



**INSTITUTO POLITÉCNICO NACIONAL**

**CENTRO DE INVESTIGACIÓN Y DESARROLLO DE  
TECNOLOGÍA DIGITAL**



**Visión computacional para instrumentación y navegación asistida  
usando métodos opto-digitales**

**TESIS**

**QUE PARA OBTENER EL GRADO DE  
MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES**

**PRESENTA**

**ING. JUAN ZHENG WU**

**BAJO LA DIRECCIÓN DE**

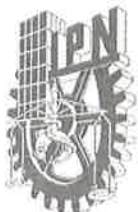
**DR. RIGOBERTO JUÁREZ SALAZAR**

**DR. VÍCTOR HUGO DÍAZ RAMÍREZ**

**Tijuana, Baja California, México**

**Julio 2020**





# INSTITUTO POLITÉCNICO NACIONAL

## SECRETARIA DE INVESTIGACIÓN Y POSGRADO

### ACTA DE REGISTRO DE TEMA DE TESIS Y DESIGNACIÓN DE DIRECTORES DE TESIS

Tijuana, B.C. a 17 de diciembre del 2018

El Colegio de Profesores de Estudios de Posgrado e Investigación de CITEDI en su sesión ordinaria No. 12/18 celebrada el día 13 del mes de diciembre, 2018 conoció la solicitud presentada por el(la) alumno(a):

**ZHENG**

Apellido paterno

**WU**

Apellido materno

**JUAN**

Nombre (s)

Con registro: **B 1 8 0 3 8 3**

Aspirante de: **MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES**

1.- Se designa al aspirante el tema de tesis titulado:

**Visión computacional para instrumentación y navegación asistida usando métodos opto-digitales.**

De manera general el tema abarcará los siguientes aspectos:

*Desarrollo de técnicas de visión computacional optimizando métricas objetivas para aplicaciones de instrumentación óptica y navegación asistida empleando métodos opto-digitales.*

2.- Se designa como Directores de Tesis a los Profesores:

**DR. RIGOBERTO JUÁREZ SALAZAR Y DR. VÍCTOR HUGO DÍAZ RAMÍREZ**

3.- El trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:

**CENTRO DE INVESTIGACIÓN DE DESARROLLO DE TECNOLOGÍA DIGITAL**

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente hasta la aceptación de la tesis por la Comisión Revisora correspondiente:

Directores de Tesis

  
DR. RIGOBERTO JUÁREZ SALAZAR

  
DR. VÍCTOR HUGO DÍAZ RAMÍREZ

Aspirante

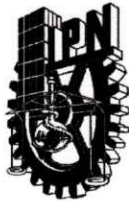
  
ING. JUAN ZHENG WU

Presidente del Colegio

  
DR. JULIO CÉSAR ROLÓN GARRIDO

INSTITUTO POLITÉCNICO NACIONAL  
CENTRO DE INVESTIGACIÓN Y DESARROLLO  
DE TECNOLOGÍA DIGITAL  
DIRECCIÓN





# INSTITUTO POLITÉCNICO NACIONAL

## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

SIP-14  
REP 2017

### ACTA DE REVISIÓN DE TESIS

En la Ciudad de Tijuana B.C. siendo las 17:00 horas del día 15 del mes de julio del 2020 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: CITEDI para examinar la tesis titulada:  
Visión computacional para instrumentación y navegación asistida usando métodos opto-digitales. (la) alumno (a):

Apellido Paterno:	ZHENG	Apellido Materno:	WU	Nombre (s):	JUAN
-------------------	-------	-------------------	----	-------------	------

Número de registro: B 1 8 0 3 8 3

Aspirante del Programa Académico de Posgrado: MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 13 % de similitud. **Se adjunta reporte de software utilizado.**  
Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo **SI** ☐ **NO** ☒ **SE CONSTITUYE UN POSIBLE PLAGIO.**

#### **JUSTIFICACIÓN DE LA CONCLUSIÓN:** *(Por ejemplo, el % de similitud se localiza en metodologías adecuadamente referidas a fuente original)*

La herramienta Turnitin arrojó un porcentaje de similitud del 13% del cual el 10% corresponde a fuentes de internet, el 7% a publicaciones, y el 8% a trabajos del estudiante. Se observa que ninguna de las fuentes primarias arroja más del 1% de similitud. Con base a lo anterior, se concluye que el trabajo del estudiante es original.

**\*\*Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**


Finalmente y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** ☒ **SUSPENDER** ☐ **NO APROBAR** ☐ la tesis por **UNANIMIDAD** ☒ o **MAYORÍA** ☐ en virtud de los motivos siguientes:

El estudiante cumplió en tiempo y forma con lo reglamentado.

Se alcanzó el objetivo general y objetivos específicos de la tesis.

El estudiante atendió las revisiones indicadas por el comité tutorial.

#### **COMISIÓN REVISORA DE TESIS**

  
Dr. Roberito Juárez Salazar  
Director de Tesis  
Nombre completo y firma

Dr. Víctor Hugo Díaz Ramírez  
2° Director de Tesis (en su caso)  
Nombre completo y firma

Dr. Eduardo Javier Moreno Valenzuela  
Nombre completo y firma

Dr. Ricardo Ramón Pérez Alcocer  
Nombre completo y firma

M. en C. Andrés Calvillo Téllez  
Nombre completo y firma

Dr. Julio César Rolón Garrido  
Nombre completo y firma  
**PRESIDENTE DEL COLEGIO DE PROFESORES**





## **INSTITUTO POLITÉCNICO NACIONAL**

### **SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

#### **CARTA CESIÓN DE DERECHOS**

En la Ciudad de **Tijuana**, Baja California, el día **15** del mes de **julio** del año **2020**, el que suscribe **Juan Zheng Wu**, alumno del Programa de **MAESTRÍA EN CIENCIAS EN SISTEMAS DIGITALES**, con número de registro **B180383**, adscrito(a) al **CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGÍA DIGITAL**, manifiesta que es el autor intelectual del presente trabajo de Tesis bajo la dirección de **Dr. Rigoberto Juárez Salazar** y **Dr. Víctor Hugo Díaz Ramírez** y cede los derechos del trabajo titulado **Visión computacional para instrumentación y navegación asistida usando métodos opto-digitales**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del (de la) autor(a) y/o director(es) del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones **Av. Instituto Politécnico Nacional No. 1310 Col Nueva Tijuana, Tijuana, Baja California, México**, correo electrónico de contacto: **posgrado@citedi.mx**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

---

**Juan Zheng Wu**





## Agradecimientos

En primer lugar, quiero agradecer a mi familiar por el apoyo brindado y por haber permitido la dedicación al trabajo de tesis.

Asimismo, quiero agradecer a mis asesores, el Dr. Rigoberto Juárez Salazar y al Dr. Víctor Hugo Díaz Ramírez por toda la atención incondicional, por compartir sus conocimientos y sus experiencias, por toda la retroalimentación recibida.

A mi comite tutorial, el Dr. Eduardo Javier Moreno Valenzuela, Dr. Ricardo Ramón Pérez Alcocer y al M.C. Andrés Calvillo Téllez por la disposición de su tiempo y por la retroalimentación recibida que permitieron concluir este trabajo de tesis.

Al Centro de Investigación y Desarrollo de Tecnología Digital, Instituto Politécnico Nacional, por abrir sus puertas para continuar mi formación académica. Agradezco a los profesores, por brindarme sus conocimientos, al personal administrativo, por su buen trato y la atención dada en las solicitudes, al personal de intendencia, por mantener siempre limpio el centro y el área de trabajo, al personal de seguridad, por mantener la seguridad del centro.

Quiero agradecer a mi equipo de trabajo, Ing. Alejandra Jocelyn Girón Vázquez, Ing. Mack Orlando Lugo Aguilar, M. C. Obed Isaí Ríos Orellana, M. C. José Enrique Hernández Beltrán y a M. C. Juana Martínez Laguna, por sus consejos y motivaciones para ser un mejor investigador. De la misma manera, agradezco a todos mis compañeros del centro, por la amistad brindada.

Por último, un agradecimiento especial al Consejo Nacional de Ciencia y Tecnología por el apoyo económico recibido durante la realización de este trabajo.



## Dedicatoria

A mi familia por su apoyo incondicional.



## Visión computacional para instrumentación y navegación asistida usando métodos opto-digitales

### Resumen

Navegación visual es un problema de investigación abierto que se puede abordar usando métodos de visión por computadora. Cada día, las aplicaciones basadas en visión son más comunes; por ejemplo, la navegación autónoma, localización de objetos, seguidor de rutas y construcciones de mapa tridimensional, entre otras. Los sistemas opto-digitales poseen grandes ventajas para el desarrollo de aplicaciones de visión por computadora. Su configuración integra un sistema óptico con un medio de adquisición de imágenes y procesamiento digital. En la actualidad, la captura de imágenes se realiza de forma rápida empleando cámaras digitales. También, las computadoras digitales permiten que el procesamiento de las imágenes capturadas pueda realizarse a altas velocidades y de forma mucho más flexible. En este trabajo de tesis se propone un método opto-digital para la estimación de posición y orientación de un robot móvil terrestre. Primeramente, se realiza una detección de contornos en la escena. Esta información permite realizar una detección de líneas de carril, a través de la transformada de Hough. Posteriormente, se determina la posición y orientación del vehículo a partir de las líneas detectadas. Se utilizan máscaras digitales para limitar la información no útil que interfiere a la detección de líneas. Asimismo, se realiza una implementación paralela de los algoritmos en desarrollados para reducir el tiempo de respuesta en el procesamiento digital de datos. El método propuesto es evaluado a través de criterios de desempeño objetivos al procesar secuencias de video tomadas del mundo real. Los resultados obtenidos mostraron que los métodos opto-digitales son factibles para aplicaciones de navegación visual e instrumentación óptica en vehículos reales.

**Palabras clave:** Estimación de posición y orientación, detección de carril, navegación visual, homografía, transformación geométrica, corrección de distorsión, transformada Hough, cámara pinhole, calibración de cámara, visión computacional.



## Computer vision for instrumentation and assisted navigation using opto-digital methods

### Abstract

Visual navigation is an open research problem that can be addressed using computer vision methods. Computer vision applications are more common in everyday life such as autonomous navigation, object localization, path follower, map construction, and amongst others. Opto-digital systems offer great advantages for the development of computer vision applications. Typical opto-digital configurations integrate an optical system with a digital image acquisition and processing unit. Nowadays, digital cameras can capture images at high speeds. On the other hand, digital computers allow the processing of the acquired images to be performed at high speeds in a much more flexible way. In this work, an opto-digital method to estimate the position and orientation of a mobile robot is proposed. First, an edge detection method is used to extract the edges of the scene. This information is useful to detect lane lines with Hough transform. A binary window mask limits non-useful information that interferes with line detection. Additionally, a parallel implementation of the developed algorithms was used to speed up the digital processing time. The efficiency of the proposed method is evaluated through an objective performance criteria by processing video sequences taken from the real world. The experimental results showed that opto-digital methods are feasible for visual navigation and optical instrumentation applications.

**Keywords:** Position and orientation estimation, lane detection, visual navigation, homography, geometric transformation, lens distortion correction, Hough transform, pinhole camera, camera calibration, computer vision.





# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Visión por computadora	1
1.2. Instrumentación para navegación asistida	2
1.3. Sistemas opto-digitales	3
1.4. Objetivos	4
1.4.1. Objetivo general	4
1.4.2. Objetivos específicos	4
1.5. Contribuciones	4
<b>2. Marco teórico</b>	<b>7</b>
2.1. Modelo de cámara	7
2.1.1. Homografías	8
2.1.2. Estimación de homografías usando puntos de correspondencia	9
2.1.3. Estimación de homografías usando líneas	10
2.1.4. Estimación de homografías usando cuadriláteros	11
2.1.5. Corrección de perspectiva	15
2.1.6. Generación de imágenes panorámicas	15
2.1.7. Construcción de imágenes compuestas	16
2.2. Calibración de cámaras usando homografías	16
2.2.1. Parámetros intrínsecos	17
2.2.2. Parámetros extrínsecos	20
2.3. Distorsión radial	21
2.3.1. Modelo polinomial	21
2.3.2. Corrección de distorsión radial	22
2.4. Detección de contornos	23
2.4.1. Método del gradiente	23
2.4.2. Método de Canny	25
2.4.3. Filtro de ventana	27
2.5. Detección de líneas	28

2.5.1. Coordenadas paralelas . . . . .	29
2.5.2. Transformada de Hough . . . . .	30
2.5.3. Máscara Gaussiana . . . . .	30
2.6. Estimación de posición y orientación . . . . .	33
2.6.1. Predicción de líneas de carril . . . . .	34
<b>3. Desarrollo experimental y resultados</b>	<b>37</b>
3.1. Plataforma experimental . . . . .	37
3.2. Implementación en cómputo paralelo . . . . .	37
3.3. Calibración de cámara . . . . .	39
3.4. Corrección de imágenes distorsionadas . . . . .	41
3.5. Detección de líneas de carril . . . . .	42
3.5.1. Filtro de ventana . . . . .	44
3.5.2. Máscara Gaussiana . . . . .	44
3.5.3. Líneas de carril curvas . . . . .	45
3.5.4. Líneas discontinuas y múltiples carriles . . . . .	47
3.6. Estimación de posición y orientación . . . . .	48
3.7. Navegación de un robot móvil terrestre . . . . .	49
3.8. Discusiones . . . . .	51
<b>4. Conclusiones</b>	<b>55</b>
<b>A. Mínimos cuadrados recursivos con factor de olvido</b>	<b>57</b>
<b>B. Rectificación de imágenes</b>	<b>61</b>
B.1. Estimación de la matriz fundamental . . . . .	62
B.2. Rectificación . . . . .	63
<b>C. Segunda transformada de Hough</b>	<b>71</b>
<b>Bibliografía</b>	<b>75</b>

# Índice de cuadros

3.1. Comparación en tiempos de ejecución para cada implementación.	40
--	----



# Índice de figuras

1.1. Posición y orientación en sistemas móviles para la navegación asistida. (a) Un vehículo aéreo presenta seis grados de libertad para definir su posición y orientación en el espacio tridimensional. (b) Un vehículo terrestre presenta tres grados de libertad para definir su posición y orientación en el plano del piso. (c) Un vehículo terrestre restringido a moverse a lo largo de un carril solo presenta dos grados de libertad: la posición $x$ perpendicular a la dirección de desplazamiento y el ángulo $\phi$ alrededor del eje vertical. . . . .	2
2.1. Diferentes configuraciones para el proceso de formación de imágenes. (a) Lente convencional. (b) Lente simple. (c) Cámara <i>pinhole</i> . . . . .	8
2.2. Proceso de formación de imágenes. (a) Caso general en donde se observan puntos $\mathbf{p}$ en el espacio tridimensional. (b) Caso particular en donde los puntos $\mathbf{p}$ son coplanares a los puntos $\boldsymbol{\rho}$ . . . . .	9
2.3. Los puntos $\boldsymbol{\mu}_a, \boldsymbol{\mu}_b$ y $\boldsymbol{\mu}_c$ se determinan por las intersecciones de las líneas generadas a partir de cuatro puntos $\boldsymbol{\mu}_k$ en el plano imagen. . . . .	12
2.4. Captura de cuatro puntos por cada rectángulo observado en la imagen. . . . .	15
2.5. Objetos seleccionados con perspectiva corregida. (a) Monitor de computadora. (b) Documento impreso en hoja de papel. (c) Pantalla de tableta electrónica. . . . .	16
2.6. Composición de imágenes panorámicas. (a)-(c) Imágenes de entrada. (d) Imagen panorámica resultante. . . . .	17
2.7. Imágenes capturadas de una escena en diferentes puntos de observación. (a)-(d) Conjunto de imágenes correspondiente a una pared con objetos. (e)-(g), (h)-(j), y (k)-(n) Conjuntos de imágenes de paredes, piso y techo de la escena observado. . . . .	18
2.8. Imágenes compuestas generadas por los conjuntos de imágenes capturadas. . . . .	19

2.9. (a) Escena tridimensional creada con imágenes compuestas. (b)-(g) Escena en diferentes ángulos. . . . .	20
2.10. La distorsión radial de acuerdo a la transformación de un punto sin distorsión $\beta$ a un punto distorsionado $\delta$ . . . . .	22
2.11. Simulación de distorsión de lentes. (a) Imagen de una rejilla con distorsión radial. (b) Imagen de la rejilla en (a) con distorsión radial corregida. . . . .	23
2.12. Corrección de perspectiva y distorsión de una imagen. (a) Imagen con perspectiva y distorsión radial. (b) Imagen corregida, y (c) la región de interés. . . . .	24
2.13. Representaciones de las líneas de carril en (a) el piso o plano de referencia, y (b) el plano imagen. . . . .	27
2.14. Representación de una línea parametrizada: (a) $y = mx + b$ y (b) $\rho = x \cos \phi + y \sin \phi$ . . . . .	28
2.15. Ejemplo de representación de nutrientes en la dieta de un paciente graficados usando coordenadas paralelas . . . . .	29
2.16. A la derecha está la representación de puntos y líneas en el plano imagen y a la izquierda su correspondiente representación en el plano de Hough. (a) Un punto $p$ en el plano imagen se representa como (b) una polilínea $\sigma$ en el plano de Hough. (c) Una línea $\ell$ con puntos $p_k$ en el plano imagen se representa como (b) un punto $\hat{\ell}$ en el plano de Hough. (e) Un conjunto de líneas concurrentes $\ell_k$ que intersectan en un punto $p$ se representan como (f) un conjunto de puntos $\hat{\ell}$ que son “colineales” a lo largo de la polilínea $\sigma$ . . . .	31
2.17. (a) $D_1, D_2$ y $D_3$ son máscaras Gaussianas centrados en el origen. (b) Máscara Gaussiana $D$ centrado en el punto $t$ con ángulo de orientación $\phi$ . . . . .	32
2.18. (a) Vista del plano de referencia cuando el vehículo se encuentra centrado y alineado. (b) Imagen observada por la cámara. . . . .	33
3.1. Plataforma para navegación y adquisición de imágenes. (a) Cámara convencional conectada a la tarjeta de desarrollo <i>Raspberry Pi 3 B+</i> . (b) Robot móvil ensamblado. . . . .	38
3.2. Plataforma para procesamiento de imágenes. (a) Computadora de escritorio. (b) Tarjeta de video NVIDIA Geforce GTX 760. . .	38
3.3. Pista de prueba para navegación de un robot móvil terrestre. . .	39
3.4. Arquitectura del algoritmo de la transformada de Hough en cómputo paralelo. . . . .	39
3.5. (a)-(d) Secuencia de imágenes de un patrón cuadro-radial en diferentes perspectivas. (e)-(h) Representación de las imágenes (a)-(d) en el plano de Hough. . . . .	40
3.6. Patrones de calibración capturados por la cámara del robot móvil terrestre. . . . .	41
3.7. Patrones de franjas capturadas por la cámara del robot móvil terrestre. (a) Franja en dirección $x$ . (b) Franja en dirección $y$ . . .	42

3.8. Extracción de fase con el método de corrimiento de fase. (a) Fase en dirección $x$ . (b) Fase en dirección $y$ . . . . .	43
3.9. Corrección de distorsión utilizando los parámetros estimados. (a) Imagen con distorsión. (b) Imagen sin distorsión. . . . .	43
3.10. Capturas de tres secuencias de video para la evaluación del método propuesto. (a) Imagen en donde la mayor parte de la escena no corresponde a la pista. (b) Imagen en donde las líneas de carril son curvas. (c) Imagen en escenas con líneas de carril discontinuas y múltiples carriles. . . . .	44
3.11. (a)-(c) Detección de líneas de carril sin el uso filtros. (d)-(f) Detección de líneas usando el filtro de ventana para discriminación de contornos. (g)-(i) Detección de líneas utilizando el filtro de ventana y la máscara Gaussiana. . . . .	45
3.12. Evaluación del funcionamiento del algoritmo propuesto en líneas curvas. (a)-(c) Detección de líneas de carril sin el uso filtros. (d)-(f) Detección de líneas usando el filtro de ventana para discriminación de contornos. (g)-(i) Detección de líneas utilizando el filtro de ventana y la máscara Gaussiana. . . . .	46
3.13. Evaluación del funcionamiento del algoritmo propuesto en escenas de la vida real. (a)-(c) Detección de líneas de carril sin el uso filtros. (d)-(f) Detección de líneas usando el filtro de ventana para discriminación de contornos. (g)-(i) Detección de líneas utilizando el filtro de ventana y la máscara Gaussiana. . . . .	47
3.14. Primer conjunto de secuencias de vídeo. (a)-(y) Capturas de secuencias de video para evaluar el desempeño del método propuesto en diferentes escenarios. . . . .	49
3.15. Segundo conjunto de secuencias de vídeo. (a)-(y) Capturas de secuencias de video para evaluar el desempeño del método propuesto en diferentes escenarios. . . . .	50
3.16. Tercer conjunto de secuencias de vídeo. (a)-(y) Capturas de secuencias de video para evaluar el desempeño del método propuesto en diferentes escenarios. . . . .	51
3.17. Cuarto conjunto de secuencias de vídeo. (a)-(y) Capturas de secuencias de video para evaluar el desempeño del método propuesto en diferentes escenarios. . . . .	52
3.18. Secuencia de imágenes para la evaluación del algoritmo propuesto en líneas de carril no visibles. (a)-(e) Predicción de líneas de carril en oclusión por un vehículo, y (f)-(i) su visualización de los contornos detectados, respectivamente. . . . .	53

3.19. Evaluación del algoritmo propuesto en una secuencia de video de la navegación de un robot móvil terrestre. (a)-(e) Imágenes de entrada con distorsión radial. (f)-(j) Detección de contornos con su respectivo filtro de ventana para las imágenes (a)-(e) con distorsión corregida. (k)-(o) Picos de intensidad máxima limitado por la región de interés en el plano de Hough. (p)-(t) Resultados de detección de líneas de carril y estimación de posición y orientación del robot móvil terrestre. . . . .	54
B.1. Escena experimental para el método de rectificación de imágenes.	
(a)-(b) Un par de imágenes capturadas en diferentes perspectivas.	
(c)-(d) Imágenes rectificadas. . . . .	68
C.1. (a)-(d) y (e)-(h) Par de ejemplos de detección de puntos de concurrencia en un patrón cuadro-radial. . . . .	73



# Introducción

## 1.1. Visión por computadora

En la actualidad, los sistemas de visión por computadora están impulsando el desarrollo de nuevas tecnologías para realizar diversos tipos de aplicaciones en ingeniería, educación, medicina, y entretenimiento entre otras [1–4]. El estudio y diseño de nuevos métodos de visión computacional es un tema de investigación actual y de gran relevancia debido a su impacto en la sociedad [5]. Algunos ejemplos de aplicación como la inspección de componentes electrónicos, digitalización tridimensional de objetos, navegación visual, análisis de irregularidades en tejido, realidad aumentada, reconocimiento, y análisis biométrico, entre otros [6–9]. Los sistemas de visión tienen como finalidad obtener información de un entorno tridimensional como lo hace el sistema visual humano [10]. Para alcanzar este comportamiento resulta muy complejo, los resultados deben ser mejor en comparación al sistema de visión humano [11, 12].

En la actualidad, los equipos digitales han mejorado considerablemente en los últimos años. Las cámaras digitales alcanzan altos niveles de resolución con sensibilidad a una amplia gama de colores. Los niveles de nitidez en las imágenes rebasan lo que el ojo humano puede distinguir. Sin embargo, para las computadoras digitales, las imágenes son simplemente arreglos de píxeles o valores numéricos de intensidad que representan colores en cada píxel. Por esta razón, desde el punto de vista computacional, comprender la información contenida en imágenes es un problema que se ha tratado de resolver desde hace décadas [10].

Los modelos de aprendizaje marcaron el inicio de trabajos sobre visión por computadora [13, 14]. Los resultados que obtuvieron propició un interés creciente por este tema de investigación y diversos enfoques se han desarrollado y reportado en la literatura científica [15–17]. A pesar de los importantes avances realizados en el campo de visión computacional, todavía permanece abierto un numeroso considerable de problemas. Por ejemplo, reducir los tiempos de respuesta de los sistemas, mejorar los niveles de exactitud, reducir el consumo de

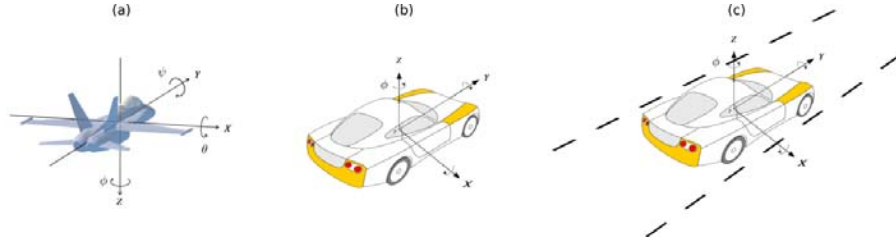


Figura 1.1: Posición y orientación en sistemas móviles para la navegación asistida<sup>2</sup>. (a) Un vehículo aéreo presenta seis grados de libertad para definir su posición y orientación en el espacio tridimensional. (b) Un vehículo terrestre presenta tres grados de libertad para definir su posición y orientación en el plano del piso. (c) Un vehículo terrestre restringido a moverse a lo largo de un carril solo presenta dos grados de libertad: la posición  $x$  perpendicular a la dirección de desplazamiento y el ángulo  $\phi$  alrededor del eje vertical.

energía, reducir el tamaño de los equipos, y simplificar los procesos de operación para facilidad de uso, entre otros.

## 1.2. Instrumentación para navegación asistida

La instrumentación se encarga de realizar mediciones físicas mediante el uso de sensores, permiten monitorear y retroalimentar el sistema de navegación [18–21]. En general, para robots móviles, la navegación asistida requiere mediciones de proximidad (distancia), posición  $(x, y, z)$ , orientación (cabeceo, alabeo, guiñada), y velocidad  $(V_x, V_y, V_z)$ , entre otras [22], tal como se muestra en la figura 1.1(a). En particular, para un vehículo terrestre, las variables de interés se reducen a la posición  $(x, y)$  en el plano de referencia, y la orientación definida por la rotación alrededor del eje  $z$  (guiñada), como se muestra en la figura 1.1(b).

En este trabajo de tesis se aborda como problema de estudio, la navegación visual de un robot móvil terrestre restringido a desplazarse a lo largo de un carril, como se ilustra en la figura 1.1(c). De esta forma, el problema de estimación de posición y orientación del vehículo se reduce a estimar la coordenada  $x$ , y la orientación dada por el ángulo  $\phi$  alrededor del eje vertical.

En la literatura se han propuesto diferentes enfoques para extraer información útil para la navegación empleando visión por computadora. Por ejemplo, se han propuesto métodos de estimación de distancias de objetos alrededor del vehículo [23, 24], mapas de entrenamiento del sistema [25], planificación de rutas [25, 26], y reconstrucción virtual del espacio tridimensional, entre

<sup>2</sup>Imágenes tomadas de

<http://www.chrobotics.com/library/understanding-euler-angles>

[https://www.researchgate.net/figure/Vehicle-with-coordinate-system\\_fig1\\_311268794](https://www.researchgate.net/figure/Vehicle-with-coordinate-system_fig1_311268794)

otras [27–29]. En este trabajo de tesis, se propone un método de navegación visual empleando transformaciones proyectivas y la transformada de Hough para la detección de líneas de carril. Los resultados obtenidos en esta tesis muestran que es posible determinar la posición y orientación del vehículo empleando las líneas de carril como fuente de información.

### 1.3. Sistemas opto-digitales

La óptica es una rama de la física que se encarga del estudio de los fenómenos en los que interviene la luz [30,31]. Los avances de la óptica se vieron motivados por la necesidad de aprovechar información visual. Los primeros avances consistieron en el análisis y aplicación de sistemas puramente ópticos; por ejemplo, holografía, microscopía, e interferencia láser, entre otros. Incluso, la captura de imágenes se realizaba con cámaras analógicas, donde las distribuciones de intensidad eran registradas empleando película fotográfica.

Los avances recientes en sensores electrónicos y computadoras digitales han permitido incrementar la eficiencia de los sistemas ópticos. En la actualidad, la captura de imágenes se realiza de forma rápida empleando cámaras digitales. También, las computadoras digitales permiten que el procesamiento de las imágenes capturadas pueda realizarse en altas velocidades y de forma mucho más flexible.

Los sistemas ópticos tienen configuraciones que incluyen sensores, lentes y filtros que permiten procesar señales análogas usando las propiedades de la luz [32–34]. Por otro lado, los sistemas digitales convierten las señales análogas a digital para realizar procesamiento de imágenes a través de secuencias específicas de instrucciones [35,36]. La integración de un sistema óptico con un medio de adquisición y procesamiento digital dio origen a los sistemas opto-digitales.

En este trabajo de tesis, aprovechamos las ventajas de los sistemas opto-digitales para el desarrollo de un método de estimación de posición y orientación de un vehículo móvil terrestre desplazándose a lo largo de un carril. En la parte óptica, consideramos las propiedades geométricas de la cámara y el proceso óptico de formación de imagen. Respecto a la parte digital, se proponen algoritmos viables para ser implementados en plataformas computacionales de alto desempeño (GPU por sus siglas en inglés *Graphics Processing Unit*), explotando la capacidad del paralelismo masivo. El método propuesto es evaluado procesando secuencias de video tomadas del mundo real. De los resultados obtenidos, se observa que el método propuesto puede ser útil para proporcionar las señales de posición y orientación requeridas por el controlador de un robot móvil.

El contenido del presente documento de tesis se organiza de la siguiente forma. En el capítulo 2 se analizan los principios teóricos utilizados para el desarrollo del método de detección de carril y estimación de posición y orientación del vehículo. En el capítulo 3 se presenta la plataforma experimental construida, los detalles de la implementación del método propuesto, y la calibración del sistema. Este capítulo finaliza con la descripción de los resultados obtenidos en la detección de carril y estimación de pose empleando secuencias de video del

mundo real. En el capítulo 4 se presentan las conclusiones de la presente tesis y trabajo a futuro.

En el trabajo de investigación desarrollado en esta tesis, se analizaron diferentes enfoques reportado en la literatura científica para visión por computadora. En el apéndice A se presenta la estimación de parámetros del sistema usando mínimos cuadrados recursivos con factor de olvido. Los apéndices B, y C presentan el método de rectificación de imágenes, y segunda transformada de Hough, respectivamente. En estos apéndices se describen los aspectos básicos de la implementación de estos métodos alternativos así como las desventajas restrictivas por las que no fueron usados en este trabajo de tesis.

## 1.4. Objetivos

El objetivo general y los objetivos específicos de este trabajo de tesis, se presentan a continuación:

### 1.4.1. Objetivo general

Desarrollo de técnicas de visión computacional optimizando métricas objetivas para aplicaciones de instrumentación óptica y navegación asistida empleando métodos opto-digitales.

### 1.4.2. Objetivos específicos

- Estimar la posición y orientación de un robot móvil terrestre usando como referencia las líneas de carril en la pista.
- Implementar el algoritmo propuesto de forma eficiente en una plataforma computacional de alto rendimiento.
- Cuantificar el desempeño del sistema propuesto utilizando medidas de desempeño objetivas a través de simulaciones por computadora.
- Verificar el funcionamiento del método propuesto empleando secuencias de video del mundo real.

## 1.5. Contribuciones

Los resultados derivados de este trabajo de tesis consisten en la publicación de un artículo científico, cuatro memorias de congreso, y un software de procesamiento de patrones de franjas. Los detalles de estos desarrollos se enlistan a continuación.

- Rigoberto Juarez-Salazar, Alejandra Giron, Juan Zheng, and Victor H. Diaz-Ramirez, “Key concepts for phase-to-coordinate conversion in fringe projection systems,” *Appl. Opt.* 58, 4828-4834 (20 June 2019). DOI: <https://doi.org/10.1364/AO.58.004828>.

- Juan Zheng, Alejandra Giron, Rigoberto Juarez-Salazar, and Victor H. Diaz-Ramirez “Image stitching by projective transformations,” Proc. SPIE 11136, Optics and Photonics for Information Processing XIII, 111360C (6 September 2019).  
DOI: <https://doi.org/10.1117/12.2528585>.
- Alejandra Giron, Juan Zheng, Rigoberto Juarez-Salazar, and Victor H. Diaz-Ramirez “Length measurements from monocular images,” Proc. SPIE 11136, Optics and Photonics for Information Processing XIII, 111360F (6 September 2019).  
DOI: <https://doi.org/10.1117/12.2528607>.
- Rigoberto Juarez-Salazar, Juan Zheng, Alejandra Giron, and Victor H. Diaz-Ramirez “Calibration of camera-projector fringe projection systems for three-dimensional scanning,” Proc. SPIE 11136, Optics and Photonics for Information Processing XIII, 111360D (6 September 2019).  
DOI: <https://doi.org/10.1117/12.2530016>.
- Juan Zheng, Rigoberto Juarez-Salazar, y Victor H. Diaz-Ramirez, “Pose estimation from projective transformations for visual guidance of a wheeled mobile robot,” Proc. SPIE, Optics and Photonics for Information Processing XIII (25 August 2020).
- Rigoberto Juarez-Salazar, Alejandra Giron, Juan Zheng, Victor H. Diaz-Ramirez, “3DFringe: Digitalización de objetos tridimensionales por proyección de franjas,” Participación en la convocatoria *Premio al mejor software* del Instituto Politécnico Nacional (27 August 2019).



## Capítulo 2

# Marco teórico

## 2.1. Modelo de cámara

Las cámaras fotográficas son dispositivos complejos que involucran la operación conjunta de un sensor foto-sensible y un sistema de formación de imagen (lente compuesta). El sistema de formación de imagen está integrado por elementos ópticos, electrónicos y mecánicos que incluyen lentes, obturadores, filtros y actuadores diseñados para generar imágenes de alta calidad.

En la figura 2.1(a) se representa el proceso de formación de imagen usando una lente compuesta. El diseño de algoritmos de procesamiento de datos a partir del modelo matemático de una lente compuesta puede resultar compleja [37,38].

El proceso de formación de imagen puede describirse completamente suponiendo que el sistema de formación de imagen consiste en una lente simple como se muestra en la figura 2.1(b). El modelo matemático resultante será suficiente para incluir efectos tales como desenfoque y distorsión radial.

Los algoritmos de procesamiento de imágenes pueden simplificarse aún más cuando el efecto de desenfoque y la distorsión radial son despreciables. En este caso, se puede considerar el modelo de una cámara *pinhole* como se muestra en la figura 2.1(c) [15].

El proceso de formación de imagen usando el modelo de cámara *pinhole* es bastante simple. Este modelo integra la información sobre los parámetros extrínsecos (posición y orientación) de la cámara y los parámetros intrínsecos tales como la longitud focal, el centro óptico, oblicuidad y tamaño de los píxeles [39]. Usando el modelo de cámara *pinhole*, un punto  $\mathbf{p}$  en el espacio tridimensional observado por la cámara, se detecta como un punto  $\boldsymbol{\mu}$  en el plano imagen como

$$\boldsymbol{\mu} = \mathcal{H}^{-1}[C\mathcal{H}[\mathbf{p}]], \quad (2.1)$$

donde  $C = K[R^T, -R^T\mathbf{t}]$  es una matriz de tamaño  $4 \times 3$  que contiene los parámetros de la cámara,  $K$  es la matriz de parámetros intrínsecos, la matriz  $R$  y el

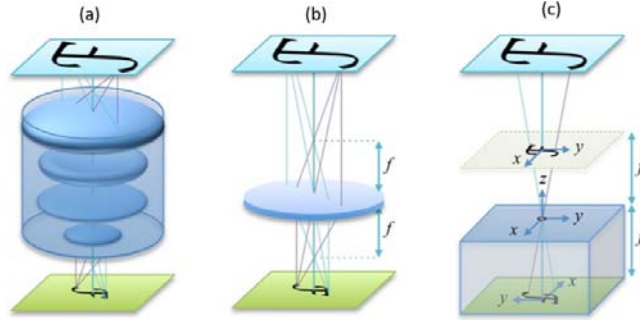


Figura 2.1: Diferentes configuraciones para el proceso de formación de imágenes. (a) Lente convencional. (b) Lente simple. (c) Cámara *pinhole*.

vector  $\mathbf{t}$  definen la orientación y posición de la cámara, respectivamente,  $[\cdot]^T$  indica transposición, y  $\mathcal{H}[\cdot]$  es el operador de coordenadas homogéneas.

El operador de coordenadas homogéneas permite realizar operaciones geométricas cambiando del espacio cartesiano al espacio homogéneo. De esta forma, las transformaciones de proyección geométricas, que son no-lineales en el espacio cartesiano, se realizan de forma simple a través de operaciones lineales. Para realizar proyecciones geométricas, el operador  $\mathcal{H}$  agrega una dimensión adicional al vector dado; por ejemplo, un punto  $\mathbf{p} = [x, y, z]$ , sus correspondientes coordenadas homogéneas son

$$H[\mathbf{p}] = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \mathbf{q}. \quad (2.2)$$

Para regresar al espacio cartesiano original, el operador  $\mathcal{H}^{-1}$  reduce una dimensión al vector  $\mathbf{q}$  y es dividido por el último elemento; específicamente,

$$H^{-1}[\mathbf{q}] = \frac{1}{s} \mathbf{p}, \quad (2.3)$$

donde  $s$  es el último elemento de  $\mathbf{q}$ .

### 2.1.1. Homografías

Existe un caso particular de puntos que pertenecen a un plano que se encuentra en el espacio tridimensional [40]. Los puntos  $\boldsymbol{\rho}$  del plano son observados y registrados en el plano imagen como puntos  $\boldsymbol{\mu}$ , como se observa en la figura 2.1.1. Dado este caso, los puntos  $\mathbf{p}$  (del espacio tridimensional) se pueden parametrizar usando los puntos  $\boldsymbol{\rho}$  (del plano bidimensional) como

$$\mathbf{p} = [\mathbf{q}_1, \quad \mathbf{q}_2, \quad s]\mathcal{H}[\boldsymbol{\rho}], \quad (2.4)$$

donde  $\mathbf{q}_1$  y  $\mathbf{q}_2$  son vectores ortogonales que definen la orientación del plano observado, y  $s$  representa la traslación del plano observado. Utilizando este



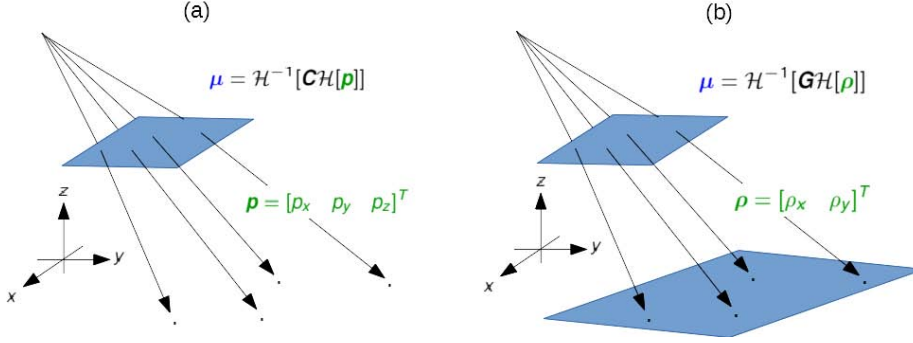


Figura 2.2: Proceso de formación de imágenes. (a) Caso general en donde se observan puntos  $\mathbf{p}$  en el espacio tridimensional. (b) Caso particular en donde los puntos  $\mathbf{p}$  son coplanares a los puntos  $\boldsymbol{\rho}$ .

caso, en la Ec. (2.1) se reduce la matriz  $\mathbf{C}$  de la cámara *pinhole* a una matriz  $\mathbf{G}$  de tamaño  $3 \times 3$  como

$$\boldsymbol{\mu} = \mathcal{H}^{-1}[\mathbf{G}\mathcal{H}[\boldsymbol{\rho}]], \quad (2.5)$$

donde  $\mathbf{G}$  es una matriz conocida como *homografía* y está dada por

$$\mathbf{G} = \mathbf{K}\mathbf{R}^T[\mathbf{q}_1, \ \mathbf{q}_2, \ \mathbf{s} - \mathbf{t}]. \quad (2.6)$$

Esta matriz puede construirse usando solamente los parámetros de posición y orientación de la cámara [41]. La homografía se puede describir como una transformación geométrica que relaciona los puntos del plano imagen y los puntos del plano de referencia. Esta matriz es de gran importancia en aplicaciones tales como calibración de cámaras, corrección de perspectiva, y construcción de imágenes panorámicas, entre otras [15, 40, 42]. A continuación, se mostrarán tres métodos de estimación de homografías, basados en puntos de correspondencia, líneas en la imagen, y esquinas de cuadriláteros.

### 2.1.2. Estimación de homografías usando puntos de correspondencia

Una homografía se representa por una matriz de tamaño  $3 \times 3$  dada por

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{g}}_1^T \\ \bar{\mathbf{g}}_2^T \\ \bar{\mathbf{g}}_3^T \end{bmatrix}, \quad (2.7)$$

donde  $\bar{\mathbf{g}}_1^T$ ,  $\bar{\mathbf{g}}_2^T$  y  $\bar{\mathbf{g}}_3^T$  son los renglones de la matriz  $\mathbf{G}$ . Usando la Ec. (2.5) y por definición de  $\mathcal{H}^{-1}$  se obtiene

$$\boldsymbol{\mu} = \frac{1}{\bar{\mathbf{g}}_3^T \mathcal{H}[\boldsymbol{\rho}]} \begin{bmatrix} \bar{\mathbf{g}}_1^T \mathcal{H}[\boldsymbol{\rho}] \\ \bar{\mathbf{g}}_2^T \mathcal{H}[\boldsymbol{\rho}] \end{bmatrix}. \quad (2.8)$$

Despejando el denominador de la Ec. (2.8) se forma el siguiente sistema de ecuaciones lineales:

$$\begin{aligned}\bar{\mathbf{g}}_3^T \mathcal{H}[\boldsymbol{\rho}] \mu_x &= \bar{\mathbf{g}}_1^T \mathcal{H}[\boldsymbol{\rho}], \\ \bar{\mathbf{g}}_3^T \mathcal{H}[\boldsymbol{\rho}] \mu_y &= \bar{\mathbf{g}}_2^T \mathcal{H}[\boldsymbol{\rho}].\end{aligned}\tag{2.9}$$

Este sistema de ecuaciones lineales se puede escribir en forma matricial como

$$\underbrace{\begin{bmatrix} \mathcal{H}[\boldsymbol{\rho}]^T & \mathbf{0}_3^T & -\mu_x \mathcal{H}[\boldsymbol{\rho}]^T \\ \mathbf{0}_3^T & \mathcal{H}[\boldsymbol{\rho}] & -\mu_y \mathcal{H}[\boldsymbol{\rho}]^T \end{bmatrix}}_A \underbrace{\begin{bmatrix} \bar{\mathbf{g}}_1 \\ \bar{\mathbf{g}}_2 \\ \bar{\mathbf{g}}_3 \end{bmatrix}}_g = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.\tag{2.10}$$

La matriz homografía tiene nueve elementos a determinar; sin embargo, solo existen ocho incógnitas porque las homografías son únicas excepto escala. Por otro lado, considerando la Ec. (2.10) se concluye que por cada correspondencia de puntos  $(\boldsymbol{\mu}, \boldsymbol{\rho})$  se obtienen dos ecuaciones. Por lo tanto, se necesitan al menos cuatro puntos de correspondencia para estimar una matriz homografía. Con  $n$  puntos de correspondencia, se obtiene la siguiente ecuación matricial:

$$\underbrace{\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ \vdots \\ A_n \end{bmatrix}}_A \mathbf{g} = \underbrace{\begin{bmatrix} \mathbf{0}_2 \\ \mathbf{0}_2 \\ \mathbf{0}_2 \\ \mathbf{0}_2 \\ \vdots \\ \mathbf{0}_2 \end{bmatrix}}_{\mathbf{0}_{2n}},\tag{2.11}$$

donde  $A_k$  representa las dos ecuaciones correspondientes a la Ec. 2.10 para un punto de correspondencia, y  $\mathbf{0}_2$  es un vector de ceros de tamaño  $2 \times 1$ . Se usa descomposición en valores singulares para encontrar el vector  $\mathbf{g}$  que minimiza  $\|A\mathbf{g}\|$ . Teniendo  $G$ , se puede corregir la distorsión generada por la perspectiva respecto al objeto de interés utilizando la Ec. (2.5) como

$$\boldsymbol{\rho} = \mathcal{H}^{-1}[G^{-1}\mathcal{H}[\boldsymbol{\mu}]].\tag{2.12}$$

Una desventaja de este proceso de estimación de homografías es la necesidad de proporcionar tantos puntos  $\boldsymbol{\mu}$  del plano imagen como puntos  $\boldsymbol{\rho}$  del plano de referencia. Por ejemplo, para escanear documentos, los usuarios no cuentan con información del plano de referencia (puntos  $\boldsymbol{\rho}$ ). Por consecuencia, no hay información a priori para conocer la relación del aspecto del documento. Para este tipo de aplicaciones se propone un método de estimación de homografías donde las coordenadas  $\boldsymbol{\rho}$  del plano de referencia no son requeridas.

### 2.1.3. Estimación de homografías usando líneas

Una línea recta se puede expresa matemáticamente como

$$y = mx + b,\tag{2.13}$$

donde  $m$  es la pendiente de la recta, y  $b$  es la intersección con el eje vertical. Observe que la ecuación de la recta se puede reescribir como  $mx - y + b = 0$ . En general, para tres constantes cualquiera  $a$ ,  $b$ , y  $c$  (excepto  $a = b = c = 0$ ), la recta puede escribirse como

$$ax + by + c = 0. \quad (2.14)$$

Esta ecuación toma una forma simple cuando se usan coordenadas homogéneas. Específicamente, las líneas rectas pueden ser representadas por un vector como

$$\ell = \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \quad (2.15)$$

De esta forma, de la Ec. (2.14), se observa que los puntos  $\mu = [x, y]^T$  que pertenecen a la recta parametrizada por el vector  $\ell$  satisfacen

$$\ell^T \mathcal{H}[\mu] = 0. \quad (2.16)$$

Si tenemos  $n$  líneas,  $\ell_1, \ell_2, \dots, \ell_n$ , que intersectan en el punto  $\mu$ , entonces se puede construir el siguiente sistema de ecuaciones

$$\begin{bmatrix} \ell_1^T \\ \ell_2^T \\ \vdots \\ \ell_n^T \end{bmatrix} \mathcal{H}[\mu] = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (2.17)$$

El punto de intersección  $\mu$  de las líneas se encuentra resolviendo la Ec. (2.17) usando el método de descomposición en valores singulares. Este enfoque se puede aplicar para encontrar los puntos  $\mu_n$  necesarios para realizar corrección de perspectiva usando el método descrito en la subsección anterior.

#### 2.1.4. Estimación de homografías usando cuadriláteros

La matriz homografía se puede representar como

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} = [\mathbf{g}_1 \quad \mathbf{g}_2 \quad \mathbf{g}_3], \quad (2.18)$$

donde  $\mathbf{g}_1$ ,  $\mathbf{g}_2$  y  $\mathbf{g}_3$  son las columnas de la matriz  $G$ . Observamos que las columnas de  $G$  corresponden a las coordenadas homogéneas de las imágenes de puntos  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ , y  $\mathbf{e}_3$  dados por

$$\begin{aligned} \mathbf{g}_1 &= G\mathcal{H}[\mathbf{e}_1], \\ \mathbf{g}_2 &= G\mathcal{H}[\mathbf{e}_2], \\ \mathbf{g}_3 &= G\mathcal{H}[\mathbf{e}_3], \end{aligned} \quad (2.19)$$

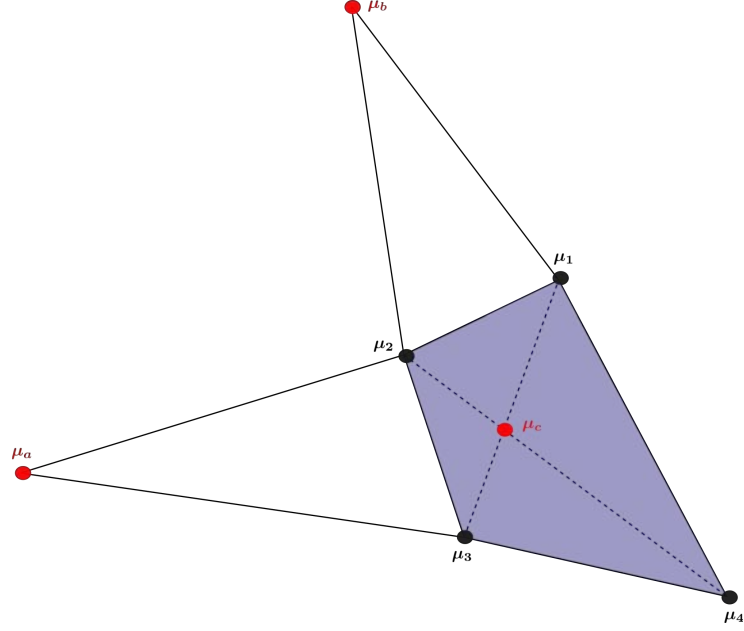


Figura 2.3: Los puntos  $\mu_a, \mu_b$  y  $\mu_c$  se determinan por las intersecciones de las líneas generadas a partir de cuatro puntos  $\mu_k$  en el plano imagen.

donde  $e_1$  y  $e_2$  son los puntos al infinito en el plano de referencia en dirección  $q_1$  y  $q_2$ , respectivamente, y  $e_3$  es el origen del plano de referencia. Las coordenadas homogéneas de los puntos  $e_1$ ,  $e_2$  y  $e_3$  están dadas por

$$\mathcal{H}[e_1] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathcal{H}[e_2] = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathcal{H}[e_3] = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.20)$$

Usando los nuevos puntos junto con la Ec. (2.5), se obtiene el siguiente conjunto de ecuaciones:

$$\begin{aligned} \mu_a &= \mathcal{H}^{-1}[G\mathcal{H}[e_1]], \\ \mu_b &= \mathcal{H}^{-1}[G\mathcal{H}[e_2]], \\ \mu_c &= \mathcal{H}^{-1}[G\mathcal{H}[e_3]]. \end{aligned} \quad (2.21)$$

Aplicando  $\mathcal{H}^{-1}$  a cada ecuación y sustituyendo las columnas de la ecuación (2.18), se obtiene

$$G = [\gamma_1 \mathcal{H}[\mu_a] \quad \gamma_2 \mathcal{H}[\mu_b] \quad \mathcal{H}[\mu_c]], \quad (2.22)$$

donde  $\gamma_1$  y  $\gamma_2$  son factores de escala desconocidos. Los puntos  $\mu_a, \mu_b$  y  $\mu_c$  se conocen por los puntos  $\mu_k$  dados por el usuario, como se observa en la figura 2.3.

Utilizando los puntos conocidos  $\mu_k$ , se obtiene el siguiente conjunto de líneas:

$$\begin{aligned}\ell_1 &= \mathcal{H}[\mu_1] \times \mathcal{H}[\mu_2], \\ \ell_2 &= \mathcal{H}[\mu_3] \times \mathcal{H}[\mu_4], \\ \ell_3 &= \mathcal{H}[\mu_1] \times \mathcal{H}[\mu_4], \\ \ell_4 &= \mathcal{H}[\mu_2] \times \mathcal{H}[\mu_3], \\ \ell_5 &= \mathcal{H}[\mu_2] \times \mathcal{H}[\mu_4], \\ \ell_6 &= \mathcal{H}[\mu_1] \times \mathcal{H}[\mu_3],\end{aligned}\tag{2.23}$$

donde  $\ell_1$  y  $\ell_2$  son líneas paralelas hacia el eje  $x$ . Las líneas  $\ell_3$  y  $\ell_4$  son paralelas hacia el eje  $y$ . La intersección de las líneas  $\ell_5$  y  $\ell_6$  indica el origen. Las intersecciones de las líneas nos permiten calcular los puntos  $\mu_a$ ,  $\mu_b$  y  $\mu_c$  como

$$\begin{aligned}\mathcal{H}[\mu_a] &= \ell_1 \times \ell_2 = [a_1 \quad a_2 \quad a_3]^T, \\ \mathcal{H}[\mu_b] &= \ell_3 \times \ell_4 = [b_1 \quad b_2 \quad b_3]^T, \\ \mathcal{H}[\mu_c] &= \ell_5 \times \ell_6 = [c_1 \quad c_2 \quad c_3]^T.\end{aligned}\tag{2.24}$$

Asumiendo que la cámara está alineada al plano de referencia en el origen ( $R = \mathbb{I}$ ,  $\mathbf{t} = \mathbf{0}_3$ ), la Ec. (2.6) se reescribe como

$$\lambda [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \mathbf{s}] = K^{-1} [\gamma_1 \mathcal{H}[\mu_a] \quad \gamma_2 \mathcal{H}[\mu_b] \quad \mathcal{H}[\mu_c]].\tag{2.25}$$

De esta manera se puede aprovechar la ortonormalidad de los vectores  $\mathbf{q}_1$  y  $\mathbf{q}_2$ . Asumiendo que  $\mathbf{q}_1^T \mathbf{q}_1 = \mathbf{q}_2^T \mathbf{q}_2$  y  $\mathbf{q}_1^T \mathbf{q}_2 = 0$ , se obtiene

$$\gamma_1^2 \mathcal{H}[\mu_a]^T W \mathcal{H}[\mu_a] = \gamma_2^2 \mathcal{H}[\mu_b]^T W \mathcal{H}[\mu_b],\tag{2.26}$$

$$\mathcal{H}[\mu_a]^T W \mathcal{H}[\mu_b] = 0,\tag{2.27}$$

donde

$$W = K^{-T} K^{-1}.\tag{2.28}$$

La matriz  $K$  se compone por los seis parámetros intrínsecos de la cámara, dados por el tamaño del píxel (ancho  $s_x$  y altura  $s_y$ ), las coordenadas del punto principal  $(\tau_x, \tau_y)$ , la distancia focal  $f$  y la oblicuidad del píxel  $\sigma$ . Para simplificar la estimación de estos parámetros, se usan las siguientes suposiciones [40]. En la actualidad, la calidad de las cámaras modernas es alta. Entonces, podemos asumir con seguridad que el sensor de la cámara tiene una oblicuidad de cero. De esta manera,  $W$  se simplifica como

$$W = \begin{bmatrix} s_x^2 & 0 & -s_x \tau_x \\ 0 & s_y^2 & -s_y \tau_y \\ -s_x \tau_x & -s_y \tau_y & \tau_x^2 + \tau_y^2 + f^2 \end{bmatrix}.\tag{2.29}$$

Además, el sensor de la cámara y sus lentes están alineados tal que el punto principal está muy cercano al centro de la imagen. Por lo tanto, se asume que

las coordenadas del punto principal son  $\tau_x = \tau_y = 0$ . De esta forma, la matriz  $W$  se vuelve diagonal, específicamente,

$$W = \text{diag}(s_x^2, s_y^2, f). \quad (2.30)$$

Más aún, asumiendo que la relación de aspecto de los píxeles es conocida, la matriz  $W$  se puede reescribir como

$$W = \text{diag}(r^2, 1, w), \quad (2.31)$$

donde  $r = s_x/s_y$  es la relación de aspecto del píxel ( $r = 1$  si los píxeles son cuadrados), y  $w = f^2/s_y^2$  es una constante desconocida. Usando la matriz  $W$ , la Ec. (2.27) se puede reescribir como

$$a_1 b_1 r^2 + a_2 b_2 + a_3 b_3 w = 0, \quad (2.32)$$

obteniendo como resultado

$$w = -\frac{r^2 a_1 b_1 + a_2 b_2}{a_3 b_3}. \quad (2.33)$$

Con base en lo anterior, la matriz  $W$  se calcula usando la Ec. (2.26) para obtener la relación del aspecto, dado por

$$\xi = \frac{\gamma_1}{\gamma_2} = \sqrt{\frac{\mathcal{H}[\boldsymbol{\mu}_b]^T W \mathcal{H}[\boldsymbol{\mu}_b]}{\mathcal{H}[\boldsymbol{\mu}_a]^T W \mathcal{H}[\boldsymbol{\mu}_a]}}. \quad (2.34)$$

En consecuencia, para cada punto  $\boldsymbol{\mu}$  en la Ec. (2.12), su correspondencia del punto  $\boldsymbol{\rho}$  es

$$\begin{bmatrix} \rho_{k,x} \\ \rho_{k,y} \end{bmatrix} = \frac{1}{(\mathcal{H}[\boldsymbol{\mu}_a] \times \mathcal{H}[\boldsymbol{\mu}_b])^T \mathcal{H}[\boldsymbol{\mu}_k]} \begin{bmatrix} (1/\gamma_1)(\mathcal{H}[\boldsymbol{\mu}_b] \times \mathcal{H}[\boldsymbol{\mu}_c])^T \\ -(1/\gamma_2)(\mathcal{H}[\boldsymbol{\mu}_a] \times \mathcal{H}[\boldsymbol{\mu}_c])^T \end{bmatrix} \mathcal{H}[\boldsymbol{\mu}_k]. \quad (2.35)$$

Por lo tanto, si se normaliza el ancho de la imagen ( $\rho_{k,x}$ ), los factores escalares puede ser calculados como

$$\gamma_1 = \frac{(\mathcal{H}[\boldsymbol{\mu}_b] \times \mathcal{H}[\boldsymbol{\mu}_c])^T \mathcal{H}[\boldsymbol{\mu}_k]}{(\mathcal{H}[\boldsymbol{\mu}_a] \times \mathcal{H}[\boldsymbol{\mu}_b])^T \mathcal{H}[\boldsymbol{\mu}_k]}, \quad \text{y} \quad \gamma_2 = \gamma_1 / \xi. \quad (2.36)$$

De forma similar, si la altura de la imagen está normalizada ( $\rho_{k,y}$ ), entonces

$$\gamma_2 = \frac{(\mathcal{H}[\boldsymbol{\mu}_a] \times \mathcal{H}[\boldsymbol{\mu}_c])^T \mathcal{H}[\boldsymbol{\mu}_k]}{(\mathcal{H}[\boldsymbol{\mu}_a] \times \mathcal{H}[\boldsymbol{\mu}_b])^T \mathcal{H}[\boldsymbol{\mu}_k]}, \quad \text{y} \quad \gamma_1 = \xi \gamma_2. \quad (2.37)$$

El algoritmo propuesto requiere solo cuatro puntos  $\boldsymbol{\mu}$  que indican las esquinas de un rectángulo observado. Los puntos  $\boldsymbol{\mu}$  dados como entrada se usan para determinar las líneas de los contornos del rectángulo observado. Las líneas resultantes se usan para determinar tres puntos:  $\boldsymbol{\mu}_a$  (la imagen del punto al infinito en dirección  $x$ ),  $\boldsymbol{\mu}_b$  (la imagen del punto al infinito en dirección  $y$ ) y  $\boldsymbol{\mu}_c$

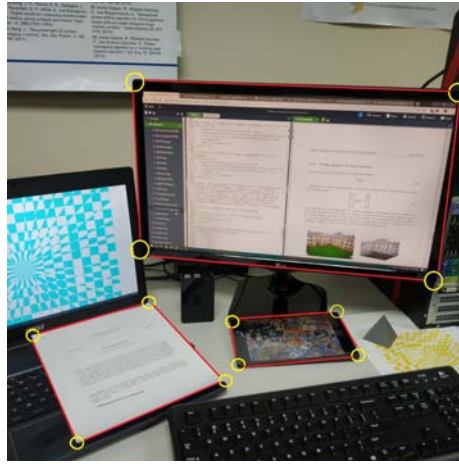


Figura 2.4: Captura de cuatro puntos por cada rectángulo observado en la imagen.

(la imagen del punto en el origen). Este algoritmo se puede aplicar al caso en donde existan  $n$  líneas paralelas a cada eje del plano observado. En la siguiente subsección se analiza el proceso de obtener los puntos de intersección de las líneas paralelas.

Como se ha mencionado previamente, las homografías son muy útiles para aplicaciones de procesamiento de imágenes. A continuación, se describirá la aplicación de las matrices homografía para realizar corrección de perspectiva, generación de imágenes panorámicas, y construcción de imágenes compuestas.

### 2.1.5. Corrección de perspectiva

En esta aplicación, se reciben como entrada cuatro puntos  $\mu$  en la imagen, y se entrega como salida una imagen sin distorsión de perspectiva. En la figura 2.4 se muestra las capturas de puntos  $\mu$  para cada objeto de interés. El conjunto de puntos  $\mu$  perteneciente al objeto de interés, se calcula una homografía para re-proyectar la imagen al plano de referencia. En la figura 2.5), se puede observar los resultados de re-proyección de cada objeto de interés.

Los resultados obtenidos validaron experimentalmente la relación entre plano imagen y plano de referencia. La geometría que establece esta relación esta dada por la homografía y se puede calcular de manera eficiente con los métodos de estimaciones descritas en las subsecciones anteriores.

### 2.1.6. Generación de imágenes panorámicas

En esta aplicación, se reciben como entrada múltiples imágenes y se entrega como resultado una imagen panorámica. Se calcula una homografía por cada

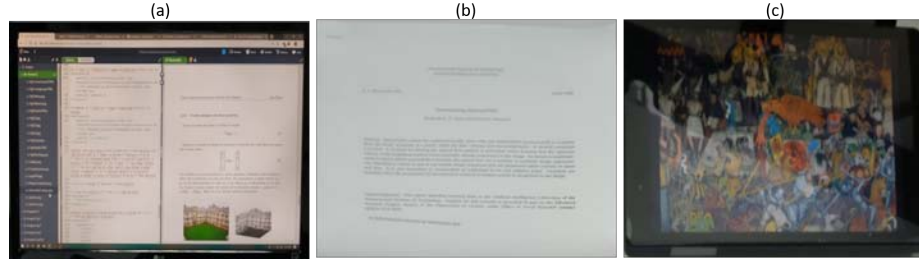


Figura 2.5: Objetos seleccionados con perspectiva corregida. (a) Monitor de computadora. (b) Documento impreso en hoja de papel. (c) Pantalla de tableta electrónica.

imagen usando rectángulos de la escena, por ejemplo, los monitores de computadora en el caso de las figuras 2.6(a)-(c). Las homografías calculadas son usadas para re-proyectar las imágenes proporcionadas a un único plano de referencia. El resultado es una imagen panorámica como se muestran en la figura 2.6(d).

Con esta aplicación, se demuestra experimentalmente el plano de referencia puede ser único y corresponder a múltiples homografías. Dado a esta particularidad, la información de interés puede ser compartida en el plano de referencia para una secuencia de video.

### 2.1.7. Construcción de imágenes compuestas

En esta aplicación, se reciben como entrada múltiples imágenes y se entrega como resultado una vista tridimensional de la escena. En la aplicación se calculan las homografías de cada imagen, ver figura 2.7. Las homografías de las imágenes correspondientes a un mismo plano (piso, paredes, techo) son usadas para generar imágenes panorámicas. Nuevamente, se determina una homografía por cada imagen panorámica generada, ver figura 2.8. Finalmente, las homografías son usadas para re-proyectar las imágenes panorámicas en las paredes de una escena tridimensional virtual como se muestra en la figura 2.9.

Los resultados obtenidos muestran la flexibilidad y la facilidad en manejo de planos. La geometría del sistema permitió realizar la re-proyección de múltiples planos hasta formar una escena tridimensional.

## 2.2. Calibración de cámaras usando homografías

Para describir el proceso de formación de imagen de una cámara, es necesario conocer sus parámetros intrínsecos y extrínsecos. Los parámetros intrínsecos representan características internas de la cámara, como la distancia focal, distorsión, oblicuidad y el punto central de la imagen. Los parámetros extrínsecos proporcionan la posición y orientación de la cámara. El proceso de la estimación de los parámetros es mejor conocido como *calibración*.



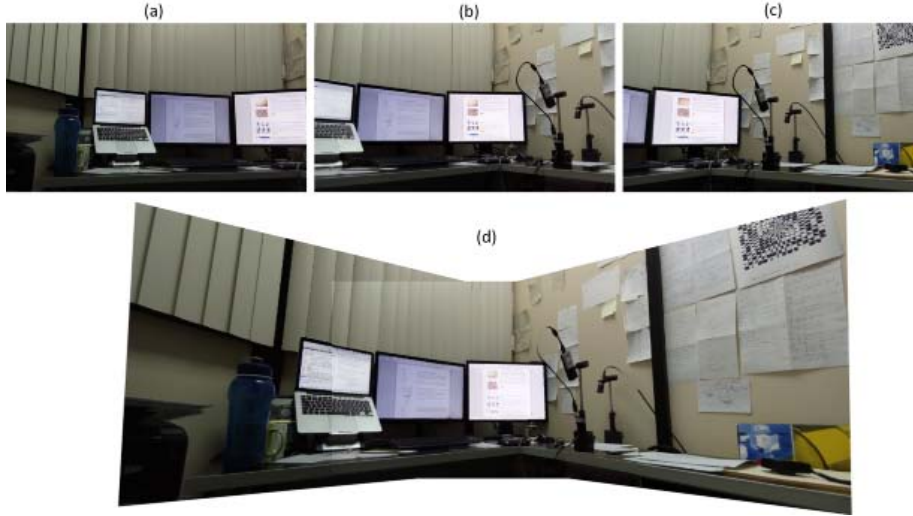


Figura 2.6: Composición de imágenes panorámicas. (a)-(c) Imágenes de entrada. (d) Imagen panorámica resultante.

La calibración es una tarea esencial para aplicaciones de procesamiento de imágenes. La disponibilidad de estos parámetros permite realizar triangulación entre dispositivos, calcular la homografía, y determinar la pose de los objetos, entre otras [40, 43]. En esta sección se presenta un método para estimar los parámetros intrínsecos y extrínsecos usando múltiples homografías.

### 2.2.1. Parámetros intrínsecos

Para estimar los parámetros intrínsecos, se toman las siguientes consideraciones. El plano de referencias coincide con el plano  $xy$  del sistema de referencia global, es decir, no existe rotación ni traslación en el plano de referencia. En este caso, la orientación del plano es  $\mathbf{q}_1 = [1 \ 0 \ 0]^T$  y  $\mathbf{q}_2 = [0 \ 1 \ 0]^T$ . Usando las consideraciones anteriores, la Ec. (2.18) se reduce a

$$G = K[\bar{\mathbf{r}}_1 \ \bar{\mathbf{r}}_2 \ -R^T \mathbf{t}], \quad (2.38)$$

donde  $\bar{\mathbf{r}}_1$  y  $\bar{\mathbf{r}}_2$  son los dos primeros renglones de la matriz de rotación  $R$  de la cámara. Por otro lado, la representación explícita de la matriz  $G$  es mostrada en la Ec. (2.18). Entonces, la Ec. (2.38) se reescribe como

$$[\bar{\mathbf{r}}_1 \ \bar{\mathbf{r}}_2 \ -R^T \mathbf{t}] = K^{-1}[\mathbf{g}_1 \ \mathbf{g}_2 \ \mathbf{g}_3]. \quad (2.39)$$

Los vectores  $\bar{\mathbf{r}}_1$  y  $\bar{\mathbf{r}}_2$  son ortogonales. Por lo tanto, se puede aprovechar que  $\bar{\mathbf{r}}_1^T \bar{\mathbf{r}}_2 = 0$  y  $\|\bar{\mathbf{r}}_1\|^2 = \|\bar{\mathbf{r}}_2\|^2$ , obteniendo

$$\mathbf{g}_1^T W \mathbf{g}_2 = 0, \quad \mathbf{g}_1^T W \mathbf{g}_1 = \mathbf{g}_2^T W \mathbf{g}_2, \quad (2.40)$$

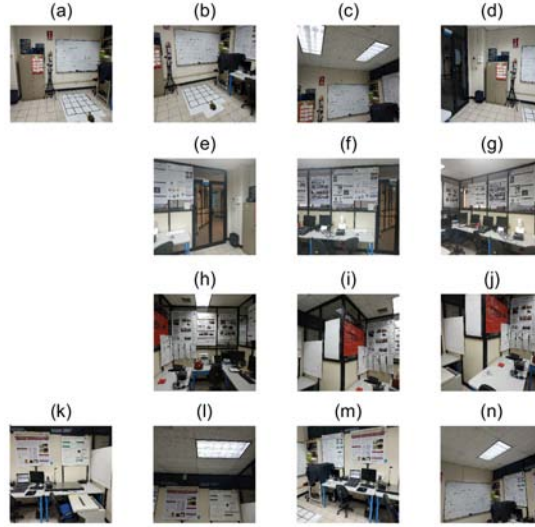


Figura 2.7: Imágenes capturadas de una escena en diferentes puntos de observación. (a)-(d) Conjunto de imágenes correspondiente a una pared con objetos. (e)-(g), (h)-(j), y (k)-(n) Conjuntos de imágenes de paredes, piso y techo de la escena observado.

donde la matriz asimétrica  $W$  es definida como

$$W = K^{-T} K^{-1} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{12} & w_{22} & w_{23} \\ w_{13} & w_{23} & w_{33} \end{bmatrix}. \quad (2.41)$$

La forma bilineal  $\mathbf{g}_i^T W \mathbf{g}_j^T$  puede expresarse como

$$\mathbf{g}_i^T W \mathbf{g}_j^T = \mathcal{V}_{ij}[G] \mathbf{w}, \quad (2.42)$$

donde

$$\mathcal{V}_{ij}[G] = \begin{bmatrix} g_{1i}g_{1j} \\ g_{2i}g_{2j} \\ g_{3i}g_{3j} \\ g_{2i}g_{1j} + g_{1i}g_{2j} \\ g_{3i}g_{1j} + g_{1i}g_{3j} \\ g_{3i}g_{2j} + g_{2i}g_{3j} \end{bmatrix}^T, \quad (2.43)$$

y

$$\mathbf{w} = [w_{11} \quad w_{22} \quad w_{33} \quad w_{12} \quad w_{13} \quad w_{23}]^T. \quad (2.44)$$

Por lo tanto, las igualdades escritas en la Ec. (2.40) se pueden expresar como

$$V[G] \mathbf{w} = \mathbf{0}_2, \quad (2.45)$$

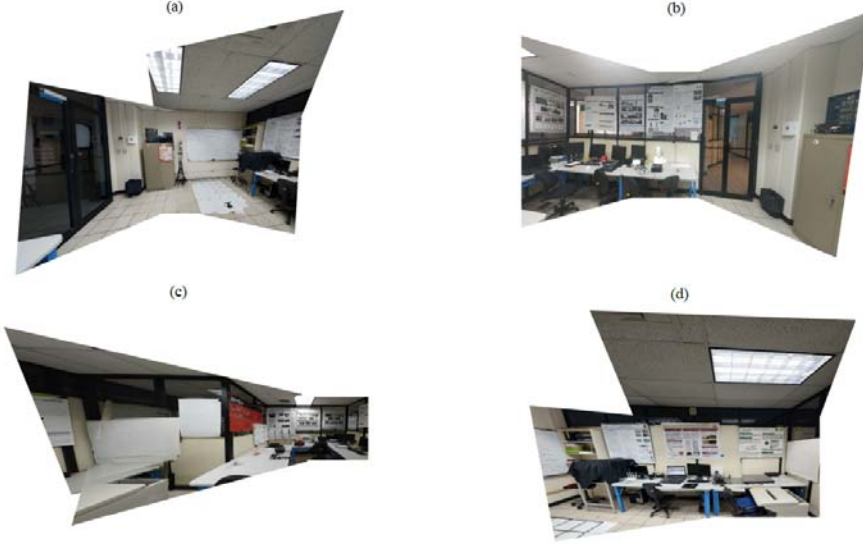


Figura 2.8: Imágenes compuestas generadas por los conjuntos de imágenes capturadas.

donde  $V[G]$  es una matriz de tamaño  $2 \times 6$  generada como

$$V[G] = \begin{bmatrix} \mathcal{V}_{12}[G] \\ \mathcal{V}_{11}[G] - \mathcal{V}_{22}[G] \end{bmatrix}. \quad (2.46)$$

La solución para encontrar  $\mathbf{w}$  en la Ec. (2.45) es usar varias homografías  $\mathbf{G}_k, k = 1, 2, 3, \dots, m$ . Para esto, las homografías se estiman en diferentes imágenes donde se varía la posición y la orientación del plano de referencia, mientras los parámetros intrínsecos se mantienen constantes. Entonces, esta solución forma una nueva matriz

$$\mathbb{V}\mathbf{w} = \mathbf{0}_{2m}, \quad (2.47)$$

donde

$$\mathbb{V} = [V[G_1]^T \quad V[G_2]^T \quad \dots \quad V[G_m]^T]^T. \quad (2.48)$$

La solución para encontrar el vector  $\mathbf{w}$  de la Ec. (2.47) se puede obtener usando el método de descomposición de valores singulares. El vector resultante debe ser normalizado debido a que la matriz  $W$  es única excepto escala. Es decir, el vector estimado, llamémosle  $\tilde{\mathbf{w}}$ , se relaciona con el producto de la matriz  $W$  por una constante  $\lambda$ . Esta relación se expresa matemáticamente como

$$\tilde{W} = \lambda W = \lambda K^{-T} K^{-1}. \quad (2.49)$$

Utilizando la matriz  $\tilde{W}$ , las entradas  $k_{ij}$  de la matriz de parámetros intrínsecos

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.50)$$

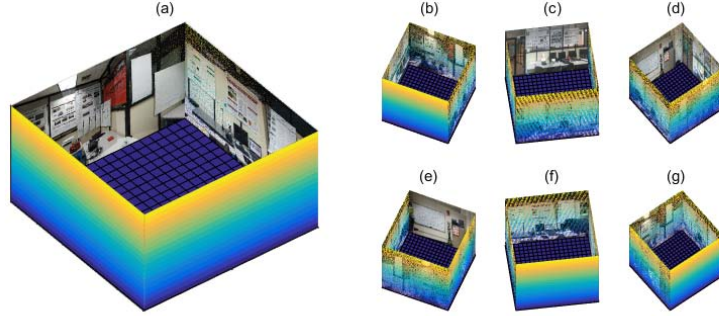


Figura 2.9: (a) Escena tridimensional creada con imágenes compuestas. (b)-(g) Escena en diferentes ángulos.

son dadas como

$$\begin{aligned}
 \lambda &= (\det \tilde{W})/d, \\
 k_{11} &= \sqrt{\lambda/\tilde{w}_{11}}, \\
 k_{22} &= \sqrt{\lambda/\tilde{w}_{11}d}, \\
 k_{12} &= -\tilde{w}_{12}\sqrt{\lambda/\tilde{w}_{11}d}, \\
 k_{13} &= (\tilde{w}_{12}\tilde{w}_{23} - \tilde{w}_{22}\tilde{w}_{13})/d, \\
 k_{13} &= (\tilde{w}_{12}\tilde{w}_{13} - \tilde{w}_{11}\tilde{w}_{23})/d.
 \end{aligned} \tag{2.51}$$

### 2.2.2. Parámetros extrínsecos

El último paso de la calibración, es encontrar los parámetros extrínsecos usando los parámetros intrínsecos disponibles. La matriz de rotación  $R$  y el vector de rotación  $\mathbf{t}$  se pueden estimar usando una homografía. Primero, se estima  $\tilde{R}^T$  de la matriz  $R^T$  como

$$\tilde{R}^T = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_1 \times \mathbf{h}_2], \tag{2.52}$$

donde a través de la Ec. (2.39), los vectores  $\mathbf{h}_1$  y  $\mathbf{h}_2$  son dados por

$$\mathbf{h}_1 = K^{-1}\mathbf{g}_1, \quad \mathbf{h}_2 = K^{-1}\mathbf{g}_2. \tag{2.53}$$

Después, la matriz de rotación  $R$  es obtenida desde  $\tilde{R}$  con la condición de ortogonalidad que mantienen las matrices de rotación. Para esto, la descomposición de valores singulares  $\tilde{R} = U\Sigma V^T$  es obtenida y la matriz de rotación requerida es determinada como

$$R = UV^T. \tag{2.54}$$

Finalmente, el vector de translación  $\mathbf{t}$  es calculado como

$$\mathbf{t} = -RK^{-1}\mathbf{g}_3. \tag{2.55}$$

## 2.3. Distorsión radial

La distorsión radial es una fuente de ruido en imágenes que introducen las lentes de la mayoría de las cámaras. Esta degradación es más susceptible en las cámaras de bajo costo. La distorsión radial afecta los resultados de algoritmos de análisis y extracción de información métrica. Por esta razón, la distorsión radial suele afectar severamente el desempeño en los algoritmos que se van a utilizar para la navegación visual. A continuación, analizaremos un método de estimación para obtener los parámetros de distorsión y a su vez corregir imágenes distorsionadas [44].

### 2.3.1. Modelo polinomial

La distorsión radial se puede modelar de acuerdo a la relación entre un punto con distorsión ( $\delta$ ) y un punto sin distorsión ( $\beta$ ), como se muestra en la figura 2.10. Esta relación se expresa matemáticamente como

$$\delta = d(r)[\beta - \mathbf{c}] + \mathbf{c}, \quad (2.56)$$

donde

$$r = \|\beta - \mathbf{c}\|, \quad (2.57)$$

con  $\|\cdot\|$  indicando la norma Euclidiana,  $\mathbf{c} = [c_x, c_y]^T$  es el punto principal de la cámara y  $d(r)$  es una función escalar modulando el nivel de la distorsión. La función  $d(r)$  se puede deducir a partir de argumentos físicos aplicando los principios de propagación de la luz [45]. Sin embargo, la función  $d(r)$  se puede aproximar a través de una serie de Taylor truncada como

$$d(r) = d_0 + d_1 r + d_2 r^2, \quad (2.58)$$

donde  $d_0$ ,  $d_1$  y  $d_2$  son coeficientes constantes. La serie de Taylor es reducida hasta el segundo orden dado que la aproximación es confiable para la función  $d(r)$ .

En las ecuaciones (2.56) y (2.58) se puede observar que la distorsión radial introducida por la lente está caracterizada por los coeficientes  $d_k$  de la función  $d(r)$  y las coordenadas  $(c_x, c_y)$  del punto principal  $\mathbf{c}$ . Estos parámetros pueden ser estimados por el algoritmo de Gauss-Newton como se describe a continuación.

Sea  $\mathbf{d}$  el vector que contiene todos los coeficientes que caracterizan la distorsión radial; es decir,

$$\mathbf{d} = [d_0 \quad d_1 \quad d_2 \quad c_x \quad c_y]^T. \quad (2.59)$$

Para el vector de parámetros  $\mathbf{d}$ , la matriz Jacobiana del modelo es

$$J(\mathbf{d}) = \frac{\partial}{\partial \mathbf{d}} \delta = \begin{bmatrix} \partial \delta_x / \partial \mathbf{d} \\ \partial \delta_y / \partial \mathbf{d} \end{bmatrix}. \quad (2.60)$$

De esta forma, los parámetros del vector se pueden estimar iterativamente por

$$\mathbf{d}_{k+1} = \mathbf{d}_k + [J^T(\mathbf{d}_k)J(\mathbf{d}_k)]^{-1} J^T(\mathbf{d}_k)[\delta - \delta(\mathbf{d}_k)], \quad (2.61)$$

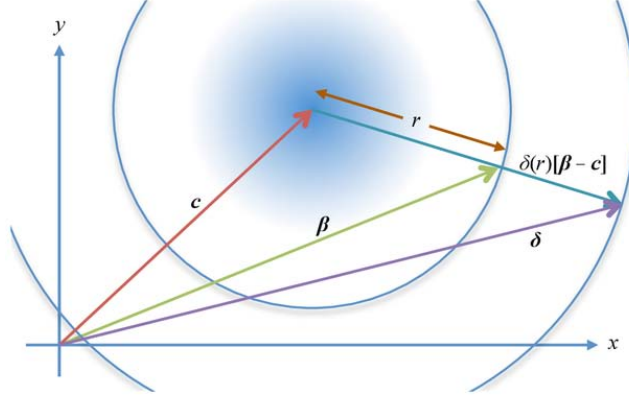


Figura 2.10: La distorsión radial de acuerdo a la transformación de un punto sin distorsión  $\beta$  a un punto distorsionado  $\delta$ .

donde  $\delta(\mathbf{d}_k)$  es el punto con distorsión calculada usando  $\mathbf{d}_k$ . Los parámetros iniciales pueden ser  $\mathbf{d}_0 = [1, 0, 0, 0, 0]^T$ , es decir, sin distorsión y el punto principal ubicado en el centro de la imagen (origen del sistema de referencia). Hay dos criterios de paro, el primero es el número de iteraciones máximas superadas para encontrar la solución. Y el segundo es la convergencia relativa descrita como  $\|\mathbf{d}_{k+1} - \mathbf{d}_k\| \leq \epsilon$ , donde  $\epsilon$  es un valor numérico dado por el usuario.

Usando los parámetros de distorsión estimados, las imágenes distorsionadas se pueden corregir utilizando la Ec. (2.56). Esto mejora significativamente los resultados para los algoritmos de detección, reduce los falsos positivos en los métodos utilizados y en general, la estabilidad al sistema.

Finalmente, utilizando los parámetros de distorsión estimados se puede corregir la distorsión radial en las imágenes capturadas usando la Ec. (2.56). En el capítulo 3 analizará de forma experimental la mejora en los resultados al procesar imágenes sin distorsión.

### 2.3.2. Corrección de distorsión radial

Para ilustrar la aplicación de las matrices homografía, así como el funcionamiento del modelo de distorsión propuesto, se presenta a continuación un ejemplo de corrección de perspectiva y distorsión de una imagen.

Primeramente, se mostrará una simulación que se realizó para validar el método de la estimación de distorsión radial. Usando la Ec. (2.58) y una distorsión predefinida  $\mathbf{d} = [1, 5, 0, 5, 0, 6, 0, 5, 0, 5]$ , se genera una rejilla distorsionada dada por

$$I = \frac{1}{4}(2 + \cos(10\pi\delta_x) + \cos(10\pi\delta_y)). \quad (2.62)$$

Enseguida, la rejilla es usada como entrada para estimar la distorsión radial; como parámetro inicial se utilizó  $\mathbf{d} = [1, 0, 0, 0, 0]$ . En la figura 2.11, se pue-

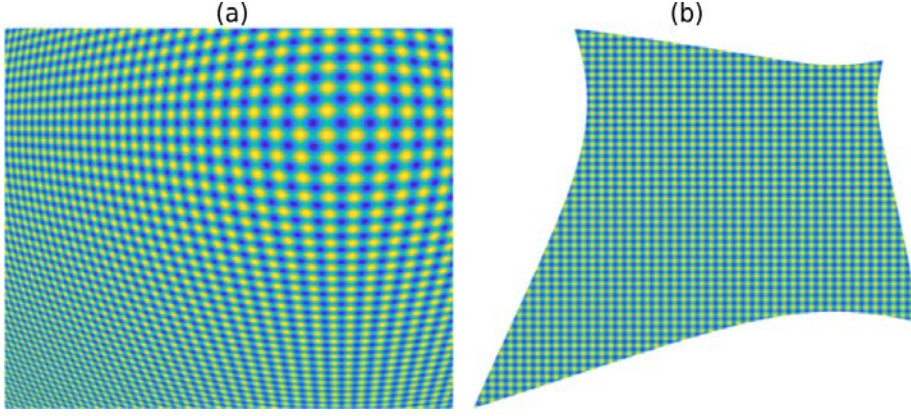


Figura 2.11: Simulación de distorsión de lentes. (a) Imagen de una rejilla con distorsión radial. (b) Imagen de la rejilla en (a) con distorsión radial corregida.

de observar la rejilla distorsionada y la corrección de la distorsión usando los parámetros estimados.

En la figura 2.12(a) se puede observar una imagen distorsionada. Los parámetros de distorsión encontrados fueron  $d = [1, 0, 1, 0, 6, 0, 0]$  en la sexta iteración con un error de  $\epsilon = 7,26e - 7$ . Con los parámetros encontrados y usando la Ec. (2.56), se calculan las escalas de la distorsión y se utilizan para corregir la perspectiva y distorsión de la imagen, como se muestra en la figura 2.12(b). En la figura 2.12(c), se centra una región de interés en la imagen corregida.

## 2.4. Detección de contornos

La detección de contornos es una tarea esencial en el procesamiento de imágenes. Específicamente, se utilizan en sistemas de detección y extracción de características. Esto permite detectar la discontinuidad o el cambio de brillo en los puntos de la imagen. El método tradicional utilizado para la detección de contornos es el método del gradiente que se describe a continuación.

### 2.4.1. Método del gradiente

Los contornos en una imagen pueden interpretarse como cambios bruscos de intensidad. En términos matemáticos, se puede asociar los contornos de la imagen con discontinuidades en las derivadas de la intensidad. Este es el principio general del funcionamiento del método del gradiente para detección de contornos.

La derivada de la intensidad  $I(x, y)$  a lo largo de la dirección  $x$  está dada por  $\partial I(x, y) / \partial x$ . Para detectar discontinuidades, consideramos las derivadas de



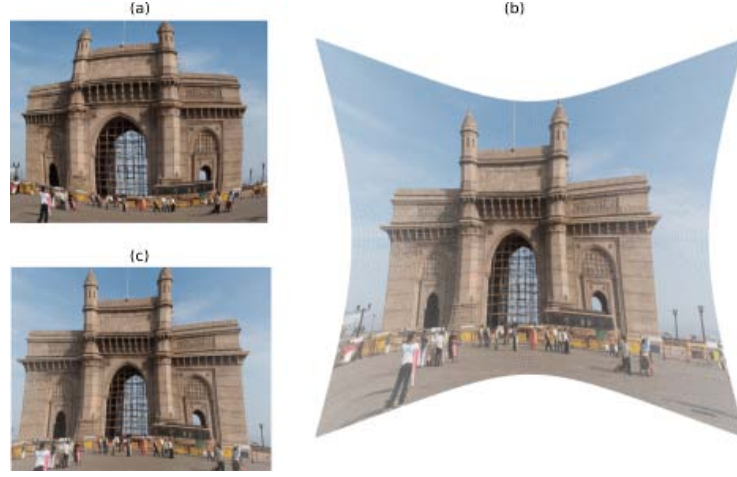


Figura 2.12: Corrección de perspectiva y distorsión de una imagen. (a) Imagen con perspectiva y distorsión radial. (b) Imagen corregida, y (c) la región de interés.

la intensidad en los ejes  $x$  e  $y$  a través del operador gradiente como

$$\nabla I(x, y) = \frac{\partial I(x, y)}{\partial x} \mathbf{i} + \frac{\partial I(x, y)}{\partial y} \mathbf{j}, \quad (2.63)$$

donde  $\nabla$  representa el operador gradiente,  $\mathbf{i}$ ,  $\mathbf{j}$  representan vectores unitarios en la dirección  $x$  e  $y$ , respectivamente, y

$$\begin{aligned} \frac{\partial I(x, y)}{\partial x} &= \lim_{\Delta x \rightarrow 0} \frac{I(x + \Delta x, y) - I(x, y)}{\Delta x}, \\ \frac{\partial I(x, y)}{\partial y} &= \lim_{\Delta y \rightarrow 0} \frac{I(x, y + \Delta y) - I(x, y)}{\Delta y}. \end{aligned} \quad (2.64)$$

De esta forma, la ubicación de contornos en la imagen corresponderá a aquellos puntos  $(x, y)$  donde la magnitud del gradiente

$$|\nabla I| = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}}, \quad (2.65)$$

tome valores altos; es decir

$$|\nabla I| \geq u, \quad (2.66)$$

donde  $|\cdot|$  representa la magnitud del argumento vectorial dado y  $u$  es un valor de umbral dado.

El método del gradiente es simple y fácil de implementar computacionalmente. Sin embargo, debido al uso de derivadas, el ruido de las imágenes suele amplificarse considerablemente. En consecuencia, el resultado de la detección



contiene contornos en puntos continuos, o de lo contrario, los puntos discontinuos son ignorados. Para limitar los efectos del ruido, se puede hacer un filtrado inicial que suprimen el ruido, y luego detectar los contornos. Existen en la literatura otras propuestas para detección de contornos en donde se incluye un filtro de ruido [46–48]. Estos métodos mejoran considerablemente el desempeño de la detección de contornos. Entre los métodos propuestos, uno de los más exitosos es el método de detección de contornos de *Canny* [49].

### 2.4.2. Método de Canny

El método de Canny toma en consideración tres criterios para mejorar el desempeño de la detección de contornos. El primer criterio es la buena detección. Este criterio maximiza la relación señal/ruido para incrementar o reducir la probabilidad en detectar puntos continuos como discontinuidad. El segundo criterio es la buena localización. Un operador decide cuáles puntos se consideran contornos de acuerdo al contorno central. Por último, el criterio de selección de contornos. Se descartan las señales falsas cuando se registran múltiples respuestas de la detección.

De acuerdo a los criterios mencionados, se suaviza la imagen para reducir el ruido mejorando la detección de contornos. La suavización se realiza a través de la convolución entre la primera derivada de una señal bidimensional Gaussiana  $G_n(x, y)$  en dirección  $\mathbf{n}$  y la imagen  $I(x, y)$ , es decir

$$G_n(x, y) * I(x, y), \quad (2.67)$$

donde “\*” es el operador de convolución,

$$G_n(x, y) = \mathbf{n} \cdot \nabla G(x, y), \quad (2.68)$$

$\mathbf{n}$  es el vector de dirección, “.” representa el operador de producto punto,

$$G(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (2.69)$$

y  $\sigma^2$  es la desviación estándar de la señal Gaussiana. La dirección  $\mathbf{n}$  no es conocida y por lo general, debe ser orientada a la misma dirección que los contornos para ser detectados. Sin embargo, se puede realizar una aproximación usando la dirección del gradiente suavizado

$$\hat{\mathbf{n}} = \frac{\nabla(G(x, y) * I(x, y))}{|\nabla(G(x, y) * I(x, y))|}. \quad (2.70)$$

La dirección  $\hat{\mathbf{n}}$  calculada resulta una buena aproximación debido a la suavidad aplicada a la dirección del gradiente del contorno. Por otro lado, un punto se considera discontinuo cuando el operador  $G_n(x, y)$  aplicado en la imagen  $I(x, y)$  es el máximo local, es decir,

$$G_n(x, y) * I(x, y) \approx 0. \quad (2.71)$$

Asociando la convolución Gaussiana con la Ec. (2.68), se puede reescribir la Ec. (2.71) como

$$\frac{\partial^2}{\partial n^2} G(x, y) * I(x, y) = 0. \quad (2.72)$$

De la misma forma, se utiliza la magnitud para determinar la intensidad de los contornos detectados

$$|G_n(x, y) * I(x, y)| = |\nabla(G(x, y) * I(x, y))|. \quad (2.73)$$

Usando la propiedad asociativa de la convolución, en la Ec. (2.72) se puede aplicar, primero, la convolución entre la señal Gaussiana  $G(x, y)$  y la imagen  $I(x, y)$ . Después, se calcula la segunda derivada parcial respecto a  $n$  para detectar los contornos. Esto es equivalente a detectar y localizar los contornos usando el operador  $G_n(x, y)$  pero sin necesidad de conocer la dirección  $\mathbf{n}$ .

La segunda derivada parcial de la Ec. (2.72) se resuelve usando un filtro Gaussiano  $n$ -dimensional [50]. Este filtro se puede construir de manera eficiente considerando múltiples filtros Gaussiano unidimensional como

$$G_{nD}(\boldsymbol{\theta}) = G_{1D}(\theta_1) * G_{1D}(\theta_2) * G_{1D}(\theta_3) * \cdots * G_{1D}(\theta_n), \quad (2.74)$$

donde  $G_{nD}(\mathbf{x})$  es el filtro Gaussiano  $n$ -dimensional y  $G_{1D}(x_k)$  es el  $k$ -ésimo filtro Gaussiano de una dimensión. Un filtro  $G_{1D}(x)$  esta compuesto por operadores de dirección para ajustar el contorno de un punto observado. El operador de dirección  $O(\theta)$  se determina por un filtro Gaussiano normalizado de una dimensión en dirección de  $\theta$ ; específicamente,

$$O(\theta) = \frac{\cos \theta}{2N + 1} \left[ \sum_{k=-N}^N \exp \left( -\frac{(kd \sin \theta)^2}{2\sigma^2} \right) \right], \quad (2.75)$$

donde  $N$  es el número de muestras, y  $d$  es la distancia del punto local con el punto observado. El tamaño de un filtro Gaussiano unidimensional es  $2N + 1$  operadores de dirección. Para casos específicos, el ángulo  $\theta$  es dado por el usuario. En casos generales, el ángulo  $\theta$  se puede determinar con incrementos de  $180/n$ . Por ejemplo, si quiero cuatro filtros Gaussiano unidimensionales, los ángulos serían  $\boldsymbol{\theta} = [0, 45, 90, 135]$ . Por la disponibilidad de los operadores, se pueden calcular los puntos discontinuos sustituyendo la segunda derivada parcial de la Ec. (2.72) por una convolución con el filtro de la Ec. (2.74).

Los contornos son una fuente de información para el algoritmo de detección de líneas. Sin embargo, en la escena observada por la cámara tendrá mucha más información de objetos que no son de interés; por ejemplo, árboles, edificios, señales de tránsito, vehículos, postes de cables eléctricos, etc. Toda esta información puede hacer más difícil la detección de las líneas de carril. Por esta razón, se propone el uso de un filtro de ventana que elimina los contornos que están fuera del área donde se espera observar el carril. Este filtro eliminará los contornos que no contribuirán a la detección de las líneas de carril. Además, de esta forma se incrementa el desempeño de la transformada de Hough debido a que se reduce el número de píxeles a procesar.

serán detectados los contornos de deferentes objetos que no son de interés,

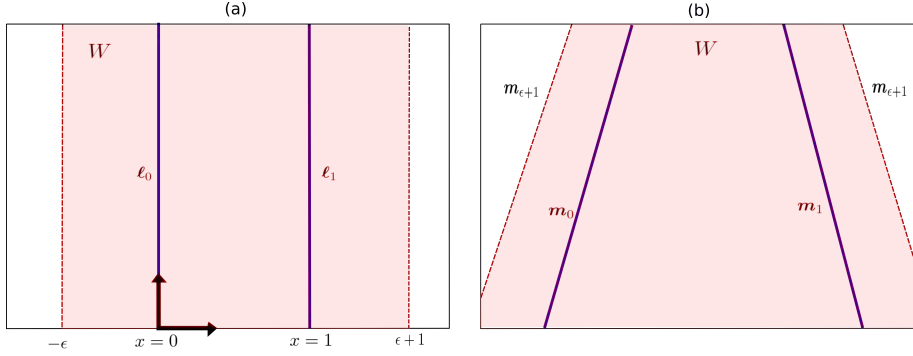


Figura 2.13: Representaciones de las líneas de carril en (a) el piso o plano de referencia, y (b) el plano imagen.

### 2.4.3. Filtro de ventana

La idea general para construir un el filtro de ventana consiste en definir un área en donde se espera que se encuentre el carril de interés. Para ello, usamos como referencia la ubicación relativa de la cámara y la pista detectada en imágenes previas. Además, partimos del hecho de que la configuración cámara-piso es constante y definida por una matriz homografía. Esta homografía se puede obtener previamente a través del método de estimación usando cuadriláteros (ver la subsección 2.1.4).

Supondremos que sobre el piso, las líneas de carril son las líneas  $x = 0$  y  $x = 1$  como se muestra en la figura 2.13(a). La representación de estas líneas de carril en coordenadas homogéneas es

$$\begin{aligned} \ell_0 &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, & (\text{Línea de carril izquierda}), \\ \ell_1 &= \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, & (\text{Línea de carril derecha}). \end{aligned} \quad (2.76)$$

En la imagen capturada por la cámara, las líneas de carril  $\ell_0$  y  $\ell_1$  serán observadas como las líneas  $m_0$  y  $m_1$  como se muestra en la figura 2.13(b). Si  $G$  es la matriz homografía que define la geometría entre el piso y el plano imagen, entonces podemos calcular  $m_0$  y  $m_1$  a partir de  $\ell_0$  y  $\ell_1$  como

$$m_k = G^{-T} \ell_k, \quad k = 0, 1. \quad (2.77)$$

Las líneas observadas  $m_0$  y  $m_1$  definen un área que es difícil de describir matemáticamente para definir la región de interés en la imagen. Por esta razón, en lugar de considerar el plano imagen, el filtro de ventana es construido basándose en el plano de referencia donde las líneas de carril  $\ell_0$  y  $\ell_1$  siempre son fijas y verticales.

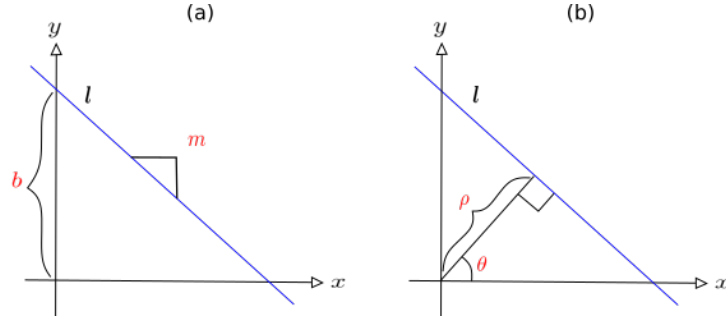


Figura 2.14: Representación de una línea parametrizada: (a)  $y = mx + b$  y (b)  $\rho = x \cos \phi + y \sin \phi$ .

Usando la matriz homografía  $G$ , transformamos las coordenadas  $\mu$  de puntos del plano imagen a sus correspondientes puntos  $\rho$  del plano de referencia como

$$\rho(\mu) = \begin{bmatrix} \rho_x(\mu) \\ \rho_y(\mu) \end{bmatrix} = \mathcal{H}^{-1}[G^{-1}\mathcal{H}[\mu]]. \quad (2.78)$$

De esta manera, con el filtro de ventana  $W$  determinamos si el punto  $\mu$  está dentro o fuera de la región de interés de acuerdo a

$$W(\mu) = \begin{cases} 0, & \text{si } \rho_x(\mu) > 1 + \epsilon, \text{ o } \rho_x(\mu) < -\epsilon, \\ 1, & \text{caso contrario,} \end{cases} \quad (2.79)$$

donde  $\epsilon$  es un valor positivo dado por el usuario, con la finalidad de que la ventana  $W(\mu)$  contenga las líneas de carril de interés, como se observa en la región sombreada mostrada en la figura 2.13.

## 2.5. Detección de líneas

La detección de líneas es un método de gran importancia para uso de aplicaciones, tales como, la navegación autónoma, calibración de cámaras, recuperación de objetos tridimensionales, compresión de datos, y rectificación de imágenes, entre otras [51–55]. Las líneas poseen propiedades geométricas para determinar los parámetros de posición y orientación [56]. Con base en lo anterior, se utiliza esta fuente de información para la navegación visual.

La transformada de Hough se usa típicamente para detección de líneas, círculos, cónicas, y, en general, curvas cuya representación paramétrica esté disponible [57]. Dado que pueden existir múltiples parametrizaciones de un objeto, la implementación de la transformada de Hough para detectar algún objeto en específico no es única [58]. Para la detección de líneas, podríamos elegir la parametrización  $(m, b)$ ,

$$y = mx + b, \quad (2.80)$$

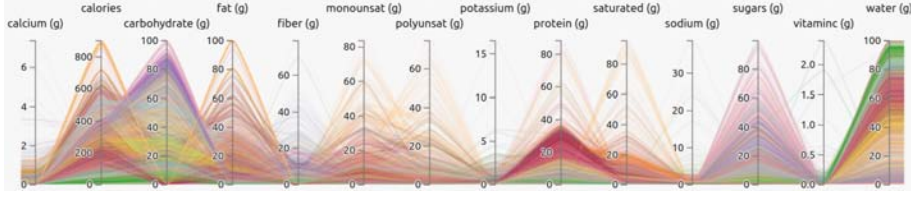


Figura 2.15: Representación de nutrientes en la dieta de un paciente graficados usando coordenadas paralelas<sup>2</sup>.

o la parametrización  $(\phi, \rho)$ ,

$$\rho = x \cos \phi + y \sin \phi. \quad (2.81)$$

En la figura 2.14 se muestra gráficamente cada una de las representaciones descritas. Es importante ver que las ecuaciones (2.80) y (2.81) conducirán a implementaciones de la transformada de Hough muy diferentes. Lo más evidente es que la parametrización dada por la ecuación (2.80) no puede representar líneas verticales. Por otro lado, la implementación de la transformada de Hough usando la parametrización dada por la ecuación (2.81) requiere mayor capacidad computacional debido al manejo de curvas cosenoidales. Este ejemplo refleja la importancia de elegir adecuadamente una parametrización que permita implementar la transformada de Hough de manera eficiente [58].

El principio de funcionamiento de la transformada de Hough se basa en acumular las posibles soluciones a la que pertenezca un punto en la imagen. Por ejemplo, si se pretende detectar líneas, entonces todo punto  $p$  en la imagen que posiblemente pertenece a una línea se representa en el plano de Hough abarcando todas las posibles líneas que pasan por  $p$ . Este proceso se puede realizar eficientemente empleando representación de puntos y líneas en coordenadas paralelas como veremos a continuación.

### 2.5.1. Coordenadas paralelas

Las coordenadas paralelas fueron introducidas con la finalidad de representar gráficamente objetos que existen en múltiples dimensiones. Este sistema de coordenadas es de mucha utilidad para el procesamiento de imágenes. Además, sirve para analizar otros problemas de diferentes áreas de estudio. Por ejemplo, se puede realizar una gráfica alimenticia sobre de la tabla nutricional de varios productos. Los ejes paralelos corresponde para cada grupo de macronutrientes y la información generada puede aportar una sana alimentación para el paciente. En la figura 2.15 se muestra el ejemplo descrito donde indica el valor nutricional de cada producto.

La literatura ha demostrado que el uso de coordenadas paralelas proporciona una forma bastante eficiente de trabajar con líneas rectas [59,60]. Por esta razón,

<sup>2</sup>Imagen tomada de <https://www.d3-graph-gallery.com/parallel>.

se utilizan coordenadas paralelas y la transformada de Hough para el diseño del algoritmo de detección de líneas.

### 2.5.2. Transformada de Hough

Antes de comenzar a describir el método de la transformada de Hough, definiremos primero el espacio de cada plano. El plano imagen utiliza las coordenadas cartesianas  $(x, y)$  y el plano de Hough utiliza las coordenadas cartesianas  $(u, v)$  como se muestran en la figura 2.16. Por lo tanto, un punto en el plano imagen se representa como  $\mathbf{p} = [p_x, p_y]^T$ , ver figura 2.16(a). Asimismo, un punto  $\mathbf{p}$  se representa en el plano de Hough como una polilínea  $\sigma$  de dos segmentos. Esta transformación se puede visualizar en la figura 2.16(b) y está definida como

$$\sigma = S_{\pm 1} \mathcal{H}[\mathbf{p}], \quad (2.82)$$

donde

$$S_{\pm 1} = \begin{bmatrix} \pm 1 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}. \quad (2.83)$$

Considere una línea  $\ell$  que contiene puntos  $\mathbf{p}_k$ ,  $k = 1, 2, \dots, n$ , como se muestra en la figura 2.16(c). Las polilíneas  $\sigma_k$  correspondientes en el plano  $uv$  intersectan en el punto  $\hat{\ell}$  dado por

$$\hat{\ell} = \mathcal{H}^{-1}[S_{\pm 1}^{-T} \ell]. \quad (2.84)$$

En consecuencia, en el plano de Hough se produce un pico de intensidad en el punto  $\hat{\ell}$  por la acumulación de las polilíneas  $\sigma_k$ , como se muestra en la figura 2.16(d). Por lo tanto, el proceso para la detección de una línea en la imagen se simplifica en el plano de Hough; solo se requiere un algoritmo simple de detección de picos.

La transformada de Hough también es útil para detectar puntos donde varias líneas intersectan. Estos puntos son conocidos como *puntos de concurrencia* como se muestra figura 2.16(e)-(f). Para ello, usamos primero la transformada de Hough para detectar líneas y, al resultado, aplicamos nuevamente la transformada de Hough para detectar los puntos de concurrencia. En este proceso, nos referimos a la *primera transformada de Hough* para detectar líneas, y *segunda transformada de Hough* para detectar puntos de concurrencia. En el apéndice C se describe de manera formal la segunda transformada de Hough.

### 2.5.3. Máscara Gaussiana

La idea principal de la máscara Gaussiana consiste en suprimir información del plano de Hough que no contiene las líneas de interés de acuerdo a estimaciones previas. Así, la búsqueda de líneas en el plano de Hough se limita al área definida por la máscara Gaussiana. La transformada de Hough acumula todos los posibles resultados de la detección de líneas. Para las líneas de carril solo se necesita dos picos de intensidad alta en el plano de Hough. Los picos detectados

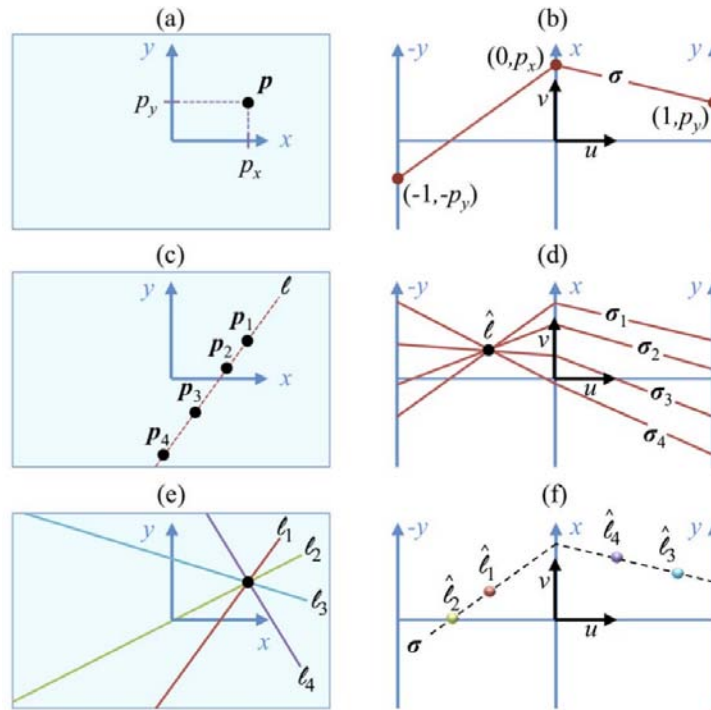


Figura 2.16: A la derecha está la representación de puntos y líneas en el plano imagen y a la izquierda su correspondiente representación en el plano de Hough. (a) Un punto  $p$  en el plano imagen se representa como (b) una polilínea  $\sigma$  en el plano de Hough. (c) Una línea  $\ell$  con puntos  $p_k$  en el plano imagen se representa como (b) un punto  $\hat{\ell}$  en el plano de Hough. (e) Un conjunto de líneas concurrentes  $\ell_k$  que intersectan en un punto  $p$  se representan como (f) un conjunto de puntos  $\hat{\ell}$  que son “colineales” a lo largo de la polilínea  $\sigma$ .

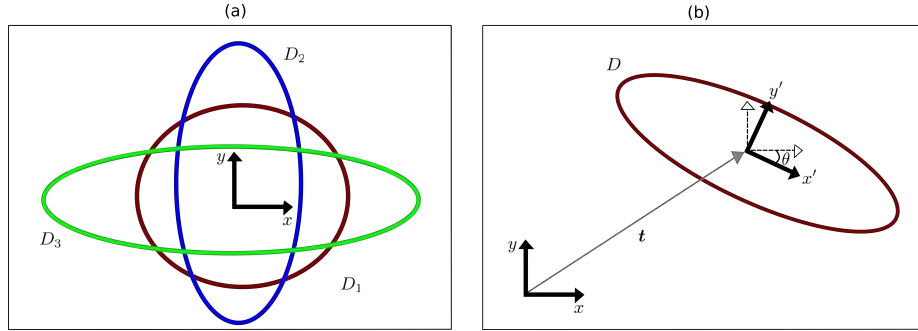


Figura 2.17: (a)  $D_1, D_2$  y  $D_3$  son máscaras Gaussianas centrados en el origen. (b) Máscara Gaussiana  $D$  centrado en el punto  $\mathbf{t}$  con ángulo de orientación  $\phi$ .

no aseguran que sean las líneas del carril de interés a pesar que los contornos fueron reducidos por el filtro de ventana. Dentro del carril aún existe información adicional que ocasiona ruido a la detección. Por esta razón, se propone el uso de una máscara Gaussiana para limitar el área de búsqueda de líneas. Este filtro eliminará todos los puntos fuera de la región donde se espera encontrar las líneas de carril.

Podemos definir la máscara Gaussiana bidimensional como

$$D(x, y) = \exp\left(\frac{-ax^2 - by^2}{2\sigma^2}\right), \quad (2.85)$$

donde  $a$  y  $b$  son los parámetros que determinan el tamaño del filtro a lo largo del eje- $x$  y del eje- $y$ , respectivamente. El filtro definido en la ecuación (2.85) está centrado en el origen del sistema de referencias como se muestra en la figura 2.17(a). Sin embargo, las líneas detectadas pueden estar en posiciones y orientaciones diferentes. Por ello, se realiza una transformación del sistema de referencia a la posición  $\mathbf{t}$  y orientación  $\phi$  esperadas de las líneas de interés como

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R(\phi) \left( \begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{t} \right), \quad (2.86)$$

donde  $R(\phi)$  es una matriz de rotación y  $\mathbf{t}$  un vector de traslación dados como

$$R(\phi) = \begin{bmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (2.87)$$

De esta forma, la máscara Gaussiana  $D(x', y')$  estará centrado en el punto  $(t_x, t_y)$  con una orientación dada por  $\phi$  como se muestra en la figura 2.17(b). Como primera instancia, se utiliza el ángulo y la posición dadas por las líneas de referencia. Después, se toman las líneas detectadas en imágenes previas.



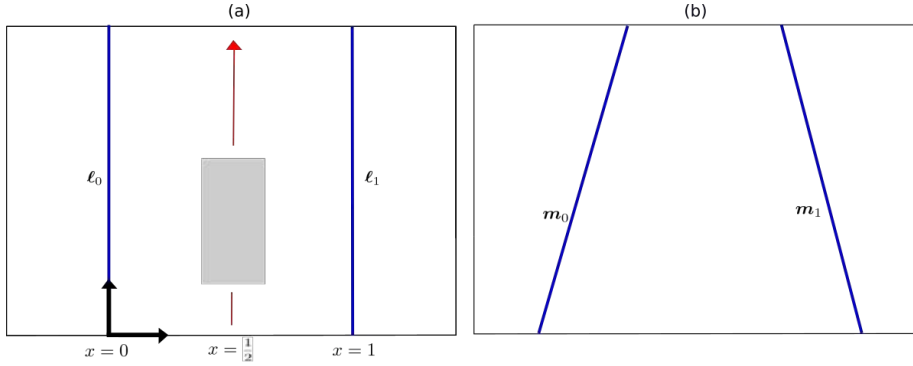


Figura 2.18: (a) Vista del plano de referencia cuando el vehículo se encuentra centrado y alineado. (b) Imagen observada por la cámara.

## 2.6. Estimación de posición y orientación

En esta tesis usamos las líneas de carril detectadas para estimar la posición y orientación del vehículo. Para esto, partimos de las definiciones de líneas de carril descritas en la Ec. (2.76) de la subsección 2.4.3. Las líneas  $\ell_0$ , y  $\ell_1$  representan el caso particular cuando el vehículo se encuentra centrado y alineado ( $x = 0.5$  y  $\phi = 0$ ), como se muestra en la figura 2.18(a). En el caso general, donde el vehículo se encuentra en cualquier otra posición y orientación  $(x, \phi)$ , las líneas de carril estarán dadas por

$$\tilde{\ell}_0 = \lambda_0 \begin{bmatrix} \cos \phi \\ \sin \phi \\ -x \end{bmatrix}, \quad \tilde{\ell}_1 = \lambda_1 \begin{bmatrix} \cos \phi \\ \sin \phi \\ -1 - x \end{bmatrix}, \quad (2.88)$$

donde  $\lambda_0$  y  $\lambda_1$  son dos escalares cualquiera diferentes de cero.

En un proceso previo de calibración, determinamos la geometría entre el piso y el plano imagen calculando la homografía  $G$  usando alguno de los métodos descritos en la sección 2.1. Una de las formas más directas es usando el cuadrilátero resultante de algún rectángulo en la escena, ver subsección 2.1.4. Así, la matriz homografía  $G$  estará disponible.

Del método propuesto para detección de líneas de carril, obtenemos como resultado los vectores  $m_0$  y  $m_1$  que representan las líneas de carril detectadas. Podemos emplear la matriz homografía  $G$  disponible para relacionar las líneas de carril en el plano del piso con las líneas detectadas en el plano imagen como

$$\begin{aligned} \tilde{\ell}_0 &= G^T \tilde{m}_0 = \begin{bmatrix} \tilde{\ell}_{0,x} \\ \tilde{\ell}_{0,y} \\ \tilde{\ell}_{0,z} \end{bmatrix}, \\ \tilde{\ell}_1 &= G^T \tilde{m}_1 = \begin{bmatrix} \tilde{\ell}_{1,x} \\ \tilde{\ell}_{1,y} \\ \tilde{\ell}_{1,z} \end{bmatrix}. \end{aligned} \quad (2.89)$$

Usando la Ec. (2.88) en (2.89), se obtienen las igualdades

$$\begin{aligned}
 \tilde{\ell}_{0,x} &= \lambda_0 \cos \phi, \\
 \tilde{\ell}_{0,y} &= \lambda_0 \sin \phi, \\
 \tilde{\ell}_{0,z} &= -\lambda_0 x, \\
 \tilde{\ell}_{1,x} &= \lambda_1 \cos \phi, \\
 \tilde{\ell}_{1,y} &= \lambda_1 \sin \phi, \\
 \tilde{\ell}_{1,z} &= -\lambda_1 - \lambda_1 x.
 \end{aligned} \tag{2.90}$$

De esta forma, podemos calcular la posición  $x$  y orientación  $\phi$  usando las componentes de  $\tilde{\ell}_0$  y  $\tilde{\ell}_1$  como

$$\begin{aligned}
 \tan \phi &= \frac{1}{2} \left( \frac{\tilde{\ell}_{0,y}}{\tilde{\ell}_{0,x}} + \frac{\tilde{\ell}_{1,y}}{\tilde{\ell}_{2,x}} \right), \\
 x &= -\frac{1}{2} \left( \frac{1}{\lambda_0} \tilde{\ell}_{0,z} + \frac{\tilde{\ell}_{1,z} + \lambda_1}{\lambda_1} \right),
 \end{aligned} \tag{2.91}$$

donde  $\lambda_0$  y  $\lambda_1$  se calculan usando

$$\begin{aligned}
 \lambda_0^2 &= \tilde{\ell}_{0,x}^2 + \tilde{\ell}_{0,y}^2, \\
 \lambda_1^2 &= \tilde{\ell}_{1,x}^2 + \tilde{\ell}_{1,y}^2.
 \end{aligned} \tag{2.92}$$

La navegación asistida utiliza la disponibilidad de la posición y orientación del vehículo para retroalimentar el sistema. Para una buena navegación, es necesario minimizar los errores de mediciones calculados. Para esto, se implementaron mejoras a la detección de líneas de carril mediante el uso de un filtro de ventana y máscara Gaussiana. Sin embargo, existen casos extremos donde las líneas de carril tiene poca o nada de visibilidad en la escena observada. Las líneas de carril se pierden y cuando se puedan volver a visualizar, la posición y orientación cambiaran drásticamente ocasionando la inestabilidad del sistema. Algunos factores que atribuyen a esta causa son la oclusión por objetos, degradación de piso, iluminación y discontinuidad de líneas, entre otras. Por esta razón, se propone implementar un algoritmo de predicción de líneas de carril para evitar los cambios drásticos en líneas de carril. A continuación describiremos formalmente este proceso de predicción usando las posiciones y orientaciones calculadas previamente para suavizar la detección de líneas de carril.

### 2.6.1. Predicción de líneas de carril

El método de predicción de líneas de carril se basa en proporcionar una nueva medición dada por las entradas de estimaciones previas. Para esto, utilizamos el método de mínimos cuadrados recursivos con factor de olvido para predecir los parámetros del sistema. El factor de olvido asigna la importancia que tiene las estimaciones previas, entre más antiguas sean, menor importancia van a ser.

El método de mínimos cuadrados recursivos con factor de olvido es descrito de manera formal en el apéndice A.

Utilizando la notación  $[\cdot]_k$  como la estimación actual,  $[\cdot]_{k-1}$  como estimaciones previas; podemos escribir la predicción de la posición del vehículo como

$$x_k = x_{k-1} + \left[ (P_{k-1} \boldsymbol{\theta}_k^T) / (\lambda + \boldsymbol{\theta}_k P_{k-1} \boldsymbol{\theta}_k^T) \right] \xi_k, \quad (2.93)$$

donde  $\boldsymbol{\theta}$  son las variables de regresión,  $\lambda$  es el factor de olvido,  $\xi$  es el error de la función, y

$$\begin{aligned} P_{k-1} &= \Theta_{k-1}^T \Lambda \Theta_{k-1}, \\ \Theta_{k-1} &= [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, \boldsymbol{\theta}_3^T, \dots, \boldsymbol{\theta}_{k-1}^T], \\ \Lambda &= \text{diag}(\lambda^{k-1}, \lambda^{k-2}, \lambda^{k-3}, \dots, \lambda^0). \end{aligned} \quad (2.94)$$

De la misma manera, la orientación del vehículo se puede predecir como

$$\phi_k = \phi_{k-1} + \left[ (P_{k-1} \boldsymbol{\theta}_k^T) / (\lambda + \boldsymbol{\theta}_k P_{k-1} \boldsymbol{\theta}_k^T) \right] \xi_k, \quad (2.95)$$

En el siguiente capítulo se evaluará el desempeño del método propuesto procesando secuencias de video del mundo real. Se realizará la descripción de las plataformas utilizadas para las pruebas experimentales. Se presentarán los detalles importantes de la implementación del método propuesto y la calibración de la cámara de un robot móvil. Finalmente, los resultados obtenidos serán analizados y discutidos.



# Capítulo 3

## Desarrollo experimental y resultados

### 3.1. Plataforma experimental

En este trabajo de tesis, se utilizan dos plataformas para la implementación del algoritmo de navegación visual de un robot. En la figura 3.1 se muestra el robot móvil terrestre *OSOYOO* utilizado para realizar la navegación en una pista experimental. El sistema está equipado con una tarjeta *Raspberry Pi 3 B+* para el control y la comunicación del vehículo. Esta tarjeta es una pequeña computadora de gran utilidad para controlar la velocidad de los motores, adquisición de imágenes por la cámara y conexión a la red para transferir los datos hacia otra plataforma. Para realizar el procesamiento de datos, se usa una computadora de escritorio equipada con el procesador gráfico NVIDIA GeForce GTX 760 para ejecutar tareas de cómputo paralelo, como se muestra en la figura 3.2. Esta computadora se utiliza para ejecutar los algoritmos de procesamiento propuestos.

En la figura 3.3, se muestra la pista experimental para la navegación del robot móvil terrestre. Las grabaciones son capturadas para el procesamiento del algoritmo desarrollado y posteriormente, son evaluado de acuerdo al desempeño y robustez de la estimación de posición y orientación del vehículo.

### 3.2. Implementación en cómputo paralelo

El paradigma de los algoritmos paralelos es la ejecución simultánea de tareas. Esto genera ventajas como la ejecución de tareas en múltiples procesadores, la capacidad de resolver problemas de alta complejidad y mejoras en tiempos de respuesta, entre otros. Para esto, se tiene que realizar un nuevo diseño paralelo para independizar las tareas. En el algoritmo de la detección de líneas de carril

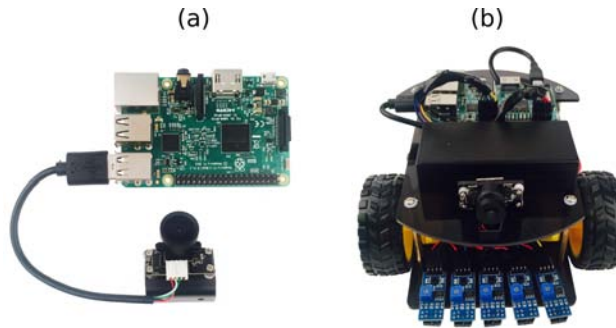


Figura 3.1: Plataforma para navegación y adquisición de imágenes. (a) Cámara convencional conectada a la tarjeta de desarrollo *Raspberry Pi 3 B+*. (b) Robot móvil ensamblado.

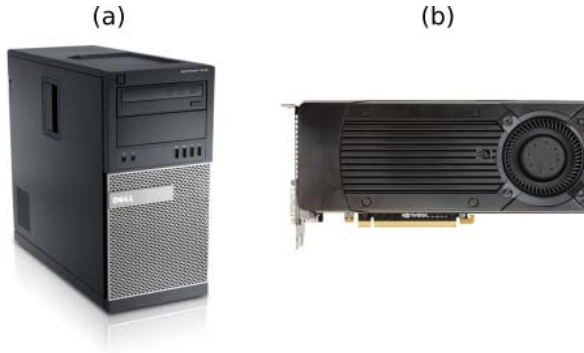


Figura 3.2: Plataforma para procesamiento de imágenes. (a) Computadora de escritorio. (b) Tarjeta de video NVIDIA Geforce GTX 760.

se paralizaron varias tareas, específicamente, la detección de contornos y la transformada de Hough como se muestra en la figura 3.4. Los detalles de cada tarea se describen a continuación.

Primeramente, se recibe la imagen capturada en la unidad central de procesamiento (CPU por sus siglas en inglés *Central Processing Unit*), y se transfiere a la memoria de unidad de procesamiento gráfico. Después, se realiza la detección de contornos para cada punto de la imagen en los procesadores del GPU. A continuación, cada punto de los contornos son procesados para encontrar su respectiva polilínea y se acumulan en cada coordenada del plano de Hough. Después, se retornan los puntos acumulados hacia la memoria del CPU. Finalmente, se detectan las dos líneas representadas por los picos de intensidad alta en el plano de Hough.

En la figura 3.5(a)-(d) se observa una secuencia de imágenes de un patrón cuadro-radial en diferentes perspectivas. Las imágenes son procesadas utilizando



Figura 3.3: Pista de prueba para navegación de un robot móvil terrestre.

el algoritmo paralelo para la detección de líneas. El resultado de las detecciones se puede visualizar en la figura 3.5(e)-(h). Las líneas detectadas son utilizadas para encontrar la concurrencia entre ellas usando la segunda transformada de Hough (consultar Apéndice C).

La implementación en cómputo paralelo (CUDA) fue favorable comparando el tiempo de respuesta con el obtenido en la implementación del algoritmo secuencial en el software MATLAB. Usando la tarjeta de procesador gráfico NVIDIA Geforce GTX 760, se obtuvieron los resultados presentados en la tabla comparativa 3.1. La implementación de cómputo paralelo tiene una mejora de 29X ( $872.12 \text{ ms} / 29.77 \text{ ms} = 29.29$ ) en el rendimiento del sistema. Debido a su tiempo de ejecución se obtiene una mejora en la estabilidad del sistema.

### 3.3. Calibración de cámara

Como hemos visto anteriormente, la estimación de parámetros intrínsecos y extrínsecos de una cámara requieren múltiples homografías. Adicionalmente,

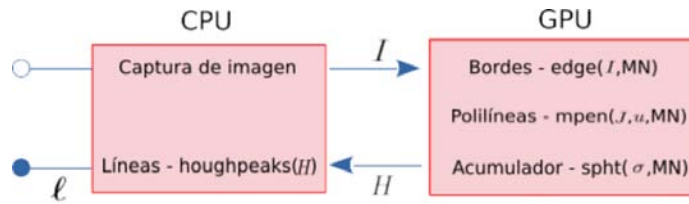


Figura 3.4: Arquitectura del algoritmo de la transformada de Hough en cómputo paralelo.

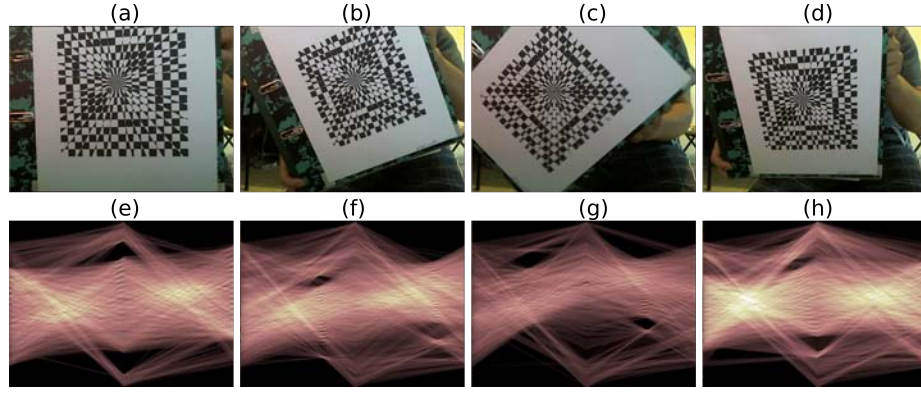


Figura 3.5: (a)-(d) Secuencia de imágenes de un patrón cuadro-radial en diferentes perspectivas. (e)-(h) Representación de las imágenes (a)-(d) en el plano de Hough.

Cuadro 3.1: Comparación en tiempos de ejecución para cada implementación.

	Detección de contornos	Transformada de Hough	Cuadros por segundo (fps)
MATLAB	12.85 ms	853.27 ms	1.15 fps
CUDA	0.23 ms	29.54 ms	33.59 fps

para la estimación de homografías se requieren tres puntos en el plano imagen:  $\mu_a$ , el punto al infinito en dirección  $x$ ,  $\mu_b$ , el punto al infinito en dirección  $y$ , y  $\mu_c$ , la imagen del origen. Por conveniencia, se utiliza la transformada Hough para detectar los puntos del plano imagen requerido. A continuación, se describirán los pasos que se realizaron para la calibración de la cámara del robot móvil.

Primeramente, se capturaron 32 imágenes en diferentes puntos de observación al patrón de calibración como se muestra en la figura 3.6. Después, se utilizó la transformada de Hough para detectar los puntos del plano imagen ( $\mu$ ) de cada imagen. Como siguiente paso, los puntos  $\mu_a$ ,  $\mu_b$  y  $\mu_c$  fueron procesados para calcular la homografía cada imagen. Finalmente, todas las homografías calculadas fueron utilizadas para la estimación de los parámetros intrínsecos y extrínsecos de la cámara. Los parámetros intrínsecos y extrínsecos resultantes



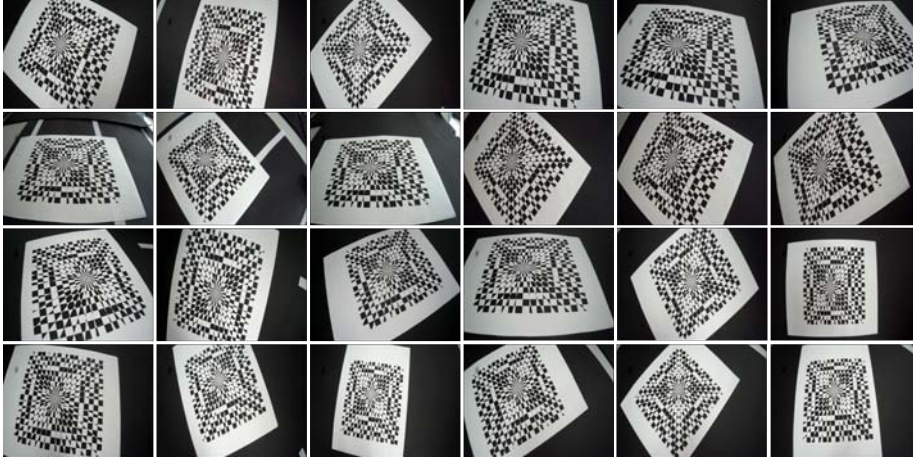


Figura 3.6: Patrones de calibración capturados por la cámara del robot móvil terrestre.

son

$$\begin{aligned}
 K &= \begin{bmatrix} 2,0184 & 0 & 0,0359 \\ 0 & 2,6129 & -0,4382 \\ 0 & 0 & 1 \end{bmatrix}, \\
 R &= \begin{bmatrix} 0,9997 & 0,0215 & -0,0109 \\ 0,0209 & -0,5473 & 0,8367 \\ 0,0121 & -0,8367 & -0,5476 \end{bmatrix}, \text{ y} \\
 t &= \begin{bmatrix} -22,1248 \\ -255,3641 \\ 158,7193 \end{bmatrix}.
 \end{aligned} \tag{3.1}$$

Los parámetros obtenidos de la cámara son utilizados para formar la homografía del sistema.

### 3.4. Corrección de imágenes distorsionadas

En esta sección se va estimar la distorsión radial de la lente que tiene la cámara del robot móvil. Para esto, se analiza la diferencia entre un plano de referencia observado y el plano imagen. Por esto, capturamos los patrones de franjas desplegados en un monitor de computadora (plano de referencia) como se muestra en la figura 3.7. Se utilizó una rejilla con una única franja (frecuencia espacial  $f = 1$ ) con 16 corrimientos de fase. No fue necesario usar rejillas de mayor frecuencia debido a que el monitor usado para desplegar las rejillas no presenta distorsión gamma. Además, al usar la frecuencia  $f = 1$ , se evita el proceso de desdoblamiento de fase [43].

Se capturaron dos conjuntos de 16 patrones de franjas (16 patrones de franjas por cada eje coordenado,  $x$ ,  $y$ ). Los 32 patrones de franjas resultantes son procesados usando el método de corrimiento de fase [61]. En la figura 3.8, se presenta la fase extraída, que corresponde a las coordenadas de los puntos del plano de referencia (sin distorsión). Los puntos del plano de referencia se comparan con los puntos del plano imagen (afectados por distorsión radial de la lente).

Las correspondencias de puntos resultantes (puntos con distorsión y sin distorsión) son procesadas para estimar los parámetros de distorsión aplicando el método analizado previamente en el capítulo 2. El resultado de la estimación de parámetros es

$$\mathbf{d} = [0,9567 \quad -0,0320 \quad 0,1911 \quad -0,0012 \quad 0,0098]^T. \quad (3.2)$$

Usando los parámetros de distorsión estimados, se puede realizar corrección de distorsión radial de la imagen de entrada. En la figura 3.9(a), se muestra una imagen de las líneas de carril capturada por la cámara del robot. Empleando los parámetros estimados, se corrige la distorsión radial de la imagen empleando la Ec. (2.56). En la figura 3.9(b), se muestra el resultado de la corrección de distorsión. Este proceso de corrección de distorsión radial mejora significativamente el proceso posterior de detección de líneas como se verá más adelante.

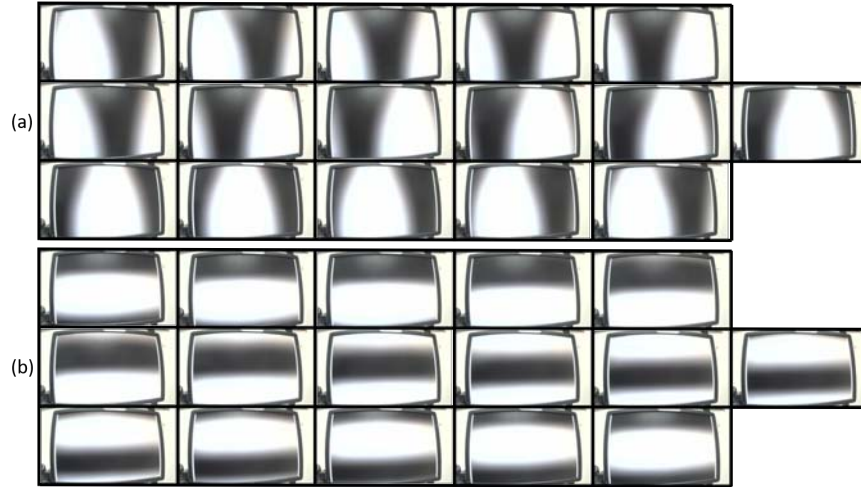


Figura 3.7: Patrones de franjas capturadas por la cámara del robot móvil terrestre. (a) Franja en dirección  $x$ . (b) Franja en dirección  $y$ .

### 3.5. Detección de líneas de carril

En esta sección se van a aplicar los métodos analizados para realizar detección de líneas de carril. Se evalúa el funcionamiento del algoritmo propuesto

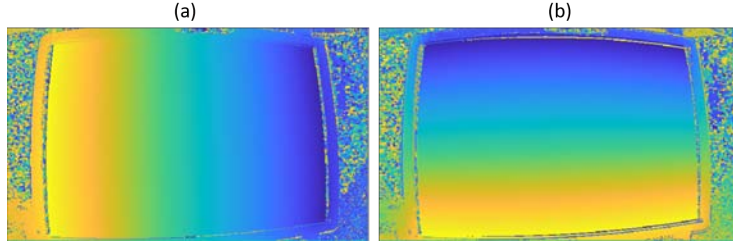


Figura 3.8: Extracción de fase con el método de corrimiento de fase. (a) Fase en dirección  $x$ . (b) Fase en dirección  $y$ .

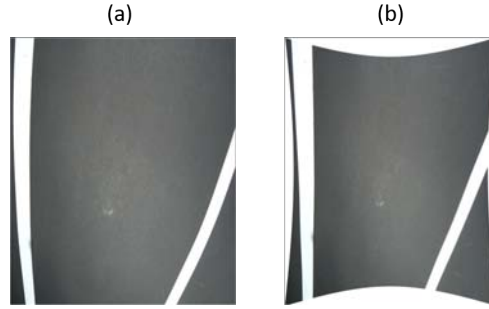


Figura 3.9: Corrección de distorsión utilizando los parámetros estimados. (a) Imagen con distorsión. (b) Imagen sin distorsión.

para la detección de líneas de carril en diferentes escenas como se muestra en la figura 3.10 (las secuencias de video fueron tomadas de una base de datos pública disponible en <https://bdd-data.berkeley.edu/>).

En el primer paso, se aplica el método de Canny para detectar los contornos de la escena observada. Para la figura 3.10(a), el resultado de la detección de contornos se muestra en la figura 3.11(a). En el segundo paso se obtiene la transformada de Hough para detección de líneas. La figura 3.11(b) muestra la transformada de Hough correspondiente a los contornos detectados mostrados en la figura 3.11(a). En el plano de Hough, las líneas de carril de interés deben corresponder a los dos picos de mayor intensidad. Para fines ilustrativos, las líneas detectadas son desplegadas en la imagen de entrada como se muestra en la figura 3.11(c).

De la figura 3.11(c) se puede observar que la detección de líneas de carril presenta una desviación considerable. Las líneas detectadas están desalineadas respecto a las líneas de carril reales debido a errores en la detección. Para mejorar la exactitud en la detección de líneas de carril, se usa un filtro de ventana para discriminar contornos ajenos a la pista, y una máscara Gaussiana para mejorar el espacio de búsqueda en el plano de Hough.



Figura 3.10: Capturas de tres secuencias de video para la evaluación del método propuesto. (a) Imagen en donde la mayor parte de la escena no corresponde a la pista. (b) Imagen en donde las líneas de carril son curvas. (c) Imagen en escenas con líneas de carril discontinuas y múltiples carriles.

### 3.5.1. Filtro de ventana

Además de los contornos que corresponden a las líneas de carril, en la escena existen contornos que corresponden a otros objetos en la escena. Por ejemplo, árboles, arbustos, postes de líneas eléctricas, otros vehículos, señales de tránsito, etc. Los contornos adicionales en la escena introducen polilíneas adicionales en el plano de Hough. Las polilíneas adicionales son ruido que incrementa la dificultad para hallar los picos de intensidad máxima que corresponden a las líneas de carril de interés. Para reducir esta fuente de ruido, se empleó un filtro de ventana con  $\epsilon =$  para suprimir todos los contornos no útiles para la detección de las líneas de interés, como se muestra en la figura 3.11(d).

Se debe mencionar que al eliminar los contornos que no corresponden a la pista también se acelera el proceso de obtención de la transformada de Hough. Esto se debe a que al reducir el número de contornos, también se reduce el número de polilíneas que se deben acumular como se muestra en la figura 3.11(e). En la figura 3.11(f) se muestra que la detección de las líneas de carril ha mejorado, pero aún existe un error que se debe reducir.

### 3.5.2. Máscara Gaussiana

En la figura 3.11(e) se puede observar que hay más de dos picos de intensidad máxima. Esto se debe primero a que cada línea de carril genera dos líneas asociadas a sus contornos. Además, en la escena no solo existen dos líneas de carril, sino tres. Por esta razón, en el plano de Hough aparecerán más de dos picos de intensidad máxima en lugar de solo dos picos como se esperaría en el caso ideal.

Para evitar este problema, se utilizaron las máscaras Gaussianas para limitar el área de búsqueda en el plano de Hough a aquellas regiones donde se espera encontrar los pico de intensidad máxima. Con una máscara, se espera detectar el pico de intensidad que representa una línea de carril de interés. Por motivos ilustrativos, las dos máscaras Gaussianas son presentadas al mismo tiempo en

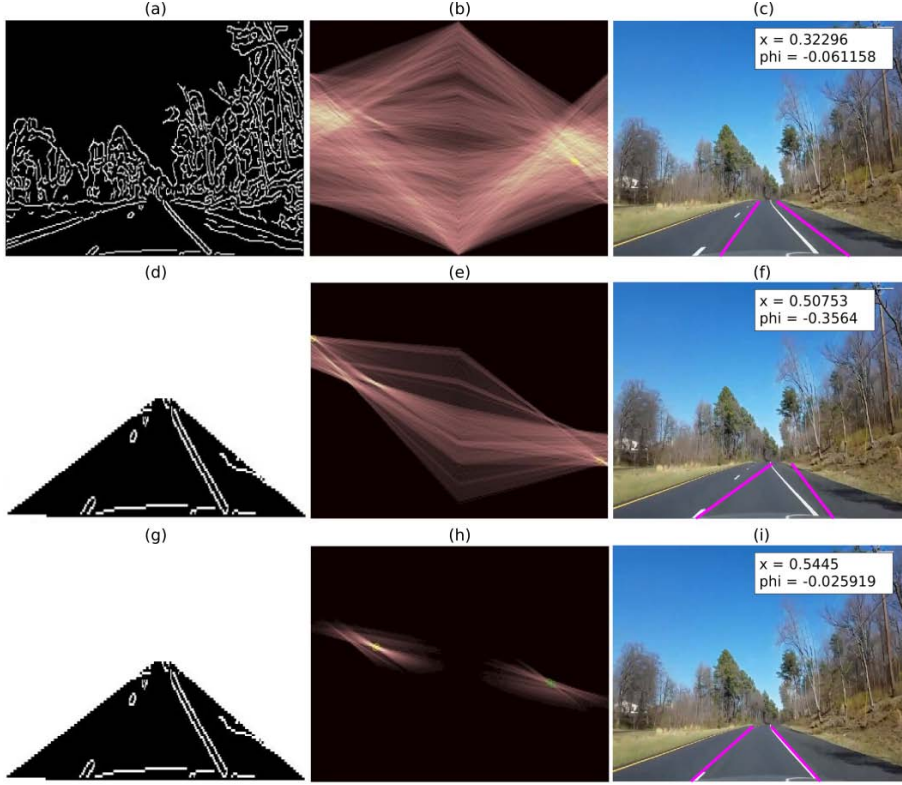


Figura 3.11: (a)-(c) Detección de líneas de carril sin el uso filtros. (d)-(f) Detección de líneas usando el filtro de ventana para discriminación de contornos. (g)-(i) Detección de líneas utilizando el filtro de ventana y la máscara Gaussiana.

el plano de Hough, como se muestra en la figura 3.11(h). De esta forma, la exactitud en la detección de las líneas de carril mejora significativamente como se observa en la figura 3.11(i).

### 3.5.3. Líneas de carril curvas

La robustez del algoritmo propuesto se verificó procesando una secuencia de video donde aparecen líneas de carril curvas, ver figura 3.10(b). En este experimento se encontró que el método propuesto funciona correctamente detectando curvas de hasta  $\phi = 10$  grados respecto a la dirección de desplazamiento. En la figura 3.12 se muestra el resultado del procesamiento de la detección de líneas de carril en la escena observada.

De forma similar a la prueba anterior, se observa que la exactitud en la detección de líneas de carril mejora con la aplicación del filtro de ventana y la máscara Gaussiana. Observe que la cámara fue colocada de tal forma que en la



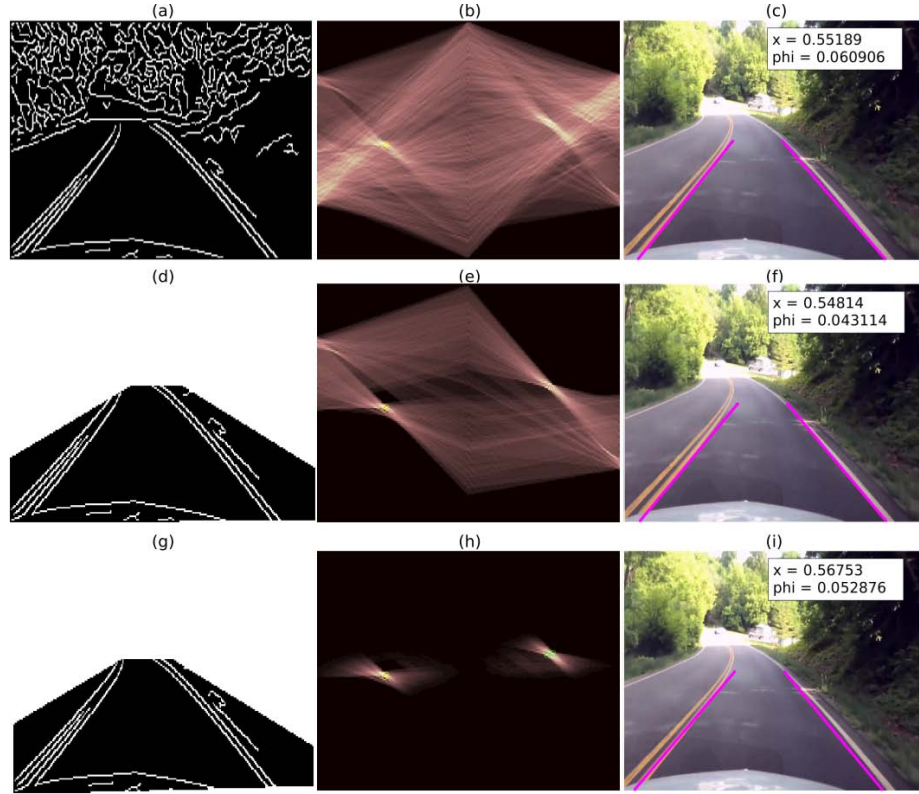


Figura 3.12: Evaluación del funcionamiento del algoritmo propuesto en líneas curvas. (a)-(c) Detección de líneas de carril sin el uso filtros. (d)-(f) Detección de líneas usando el filtro de ventana para discriminación de contornos. (g)-(i) Detección de líneas utilizando el filtro de ventana y la máscara Gaussiana.

imagen la pista abarca la mayor parte de la imagen. Esto es conveniente debido a que la información capturada por la imagen corresponden en mayor medida a las líneas de carril de interés. Por lo tanto, la acumulación de polilíneas asociadas a los contornos detectados forman picos de intensidad bien definidos en el plano de Hough, como se muestra en la segunda columna de la figura 3.12.

Con esta prueba, se confirma que es recomendable, tanto el uso de los filtros propuestos, como una buena colocación de la cámara para capturar en mayor medida la información de la pista reduciendo al mismo tiempo los contornos innecesarios para la detección de líneas de interés.

También se puede mencionar que el algoritmo propuesto es capaz de detectar líneas rectas con bajo error cuando las líneas de carril son curvas. Para esto, se reduce el campo de visión de la cámara para limitar la longitud observada del carril en la escena. En otras palabras, las líneas de carril curvas se pueden aproximar a líneas rectas a corta distancia desde la cámara. Así, ajustando

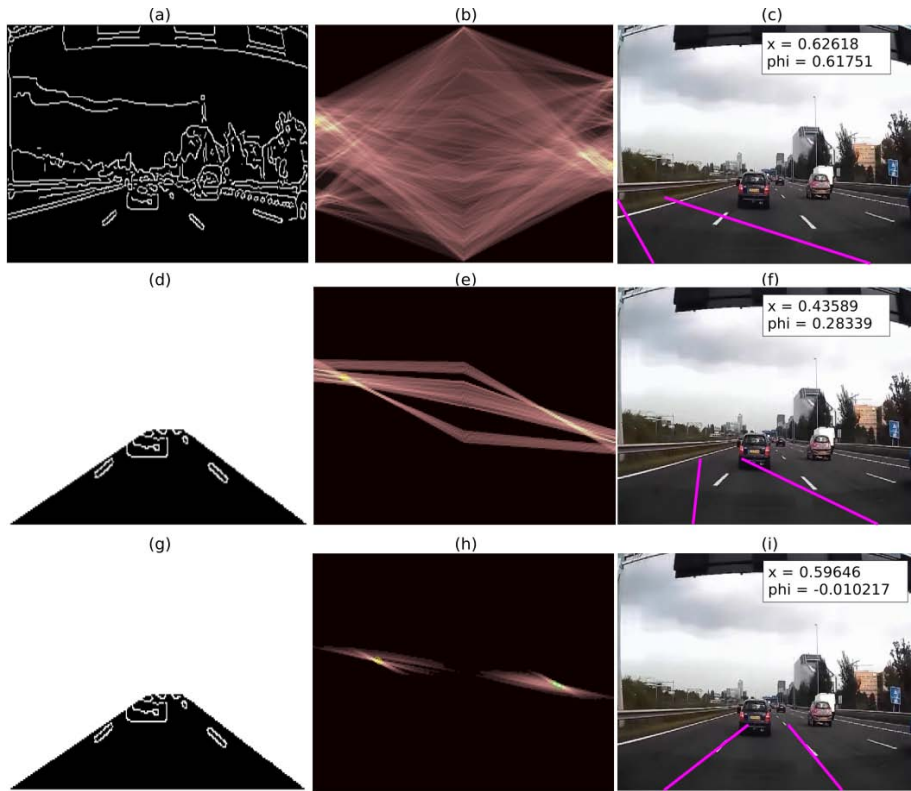


Figura 3.13: Evaluación del funcionamiento del algoritmo propuesto en escenas de la vida real. (a)-(c) Detección de líneas de carril sin el uso filtros. (d)-(f) Detección de líneas usando el filtro de ventana para discriminación de contornos. (g)-(i) Detección de líneas utilizando el filtro de ventana y la máscara Gaussiana.

la extensión del campo observado en la escena se puede obtener una buena detección de las líneas de carril aún cuando estas presenten curvatura.

#### 3.5.4. Líneas discontinuas y múltiples carriles

Se realizó otra prueba en donde el método propuesto se evaluó procesando una secuencia de video capturada en una zona urbana típica. En este nuevo escenario, aparecen otras dificultades tales como líneas discontinuas, múltiples carriles alrededor de la pista observada, oclusiones por otros vehículos, líneas de contención, señales de tránsito, y fachadas de edificios, entre otros, ver figura 3.10(c).

En esta prueba, la transformada de Hough detecta una mayor cantidad de líneas que están presentes en la escena, mientras que las líneas de carril de interés se vuelven menos visibles debido a las discontinuidades de los trazos y

oclusiones. Aún en este escenario complejo, el método de detección de líneas de carril propuesto logra buenos resultados y destaca la mejora que proporciona la aplicación del filtro de ventana y Gaussiano como se muestra en la figura 3.13.

En la implementación actual, el método propuesto detecta un carril. Sin embargo, cuando se realiza un cambio de carril, el método propuesto mantendrá la detección en el carril inicial y no en el carril actual. Este caso se puede detectar usando la variable  $x$  que indica la posición relativa respecto al carril. En el plano de referencia, las líneas de carril son  $x = 0$  y  $x = 1$ . Por lo tanto, la posición del vehículo será  $x = 1/2$  cuando el vehículo está centrado en el carril. De esta forma si la posición  $x$  es mayor que  $+1$ , indicará que el vehículo ha cambiado al carril derecho. De forma similar, si la posición  $x$  es menor que cero, entonces el vehículo se encontrará en el carril izquierdo. Se puede desarrollar una rutina que detecte estas transiciones y así restablecer el enfoque en el nuevo carril del vehículo.

### 3.6. Estimación de posición y orientación

Se utilizaron 100 secuencias de video que presentan diferentes escenarios de manejo, horas del día, y cambios climáticos (las secuencias de video fueron tomadas de una base de datos pública disponible en <https://bdd-data.berkeley.edu/>). Cada secuencia de video es procesada con el algoritmo desarrollado. Los resultados son capturados y presentados en las figuras 3.14-3.17. De los resultados obtenidos, se encontraron escenarios donde la detección de líneas es más complicada, tales como, líneas de carril con oclusión, líneas curvas, cambio de carril, carril sin líneas y oclusión por vehículos. Algunos de estos problemas fueron mencionados en las evaluaciones anteriores.

El problema de oclusión de líneas se encuentra presente en múltiples secuencias de video. Las líneas de carril son fuentes importantes para el método propuesto de estimación de posición y orientación del vehículo. La pérdida de esta fuente de información afecta los resultados de las mediciones debido a que la transformada de Hough no puede detectar líneas ocluidas. En consecuencia, la posición y orientación determinada son erróneas. Para evitar este problema, se propuso un método de estimación usando mínimos cuadrados recursivos con factor de olvido que aprovecha el resultado de estimaciones previas para calcular una nueva posición y orientación actual.

En la figura 3.18 se presenta una secuencia de video donde las líneas de carril se pierden de vista por oclusión de un vehículo. Sin embargo, el algoritmo propuesto es capaz de mantener una estimación de línea estable aprovechando las estimaciones realizadas antes de la oclusión. Para esto, las líneas son determinadas por la posición y orientación calculada por el método de mínimos cuadrados recursivos con factor de olvido analizado previamente en el capítulo 2.5. El resultado de la predicción de líneas de carril en esta prueba se muestra en la figura 3.18(a)-(e). Estos resultados demuestran la robustez del algoritmo propuesto aún cuando las líneas de carril no son visibles temporalmente en la pista, como se muestra en la figura 3.18(f)-(j).



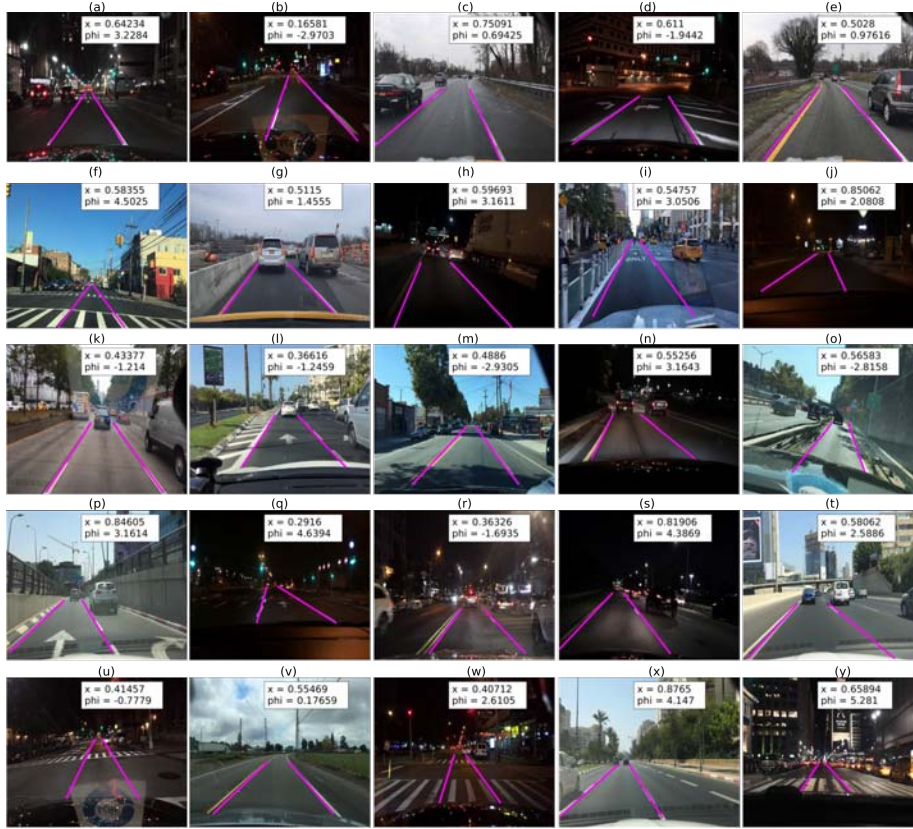


Figura 3.14: Primer conjunto de secuencias de vídeo. (a)-(y) Capturas de secuencias de vídeo para evaluar el desempeño del método propuesto en diferentes escenarios.

### 3.7. Navegación de un robot móvil terrestre

Las evaluaciones previas del algoritmo propuesto fueron realizadas usando secuencias de vídeo obtenidas de una base de datos que colecta rutinas de navegación de automóviles. De estas evaluaciones se encontró que la estimación de posición y orientación del vehículo usando es método propuesto es eficiente y robusta. Para el siguiente experimento, se evaluará una secuencia de vídeo de navegación de un robot móvil terrestre en una pista experimental.

El procesamiento de la secuencia de vídeo es similar a las evaluaciones anteriores. Sin embargo, la cámara utilizada en el robot móvil presenta distorsión radial considerable, como se muestra en la figura 3.19(a)-(e). La distorsión radial ocasiona una mala visualización y alto error en la detección de líneas de carril debido a que las líneas rectas se convierten en líneas curvas. Este problema se resolvió usando el método de corrección de distorsión radial descrito



Figura 3.15: Segundo conjunto de secuencias de vídeo. (a)-(y) Capturas de secuencias de video para evaluar el desempeño del método propuesto en diferentes escenarios.

previamente en esta tesis. Los parámetros de distorsión fueron estimados previamente en la sección 3.4. La imagen corregida proporciona nuevos puntos en las coordenadas del plano y son utilizados para detectar los contornos de la imagen, como se observa en la figura 3.19(f)-(j). Después, se detectan las líneas de carril con la transformada de Hough y se limitan la búsqueda de picos de intensidad máxima con las máscaras Gaussianas, como se ilustra en la figura 3.19(k)-(o). Finalmente, se estiman la posición y orientación usando las líneas de carril disponibles.

En la figura 3.19(p)-(t), se presentan los resultados obtenidos y se mostró la robustez del algoritmo propuesto aún usando cámaras con distorsión radial. La estimación de posición y orientación de un robot móvil es eficiente y las estimaciones realizadas pueden emplearse para realizar retroalimentación para el controlador del robot móvil.



Figura 3.16: Tercer conjunto de secuencias de vídeo. (a)-(y) Capturas de secuencias de video para evaluar el desempeño del método propuesto en diferentes escenarios.

### 3.8. Discusiones

Los resultados experimentales obtenidos han demostrado la eficiencia y la robustez del método propuesto en diferentes escenarios del mundo real. Se mostró la utilidad del modelo de cámara pinhole aún cuando se utilizan cámaras que presentan distorsión radial. Esto es posible gracias a la estimación de parámetros de distorsión radial y su uso para corrección de distorsión en las imágenes procesadas.

La primera etapa del método propuesto consiste en detectar los contornos de la escena. Se empleó el método de Canny para detección de contornos debido a su eficiencia computacional. Se usó un filtro Gaussiano con una desviación estándar de  $\sqrt{2}$  para minimizar el ruido de las imágenes de entrada. Los contornos son suprimidos mediante un umbral para las intensidades que sea mayor a 0,35 y un umbral para las intensidades que sea menor a 0,65. Posteriormente, se usó



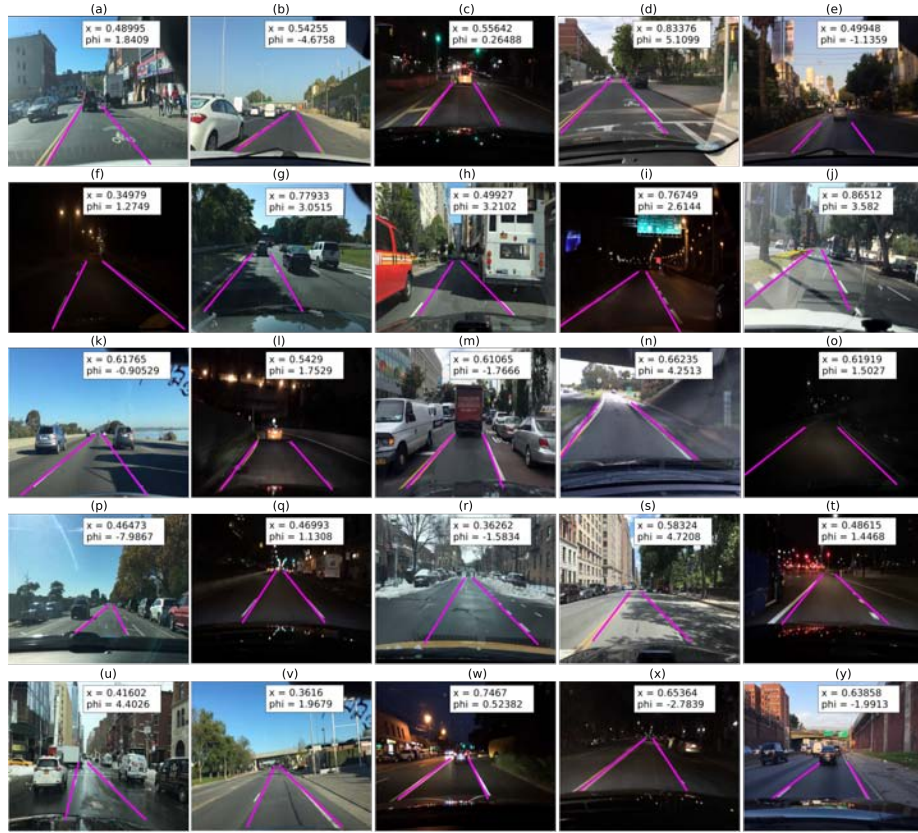


Figura 3.17: Cuarto conjunto de secuencias de vídeo. (a)-(y) Capturas de secuencias de vídeo para evaluar el desempeño del método propuesto en diferentes escenarios.

un filtro de ventana para separar la información entre la pista y otros objetos de la escena.

En una segunda etapa, se aplicó la transformada de Hough en coordenadas paralelas para detección de líneas. En el plano de Hough, se usó un máscara Gaussiana para limitar la región de búsqueda de las líneas de carril esperadas en un área específica. Las regiones de interés fueron determinadas usando como referencia las detecciones previas de líneas de carril. De los resultados obtenidos, se comprobó que los filtros usados mejoraron significativamente el proceso de detección de líneas. Asimismo, se redujo el tiempo de respuesta del método propuesto al implementar los algoritmos usando cómputo paralelo.

En la última etapa, la posición y orientación del vehículo se estimó usando la matriz homografía del sistema y las líneas de carril detectadas. Finalmente, se utilizó el método de mínimos cuadrados recursivos con factor de olvido para proporcionar estimaciones cuando las líneas de carril no son visibles o varían sig-

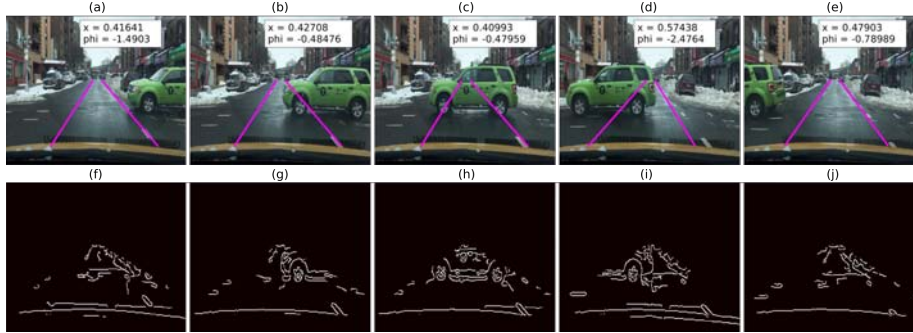


Figura 3.18: Secuencia de imágenes para la evaluación del algoritmo propuesto en líneas de carril no visibles. (a)-(e) Predicción de líneas de carril en oclusión por un vehículo, y (f)-(i) su visualización de los contornos detectados, respectivamente.

nificativamente por ruido aleatorio. Este método permite suavizar la transición entre detecciones de líneas de carril, y en general, la estabilidad del sistema.

Los resultados mostraron que es posible detectar líneas de carril curvas aún cuando el método propuesto fue diseñado asumiendo líneas de carril rectas. Sin embargo, si la curvatura de las líneas es mayor que 10 grados respecto a la dirección de desplazamiento, la precisión en la estimación de posición y orientación se reduce. En este caso, se puede incrementar la precisión reduciendo el campo de visión para limitar la longitud observada de la pista y la aproximación a líneas rectas siga siendo válida.

El método propuesto es robusto ante la presencia de brillo ambiental y saturación. Esta fuente de ruido ocasiona cambios en la intensidad de los píxeles provocando líneas adicionales en la transformada de Hough. Aún así, se logró una buena detección de líneas de carril mediante el uso de las máscaras Gaussianas propuestas.

De la navegación del robot móvil en la pista experimental se observaron los siguientes comportamientos del algoritmo desarrollado. Primero, los azulejos del piso ocasionaron líneas adicionales en la detección de contornos, y en el plano de Hough, el pico de intensidad máxima pertenecía a las líneas de los azulejos debido a su trazo recto, en contraste a la curvatura de las líneas de interés que limitan la pista. Para atenuar esta fuente de ruido, se redujo el tamaño del filtro de ventana en la detección de contornos hasta ajustar las líneas de carril de interés. Segundo, se ajustó el campo de observación en la escena para detectar las líneas curvas del carril en la pista experimental. Cuando el campo de observación es pequeño, las líneas de carril no se visualizan correctamente. Cuando el campo de observación es grande, se visualizaban líneas adicionales que no son útiles. Ambos casos provocan errores en la detección de líneas de carril. Para reducir el efecto de estas fuentes de error, se incrementó el factor de olvido del estimador de mínimos cuadrados y se ajustó el campo de observación a un punto medio entre el campo cercano y lejano de la cámara. Por último,

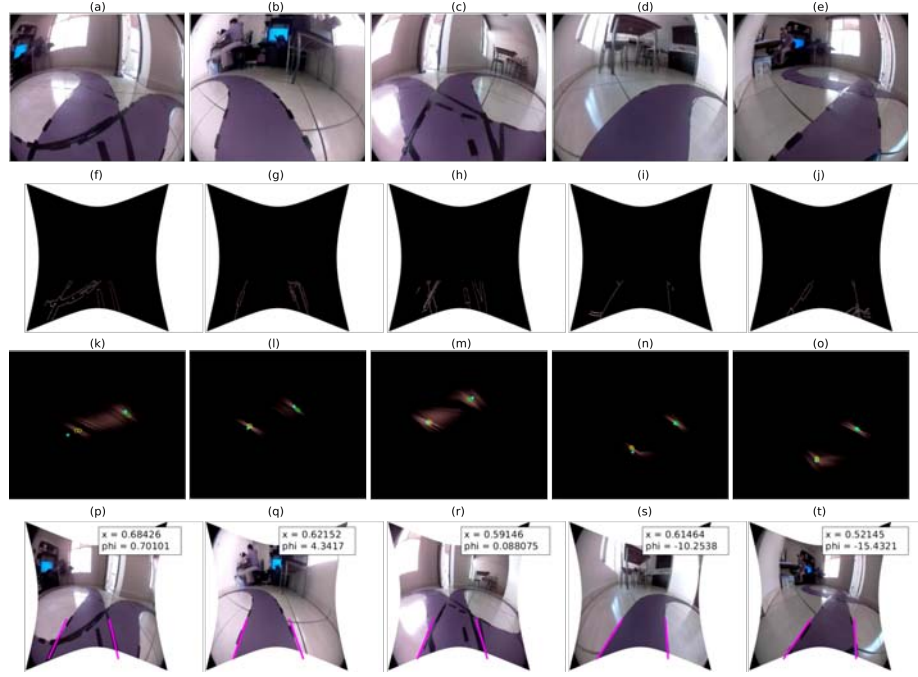


Figura 3.19: Evaluación del algoritmo propuesto en una secuencia de video de la navegación de un robot móvil terrestre. (a)-(e) Imágenes de entrada con distorsión radial. (f)-(j) Detección de contornos con su respectivo filtro de ventana para las imágenes (a)-(e) con distorsión corregida. (k)-(o) Picos de intensidad máxima limitado por la región de interés en el plano de Hough. (p)-(t) Resultados de detección de líneas de carril y estimación de posición y orientación del robot móvil terrestre.

los giros del robot móvil en las curvas no son constantes debido al diseño del carril. Las estimaciones son erróneas cuando el robot realiza rotaciones bruscas. En estos casos, el factor de olvido del predictor por mínimos cuadrados se debe reducir; sin embargo, esto hace que el algoritmo sea sensible al ruido aleatorio como se describió anteriormente.

# Capítulo 4

## Conclusiones

En este trabajo de tesis se abordó el problema de estimación de la posición y orientación de un robot móvil terrestre usando líneas de carril como fuente de información. El método propuesto se diseñó considerando el enfoque de sistemas opto-digitales para el procesamiento de imágenes. Para ello, se analizaron los modelos físicos del proceso de formación de imagen y la flexibilidad que ofrece la implementación de algoritmos en computadoras digitales para procesamiento de datos.

En los algoritmos de estimación propuestos, se consideró el modelo de cámara *pinhole* para simplificar el análisis y reducir la complejidad de los modelos matemáticos. Para hacer válidos estos algoritmos en cámaras que presentan distorsión debido a la lente, se consideró una etapa adicional de corrección de distorsión radial. Para definir la posición y orientación relativa entre un plano de referencia y la cámara, se consideró la relación entre homografías y los parámetros intrínsecos y extrínsecos de la cámara.

Para la estimación de posición y orientación usando líneas de carril como referencia, se propuso un algoritmo que consiste en dos etapas. Primero, se realiza detección de las líneas de carril, y, segundo, se determina la posición y orientación del robot móvil terrestre a partir de las líneas de carril detectadas. Se empleó el método de Canny para detección de contornos en la escena. Se implementó la transformada de Hough en coordenadas paralelas para detección de líneas de carril.

Para mejorar la exactitud en la detección de líneas de carril, se propuso el uso de un filtro de ventana para separar la información entre la pista y otros objetos de la escena que interfieren con la detección. En el plano de Hough, se usó una máscara Gaussiana para limitar la región de búsqueda de las líneas de carril al área donde se esperaba encontrarlas en base a imágenes previas. Se agregó una etapa de corrección de distorsión radial para mejorar los resultados de la detección aún usando una cámara convencional de bajo costo.

El algoritmo de navegación visual propuesto fue validado procesando secuen-

cias de vídeo tomadas del mundo real. La cámara fue calibrada para obtener tanto los parámetros intrínsecos, extrínsecos y distorsión radial. Los resultados obtenidos mostraron la eficiencia y robustez del método propuesto. El costo computacional fue reducido por la discriminación de contornos en áreas que no corresponden a la pista. También, se incrementó la velocidad de respuesta implementando los algoritmos en computo paralelo usando una unidad de procesamiento gráfica (GPU) con arquitectura unificada de dispositivos de cómputo (CUDA por sus siglas en inglés *Compute Unified Device Architecture*). Más aún, debido a su simplicidad, los algoritmos desarrollados pueden ser implementados fácilmente en otras plataformas computacionales y con otros lenguajes de programación. Los resultados obtenidos mostraron que los métodos opto-digitales son factibles para aplicaciones de navegación visual e instrumentación óptica en vehículos reales.

Como trabajo futuro, se debe considerar el análisis de fuentes de error para robustecer el método propuesto a perturbaciones tales como líneas de carril curvas, señalamientos, semáforos, automóviles, cruces peatonales, y baja visibilidad por lluvia, sombras, y neblina entre otras.



# Apéndice A

## Mínimos cuadrados recursivos con factor de olvido

El método de mínimos cuadrados es un proceso que se usa para predecir los parámetros de un sistema optimizando los resultados de datos experimentales. Este método utiliza un criterio basado en minimizar el error obtenido entre los datos experimentales y la salida del modelo teórico dado por

$$\xi = (y - \tilde{y})^2 = (y - \phi^T \tilde{\theta})^2, \quad (\text{A.1})$$

donde  $\phi$  es el vector de variables de regresión y  $\tilde{\theta}$  son los parámetros del sistema. En el caso específico para múltiples mediciones, la Ec. (A.1) se expande como

$$J(\tilde{\theta}) = \sum_{i=0}^n \left( y_i - \phi_i^T \tilde{\theta}_i \right)^2 = E^T E, \quad (\text{A.2})$$

donde

$$E = [\xi_1, \xi_2, \xi_3, \dots, \xi_n]^T. \quad (\text{A.3})$$

Agregando el factor de olvido para reducir la importancia de las mediciones antiguas, la Ec. (A.2) se reescribe como

$$J(\tilde{\theta}, \lambda) = \sum_{i=0}^n \lambda^{n-i} \left( y_i - \phi_i^T \tilde{\theta}_i \right)^2 = [Y - \Phi \tilde{\Theta}]^T \Lambda [Y - \Phi \tilde{\Theta}], \quad (\text{A.4})$$

donde  $\lambda$  es el factor de olvido, y

$$\begin{aligned} \Phi &= [\phi_1^T, \phi_2^T, \phi_3^T, \dots, \phi_n^T], \\ \tilde{\Theta} &= [\tilde{\theta}_1, \tilde{\theta}_2, \tilde{\theta}_3, \dots, \tilde{\theta}_n], \\ Y &= [y_1, y_2, y_3, \dots, y_n]^T, \\ \Lambda &= \text{diag}(\lambda^{n-1}, \lambda^{n-2}, \lambda^{n-3}, \dots, \lambda^0). \end{aligned} \quad (\text{A.5})$$

Desarrollando la Ec. (A.4), se obtiene la expresión a minimizar como

$$J(\tilde{\theta}, \lambda) = Y^T Y - Y^T \Lambda \Phi \tilde{\Theta} - \Phi^T \tilde{\Theta}^T \Lambda Y + \Phi^T \tilde{\Theta}^T \Lambda \Phi \tilde{\Theta}. \quad (\text{A.6})$$

Los parámetros del sistema se pueden estimar usando la derivada de la función  $J$  con respecto al vector de parámetros e igualando a cero como

$$\begin{aligned} \frac{\partial J}{\partial \tilde{\theta}} &= -2\Phi^T \Lambda Y + 2\Phi^T \Lambda \Phi \tilde{\Theta} = 0, \\ -\Phi^T \Lambda Y + \Phi^T \Lambda \Phi \tilde{\Theta} &= 0. \end{aligned} \quad (\text{A.7})$$

La solución de esta ecuación nos lleva a la estimación de los parámetros  $\tilde{\Theta}$  del modelo dados por

$$\tilde{\Theta} = (\Phi^T \Lambda \Phi)^{-1} \Phi^T \Lambda Y. \quad (\text{A.8})$$

La Ec. (A.8) se puede utilizar recursivamente para realizar estimaciones iterativamente usando estimaciones previas. La recursividad permite optimizar adaptativamente el cálculo de una nueva estimación y despreocupar los parámetros antiguos del sistema. Para esto, definimos las estimaciones de los parámetros como

$$\tilde{\Theta}_{k-1} = D_{k-1}^{-1} Z_{k-1}, \quad (\text{A.9})$$

donde

$$\begin{aligned} D_{k-1} &= \Phi_{k-1}^T \Lambda \Phi_{k-1}, \\ Z_{k-1} &= \Phi_{k-1}^T \Lambda Y_{k-1}, \\ \Phi_{k-1} &= [\phi_1^T, \phi_2^T, \phi_3^T, \dots, \phi_{k-1}^T], \\ Y_{k-1} &= [y_1, y_2, y_3, \dots, y_{k-1}]^T. \end{aligned} \quad (\text{A.10})$$

De la misma manera, podemos deducir la estimación actual como

$$\tilde{\Theta}_k = D_k^{-1} Z_k, \quad (\text{A.11})$$

donde

$$D_k = \lambda D_{k-1} + \phi_k^T \phi_k, \quad (\text{A.12})$$

$$Z_k = \lambda Z_{k-1} + \phi_k^T y_k. \quad (\text{A.13})$$

Dado las definiciones previas, se pueden determinar los parámetros del sistema  $\Theta_k$ . Sin embargo, la inversa de la Ec. (A.12) no puede ser determinada directamente. Para esto, se describirá una serie de pasos para calcular  $D_k^{-1}$  y simplificar la Ec. (A.11).

Partiendo de la Ec. (A.12) y multiplicando  $D_k^{-1}$  por la izquierda, se obtiene

$$\mathbb{I} = \lambda D_k^{-1} D_{k-1} + D_k^{-1} \phi_k^T \phi_k. \quad (\text{A.14})$$

Después, se multiplica  $D_{k-1}^{-1}$  por la derecha a la ecuación previa, obteniendo

$$D_{k-1} = \lambda D_k^{-1} + D_k^{-1} \phi_k^T \phi_k D_{k-1}, \quad (\text{A.15})$$

y multiplicando  $\phi_k^T$  por la derecha a la Ec. (A.15), se tiene

$$D_{k-1} \phi_k^T = \lambda D_k^{-1} \phi_k^T + D_k^{-1} \phi_k^T \phi_k D_{k-1} \phi_k^T. \quad (\text{A.16})$$

Por comodidad, la ecuación anterior se renombra como  $K$ , es decir,

$$K = \frac{D_{k-1} \phi_k^T}{\lambda + \phi_k D_{k-1} \phi_k^T} = D_{k-1} \phi_k^T. \quad (\text{A.17})$$

Al sustituir la Ec. (A.17) con la Ec. (A.15), se obtiene

$$D_{k-1} = \lambda D_k^{-1} + K \phi_k D_{k-1} \phi_k^T, \quad (\text{A.18})$$

y finalmente  $D_k^{-1}$  se puede determinar como

$$D_k^{-1} = \frac{1}{\lambda} (D_{k-1} - K \phi_k D_{k-1}). \quad (\text{A.19})$$

Los parámetros del sistema  $\Theta_k$  se obtiene usando las Ec. (A.11) y (A.19) como

$$\begin{aligned} \tilde{\Theta}_k &= \frac{1}{\lambda} (D_{k-1} - K \phi_k D_{k-1} \phi_k^T) (\lambda Z_{k-1} + \phi_k^T y_k), \\ &= \tilde{\Theta}_{k-1} + \frac{1}{\lambda} D_{k-1}^{-1} \phi_k^T y_k - K \tilde{y}_k - \frac{1}{\lambda} K \phi_k \phi_k^T y_k, \end{aligned} \quad (\text{A.20})$$

que se puede simplificar aún más usando la Ec. (A.18) obteniendo

$$\begin{aligned} \tilde{\Theta}_k &= \tilde{\Theta}_{k-1} + D_k^{-1} \phi_k^T y_k - K \tilde{y} + \frac{1}{\lambda} K \phi_k \phi_k^T y_k - \frac{1}{\lambda} K \phi_k \phi_k^T y_k, \\ \tilde{\Theta}_k &= \tilde{\Theta}_{k-1} + K \xi_k. \end{aligned} \quad (\text{A.21})$$

De acuerdo al análisis de mínimos cuadrados recursivo con factor de olvido, la implementación del algoritmo es simple para cualquier lenguaje de programación. El método posee grandes ventajas para su uso en diversas aplicaciones de visión computacional. Para el caso específico de aplicación en navegación asistida de robots móviles, se utiliza para predecir la posición y orientación del vehículo. Las estimaciones previas permiten generar un filtro para suavizar la detección de líneas de carril. Debido a esto, se logra mantener estable la detección de líneas aún cuando el carril no es visible en algunos instantes.



# Apéndice B

## Rectificación de imágenes

La rectificación de imágenes es una técnica de gran importancia debido a que permite el desarrollo de aplicaciones de captura de escenas tridimensionales [62]. Específicamente, esta técnica ayuda a resolver el problema de búsqueda de puntos de correspondencia para calcular la posición de objetos en el espacio. La rectificación de imágenes permite determinar la profundidad de la escena y posición relativa de los objetos usando visión estéreo [63, 64].

En la literatura se han reportado varios métodos de rectificación de imágenes explotando la geometría epipolar [15, 65–67]. El concepto de *geometría epipolar* simplifica el problema de búsqueda de puntos de correspondencia en un sistema de cámara estéreo. En un sistema estéreo, las cámaras pueden ser diferentes y estar desalineadas. En esta configuración, el punto de intersección entre la línea que une los centros de proyección de las cámaras y el plano imagen se denomina epípolos. Las líneas que pasan por el epípolo se conocen como líneas epipolares. Si  $e$  y  $e'$  son los epípolos en el plano imagen de la primera y segunda cámara, respectivamente, entonces las líneas epipolares están dadas como

$$\ell = \mathcal{H}[e] \times \mathcal{H}[x], \quad (\text{B.1})$$

$$\ell' = \mathcal{H}[e'] \times \mathcal{H}[x'], \quad (\text{B.2})$$

donde  $x$  y  $x'$  son puntos en la imagen de la primera y segunda cámara, respectivamente. La relación entre ambos puntos se puede definir como una transformación proyectiva  $G_\pi$ , dado que

$$x' = \mathcal{H}^{-1}[G_\pi \mathcal{H}[x]]. \quad (\text{B.3})$$

Al sustituir la Ec. (B.1) en la Ec. (B.3), se obtiene

$$\ell' = e' \times G_\pi \mathcal{H}[x]. \quad (\text{B.4})$$

Esta ecuación se puede reescribir usando la matriz antisimétrica  $[e']_\times$  para re-

presentar el producto cruz como

$$\ell' = \underbrace{[e']_{\times} G_{\pi}}_F \mathcal{H}[\mathbf{x}], \quad (\text{B.5})$$

donde  $F$  es la matriz fundamental y

$$[e']_{\times} = \begin{bmatrix} 0 & -e'_z & e'_y \\ e'_z & 0 & -e'_x \\ -e'_y & e'_x & 0 \end{bmatrix}. \quad (\text{B.6})$$

En la siguiente sección, se desarrollará el método de transformación lineal directa para estimación de la matriz fundamental. Posteriormente, se mostrará cómo usar la matriz fundamental para realizar rectificación de imágenes estéreo.

## B.1. Estimación de la matriz fundamental

La matriz fundamental  $F$  establece la relación entre los puntos de correspondencia de dos imágenes que capturan la misma escena. Este método de estimación, no requiere una calibración explícita de las cámaras. Usando puntos de correspondencia  $(\mathbf{x}, \mathbf{x}')$ , la matriz fundamental se puede definir como

$$\mathcal{H}[\mathbf{x}'^T] F \mathcal{H}[\mathbf{x}] = 0. \quad (\text{B.7})$$

Si las coordenadas de los puntos de correspondencia  $(\mathbf{x}, \mathbf{x}')$  son conocidas, entonces, usando la Ec. (B.7), se puede calcular los elementos de la matriz  $F$ . Específicamente, la ecuación correspondiente al punto de correspondencia  $(x, y, 1)$  y  $(x', y', 1)$  es

$$\begin{aligned} x'x F_{11} + x'y F_{12} + x' F_{13} + \\ + y'x F_{21} + y'y F_{22} + y' F_{23} + \\ + x F_{31} + y F_{32} + F_{33} = 0. \end{aligned} \quad (\text{B.8})$$

Factorizando las variables desconocidas, se obtiene

$$[x'x, x'y, x', y'x, y'y, y', x, y, 1] \mathbf{f} = 0_n, \quad (\text{B.9})$$

donde  $\mathbf{f} = [F_{11} \ F_{12} \ F_{13} \ F_{21} \ F_{22} \ F_{23} \ F_{31} \ F_{32} \ 1]^T$ .

La matriz fundamental tiene nueve elementos a determinar; sin embargo, solo existen ocho incógnitas porque la matrices fundamentales son únicas excepto escala. Por lo tanto, se necesitan al menos ocho puntos de correspondencia para estimar una matriz fundamental. Con  $n$  puntos de correspondencia, se obtiene la siguiente ecuación matricial:

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & x_1 y'_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2 x_2 & x'_2 y_2 & x'_2 & x_2 y'_2 & y'_2 y_2 & y'_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 1 \\ x'_n x_n & x'_n y_n & x'_n & x_n y'_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = A \mathbf{f} = \mathbf{0}_n, \quad (\text{B.10})$$

que se puede resolver para  $\mathbf{f}$  usando el método de descomposición en valores singulares. En la siguiente sección, se mostrará cómo emplear la matriz fundamental para realizar rectificación de imágenes capturadas por un sistema estéreo.

## B.2. Rectificación

Partiendo de la Ec. (B.5), observamos que la matriz fundamental

$$F = [\mathbf{e}']_{\times} G_{\pi} \quad (\text{B.11})$$

contiene la homografía del plano epipolar  $G_{\pi}$  y los epípolos  $\mathbf{e}$  y  $\mathbf{e}'$  de ambas cámaras. Esta matriz  $F$  relaciona los puntos de ambas imágenes. Por lo tanto, para cada par de puntos de correspondencia  $(\mathbf{x}, \mathbf{x}')$ , la matriz fundamental  $F$  satisface

$$\mathcal{H}[\mathbf{x}']^T F \mathcal{H}[\mathbf{x}] = 0. \quad (\text{B.12})$$

Una propiedad de las imágenes rectificadas es que los epípolos se encuentran en la coordenada  $\mathbf{i} = [1, 0, 0]^T$ . En este caso, la matriz antisimétrica de  $\mathbf{i}$  es la matriz fundamental del sistema rectificado. La nueva matriz fundamental se define como

$$\bar{F} = [\mathbf{i}]_{\times} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (\text{B.13})$$

Sin embargo, para llevar un sistema estéreo general a un sistema estéreo rectificado equivalente, es necesario transformar las imágenes originales en un par de imágenes rectificadas. Esta transformación implica el uso de un par de homografías que se construyen a partir de la matriz fundamental del sistema.

Las homografías requeridas deben llevar los epípolos del sistema al punto al infinito  $\mathbf{i} = [1, 0, 0]^T$ . Representamos las homografías necesarias como  $G$  y  $G'$  que se aplicarán a las imágenes  $I$  e  $I'$ , respectivamente. Los puntos  $\mathbf{x}$  y  $\mathbf{x}'$  del par de imágenes son transformados de tal forma que la imagen resultante satisfaga la geometría epipolar de un sistema estéreo rectificado. Estos puntos están dados por

$$\mathcal{H}[\bar{\mathbf{x}}] = G \mathcal{H}[\mathbf{x}] \quad \text{y} \quad \mathcal{H}[\bar{\mathbf{x}}'] = G' \mathcal{H}[\mathbf{x}']. \quad (\text{B.14})$$

Los puntos de las imágenes rectificadas cumplen la misma restricción dada por la Ec. (B.12). Por lo tanto, sustituyendo los puntos proyectados, se obtiene

$$\mathcal{H}[\mathbf{x}']^T G'^T [\mathbf{i}]_{\times} G \mathcal{H}[\mathbf{x}] = 0. \quad (\text{B.15})$$

De la Ec. (B.15) se puede observar que la matriz fundamental  $F$  del sistema y la matriz fundamental  $\bar{F}$  de un sistema rectificado están relacionadas como

$$F = G'^T \bar{F} G. \quad (\text{B.16})$$

Las homografías  $G$  y  $G'$  son independientes. Cada una de estas matrices contiene ocho incógnitas por ser únicas excepto escala; es decir

$$G = \begin{bmatrix} u_a & u_b & u_c \\ v_a & v_b & v_c \\ w_a & w_b & 1 \end{bmatrix}, \quad (\text{B.17})$$

y de forma similar para  $G'$ .

La matriz  $G$  se puede descomponer en dos transformaciones simples para simplificar el proceso de estimación. Específicamente, la matriz  $G$  se descompone en

$$G = G_a G_p, \quad (\text{B.18})$$

donde  $G_a$  es una transformación afín y  $G_p$  es la transformación proyectiva que minimiza la distorsión. La transformación  $G_p$  se define como

$$G_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{bmatrix}, \quad (\text{B.19})$$

por lo que podemos escribir  $G_a$  como

$$G_a = G G_p^{-1} = \begin{bmatrix} u_a - u_c w_a & u_b - u_c w_b & u_c \\ v_a - v_c w_a & v_b - v_c w_b & v_c \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.20})$$

De la misma forma definimos las homografías  $G'_a$  y  $G'_p$ .

Anteriormente, se mencionó que las homografías  $G$  y  $G'$  son independientes para cada imagen. Sin embargo,  $\mathbf{w}$  y  $\mathbf{w}'$  son vectores de correspondencia que dependen de la dirección  $\mathbf{z} = [\mu, \lambda, 0]^T$  y su epípolo  $\mathbf{e}$  en  $I$ , tales que

$$\mathbf{w} = [\mathbf{e}]_{\times} \mathbf{z} \quad \text{y} \quad \mathbf{w}' = F \mathbf{z}. \quad (\text{B.21})$$

El par resultante  $\mathbf{w}$  y  $\mathbf{w}'$  representan líneas epipolares. Existen varios vectores de dirección  $\mathbf{z}$  que satisfacen la condición para la rectificación de imagen. Por lo tanto, es necesario encontrar un vector  $\mathbf{z}$  que minimice la distorsión. Para esto, se usa el siguiente criterio de minimización

$$\sum_{i=1}^n \left[ \frac{\mathbf{w}^T (\mathbf{p}_i - \mathbf{p}_c)}{\mathbf{w}^T \mathbf{p}_c} \right]^2. \quad (\text{B.22})$$

Esta sumatoria incluye la transformación que sufre cada píxel en la imagen original y su respectiva imagen transformada por la homografía correspondiente. En el proceso de minimización, se usa el promedio de los puntos  $\mathbf{p}_c = \sum_{i=1}^n \mathbf{p}_i$  para normalizar las coordenadas píxel. Asimismo, se busca que el vector  $\mathbf{w}$  introduzca la mínima diferencia entre todos los puntos de la imagen.

La sumatoria (B.22) se puede expresar de forma matricial como

$$\frac{\mathbf{w}^T P P^T \mathbf{w}}{\mathbf{w}^T \mathbf{p}_c \mathbf{p}_c^T \mathbf{w}}, \quad (\text{B.23})$$

donde  $P$  es una matriz  $3 \times n$  definida como

$$P = [\mathbf{p}_1 - \mathbf{p}_c \quad \mathbf{p}_2 - \mathbf{p}_c \quad \dots \quad \mathbf{p}_n - \mathbf{p}_c]. \quad (\text{B.24})$$



Sustituyendo las definiciones de la Ec. (B.21) en la Ec. (B.23) para ambas imágenes se obtiene

$$\frac{\overbrace{z^T [e]_{\times}^T P P^T [e]_{\times}^T z}^A}{\underbrace{z^T [e]_{\times}^T p_c p_c^T [e]_{\times}^T z}_B} + \frac{\overbrace{z^T F^T P P^T F z}^{A'}}{\underbrace{z^T F^T p_c p_c^T F z}_{B'}}, \quad (\text{B.25})$$

que podemos reescribir de forma compacta como

$$\frac{z^T A z}{z^T B z} + \frac{z^T A' z}{z^T B' z}, \quad (\text{B.26})$$

donde  $A, B, A'$  y  $B'$  son matrices de  $3 \times 3$  que dependen de los puntos de las imágenes. La componente  $w$  del vector  $z$  es 0, por lo tanto, la información útil se encontrará en las primeras dos columnas de las matrices  $A, B, A'$  y  $B'$ . Esto permite reducir las matrices a bloques de  $2 \times 2$ . Explotando esta característica, podemos reducir el bloque superior izquierdo de  $P P^T$  a

$$P P^T = \frac{M N}{12} \begin{bmatrix} N^2 - 1 & 0 \\ 0 & M^2 - 1 \end{bmatrix}, \quad (\text{B.27})$$

donde  $N$  y  $M$  son el ancho y alto de la imagen, respectivamente. De forma similar, para  $p_c p_c^T$  tenemos

$$p_c p_c^T = \frac{1}{4} \begin{bmatrix} (N-1)^2 & (N-1)(M-1) \\ (N-1)(M-1) & (M-1)^2 \end{bmatrix}. \quad (\text{B.28})$$

Usando estos resultados, las matrices  $2 \times 2$  de  $A, B, A'$  y  $B'$  pueden ser obtenidos. Dicho lo anterior, solamente el vector de dirección  $z$  esta por determinar. Se divide en términos la suma (B.26) obteniendo dos estimaciones del vector  $z$ . Para cada vector  $z$ , se busca el eigenvector con el menor eigenvalor de las matrices simétricas positivas ( $A$  y  $A'$ ). El promedio de los vectores  $z$  es el adecuado para cumplir el criterio de la minimización, es decir

$$z = \left( \frac{\tilde{z}_1}{\|\tilde{z}_1\|} + \frac{\tilde{z}_2}{\|\tilde{z}_2\|} \right) / 2. \quad (\text{B.29})$$

De esta manera, las homografías  $G_p$  y  $G'_p$  están disponibles.

Con las homografías calculadas, los epípolos se encontrarán en el infinito. Sin embargo, para la rectificación es necesario asegurar que estén a la dirección  $i = [1, 0, 0]^T$ . La transformación afín se encarga de mantener esa dirección. Para fines prácticos se descompone esta transformación como

$$G_a = G_s G_r, \quad (\text{B.30})$$

donde  $G_r$  y  $G_s$  son la transformaciones de similitud y de corte, respectivamente. La transformación de similitud realiza una rotación y traslación a la imagen para

que el epípolo se encuentre en la dirección  $\mathbf{i} = [1, 0, 0]^T$  alineando a la dirección  $v$ . Esta homografía se define como

$$G_r = \begin{bmatrix} v_b - v_c w_b & v_c w_a - v_a & 0 \\ v_a - v_c w_a & v_b - v_c w_b & v_c \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.31})$$

La matriz de la Ec. (B.31) contiene elementos desconocidos. Únicamente se ha calculado  $\mathbf{w}$  para la transformación proyectiva. Los elementos de  $\mathbf{v}$  se eliminan desarrollando la Ec. (B.16), como

$$F = \begin{bmatrix} v_a w'_a - v'_a w_a & v_b w'_a - v'_a w_b & v_c w'_a - v'_a \\ v_a w'_b - v'_b w_a & v_b w'_b - v'_b w_b & v_c w'_b - v'_b \\ v_a - v'_c w_a & v_b - v'_c w_b & v_c - v'_c \end{bmatrix}. \quad (\text{B.32})$$

Usando el último renglón y columna de la matriz, se determina que

$$v_a = F_{31} + v'_c w_a, \quad (\text{B.33})$$

$$v_b = F_{32} + v'_c w_b, \quad (\text{B.34})$$

$$v_c = F_{33} + v'_c, \quad (\text{B.35})$$

$$v'_a = F_{13} - v_c w'_a, \quad (\text{B.36})$$

$$v'_b = F_{23} - v_c w'_b. \quad (\text{B.37})$$

Sustituyendo las Ec. (B.33)-(B.37) a la Ec. (B.31), se obtiene

$$G_r = \begin{bmatrix} F_{32} - w_b F_{33} & w_a F_{33} - F_{31} & 0 \\ F_{31} - w_a F_{33} & F_{32} - w_b F_{33} & F_{33} + v'_c \\ 0 & 0 & 1 \end{bmatrix}, \quad (\text{B.38})$$

y del mismo modo para  $G'_r$  tenemos

$$G'_r = \begin{bmatrix} w'_b F_{33} - F_{23} & F_{13} - w'_a F_{33} & 0 \\ w'_a F_{33} - F_{13} & w'_b F_{33} - F_{23} & v'_c \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.39})$$

Ambas matrices contienen el elemento  $v'_c$  que involucra una traslación relacionada a las imágenes, alineando los puntos de manera horizontal. El valor numérico de  $v'_c$  es cero para la mínima traslación para cualquier imagen.

Las transformaciones  $G_p$  y  $G_r$  son suficientes para realizar la rectificación de imágenes. Sin embargo, la rectificación usando las dos homografías genera una distorsión en la parte horizontal de las imágenes. Por otro lado, la transformación de corte  $G_s$  minimiza la distorsión generada por la rectificación. Esta transformación solo afecta los puntos de la coordenada  $u$ , tales que

$$G_s = \begin{bmatrix} S_a & S_b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.40})$$

El primer renglón de la transformación  $G_s$  se determina usando dos criterios que evalúan el comportamiento de la transformación proyectiva de la imagen. El primer criterio mantiene la perpendicularidad de dos líneas en la imagen (el cruce de la línea vertical  $\mathbf{m}$  y la línea horizontal  $\mathbf{n}$  en el punto central), tales que

$$(G_s \mathbf{n})^T (G_s \mathbf{m}) = 0. \quad (\text{B.41})$$

El segundo criterio conserva la relación de aspecto de la imagen

$$\frac{(G_s \mathbf{n})^T (G_s \mathbf{n})}{(G_s \mathbf{m})^T (G_s \mathbf{m})} = \frac{w^2}{h^2}. \quad (\text{B.42})$$

Las Ec. (B.41) y (B.42) son polinomios cuadrados en dos dimensiones. Utilizando funciones paramétricas y curvas algebraicas se puede encontrar la solución que satisface ambos criterios, como

$$S_a = \frac{h^2 x_v^2 + w^2 y_v^2}{hw(x_v y_u - x_u y_v)} \quad \text{y} \quad S_b = \frac{h^2 x_u x_v + w^2 y_u y_v}{hw(x_v y_u - x_u y_v)}. \quad (\text{B.43})$$

La parte prima de esta transformación de corte se calcula de la misma manera.

La combinación de las transformaciones  $G_s G_r G_p$  y  $G'_s G'_r G'_p$ , rectifican las imágenes  $I$  e  $I'$ , respectivamente, con la mínima distorsión. En la siguiente sección, se comprueba la validez de este método para la rectificación de imágenes simulando un sistema estéreo no alineado.

Se realizó una prueba usando las imágenes de un sistema estéreo no alineado que se muestran en la figura B.1. Trece puntos de correspondencia fueron seleccionados manualmente de la escena. Usando el método de transformación lineal directa, ver sección B.1, se encontró la siguiente matriz fundamental

$$F = \begin{bmatrix} -0,0011 & -0,1031 & -0,1004 \\ -0,1129 & -0,0222 & -0,7153 \\ -0,1005 & 0,6659 & -0,0135 \end{bmatrix}.$$

Con la matriz fundamental estimada, es posible encontrar líneas epipolares a partir de un punto en la imagen. Debido a esta propiedad, se puede validar que la línea epipolar en dirección  $\mathbf{z}$  es el vector  $\mathbf{w}$  de la homografía. Usando el criterio de minimización descrito se encontró el vector  $\mathbf{w}$  que minimiza la distorsión al transformar los epípolos hacia el infinito. Por lo tanto, con la Ec. (B.19) se obtuvieron las siguientes homografías

$$\mathbf{G}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0,1273 & 1,9614 & 1 \end{bmatrix} \quad \text{y} \quad \mathbf{G}'_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0,1354 & 1,9940 & 1 \end{bmatrix}.$$

Después, las transformaciones de similitud se calculan usando las Ec. (B.38) y (B.39). Las matrices resultantes fueron

$$\mathbf{G}_r = \begin{bmatrix} 0,6924 & 0,1023 & 0 \\ -0,1023 & 0,6924 & -0,0135 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{y} \quad \mathbf{G}'_r = \begin{bmatrix} 0,6883 & -0,0985 & 0 \\ 0,0985 & 0,6883 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

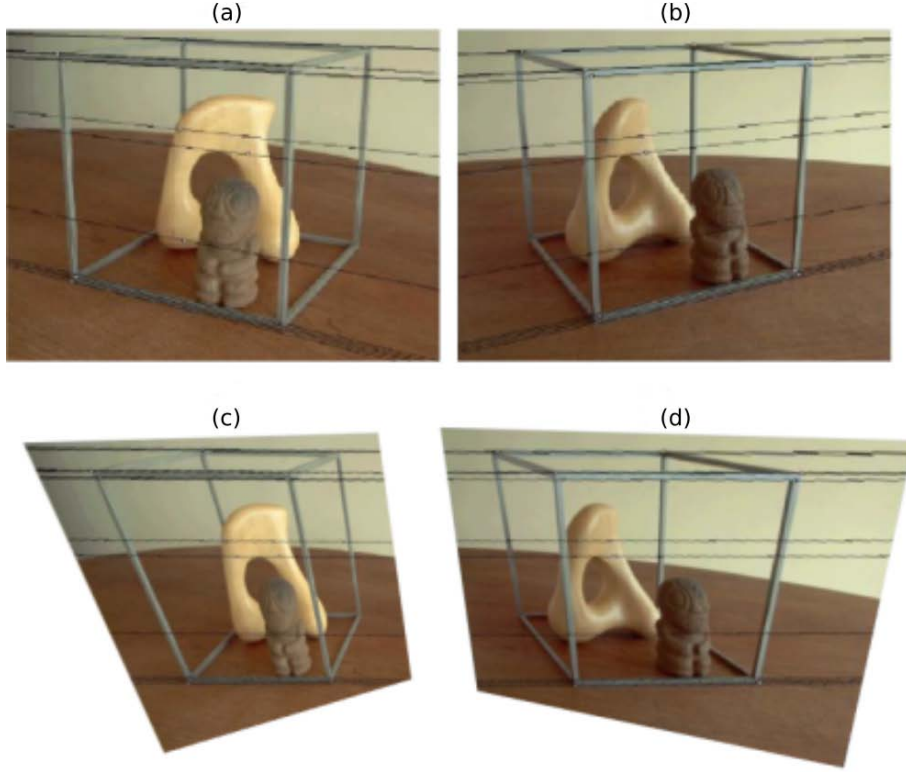


Figura B.1: Escena experimental para el método de rectificación de imágenes. (a)-(b) Un par de imágenes capturadas en diferentes perspectivas. (c)-(d) Imágenes rectificadas.

Para la rectificación de imágenes es suficiente usar las homografías y las matrices de similitud. Sin embargo, la distorsión se puede minimizar con la transformación de corte. Esta transformación trata de mantener la relación de aspecto y la forma de la imagen. Por consiguiente, usando la Ec. (B.40) se obtuvieron las siguientes transformaciones

$$\mathbf{G}_s = \begin{bmatrix} 0,3357 & -2,9149 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{y} \quad \mathbf{G}'_s = \begin{bmatrix} 0,3157 & -2,6452 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Finalmente, se generó una homografía por cada imagen con las combinaciones de las transformaciones  $G_s G_r G_p$  y  $G'_s G'_r G'_p$ . Las homografías resultantes se validaron calculando las imágenes rectificadas correspondientes como se muestra en la figura B.1. Se puede observar que usando las imágenes rectificadas, la búsqueda de puntos en la imagen (2D) se reduce a una búsqueda a lo largo

de un renglón (1D). Además, al observar un punto de interés en la imagen  $I$ , se forma una línea recta con el mismo renglón en la imagen  $I'$  que contiene el punto de correspondencia.

Para la aplicación de estimación de posición y orientación de un robot móvil, el método de rectificación de imágenes estéreo puede ser de utilidad para obtener información tridimensional de la escena. Así, la posición y orientación del robot en el espacio tridimensional puede estimarse con respecto a puntos de referencias en la escena. Mientras el robot se desplaza, la cámara captura una secuencia de imágenes que pueden considerarse fueron capturadas por un sistema estéreo cuyas cámaras son idénticas, pero en diferentes poses. Como trabajo a futuro, se propone analizar la implementación del método de rectificación de imágenes para el problema general de navegación en el espacio tridimensional.



## Segunda transformada de Hough

Se ha mostrado que las columnas de una matriz homografía corresponden a las imágenes de puntos específicos en el plano de referencia. Específicamente, el punto al infinito en la dirección  $x$ , el punto al infinito en la dirección  $y$ , y el punto en el origen. En este trabajo de tesis, se analizó la posibilidad de construir matrices homografía detectando las imágenes de los puntos que forman sus columnas. La detección del punto en el origen no suele ser complicada. Por otro lado, la detección de puntos al infinito puede llegar a ser difícil.

Un punto al infinito puede detectarse indirectamente como el punto de intersección de líneas paralelas. Para detectar líneas, podemos usar la primera transformada de Hough. Y, para detectar puntos de intersección, podemos usar la segunda transformada de Hough con las líneas detectadas por la primera transformada.

La matriz homografía que contiene la posición y orientación del robot móvil terrestre requiere dos puntos al infinito (uno en la dirección  $x$  y otro en la dirección  $y$ ). Podemos usar las líneas de carril para detectar el punto al infinito en la dirección  $y$ ; sin embargo, no hay información suficiente para detectar el punto al infinito en la dirección  $x$ . En otras aplicaciones de navegación donde estén disponibles patrones de guía, como pistas de aterrizaje de drones, la segunda transformada de Hough sería aplicable directamente. En este apéndice, se describe el funcionamiento de la segunda transformada de Hough para trabajo futuro en otras aplicaciones de navegación.

Considere un conjunto de líneas  $\mathbf{l}_k$ ,  $k = 1, \dots, n$  que intersectan en el punto  $\mathbf{p}$ . La segunda transformada de Hough permite encontrar con facilidad el punto de concurrencia  $\mathbf{p}$  representando las líneas detectadas en el *espacio diamante* [68].

Partiendo de la primera transformada de Hough, al sustituir la Ec. (2.82) en la Ec. (2.84), se obtiene

$$\mathbf{d} = \mathcal{H}^{-1}[S^{-T}S\mathcal{H}[\mathbf{p}]] = \mathcal{P}_{S^{-T}S}[\mathbf{p}]. \quad (\text{C.1})$$

Usando las matrices  $S_+$  y  $S_-$  de la Ec. (2.83) y al sustituir en la Ec. (C.1), se obtiene

$$\mathbf{d} = \begin{cases} \mathcal{P}_{D_1}, & D_1 = S_{-1}^{-T} S_{-1}, & \text{si } \mathbf{p} \in \text{I}, \\ \mathcal{P}_{D_2}, & D_2 = S_{-1}^{-T} S_{+1}, & \text{si } \mathbf{p} \in \text{II}, \\ \mathcal{P}_{D_3}, & D_3 = S_{+1}^{-T} S_{-1}, & \text{si } \mathbf{p} \in \text{III}, \\ \mathcal{P}_{D_4}, & D_4 = S_{+1}^{-T} S_{+1}, & \text{si } \mathbf{p} \in \text{IV}, \end{cases} \quad (\text{C.2})$$

donde I, II, III, IV son los cuadrantes del plano de la imagen.

Al utilizar el método del espacio diamante surge un problema al manejar los arreglos de memoria en la computadora. Por un lado, la formación de la imagen utiliza un espacio rectangular para recorrer todos los píxeles en filas y columnas. Por otro lado, el espacio diamante exige la acumulación de polilíneas en puntos que están dispuestos en forma diagonal. Por lo tanto, el manejo de memoria, y el proceso de acumulación de polilíneas en el espacio diamante es ineficiente. Para acelerar este proceso se realiza una modificación al método con una rotación de  $\pi/4$  radianes, un escalamiento de  $\sqrt{2}/2$  y una traslación de  $[1/2, 1/2]^T$ . El resultando es

$$\hat{\mathbf{p}} = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \mathbf{d} + \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad (\text{C.3})$$

y en coordenadas homogéneas,

$$\hat{\mathbf{p}} = \mathcal{H}^{-1}[T\mathcal{H}[\mathbf{d}]], \quad \text{donde } T = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}. \quad (\text{C.4})$$

Por lo tanto, se obtiene una proyección lineal como

$$\hat{\mathbf{p}} = \mathcal{P}_{R_i}[\mathbf{p}], \quad \text{donde } R_i = TD_i. \quad (\text{C.5})$$

La segunda transformada de Hough se utiliza para la detección de puntos de concurrencia, mejor conocidos como *vanishing points* en la literatura y se utilizan para estimar homografías [60]. A continuación, se describen los pasos para la detección de puntos de concurrencia de un patrón cuadro-radial como se muestra en la figura C.1.

Primero, se detectan los contornos del patrón usando el método de Canny como se muestra en la figura C.1(b) y C.1(f). Los contornos detectados son procesados por la primera transformada de Hough para detectar líneas como se muestra en la figura C.1(c) y C.1(g). Entonces, las líneas detectadas son procesadas por la segunda transformada de Hough para detectar los tres puntos de concurrencia requeridos como se muestra en la figura C.1(d) y C.1(h). Finalmente, los puntos de concurrencia detectados son usados para construir la matriz homografía asociada al plano de referencia observado y al plano imagen de la cámara.

Como trabajo futuro, se puede emplear la primera y segunda transformada de Hough para estimación de posición y orientación en aplicaciones donde la escena contiene cuadrículas. Algunos escenarios posibles son ambientes urbanos, pistas de aterrizaje para drones, líneas de producción industrial, y guías por



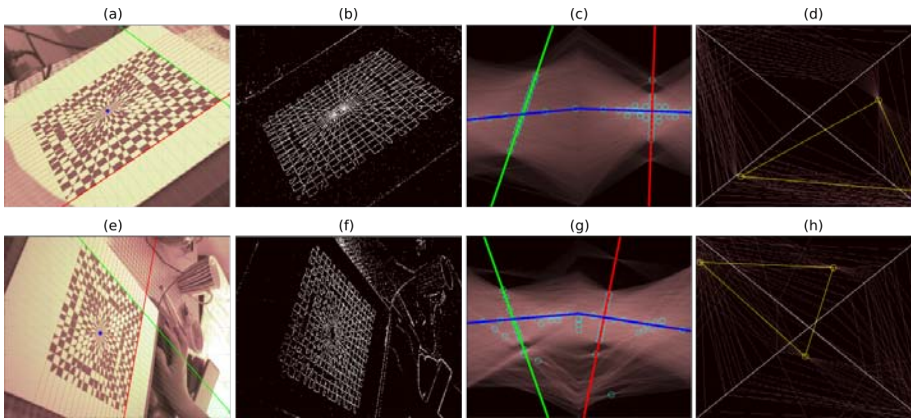


Figura C.1: (a)-(d) y (e)-(h) Par de ejemplos de detección de puntos de concurrencia en un patrón cuadro-radial.

detección de códigos de respuesta rápida (QR, por las siglas en inglés: *Quick Response*).



# Bibliografía

- [1] N. M. Oliver, B. Rosario, A. P. Pentland, A bayesian computer vision system for modeling human interactions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8) (2000) 831–843.
- [2] D. N. Metaxas, *Physics-Based Deformable Models: Applications to Computer Vision, Graphics, and Medical Imaging*, 1st Edition, Kluwer Academic Publishers, USA, 1996.
- [3] T. Starner, J. Auxier, D. Ashbrook, M. Gandy, The gesture pendant: a self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring, in: *Digest of Papers. Fourth International Symposium on Wearable Computers*, 2000, pp. 87–94.
- [4] R. C. Jain, T. O. Binford, Ignorance, myopia, and naiveté in computer vision systems, *CVGIP: Image Understanding* 53 (1) (1991) 112 – 117.
- [5] R. Szeliski, *Computer vision: algorithms and applications*, Springer Science & Business Media, 2010.
- [6] C. Wöhler, *3D computer vision: efficient methods and applications*, Springer Science & Business Media, 2012.
- [7] A. Waxman, J. Moigne, B. Srinivasan, Visual navigation of roadways, in: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, Vol. 2, 1985, pp. 862–867.
- [8] S. Zhang, Recent progresses on real-time 3d shape measurement using digital fringe projection techniques, *Optics and Lasers in Engineering* 48 (2) (2010) 149 – 158.
- [9] A. K. Jain, S. Z. Li, *Handbook of face recognition*, Vol. 1, Springer, 2011.
- [10] T. Huang, *Computer vision: Evolution and promise* (1996).

- [11] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, S. Yogamani, Computer vision in automated parking systems: Design, implementation and challenges, *Image and Vision Computing* 68 (2017) 88 – 101.
- [12] H. Fathi, F. Dai, M. Lourakis, Automated as-built 3d reconstruction of civil infrastructure using computer vision: Achievements, opportunities, and challenges, *Advanced Engineering Informatics* 29 (2) (2015) 149 – 161.
- [13] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*, MIT press, 1993.
- [14] M. Ejiri, Machine vision in early days: Japan's pioneering contributions, in: *Computer Vision – ACCV 2007*, Springer Berlin Heidelberg, 2007, pp. 35–53.
- [15] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, 2004.
- [16] D. A. Forsyth, J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall Professional Technical Reference, 2002.
- [17] B. Cyganek, J. Siebert, An introduction to 3d computer vision techniques and algorithms (2009) 459–474.
- [18] W. F. Riley, K. G. McConnell, J. Dally, J. Dally, J. Dally, *Instrumentation for engineering measurements*, Wiley, 1993.
- [19] J. Dunnicliff, *Geotechnical instrumentation for monitoring field performance*, John Wiley & Sons, 1993.
- [20] J. G. Webster, *Medical instrumentation: application and design*, John Wiley & Sons, 2009.
- [21] W. Demtröder, *Laser spectroscopy: basic concepts and instrumentation*, Springer Science & Business Media, 2013.
- [22] W. R. Williamson, M. F. Abdel-Hafez, I. Rhee, E. Song, J. D. Wolfe, D. F. Chichka, J. L. Speyer, An instrumentation system applied to formation flight, *IEEE Transactions on Control Systems Technology* 15 (1) (2007) 75–85.
- [23] J. Han, D. Kim, M. Lee, M. Sunwoo, Enhanced road boundary and obstacle detection using a downward-looking lidar sensor, *IEEE Transactions on Vehicular Technology* 61 (3) (2012) 971–985.
- [24] H. Wang, B. Wang, B. Liu, X. Meng, G. Yang, Pedestrian recognition and tracking using 3d lidar for autonomous vehicle, *Robotics and Autonomous Systems* 88 (2017) 71 – 78.
- [25] F. Bonin-Font, A. Ortiz, G. Oliver, Visual navigation for mobile robots: A survey, *Journal of Intelligent and Robotic Systems* 53 (3) (2008) 263.

- [26] S. G. Tzafestas, 11 - mobile robot path, motion, and task planning, in: *Introduction to Mobile Robot Control*, Elsevier, Oxford, 2014, pp. 429 – 478.
- [27] W. Fei, C. Jin-Qiang, C. Ben-Mei, L. H. Tong, A comprehensive uav indoor navigation system based on vision optical flow and laser fastslam, *Acta Automatica Sinica* 39 (11) (2013) 1889 – 1899.
- [28] M. P. Ananda, H. Bernstein, K. E. Cunningham, W. A. Feess, E. G. Stroud, Global positioning system (gps) autonomous navigation, in: *IEEE Symposium on Position Location and Navigation. A Decade of Excellence in the Navigation Sciences*, 1990, pp. 497–508.
- [29] J. Cheng, Z. Xiang, T. Cao, J. Liu, Robust vehicle detection using 3d lidar under complex urban environment, in: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 691–696.
- [30] E. Hecht, *Optics*, Pearson Education India, 2012.
- [31] D. Malacara, *Óptica básica*, Fondo de cultura económica, 2015.
- [32] R. E. Fischer, B. Tadic-Galeb, P. R. Yoder, R. Galeb, B. C. Kress, S. C. McClain, T. Baur, R. Plympton, B. Wiederhold, B. G. A. J, *Optical system design*, Vol. 599, Citeseer, 2000.
- [33] J. R. Fienup, Phase-retrieval algorithms for a complicated optical system, *Appl. Opt.* 32 (10) (1993) 1737–1746.
- [34] N. Collings, *Fourier Optics in Image Processing*, 2018.
- [35] I. Pitas, *Digital image processing algorithms and applications*, John Wiley & Sons, 2000.
- [36] R. C. Gonzalez, R. E. Woods, S. L. Eddins, *Digital image processing using MATLAB*, Pearson Education India, 2004.
- [37] J. Kannala, S. S. Brandt, A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (8) (2006) 1335–1340.
- [38] C. Kolb, D. Mitchell, P. Hanrahan, A realistic camera model for computer graphics, in: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 317–324.
- [39] Z. Zhang, A Flexible New Technique for Camera Calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (dec 2000).
- [40] R. Juarez-Salazar, V. H. Díaz-Ramírez, Operator-based homogeneous coordinates: application in camera document scanning, *Optical Engineering* 56 (7) (2017) 070801.

- [41] R. Juárez-Salazar, V. H. Díaz-Ramírez, Homography estimation for camera document scanning applications, in: Optics and Photonics for Information Processing XII, Vol. 10751, SPIE, 2018, p. 17.
- [42] J. Zheng, A. Giron, R. Juárez-Salazar, V. H. Díaz-Ramírez, Image stitching by projective transformations, in: Optics and Photonics for Information Processing XIII, International Society for Optics and Photonics, SPIE, 2019, pp. 51 – 55.
- [43] R. Juárez-Salazar, A. Giron, J. Zheng, V. H. Díaz-Ramírez, Key concepts for phase-to-coordinate conversion in fringe projection systems, Appl. Opt. 58 (2019) 4828–4834.
- [44] L. N. Gaxiola, R. Juárez-Salazar, V. H. Díaz-Ramírez, Simple method for correction of distortion in images, in: Optics and Photonics for Information Processing X, Vol. 9970, 2016, pp. 147 – 153.
- [45] D. Malacara-Hernández, Z. Malacara-Hernández, Handbook of Optical Design, 3rd Edition, CCRC Press, 2013.
- [46] V. Torre, T. A. Poggio, On edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (2) (1986) 147–163.
- [47] D. Marr, E. Hildreth, S. Brenner, Theory of edge detection, Proceedings of the Royal Society of London. Series B. Biological Sciences 207 (1167) (1980) 187–217.
- [48] L. Ding, A. Goshtasby, On the canny edge detector, Pattern Recognition 34 (3) (2001) 721 – 725.
- [49] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (6) (1986) 679–698.
- [50] M. A. Gennert, Detecting half-edges and vertices in images, Worcester Polytechnic Institute, 1986.
- [51] S. Lee, J. Kim, J. Shin Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. Seok Hong, S.-H. Han, I. So Kweon, Vpgnet: Vanishing point guided network for lane and road marking detection and recognition, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 1947–1955.
- [52] B. Caprile, V. Torre, Using vanishing points for camera calibration, International journal of computer vision 4 (2) (1990) 127–139.
- [53] B. S. Song, K. M. Lee, S. U. Lee, Model-based object recognition using geometric invariants of points and lines, Computer Vision and Image Understanding 84 (3) (2001) 361 – 383.

- [54] C. K. Fong, W. K. Cham, 3d object reconstruction from single distorted line drawing image using vanishing points, in: 2005 International Symposium on Intelligent Signal Processing and Communication Systems, 2005, pp. 53–56.
- [55] P. Clark, M. Mirmehdi, Rectifying perspective views of text in 3d scenes using vanishing points, *Pattern Recognition* 36 (11) (2003) 2673 – 2686.
- [56] H. Pottmann, J. Wallner, *Computational line geometry*, Springer Science & Business Media, 2009.
- [57] A. Herout, M. Dubska, J. Havel, *Real-Time Detection of Lines and Grids: By PClines and Other Approaches*, Springer Science & Business Media, 2012.
- [58] A. Ardershir, *Feature Line Detection*, John Wiley & Sons, Ltd, 2017, Ch. 4, pp. 75–131.
- [59] D. H. Ballard, Generalizing the hough transform to detect arbitrary shapes, *Pattern Recognition* 13 (2) (1981) 111 – 122.
- [60] R. Juarez-Salazar, V. H. Diaz-Ramirez, Homography estimation by two pclines hough transforms and a square-radial checkerboard pattern, *Appl. Opt.* 57 (12) (2018) 3316–3322.
- [61] R. Juarez-Salazar, C. Mendoza-Rodriguez, J. E. Hernandez-Beltran, C. Robledo-Sanchez, How do phase-shifting algorithms work?, *European Journal of Physics* 39 (6) (2018) 065302.
- [62] B. D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'81*, Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.
- [63] M. Bertozzi, A. Broggi, Gold: a parallel real-time stereo vision system for generic obstacle and lane detection, *IEEE Transactions on Image Processing* 7 (1) (1998) 62–81.
- [64] D. Murray, J. J. Little, Using real-time stereo vision for mobile robot, *Autonomous Robots* 8 (2) (2000) 161–171.
- [65] R. I. Hartley, Theory and practice of projective rectification, *International Journal of Computer Vision* 35 (2) (1999) 115–127.
- [66] C. Loop, Z. Zhang, Computing rectifying homographies for stereo vision, in: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Vol. 1, 1999, pp. 125–131.
- [67] M. Pascal, M. Jean-Michel, T. Zhongwei, Three-step image rectification, in: *Proceedings of the British Machine Vision Conference*, BMVA Press, 2010, pp. 89.1–89.10.

- [68] M. Dubská, A. Herout, Real projective plane mapping for detection of orthogonal vanishing points., in: BMVC, 2013.