

CENTRO DE INVESTIGACIONES EN ÓPTICA, A. C.



**CENTRO DE INVESTIGACIONES
EN OPTICA, A.C.**

TESIS DE MAESTRÍA

**DISEÑO DE ROBOT MÓVIL Y CONTROL MEDIANTE VISIÓN POR
COMPUTADORA PARA LA NAVEGACIÓN EN UN ESPACIO
CONOCIDO.**

Por

**JOSÉ MANUEL SÁMANO MOLINA
Ing. en Electrónica**

SEPTIEMBRE 2013

Asesor

Dr. Julio César Estrada Rico

Sinodales

Dr. María del Socorro Hernández Montes

Dr. Apolinar Muñoz Rodríguez

Fecha de Graduación

Septiembre 2013

Índice general

1. Introducción	3
1.1. Antecedentes	3
1.2. Funcionamiento del sistema de control para el robot móvil.	5
1.3. Condición del modelo	5
2. Calibración de Cámara y Geometría Projectiva.	7
2.1. Geometría Projectiva	8
2.1.1. Planos, puntos y líneas rectas	9
2.1.2. Transformaciones Projectivas	12
2.1.3. Rectificación de distorsión projectiva	14
2.1.4. Transformación Projectiva General	16
2.2. Modelo básico de cámara, punta de alfiler (<i>pinhole</i>)	17
2.3. Calibración de cámara	21
2.3.1. Ecuaciones básicas	22
2.3.2. Restricciones sobre los parámetros intrínsecos	24
2.3.3. Interpretación Geométrica	24
2.3.4. Solución de la Calibración de cámara	25
3. Diseño del Robot Móvil y Sistema de Control Remoto	31
3.1. Descripción de los principales componentes	32
3.1.1. El microcontrolador	32
3.1.1.1. Características de los microcontroladores	33
3.1.2. Módulos de radiofrecuencia	35
3.1.2.1. Módulo de Transmisión	35
3.1.2.2. Módulo de Recepción	35
3.1.3. Conversor TTL-RS232 (MAX232)	36

3.1.4.	Motores paso a paso	37
3.1.4.1.	Características comunes de los motores paso a paso . . .	38
3.2.	Subsistema de Procesamiento del robot móvil	40
3.3.	Subsistema actuador	41
3.4.	Subsistema de comunicación	42
3.4.1.	Envío y recepción	44
3.5.	Diseño del prototipo	45
4.	Planeación Trayectoria y Movimiento	49
4.1.	Exploración Rápida de Árboles Aleatorios (Rapidly-Exploring Random Trees)	50
4.2.	Análisis y Modelo Cinemático del Robot Móvil	53
4.3.	Controlabilidad sobre una Trayectoria	57
5.	Implementación del Sistema de Visión por Computadora	59
5.1.	Implementación de Hardware	59
5.1.1.	Cámara	59
5.1.2.	Sistema de iluminación	60
5.1.3.	Interfaz control remoto PC- Robot Móvil	63
5.2.	Implementación del Software	63
5.2.1.	Control del Robot Móvil	64
5.2.1.1.	Descripción del funcionamiento	64
5.2.2.	Calibración de cámara	65
5.2.3.	Transformación proyectiva del entorno	65
5.2.4.	Preprocesamiento de la imagen	67
5.2.4.1.	Filtrado	69
5.2.4.2.	Segmentación por filtro de color	70
5.2.4.3.	Umbralización	75
5.2.4.4.	Binarización	75
5.2.4.5.	Erosión y Dilatación	75
5.2.5.	Generación del entorno	79
5.2.5.1.	Detección de obstáculos	82
5.2.5.2.	Detección del Robot Móvil	83
5.2.6.	Planeación de Trayectoria	85

6. Conclusiones	91
7. Trabajo a futuro	93
Bibliografía	95

***i.* Resumen**

El diseño de un robot móvil y el control de sus movimientos por computadora se presenta en este trabajo. El proyecto consistió en el diseño de un robot móvil de bajo costo para la navegación de un espacio conocido mediante el control por visión por computadora y algoritmos RRT (Rapidly-Exploring Random Trees) para su navegación. Con este diseño se resolvió la coordinación del robot móvil logrando el movimiento y desplazamiento de un lugar a otro ajustándose a la cinemática de un automóvil actual con la opción de estacionarse en espacios estrechos y distintas posiciones de forma completamente autónoma y evitando todos los obstáculos en su camino. El desplazamiento del robot móvil es controlado automáticamente por un computador a través de un control remoto por radio frecuencia basada en la información que proporciona el computador por una cámara mediante la captura de imágenes. Una de las ventajas del robot móvil construido es que este obedece tareas similares a otros robots que cuentan con múltiples sensores incorporados para moverse por interiores. Este trabajo muestra la importancia en el área de visión por computadora para usos en el control de la navegación móvil.

***ii.* Justificación**

En esta tesis se aborda el problema de la planeación de trayectoria en sistemas robóticos móviles. Se construyó un robot móvil de bajo costo para múltiples tareas haciendo uso de una cámara digital como único sensor externo al robot móvil. Una computadora realiza el control general para la evasión de obstáculos del robot móvil que se encuentra libre de sensores para su navegación, es decir; la única información entrante para la navegación será por la cámara.

iii. Objetivos

- Diseñar y construir un prototipo robot móvil inalámbrico para uso general.
- Diseñar y construir una interfaz de control remoto para el robot móvil, capaz de ser controlado por un computador a través de un periférico.
- Generar un algoritmo para generar el entorno de trabajo en un ambiente de dos dimensiones.
- Generar un algoritmo de planeación de trayectoria basado en RRT sobre el entorno generado en dos dimensiones.
- Ejecutar instrucciones del movimiento del robot móvil para seguir la ruta planeada.

iv. Prólogo

El capítulo I aborda la introducción al tema de la robótica y visión por computadora como también sus antecedentes, función del sistema y las condiciones del sistema.

En el capítulo II, se presenta teoría e información necesaria para la completa comprensión del trabajo realizado. Se explica el uso y aplicación de la geometría proyectiva, modelo de cámara, y calibración de cámara que se usaron en el sistema.

En el capítulo III, se muestra el diseño del robot móvil de bajo costo, su construcción y explicación de los bloques que la componen, así como también el diseño y construcción de control remoto que es conectado a una computadora terminal para mandar las instrucciones al robot móvil. También se presenta como la terminal puede conectarse con el programa principal del sistema desde otra computadora por medio de una plataforma TCP/IP.

En el capítulo IV, se describe la metodología de la planeación de trayectoria que se uso en el sistema, consideraciones y modelo, así como también los resultados obtenidos ya en conjunto y el sistema completo funcionando.

La implementación del sistema se presenta en el capítulo V describiendo las etapas del sistema de visión y explicando métodos usados.

El capítulo VI y VII, se dan las conclusiones finales sobre el desempeño de la propuesta y se discuten consideraciones para trabajo a futuro con esta línea de investigación.

1 Introducción

La robótica móvil es un área de interés actual por sus potenciales aplicaciones en diversas áreas de la ciencia e ingeniería. Hoy en día, los principales objetivos es el desarrollo de robots autónomos que puedan desenvolverse en entornos estructurados dinámicos o estáticos. En robótica móvil se pueden encontrar tres problemas fundamentales:

- La navegación: Desplazarse a través de un entorno para alcanzar una meta o una posición determinada.
- La localización: La capacidad de poder ubicarse en el entorno haciendo uso de un sistema sensorial que pueda proporcionar información del ambiente en el que se encuentra.
- Planeación de trayectoria y construcción de mapas de entorno: La reconstrucción un mapa a medida que se desplazan en el entorno estático, dinámico o ya sea reconstruido virtualmente y planear trayectorias a través de este para alcanzar el objetivo deseado.

1.1. Antecedentes

Actualmente se desarrollan diversas investigaciones y trabajos en el que se han propuesto una variedad de técnicas de posicionamiento de robots móviles en ambientes, como interiores o exteriores. Las técnicas varían ampliamente en función del ambiente en el que se desplace o de acuerdo al conocimiento que tenga del entorno, la tarea a realizar y/o del conjunto de sensores de los que se disponga. Algunas técnicas son basadas en sistemas de medición láser, técnicas de control predictivo neuronal, técnicas por visión de computador entre otras más. El sistema sensorial del que disponga permitirá enfrentarse ante obstáculos que se presenten en su entorno mientras se desplaza

a través de él. En general, la estimación de los parámetros que definen la localización de un robot móvil son estimados a partir de una serie de medidas proporcionadas por los sensores.

En la actualidad es posible encontrar diversos tipos de robots móviles, en el que podemos considerar el precio como un factor importante para la adquisición de alguno. La compra o adquisición de estos dispositivos generalmente se asocia con altos costos, ya sean por razones de trámites de importación, adquisición de repuestos, licencias de software, soporte técnico, etc.

Con este proyecto se pretende lograr algunos objetivos, como dar uso a la aplicación de visión artificial a la robótica móvil autónoma, en donde a parte de reconocer y manipular objetos de orientación y posición desconocida, se pueda implementar el propio sistema de control del robot basado únicamente en la retroalimentación visual y ausencia de sensores en el dispositivo. Además, dicho estudio puede dar inicio a la aplicación de sistemas de visión a otras modalidades de la robótica, por ejemplo sistemas autocontrolados por visión artificial para implementar tareas completamente automáticas con robots móviles de bajo costo.

La captura de la imagen es el conjunto de operaciones que se efectúan para transformar una escena en una señal digital. Las imágenes no siempre se presentan en un formato adecuado para su análisis, por lo que el siguiente proceso es el preprocesamiento de una imagen, en el cual se utilizan técnicas encaminadas a realizar la mejora de la imagen, como son el nivel de gris, contraste, eliminación del ruido, el realce de algunas características de interés, etc. Una vez que esta imagen está en condiciones de ser procesada, se aplica corrección de perspectiva a la imagen para tener una relación con el mundo real, posteriormente hallar los objetos dentro de la imagen de forma independiente, y esto se hace a través de la segmentación, que es un proceso que divide la escena en objetos. Cada uno de los objetos puede ser clasificado, por lo que la siguiente tarea es la clasificación o extracción de características para el reconocimiento. El reconocimiento es la identificación de cada objeto en la escena mediante una serie de puntos que determinan la forma de los objetos.

1.2. Funcionamiento del sistema de control para el robot móvil.

De forma generalizada el sistema se basa en las siguientes acciones:

- Moverse de su posición actual a la posición indicada por el usuario.
- Evasión de los obstáculos.

De forma más particular, podemos observar el siguiente diagrama a bloques del sistema mostrado en la Figura 1.2.1, en donde se muestran los pasos a seguir para lograr las acciones generales. La figura nos muestra dos bloques, el primer bloque situado en la parte superior de la imagen indica los pasos a seguir una vez iniciado el sistema. Posteriormente el segundo bloque, es la parte autónoma del sistema.

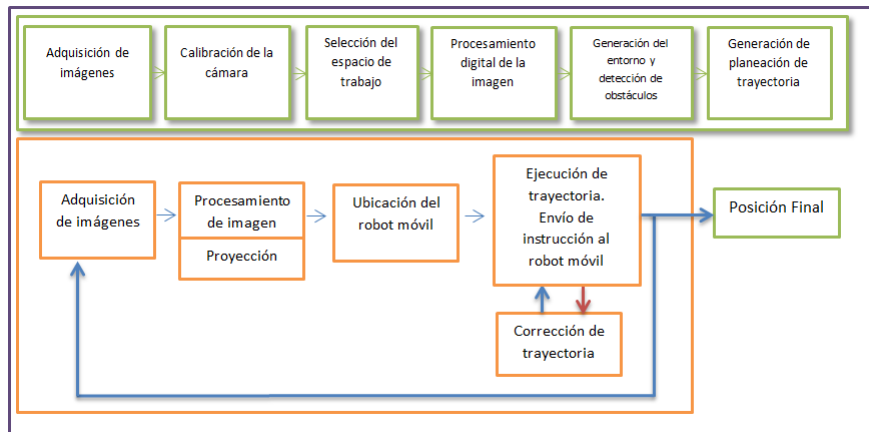


Figura 1.2.1: Diagrama a bloques del funcionamiento del Sistema de planeación de trayectoria para el robot móvil.

1.3. Condición del modelo

El prototipo tiene las siguientes condiciones:

- El área de trabajo del robot móvil tiene medidas de $150\text{ cm} \times 180\text{ cm}$.
- El espacio de trabajo de ser completamente estático., no se permite alteraciones del medio.

- Iluminación homogénea dentro del escenario.
- La posición de la cámara debe ser estática y no se permite moverla después de iniciarse el sistema.
- Los obstáculos se manejarán como plantillas sin altura.

2 Calibración de Cámara y Geometría Projectiva.

El uso regular de cámaras no profesionales para usos métricos ha colocado la cuestión de la calibración de las cámaras en un lugar privilegiado de estudio. Así, encontramos abundante literatura reciente que aborda este tema, aportando nuevas propuestas, aplicaciones y formulaciones.

Una cámara es una transformación entre el mundo 3D (espacio objeto) y una imagen 2D. El modelo de cámara de interés de la que hablaremos será *proyección central*. Trataremos en la segunda sección un *modelo* de cámara que son matrices con propiedades particulares que representan el mapeo de la cámara. Sabemos que todas las cámaras de modelo de proyección central son especializaciones de *cámara de proyección general*. La estructura de este modelo de cámara general es examinado usando las herramientas de *Geometría Projectiva*. Veremos que entidades geométricas de la cámara, tal como el centro de proyección y el plano imagen, pueden ser calculados simplemente desde su representación matricial.

Algunos modelos especializados caen dentro de dos clases principales, los modelos de cámara con centro finito y los modelos de cámaras con centro en “*infinito*”. De las cámaras con centro en el infinito, la *cámara afín* es de particular importancia porque es la generalización natural de la proyección paralela. Para este trabajo sólo nos enfocaremos en un modelo acerca de cámaras finitas como el modelo básico *pinhole* utilizado en este trabajo.

Para la mayor comprensión del trabajo realizado, necesitamos de una herramienta algebraica muy importante como es la *Geometría Projectiva*. En la siguiente sección a continuación se ahonda más sobre geometría projectiva.

2.1. Geometría Proyectiva

Estamos familiarizados con las transformaciones proyectivas. Cuando miramos una imagen, nosotros vemos cuadros que no son cuadros, o círculos que no son círculos. La transformación de estos objetos planos en la imagen es un ejemplo de una transformación proyectiva.

Entonces, ¿qué propiedades de la geometría son preservadas por las transformaciones proyectivas?, sin duda, la forma no es, ya que un círculo puede aparecer como una elipse. Tampoco lo son las longitudes de dos radios perpendiculares de un círculo que se extendieron por diferentes cantidades por la transformación proyectiva. Ángulos, distancias, proporciones de distancias, ninguno de ellos se conservan, y puede parecer que la geometría muy poco se conserva de una transformación proyectiva. Sin embargo, una propiedad que se conserva es el de la **rectitud**. Resulta que este es el requisito más general sobre un plano, y podemos definir una transformación proyectiva de un plano como cualquier asignación de los puntos en el plano que conserva las líneas rectas.

Mediante estas transformaciones es posible modelar la distorsión geométrica que le ocurre a un plano que es visto por una cámara perspectiva. Con estas transformaciones algunas propiedades se conservan como la colinealidad (líneas rectas son vistas como líneas rectas), mientras que otras propiedades no (las líneas paralelas por lo general no son vistas como líneas paralelas).

Para ver por qué requerimos de la geometría proyectiva comenzaremos desde la geometría euclidiana. Esta es la geometría que describe los ángulos y las formas de los objetos. La geometría euclidiana es problemático en un aspecto importante, tenemos que seguir haciendo una excepción a la razón de algunos de los conceptos básicos de la geometría como la intersección de las líneas. Dos líneas (estamos pensando en dos dimensiones la geometría), casi siempre se reúnen en un punto, pero hay algunos pares de líneas que no lo hacen, los que llamamos paralelos. Una forma lingüística común para entender alrededor de esto es, decir que las líneas paralelas se encuentran "en el infinito". Sin embargo, esto no es del todo convincente, y en conflicto con otra sentencia, que el infinito no existe, y es sólo una ficción conveniente.

Podemos evitar esto mejorando el plano euclidiano con la adición de estos puntos en el infinito en paralelo líneas se encuentran, y resolver las dificultades con el infinito llamando a los "puntos ideales". Al añadir estos puntos en el infinito, el espacio euclidiano

se transforma en un nuevo tipo de objeto geométrico, el espacio proyectivo. Esta es una manera muy útil de pensar, ya que estamos familiarizados con las propiedades del espacio euclidiano, que implican conceptos como las distancias, ángulos, puntos, líneas y la incidencia. No hay nada misterioso en el espacio de proyección es sólo una extensión del espacio euclidiano en el que dos líneas siempre se reúnen en un punto, aunque a veces en los puntos misteriosos en el infinito.

2.1.1. Planos, puntos y líneas rectas

Un punto en un plano se representa como un par de coordenadas (x, y) en \mathfrak{R}^2 . Comúnmente, \mathfrak{R}^2 se identifica con un plano. Se puede considerar entonces \mathfrak{R}^2 como un espacio vectorial en el que (x, y) es un vector. Se asocia así, un punto a un vector. Los vectores los representaremos en letra negrita y sus elementos serán dispuestos de manera vertical, $\mathbf{m} = (x, y)^T$. Una línea recta en el plano (x, y) está representada por la ecuación: $ax + by + c = 0$. [14]

De esta forma, una línea puede ser naturalmente representado por el vector $(a, b, c)^T$. La correspondencia entre líneas rectas y vectores no es uno a uno, ya que $ax + by + c = 0$ y $(ka)x + (kb)y + (kc) = 0$ representan la misma línea recta para $k \neq 0$. De esta forma los vectores $(a, b, c)^T$ y $k(a, b, c)^T$ representan de igual forma la misma línea para $k \neq 0$. De hecho estos vectores relacionados por una escala global son considerados como equivalentes y son definidos como vectores homogéneos.

Podemos representar un punto homogéneo de la siguiente forma. Un punto $\mathbf{x} = (x, y)^T$ se encuentra sobre la línea $\mathbf{l} = (a, b, c)^T$ si y sólo si $ax + by + c = 0$. Esto puede ser escrito en términos de un producto punto de vectores representando el punto como $\mathbf{l} \cdot \mathbf{m} = \mathbf{l}^T \mathbf{m} = [x \ y \ 1][a \ b \ c]^T = ax + by + c = 0$. [14]

Así el punto (x, y) es representado por un vector $(x, y, 1)^T$. Sin embargo, los vectores (kx, ky, k) , para $k \neq 0$, pueden considerarse también como representaciones del mismo punto (x, y) ya que satisfacen la ecuación $[a \ b \ c]^T [kx \ ky \ kz] = k(ax + by + c) = 0$. Se dice entonces, que los puntos en un plano bidimensional pueden ser representados por vectores homogéneos de tres dimensiones cuyos dos primeros elementos son las coordenadas del punto en el plano y el tercer elemento es 1.

En términos generales, si se tiene un vector homogéneo de tres dimensiones dado por

$(x_1, x_2, x_3)^\top$ que representa un punto en un plano, las coordenadas de este punto en el plano están definidas como $(x, y) = (x_1/x_3, x_2/x_3)$. [14]

Intersección de Líneas. Dadas dos rectas $\mathbf{l} = (a, b, c)^\top$ y $\mathbf{l}' = (a', b', c')^\top$ deseamos encontrar el punto de intersección de ambas rectas.

Del álgebra vectorial se sabe que el producto cruz de dos vectores se obtiene un vector que es perpendicular a ellos. También se sabe que el producto punto entre dos vectores perpendiculares es igual a cero. Por lo tanto, se puede afirmar que $\mathbf{l} \cdot (\mathbf{l} \times \mathbf{l}') = \mathbf{l}' \cdot (\mathbf{l} \times \mathbf{l}') = 0$. Si se define el punto \mathbf{x} como $\mathbf{l} \times \mathbf{l}'$ obtenemos la representación homogénea que satisface las ecuaciones $\mathbf{l}^\top \mathbf{x} = 0$ y $\mathbf{l}'^\top \mathbf{x} = 0$ lo que quiere decir que el punto \mathbf{x} es la intersección de las rectas ya que pertenece a ellas.

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' \quad (2.1.1)$$

Nótese que la simplicidad de esta expresión para la intersección de dos líneas es una consecuencia directa de el uso de representaciones de vectores homogéneos de líneas y puntos.

Línea recta que pasa por dos puntos. Mediante el mismo razonamiento podremos ejemplificar una línea recta que pasa por dos puntos. Dado dos puntos $\mathbf{x} = (x, y, 1)^\top$ y $\mathbf{x}' = (x', y', 1)^\top$ y que se desea encontrar la línea recta que pasa por ambos puntos, podemos definirlo de la siguiente forma:

$$\mathbf{l} = \mathbf{x} \times \mathbf{x}' \quad (2.1.2)$$

ya que $\mathbf{l}^\top \mathbf{x} = 0$ y $\mathbf{l}^\top \mathbf{x}' = 0$, lo cual quiere decir que tanto como \mathbf{x} como \mathbf{x}' pertenecen a la misma recta.

Intersección de líneas paralelas. Consideremos dos líneas, $ax + by + c = 0$ y $ax + by + c' = 0$. Estas son representados por los vectores $\mathbf{l} = (a, b, c)^\top$ y $\mathbf{l}' = (a, b, c')^\top$ en el cual las primeras dos coordenadas son las mismas, y se desea encontrar el punto de intersección. Según Ecuación 2.1.1 la representación homogénea del punto de intersección está dada por $\mathbf{x} = \mathbf{l} \times \mathbf{l}' = (c' - c)(b, -a, 0)^\top$. Ahora si nosotros intentamos encontrar la representación no homogénea de este punto, obtenemos $(b/0, -a/0)^\top$ que no tiene sentido, salvo que sugiere que el punto de intersección tiene coordenadas infinitamente grandes. La representación no homogénea de este punto representa una

singularidad debido a la división por cero. El punto obtenido de la división por cero puede ser interpretado como un punto en el infinito. En general, puntos con coordenadas homogéneas de este tipo $(x, y, 0)^\top$ no corresponde a algún punto finito en \mathbb{R}^2 y son conocidos como *puntos ideales*. Esta observación concuerda con la idea usual que las líneas paralelas se intersectan en el infinito. Ejemplificando esto, consideraremos dos líneas $x = 1$ y $x = 2$. Las dos líneas son paralelas, y consecuentemente se intersectan “en el infinito”. En notación homogénea las líneas son $\mathbf{l} = (-1, 0, 1)^\top$, $\mathbf{l}' = (-1, 0, 2)^\top$, aplicando la Ecuación 2.1.2 el punto de intersección es

$$\mathbf{x} = \mathbf{l} \times \mathbf{l}' = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -1 & 0 & 1 \\ -1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

que es el punto en el infinito en la dirección del *eje y*. Todos los puntos ideales pertenecen a la línea en el infinito $\mathbf{l}_\infty = [0 \ 0 \ 1]^\top$, ya que $[x_1 \ x_2 \ 0]^\top [0 \ 0 \ 1] = 0$. De esta manera una recta $\mathbf{l} = (a, b, c)^\top$ intersecta a \mathbf{l}_∞ en $\mathbf{x} = \mathbf{l} \times \mathbf{l}_\infty = [b \ -a \ 0]^\top$. Entonces, una línea \mathbf{l}' paralela a \mathbf{l} intersecta a \mathbf{l}_∞ en el mismo punto ideal. Es interesante observar que la *Geometría Proyectiva* de dos líneas rectas paralelas se intersectan en un punto “ideal”, mientras que; en la *Geometría Euclidiana* la intersección de dos rectas paralelas no está definida.

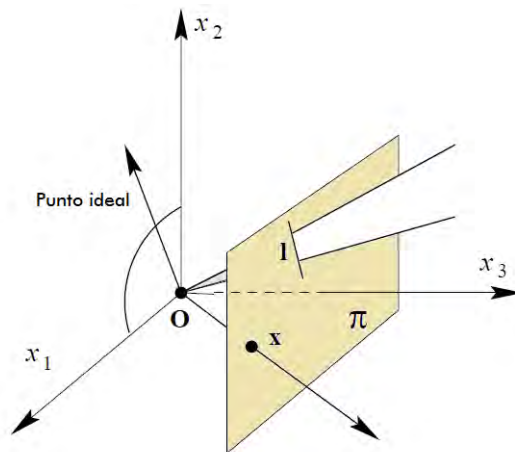


Figura 2.1.1: Un modelo del plano proyectivo.

Como señalamos anteriormente, un punto (x, y) en un plano tiene una representación homogénea en \mathbb{R}^3 dada por $\mathbf{x} = [x_1 \ x_2 \ x_3]^\top$ con $x = x_1/x_3$ y $y = x_2/x_3$. El plano proyec-

tivo \mathbb{P}^2 en el que están representados los puntos no homogéneos $x = x_1/x_3$ y $y = x_2/x_3$ puede ser interpretado utilizando la representación de la Figura 2.1.1. En este esquema la proyección de (x_1, x_2, x_3) es un plano (x, y) paralelo al plano $(x_1, x_2, 0)$ ubicado en $x_3 = 1$, está dado por el punto (x, y) el cual puede ser calculado aplicando el Teorema de Tales como $x : x_1 = y : x_2 = 1 : x_3$. Se obtiene entonces $x = x_1/x_3$ y $y = x_2/x_3$. Podemos observar que cualquier punto 3D sobre la línea de proyección produce el mismo punto proyectado en el plano (x, y) . Eso es lo mismo que decir que la proyección de $k(x_1, x_2, x_3)$ es igual a la proyección de (x_1, x_2, x_3) , para $k \neq 0$, y que esta proyección está dada por $x = x_1/x_3$ y $y = x_2/x_3$. En otras palabras, la clase de vectores homogéneos $k(x_1, x_2, x_3)$ representan el mismo punto en \mathbb{P}^2 . [14]

2.1.2. Transformaciones Projectivas

Geometría es el estudio de propiedades invariantes bajo grupos de transformaciones. Desde este punto de vista, geometría projectiva 2D es el estudio de las propiedades del plano projectivo \mathbb{P}^2 que son invariantes bajo un grupo de transformaciones conocidas como *projectividades*. [27, 14]

Una projectividad es una transformación invertible de puntos in $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ de manera tal que una línea recta es transformada como otra línea recta. También, la projectividad es llamada *colineación* y, una *transformación projectiva* es llamada *homografía*: los términos son sinónimos. Una definición equivalente algebraica de projectividad es posible basado en el siguiente resultado. *Una transformación $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ es una projectividad si y sólo si existe una matriz \mathbf{H} no singular de 3×3 tal que para un punto en \mathbb{P}^2 es representado por un vector \mathbf{x} , entonces definimos la projectividad como:*

$$h(\mathbf{x}) = \mathbf{x}' = \mathbf{H}\mathbf{x}, \quad (2.1.3)$$

donde \mathbf{H} es una matriz 3×3 no singular. Se dice entonces que \mathbf{x}' es la transformación lineal \mathbf{H} del vector \mathbf{x} . Esta transformación es biunívoca entre dos planos 2D, cuyos puntos son representados homogéneamente por \mathbf{x} y \mathbf{x}' . Es decir; un punto en un plano 2D tiene una única correspondencia en un punto de otro plano 2D, y cada punto en un plano tiene un solo punto correspondiente en el otro plano.

Decir entonces que, una recta de un \mathbb{P}^2 puede ser transformada en otra recta en otro

plano \mathbb{P}^2 puede comprobarse de la siguiente manera: si $\mathbf{x}_1, \mathbf{x}_2$ y \mathbf{x}_3 se encuentran en la misma línea recta \mathbf{l} , entonces $\mathbf{l}^\top \mathbf{x}_i$ para $i = 1, 2, 3$. Esta ecuación puede ser escrita como $\mathbf{l}^\top \mathbf{H}^{-1} \mathbf{H} \mathbf{x}_i = 0$. Todos los puntos $\mathbf{H} \mathbf{x}_i$ se encuentra sobre la línea $\mathbf{H}^{-\top} \mathbf{l}$, y la colinealidad es preservada por la transformación.

De manera explícita podemos representar la Ecuación 2.1.3 como:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (2.1.4)$$

Note que la matriz \mathbf{H} en está ecuación puede ser cambiada por un factor escala arbitraria diferente de cero sin alterar la transformación proyectiva, es decir; \mathbf{H} por $k\mathbf{H}$ para $k \neq 0$. Consecuentemente podemos decir que \mathbf{H} es una matriz *homogénea*, ya que como en la representación homogénea de un punto, sólo el radio de los elementos de la matriz es significativo.

Para una mayor perspectiva, podemos ejemplificar lo anterior calculando la matriz proyectiva entre dos planos paralelos como se muestra en la Figura 2.1.2.

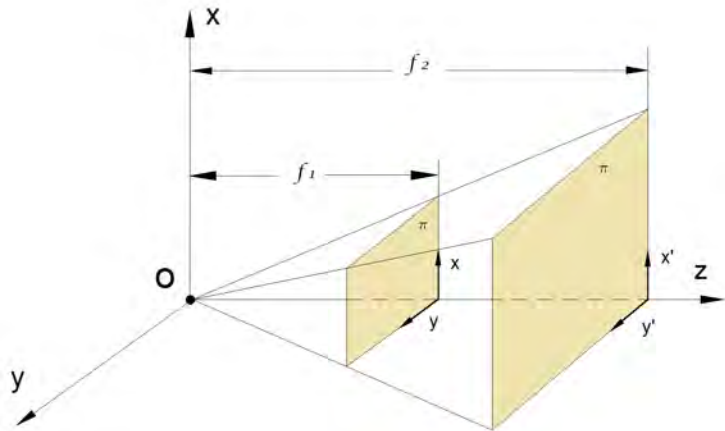


Figura 2.1.2: Proyección en planos paralelos

Utilizando el teorema de Tales, obtenemos las siguientes relaciones:

$$\frac{x'}{f_2} = \frac{x}{f_1} \quad ; \quad \frac{y'}{f_2} = \frac{y}{f_1}$$

Estas ecuaciones se pueden escribir usando la forma matricial $\mathbf{x}' = \mathbf{H}\mathbf{x}$, con $\mathbf{x} = [x \ y \ 1]^\top$, $\mathbf{x}' = [x' \ y' \ 1]^\top$ y

$$\mathbf{H} = \begin{bmatrix} \frac{f_2}{f_1} & 0 & 0 \\ 0 & \frac{f_2}{f_1} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Para el caso de dos planos no paralelos se pueden establecer dos características de la transformación proyectiva: *i)* hay una correspondencia biunívoca entre los puntos pertenecientes a ambos planos y *ii)* una línea recta en un plano corresponde a una línea recta en el otro plano. De esta manera se puede afirmar que la relación proyectiva entre ambos planos está dada por la ecuación general Ecuación 2.1.4. [14]

2.1.3. Rectificación de distorsión proyectiva

Buscamos la matriz \mathbf{H} que logre transformar una imagen que presente una distorsión proyectiva, en una imagen libre de distorsión.

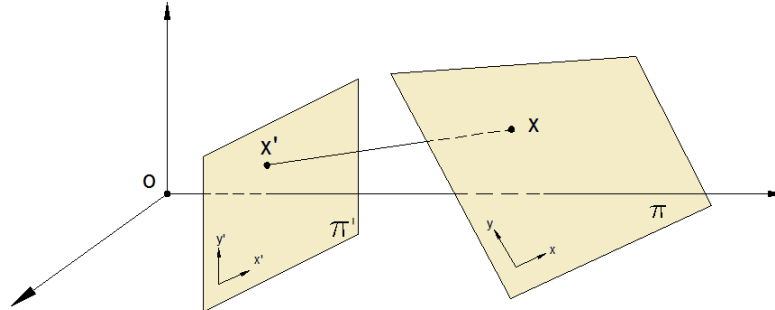


Figura 2.1.3: Proyección en dos planos no paralelos.

La operación de rectificación de distorsión se muestra en la Figura 2.1.3. Los datos de entrada en este problema son las coordenadas de n puntos (x'_i, y'_i) , $i = 1, 2, \dots, n$, en la imagen original y las coordenadas deseadas de estos puntos (x_i, y_i) en la imagen rectificada. Por lo general se escogen puntos (x'_i, y'_i) que pertenezcan a un rectángulo. Usamos las coordenadas no homogéneas aquí en lugar de coordenadas homogéneas de los puntos, porque son esas coordenadas no homogéneas que se miden directamente en

la imagen y desde el plano del mundo. A partir de la Ecuación 2.1.4 se obtiene para cada punto (x', y') y su correspondencia (x, y) las ecuaciones[14, 10]:

$$x' = \frac{x'_1}{x'_2} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \quad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}.$$

Cada punto correspondiente genera 2 ecuaciones para los elementos de \mathbf{H} , que después de multiplicarse queda

$$x'(h_{31}x + h_{32}y + h_{33}) = h_{11}x + h_{12}y + h_{13}$$

$$y'(h_{31}x + h_{32}y + h_{33}) = h_{21}x + h_{22}y + h_{23}.$$

Estas ecuaciones son lineales en los elementos de \mathbf{H} y lo que nos interesa es encontrar los 9 elementos de \mathbf{H} . Como la matriz \mathbf{H} es homogénea, $k\mathbf{H}$ también sería solución al problema. De esta manera es posible dividir cada elemento de \mathbf{H} por h_{33} para obtener una matriz \mathbf{H} con sólo 8 elementos desconocidos, ya que el último sería igual a uno.[14]

Entonces, con $h_{33} = 1$ las dos últimas ecuaciones pueden ser escritas de manera matricial como:

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (2.1.5)$$

o bien $\mathbf{AH} = \mathbf{b}$. Se observa que para cada correspondencia de puntos se obtienen dos ecuaciones. Suponiendo n pares de puntos correspondientes se puede establecer el siguiente sistema de $2n$ ecuaciones y 8 incógnitas.

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_n \end{bmatrix} \mathbf{H} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{bmatrix}$$

donde \mathbf{A}_i es el vector de puntos a transformar y el vector \mathbf{b} obtenido en la Ecuación 2.1.5 para el punto i . El sistema de ecuaciones anterior puede ser expresado como $\tilde{\mathbf{A}}\mathbf{H} = \mathbf{b}$. Para $n = 4$ existe una solución directa dada por $\mathbf{H} = \tilde{\mathbf{A}}^{-1}\mathbf{b}$. Si $n > 4$ sin embargo el sistema queda sobredeterminado. En este caso se utiliza el método de los mínimos cuadrados en el que se encuentra un vector \mathbf{H} tal que minimice $\|\tilde{\mathbf{A}}\mathbf{H} - \mathbf{b}\|$. La solución entonces estaría dada por $\mathbf{H} = [\tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}]^{-1} \tilde{\mathbf{A}}^\top \mathbf{b}$. [14]

Cabe señalar que el método aquí expuesto funciona sólo si $h_{33} \neq 0$. Para el caso en que $h_{33} = 0$ se recomienda un método alternativo en el que la restricción para \mathbf{H} no es $h_{33} = 1$ sino $\|\tilde{\mathbf{H}}\| = 1$ donde $\tilde{\mathbf{H}} = [h_{11} \ \dots \ h_{33}]^\top$. En la Figura 2.1.4 se muestra gráficamente una transformación projectiva.

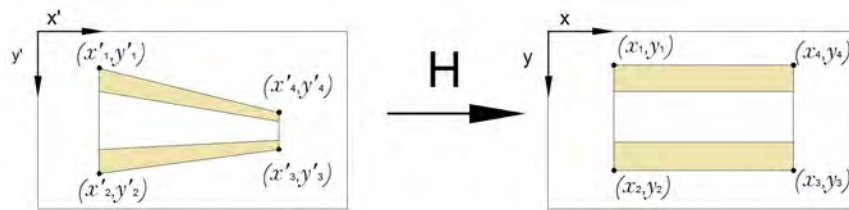


Figura 2.1.4: Removiendo distorsión de perspectiva.

2.1.4. Transformación Projectiva General

La transformación projectiva es la generalización de las transformaciones lineales $\mathcal{R}^2 \rightarrow \mathcal{R}^2$ en la que las líneas paralelas no son transformadas necesariamente como tales. La transformación se de nota como \mathbf{H}_p .

La invariante de esta transformada es la razón de cruz. [14]

Transformación	Matriz \mathbf{H}	Invariantes
Euclídea	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$	longitud entre puntos
Similitud	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$	ángulos entre rectas, razón entre dos distancias.
Afín	$\begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$	líneas paralelas, razón entre dos áreas.
General	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$	razón de cruz.

2.2. Modelo básico de cámara, punta de alfiler (*pinhole*)

La gran mayoría de los procedimientos de calibración se basan en el modelo de cámara de agujero o *pinhole*, que es el modelo más simple que se puede obtener de una cámara.

El modelo *pinhole* consiste en un centro óptico C , en donde convergen todos los rayos de la proyección, y un plano de imagen R en el cual la imagen es proyectada. El plano de la imagen está ubicado a una distancia focal f del centro óptico y perpendicular al eje óptico Z .

Consideremos la proyección central de puntos en el espacio dentro de un plano. Sea el centro de proyección el origen de un sistema de coordenadas Euclidiano, y considerar el plano $z = f$, el cual es llamado *plano imagen* o *plano focal*. Bajo el modelo de cámara *pinhole*, un punto en el espacio con coordenadas $\mathbf{X} = (X, Y, Z)^\top$ es asignado a el punto sobre el plano imagen, donde una la línea que une el punto \mathbf{X} al centro de proyección se encuentra con el plano imagen, tal como se muestra en la Figura 2.2.1.

Por triángulos semejantes, rápidamente calculamos que el punto $\mathbf{X} = (X, Y, Z)^\top$ es asignado al punto $\mathbf{X} = (fX/Z, fY/Z, Z)^\top$ sobre el plano imagen. Ignorando la coordenada de la imagen fina, observamos que

$$(X, Y, Z)^\top \rightarrow (fX/Z, fY/Z, Z)^\top \quad (2.2.1)$$

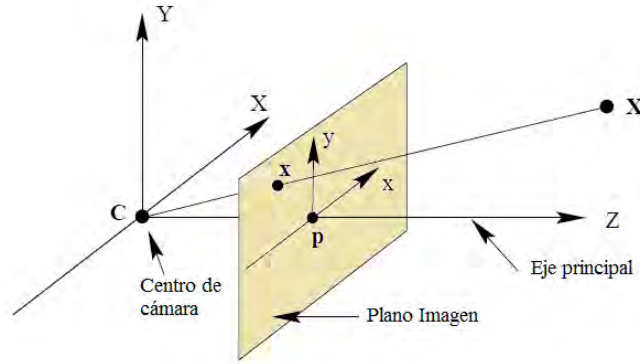


Figura 2.2.1: Modelo geométrico de la cámara *pinhole*.

describe la proyección central asignado desde el mundo a la coordenadas de la imagen. Esto es un mapeo desde un espacio Euclidiano \mathbb{R}^3 a un espacio Euclidiano \mathbb{R}^2 .

El centro de proyección es llamado el *centro de cámara*. También es conocido como el *centro óptico*. La línea desde el centro de cámara perpendicular al plano imagen es llamado *eje principal* o *rayo principal* de la cámara, y el punto donde el eje principal encuentra el plano imagen es llamado el *punto principal*. El plano a través del centro de cámara paralelo al plano imagen es llamado el *plano principal* de la cámara.

Proyección central en coordenadas homogéneas. Si el mundo y los puntos de la imagen son representados por vectores homogéneos, entonces la proyección central es expresado muy simple como un mapeo lineal entre sus coordenadas homogéneas. En particular, la Ecuación 2.2.1 puede ser escrita en términos de una multiplicación de matrices como[14, 41]

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & \\ & f & & \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.2.2)$$

Ahora introduciremos la \mathbf{X} para el punto del mundo representado por un vector homogéneo $\mathbf{X} = (X, Y, Z, 1)^\top$, X para el punto de la imagen representado por un 3-vector homogéneo y P para un vector 3×4 homogéneo *matriz de proyección de cámara*.

Entonces podemos escribir Ecuación 2.2.2 compáctamente como:

$$X = \mathbf{P}\mathbf{X}$$

el cual define la matriz de cámara para el modelo pinhole de proyección central.

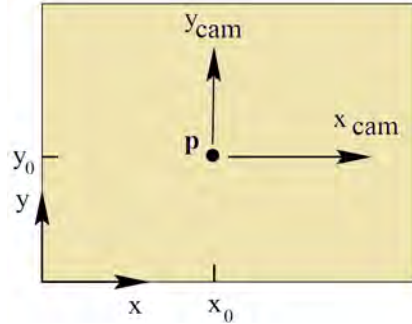


Figura 2.2.2: Sistema de coordenadas de la imagen (x, y) y cámara (x_{cam}, y_{cam}) .

Punto principal offset. La expresión Ecuación 2.2.1 asume que el origen de coordenadas en el plano imagen es el punto principal. En la práctica, puede no ser así, así que en general hay una asignación

$$(X, Y, Z)^T \rightarrow (fX/Z + p_x, fY/Z + p_y, Z)^T$$

donde $(p_x, p_y)^T$ son las coordenadas del punto principal. Esto se muestra de forma gráfica en la figura Figura 2.2.2 esta ecuación puede ser expresado convenientemente en coordenadas homogéneas como

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX + p_x \\ fY + p_y \\ Z \end{pmatrix} = \begin{bmatrix} f & p_x & 0 \\ f & p_y & 0 \\ & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.2.3)$$

Ahora, escribiendo

$$\mathbf{K} = \begin{bmatrix} f & p_x \\ f & p_y \\ & 10 \end{bmatrix} \quad (2.2.4)$$

entonces Ecuación 2.2.3 tiene la forma concisa

$$\mathbf{x} = \mathbf{K} [\mathbf{I}|0] \mathbf{X}_{\text{cam}}. \quad (2.2.5)$$

La matriz \mathbf{K} es llamado la *matriz de calibración de cámara*. En Ecuación 2.2.5 hemos escrito $(X, Y, Z, 1)^\top$ como \mathbf{X}_{cam} para enfatizar que la cámara se supone que se encuentra en el origen de un sistema de coordenadas euclidianas con el eje principal de la cámara apuntando hacia abajo del eje z, y el punto \mathbf{X}_{cam} es expresado en este sistema de coordenadas.

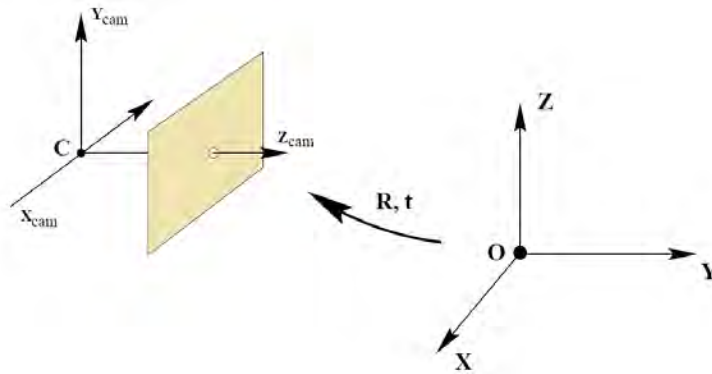


Figura 2.2.3: Transformación Euclidiana entre el mundo y el marco de coordenadas de la cámara.

Rotación y Traslación de cámara. En general, los puntos en el espacio se expresan en términos de un marco de coordenadas euclidianas diferente, conocido como el *marco de coordenadas del mundo*. Los dos marcos de coordenadas son relacionados a través de una rotación y una traslación. Ver Figura 2.2.3 Si $\tilde{\mathbf{X}}$ es un vector no homogéneo representado las coordenadas de un punto en el marco de coordenadas del mundo, y $\tilde{\mathbf{X}}_{\text{cam}}$ representa el mismo punto en la marco de coordenadas de la cámara, entonces puede escribirse $\tilde{\mathbf{X}}_{\text{cam}} = R(\tilde{\mathbf{X}} - \tilde{\mathbf{C}})$, donde $\tilde{\mathbf{C}}$ representa las coordenadas del centro de cámara en el marco de coordenadas del mundo, y R es una matriz de 3×3 representando la orientación del marco de coordenadas de la cámara. Esta ecuación puede ser escrita en coordenadas homogéneas como [14, 41]

$$\mathbf{X}_{cam} = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{X}. \quad (2.2.6)$$

Colocando junto con Ecuación 2.2.4 conlleva a la siguiente fórmula

$$\mathbf{x} = \mathbf{K}R [\mathbf{I} | -\tilde{\mathbf{C}}] \mathbf{X} \quad (2.2.7)$$

donde \mathbf{X} es ahora un marco de coordenadas del mundo. Esto es el mapeo general tomado por una cámara pinhole. Uno ve que una cámara pinhole, $P = \mathbf{K}R [\mathbf{I} | -\tilde{\mathbf{C}}]$, tiene 9 grados de libertad: 3 para K (los elementos f, p_x, p_y), 3 para R , y 3 para $\tilde{\mathbf{C}}$. Los parámetros contenidos en K son llamados parámetros de cámara *internos*, o la *orientación interna* de la cámara. Los parámetros de R y $\tilde{\mathbf{C}}$ que relacionan la orientación de la cámara y posición a un sistema de coordenadas del mundo son llamados parámetros *externos* o la *orientación externa*.

2.3. Calibración de cámara

Hablando de calibración de cámaras podemos encontrar dos grandes campos de aplicaciones como son la fotogrametría y la visión por computador.

Con respecto a visión por computador, la precisión métrica se excluye en beneficio de la automatización y la inmediatez del proceso de calibración, estas causas hacen que se extienda el uso de la autocalibración, en la que no se utiliza un objeto para calibrar expresamente la cámara, sino que este proceso se realiza aunado a las tareas de orientación y de restitución del objeto aprovechando la propia escena fotografiada o filmada aplicando restricciones a las condiciones de la captura.

Por esto mismo el concepto de calibración y los consiguientes tratamientos de los parámetros implicados varían según los puntos de vistas.

El uso de cámaras calibradas puede ser útil para resolver una serie de problemas relacionados con la obtención de la posición 3D de los objetos en el espacio a partir de sus imágenes o para su reconstrucción tridimensional. Esto puede permitir, entre

otras tareas, la realización de mapas del entorno de la cámara, el seguimiento de un objeto específico o la obtención de la posición de la cámara respecto a objetos que la rodeen. También puede facilitar la navegación por su entorno de un robot móvil, permitiendo evitar obstáculos, dirigirse a objetos determinados o facilitar la definición de la trayectoria más adecuada para alcanzar su destino.

Por calibración de cámara podríamos decir que es un proceso en que tratan de determinarse los parámetros geométricos y físicos como la distancia focal, coordenadas del punto principal, los parámetros de la distorsión radial y tangencial, de la toma de fotografías. Todo esto debido a la distorsión radial, a la falta de planaridad del plano imagen, la imposibilidad física de llegar a conocer el valor verdadero del plano focal de la toma que se calcula en función de la distorsión que produce, que también es desconocida y como resultado se obtienen los parámetros intrínsecos y extrínsecos de la cámara.

En esta sección se realiza la explicación de las técnicas del proceso de calibración de cámara empleado en el sistema descritas por Zhengyou Zhang [43].

El propuesta de la técnica de calibración sólo requiere de la cámara para observar un patrón de cuadros mostrado en diferentes orientaciones. El patrón puede ser impreso por una impresora láser y unido a una superficie plana razonable. Ya sea la cámara o el patrón plano, puede ser movido por la mano. El movimiento no necesita ser conocido.

El enfoque propuesto se encuentra entre la calibración fotogramétrica y la autocalibración, debido a que usamos la información métrica 2D en lugar de un 3D o implícita, en comparación con las técnicas clásicas, la técnica descrita por Zhang es bastante más flexible. En comparación con la auto-calibración, gana considerable grado de robustez.

2.3.1. Ecuaciones básicas

Se examina las limitaciones sobre los parámetros intrínsecos de la cámara provista por la observación de un solo plano.

Un punto 2D es denotado por $\mathbf{m} = [u, v]^\top$. Un punto 3D es denotado por $\mathbf{M} = [X, Y, Z]$. Usamos $\tilde{\mathbf{m}}$ para denotar el vector aumentado por la adición de 1 como el último elemento: $\tilde{\mathbf{m}} = [u, v, 1]^\top$ y $\tilde{\mathbf{M}} = [X, Y, Z]^\top$. Un modelo de cámara pinhole es usado: la relación entre un punto \mathbf{M} en 3D y su proyección \mathbf{m} está dado por

$$s\tilde{\mathbf{m}} = \mathbf{A}[\mathbf{R} \ \mathbf{t}]\tilde{\mathbf{M}} \quad (2.3.1)$$

donde s es un factor de escala arbitrario, (\mathbf{R}, \mathbf{t}) , son llamados parámetros extrínsecos, es la rotación y la traslación que relaciona el sistema de coordenadas del mundo al sistema de coordenadas de la cámara, y \mathbf{A} , es llamado la matriz intrínseca de la cámara, está dado por:

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ & \beta & v_0 \\ & & 1 \end{bmatrix}$$

con (u_0, v_0) las principales coordenadas del punto principal, α y β son los factores de escala en la imagen y u y el piso de lv son los ejes, y γ el parámetro que describe la asimetría de los dos ejes de la imagen.

Usaremos la abreviación $\mathbf{A}^{-\top}$ para $(\mathbf{A}^{-1})^{\top}$ o $(\mathbf{A}^{\top})^{-1}$.

Sin pérdida de generalidad, asumimos que el modelo del plano es $Z = 0$ del sistema de coordenadas mundo.

Indicamos la i – *ésima* columna de la matriz de rotación \mathbf{R} por \mathbf{r}_i . A partir de la Ecuación 2.3.1 tenemos:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}.$$

Por abuso de notación, aún usamos M para denotar un punto sobre el modelo del plano, pero $M = [X, Y]^{\top}$ ya que Z siempre es igual a 0. A su vez, $M = [\tilde{X}, Y, 1]^{\top}$. Entonces, un punto M del modelo y su imagen \mathbf{m} es relacionado por una homografía \mathbf{H} :

$$s\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathbf{M}} \quad \text{con} \quad \mathbf{H} = \mathbf{A} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \quad (2.3.2)$$

como es evidente la matriz \mathbf{H} de 3×3 es definida por un factor de escala.

2.3.2. Restricciones sobre los parámetros intrínsecos

Dada una imagen del modelo de un plano, una homografía puede ser estimada. Denotemos $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3]$. A partir de la Ecuación 2.3.2 tenemos[14]

$$[\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] = \lambda \mathbf{A} [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}],$$

donde λ es un escalar arbitrario. Sabiendo que \mathbf{r}_1 y \mathbf{r}_2 son ortonormales, tenemos

$$\mathbf{h}_1^\top \mathbf{A}^{-\top} \mathbf{A}^{-1} \mathbf{h}_2 = 0 \quad (2.3.3)$$

$$\mathbf{h}_1^\top \mathbf{A}^{-\top} \mathbf{A}^{-1} \mathbf{h}_2 = \mathbf{h}_2^\top \mathbf{A}^{-\top} \mathbf{A}^{-1} \mathbf{h}_1 = 0. \quad (2.3.4)$$

Estas son las dos restricciones básicas sobre los parámetros intrínsecos, dada una homografía. Dado a que una homografía tiene 8 grados de libertad y hay 6 parámetros extrínsecos (3 de rotación y 3 de traslación), sólo podemos obtener 2 restricciones sobre los parámetros intrínsecos. Note que $\mathbf{A}^{-\top} \mathbf{A}^{-1}$ actualmente describe la imagen del cónico absoluto.

2.3.3. Interpretación Geométrica

No es difícil verificar que el modelo del plano, bajo nuestra convención, se describe en el sistema de coordenadas de la cámara por la siguiente ecuación[14]:

$$\begin{bmatrix} \mathbf{r}_3 \\ \mathbf{r}_3^\top \mathbf{t} \end{bmatrix}^\top \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = 0,$$

donde $w = 0$ para puntos en el infinito y $w = 1$ para cualquier otro caso. Este plano intersecta al plano en el infinito en una línea y podemos fácilmente ver que $\begin{bmatrix} \mathbf{r}_1 \\ 0 \end{bmatrix}$ y $\begin{bmatrix} \mathbf{r}_2 \\ 0 \end{bmatrix}$ son dos puntos particulares sobre la línea. Algún punto sobre de esta línea es una

combinación lineal de estos dos puntos, por ejemplo:

$$\mathbf{x}_\infty = a \begin{bmatrix} \mathbf{r}_1 \\ 0 \end{bmatrix} + b \begin{bmatrix} \mathbf{r}_2 \\ 0 \end{bmatrix} = \begin{bmatrix} a\mathbf{r}_1 + b\mathbf{r}_2 \\ 0 \end{bmatrix}.$$

Ahora, calculamos la intersección de la línea anterior con la cónica absoluta. Por definición, el punto \mathbf{x}_∞ , conocido como el *punto circular*, satisface: $\mathbf{x}_\infty^\top \mathbf{x}_\infty = 0$, por ejemplo,

$$(a\mathbf{r}_1 + b\mathbf{r}_2)^\top (a\mathbf{r}_1 + b\mathbf{r}_2) = 0 \quad \text{o} \quad a^2 + b^2 = 0.$$

La solución es $b = \pm ai$, donde $i^2 = -1$. Es decir, los dos puntos de intersección son

$$\mathbf{x}_\infty = a \begin{bmatrix} \mathbf{r}_1 \pm i\mathbf{r}_2 \\ 0 \end{bmatrix}.$$

Su proyección en el plano de la imagen se le da, hasta un factor de escalar, por

$$\mathbf{m}_\infty = \mathbf{A}(\mathbf{r}_1 \pm i\mathbf{r}_2) = \mathbf{h}_1 + i\mathbf{h}_2.$$

El punto \mathbf{m}_∞ está sobre la imagen de el cónico absoluto, descrito por $\mathbf{A}^{-\top} \mathbf{A}^{-1}$. Esto nos da

$$(\mathbf{h}_1 + i\mathbf{h}_2)^\top \mathbf{A}^{-\top} \mathbf{A}^{-1} (\mathbf{h}_1 + i\mathbf{h}_2) = 0.$$

Exige que tanto partes real e imaginaria sean cero, resultado de la Ecuación 2.3.3 y Ecuación 2.3.4.

2.3.4. Solución de la Calibración de cámara

Comenzaremos con una solución analítica, seguido por una técnica de optimización no lineal basada en un criterio de máxima probabilidad. Finalmente, se tomará en cuenta la distorsión de las lentes, dando ambas soluciones analíticas y no lineales[42].

- Solución de Forma Cerrada.

Sea

$$\mathbf{B} = \mathbf{A}^{-\top} \mathbf{A}^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{c}{\alpha^2\beta} & \frac{cv_0 - u_0\beta}{\alpha^2\beta} \\ -\frac{c}{\alpha^2\beta} & \frac{c}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{c(cv_0 - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{cv_0 - u_0\beta}{\alpha^2\beta} & -\frac{c(cv_0 - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(cv_0 - u_0\beta)^2}{\alpha^2\beta^2} - \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}. \quad (2.3.5)$$

Note que \mathbf{B} es simétrico, definido por un vector 6D.

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^\top. \quad (2.3.6)$$

Deje que la i -ésima vector columna de \mathbf{H} sea $\mathbf{h}_i = [h_{i1}, h_{i2}, h_{i3}]^\top$.

Entonces tenemos

$$\mathbf{h}_i^\top \mathbf{B} \mathbf{h}_j = \mathbf{v}_{ij}^\top \mathbf{b} \quad (2.3.7)$$

con $\mathbf{v}_{ij} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^\top$.

Por lo tanto, las dos restricciones fundamentales Ecuación 2.3.3 y Ecuación 2.3.4, a partir de una homografía dada, puede ser escrita como 2 ecuaciones homogéneas en \mathbf{b} :

$$\begin{bmatrix} \mathbf{v}_{12}^\top \\ (\mathbf{v}_{11} - \mathbf{v}_{22}) \end{bmatrix} = \mathbf{b} = 0. \quad (2.3.8)$$

Si n imágenes del modelo plano son observados, apilando n ecuaciones como la Ecuación 2.3.8 tenemos

$$\mathbf{V} \mathbf{b} = 0, \quad (2.3.9)$$

donde \mathbf{V} es una matriz $2n \times 6$. Si $n \geq 3$, tendremos en general solución única \mathbf{b} definida hasta el factor de escala. Si $n = 2$, podemos imponer la restricción asimétrica $c = 0$, por ejemplo; $[0, 1, 0, 0, 0, 0] \mathbf{b} = 0$, que es añadido como una ecuación adicional a la Ecuación 2.3.9. La solución de la Ecuación 2.3.9 es bien conocida como el eigenvector de $\mathbf{V}^\top \mathbf{V}$ asociado con el más pequeño eigenvalor.

Una vez que \mathbf{b} es estimado, podemos calcular la matriz intrínseca \mathbf{A} . Una vez conocida \mathbf{A} , los parámetros extrínsecos para cada imagen es fácilmente calculado. A partir de la Ecuación 2.3.1 tenemos:

$$\mathbf{r}_1 = \lambda \mathbf{A}^{-1} \mathbf{h}_1$$

$$\mathbf{r}_2 = \lambda \mathbf{A}^{-1} \mathbf{h}_2$$

$$\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$$

$$\mathbf{t} = \lambda \mathbf{A}^{-1} \mathbf{h}_3$$

con $\lambda = 1 / \|\mathbf{A}^{-1} \mathbf{h}_1\| = 1 / \|\mathbf{A}^{-1} \mathbf{h}_2\|$. A causa del ruido en un dato, la matriz calculada $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]$ no cumple con las propiedades generales de la matriz de rotación.

■ **Estimación de máxima probabilidad.**

La solución anterior se obtiene a través de minimizar una distancia algebraica que no es físicamente significativa. Podemos refinarla a través de la inferencia de máxima probabilidad. Se nos ha dado n imágenes de un modelo del plano y hay m puntos en el modelo del plano. Supongamos que los puntos de la imagen están dañados por el ruido independientes e idénticamente distribuidos. La estimación de máxima probabilidad se pueden obtener al minimizar el siguiente funcional:

$$\sum_{i=1}^n \sum_{j=1}^m \|\mathbf{m}_{ij} - \hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i \mathbf{M}_j)\|^2, \quad (2.3.10)$$

donde $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i \mathbf{M}_j)$ es la proyección del punto \mathbf{M}_j en la imagen i , acordando la Ecuación 2.3.2. Una rotación \mathbf{R} es parametrizada por un vector de 3 parámetros, denotado por \mathbf{r} , que es paralela al eje de rotación y cuya magnitud es igual al ángulo de rotación. \mathbf{R} y \mathbf{r} son relacionados por la fórmula de Rodrigues [10]. Minimizando la Ecuación 2.3.10 es un problema de minimización no lineal, el cual es resuelto con el Algoritmo Levenber-Marquardt como se aplica en Minipack [24]. Se requiere una estimación inicial de \mathbf{A} , $\{\mathbf{R}_i, \mathbf{t}_i | i = 1 \dots n\}$ que se puede obtener mediante la técnica me en la subsección anterior.

■ **Distorsión Radial**

Hasta ahora no hemos considerado distorsión de la lente de una cámara. Sin embargo, una cámara de escritorio usualmente presenta la distorsión del objeto importante, especialmente lentes de distorsión radial. En esta sección, solamente consideramos los dos primeros términos de la distorsión radial. El lector puede ver las referencias [36, 3, 9, 15] para modelos más complicados. Basado en los reportes de la literatura [3, 38, 40], es observado que la función de distorsión es totalmente dominada por los componentes radiales, y especialmente dominada por el primer término. También se ha encontrado que los modelos más elaborados no sólo no ayudarían (insignificante en comparación con cuantificación del sensor), sino también podrían causar una inestabilidad numérica [38, 40]. Sea (u, v) el ideal (no observables sin distorsión) las coordenadas pixel de la imagen y (\check{u}, \check{v}) las coordenadas a la imagen real observada. Los puntos ideales son la proyección de los puntos del modelo de acuerdo con el modelo *pinhole*. Similarmente (x, y) and (\check{x}, \check{y}) son los ideales (sin distorsión) y real (distorsionada) de la imagen normalizada de coordenadas. Tenemos [3, 40]

$$\begin{aligned}\check{x} &= x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \check{y} &= y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2],\end{aligned}$$

donde k_1 y k_2 son los coeficientes de la distorsión radial. El centro de la distorsión radial es el mismo punto principal. A partir de $\check{u} = u_0 + \alpha\check{x} + \gamma\check{y}$ y $\check{v} = v_0 + \beta\check{y}$ y asumiendo que $\gamma = 0$, tenemos

$$\check{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (2.3.11)$$

$$\check{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]. \quad (2.3.12)$$

■ **Estimación de la distorsión radial por la alternancia.**

A medida que la distorsión radial se espera que sea pequeña, una se espera para estimar los otros cinco parámetros intrínsecos, mediante la técnica descrita en el segundo punto de esta sección, razonable y por hacer caso omiso de la distorsión. Una estrategia es entonces para calcular k_1 y k_2 , después de haber estimado los otros parámetros, lo que nos dará las coordenadas en píxeles ideales (u, v) . Entonces, de la Ecuación 2.3.11 y Ecuación 2.3.12, tenemos dos ecuaciones para cada punto en cada imagen:

$$\begin{bmatrix} (u - u_0)(x^2 + y^2) & (u - u_0)(x^2 + y^2)^2 \\ (v - v_0)(x^2 + y^2) & (v - v_0)(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \check{u} - u \\ \check{v} - v \end{bmatrix}.$$

Dado m puntos en n imágenes, podemos apilar todas las ecuaciones juntas para obtener un total de $2mn$ ecuaciones, o en forma matricial $\mathbf{D}\mathbf{k} = \mathbf{d}$, donde $\mathbf{k} = [k_1, k_2]^\top$. La solución lineal de mínimos cuadrados está dado por

$$\mathbf{k} = (\mathbf{D}^\top \mathbf{D})^{-1} \mathbf{D}^\top \mathbf{d}.$$

Una vez que k_1 y k_2 son estimados, uno puede refinar la estimación de los otros parámetros por solución de la Ecuación 2.3.10 con $\hat{\mathbf{m}}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, \mathbf{M}_j)$ remplazado por la Ecuación 2.3.11 y Ecuación 2.3.12. Podemos alternar estos procedimientos hasta la convergencia.

3 Diseño del Robot Móvil y Sistema de Control Remoto

En este trabajo de tesis, se propone el diseño y construcción de una plataforma robótica móvil de bajo costo, con características como se muestran en la Figura 3.0.1, adecuado para entornos estáticos para el uso general de algoritmos de navegación y control. Este robot móvil tiene la capacidad de comunicación con un computador a través de radiofrecuencia y está impulsado por un motores paso a paso y otro para dar dirección.

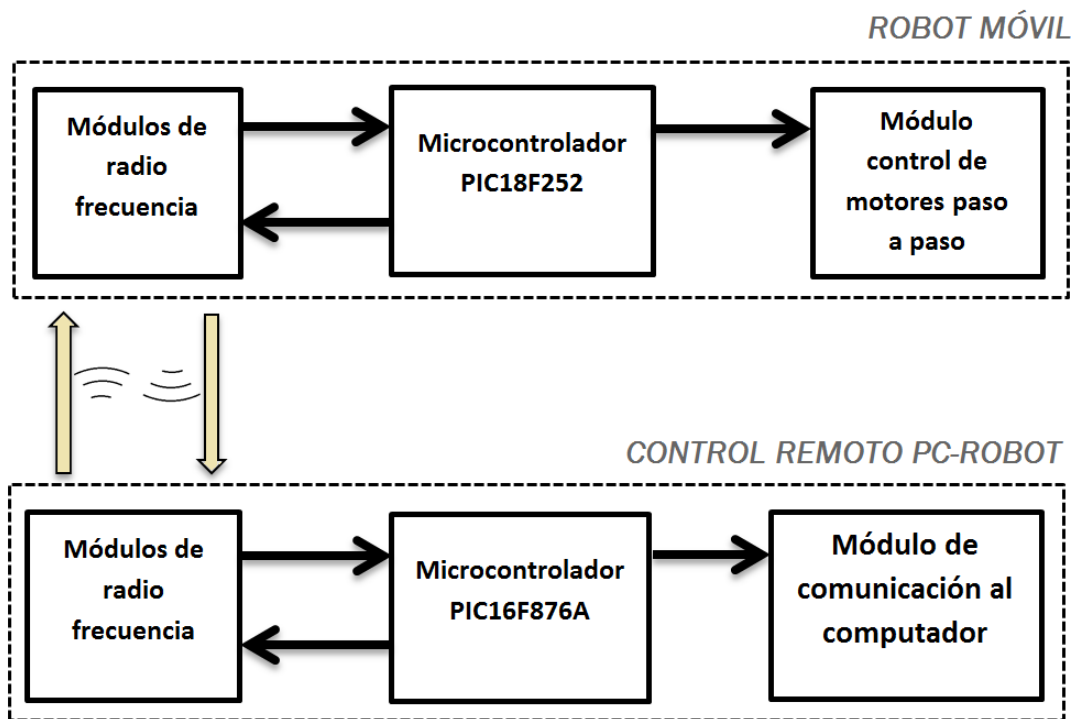


Figura 3.0.1: Diagrama a bloques general del robot móvil y control remoto.

El prototipo se diseñó de forma modular por 3 subsistemas fundamentales.

- **Subsistema de Procesamiento.** Se encarga de controlar y ejecutar todas las órdenes provenientes del computador para el robot móvil a través del subsistema de comunicación validando las instrucciones recibidas, control de los motores o ejecución de conjunto de instrucciones.
- **Subsistema de comunicación.** Permite la interacción con el robot móvil a través de un protocolo de comunicación y una interfaz RS-232 para conectarse de forma serial con el computador y de este al robot móvil a través de módulos de comunicación inalámbrica por radio frecuencia, permitiendo la comunicación entre el computador y el robot a una velocidad de 9600 baudios.
- **Subsistema actudador.** Las rueda de tracción del robot móvil y su eje de dirección son accionadas por motores paso a paso, acoplado mecánicamente y directamente sobre estas. La potencia eléctrica suministrada a los motores está controlada mediante la técnica de arreglos Darlington.

3.1. Descripción de los principales componentes

3.1.1. El microcontrolador

Un microcontrolador es un computador digital en un sólo chip que cuenta con un microprocesador o unidad central de proceso (CPU), una memoria para almacenar el programa, una memoria para almacenar datos y puertos de entrada y salida. Este dispositivo se destina a gobernar una sola tarea que no se puede modificar. Los microcontroladores disponen de los bloques esenciales: CPU, memorias de datos (RAM) y de programa (ROM), reloj, periféricos de entradas/salidas, etc.

El funcionamiento de los microcontroladores está determinado por el programa almacenado en su memoria. Este puede escribirse en distintos lenguajes de programación, además; de que pueden reprogramarse repetidas veces. Una vez programado y configurado, el microcontrolador solamente sirve para gobernar la tarea asignada.

El uso de estos dispositivos nos permite elegir el número de líneas de entrada y salida, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, usos de los módulos con los que cuente, entre otros.

Existe una gran diversidad de modelos de microcontroladores, estos se diferencian por varias características de acuerdo a las necesidades, como módulos de conversión analógico - digital, módulos de comunicación, cantidad de memoria RAM y ROM, número de puertos de comunicación, comparadores, entre otros más.

Con el uso de microcontroladores podemos encontrar *ventajas* como:

- **Prestación:** La alta integración de subsistemas que componen a un microcontrolador reduce el número de chips, la cantidad de pistas y espacio que requeriría en un circuito impreso si se implementase en un sistema equivalente usando chips separados.
- **Fiabilidad:** La sustitución de una gran cantidad de componentes a un sólo circuito programable disminuye el riesgo de fallas y precisa de menos calibraciones.
- **Flexibilidad:** La modificación del modo de funcionamiento se reduce a cambios en la programación del dispositivo.

3.1.1.1. Características de los microcontroladores

Los microcontroladores disponen de características principales como:

- **Unidad de Proceso Centra (CPU).**
Típicamente de 8 bits, pero también las hay de 4, 32 y hasta 64 bits con arquitectura Harvard, con memoria/bus de datos separada de la memoria/bus de instrucciones de programa, o arquitectura de von Neumann, también llamada arquitectura Princeton, con memoria/bus de datos y memoria/ bus de programa compartidas
- **Memoria de Programa.**
Es una memoria ROM (Read-Only Memory), EPROM (Electrically Programmable ROM), EEPROM (Electrically Erasable/Programmable ROM) o Flash que almacena el código del programa que típicamente puede ser de 1 kilobyte a varios megabytes.
- **Memoria de Datos.**
Es una memoria RAM (Random Access Memory) que típicamente puede ser de 1, 2, 4, 8, 16, 32 kilobytes.

- **Generador de reloj.**

Usualmente un cristal de cuarzo de frecuencias que genera una señal oscilatoria de entre 1 a 40 MHz, o también resonadores o circuitos RC.

- **Interfaz de Entrada/Salida.**

Puertos paralelos, seriales (UARTs, Universal Asynchronous Receiver/Transmitter), I2C (Inter-Integrated Circuit), Interfaces de Periféricos 4 Seriales (SPIs, Serial Peripheral Interfaces), Red de Área de Controladores (CAN, Controller Area Network), USB (Universal Serial Bus).

- **Otras características.**

Convertor Analógico-Digital (A/D, analog-to-digital) para convertir un nivel de voltaje en un cierto pin de un puerto de entrada a un valor digital manipulable digital por el programa del microcontrolador. Moduladores por Ancho de Pulso (PWM, Pulse-Width Modulation) para generar ondas cuadradas de frecuencia fija pero con ancho de pulso modificable.

Los microcontroladores usados en este trabajo es el PIC18F252 y el PIC16F876A, algunas de sus principales características se mencionan en la Tabla 3.1.

CARACTERÍSTICAS	PIC18F252	PIC16F876A
Frecuencia de Operación	DC - 40 MHz	DC - 20 MHz
Memoria de programa (Bytes)	32K	8K
Memoria de programa (Instrucciones)	16384	4096
Memoria de Datos (Bytes)	1536	368
Memoria de datos EEPROM(Bytes)	256	256
Fuentes de interrupción	17	14
Puertos Entrada/Salida	Puertos A, B, C	Puertos A, B, C
Temporizadores	4	3
Módulos Captura/Comparación/PWM	2	2
Comunicación Serial	MSSP(I2C,SPI), USART	MSSP(I2C,SPI), USART
Convertor Analógico - Digital 10 bits	5 Canales de entrada	5 Canales de entrada
Conjunto de instrucciones	75 Instrucciones	35 Instrucciones

Cuadro 3.1: Tabla de las principales características del PIC18f252.

3.1.2. Módulos de radiofrecuencia

Este módulo de radio frecuencia de muy bajo costo permitirá la comunicación entre dos dispositivos, a través transmisores (TWS 434/333) y receptores (RWS 434/333) de una sola vía. Por lo que se ha optado por usar dos módulos de transmisión y recepción en distintas frecuencias(433 MHz y 333 Mhz) por modulación de Amplitud (AM). Para estos dispositivos no es necesario el uso de licencias de radio frecuencias para su funcionamiento, por lo que no infringimos ningún tipo de norma legal.

3.1.2.1. Módulo de Transmisión

Los módulos de transmisión usados trabajo llamados TWS434 y TWS333 (se muestra en la Figura 3.1.1) permiten la transmisión de datos en serie de una forma sencilla, tiene una potencia de salida de hasta 8 mW, alcanzando distancias de hasta 100 m aproximadamente en espacios abiertos y 40 m en espacios cerrado. Este tipo de transmisor maneja señales digitales y lineales de entrada y opera con un voltaje de 2v - 12v de corriente continua. La Tabla 3.2 muestra las principales especificaciones del módulo.



Figura 3.1.1: TWS 433 - TWS333. Los dos componentes de transmisión comparten la misma apariencia y configuración de pines.

3.1.2.2. Módulo de Recepción

Los módulos de recepción usados en este trabajo llamados RWS434 y RWS333(se muestra en la Figura 3.1.2) permiten recepción de datos en serie, mantienen una sensibilidad de 3 uV, operan con un voltaje de alimentación de 3v - 5.5v de corriente

Símbolo	Características	Condiciones	Mín.	Típ.	Máx	Unidad
Vcc	Voltaje de alimentación		2.0	-	12.0	V
Icc	Corriente máxima		-	1.64/19.4	-	mA
Vh	Voltaje máximo entrada	Idato=100µA (alto)	Vcc-0.5	Vcc	Vcc+0.5	V
Vl	Voltaje mínimo entrada	Idato=0µA (bajo)	-	-	0.3	V
F0	Frecuencia de Operación		433.90/333.90	433.92/433.92	433.94/333.94	MHz
Pout	Potencia RF sobre 50 ohm	Vcc = 5 a 6V	-	14	-	dBm
Tbw	Ancho banda modulación	Codificación externa	-	5	-	Khz
Tr	Flanco subida modulación		-	-	100	uS
Tf	Flanco bajada modulación		-	-	100	uS
Dr	Velocidad de datos		-	2.4K	3K	Bps

Cuadro 3.2: Especificaciones de los módulos TWS433/TWS333

continua, también posee una salida digital y una lineal. La principales especificaciones de los módulos se muestran en la Tabla 3.3.



- Pin 1: Gnd
- Pin 2: Salida Digital
- Pin 3: Salida Lineal
- Pin 4: Vcc
- Pin 5: Vcc
- Pin 6: Gnd
- Pin 7: Gnd
- Pin 8: Ant

Figura 3.1.2: RWS433 - RWS333. Los dos componentes de transmisión comparten la misma apariencia y configuración de pines.

3.1.3. Conversor TTL-RS232 (MAX232)

El circuito integrado MAX232 usado en este trabajo para permitir la comunicación entre el tipo de señal del computador y el microcontrolador, dispone internamente de 4 conversores de niveles TTL al bus standard RS232 y viceversa, para comunicación serial.

Este circuito integrado dispone internamente de 2 conversores de nivel de TTL a RS232 y otros 2 de RS232 a TTL con lo que en total se podría manejar 4 señales del puerto

3.1 Descripción de los principales componentes

Símbolo	Características	Condiciones	Mín.	Típ.	Máx	Unidad
Vcc	Voltaje de alimentación		3	5	5.5	V
Itot	Corriente de operación		-	3.5	4.5	mA
Vdato	Salida datos	Idato=+200 μ A (alto)	Vcc - 0.5	-	Vcc	V
		Idato=-10 μ A (bajo)	-	-	0.3	V
Fc	Frecuencia operación		-	433.92/333.92	-	MHz
Pref	Sensibilidad				-106	dBm
	Ancho de canal		\pm 500			kHz
NEB	Ancho banda equival. ruido		-	5	4	kHz
Dr	Velocidad transferencia datos				3	kb/s
	Tiempo de encendido		-	-	5	mS

Cuadro 3.3: Especificaciones de los módulos RWS433/RWS333

serie del PC, por lo general las más usadas son: TX (para transmisión), RX (para recepción). Para el funcionamiento correcto del MAX232, se le conecta dos condensadores externos como se muestra en la Figura 3.1.3 en la que solo se han cableado las líneas TX y RX que son las más usadas para casi cualquier aplicación.

3.1.4. Motores paso a paso

Los motores paso a paso son máquinas eléctricas capaces de transformar impulsos eléctricos en giros o desplazamientos finitos.

Los motores paso a paso son ideales para la construcción de mecanismos donde se requieren movimientos muy precisos. La principal característica de estos es el de poder moverlos un paso a la vez para cada pulso que se le apliqué. El giro o paso puede variar desde 0.72° hasta pasos de 90° [23].

Existen tres tipos de motores paso a paso de imán permanente, los unipolares, bipolares y multifase, las diferencias entre estas son el tipo de bobinado, nos concentraremos en los motores usados en este trabajo que son los unipolares.

Relativamente los motores unipolares son fáciles de controlar debido a que poseen devanados duplicados c. Aunque para facilitar el esquema se dibuja este devanado como una bobina con punto medio, en realidad tienen dos bobinas en cada eje del estator, que están unidas por extremos opuestos, de tal modo que al ser alimentada una u otra, generan cada una un campo magnético inverso al de la otra. Nunca se

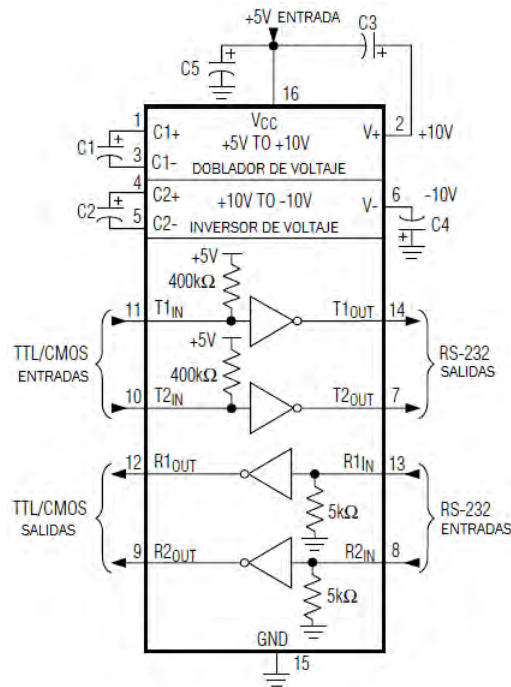


Figura 3.1.3: Diagrama de funcionamiento del circuito integrado MAX232.

energizan juntas: por eso lo correcto es decir que tienen una doble bobina, en lugar de decir (como se hace habitualmente) que es una bobina con punto medio. Esta duplicación se hace para facilitar el diseño del circuito de manejo, ya que permite el uso en la parte de potencia de un transistor único por cada uno de los bobinados.

En el esquema más común de conexión se unen los puntos medios de ambos ejes (a y b en la Figura 3.1.5) y se les conecta al positivo de la alimentación del motor. El circuito de control de potencia, entonces, se limita a poner a masa los bobinados de manera secuencial[23].

3.1.4.1. Características comunes de los motores paso a paso

Las características básicas y más comunes de un motor paso a paso son la siguientes:

- **Voltaje.** Los motores paso a paso tienen una tensión eléctrica de trabajo. Este valor viene impreso en su carcasa o por lo menos se especifica en su hoja de datos. Algunas veces puede ser necesario aplicar un voltaje superior para lograr que un determinado motor cumpla con el torque deseado, pero esto producirá

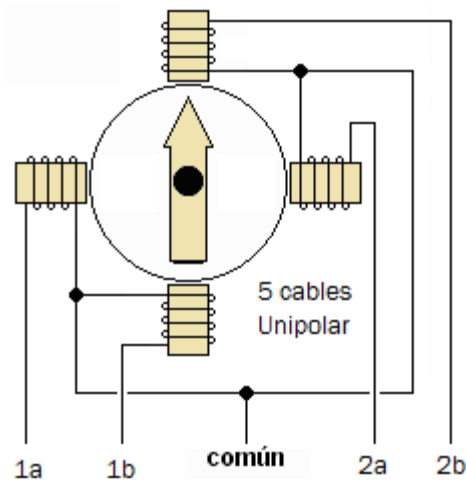


Figura 3.1.4: Distribución del bobinado de un motor unipolar.

un calentamiento excesivo y/o acortará la vida útil del motor.

- **Resistencia eléctrica.** La resistencia eléctrica de los bobinados determinará la corriente que consumirá el motor, y su valor afecta la curva de torque del motor y su velocidad máxima de operación.
- **Grados por paso.** Generalmente, este es el factor más importante al elegir un motor paso a paso para un uso determinado. Este factor define la cantidad de grados que rotará el eje para cada paso completo. Una operación de medio-paso o semi-paso (half step) del motor duplicará la cantidad de pasos por revolución al reducir la cantidad de grados por paso. Cuando el valor de grados por paso no está indicado en el motor, es posible contar a mano la cantidad de pasos por vuelta, haciendo girar el motor y sintiendo por el tacto cada "diente" magnético. Los grados por paso se calculan dividiendo 360 (una vuelta completa) por la cantidad de pasos que se contaron. Las cantidades más comunes de grados por paso son: $0,72^\circ$, $1,8^\circ$, $3,6^\circ$, $7,5^\circ$, 15° y hasta 90° . A este valor de grados por paso usualmente se le llama la resolución del motor. En el caso de que un motor no indique los grados por paso en su carcasa, pero sí la cantidad de pasos por revolución, al dividir 360 por ese valor se obtiene la cantidad de grados por paso. Un motor de 200 pasos por vuelta, por ejemplo, tendrá una resolución de $1,8^\circ$ por paso.

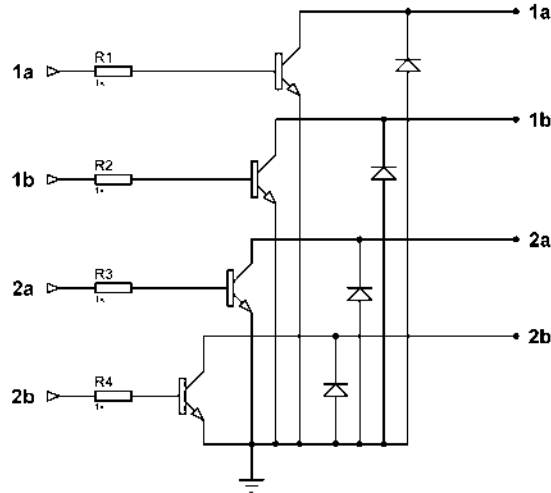


Figura 3.1.5: Circuito común para controlar un motor unipolar.

3.2. Subsistema de Procesamiento del robot móvil

Este es el principal subsistema del robot móvil, basado en un microcontrolador de 8 bits, el PIC18F252; se encarga de controlar y ejecutar todas las órdenes provenientes del computador para el robot móvil a través del subsistema de comunicación validando las instrucciones recibidas, control de los motores o ejecución de conjunto de instrucciones. Es importante recalcar que el robot está desprovisto de sensores para fines de uso de este sistema, sin embargo; está provisto por un bus I^2C (Inter-Integrated Circuit), lo que facilita la adaptación de nuevos dispositivos en el robot móvil como lo es el subsistema actuador.

En la Figura 3.2.1, se muestra el circuito electrónico del microcontrolador y las referencias de los puertos a los distintos circuitos.

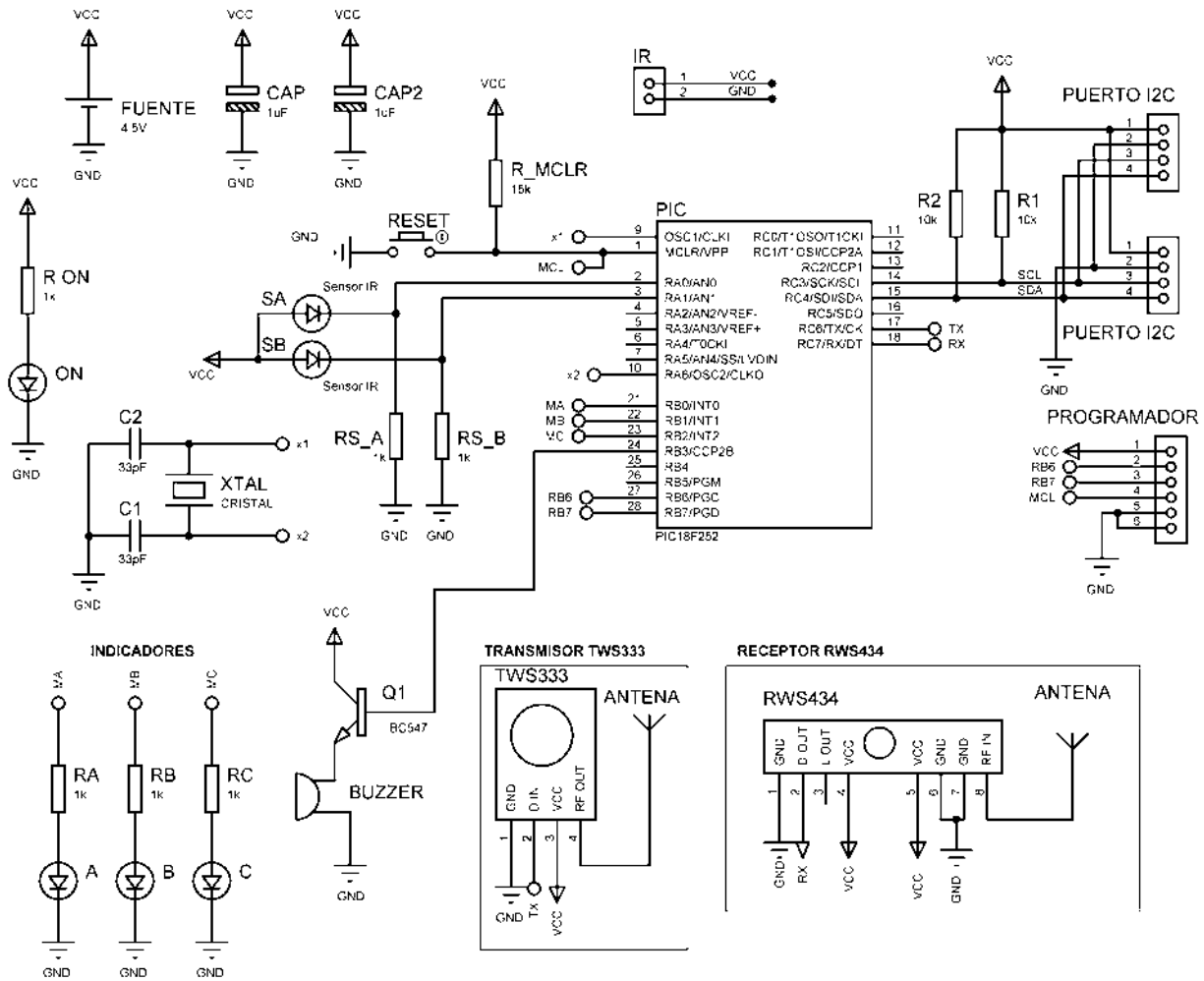


Figura 3.2.1: Diagrama electrónico del subsistema de procesamiento de información del Robot Móvil.

3.3. Subsistema actuador

Este subsistema actuador como se muestran en el diagrama electrónico de la Figura 3.3.1 se encarga de darle potencia a los motores que le dan movimiento y dirección al robot móvil. El subsistema es gobernado por el subsistema de procesamiento a través de un bus I^2C . Cuenta con una conexión de 4 terminales que se conecta a otra igual del subsistema de procesamiento que le provee de fuente de alimentación y comunicación. El subsistema de procesamiento rige el control de los los puertos expansores de 8 bits

MCP23008 del subsistema actuador para los motores a través del bus I^2C permitiendo expandir los puertos de comunicación hasta un máximo de 8 componentes del mismo tipo, es decir; usando el circuito integrado MCP23008 podemos obtener una combinación de hasta 64 terminales de entradas o salidas con velocidades de comunicación desde 100 KHz hasta 1 MHz, lo que permitirá su alta expansibilidad para otros módulos de interés.

La parte de potencia para los motores paso a paso es provisto por el circuito ULN2804 que es un arreglo driver que incrementan la corriente de salida para cada uno en 500 mA, corriente esencial para obtener un buen torque en los motores paso a paso.

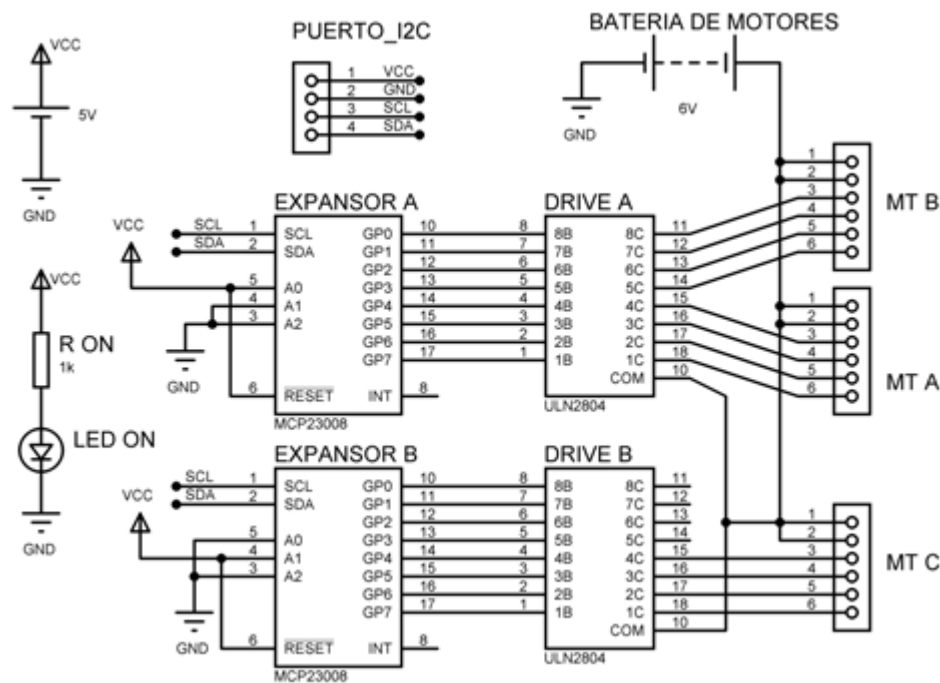


Figura 3.3.1: Diagrama electrónico del subsistema actuador para los motores paso a paso.

3.4. Subsistema de comunicación

Este subsistema de comunicación permite la interacción entre el computador y el robot móvil. La comunicación se realiza a través del puerto serial del computador, si bien

3.4 Subsistema de comunicación

este ya es obsoleto, aún se pueden conseguir dispositivos de este tipo con conexión USB. La razón de usar el tipo de señales RS-232 de un puerto serial es para aprovechar el uso de los módulos de radio frecuencia pueden manejar este tipo velocidades y que también el microcontrolador posee este tipo de módulo de comunicación incorporado.

Los niveles de señales RS-232 son incompatibles con los niveles que un microcontrolador que son los niveles lógicos TTL (transistor-transistor logic), por lo que usamos el convertidor MAX232 para adaptar los tipos de señales.

La función del microcontrolador PIC16F876A es la de recibir la comunicación proveniente del computador y encriptarla o blindarla para enviarla por radio frecuencia y así evitar perdidas de comunicación validando estas.

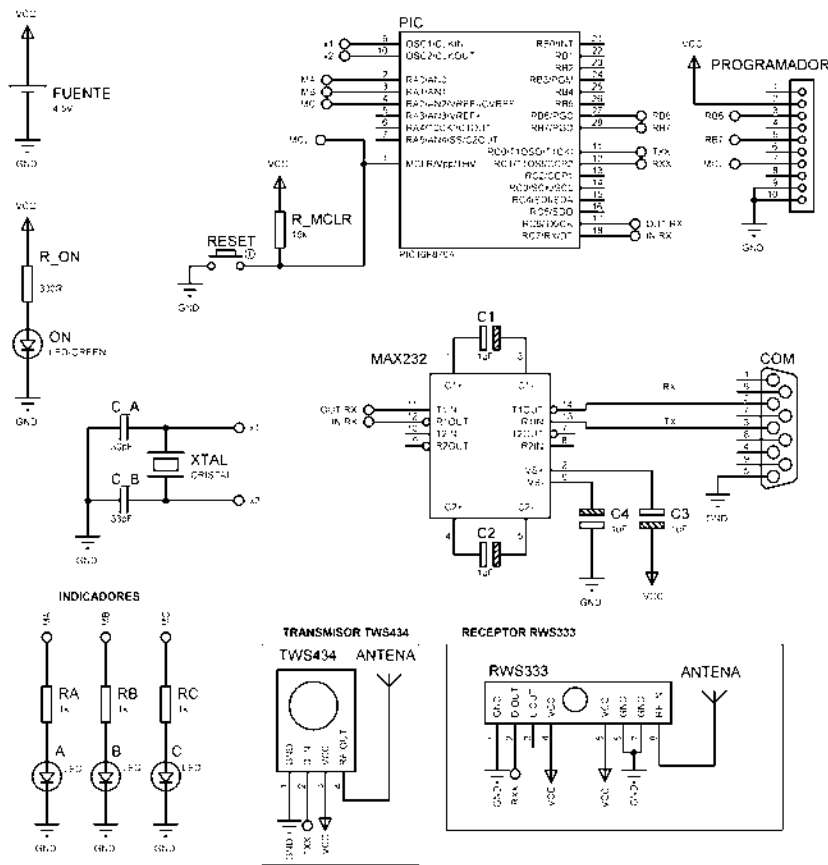


Figura 3.4.1: Diagrama electrónico del subsistema de comunicación.

3.4.1. Envío y recepción

Para el envío y recepción de datos al robot móvil es necesario cerciorarnos de que los datos enviados sean recibidos correctamente. Para esto hemos implementado un algoritmo que asegure que el dato e instrucción enviado sean válidas en su totalidad. El funcionamiento se observa en el diagrama a bloques de la Figura 3.4.2. El primer dato se envía desde el ordenador hacia el microcontrolador del control remoto a través de un transceptor de niveles lógicos RS232-TTL. El dato recibido por el microcontrolador es preparado para ser enviado por RF, usando un algoritmo de suma (checksum) el cual contiene el número contador de instrucciones, la instrucción que debe ejecutar el robot móvil, el parámetro o dato y la suma de validación.

Una vez ya recibida la instrucción por el subsistema de comunicaciones del robot móvil, se comparan los datos con la suma de validación. En caso de ser correctos los datos se envía confirmación de reconocimiento. El robot móvil tiene un límite de espera de 15 milisegundos, en caso de no recibir el reconocimiento se vuelve a enviar el conjunto de datos hasta recibir dicho reconocimiento. El robot no puede ejecutar instrucciones con el mismo número del contador de instrucción, con esto blindamos el funcionamiento de envío y recepción de datos.

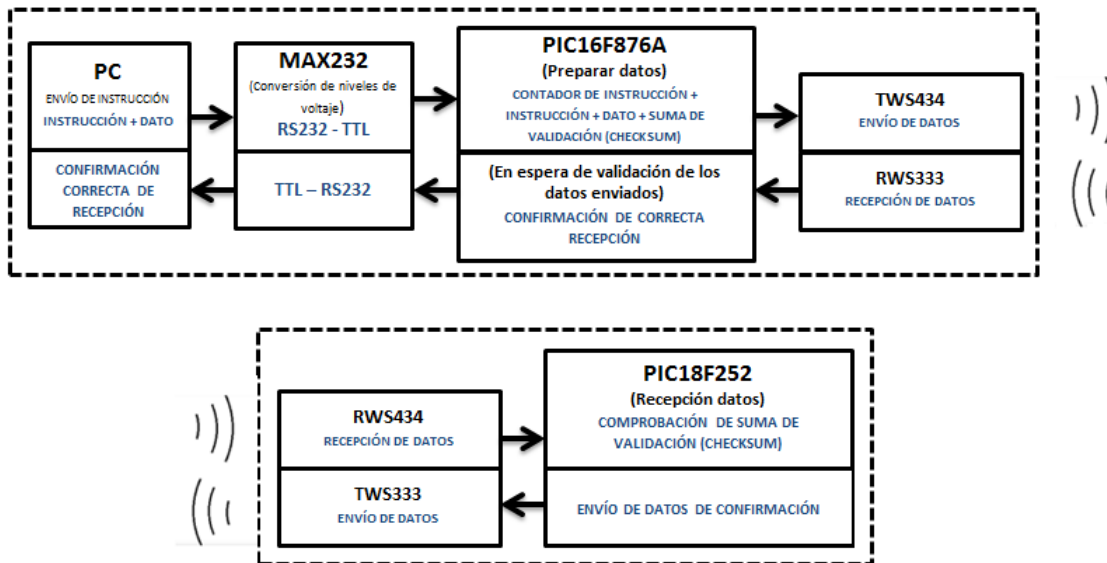


Figura 3.4.2: Diagrama a bloques del procedimiento de envío y recepción de datos.

3.5. Diseño del prototipo

Todo diseño se hizo mediante el software ISIS Proteus e ISIS Ares de Labcenter Electronics, los diseños se presentan a continuación. En la Figura 3.5.1 se muestra el diseño del subsistema de procesamiento del robot móvil.

Este diseño del robot móvil es un sistema de control en lazo abierto debido a que es un proceso secuencial en el que no se tiene una regulación de las variables a controlar. Desde el punto de estabilidad el desarrollo del sistema es simple, el cual la salida no requiere ser medida ni retroalimentada para compararla con la entrada a demás de que es fácil de implementar y muy económico,

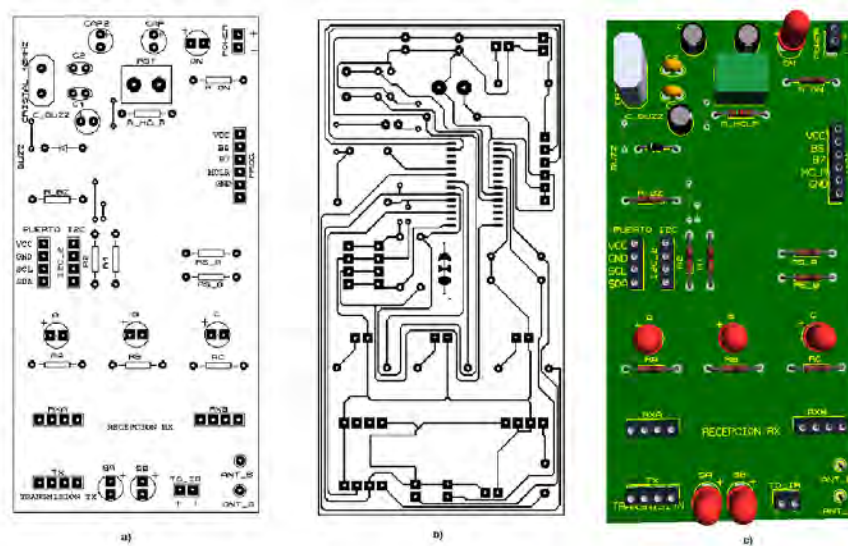


Figura 3.5.1: Diseño del subsistema de procesamiento del robot móvil. a) Diseño de la parte superior de la placa. b) Diseño de la parte inferior de la placa. c) Simulación 3D de la placa.

En la figura Figura 3.5.2 se muestra los diseños realizados para el subsistema actuador de los motores. Este subsistema es alimentado por un paquete de 4 baterías de 1.2 V y 2500 mAh, suficiente para mantener funcionando los motores por una hora aproximadamente debido a la poca resistencia de los bobinados pero compensados en torque.

La Figura 3.5.3 muestra el diseño del subsistema de comunicación, este diseño y el subsistema de procesamiento se usaron microcontroladores de montaje superficial para

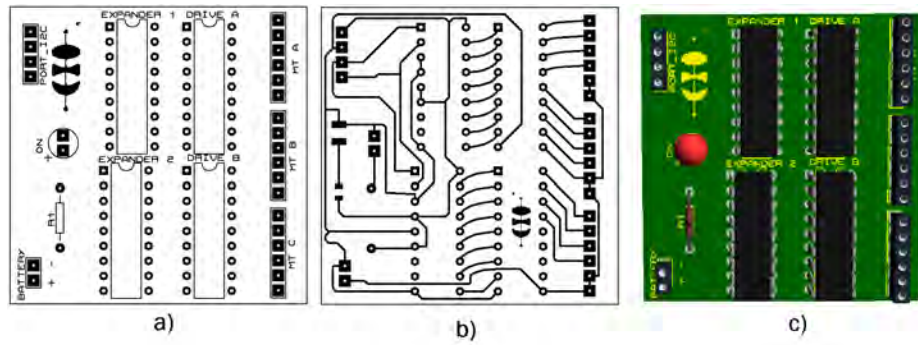


Figura 3.5.2: Diseño del subsistema actuador. a) Diseño de la parte superior de la placa. b) Diseño de la parte inferior de la placa. c) Simulación 3D de la placa.

limitar el mayor espacio.

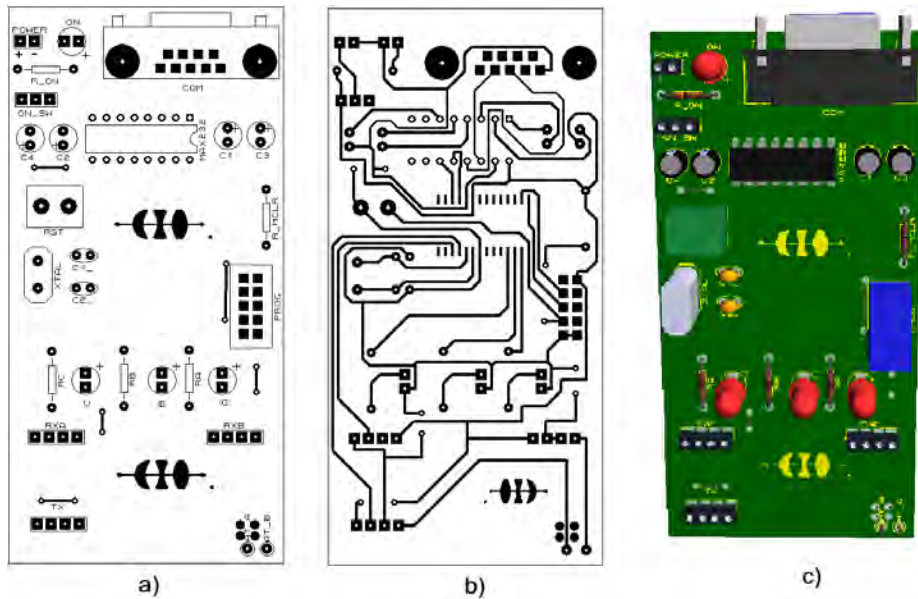


Figura 3.5.3: Diseño del subsistema de comunicación. a) Diseño de la parte superior de la placa. b) Diseño de la parte inferior de la placa. c) Simulación 3D de la placa.

El robot móvil simula lo más parecido a un carro de tres llantas como se muestra en la Figura 3.5.5, la tracción del robot móvil es proporcionada por un motor B acoplado en la llanta delantera, la dirección se lo otorga el motor A. El robot móvil tiene un largo de 25 cm de largo, 15 cm de alto y 12 cm de ancho.

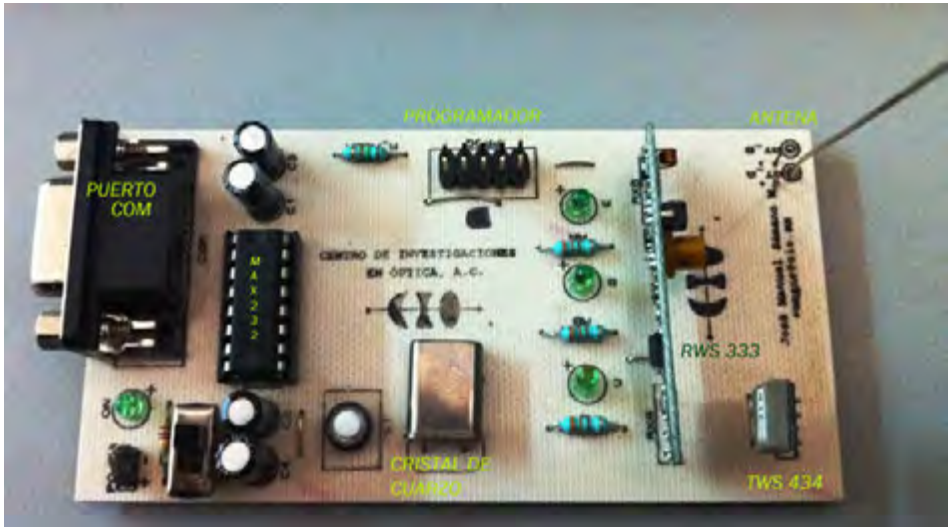


Figura 3.5.4: Diseño real del subsistema de comunicaciones.

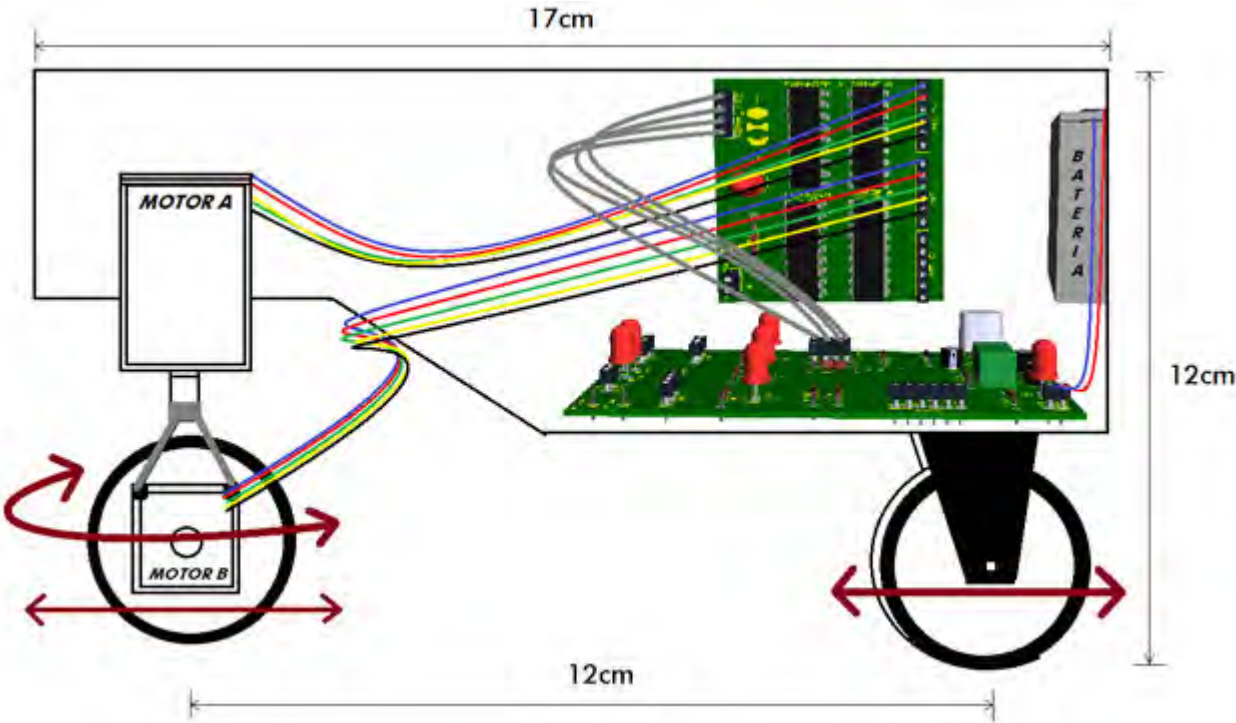


Figura 3.5.5: Diseño del prototipo del robot móvil

4 Planeación Trayectoria y Movimiento

Planificación es un término que significan diferentes cosas para diferente grupo de personas. En robótica se refiere a la automatización de los sistemas mecánicos que tienen sensores, actuación y capacidades de cálculo. Una necesidad fundamental en la robótica es que los algoritmos convierten las especificaciones de alto nivel de las tareas de los seres humanos en el bajo nivel de las descripciones de cómo moverse. Los términos de planificación de movimientos y la planificación de la trayectoria son de uso frecuente para este tipo de problemas. Planificación trayectoria generalmente se refiere al problema de tomar la solución a partir de un algoritmo de planificación de movimiento del robot y determinar cómo se mueven a lo largo de la solución en una forma que se respete las limitaciones mecánicas del robot de acuerdo a su *holonomía*. Una de las posibles formas de clasificar los robots es distinguiendo si son holonómicos o no, como es el caso de nuestro robot móvil. Podemos decir que la holonomía es una distinción que está relacionada con su movilidad. Un vehículo con un sistema como el nuestro de tres llantas similar a un triciclo, por ejemplo, no lo es, porque para poder desplazarse en el sentido lateral tiene que realizar varias maniobras previas.

Con el fin de obtener los controladores de retroalimentación más adecuados para cada caso, clasificamos las tareas de movimiento posibles de la siguiente manera:

- Movimiento punto a punto: El robot debe alcanzar una meta con la configuración deseada a partir de una configuración inicial dada.
- Seguimiento de Ruta: El robot debe alcanzar y seguir una trayectoria geométrica en el espacio cartesiano a partir de una configuración inicial dada (dentro o fuera de la ruta).
- Seguimiento de la Trayectoria: El robot debe alcanzar y seguir una trayectoria

en el espacio cartesiano (es decir, una trayectoria geométrica con una ley de sincronización asociada) a partir de una configuración inicial dado (dentro o fuera de la trayectoria).

Las tres tareas se muestran en la Figura 4.0.1 con referencia a nuestro robot móvil.

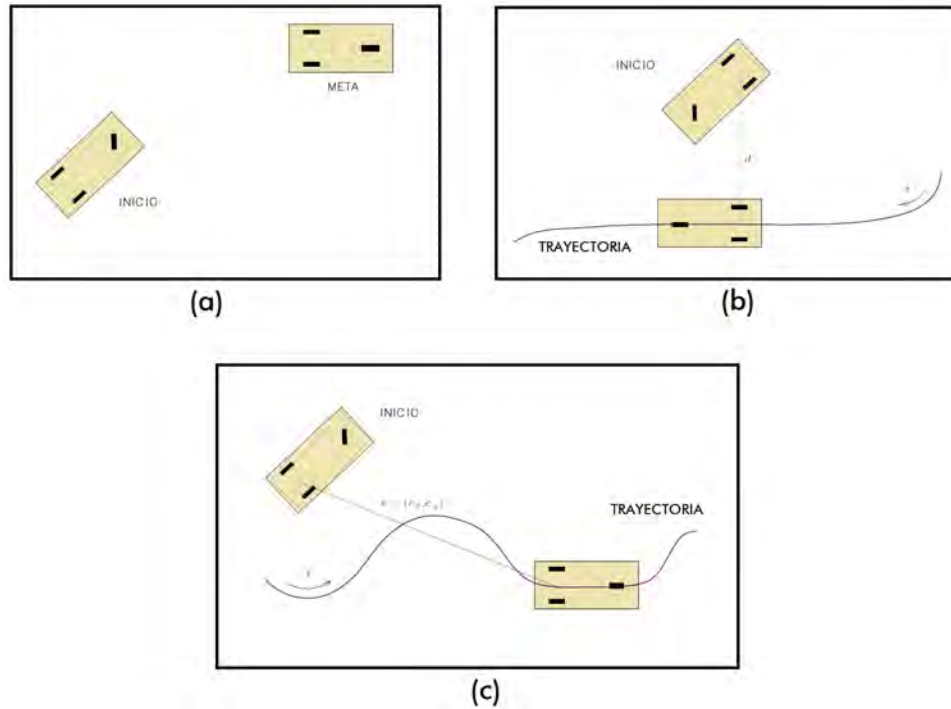


Figura 4.0.1: Tareas de Movimiento: Movimiento punto a punto (a), Seguimiento de Ruta (b), Seguimiento de Trayectoria (c).

4.1. Exploración Rápida de Árboles Aleatorios (Rapidly-Exploring Random Trees)

La exploración rápida de árboles aleatorios o por sus siglas en inglés RRT fue introducido en [21] como una estructura de datos eficiente y esquema de muestreo para buscar rápidamente un alto espacio dimensional que tienen tanto las restricciones algebraicas (derivado desde obstáculos) y las limitaciones diferenciales (derivado de noholonomía y dinámica). La idea clave es que la tendencia de exploración hacia las partes inexploradas del espacio mediante el muestreo de puntos en el espacio de estados, y de manera

progresiva "tirando" el árbol de búsqueda hacia ellos. Al menos otros dos algoritmos de técnicas de planeación han sido propuestos y que generan árboles de búsqueda: The Ariadne's clew algorithm[1] y, the planners in [11, 25]. Intuitivamente, estos planificadores tratan de "empujar" el camino del árbol de búsqueda desde vértices previamente construidos, que en contraste con el algoritmo RRT, utiliza el espacio que rodea para "arrastrar" el árbol de búsqueda, en última instancia conduce a una cobertura uniforme del espacio de estados.

```

BUILD_RRT( $x_{init}$ )
1   $\mathcal{T}.init(x_{init});$ 
2  for  $k = 1$  to  $K$  do
3     $x_{rand} \leftarrow RANDOM\_STATE();$ 
4     $EXTEND(\mathcal{T}, x_{rand});$ 
5  Return  $\mathcal{T}$ 

EXTEND( $\mathcal{T}, x$ )
1   $x_{near} \leftarrow NEAREST\_NEIGHBOR(x, \mathcal{T});$ 
2  if  $NEW\_STATE(x, x_{near}, x_{new}, u_{new})$  then
3     $\mathcal{T}.add\_vertex(x_{new});$ 
4     $\mathcal{T}.add\_edge(x_{near}, x_{new}, u_{new});$ 
5    if  $x_{new} = x$  then
6      Return Reached;
7    else
8      Return Advanced;
9  Return Trapped;

```

Figura 4.1.1: Algoritmo de construcción básico RRT

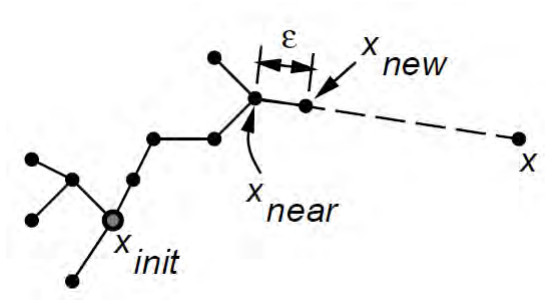


Figura 4.1.2: Función EXTEND

El algoritmo básico de construcción RRT está dado en la Figura 4.1.1. Se lleva a cabo una iteración en el que cada paso intenta extender el RRT mediante la adición de

un nuevo vértice que es empujado por el estado de selección aleatoria. La función EXTEND, ilustrada en la Figura 4.1.2, selecciona el vértice más cercano en el RRT para un estado de muestra dado. El vértice “cercano” es elegido de acuerdo con la métrica, ρ . La función NEW_STATE hace un movimiento hacia x mediante la aplicación de una entrada $u \in U$ por algún incremento de tiempo Δt . Esta entrada puede ser elegida aleatoriamente, o seleccionando todas las posibles entradas y elegir la que obtiene un nuevo estado lo más cerca posible a la muestra, x (si U es infinito, entonces una aproximación o técnica analítica puede ser usada). En el caso de planeación holonómica, el valor óptimo para u puede ser elegida fácilmente mediante el cálculo de un vector simple. NEW_STATE implícitamente también usa la función de detección de colisión para determinar si el nuevo estado (y todo estado intermedio) satisface las restricciones globales. Para muchos problemas, esto puede llevarse rápidamente (casi en tiempo constante) usando algoritmos de cálculo de incrementos de distancia [11, 4] mediante el almacenamiento de las invariantes apropiadas con cada uno de los vértices RRT. Si NEW_STATE es correcto, el nuevo estado y entrada son representados en x_{new} and u_{new} , respectivamente. Tres situaciones pueden ocurrir: *Alcanzado*, en el cual el nuevo vértice alcanza la muestra x (para el caso de planeación noholonómica, podrias en vez de tener un umbral, $\|x_{new} - x\| < \epsilon$ por uno pequeño $\epsilon > 0$); *Avanzado*, en el que un vértice $x_{new} \neq x$ es agregado al RRT; *Atrapado*, en el que NEW_STATE falla para producir un nuevo estado que se encuentra en x_{free} .

La parte superior de la Figura 4.1.3 muestra un RRT para un problema de planeación holonómica, construida en un espacio 2D rectangular. La parte inferior de la misma figura muestra el diagrama Voronoi de los vértices RRT; note que la probabilidad de que el vértice sea seleccionado para extensión es proporcional a el área de su región Voronoi. Esto predispone al RRT para explorar rápidamente.

En simplificación enumeraremos los pasos para generar nuestra planeación de trayectoria para el robot móvil.

PASOS DE NUESTRO RRT:

1. Generar un punto al azar en el espacio de configuración.
2. Busque el nodo más cercano (usando Distancia, D) para expandir el árbol.
3. Generar un ángulo de dirección aleatoria comprendidos entre -30 y 30 grados.
4. En $D < \text{Zona Crítica Distancia}$.

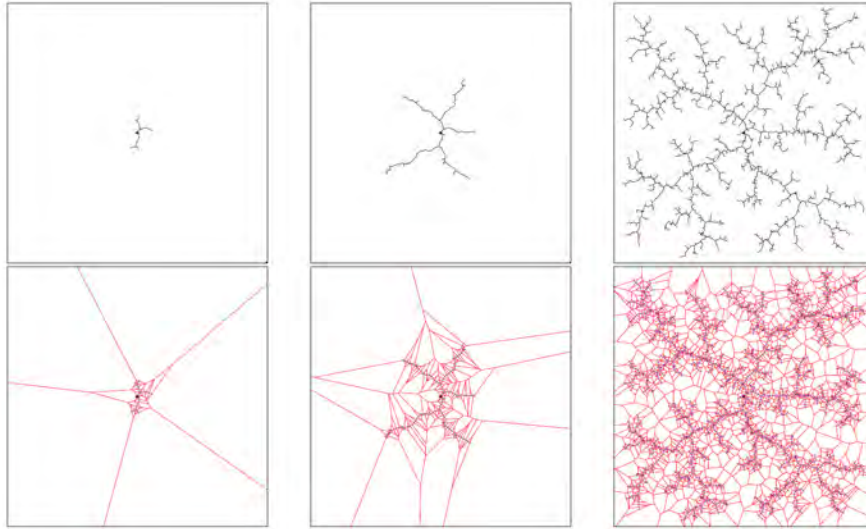


Figura 4.1.3: RRT predispuesto por las grandes regiones Voronoi para explorar rápidamente, antes de cubrir uniformemente el espacio.

i) de doble sentido de movimiento de habilitación (hacia delante y hacia atrás), de lo contrario el robot sólo podrá avanzar.

ii) Si un nodo que ya está dentro de la zona crítica, los otros nodos tienen que estar dentro de la zona.

5. Si la colisión es libre, expanda el árbol de utilizar el ángulo de dirección y la velocidad fija basada en las leyes de control.

6. Si las configuraciones del robot está dentro de un rango de tolerancia de sus configuraciones meta, volver al *Alcance*.

7. De otra forma, si el árbol se expande luego regresar a *Avanzado*.

8. De lo contrario regresar a *Atrapado*.

Des esta forma y haciendo uso del modelo cinemático de nuestro robot se obtiene el siguiente resultado que se muestra en la Figura 4.1.4

4.2. Análisis y Modelo Cinemático del Robot Móvil

La característica principal del modelo cinemático de los robots móviles de ruedas es la presencia de limitaciones no holonómicas debido a la condición de rodadura sin

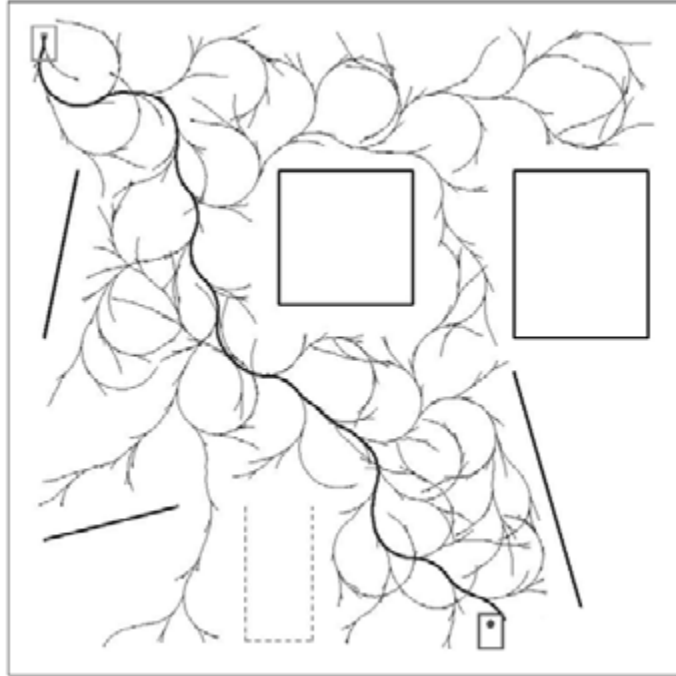


Figura 4.1.4: Planeación de trayectoria usando RRT y modelo Cinemático de nuestro robot móvil.

deslizamiento entre las ruedas y el suelo. El caso de una única rueda se analiza primero. Considere la posibilidad de una rueda que rueda sobre un plano mientras se mantiene su cuerpo vertical, como se muestra en la Figura 4.2.1. Este tipo de sistema también se conoce como un monociclo. Su configuración puede ser descrito por un vector q de tres coordenadas generalizadas, es decir; las coordenadas de posición x , y del punto de contacto con el suelo en un marco de fijo y el ángulo θ midiendo la orientación de la rueda con respecto al eje x . El sistema generalizado de velocidades \dot{q} no puede asumir valores independientes, en particular, deben satisfacer la restricción

$$\begin{bmatrix} \sin \theta & \cos \theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0, \quad (4.2.1)$$

que supone que la velocidad lineal del centro de la rueda se encuentra en el plano del cuerpo de la rueda (velocidad lateral igual a cero).

La Ecuación 4.2.1 es un ejemplo típico de una restricción Pfaffian ¹ $C(q)\dot{q}=0$, es decir, lineal en las velocidades generalizadas. Como una consecuencia, todas las velocidades generalizadas admisibles están contenidas en el espacio nulo de la matriz de restricción $C(q)$. En este caso, se obtiene

$$\dot{q} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2, \quad (4.2.2)$$

donde v_1 y v_2 son respectivamente la velocidad lineal de la rueda y su velocidad angular alrededor del eje vertical. Como la elección de una base para el espacio nulo de la matriz C no es única, los componentes de v puede también asumir diferentes significados. Por otra parte, puede no tener relación directa con los controles reales disponibles, que son en general fuerzas o torques. Por esta razón la Ecuación 4.2.2 es llamada *modelo cinemático* del unicycle.

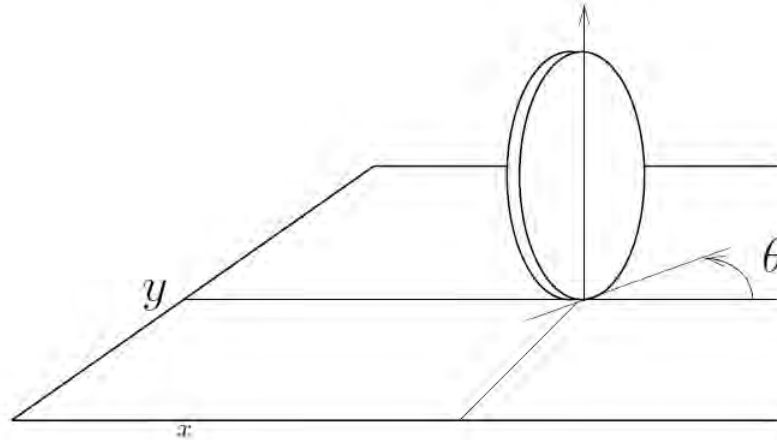


Figura 4.2.1: Coordenadas Generalizadas de un Unicycle

Pasando ahora a nuestro robot que tiene las mismas características cinemáticas de un automóvil triciclo como se muestra en la Figura 4.2.2. Por simplicidad, supongamos que las dos del eje trasero colapsan en el punto medio del eje. La rueda delantera se puede pilotear con la orientación de la rueda trasera fija. Las coordenadas generalizadas son

¹En matemáticas, el determinante de una matriz antisimétrica siempre se puede escribir como el cuadrado de un polinomio en las entradas de la matriz. Este polinomio se llama la Pfaffain de la matriz.

$q = (x, y, \theta, \phi)$, donde x, y son las coordenadas cartesianas de la rueda trasera, θ mide la orientación del cuerpo del carro con respecto al eje x, y y ϕ es el ángulo de dirección. El sistema está sujeto a dos restricciones holonómicas, uno para cada rueda:

$$\begin{aligned} \dot{x}_f \sin(\theta + \phi) - \dot{y}_f(\theta + \phi) &= 0 \\ \dot{x}_f \sin \theta - \dot{y}_f \theta &= 0, \end{aligned}$$

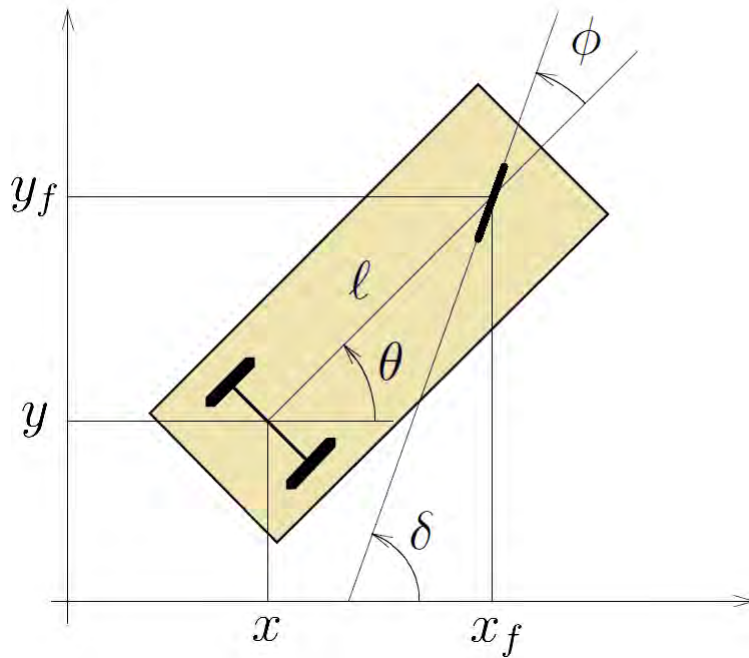


Figura 4.2.2: Coordenadas generalizadas del robot móvil

con x_f, y_f indicando las coordenadas cartesianas de la rueda delantera. Mediante el uso de la restricción de cuerpo rígido

$$x_f = x + l \sin \theta,$$

donde l es la distancia entre las ruedas, la primer restricción cinemática viene siendo

$$\dot{x}_f \sin(\theta + \phi) - \dot{y}_f(\theta + \phi) - \dot{\theta} l \cos \theta = 0.$$

La matriz de restricción Pfaffain es

$$C(q) = \begin{bmatrix} \sin(\theta + \phi) & -\cos(\theta + \phi) & -l\cos\phi & 0 \\ \sin\theta & -\cos\theta & 0 & 0 \end{bmatrix}$$

El modelo para la conducción de las ruedas delanteras se obtiene como

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\sin\phi \\ \sin\phi/l \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2, \quad (4.2.3)$$

donde la velocidad v_1 de conducción se refiere ahora a la rueda delantera. Tenga en cuenta que la singularidad anterior no se produce en este modelo; de hecho, en $\phi = \pm\pi/2$ el carro puede pivotar alrededor de su rueda trasera.

4.3. Controlabilidad sobre una Trayectoria

Consideremos ahora una trayectoria de estado de referencia deseado $q_d(t) = (x_d(t), y_d(t), \theta_d(t), \phi_d(t))$ para nuestro robot. Con el fin de ser factible, esta trayectoria debe satisfacer las restricciones no holonómicas sobre el movimiento del robot móvil. La generación de este tipo de trayectorias, así como de las correspondientes entradas de velocidad de referencia v_{d1} y v_{d2} se abordan con más detalles en [33] capítulo 4, sección 3.

Definiendo $\tilde{q}(t) = q(t) - q_d(t)$ y $\tilde{v}(t) = v(t) - v_d(t)$, la linealización aproximada del sistema Ecuación 4.2.3 sobre la trayectoria de referencia se obtiene como

$$\dot{\tilde{q}} = A(t) + B(t)\tilde{v}, \quad (4.3.1)$$

con

$$A(t) = \sum_{i=1}^2 v_{di}(t) \frac{\partial g_i}{\partial q} \Big|_{q=q_d(t)}, \quad B(t) = G(q_d(t)).$$

Procedimiento de cálculo simple

$$A(t) = \begin{bmatrix} 0 & 0 & -\sin\theta_d(t)v_{d1}(t) & 0 \\ 0 & 0 & \cos\theta_d(t)v_{d1}(t) & 0 \\ 0 & 0 & 0 & v_{d1}(t)/\ell\cos^2\theta_d(t) \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B(t) = \begin{bmatrix} \cos\theta_d(t) & 0 \\ \sin\theta_d(t) & 0 \\ \tan\theta_d(t)/\ell & 0 \\ 0 & 1 \end{bmatrix}$$

Tenga en cuenta que el sistema linealizado es variable en el tiempo a través de la dependencia de tiempo de la trayectoria de referencia. Como consecuencia, el análisis de controlabilidad es más complicado que en el caso invariante en el tiempo, y que consiste en comprobar si la controlabilidad Gramian es no singular [16].

Por ejemplo, consideramos el caso especial de una trayectoria lineal con velocidad constante, en el que uno cae en un sistema invariante en el tiempo. De hecho, en esta situación tenemos $v_{d1}(t) \equiv v_{d1}$ (un valor distinto de cero constante) y $v_{d2}(t) \equiv 0$. Además, $\theta_d(t) \equiv \theta_d(t_0)$ y $\phi(t) \equiv 0$. La condición de controlabilidad es

$$\text{rank} \begin{bmatrix} B & AB & A^2B & A^3B \end{bmatrix} = 4.$$

Es sencillo verificar que la matriz de controlabilidad tiene una única matriz 4x4 no nula. Por lo tanto, el sistema linealizado es controlable, siempre y cuando $\theta_d \neq \pm\pi/2$ y $v_{d1} \neq 0$ (lo cual no es inesperado, ya que para $v_{d1} = 0$ la trayectoria colapsaría a un punto). Esto implica que el sistema Ecuación 4.2.3 puede ser localmente estabilizado sobre la trayectoria de referencia por una realimentación lineal.

Cabe mencionar que el estudio referente a este tema es muy amplio y minucioso por lo que hemos referido lo más esencial aplicado a nuestro proyecto.

5 Implementación del Sistema de Visión por Computadora

5.1. Implementación de Hardware

5.1.1. Cámara

Lo primero que hay que elegir a la hora de diseñar un sistema de visión artificial es el tipo de cámara. En la mayoría de los casos en los que se utiliza un sistema de visión artificial la adquisición se realiza mediante cámaras monocromáticas, notándose ventajas principales en tiempo de cómputo y económicas. Note que una imagen a color necesita más tiempo de procesamiento y para procesos en tiempo real la velocidad de cálculo es un parámetro importante a tomar en consideración. Por otra parte, muchos de los procesos se basan en el reconocimiento de formas a partir de la determinación de los contornos de los objetos, siendo innecesaria la información cromática de los objetos. Aún así, hay muchas aplicaciones en las que la información de color es fundamental como en el caso de este proyecto. La imagen en color realmente proporciona una información más completa que puede servirnos para etapas de segmentación y extracción de parámetros como es el caso de este trabajo.

De acuerdo a lo anterior, elegimos una cámara web “Rocketfish” de alta calidad con las ventajas de costo-beneficio, captura de vídeo con resolución de hasta 2 megapíxeles para detalles realistas y movimiento, tecnología de enfoque automático que permite captar imágenes claras y detalladas y conexión USB 2.0.

5.1.2. Sistema de iluminación

Un entorno debidamente controlado es imprescindible para obtener condiciones de partida óptimas que aseguren una perfecta adquisición. Dentro del entorno se encuentran partes fundamentales que influyen en las expectativas tanto de calidad como de la imagen buscada como es la iluminación, el fondo, la posición de la cámara, entre otros.

La iluminación es un factor importante en el procesamiento de imágenes para obtener resultados óptimos. Una carencia de iluminación implica obtener imágenes de menor calidad lo que puede ocasionar resultados con mayores porcentajes de errores. La iluminación de la escena tiene que realizarse de un forma correcta, dada la importancia que tiene. Podemos encontrar dos formas de iluminación fundamentales como la iluminación frontal e iluminación trasera o retroiluminación. En la iluminación frontal incide directamente sobre el objeto, ya sea horizontal o verticalmente de forma oblicua o de forma difusa. La iluminación trasera o retroiluminación se ilumina una pantalla de forma que lo que se busca es el contorno del objeto a modo de sombra.

La iluminación frontal permite distinguir los detalles de los objetos como su forma, permitiendo de esta manera extraer más parámetros de cada objeto, como el color, detalles internos, entre otros; que permitan una mejor segmentación que a diferencia de la retroiluminación sólo sirve para la detección de contornos simplificados.

En la Figura 5.1.1 siguiente observamos una imagen con sólo la iluminación ambiental en un espacio cerrado. Se observan los objetos fácilmente, sin embargo; se presentan problemas de ruido en el procesamiento de la imagen lo que dificulta la segmentación para identificar cada objeto. La Figura 5.1.2 muestra la imagen con una adecuada iluminación.



Figura 5.1.1: Escena sin iluminación

En la colocación del sistema de iluminación se ha considerado los siguientes factores:

- La ubicación del entorno del robot: El espacio donde el robot interactuará.
- Cámara: Hacemos uso del control automático de ganancia de la cámara. Esto permite disminuir el brillo excedente producido por el robot móvil o los obstáculos.
- Iluminación ambiental: La incidencia de toda iluminación espontánea del exterior puede interferir en el proceso, por lo que se reducirá lo más posible la entrada de la luz ambiental.
- Reflexión: Posibles reflexiones del robot móvil, entorno u obstáculos.

Atendiendo dichos factores, se decidió hacer uso de lámparas de luz blanca usando la disposición frontal, tal como se muestra en la Figura 5.1.3.



Figura 5.1.2: Escena con iluminación.

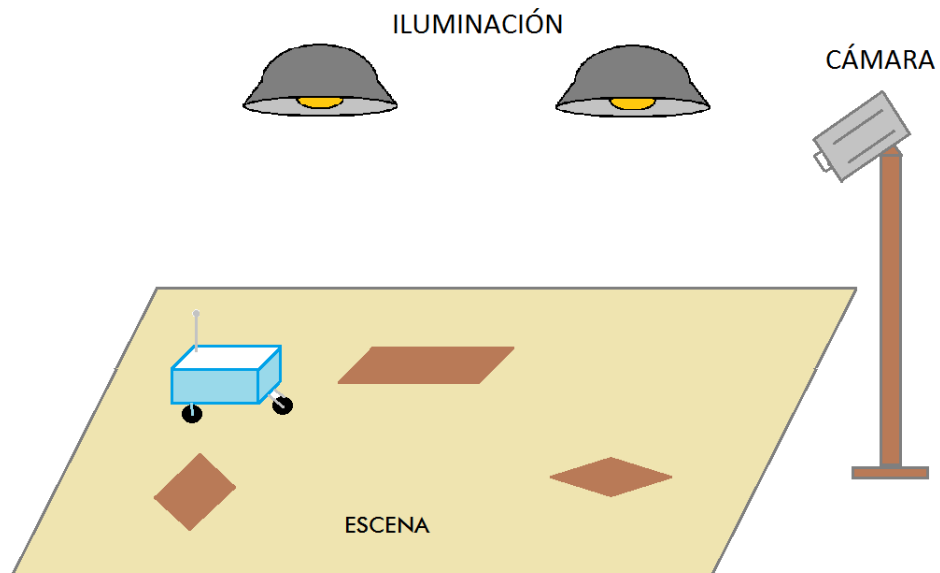


Figura 5.1.3: Representación ilustrativa para mostrar la iluminación frontal empleada en la escena.

5.1.3. Interfaz control remoto PC- Robot Móvil

El software de visión artificial se desarrolla en una computadora con velocidad de procesador de 2 GHz de velocidad para un óptimo desempeño. La computadora envía instrucciones al robot móvil como resultado del análisis realizado. El resultado del análisis del entorno nos da como resultado la posición de los objetos, orientación, número de objetos, posición del robot móvil, etc. De acuerdo a esto el robot móvil ejecutará las instrucciones que recibe del computador a través del control inalámbrico del robot que se encuentra conectado al computador mediante una conexión serial.

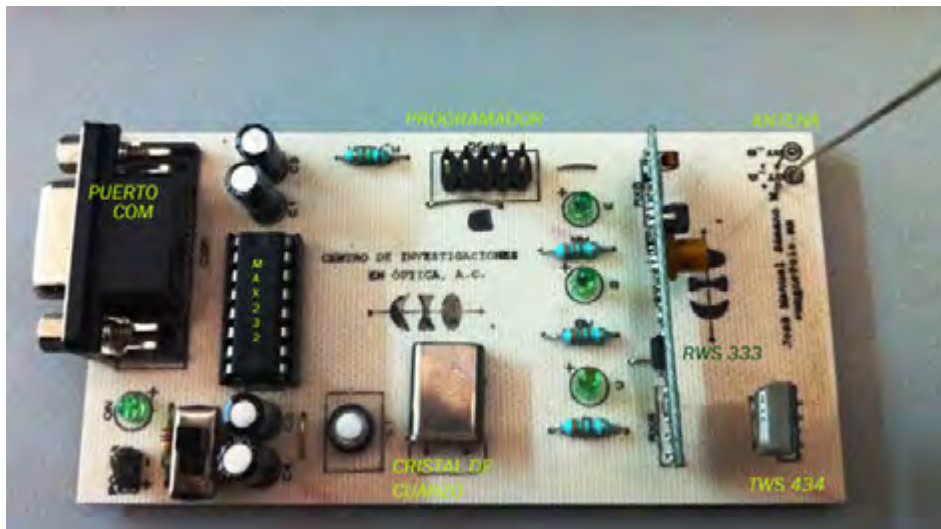


Figura 5.1.4: Figura de Computadora y puerto COM - control - robot

5.2. Implementación del Software

Se implementó el desarrollo de software del proyecto bajo la plataforma Microsoft Visual C++. Para ayudarnos más con el software del sistema de visión, se ha hecho uso de la librería OpenCV.

OpenCV es una librería de código abierto de visión por computadora. La librería está escrita en C y C++, corre bajo Linux, Windows y Mac OS X. La razón de usar esta librería es porque fue diseñado para eficiencia computacional y con un fuerte enfoque para aplicaciones en tiempo real, ideal para este proyecto, además que está a disposición de cualquier persona.

5.2.1. Control del Robot Móvil

Para permitir el manejo del robot móvil mediante el control remoto, se desarrolló una aplicación que nos permite trabajar de forma manual o bien; mediante el envío de instrucciones desde cualquier otra aplicación a fin por medio de comunicación TCP/IP ya sea desde el mismo ordenador o desde otro local.

5.2.1.1. Descripción del funcionamiento

El funcionamiento radica en el análisis y procesamiento de los datos enviados y recibidos a cargo de los microcontroladores de ambos dispositivos por vía RF. Cada orden a ejercer es enviado desde el software del computador hacia el microcontrolador del control remoto. El software de control remoto sirve para propósitos generales y bien puede mandar instrucciones manualmente desde su interfaz o ser controlado de forma remota, es decir; enviar una acción con ciertos parámetros en número de pasos para los motores desde su interfaz o bien iniciar un host para ser controlado por TCP/IP ya sea localmente desde el mismo ordenador o un ordenador remoto por medio de otro software. (Véase la figura Figura 5.2.1)



Figura 5.2.1: Interfaz de la aplicación creada especialmente para el control remoto del robot móvil.

El microcontrolador del control remoto se encarga de recibir el dato y se blindo para ser enviado por RF por un método de suma o checksum que involucra un resultado de todos los bytes enviados. El método consiste enviar bytes especificando el número de

instrucción, tipo de acción, parámetros de la acción, checksum, fin de la transmisión. Más detalles sobre el hardware en capítulo 3.4.

Una vez agregando nuestra librería de C/C++ para comunicación con el host, las instrucciones a enviar son de lo más simple como se muestran en la Figura 5.2.2 donde la unidad del parámetro enviado corresponde a 1.875° equivalente a un paso para los motores.

1	<code>centrar();</code>
2	<code>avanza(50);</code>
3	<code>gira_derecha(25);</code>
4	<code>avanza(140);</code>
5	<code>retrocede(20);</code>
6	<code>detener();</code>

Figura 5.2.2: Ejemplo de secuencia de órdenes de alto nivel para el robot móvil.

5.2.2. Calibración de cámara

Para obtener un imagen correcta sin deformación, es necesario efectuar un calibración de la posición de la cámara. Recordemos que la calibración de una cámara es el proceso que permite dentro del campo de la visión artificial, la obtención de los parámetros que definen las condiciones de formación de la imagen, incluyendo la geometría interna y la óptica de la cámara, así como su posición y orientación respecto a un objeto de referencia o patrón de calibración. En sí; la calibración de cámara es la determinación de los parámetros extrínsecos en intrínsecos de un sistema de visión. La Figura 5.2.3 nos muestra el resultado del procesamiento de calibración de cámara, se observa como se ha corregido la curvatura de la imagen ocasionada por la lente de la cámara.

5.2.3. Transformación proyectiva del entorno

La necesidad de efectuar transformaciones proyectivas en el sistema de visión empleado en este trabajo, nos favorece en gran medida a la eliminación de distorsión de perspectiva de la escena, permitiendo una vista frontal ortogonal sintetizada de la misma. De

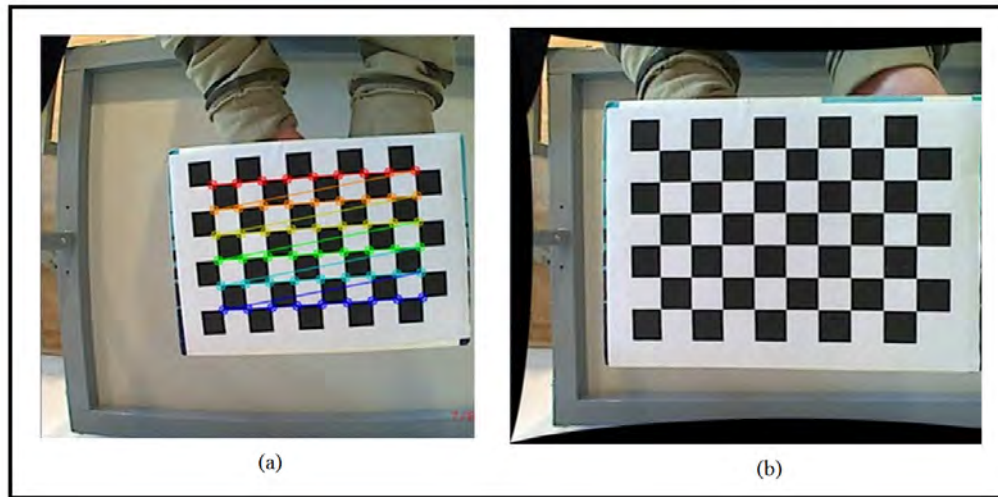


Figura 5.2.3: Calibración de cámara. (a) Detección de bordes para obtener parámetros extrínsecos. (b) Resultado de la calibración de cámara.

esta forma, se facilitará el uso de la imagen para el procesamiento a fin, que se requiere para generar un entorno libre de distorsión.

Después de cada captura de imagen realizada al espacio de trabajo donde se desenvolverá el robot móvil, se calcula la transformación inversa mediante la asignación de cuatro puntos que representan las cuatro esquinas del ambiente de trabajo, observe la Figura 5.2.4.

Note que la cámara se encuentra a una altura y ángulo dispuesto que nos ofrece una perspectiva del ambiente de trabajo.

De antemano se sabe la información métrica y forma geométrica del espacio de trabajo, por lo que; el siguiente paso será aplicar una transformación proyectiva de forma que describa proporcionalmente la forma métrica y geométrica en una vista frontal ortogonal sintetizada, es decir; si nuestro espacio de trabajo tiene una forma geométrica rectangular se adecuaran los puntos correspondientes de forma tal que describa los más aproximado el espacio de trabajo. Esto se logra mediante una transformación lineal que esta dada por la ecuación:

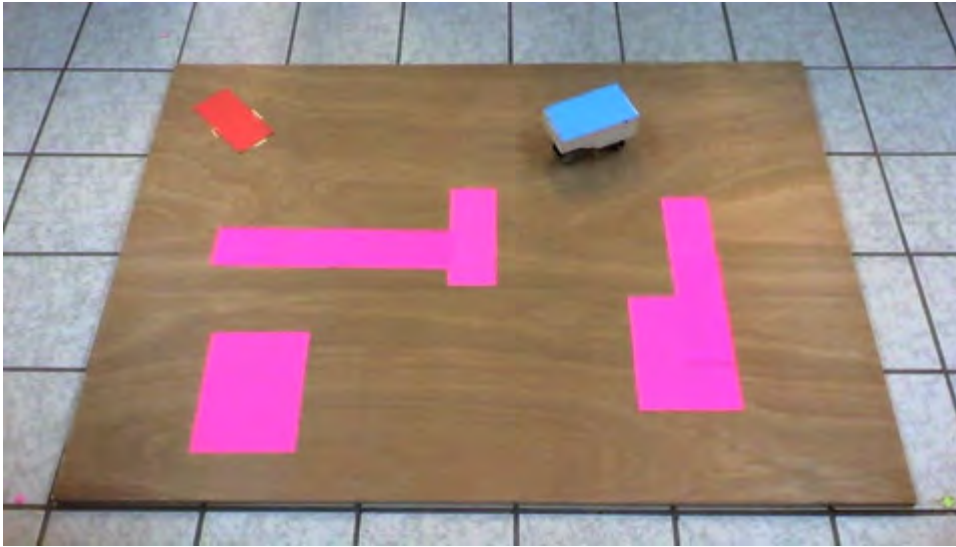


Figura 5.2.4: Imagen de la escena con la perspectiva inicial.

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_n \end{bmatrix} \mathbf{H} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{bmatrix} \quad (5.2.1)$$

Donde la matriz \mathbf{H} representa la transformación lineal que elimina la distorsión de perspectiva de la escena, se puede comenzar a hacer un mapeo para toda la escena, es decir; el resultado se observa en la Figura 5.2.5. A y B son los puntos correspondientes entre ellos. Mayor detalles se explica en el capítulo 2.1.3

5.2.4. Preprocesamiento de la imagen

La degradación de una imagen que regularmente es proveniente por el ruido de los sensores de captura, imprecisiones en el enfoque de la cámara, movimientos de la misma o perturbaciones aleatorias, se manifiestan en la imagen captura en forma de ruido, pérdida de definición y fidelidad de la imagen, el poco o demasiado brillo, contraste, entre otros.

El preprocesado de una imagen pretende reparar los desperfectos producidos o no eliminados por el hardware, mediante operaciones de restauración. Los algoritmos de

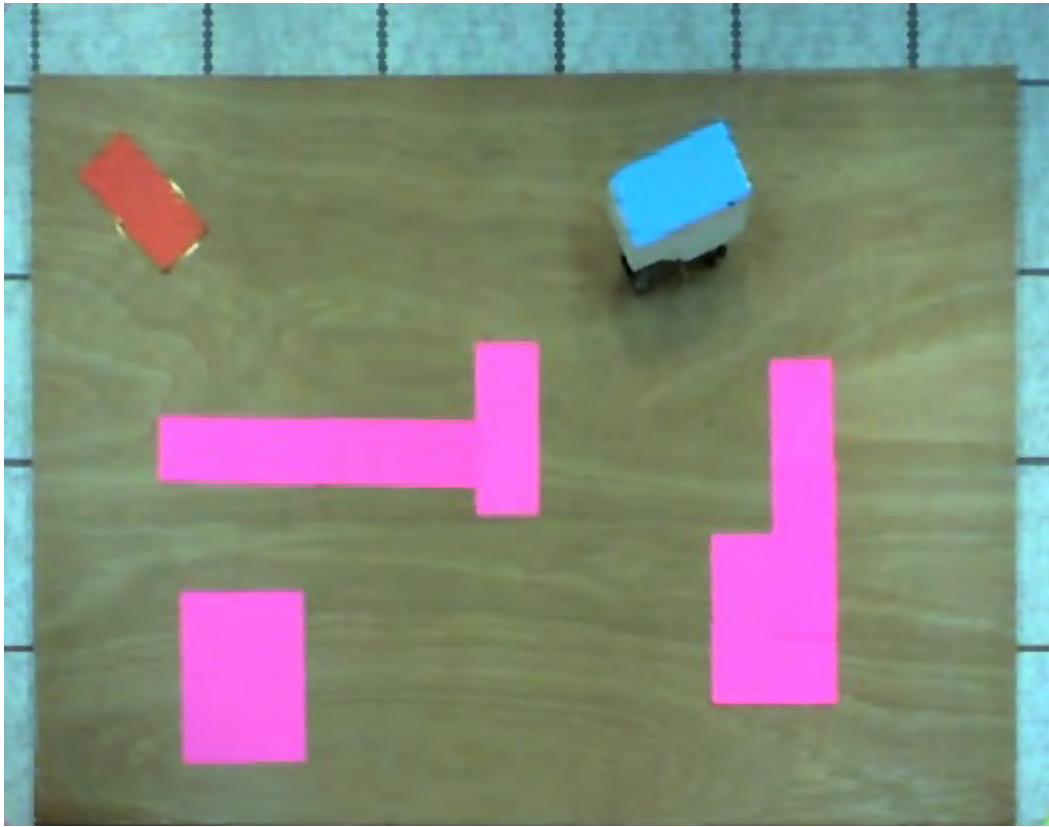


Figura 5.2.5: Imagen de la escena sin distorsión de perspectiva

preprocesado permiten modificar la imagen para eliminar ruido, transformarla geoméricamente, mejorar la intensidad o el contraste, etc.; proceso que intentan mejorar el resultado final de la imagen capturada.

En el proceso de visión artificial estos algoritmos tienen que ser utilizados lo menos posible, ya que un uso excesivo de ellos repercutirá en el tiempo de proceso e indicará que la calibración, iluminación y selección de los elementos de la etapa de adquisición no han sido los adecuados.

Los pasos a considerar en esta etapa para el preprocesamiento de la imagen son los siguientes:

- Filtrado.
- Segmentación.
- Umbralización

- Binarización
- Erosión y Dilatación.

5.2.4.1. Filtrado

La mayoría de las implementaciones de filtros se realizan en dos dominios: el dominio espacial y el dominio frecuencial.

Los métodos basados en el dominio espacial hacen referencia a la manipulación directa de la luminancia de los píxeles, mientras que los métodos basados en el dominio frecuencial modifican indirectamente la luminancia de cada píxel utilizando como factores de ponderación los valores de los otros píxeles de la imagen o del entorno del punto y las relaciones numéricas entre ellos.

Este trabajo se ha hecho uso de filtros espaciales para el propósito requerido. Este tipo de filtros, suele consistir en recorrer toda la imagen, píxel a píxel, y realizar alguna operación aritmética con un número concreto de píxeles vecinos. Al conjunto de estos vecinos se les denomina ventana, la cual podrá ser de 3×3 , 5×5 ,..., etc.

En la Figura 5.2.6 se observa la representación de una ventana de 3×3 y 5×5 , en la que los píxeles vecinos van del 1 al 9 y del 1 al 25 respectivamente, siendo el píxel central sobre el que se aplica un algoritmo. La mayoría de los algoritmos suelen ser combinaciones lineales de los píxeles de la ventana. Para representar un filtro se suele dar el vector o matriz de convolución o núcleo que representa la combinación lineal entre ellos.

El filtro que se ha usado en este trabajo para reducir el ruido es el filtro de la mediana ya que produce un efecto de suavizado y bordes más abruptos. El filtro de la mediana consiste en obtener la lista de todos los valores de los píxeles de la ventana, ordenarlos y tomar el valor del media, es decir; el valor que tenga igual número de valores superiores e inferior dentro de la ventana. De esta forma se harán correcciones en la imagen para una mejor segmentación.

Este filtro es muy usado para eliminar ruido impulsivo en una imagen. La ventaja de este filtro es que el valor final del píxel es un valor real presente en la imagen y no un promedio, de este modo se reduce el efecto borroso que tienen las imágenes que han sufrido un filtro de media. Además el filtro de la mediana es menos sensible a valores

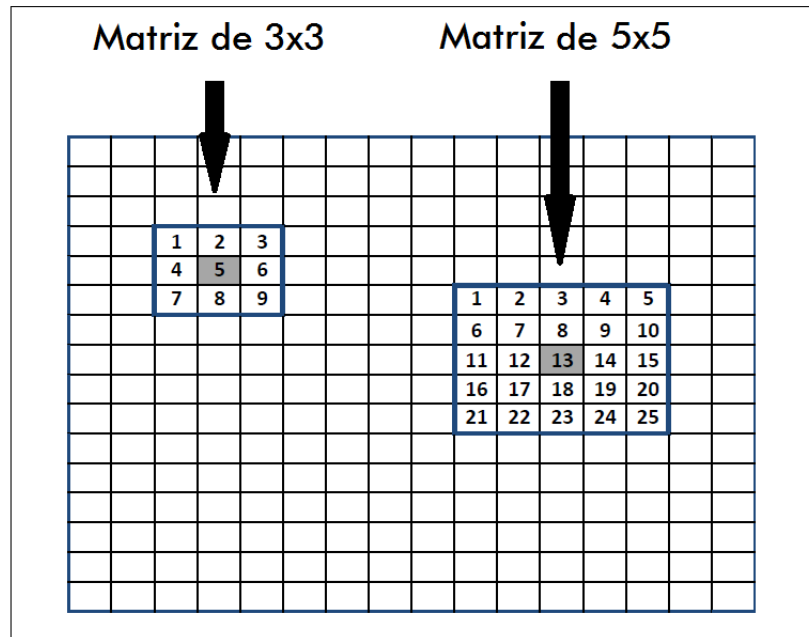


Figura 5.2.6: Representación de dos tipos de ventanas para el filtrado de una imagen.

extremos. El inconveniente es que resulta más complejo de calcular ya que hay que ordenar los diferentes valores que aparecen en los píxeles incluidos en la ventana y determinar cual es el valor central.

En la Figura 5.2.7 se puede observar la imagen de la escena sin filtrar, y la Figura 5.2.8 se observa con la respuesta al filtro de la mediana. Este ligero cambio ayuda a remover pequeñas imperfecciones en la imagen que nos permitirá una mejor segmentación.

5.2.4.2. Segmentación por filtro de color

Segmentar una imagen digital significa dividirla en zonas disjuntas e individualizarlas, es decir; consiste en diferenciar los diversos objetos y dónde se encuentra del fondo, que puede ser más o menos complejo de la imagen.

Terminada la etapa de segmentación, se tienen que conocer perfectamente los objetos que hay para extraer las características propias de cada uno de ellos. La etapa de segmentación es importante para el reconocimiento de formas y para simplificar la escena en una menos compleja.

La técnica de segmentación empleada en este trabajo se aplica para disposiciones de

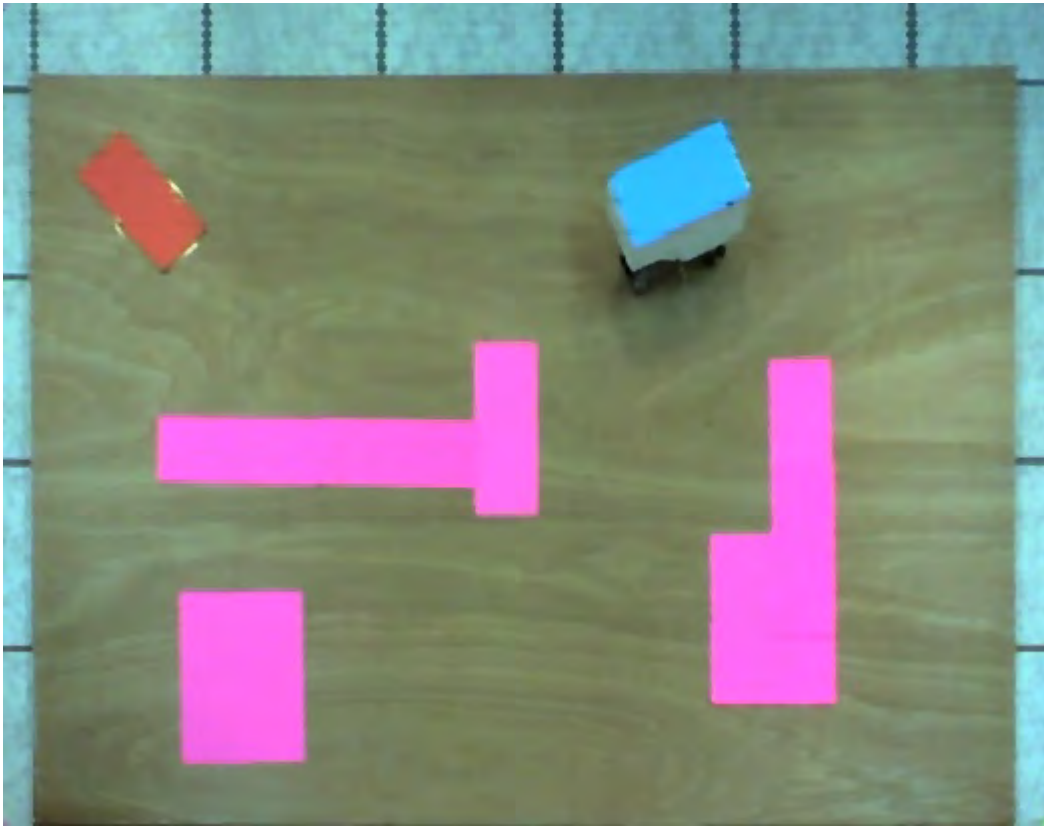


Figura 5.2.7: Imagen sin filtro de la mediana

objetos no muy complejos, donde los objetos no están montados unos sobre otros y con niveles de intensidad uniformes o cuasiuniformes, que permitan distinguirlos del fondo fácilmente. Podemos decir que la etapa de segmentación es, en sí, la verdadera etapa de reconocimiento debido a que a partir de esta etapa se tendrán los objetos perfectamente ubicados en la escena.

La técnica usada para el rastreo de los objetos, se realiza mediante la segmentación por ángulo “hue” del objeto que se desea en el espacio HSV (Hue, Saturation, Value)[13]. La metodología aplicada es mediante la selección por el usuario del nivel óptimo del umbral para segmentación de los objetos a rastrear en la imagen, para la identificación del color fundamental del objeto en el espacio HSV a través de esta selección.

RGB (Red, Blue, Green) es uno de los espacios de color mayormente usados pero HSV es algunas veces preferido para tareas de visión. “Hue” o tono de color en castellano, es una especificación del color intrínseco. “Saturation” o Saturación, describe, ¿Cuán puro

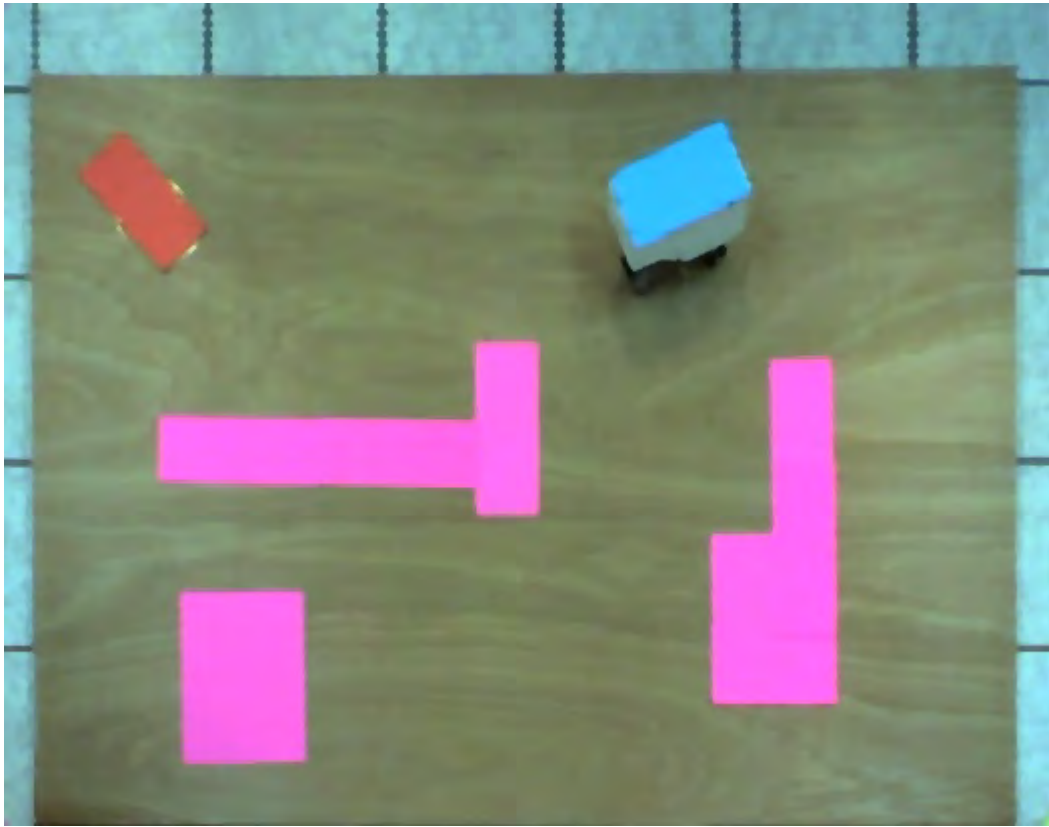


Figura 5.2.8: Imagen con filtro de la mediana

es el color?. La última componente del espacio de color HSV, “Value” o Valor es una medición de ¿cuán brillante es el color?. Los espacios de color HSI (Hue, Saturation, Intensity), HSB (Hue, Saturation, Brightness), y HSL (Hue, Saturation, lightness) son formas variantes del espacio de color HSV [7]. El modelo del espacio de color HSV se representa en la Figura 5.2.9.

HSV ha sido usado mayormente en tareas de visión [41, 6]. Las ventajas de HSV sobre RGB son[6, 7]:

- *Hue* es invariante a ciertos de tipos de realces, sombras y tonalidades.
- La segmentación es realizado sobre una única dimensión (H) y el resultado de la segmentación tiene menos segmentos que al usar RGB.

El primer paso para la segmentación será cambiar el espacio de color RGB a HSV mediante la siguiente transformación:

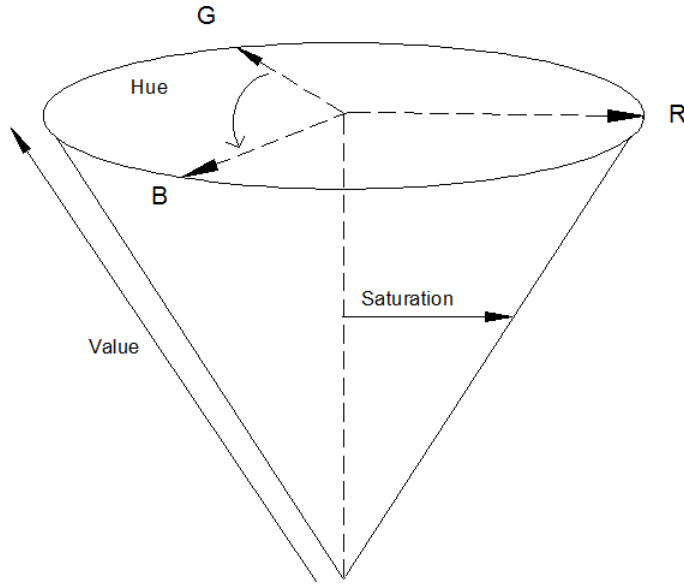


Figura 5.2.9: Representación del espacio de color HSV

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

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

$$V = \{MAX\}$$

En la Figura 5.2.10 se observa la conversión de la escena al espacio HSV. Variando el ángulo del *Hue*, el valor y la saturación se ha ajustado el filtro de colores deseados en la imagen, permitiendo de esta manera una segmentación de la escena para la fácil identificación de los obstáculos y el robot móvil. Observe la Figura 5.2.11 en donde se muestra un ejemplo de la imagen original de la escena, y otra con un filtro del ángulo del *hue* en el espacio de colores HSV, se visualiza sólo la plataforma y se muestra en el espacio de color RGB para mejor visualización. Note que segmentar la imagen después de obtener este resultado se facilita en gran medida.

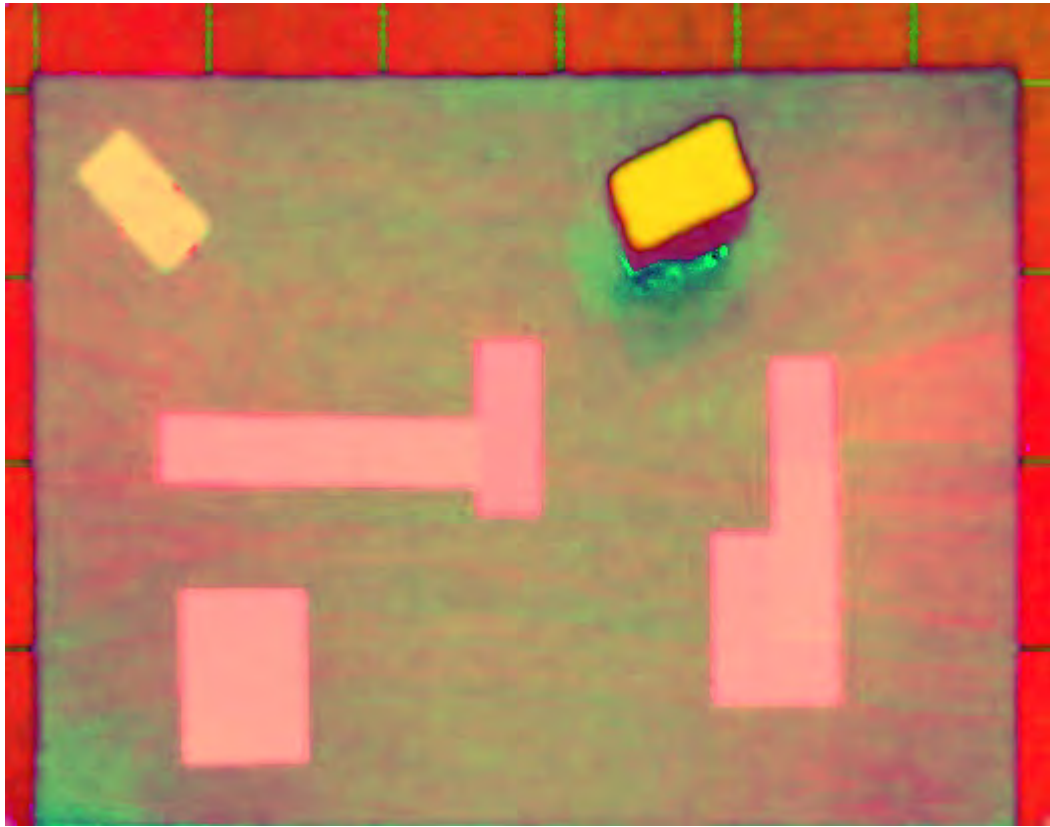


Figura 5.2.10: Conversión de espacio de color RGB a HSV

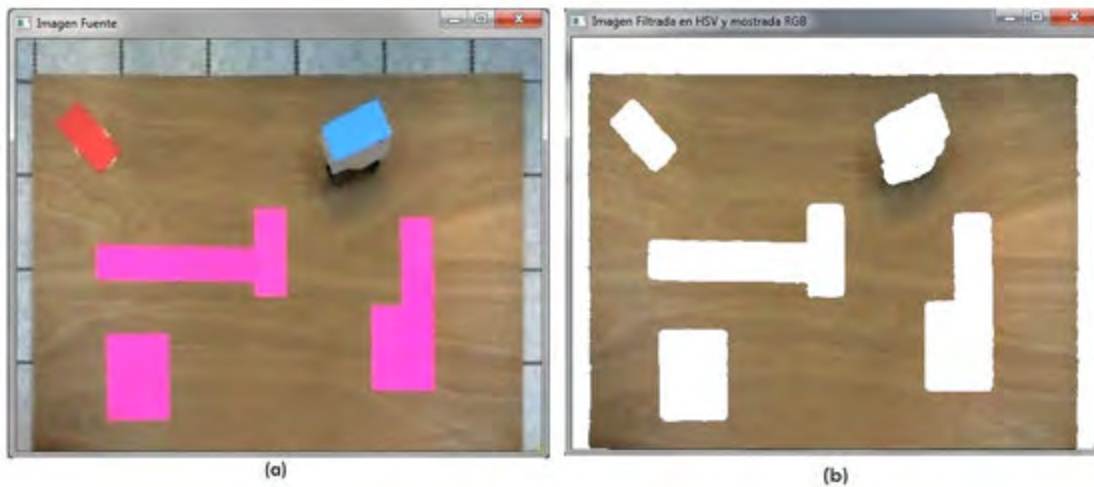


Figura 5.2.11: (a) Imagen Original (b) Imagen filtrada en el espacio de color HSV, se muestra en espacio de color RGB.

5.2.4.3. Umbralización

La umbralización o también conocida como *thresholding* consiste en eliminar los valores superiores o inferiores (poniéndolos a cero) respecto a un valor conocido como **umbral**.

$$ImgB(i, j) = ImgA(i, j) * (ImgA(i, j) \geq umbral) \quad (5.2.2)$$

El umbral que se ha usado para la detección del robot móvil y de los obstáculos es en base al paso de sólo una sección de color, es decir; sólo se permite el paso de ciertos colores limitando el ángulo del *Hue*. Para la detección de los obstáculos se ha utilizado un rango de ángulo de $123^\circ - 152^\circ$ y para el robot móvil un rango de ángulo hue de $5^\circ - 25^\circ$, en la Figura 5.2.11 se observa este rango de ángulo.

5.2.4.4. Binarización

La binarización es una variante de la umbralización y consiste en dejar en cero todos los pixeles menores de un umbral y a uno aquellos que son iguales o mayores, quedando constituida la imagen final por un conjunto de unos y ceros. La implementación del algoritmo queda de la siguiente manera para el caso en el espacio de color HSV:

$$ImgB(i, j) = \begin{cases} 1, & HueMayor \geq ImgA(i, j) \geq HueMenor \\ 0, & \text{en otro caso} \end{cases} \quad (5.2.3)$$

El resultado de la condición será 1 si se encuentra en el rango de umbral, o 0 para otro caso. Observe la Figura 5.2.12 y la donde se muestran los resultados de esta operación.

5.2.4.5. Erosión y Dilatación

Erosión y dilatación son operaciones morfológicas que se encargan de cambiar la forma y estructura de los objetos en un imagen. Está herramienta permitirá obtener componentes que dan una idea de la forma y estructura de los objetos que forman la imagen. Además, permiten modificar estas formar para separar los objetos unos de otros, obtener contornos primarios, descomponer formar complejas en formas más simples,

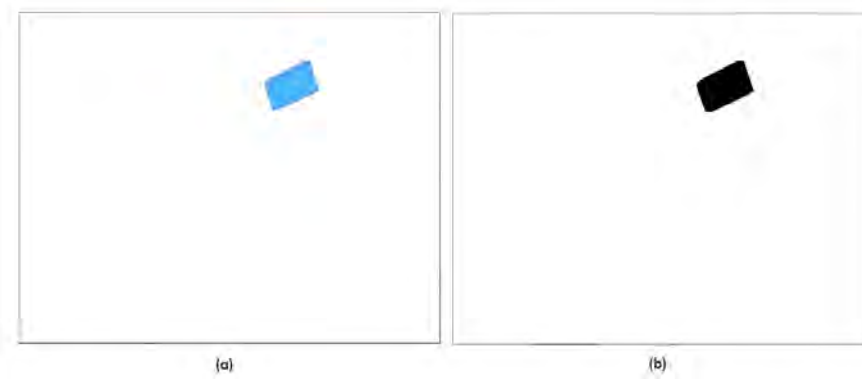


Figura 5.2.12: (a) Imagen filtrada del robot móvil. (b) Imagen binarizada para el robot móvil

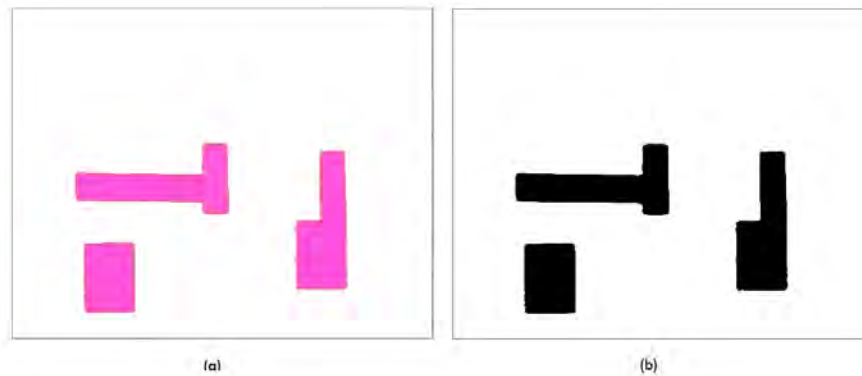


Figura 5.2.13: (a) Imagen filtrada de los obstáculos. (b) Imagen binarizada del robot móvil

obtener contornos dentro de entornos ruidosos, reconstruir elementos distorsionados, entre otros.

Esta serie de transformaciones se realizan en imágenes previamente binarizadas, aunque también existen estudios de aplicación de estas técnicas en imágenes con niveles de grises[13].

- **Dilatación Binaria.** La dilatación es una convolución de una imagen o región de una imagen A con una máscara que llamaremos B , la dilatación de $A \oplus B$ es el conjunto de todos los desplazamientos, tales que B y A se solapen en al menos un elemento distinto de cero.

Por ejemplo, dada una máscara B formada por una matriz:

0	1	0
1	1	1
0	1	0

Figura 5.2.14: Máscara B

y una figura A formada por una matriz:

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Figura 5.2.15: Figura A

El proceso de dilatación $A \oplus B$ consiste en que para cada punto $A(i, j)$ sobreponer la máscara B en la imagen A , siendo el punto central de B el (i, j) , y ver si algún uno de B coincide con algún de A y si es así, colocar el punto $A(i, j)$ a uno. Es decir, la dilatación consiste en ir desplazando la máscara B por toda la imagen, y si alguno de los elementos de la matriz coincide con un pixel del entorno, entonces el pixel donde está centra A se pone a uno. Este crecimiento como se observa en la Figura 5.2.16 y que produce un engrosamiento en los bordes del objeto es el origen del término “operador dilatación”.

- **Erosión Binaria.** La erosión binaria es la función dual de la dilatación pero no es la inversa, es decir; si se hace una erosión y luego una dilatación, la imagen no se queda como anteriormente estaba. La erosión binaria es denotada por la convolución $A \ominus B$.

Si la dilatación expandía los bordes y contornos de los objetos, la erosión reduce los contornos. Se puede utilizar para separar objetos que están unidos por una pequeña parte de sus contornos.

0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Figura 5.2.16: Resultado de $A \oplus B$

Ejemplo, usando la máscara anterior de la Figura 5.2.14 y la siguiente imagen:

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Figura 5.2.17: Imagen A

De forma parecida que el proceso anterior el proceso de erosión $A \ominus B$ consiste en que para cada punto $A(i, j)$ superponer la máscara B en A , siendo el punto centra el centro de B , y checar que todos los puntos a uno de la máscara coincidan con los de la imagen, poniéndolo a uno. En caso contrario se ponen a cero. Dependiendo de la forma de la máscara se puede utilizar para realizar erosión en dirección o direcciones de cierto interés.

Aplicando éstas dos transformaciones morfológicas en la imagen binarizada del resultado de la segmentación, se pueden describir de mejor manera los objetos, es decir; de esta forma se trata de dejar más definido los objetos a identificar.

El resultado de aplicar el operador dilatador y erosión en las imágenes binarizada productos del resultado de la segmentación se observa en laFigura 5.2.19 y Figura 5.2.18.

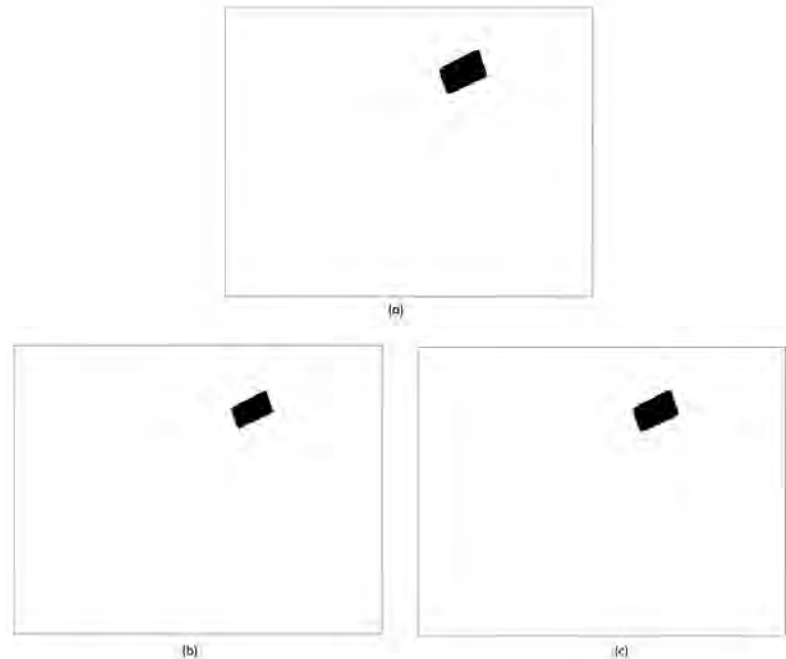


Figura 5.2.18: (a) Imagen binarizada, resultado de la segmentación para el robot móvil. (b) Resultado de aplicar el operador dilatador a la imagen (a). (c) Resultado de aplicar el operador erosión al imagen (b).

Este proceso ha ayudado a corregir y definir la geometría de los objetos para el proceso de identificación que se explica en la subsección siguiente.

5.2.5. Generación del entorno

Una vez obtenido los bordes de los objetos bien definidos después del procesado de la segmentación, el siguiente paso es la obtención de los parámetros que definan las características de cada objeto en la escena como la forma y orientación.

Dentro los parámetros o descriptores que se obtengan, habrá que seleccionar los que tengan las siguientes características:

Discriminantes. Que se diferencien lo mejor posible de una familia o clase con los de las otras.

Independientes entre sí. Los descriptores que definan cada objeto no tienen que estar relacionados, de forma que si varía alguno de ellos los demás no varíen por este.

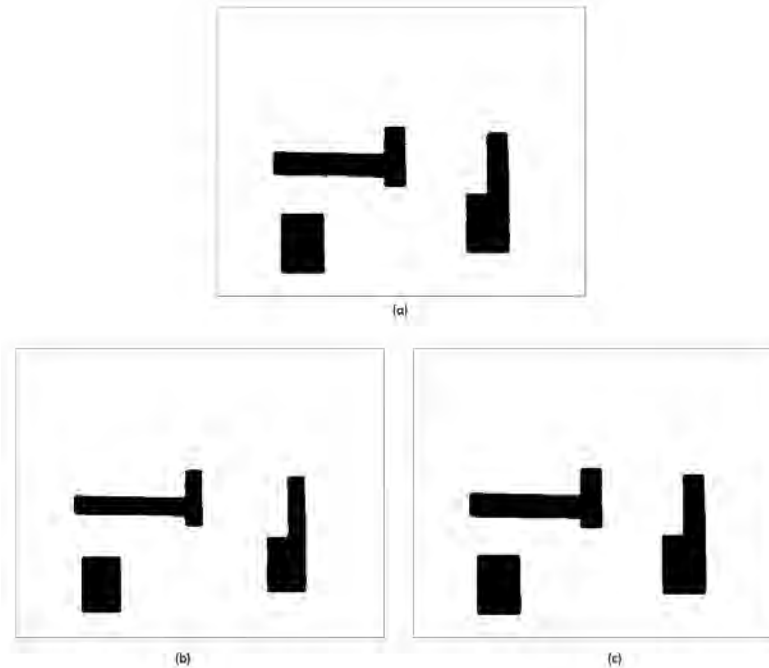


Figura 5.2.19: (a) Imagen binarizada, resultado de la segmentación para los obstáculos de la escena. (b) Resultado de aplicar el operador dilatador a la imagen (a). (c) Resultado de aplicar el operador erosión al imagen (b)

Suficientes. Delimitar de forma suficiente la pertenencia de un objeto a una clase determinada.

Los descriptores se basan en la evaluación de alguna característica del objeto, como:

- Descriptores unidimensionales o de contorno: códigos de cadena, perímetro, forma del perímetro.
- Descriptores bidimensionales o de región: áreas, momentos de inercia, centro de gravedad, etc.
- Descriptores específicos: números de agujeros, área de agujeros, posición relativa de agujeros, rasgos diferenciadores de un objeto, entre otros.

En este trabajo se ataca el problema usando los primeros descriptores, unidimensional y bidimensional. Se reduce el problema en la condición de que los objetos sólidos, es decir; no tienen agujeros.

Generalmente una imagen binaria es representada como una matriz que contiene información del objeto y el fondo. La representación más compacta de la imagen se

puede obtener al almacenar en una estructura de datos la información de los píxeles que acotan al objeto. Las técnicas basadas en la utilización de estructuras de datos, para almacenar los puntos que acotan al objetos se llaman *técnicas de codificación de contornos*.

Se hace uso de un algoritmo de seguimiento de contornos llamado *código cadena* para la extracción de contornos de la imagen. Los códigos cadena son usados para definir, a partir de los puntos del contorno de un objeto, las direcciones que van tomando el conjunto de segmentos que forman la frontera del objeto. En la práctica se suelen utilizar el *código 4* (usado en la implementación) o el *código 8*, aunque se pueden utilizar códigos con más direcciones. En la Figura 5.2.20 se muestra dos códigos cadena y la Figura 5.2.21 muestra la codificación de una frontera.

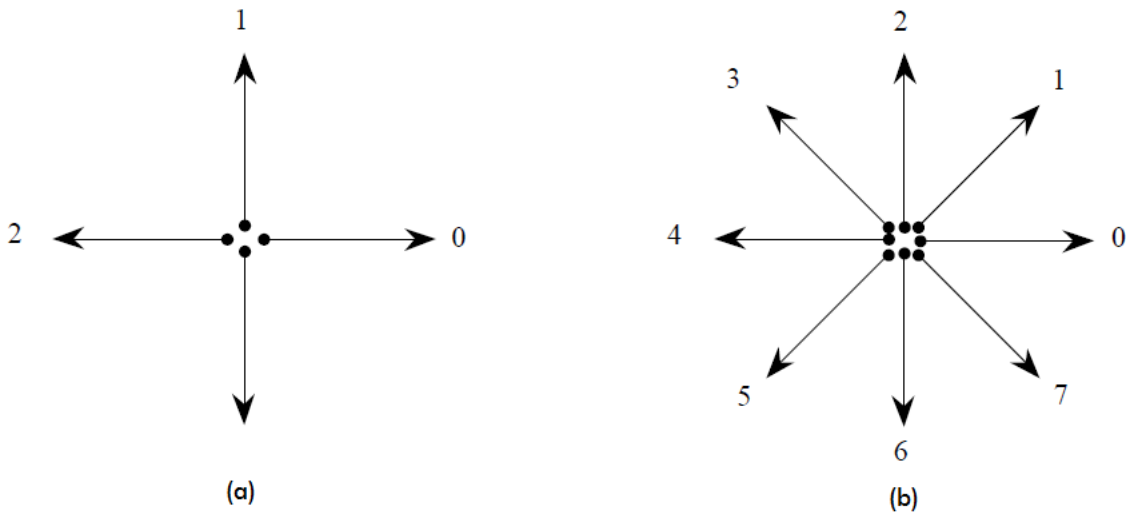


Figura 5.2.20: Códigos de cadena. (a) Cuatro direcciones. (b) Ocho direcciones.

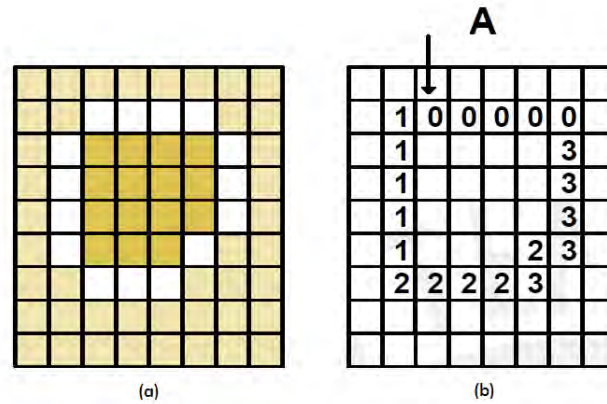


Figura 5.2.21: Codificación de una frontera. (a) Frontera. (b) Conectividad usando código 4 direcciones: A00000333323222211111.

5.2.5.1. Detección de obstáculos

Haciendo uso de la técnica de codificación de contorno que ya está implementado en OpenCV, se identifican los obstáculos en la imagen binarizada producto de la segmentación para los obstáculos, almacenando estos datos en una secuencia de puntos que será útil para evitar la colisión al momento de generar la planeación de movimiento.

En la Figura 5.2.22 se muestra el resultado de dibujar los contornos obtenidos a partir de la imagen binarizada de los obstáculos.

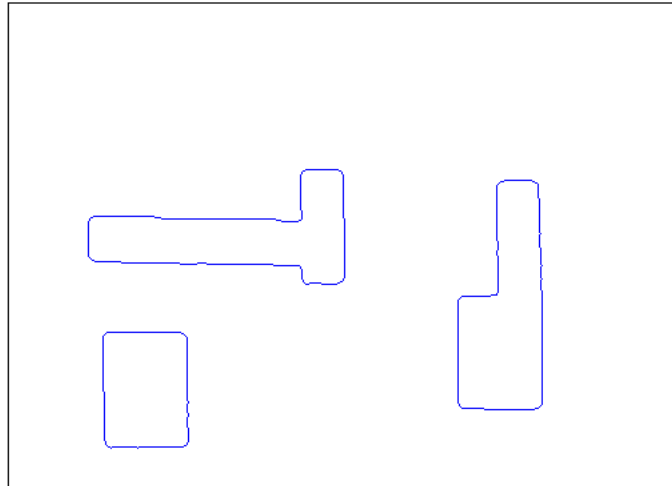


Figura 5.2.22: Detección de los contornos de los objetos en la escena. *Resultado de la obtención de contornos.*

5.2.5.2. Detección del Robot Móvil

Para la detección del robot móvil se obtendrá la forma aproximada de este para obtener una orientación certera, para su uso en la implementación de la planeación de trayectoria y el control del Robot Móvil. La forma geométrica del robot móvil es un rectángulo por lo que es necesario aproximar la misma forma geométrica, es decir; cuatro puntos que indicarán la posición y orientación del robot móvil en el espacio de la escena.

El paso para atacar el problema de la detección del robot móvil se ha realizado ajustando una *caja de contorno* que corresponde a un rectángulo que se ajusta al contorno del objeto, puesto que el objeto es un rectángulo también, se toma la caja de contorno para representación del robot. La caja de contorno está formado por las coordenadas $XMIN$, $YMIN$, $XMAX$ y $YMAX$.

Realizando los mismos pasos para la detección de obstáculos, una vez obtenido el contorno que identifica al robot móvil se hace una aproximación lineal que capture la esencia del contorno. En el siguiente recuadro se muestra el algoritmo desarrollado para determinar la ubicación del robot móvil.

```

//Esta función determina los cuatro puntos que conforman el área del robot móvil.
//@param imgDST Imágen para determinar el área.
//@param sqarea Secuencia del contorno del robot móvil en la imágen
void reconstructor::area(IplImage *imgDST, CvSeq* sqarea){
    CvPoint pt0, ptt; CvBox2D box; CvPoint2D32f box_pt[4];
    int z;
    //Determinando un rectángulo circunscrito del área mínima para el punto 2D.
    box = cvMinAreaRect2( sqarea, 0 );
    //Calcula los vértices del cuadro de entrada 2d.
    cvBoxPoints( box, box_pt );
    pt0.x = cvRound(0.94*(box_pt[3].x)+18);
    pt0.y = cvRound(0.936*(box_pt[3].y-7)+41);
    //Ajustando los puntos y guardando en la variable global Points[z]
    for( z = 0; z < 4; z++ ) {
        box_pt[z].y = 0.936*(box_pt[z].y-7)+41;
        box_pt[z].x = 0.94*(box_pt[z].x)+18;
        ptt.x = cvRound(box_pt[z].x);
        ptt.y = cvRound(box_pt[z].y);
        Points[z].x = ptt.x; Points[z].y = ptt.y;
    }
    //Dibujando la proyección del borde superior del robot móvil en la imagen
    cvLine(imgDST, pt0, ptt, CV_RGB(0, 255, 0), 1, CV_AA, 0);
    pt0 = ptt;
}
}

```

En la Figura 5.2.23 podemos observar el resultado del algoritmo anterior, se observa un rectángulo de color verde que muestra la posición real del robot móvil sobre el piso del escenario.



Figura 5.2.23: Ubicación del robot móvil y proyección al piso de la escena.

5.2.6. Planeación de Trayectoria

De acuerdo a las subsecciones anteriores tenemos todo lo necesario para comenzar a implementar nuestras funciones para el desarrollo de planeación de trayectoria y de movimiento, las cuales se anexan en el CD adjunto a este trabajo.

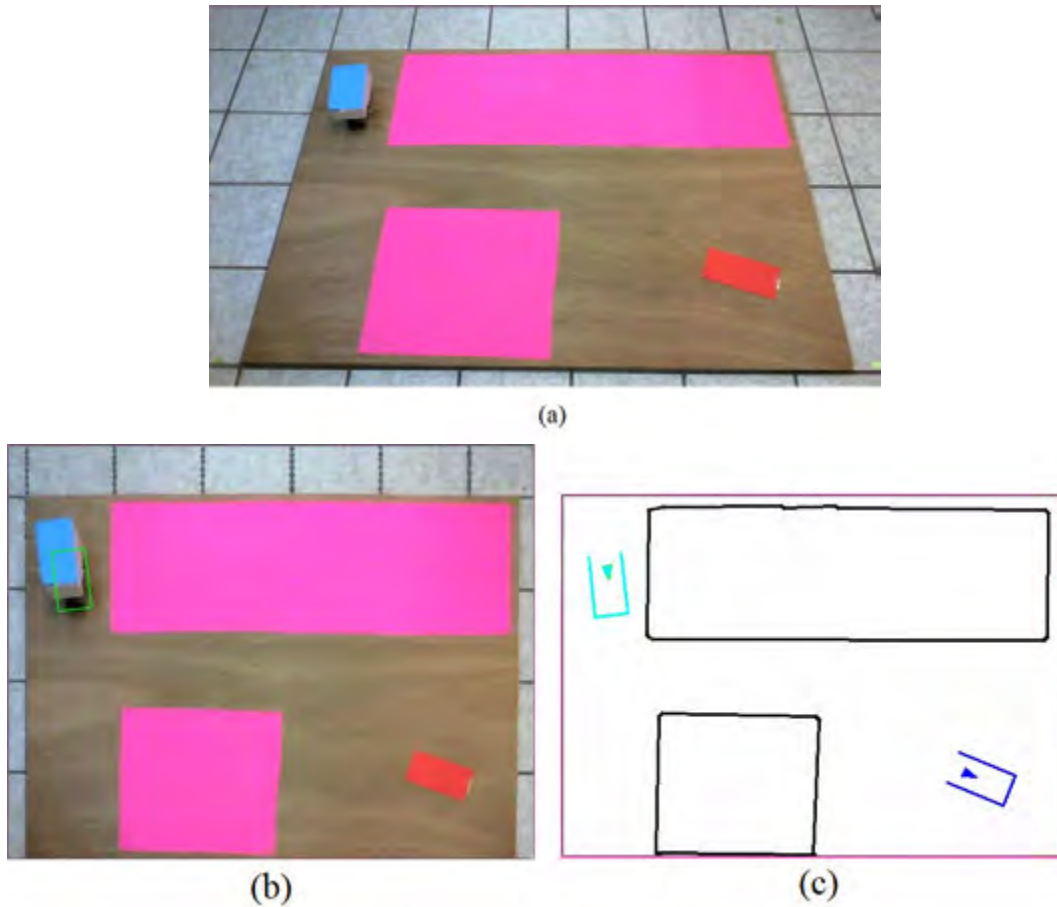


Figura 5.2.24: Generando entorno Virtual. (a) Imagen de entrada desde la cámara. (b) Imagen con corrección de perspectiva. (c) Entorno virtual generado.

Observemos la Figura 5.2.24, en la imagen se puede observar la escena original tomada desde la cámara, observamos los obstáculos de forma representativa de color rosado, el robot móvil de color azul y de color nuestro punto de meta donde el robot móvil de llegar. La imagen (b) se observa la escena con la corrección de perspectiva deseada idea para nuestra aplicación, y por último la imagen (c) el entorno virtual generado para visualizar y comparar el avance del robot con la ruta planeada.

Nuestro entorno virtual será nuestra herramienta principal de trabajo puesto que es el resultado de todo el proceso de detección de obstáculos, detección del robot móvil, y donde visualizaremos el avance a lo largo de la ruta planeada para nuestros fines.

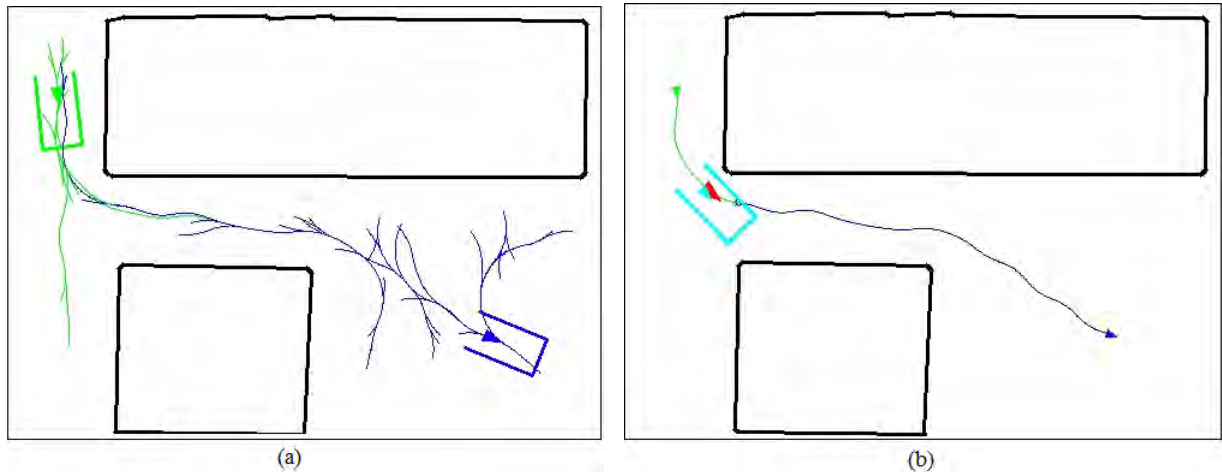


Figura 5.2.25: Generando Planeación de Trayectoria

Una vez que teniendo nuestro entorno virtual generado comenzamos a generar nuestra planeación de ruta, en la Figura 5.2.25 (a) se puede observar como hacemos crecer nuestro RRT con las condiciones del modelo cinemático de nuestro robot móvil, (véase sección 4.2). Para facilitar la generación de la ruta hacemos crecer el RRT desde el punto de inicio y el punto de meta, así cuando convergen las dos en un punto tendremos una ruta generada. Esto agiliza el proceso de nuestro RRT ya que generandolo desde solo un punto tardaría más en converger a nuestro punto de meta. En la imagen (b) se observa la ruta generada, la ruta en color de verde en la cual se hizo crecer el RRT desde el inicio y en azul la ruta que se hizo crecer desde el punto de meta. Es necesario recalcar que en nuestro sistema es posible decidir la posición final del robot móvil y dirección que se desee.

La Figura 5.2.26 muestra el progreso del seguimiento de trayectoria del robot móvil hasta la posición de meta en dirección de frente. La Figura 5.2.27 muestra como la única forma de llegar a posición de meta sería en dirección reversa.

En la Figura 5.2.28 se observa que el robot móvil se desplaza en reversa hasta la posición de meta, esto hace muy versátil a nuestro sistema ya que permite estacionarse también en paralelo usando las dos dirección, frontal y reversa.

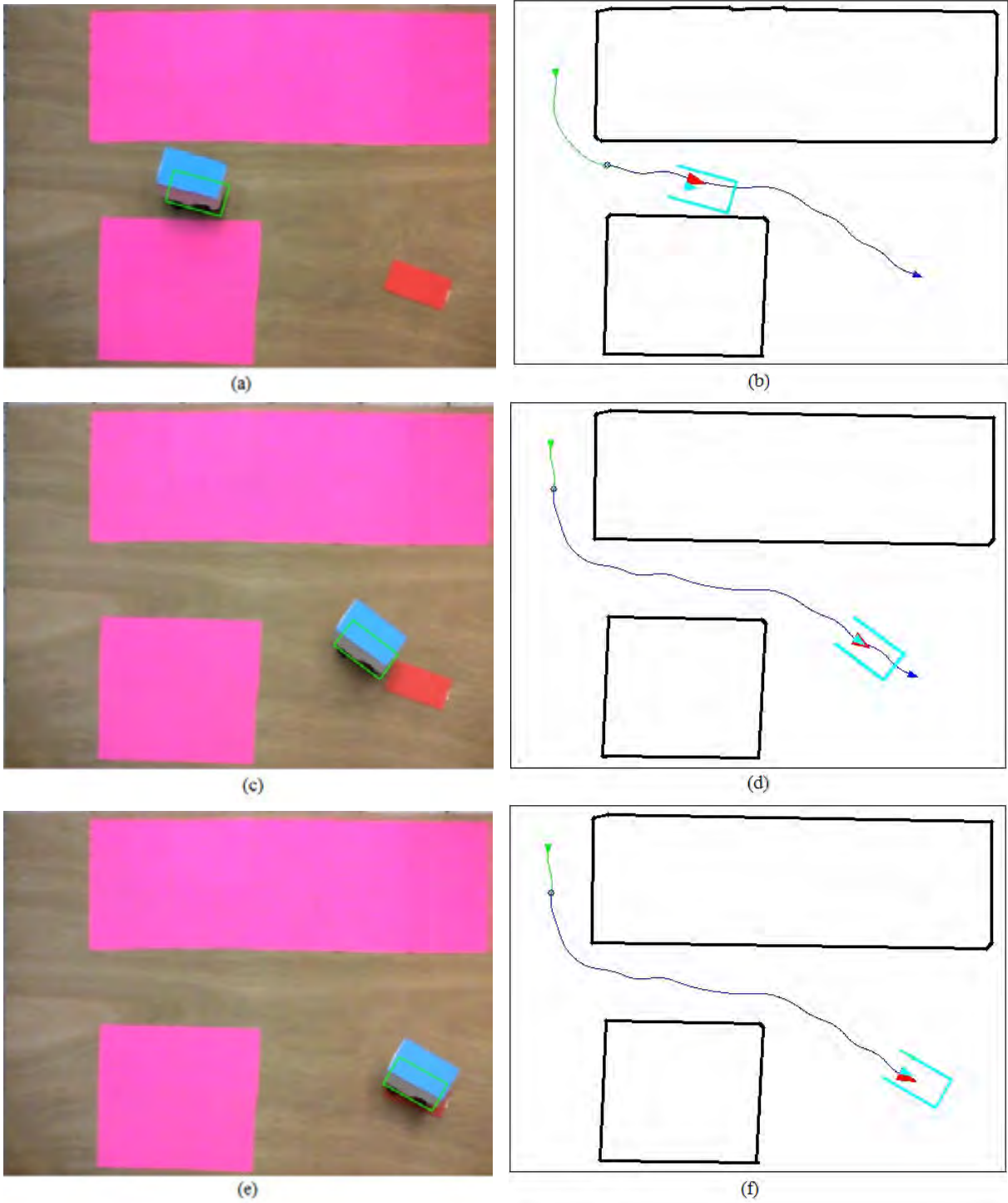


Figura 5.2.26: Seguimiento de Trayectoria.

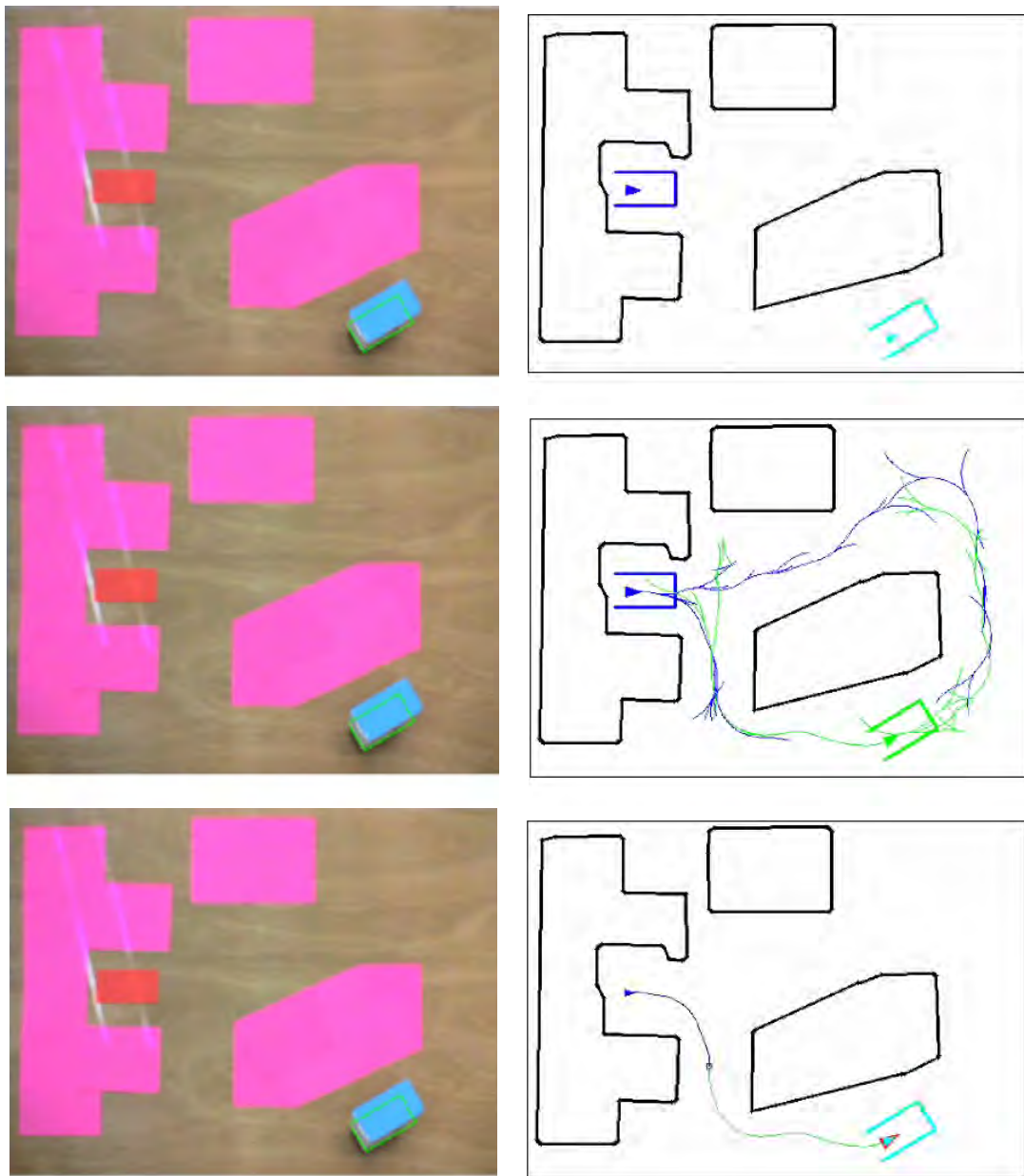


Figura 5.2.27: Generando Planeación de Trayectoria del Arreglo 2

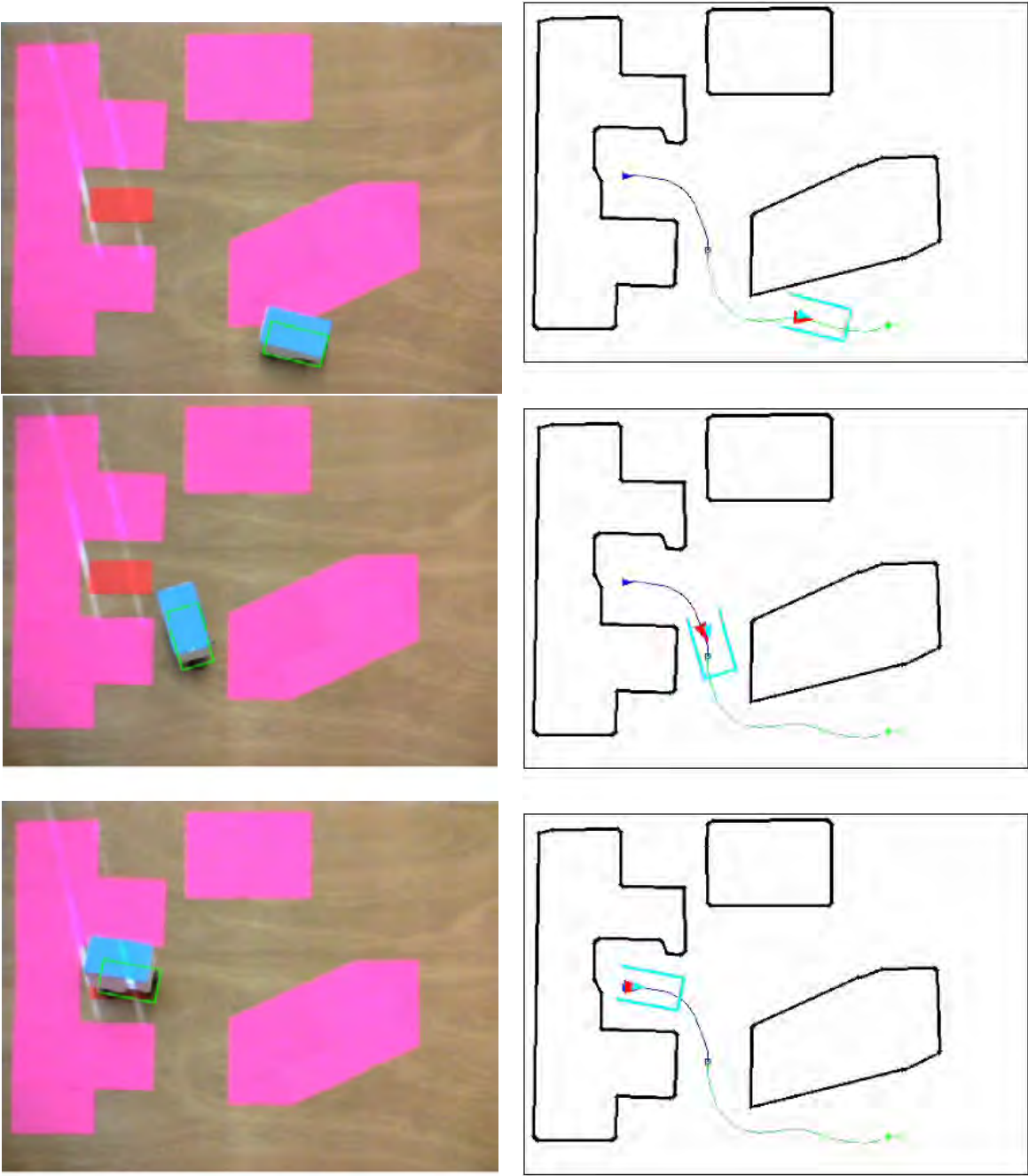


Figura 5.2.28: Seguimiento de Trayectoria del Arreglo 2

6 Conclusiones

En este trabajo como objetivo de desarrollo tecnológico se ha implementado y desarrollado un robot móvil de bajo costo para la navegación, que permite seguir la ruta generada por una planeación de trayectoria basada exploración rápida de arboles aleatorios o por sus siglas en inglés RRT (Rapidly-exploring Random Trees) controlado por un sistema de visión artificial. El principal interés se encuentra en la versatilidad y funcionamiento de un sistema de visión artificial y el desarrollo de un robot todo a bajo costo para múltiples propósitos.

Se han seguido técnicas de visión por computadora para la localización del robot a partir de imágenes capturadas por una cámara web ordinaria, es decir; haciendo uso de las capacidades de visión simple.

Los algoritmos usados para el procesamiento de las imágenes en este trabajo han permitido satisfactoriamente una adecuada aproximación de la descripción de las características de los objetos y el robot móvil, todo en tiempo real, pues bien se sabe; que en la implementación de algoritmos de procesamiento de imágenes el tiempo de procesamiento es un factor importante para tareas en tiempo real.

También se logró el objetivo de una planeación de trayectoria más natural para el robot móvil de tal manera que el desplazamiento del robot se asemeje al de un vehículo controlado por un humano.

Se han seguido técnicas de visión artificial para la localización del robot a partir de imágenes capturadas por una cámara web ordinaria, es decir; haciendo uso de las capacidades de visión simple.

Los algoritmos usados para el procesamiento de las imágenes en este trabajo han permitido satisfactoriamente una adecuada aproximación de la descripción de las características de los objetos y el robot móvil, todo en tiempo real, pues bien se sabe;

que en la implementación de algoritmos de procesamiento de imágenes el tiempo de procesamiento es un factor importante para tareas en tiempo real.

También se logró el objetivo de una planeación de trayectoria más natural para el robot móvil, de tal manera que el desplazamiento del robot se asemeje al de un vehículo controlado por un humano.

Se ha realizado con éxito la integración del sistema de visión implementado junto con el robot móvil y la parte de planeación de trayectoria. Aunque no dispone de un aplicación específica, este trabajo permitirá desarrollo de aplicaciones más complejas dando mayor importancia al control por visión.

7 Trabajo a futuro

Se ha visto como ejemplo este trabajo que la visión artificial permite automatizar y mejorar muchos procesos, como el movimiento del robot móvil.

Una de las cosas para trabajos a futuros es agregar más cámaras para una reconstrucción tridimensional del espacio de trabajo, de esta forma se podrá observar profundidad y tener una mejor perspectiva del espacio, además que con múltiples cámaras se podrían evitar oclusiones.

Futuros estudios irán dirigidos a otras técnicas de planeación de trayectoria y control de ella, como la aplicación de lógica difusa o redes neuronales de manera que se logre comparar y reducir tasas de fallos, incrementar la optimización de la planeación de trayectoria y el control hasta llegar a desarrollar un algoritmo completamente operativo.

Bibliografía

- [1] E. Mazer J.M Ahuactzin. The ariadne's clew algorithm. *J. Artificial Intell. Res.*, pages 295–316, 1998.
- [2] Pedro E. López-de-Teruel Alberto Ruiz and Lorenzo Fernández. Practical planar metric rectification. 2011.
- [3] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*,, 37(8):855–866, 1971.
- [4] M. C. Lin J. F. Canny. Efficient algorithms for incremental distance computation. In *Proc. IEEE Int'l Conf. on Robotics and Automatization*, 1991.
- [5] K.R. Castleman. *Digital Image Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1996.
- [6] H. D. Cheng and Y. Sun. A hiererchical approach to color image segmentation using homogeneity. *IEE Trans. Image Processing*, 9(12):2071–2082, 2000.
- [7] H.D. Cheng. Color image segmentation: Advances adn prospects. *Pattern Recognition*, 34:2259–2281, 2001.
- [8] A. González Marcos F. J. Martínez de Pisón A. V. Pernía Espinoza. *Técnicas y algoritmos básicos de Visión Artificial*. Universidad de la Rioja, 2006.
- [9] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12):1479–1486, 1975.
- [10] Olivier Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, London, England, 1993.
- [11] Y. Yu K. Gupta. On sensor-based roadmap: A framework for motion plannin for a manipulator arm in unknown enviroments. *IEEE/RSJ int. Conf on Intelligent Robots and Systems*, pages 1919–1924, 1998.

- [12] Tobias Hanning. *High Precision camera calibration*. Vieweg Teubner, 2009.
- [13] R.M. Haralick and L.G. Shapiro. *Computer and robot vision*. Addison-Wesley Publishing Co, New York, 1992.
- [14] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2010.
- [15] P. Cohen J. Weng and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992.
- [16] T. Kailath. *Linear Systems*. Prentice-Hall, 1980.
- [17] J. Kim and J. P. Ostrowski. Motion planning of aerial robot using rapidly exploring random trees with dynamic constraints. *Proceedings IEEE International Conference on Robotics & Automation*, 2003.
- [18] Steven M. LaValle James J. Kuffner. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and Computational Robotics: New Directions*, pages 293–08, 2001.
- [19] D. Hsu L. J. Guibas and L. Zhang. H-walk: Hierarchical distance computation for moving convex bodies. *Proc. ACM symposium on Computational Geometry*, 1999.
- [20] J.-C. Latombe. Robot motion planning. *Kluwer Academic Publishers*, 1991.
- [21] Steven M. LaValle. Rapidly-exploring random trees: A new tool for path planning. *Computer Science dept. Iowa University*, 1999.
- [22] J. Llinares. *Electromagnetismo y semiconductores*. Universidad Politécnica de Valencia.
- [23] Timothy J. Maloney. *Electrónica Industrial Moderna*. Pearson Education, 2006.
- [24] J. More. The levenberg-marquardt algorithm, implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, volume Lecture Notes in Mathematics, page 630, Springer-Verlag, 1977.
- [25] D. Hsu J.-C. Latombe R. Motwani. Aproximate nearest neighbors: Towards removing the curse of dimensionality. *Proceedings of the 30th Annual ACM symposium on Theory of Computing*, 1998.

- [26] Steven M. LaValle D. Lin L.J. Guibas J.C. Latombe R. Motwani. Finding unpredictable target in a workspace with obstacles. *Proceedings of the IEEE International Conference on Robotics and Automatization*, pages 737–742, 1997.
- [27] Q.-T. Luong O. Faugeras and T. Papadopoulos. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. The MIT Press, Cambridge MA, London, 2001.
- [28] Javier Ruiz del Solar Pablo Guerrero. Auto-localización de un robot móvil aibo mediante el método de monte carlo. *Dept. Ing. Eléctrica,*, Casilla 412-3, 2004.
- [29] Ioannis Pitas. *Digital Image Processing Algorithms and Applications*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2000.
- [30] J. A Muñoz Rodríguez. Calibration modeling for mobile vision based laser imaging and approximation networks. *Journal of Modern Optics*, 57:1583–1597, 2010.
- [31] J. A. Muñoz Rodríguez. Online self-calibration for mobile vision based on laser imaging and computer algorithms. *Optics and Lasers in Engineering*, 49:680–692, 2011.
- [32] Alain Lambert Romain Pepy and Hugues Mounier. Path planning using a dynamic vehicle model. *Insitut d’Électronique Fondamentale*.
- [33] A. De Luca G. Oriolo C. Samson. *Robot Motion Planning and control*. Lectures Notes in Control and Information Sciences 229.
- [34] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA, 1983.
- [35] Sakti Pramanik Shamik Sural, Gang Qian. Segmentation and histogram generation using the hsv color space for image retrieval. *Dept. of Computer Science and Engineering*.
- [36] C.C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, fourth edition, 1980.
- [37] González Aguilera Gómez Lahoz. Sánchez Martín, Arias Pérez. Analisis aplicado de métodos de calibración de cámaras para usos fotogramétricos. In *Topcart*, 2004.
- [38] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, August 1987.

- [39] Hanzi Wang and David Suter. Color image segmentation using global informatin and local homogeneity. *Departament of Electrical and Computer Systems Engineering Monash University, Clayton 3800, Victoria, Australia*.
- [40] G. Wei and S. Ma. Implicit and explicit camera calibration: Theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):469–480, 1994.
- [41] C. Zhang and P. Wang. A new method of color image segmentation based on intensity and hue clustering. In *International Conference on Pattern Recognition (ICPR)*, pages 613–616, 2000.
- [42] Zhengyou Zhang. Flexible camara calibration by viewing a plane from unknow orientations. In *Proceedings of the 7th International Conference on Computer Vision*, pages 666–673, Corfu, september 1999.
- [43] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000.