Universidad Autónoma del Carmen

FACULTAD DE INGENIERÍA Y TECNOLOGÍA

"SISTEMA DE LOCALIZACIÓN USANDO UN FILTRO DE PARTÍCULAS PARA UN ROBOT HUMANOIDE EN UN CAMPO DE FÚTBOL."

TESIS

PARA OBTENER EL TíTULO DE:

Maestria en Ingeniería Mecatronica

PRESENTA:

Daniel Silva Medina

Cd. Del Carmen, Campeche, 2018

Universidad Autónoma del Carmen

FACULTAD DE INGENIERÍA Y TECNOLOGÍA

"SISTEMA DE LOCALIZACIÓN USANDO UN FILTRO DE PARTÍCULAS PARA UN ROBOT HUMANOIDE EN UN CAMPO DE FÚTBOL."

TESIS

PARA OBTENER EL TíTULO DE:

Maestria en Ingeniería Mecatronica

PRESENTA:

Ing. Daniel Silva Medina

DIRECTOR:

Dr. José Luis Vazquez Avila.

CO-DIRECTOR:

Dr. Alberto Elías Petrilli Barceló

Cd. Del Carmen, Campeche, 2018

Índice general

Lista de figuras **7**

Lista de tablas **9**

1. Aspectos generales **10**

1.1. Introducción. 10

1.2. Antecedentes 12

1.3. Definición del problema. 13

1.4. Objetivos. 14

1.4.1. Objetivo general. 14

1.4.2. Objetivos particulares. 14

1.5. Propuesta. 14

1.6. Metodología. 17

2. Marco teórico. **19**

2.1. Pre procesamiento de imágenes. 19

2.1.1. Obtención de imágenes. 19

2.1.2. Espacio de color RGB. 20

2.1.3. Espacio de color HSV. 20
2.1.4. Cromaticidad. 23
2.2. Búsqueda de características. 24
2.2.1. Transformada de Hough. 24
2.2.2. RANSAC 27
2.3. Perspectiva 28
2.3.1. Modelo Pinhole. 28
2.3.2. Linea al horizonte. 30
2.3.3. Homografias. 31
2.4. Filtro de Partículas. 32

3. Descripción de la metodología. 35

3.1. Obtención de la imagen. 35
3.2. Segmentación. 36
3.2.1. Segmentación HSV. 37
3.2.2. Segmentación por Cromaticidad. 37
3.3. Cámara. 38

4. Resultados 40

4.1. Procesamiento de la imagen. 41
4.1.1. Segmentación. 41
4.1.2. Rectificación de la imagen. 48
4.1.3. Búsqueda de lineas y circulo. 50
4.2. Conclusiones 51

4.3. Trabajo a futuro. 53

Índice de figuras

1.1. Secuencia de imágenes representativas de la aplicación usando el filtro de canny y la transformada de hough. 16

2.1. Imagen del cubo RGB mostrando sus componentes en el plano cartesiano. . . 21

2.2. Diagrama de HSV fuente: [6]. 21

2.3. Diagrama de cromaticidad fuente: [6]. 24

2.4. Imagen demostrativo de una linea en el espacio cartesiano (a) y el espacio de Hough (b). 26

2.5. Modelo geométrico de la cámara pinhole. **C** es el centro de la cámara y **p** es el punto principal fuente: [6]. 28

2.6. Intersección de los planos P y H fuente:[8]. 31

3.1. Infraestructura de naoqi. 36

3.2. Tablero de ajedrez para la obtención de puntos en la imagen. 38

3.3. Ejemplo de las dimensiones del tablero para la función de OpenCv. 39

4.1. Diagrama de flujo del sistema completo. 40

4.2. a) imagen original en resolución 320x240 px, b) imagen segmentada por el método de hsv. 41

4.3. a) imagen original en resolución 640x480 px, b) imagen segmentada por el método de hsv. 41

4.4. a) imagen original en resolución 1280x960 px, b) imagen segmentada por el método de hsv. 42

4.5. a) imagen original en resolución 320x240 px, b) imagen segmentada por el método de cromaticidad. 42

4.6. a) imagen original en resolución 640x480 px, b) imagen segmentada por el método de cromaticidad. 42

4.7. a) imagen original en resolución 1280x960 px, b) imagen segmentada por el método de cromaticidad. 43

4.8. Diseño de la interfaz utilizada para calibrar el color del entorno a procesar. . . 45

4.9. a), c), e) Muestras tomadas a la luz exterior, b), d), f) Resultados de la segmentación con umbralamiento. 46

4.10. Calculo de la linea al horizonte representada con una linea azul. 47

4.11. Clasificación del color verde mediante el algoritmo de scan lines. 47

4.12. Clasificación del color verde mediante el algoritmo de scan lines. 48

4.13. Imagen rectificada con la técnica IPM. 49

4.14. Rectificación a la nube de puntos obtenida con las técnicas de segmentación. . 49

4.15. Resultado de la palicacion del algoritmo de búsqueda de lineas, mostrando de color rojo las lineas encontradas dentro del campo de fútbol. 50

4.16. Muestra de la eficiencia del algoritmo de RANSAC. 51

4.17. Percepción del robot del entorno. 51

4.18. Suma de las imágenes para rectificar que es el circulo central del campo. . . . 52

Índice de tablas

4.1. Comparación de algoritmos de segmentación resolución 320x240 43

4.2. Comparación de algoritmos de segmentación resolución 640x480 43

4.3. Comparación de algoritmos de segmentación resolución 1280x960 44

Capítulo 1

Aspectos generales

1.1. Introducción.

Con los avances tecnológicos de los últimos años, la comunidad científica se ha propuesto una serie de retos para medir el avance tecnológico y científico con el pasar de los años, debido a ello se propuso un reto llamado: ,”El gran reto", el cual habla de que en el año 2050 se forme un equipo de robots humanoides que sea capaz de ganarle a la selección de fútbol que sea campeona mundial de fútbol con humanos de ese mismo año.

Debido a el gran reto, nació un proyecto internacional con el nombre de ”RoboCup", el cual fue fundado en el año 1997 que tiene propósito de promover la investigación y educación sobre inteligencia artificial a través de competencias integradas por robots autónomos.

La competencia RoboCup se conforma de diferentes categorías, una de ellas es la de RoboCup Soccer la cual consta de jugar fútbol con un equipo de robots humanoides, es decir con una morfología parecida a los humanos, como sabemos el fútbol es uno de los deportes donde el ambiente es muy dinámico, ya que los jugadores del mismo equipo y del equipo contrario siempre están en constante movimiento, esto no contando con la posición del balón.

Debido a que el fútbol es un ambiente muy dinámico se presenta con un gran problema, la localización la cual no solo es la localización del robot en el campo de fútbol, si no que de igual forma la de los compañeros. El problema de localización se puede dividir en dos fases, localización local y localización global.

La localización local consta de que se supone conocida la posición actual del robot y el objetivo, y esta misma consiste en mantener dicho conocimiento según se vaya desplazando el robot y acumule los diversos errores que los sensores tienen al realizar la medición.

Por otro lado, la localización global consiste que partiendo de un desconocimiento absoluto de la posición inicial en la que se encuentre el robot, este problema se torna un poco más complicado debido a que la única información con la que se cuenta es proveniente de sensores que presentan ruido y en su modelo, linealidades, de igual manera se puede contar con un mapa pre guardado por el robot[9].

Otra problemática que se puede presentar es que, si nuestro entorno muestra una simetría desde el punto de vista de los sensores montados en nuestro sistema robótico, puede llegar a ser imposible culminar con certeza la localización global.

En la localización siempre se supone que se conoce el mapa o el entorno donde opera el robot, este puede ser conocido con cierto grado de incertidumbre. El mapa puede ser descrito de muchas maneras, pero las más usadas es mediante marcas (“landmarks”). En este caso el robot debe ser capaz de reconocer dichas marcas visuales para poder tener una perspectiva del mapa (mediante métodos de visión artificial, por ejemplo.).

La localización también puede ser vista como el proceso de establecer una correspondencia entre un sistema de coordenadas global y un sistema de coordenadas local del robot. Al conocer esta transformación el robot tiene la capacidad de ubicar objetos respecto a su sistema de coordenadas(local), lo cual es muy importante en la resolución de problemas de navegación y evasión de obstáculos. Dicho lo anterior podemos deducir que el problema de localización

se resume a encontrar los parámetros de dicha transformación.

El filtro de partículas es un método comúnmente usado para el estudio de un sistema que cambia a lo largo del tiempo, más concretamente es un método de Montecarlo usado comúnmente en visión artificial para el seguimiento de objetos mediante una secuencia de imágenes.

1.2. Antecedentes

El problema de localización en robótica ha sido constantemente abordado por la comunidad científica. Además, se cuenta con una numerosa cantidad de artículos científicos dirigidos a este tema acerca de los problemas y sus sub-problemas. La localización es una tarea inherente de la navegación de robots, incluidos los humanoides. Si solo ordenáramos tareas a los robots tales como trayectorias y ejecución de movimientos, estos no serían capaces de desempeñar la tarea que se les asigno; esto debido a distintas razones como el deslizamiento de las extremidades del robot con el suelo. Hacer la ejecución de trayectorias en lazo cerrado es la solución y para ello se requiere la localización.

Existen una gran cantidad de maneras de realizar la tarea de localización para los robots humanoides. Y cada una de estas conlleva una buena cantidad de variantes. Mayormente para esta tarea se emplean técnicas probabilísticas, tales como el Monte Carlo (MCL) en nuestro caso o filtros de Kalman en todas sus variantes. La similitud de estos métodos es que integran la información de los sensores internos del robot (por ejemplo, giroscopios, acelerómetros, etc.) con la información de sensores exteroceptivos (cámaras, láser, sonares, etc.).

Generalmente la pose del robot es información que se desconoce, y la única solución al problema es obtenerla indirectamente, es decir a través de la integración de datos de los sensores a través del tiempo de los diversos sensores que posee el sistema.

1.3. Definición del problema.

Para la realización de este proyecto se presentan una serie de problemas, en primer instancia es la extracción de las características del mapa, es decir usar al algoritmo que sea capaz de detectar las “landmarks”, como en esta ocasión el ambiente de trabajo será un campo de fútbol se pretende buscar las marcas que delimitan el campo para que de esa forma se obtenga un mapa construido por el robot.

El siguiente problema con el que se encuentra es que nuestra campo de fútbol es una figura geométrica con forma simétrica, eso es una gran problemática debido a que necesitaremos un buen algoritmo que nos ayude a integrar las medición internas y las externas de nuestro robot para poder conocer la pose.

1.4. Objetivos.

1.4.1. Objetivo general.

EL objetivo principal es crear un sistema de localización para un robot humanoide en un entorno conocido, mediante información visual y utilizando un filtro de partículas.

1.4.2. Objetivos particulares.

Para la realización de este proyecto se encuentran varias problemáticas y se tiene un objetivo para cada uno de ellas.

- Elegir un algoritmo que sea eficaz para la detección de características, en este caso particular para las líneas que conforman el campo de fútbol.
- Realizar la calibración de las camaras del robot.
- Calcular la línea al horizonte.
- Realizar pruebas para medir la eficacia del filtro de partículas y realizar comparaciones con otros filtros por trabajos ya hechos.

1.5. Propuesta.

Para la realización de este proyecto se propone implementar el filtro de partículas, obteniendo información dada por una cámara que va montada en el robot humanoide, esta información será procesada mediante técnicas de procesamiento digital de imágenes, de la cual obtendremos datos de “landmarks” para así poder fusionar esos datos con los sensores propioceptivos del robot humanoide.

Se pretende realizar un programa en C++, el cual se encargara de la obtención y procesamiento de los diferentes sensores y de asociar esos datos, de igual manera hará los cálculos necesarios para poder obtener la información de las coordenadas del robot y la pose en el marco de referencia local; para la recopilación de datos con la cámara se utilizara la biblioteca de código abierto OpenCV (Open Source Computer Vision Library), se eligió esta librería ya que es de código abierto y se puede manipular de la misma manera que esta misma cuenta con cientos de algoritmos de visión por computadora.

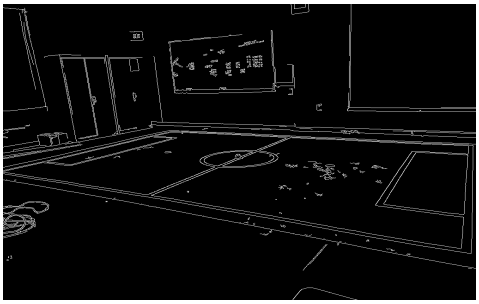
Uno de los algoritmos que se proponen para pre-procesar la imagen es “detector de bordes de canny”, el cual nos ayudara a eliminar la información que no necesitamos en la imagen, para poder marcar esos bordes dados se propone usar la transformada de hough, cabe mencionar que estos dos algoritmos solo son propuestas ya que aún pasaran por un proceso de evaluación y pruebas, posteriormente se elegirá el que mejor se adapte a nuestra aplicación ya que es de gran importancia que sean eficaces ya que la aplicación estará funcionando en línea. En las siguientes imágenes se muestra una muestra de cómo funcionan los filtros, cabe mencionar que fueron aplicados a una imagen estática.

(a) Imagen original.

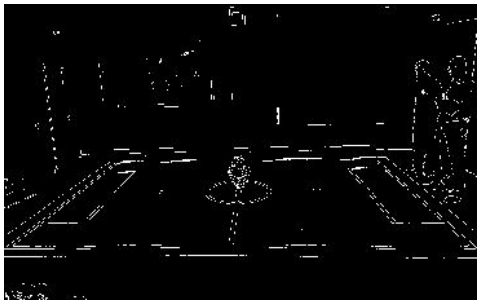
(b) Imagen original.

(c) Imagen en escala de grises.

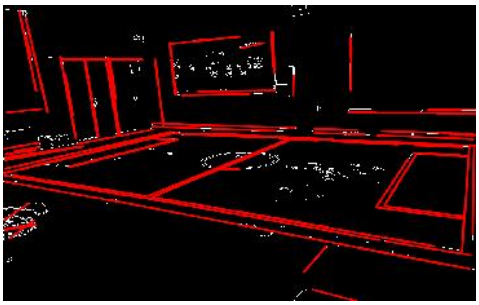
(d) Imagen en escala de grises.



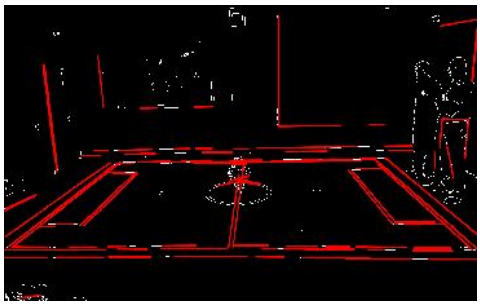
(e) Imagen con el filtro de canny.



(f) Imagen con el filtro de canny.



(g) Imagen con la trasformada de Hough.



(h) Imagen con la trasformada de Hough.

Figura 1.1: Secuencia de imágenes representativas de la aplicación usando el filtro de canny y la transformada de hough.

1.6. Metodología.

En primer lugar, se necesita tener un mapa del entorno en el cual se va a navegar, para esto se buscarán algoritmos de visión por computadora ya que usaremos la cámara como sensor principal para la localización, con ello se extraen las características del mismo mediante técnicas de visión por computadora. Una vez obtenidas las características se procederá a aplicar el filtro de partículas, el cual se compone de cuatro etapas principales:

1. Inicialización.
2. Predicción.
3. Estimación.
4. Actualización.

El proceso consiste en "lanzar" al azar un conjunto de puntos sobre la imagen con estado aleatoria las cuales serán nuestras partículas, esto entraría en la etapa de inicialización, posteriormente mediante cálculos se les asignarán valores o pesos a las partículas este sería el caso de la etapa de actualización. A partir de esos puntos con sus nuevos valores van a reemplazar a los anteriores, esta elección también será al azar, pero con un cálculo previo que provocará que sea más probable que estos puntos se sitúen en el lugar donde se encuentre nuestro robot, esta parte seria la parte de estimación. Una vez que se crea el nuevo conjunto de puntos, se realiza una leve modificación al estado (posición) de cada uno de ellos, con el fin de estimar el estado del objeto en el instante siguiente, esta sería la etapa de predicción. Al final de la etapa de predicción se vuelve a aplicar la etapa de actualización, repitiéndose este bucle.

En el capitulo 2 se describen las técnicas usadas para la extracción de las características usadas a lo largo de este trabajo. En el capitulo 3 se detalla de una manera mas precisa la aplicación de las técnicas para lograr el objetivo. En el capitulo 4 se muestran los resultados

obtenidos del trabajo. En el capitulo 5 se redacta una breve conclusión y los trabajos a futuro que se proponen a partir de este trabajo.

Capítulo 2

Marco teórico.

En este capitulo se detalla a fondo las técnicas usadas desde el pre procesamiento de las imágenes hasta la aplicación del filtro de partículas, el cual lo veremos en tres secciones las cuales son: Obtencion de imagenes, segmentación y la busqueda de características del entorno.

2.1. Pre procesamiento de imágenes.

En esta sección vamos a describir el proceso que realizamos para la obtención de las imágenes hasta llegar a la extracción de las lineas dentro de la imagen, así como los métodos y teoría utilizada para la realización de este proyecto.

2.1.1. Obtención de imágenes.

Para la obtención de las imágenes de la cancha se utiliza una cámara modelo MT9M114 con una resolución máxima de 1280x960 píxeles, la cual se describirá a fondo en el apartado de herramientas del sistema.

Para la obtención de las imágenes se utilizo el framework que trae el robot por defecto, el cual se llama naoqi, con este mismo se pueden crear un modulo para obtener una conexión hacia el robot por medio de una conexión de red, permitiéndonos así obtener los datos captados por la cámara. La cámara por defecto entrega los datos en un espacio de color RGB, pero en nuestro caso usaremos una técnica de segmentación en el espacio de color HSV. dichos espacios y técnica de segmentación se explicaran en las siguientes secciones.

2.1.2. Espacio de color RGB.

El espacio del color RGB es el modelo más conocido y utilizado. En el modelo RGB cada color está representado por tres valores de rojo (R), verde (G) y azul (B), situados a lo largo de los ejes del sistema de coordenadas cartesianas sobre un cubo. Los valores de RGB van en un rango de [0,1] o de [0, 255]. De esta manera el negro se representa como (0, 0, 0), el blanco como (1, 1, 1) o (255, 255, 255). El color negro y el blanco están representados por dos esquinas opuestas del cubo que se pueden definir por los ejes de R, G, B, del sistema de coordenadas cartesianas. Las otras puntas del cubo representan el rojo, el verde, el azul, el cían, el magenta y el amarillo. Los colores en escala de grises se representan con componentes idénticas de R, G, B.

2.1.3. Espacio de color HSV.

Por sus siglas en ingles HSV (Hue, Saturation, Value - Matiz, Saturación, Valor), también HSB (Hue, Saturation, Brightness - Matiz, Saturación, Brillo).

- Matiz (Hue): Se refiere a la frecuencia dominante del color dentro del espectro visible. Es la percepción de un color, normalmente a lo que uno distingue en un arcoiris, es

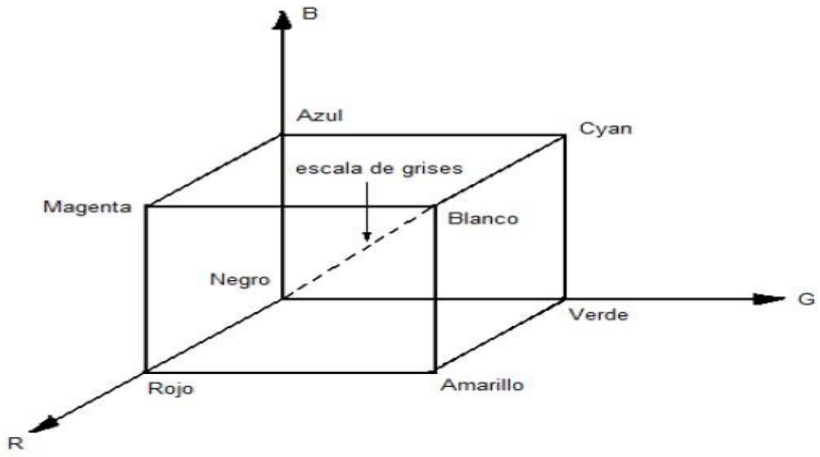


Figura 2.1: Imagen del cubo RGB mostrando sus componentes en el plano cartesiano.

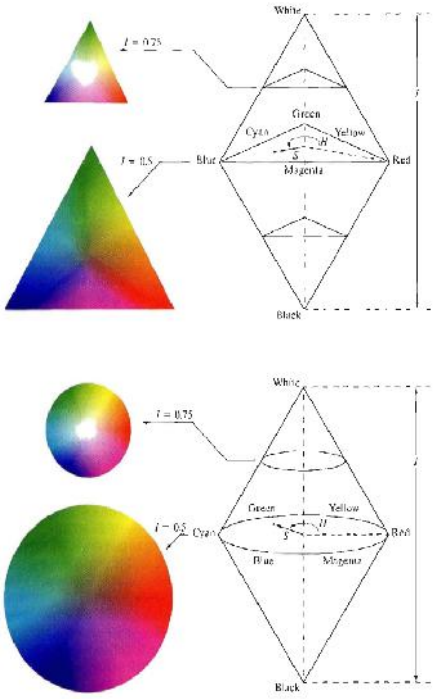


Figura 2.2: Diagrama de HSV fuente: [6].

decir, es la sensación humana de acuerdo a la cual un área perece similar a otra o cuando existe un tipo de longitud de onda dominante. Incrementa su valor mientras nos movemos de forma antihorario en el cono, con el rojo en ángulo 0.

- Saturación (Saturation): Se refiere a la cantidad de color o a la "pureza"de éste. Va de un color çlaro.[a] un color más vivo (azul cielo - azul oscuro). También se puede considerar como la mezcla de un color blanco con gris.

- Valor(Value): Es la cantidad de luz en un color, en otras palabras es la cantidad de blanco o de negro que posee un color.

Matiz y saturación juntos son llamados cromaticidad, y sin embargo un color puede ser caracterizado por su brillo y su cromaticidad, los valores de rojo, verde y azul son necesarios para formar color llamado tristimulus, y sus valores son denotados X, Y, Z respectivamente. Un color es especificado por los coeficientes tricromaticos, definidos como:

$$x = \frac{X}{X+Y+Z} \tag{2.1}$$

$$y = \frac{Y}{X+Y+Z} \tag{2.2}$$

$$z = \frac{Z}{X+Y+Z} \tag{2.3}$$

Nótese que de las ecuaciones se deduce que:

$$x+y+z = 1 \tag{2.4}$$

Para moverse del espacio de color RGB a HSV se utilizan las siguientes formulas:

Para matiz:

$$H = \begin{cases} \theta & if B \leqslant G \\ 360 - \theta & if B > G \end{cases} \tag{2.5}$$

con:

$$\theta = cos^{-1}\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}}\}$$

Para la Saturación:

$$S = 1 - \frac{3}{(R+G+B)}[min(R,G,B)]. \tag{2.6}$$

y finalmente para la intensidad:

$$I = \frac{R+G+B}{3} \tag{2.7}$$

2.1.4. Cromaticidad.

Otra manera de especificar el color es CIE cromaticidad ver fig:2.3, la cual muestra la composicion del color como una funcion de x(rojo), y (verde). para cualquier valor de x y y el valor correspondiente de z es obtenido por la ecuación 2.4 denotando que $z = 1 - (x+y)$.

El diagrama de cromaticidad es usado para la combinación de colores, debido a que un segmento de linea puede unir dos puntos definidos en todas las variaciones de colores diferentes que pueden ser obtenidos de la combinación de esos dos colores. En palabras resumidas la cromaticidad es la pureza que existe de un color en la composición de un color en el espacio RGB y para calcular la cromaticidad de un color se utiliza la siguiente ecuacion:

$$cr = \frac{G}{(R+G+B)} \tag{2.8}$$

En este caso G es el valor del pixel verde, R del color rojo, y B del color azul, para calcular la cromaticidad de algun otro color solo de cambia el denominador por la cromatricidad que quieras calcular.

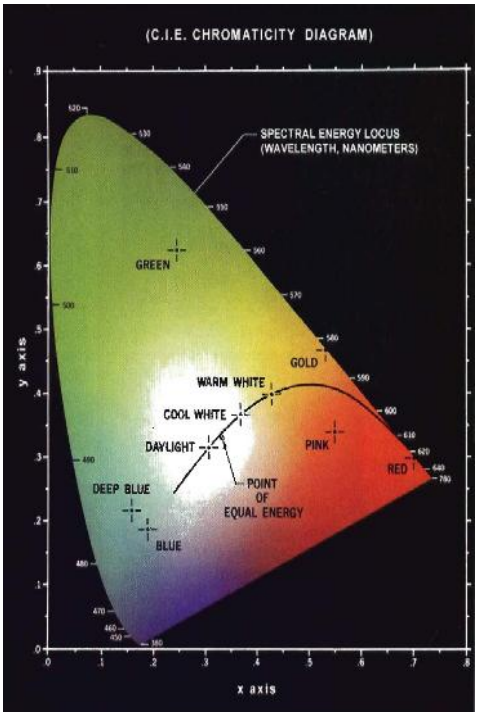


Figura 2.3: Diagrama de cromaticidad fuente: [6].

2.2. Búsqueda de características.

2.2.1. Transformada de Hough.

La transformada de Hough fue propuesta y patentada en 1962, por Paul Hough,1 inicialmente esta técnica solo se aplicaba a la detección de rectas en una imagen. En el análisis automatizado de imágenes, es común encontrar el problema de detectar figuras simples, tales como rectas o circunferencias. Como primer paso, se puede usar un detector de bordes para obtener los puntos de la imagen que pertenecen a la frontera de la figura deseada. Debido

a las imperfecciones, ya sea de la imagen captada o del detector de bordes, existen muchos puntos que pertenecen a la curva y que faltan en la imagen; también pueden existir desviaciones espaciales entre la figura ideal (por ejemplo, una recta) y los puntos ruidosos del borde detectado. Por estas razones, usualmente no es trivial agrupar los bordes detectados en un conjunto apropiado de figuras, ya sean circunferencias o cualquier otra figura. El objetivo de la transformada de Hough es resolver este problema, haciendo posible realizar agrupaciones de los puntos que pertenecen a los bordes de posibles figuras a través de un procedimiento de votación sobre un conjunto de figuras parametrizadas. El caso más simple para la transformada de Hough es la transformación lineal para detectar líneas rectas. En el espacio de la imagen, la recta se puede representar con la ecuación, :

$$y = mx + b \tag{2.9}$$

donde y es la intersección con el eje y, m la pendiente y b la ordenada al origen, y se puede graficar para cada par (x,y) de la imagen. En la transformada de Hough, la idea principal es considerar las características de una recta en término de sus parámetros (*m*,*n*), y no como puntos de la imagen $(x1,y1)$, . . . , (xn,yn). Basándose en lo anterior, la recta se puede representar como un punto (m,n) en el espacio de parámetros. Sin embargo, cuando se tienen rectas verticales, los parámetros de la recta (m,n) se indefinen. Por esta razón es mejor usar los parámetros que describen una recta en coordenada polares, denotados (ρ,θ).

El parámetro ρ representa la distancia entre el origen de coordenadas y el punto (x,y), mientras θ que es el ángulo del vector director de la recta perpendicular a la recta original y que pasa por el origen de coordenadas. Usando esta parametrización, la ecuación de una recta se puede escribir de la siguiente forma:

$$y = -\frac{cos\theta}{\operatorname{sen}\theta} * x + \frac{\rho}{sen\theta} \tag{2.10}$$

y que se puede reescribir de la siguiente manera:

$$\rho = xcos\theta + ysen\theta \tag{2.11}$$

Entonces, es posible asociar a cada recta un par (ρ, θ) que es único si $\theta \varepsilon [0, \pi)$ y $\rho \varepsilon R$ o $\theta \varepsilon [0, 2\pi)$ y $\rho \geq 0$. El espacio (ρ, θ) se denomina espacio de Hough para el conjunto de rectas en dos dimensiones. Para un punto arbitrario en la imagen con coordenadas (x_0, y_0) , las rectas que pasan por ese punto son los pares (ρ, θ) con 2.11 donde ρ (la distancia entre la línea y el origen) está determinado por θ . Esto corresponde a una curva senosoidal en el espacio (ρ, θ), que es única para ese punto. Si las curvas correspondientes a dos puntos se intersecan, el punto de intersección en el espacio de Hough corresponde a una línea en el espacio de la imagen que pasa por estos dos puntos. Generalizando, un conjunto de puntos que forman una recta, producirán senosoides que se intersecan en los parámetros de esa línea.

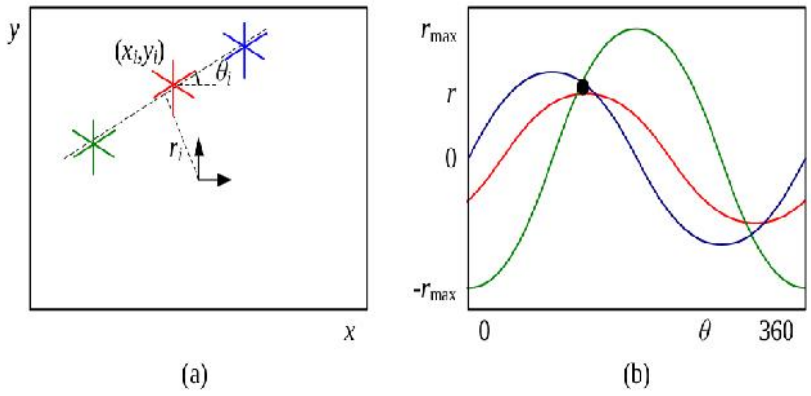


Figura 2.4: Imagen demostrativo de una linea en el espacio cartesiano (a) y el espacio de Hough (b).

2.2.2. RANSAC

Random sample consensus (RANSAC) es un método iterativo para calcular los parámetros de un modelo matemático de un conjunto de datos observados que contiene valores atípicos. Es un algoritmo no determinista en el sentido de que produce un resultado razonable sólo con una cierta probabilidad, mayor a medida que se permiten más iteraciones. El algoritmo fue publicado por primera vez por Fischler y Bolles SRI International en 1981.

Los datos consisten en ïnliers", es decir, los datos cuya distribución se explica por un conjunto de parámetros del modelo, aunque pueden estar sujetos a ruido, y "valores atípicos", que son datos que no encajan en el modelo. Los valores atípicos pueden provenir, por ejemplo, de valores extremos del ruido o de mediciones erróneas o hipótesis incorrectas sobre la interpretación de los datos. RANSAC también asume que, dada un conjunto de inliers (generalmente pequeño), existe un procedimiento que puede estimar los parámetros de un modelo que explica de manera óptima o se ajusta a esta información.

El algoritmo consta de los siguientes pasos:

- Seleccionar 3 puntos al azar.
- Se calcula el circulo que pasa por esos tres puntos.
- se calculo el conjunto consenso, es decir que los puntos seleccionados se ajusten a una circunferencia con una tolerancia dad.
- si el numero de votos para ese conjunto es mas grande que un umbral, se guarda el modelo propuesto.

2.3. Perspectiva

En las siguientes secciones, se describirá un poco de teoría sobre la geometría proyectiva aplicada a las cámaras, para realizar la rectificación de las imágenes, y así obtener nuevamente propiedades perdidas debido a la deformación proyectiva que existe en las cámaras.

2.3.1. Modelo Pinhole.

El modelo de cámara pinhole es uno de los modelos más utilizados, en este modelo se considera la proyección central de puntos en el espacio sobre un plano, dado el centro de proyección se toma como el origen de un sistema coordenadas cartesiano, este punto sera la apertura de la cámara el cual de ahi toma su nombre pinhole o su traducción al español "hoyo de aguja".

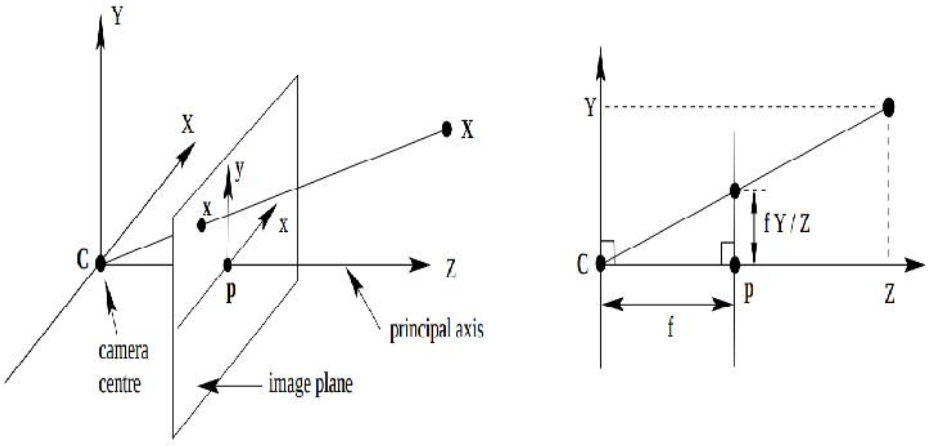


Figura 2.5: Modelo geométrico de la cámara pinhole. **C** es el centro de la cámara y **p** es el punto principal fuente: [6].

Consideramos el plano $Z = f$, el cual es llamado el plano de la imagen o plano focal sobre el modelo de cámara, un punto en el espacio con coordenada $X = (X,Y,Z)^T$ es mapeado a el punto de el plano de la imagen.

Calculamos que el punto $(X,Y,Z)^T$ es mapeado a el punto $(fX/Z, fY/Z, f)$ sobre el plano de la imagen, ignorando la coordenada final vemos que se describe la proyección central mapeando de el mundo real a coordenadas de la imagen.

$$(X,Y,Z)^T \mapsto (fX/Z, fY/Z)^T \tag{2.12}$$

Este es un mapeo de espacio euclidiano $\mathbb{R}^3$ a un espacio euclidiano $\mathbb{R}^2$.

EL centro de proyección es llamado el centro de la cámara, o también conocido como el centro óptico. La linea perpendicular al centro de la cámara a el plano de la imagen es llamada el eje principal o rayo principal de la cámara, y el punto donde el eje principal intersecta a el plano de la imagen es llamado punto principal.

Si los puntos de el mundo son representados por vectores homogeneos, entonces la proyección central se puede representar como un mapeo lineal entre sus coordenadas homogeneas, en particular 2.12 puede ser escrito en términos matriciales como:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.13}$$

Si usamos a X como notación para describir el vector que contiene las coordenadas del mundo real 2.13 se puede reescribir como:

$$x = PX \tag{2.14}$$

Donde la matriz de proyección de la cámara en modelo pinhole queda:

$$P = diag(f, f, 1)[I|0] \tag{2.15}$$

La expresión 2.12 asume que el origen de las coordenadas de el plano de la imagen es el punto principal. En la practica esto no es posible, entonces el mapeo queda de la siguiente manera:

$$(X, Y, Z)^T \mapsto (fX/Z + p_x, fY/Z + p_y)^T \tag{2.16}$$

2.3.2. Linea al horizonte.

La linea al horizonte es un concepto que se maneja en el arte, principalmente en las pinturas, ya que esta es la linea donde el pintor puede hacer referencia a la zona más importante de la pintura, como por ejemplo donde el artista quiera resaltar el color de la luna, o algunos animales que anden caminando en la tierra, para ello tendría que ir atenuando colores desde su linea al horizonte hacia la región de interés.

La linea al horizonte no esta relegada a escenas exteriores, para los interiores, el termino "nivel de ojos"se usa generalmente y tiene el mismo propósito de darle al artista el control de donde se enfoca el espectador.

En este caso nosotros lo que queremos enfocarnos en la cancha de fútbol, la cual por lógica sabemos que se encuentra sobre el piso, entonces nuestra linea al horizonte sera la división entre el cielo y la tierra.

En nuestro caso la linea al horizonte se debe calcular con la matriz de rotación de la cámara un método fue propuesto en [8], en el cual consta de buscar la intersección de dos planos. El plano P de la cámara y un plano paralelo al piso H que pase por el centro c de la cámara.

Para calcular la linea al horizonte $\overrightarrow{b}_{l/r}$ de izquierda a derecha en los bordes de la imagen y

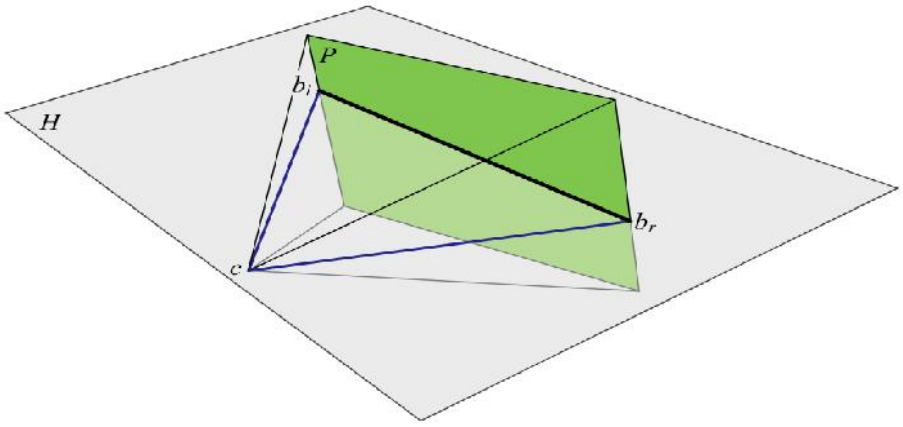


Figura 2.6: Intersección de los planos P y H fuente:[8].

los puntos $\overrightarrow{h}$ sobre el plano H ver fig.2.6:

2.3.3. Homografias.

Las homografias en el ámbito de la geometría son transformaciones proyectivas ya sea entre planos o figuras geometricas, el fin de las homografias es que exista una correspondencia entre las figuras contenidas en los planos.

En este trabajo se presenta una transformación proyectiva entre dos planos, dicha transformación de realiza para obtener una relación directa entre los dos planos que están interactuando (plano de el mundo 3D y plano de la imagen), ya que por naturaleza la obtención de los datos de la cámara forma un plano de coordenadas (x,y) y estamos mapeando al mundo real el cual sus coordenadas son (x,y,z), es lo que principalmente causa una deformación proyectiva en nuestra obtención de datos, causa un gran problema ya que dichas deformación hace que se pierdan varias propiedades de figuras geométricas, paralelismo entre rectas y deformación de ángulos.

Dicha transformación de representa con la siguiente expresión:

$$\mathbf{U = K*R*V} \tag{2.17}$$

Donde **K** es la matriz de parámetros intrínsecos de la cámara, multiplicada por la matriz de parámetros extrínsecos:

$$\mathbf{K = P * Mext} \tag{2.18}$$

En la cual M_{ext} es la matriz que modela el movimiento de la rotación de la cabeza del robot, la cual contiene la cámara, esta es indispensable para conservar la perspectiva obtenida aun cuando hayamos realizado algún movimiento de dicha articulación.

R es una matriz de rotación y translación, la cual define la vista que obtendremos de resultado, para este caso en particular se usara la bird eye, en la cual debemos hacer una rotación sobre el eje x y una rotación de 90 grados en los dos otros ejes correspondientes.

V es el vector de los puntos que vamos a mapear, este vector debe estar expresado en coordenadas homogéneas.

2.4. Filtro de Partículas.

Los filtros de partículas o metodo secuencial de montecarlo (SMC) son un conjunto de algoritmos genéticos utilizados para resolver problemas de filtrado que surgen en el procesamiento de señales y la inferencia estadística bayesiana . El problema de filtrado consiste en estimar los estados internos en sistemas dinámicos cuando se realizan observaciones parciales, y las perturbaciones aleatorias están presentes tanto en los sensores como en el sistema dinámico. El objetivo es calcular las distribuciones posteriores de los estados de algún proceso de Markov, dadas algunas observaciones ruidosas y parciales.

Los filtros de partículas implementan las transiciones de actualización de predicción de la ecuación de filtrado directamente mediante el uso de un algoritmo de partículas de selección de mutación de tipo genético . Las muestras de la distribución están representadas por un conjunto de partículas; cada partícula tiene asignado un peso de verosimilitud que representa la probabilidad de que esa partícula se muestree desde la función de densidad de probabilidad. La disparidad de peso que conduce al colapso del peso es un problema común que se encuentra en estos algoritmos de filtrado; sin embargo, puede mitigarse incluyendo un paso de remuestreo antes de que los pesos se vuelvan demasiado desiguales. Se pueden usar varios criterios de remuestreo adaptativo, incluida la varianza de los pesos y la entropía relativa con respecto a la distribución uniforme. En la etapa de remuestreo, las partículas con pesos despreciables son reemplazadas por nuevas partículas en la proximidad de las partículas con pesos mayores.

Desde el punto de vista estadístico y probabilístico, los filtros de partículas pertenecen a la clase de algoritmos de tipo genético / de ramificación , y a las metodologías de partículas de campo de interacción media. La interpretación de estos métodos de partículas depende de la disciplina científica. En Evolutionary Computing , las metodologías de partículas de tipo genético medio de campo a menudo se usan como algoritmos de búsqueda heurísticos y naturales (también conocidos como Metaheuristic). En física computacional y química molecular, se usan para resolver problemas de integración de rutas Feynman-Kac, o calculan medidas de Boltzmann-Gibbs, valores propios superiores y estados fundamentales de Schrödinger.operadores. En Biología y Genética también representan la evolución de una población de individuos o genes en algún ambiente.

El objetivo de un filtro de partículas es estimar la densidad posterior de las variables de estado dadas las variables de observación. El filtro de partículas está diseñado para un modelo oculto de Markov , donde el sistema consiste en variables ocultas y observables. Las variables

observables (proceso de observación) están relacionadas con las variables ocultas (proceso de estado) por alguna forma funcional conocida. De forma similar, el sistema dinámico que describe la evolución de las variables de estado también se conoce de forma probabilística.

Capítulo 3

Descripción de la metodología.

En este capítulo se van a describir los métodos propuestos en el marco teórico, así como la extracción de los parámetros físicos del entorno en el que se va a trabajar, como se menciono en capítulos anteriores sera un campo de fútbol adecuado para robots.

3.1. Obtención de la imagen.

Para la obtención de la imagen usando las cámaras del robot es necesario tener conocimiento de que existe una interfaz de programación de aplicaciones (API), la cual es conocida como NAOQI.

Este mismo es el software principal que controla el robot, responde a las necesidades de robótica comunes, que incluyen: paralelismo, recursos, sincronización, eventos. Este marco permite la comunicación homogénea entre diferentes módulos (movimiento, audio, vídeo), programación homogénea e intercambio homogéneo de información.

El proceso de naoqi consta de la carga de módulos al robot al momento de encenderse, estos módulos se utilizan como intermediarios para hacer la publicación de los métodos ver fig..

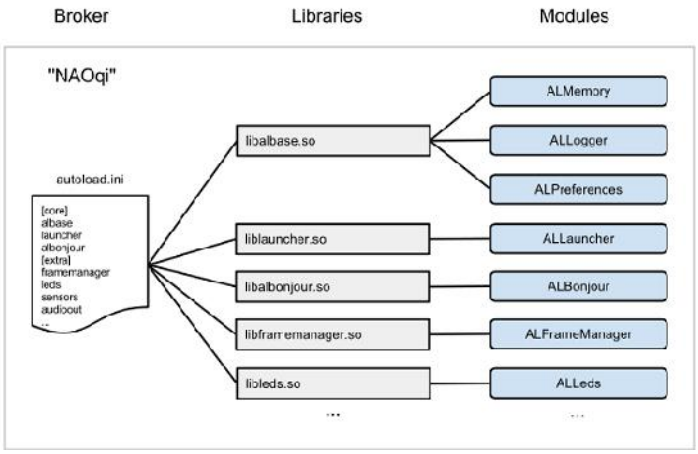


Figura 3.1: Infraestructura de naoqi.

Estos módulos también tiene la versatilidad de obtener acceso mediante una red, ya sea cableada o por WIFI.

Haciendo uso de estos módulos y la ayuda de la biblioteca OpenCV, se crea un programa en c++ para obtener el acceso al modulo de visión, una vez obtenidos el acceso se guardan los datos del bufer de la imagen y se muestran con ayuda de OpenCV.

3.2. Segmentación.

En este trabajo se propusieron dos técnicas de segmentación por color, los cuales describiremos a continuación el proceso utilizado para realizar dichos proceso, estos métodos fueron los de segmentación por filtros, uno en el espacio de color HSV ya descrito en el capitulo 2 y segmentacion por umbralamiento en el espacio cromatico, se utilizaron estos metodos debido a que en muchos trabajos son los más populares y han dado buenos resultados a la hora de trabajar en ambientes no controlados, una de las contra que podemos encontrar en los filtros de este tipo es que si la iluminacion cambia mucho se tendria que calcular los valores del filtro, esto solo para la tecnica del espacio HSV, a diferencia de la segmentacion por cromaticidad

que se cuenta con un modelo de color mas preciso del ambiente y sus variantes.

3.2.1. Segmentación HSV.

Para el diseño del filtro por color en espacio HSV a segmentacióner en consideración el color que vamos a utilizar, en este caso no tenemos tanto problema, ya que el color verde en la representación de RGB es un color que si esta definido, lo siguiente es obtener el valor de todos los verdes con la ayuda de algún programa de edición fotográfica, herramientas con tablas de colores etc..., una vez obtenidos los valores de todas las variaciones de los verdes, es decir desde los verdes mas bajos, hasta los verdes mas altos. Se aplican las ecuaciones: 2.5, 2.6, 2.7. y de esta manera se obtienen los umbrales para los filtros.

3.2.2. Segmentación por Cromaticidad.

Para la realización de este método es necesario conocer el entorno donde se va a trabajar, y tener acceso a el mismo.

Este metodo se propuso en [8] y consta de los siguientes pasos:

1. Tomar imágenes del entorno enfocando las regiones de interés a segmentar, en este caso utilizamos la cámara baja del robot ya que obtiene el mayor numero de muestras del color de interés.

2. Calcular la cromaticidad (pureza del color) de la región de interés utilizando la ecuación 2.8.

3. Realizar un histograma de los resultados de la obtención de la cromaticidad para obtener la función de densidad de nuestra zona de interés.

4. Aplicar la segmentación con el modelo de densidad obtenido.

3.3. Cámara.

Para este proceso es necesario contar con un sistema de ecuaciones, el cual se va a generar mediante una coleccion de puntos vistos por la camara, para esta adquisicion es necesario contar con un tablero como el que se muestra en 3.2

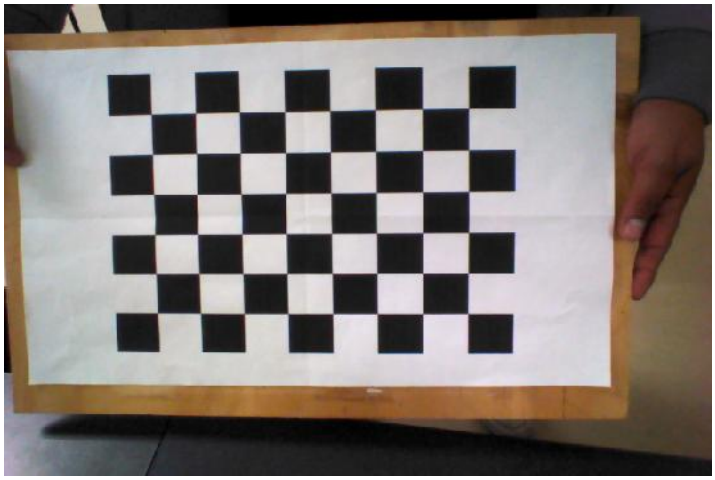


Figura 3.2: Tablero de ajedrez para la obtención de puntos en la imagen.

Este tablero debe ser un plano, por tanto es conveniente fijarlo sobre una base plana rígida. En mi caso he usado un pedazo de tabla de madera. Lo importante es que se mantenga plano el interior del patrón.

La función *findChessboardCorners* recibe una imagen donde aparezca el tablero, encuentra sus esquinas interiores y devuelve las coordenadas en un vector. Además, hay que indicarle la cantidad de filas y columnas de nuestro tablero, pero no de los casilleros, sino de las esquinas interiores ver fig.3.3:

Con esto el programa de Opencv, tiene la infomacion de cuantos puntos buscar en la imagen y le es posible almacenarlos. Los cuales se usaran para calcular los parametros intrinsecos de la camara y a su vez los coeficientes de distorsion.

Dichas matrices se usaran para la rectificacion de la nube de puntos obtenidos como de igual

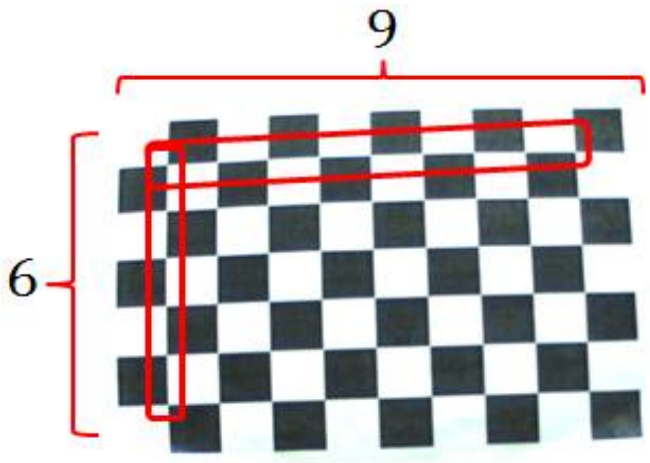


Figura 3.3: Ejemplo de las dimensiones del tablero para la función de OpenCv.

manera para el calculo de la linea al horizonte, lo cual veremos en la sección de los resultados.

Capítulo 4

Resultados

En este capitulo de muestra una evaluación de los resultados obtenidos al aplicar las técnicas propuestas.

Para un mejor entendimiento del sistema completo se muestra un diagrama de flujo del mismo.

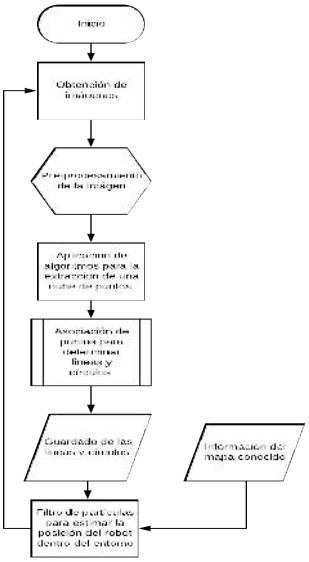
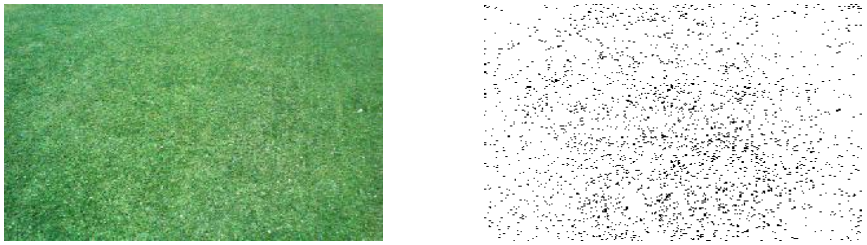


Figura 4.1: Diagrama de flujo del sistema completo.

4.1. Procesamiento de la imagen.

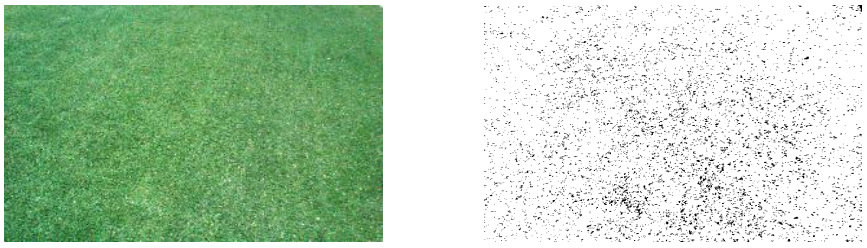
4.1.1. Segmentación.

Se aplicó el algoritmo de segmentación por color en el espacio HSV, para diferentes resoluciones de las imágenes, esto se hizo ya que a mayor resolución de imagen podemos obtener mayor información.



(a) (b)

Figura 4.2: a) imagen original en resolución 320x240 px, b) imagen segmentada por el método de hsv.

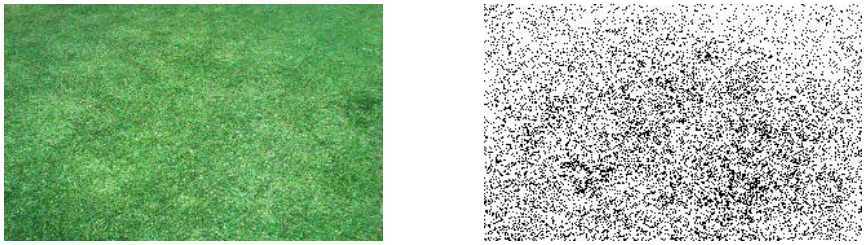


(a) (b)

Figura 4.3: a) imagen original en resolución 640x480 px, b) imagen segmentada por el método de hsv.

como se observa en las imágenes (imágenes) a mayor resolución mejoran los resultados, a lo que nos lleva a elegir una resolución adecuada para el proceso de segmentación, sin afectar los tiempos de procesamiento de la computadora del robot.

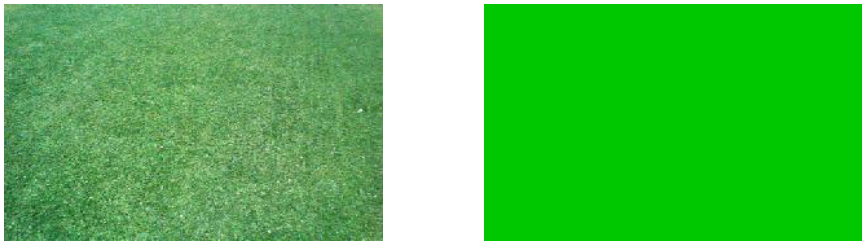
De igual manera se opto por aplicar el método de segmentación por color en el espacio de cromaticidad, la cual nos arrojo los siguientes resultados en las diferentes resoluciones de



(a) (b)

Figura 4.4: a) imagen original en resolución 1280x960 px, b) imagen segmentada por el método de hsv.

imagen.



(a) (b)

Figura 4.5: a) imagen original en resolución 320x240 px, b) imagen segmentada por el método de cromaticidad.

(a) (b)

Figura 4.6: a) imagen original en resolución 640x480 px, b) imagen segmentada por el método de cromaticidad.

A simple vista se ve una mejora en las segmentación, pero se hicieron un par de experimentos más donde comparamos los tiempos de ejecución en el robot para cada uno de los algoritmos utilizados, los resultados se pueden ver en 4.1,4.2, 4.3 cabe mencionar que las pruebas fueron

(a) (b)

Figura 4.7: a) imagen original en resolución 1280x960 px, b) imagen segmentada por el método de cromaticidad.

realizadas en el robot humanoide NAO, el el cual cuenta con un procesador ATOM Z530 1.6GHz CPU 1 GB RAM

A continuación se muestran las tablas comparativas de la efectividad de los filtros.

Comparación de algoritmos de segmentación resolución 320x240				
	hsv-time (ms)	crom-time (ms)	% hsv-efectivo	% crom-efectivo
1	15.15	20.496	94.8724	100
2	8.416	20.383	94.8802	100
3	9.04	20.076	94.9622	99.9987
4	9.956	19.793	94.8424	99.9987
5	8.441	19.792	94.5234	100
6	8.267	19.9	94.6432	99.9987
7	9.979	19.725	94.6458	100

Tabla 4.1: Comparación de algoritmos de segmentación resolución 320x240

Comparación de algoritmos de segmentación resolución 640x480				
	hsv-time (ms)	crom-time (ms)	% hsv-efectivo	% crom-efectivo
1	38.367	78.729	94.542	99.9997
2	38.333	78.603	94.6634	100
3	37.792	79.251	94.5413	99.9997
4	35.804	79.037	94.6217	99.9997
5	36.004	79.753	89.6393	99.9997
6	37.78	78.677	89.9967	100
7	37.792	79.251	94.5413	99.9997

Tabla 4.2: Comparación de algoritmos de segmentación resolución 640x480

Comparación de algoritmos de segmentación resolución 1280x960				
	hsv-time (ms)	crom-time (ms)	% hsv-efectivo	% crom-efectivo
1	155.091	321.596	79.0863	99.9265
2	169.325	317.679	79.1078	99.9272
3	151.335	315.608	79.0245	99.9304
4	165.025	339.433	79.1885	99.922
5	166.816	327.546	78.1133	99.9422
6	165.554	345.903	78.1853	99.9374
7	152.137	331.04	78.1771	99.939

Tabla 4.3: Comparación de algoritmos de segmentación resolución 1280x960

Como podemos ver el método de segmentación por cromaticidad es más efectivo que el de hsv, cabe mencionar que de igual manera es más robusto ante cambios de iluminación.

Adicionalmente a esto se creo una interfaz gráfica para realizar la calibración del filtro en espacio de cromaticidad ver fig.4.8, la cual fue desarrollada en el entorno de desarrollo qt4 en conjunto con la API del robot nao (NAOQI.).

Dicha interfaz permite al usuario acceder a la cámara del robot nao, esto con el fin de realizar una colecta de imágenes las cuales mediante un botón son procesadas para obtener la función de densidad, una vez obtenida dicha función se seleccionan con una click los umbrales del filtro, los cuales se guardan en un archivo de configuración para posteriormente ser llamados a el programa que se utilizara.

Con esta interfaz se proporciona una manera rápida y eficaz de obtener los umbrales para la segmentación, posteriormente estos umbrales pueden ser utilizados en la cámara alta del robot, esto sin importar que las muestras fueron tomadas con la cámara baja del robot.

En las siguientes imágenes se muestran fotogramas tomados por la cámara alta y su segmentación en un entorno no controlado, esto para hacer una prueba de robustes con el algoritmo.

Observando las imágenes resultantes en un entorno y los resultados satisfactorios, se elige este algoritmo para la aplicación final, con lo que damos paso al calculo de la linea al horizonte, la cual nos otorgara una mejora de rendimiento computacional. Para este calculo se

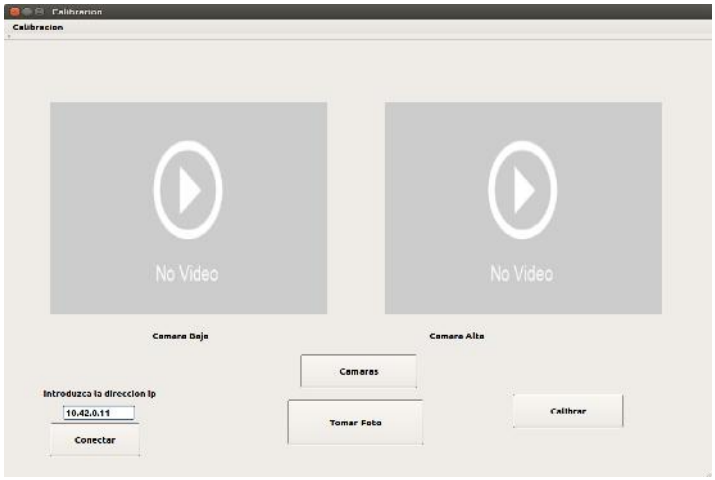
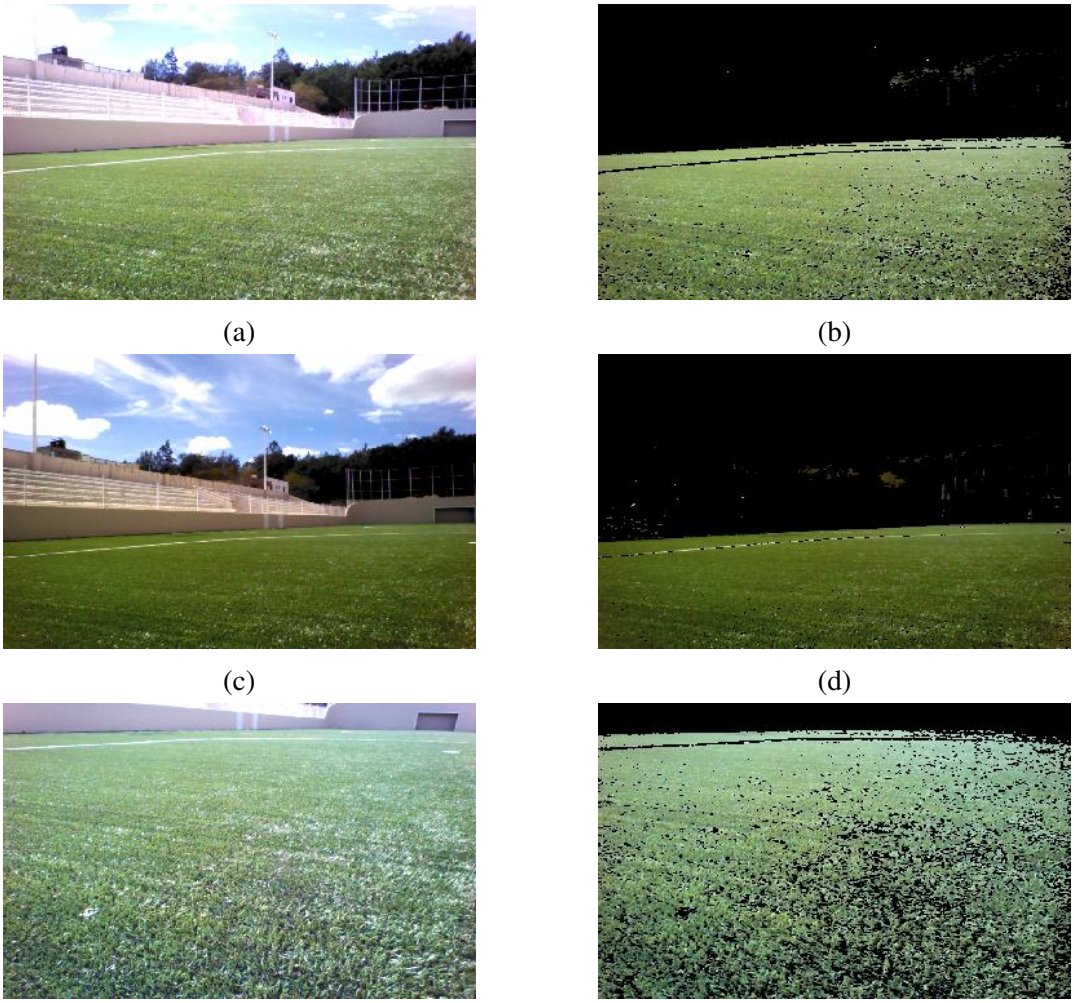


Figura 4.8: Diseño de la interfaz utilizada para calibrar el color del entorno a procesar.

utilizo la matriz de parámetros intrínsecos y la formula propuesta y escrita en el cap. 3. Con la cual se obtuvieron los siguientes resultados.

En las imágenes podemos observar que gracias a el modelo de movimiento de la articulación de la cabeza del robot en el calculo de la linea al horizonte ni influye mucho en que posición este la cabeza, este se ajustara a esos movimientos.



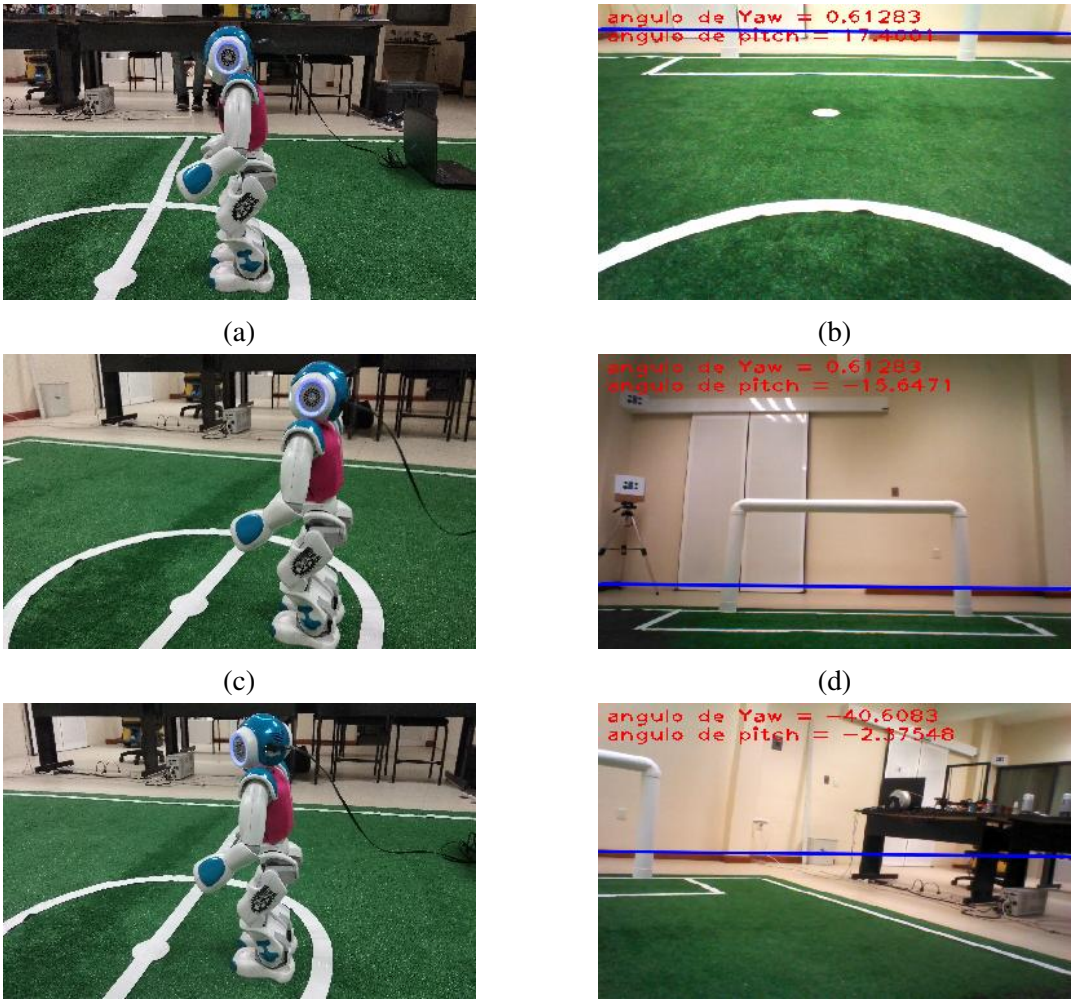
(a) (b)

(c) (d)

(e) (f)

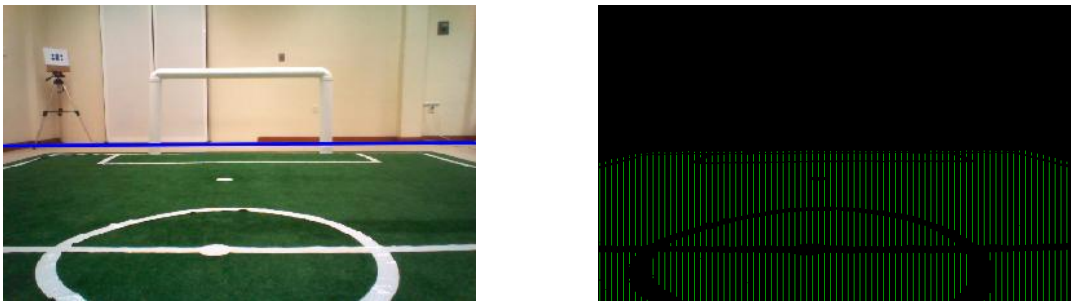
Figura 4.9: a), c), e) Muestras tomadas a la luz exterior, b), d), f) Resultados de la segmentación con umbralamiento.

El algoritmo de scan lines consiste en lineas perpendiculares por debajo de la linea al horizonte. De esta manera se analizan los píxeles que están a lo largo de la linea de búsqueda, y solo se almacenan los píxeles que pertenezcan a una clase requerida Esto lo aplicamos junto con nuestra clase de color verde, cabe mencionar que estas lineas se aplican bajo la linea al horizonte. Y los resultados son los siguientes:



(a) (b) (c) (d) (e) (f)

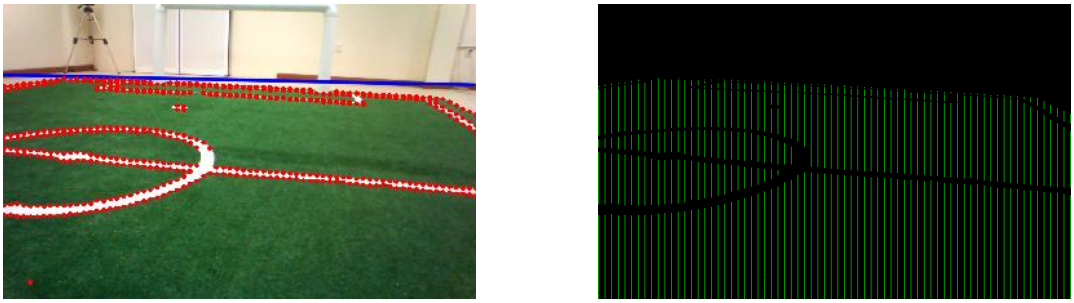
Figura 4.10: Calculo de la linea al horizonte representada con una linea azul.



(a) (b)

Figura 4.11: Clasificación del color verde mediante el algoritmo de scan lines.

Para la detección de los bordes se aplica la derivada a los resultados de la clasificación del color verde, para este caso nos da un resultado de la obtención de los bordes, los cuales se muestran en la siguiente imagen:



(a) (b)

Figura 4.12: Clasificación del color verde mediante el algoritmo de scan lines.

Después de este proceso lo que obtenemos es una nube de puntos en la cual se encuentran las lineas y el circulo principal de la cancha, para ello debemos procesar esta información para obtener las lineas existentes y el circulo central.

4.1.2. Rectificación de la imagen.

Ya teniendo la nube de puntos que pertenece al campo de fútbol, lo normal seria comenzar a buscar las lineas y el circulo central, pero debido a un fenómeno causa por la lente de la cámara esto no es posible.

Para ello se utiliza la técnica llamada IPM ya descrita en la sección de marco teórico.

Una vez obtenida la matriz de la cámara y el modelo de movimiento cinemático de la cabeza del robot, en este caso se añadió el modelo que describe el shaking del robot, esto funciona como un .estabilizador"de la imagen.

Aplicada la técnica nos dio como resultado lo siguiente:

Figura 4.13: Imagen rectificada con la técnica IPM.

Para un fin ilustrativo se aplico a toda la imagen, cabe mencionar que esta técnica solo se aplicara a los puntos que contienen las lineas, un ejemplo del resultado es el siguiente:

(a) (b)

Figura 4.14: Rectificación a la nube de puntos obtenida con las técnicas de segmentación.

con la rectificación se observa que recuperamos los ángulos que forma la cancha los cuales son rectos, de igual manera podemos obtener una relación directa entre cm y píxeles. Con la cual ya podemos estimar una profundidad, no olvidemos que todo esto esta dentro del plano x,y y no tenemos información del pĺano z cuando hablamos de la profundidad.

4.1.3. Búsqueda de lineas y circulo.

Para la búsqueda de lineas se utiliza la nube de puntos rectificada creada anteriormente, esto sera la entrada a nuestro algoritmo de la transformada de Hough. Para un panorama más amplio se muestran 4 imágenes esenciales para lograr el resultado esperado.

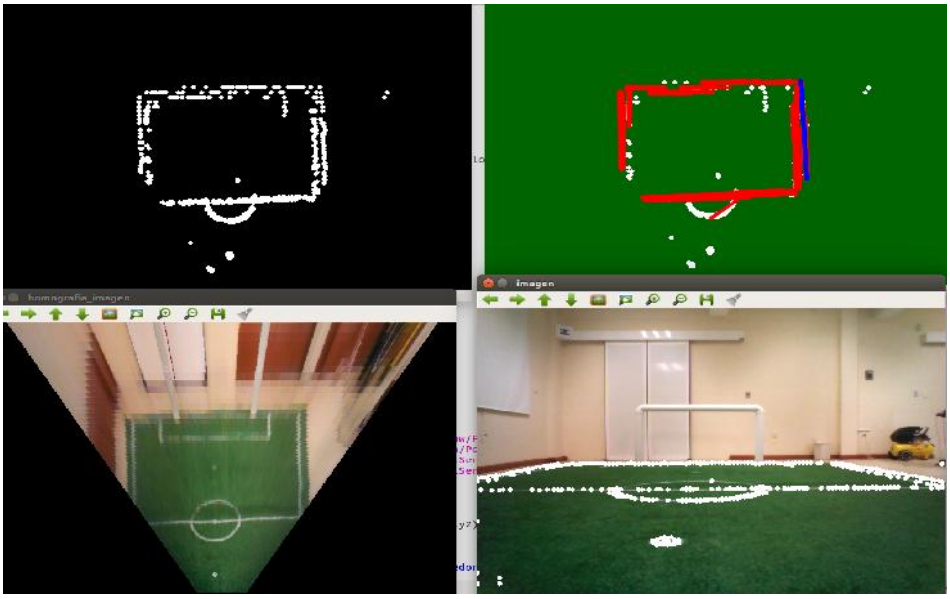


Figura 4.15: Resultado de la palicacion del algoritmo de búsqueda de lineas, mostrando de color rojo las lineas encontradas dentro del campo de fútbol.

Para la búsqueda del circulo se implemento el algoritmo de RANSAC no sin antes medir la eficiencia, ya que venimos cuidando los tiempos de procesamiento, se hizo una prueba con una imagen genérica y el resultado se plasma en la siguiente imagen:

Una vez satisfechos con los resultados se aplica a la nube de puntos rectificadas, y el resultado es el siguiente, cabe mencionar que se hizo una suma de imagenes para corroborar que efectivamente es el circulo que pertenece a el campo de fútbol.

Esta información sera la que va a alimentar el filtro de partículas para lograr la localización del robot en este entorno.

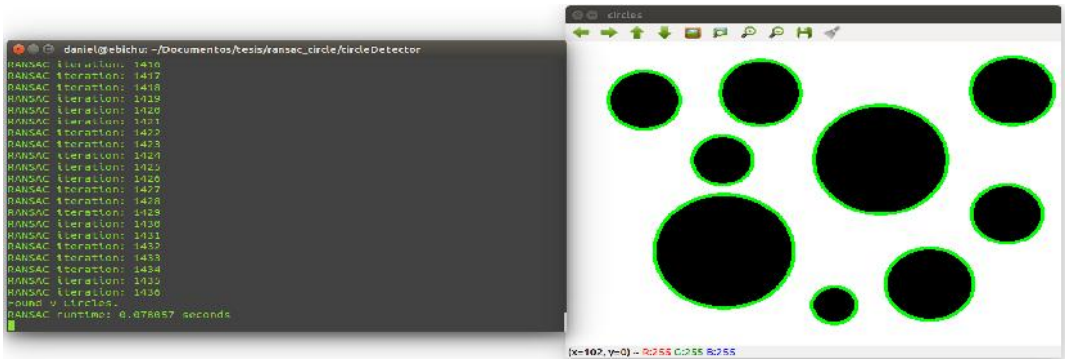


Figura 4.16: Muestra de la eficiencia del algoritmo de RANSAC.

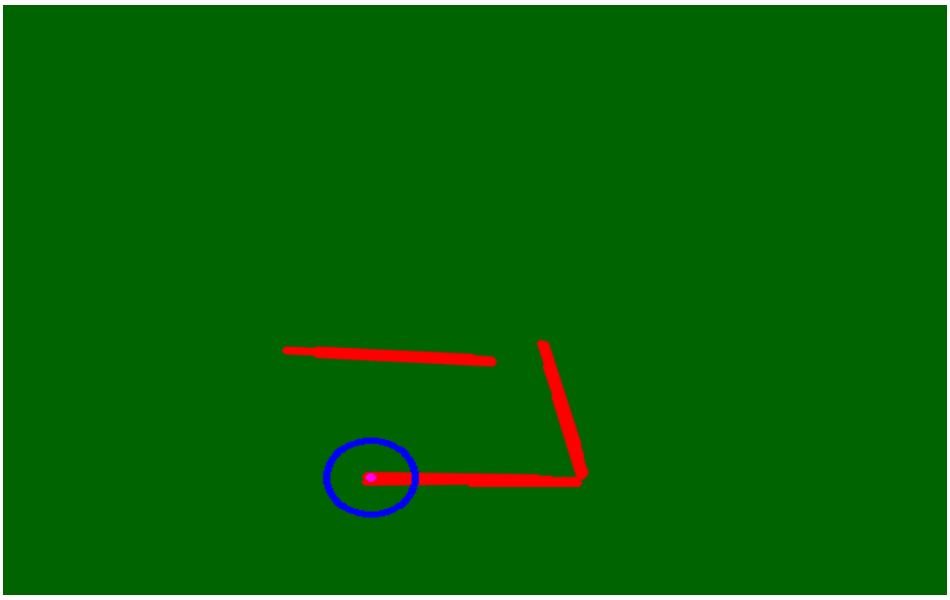


Figura 4.17: Percepción del robot del entorno.

4.2. Conclusiones

En la elaboración del sistema de localización, nos enfrentamos a varios problemas, los cuales se fueron resolviendo siempre procurando optimizar el computo.

Para el proceso de segmentación se optimizo y se realizo una interfaz gráfica, la cual mostramos en los resultados. Con esta interfaz se garantiza una obtención de los umbrales a segmentar de una manera rápida y sencilla, la cual permita a usuarios promedio poder utilizar este sistema.

Figura 4.18: Suma de las imágenes para rectificar que es el circulo central del campo.

En el proceso de la obtención de la nube de puntos las diferentes técnicas de procesamiento y geométria proyectiva se conjuntaron para lograr el objetivo.

El cual era extraer las características tratando de tener los mínimos falsos negativos o algún otro ruido que pudiera entorpecer el funcionamiento del sistema completo.

Se calibró la cámara y se obtuvo la linea al horizonte, la cual fue de gran ayuda a la hora de optimizar tiempos de procesamiento.

Con la exitosa creación de la nube de puntos se eligieron uno de los mejores algoritmos, hablando en eficiencia y confiabilidad, ya que os resultados fueron los esperados, no omitiendo el ajuste de parámetros necesario que se hicieron para llegar al resultado esperado.

Se logró la localizacion con el filtro de particulas, de una manera exitosa.

4.3. Trabajo a futuro.

Con la obtención de la linea al horizonte se planea crear un sistema de estimación para el robot, el cual solo funcionaria para medir desplazamiento de manera vertical.

De igual manera hacer control de posición con el robot dentro del campo de fútbol, así como la ejecución de trayectorias de un robot.

Se planea usar este sistema de localización para que los robots puedan hacer jugadas dentro del campo que involucren pases a sus otros compañeros.

Bibliografía

[1] Arulampalam, B. R. S. (2004). *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House.

[2] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188.

[3] Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208.

[4] Duygulu, P., Barnard, K., de Freitas, J. F., and Forsyth, D. A. (2002). Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European conference on computer vision*, pages 97–112. Springer.

[5] Fu, K. S., Gonzalez, R., and Lee, C. G. (1988). *Robotics: Control Sensing. Vis.* Tata McGraw-Hill Education.

[6] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[7] Hornung, A., Wurm, K. M., and Bennewitz, M. (2010). Humanoid robot localization in complex indoor environments. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1690–1695. IEEE.

[8] Jüngel, M., Hoffmann, J., and Lötzsch, M. (2003). A real-time auto-adjusting vision system for robotic soccer. In *Robot Soccer World Cup*, pages 214–225. Springer.

[9] Rodriguez Luque, R. (2006). *Localización multirrobot basada en filtro de partículas*. PhD thesis, Universidad de Alcalá.

[10] Sossa Azuela, J. H. (2013). Visión artificial. *España: Editorial RA-MA*.

UNACAR
Universidad Autónoma del Carmen
"Por la Grandeza de México"

ACUERDO PARA USO DE OBRA

A quien corresponda
PRESENTE

Por medio del presente escrito, **Daniel Silva Medina** (en lo sucesivo EL AUTOR) hace constar que es titular intelectual de la obra denominada, "**Sistema de localización usando un filtro de partículas para un robot humanoide en un campo de fútbol**" (en lo sucesivo la OBRA), en virtud de lo cual autoriza a la Universidad Autónoma del Carmen (en lo sucesivo la UNACAR) para que efectué resguardo físico y/o electrónico mediante copia digital o impresa para asegurar su disponibilidad, divulgación, comunicación pública, distribución, transmisión, reproducción, así como digitalización de la misma con fines académicos y sin fines de lucro como parte de Repositorio Institucional de la UNACAR (*Runacar*).

De igual manera, es deseo del AUTOR establecer que esta autorización es voluntaria y gratuita, y que de acuerdo a lo señalado en la Ley Federal del Derecho de Autor y la Ley de Propiedad Industrial, la UNACAR cuenta con mi autorización para la utilización de la información antes señalada, estableciendo que se utilizará única y exclusivamente para los fines antes señalados. EL AUTOR autoriza a la UNACAR a utilizar las obras en los términos y condiciones aquí expresados, sin que ello implique que se le conceda licencia o autorización alguna o algún tipo de derecho distinto al mencionado respecto a la "propiedad intelectual" de la misma OBRA; incluyendo todo tipo de derechos patrimoniales sobre obras y creaciones protegidas por derechos de autor y demás formas de propiedad intelectual reconocida o que lleguen a reconocer las leyes correspondientes. Al reutilizar, reproducir, trasmitir y/o distribuir la OBRA se deberá reconocer y dar crédito de autoría de la obra intelectual en los términos especificados por el propio AUTOR, y el no hacerlo implica el término de uso de esta licencia para los fines estipulados. Nada de esta licencia menoscaba o restringe los derechos patrimoniales y morales del AUTOR.

De la misma manera, se hace manifiesto que el contenido académico, literario, la edición y en general de cualquier parte de la OBRA son responsabilidad del AUTOR, por lo que se deslinda a la UNACAR por cualquier violación a los derechos de autor y/o propiedad intelectual, así como cualquier responsabilidad relacionada con la misma frente a terceros. Finalmente, el AUTOR manifiesta que estará depositando la versión final de su Tesis de maestría, OBRA y cuenta con los derechos morales y patrimoniales correspondientes para otorgar la presente autorización de uso.

En la ciudad de Carmen, del estado de Campeche a los 12 días el mes de octubre de 2018.

Atentamente,

DANIEL SILVA MEDINA

Nombre y Firma Autógrafa de EL AUTOR

Escriba la Facultad, Escuela, Centro a la que está suscrita la obra: Facultad de Ingeniería, Universidad Autónoma del Carmen, Campus III.