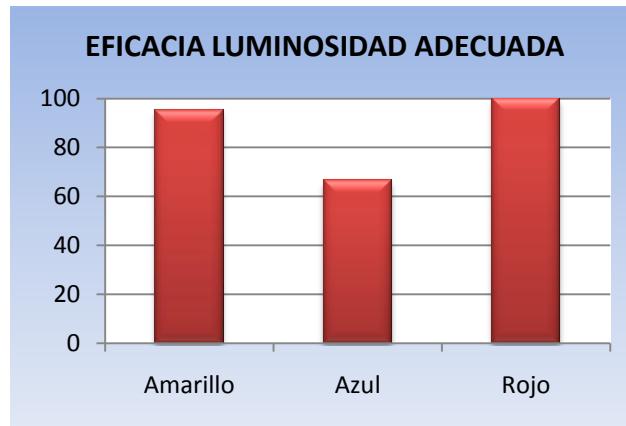


Figura 8.6: Eficacia luminosa adecuada

Como se puede observar en el gráfico presentado en la figura 8.6 las pelotas de color amarillo (95.5%) y rojo (100%) se detectan casi al perfección. Es el color azul (66.6%) el que hace que descienda la eficiencia de todo el sistema.

9. Conclusiones

El principal objetivo del trabajo ha sido desarrollar una función recogepelotas en la que el robot localiza y recoge una pelota que puede ser de varios colores. Para ello se han creado distintos filtros para determinados filtros de color y un reconocedor de formas. El filtro de color es el encargado de reconocer el color de la pelota que se quiere recoger en la imagen capturada de la cámara y posteriormente el reconocedor de formas es el responsable de reconocer la pelota en la imagen.

En la experimentación efectuada se ha conseguido una eficiencia del 71.1% aunque el sistema está pensado para su utilización con una iluminación adecuada en la que su efectividad aumenta hasta el 87.4%.

En general se dan por satisfechas las expectativas puestas en este proyecto puesto que se han conseguido los objetivos con una gran tasa de efectividad.

Referencias:

- [1] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8(6), pp. 679-698, 1986.
- [2] Deriche, R., 1990. Fast Algorithms for Low-Level Vision, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, pp. 78-87.
- [3] Hua R. C. K., De Silva L. C., and Vaddakkepat P. Detection and tracking of faces in realtime environments. *International Conf. on Imaging Science, Systems and Technology*, 2002.
- [4] Finlayson G. D. (2000) *Computational color Constancy*, in International Conference on Pattern Recognition, vol. 1, Barcelona, Spain, pp. 191-196.
- [5] Kalman, R. E. 1960. "A New Approach to Linear Filtering and Prediction Problems," Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960).
- [6] Kravtchenko V. (1999) *Tracking color objects in real time*, Master's thesis, University of British Columbia, Vancouver, British Columbia.
- [7] G. S. Pingali, Y. Jean, and I. Carlbom. Real time tracking for enhanced tennis broadcasts. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 1998.
- [8] G. S. Pingali, A. Opalach, and Y. Jean. Ball tracking and virtual replays for innovative tennis broadcasts. In *IEEE International Conference on Pattern Recognition*, 2000.
- [9] J. M. S. Prewitt. Object enhancement and extraction. In A. Rosenfeld and B. S. Lipkin, editors, *Picture Processing and Psychophysics*, pages 75-149. Academic Press, New York, 1970.
- [10] J. Ren, J. Orwell, G. A. Jones and M. Xu, "A novel framework for 3D soccer ball estimation and tracking", *Proc. ICIP*, 2004.
- [11] Y. Seo, S. Choi, H. Kim, K. S. Hong, "Where are the ball and players?: soccer game analysis with color based tracking and image mosaick", *Proc. ICIAP*, 196-203, 1997.
- [12] Shen, J. Castan, S. An Optimal Linear Operator for Step Edge Detection. *Computer Vision, Graphics, and Image Processing: Graphical Models and Understanding*. Vol. 54, No.2. pp. 112-133. 1992.
- [13] V. Vezhnevets, V. Sazonov, A. Andreeva. A Survey on Pixel-Based Skin Color Detection Techniques. *GraphiCon Conference* 2003.

- [14] Yan, F., Christmas, W., Kittler, J.: Layered Data Association Using Graph-Theoretic Formulation with Application to Tennis Ball Tracking in Monocular Sequences. *IEEE Trans. on Pattern Anal. and Machine Intel.* 30(10) (2008).
- [15] Yu, X. and Farin, D. (2005). Current and emerging topics in sports video processing. In *IEEE International Conf. on Multimedia and Expo (ICME)*.