INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGÍA DIGITAL

Navegación por retroalimentación visual usando métodos de procesamiento multidimensional

TESIS

QUE PARA OBTENER EL GRADO DE

DOCTORADO EN CIENCIAS EN SISTEMAS DIGITALES

PRESENTA

M.C. JUAN ZHENG WU

BAJO LA DIRECCIÓN DE

DR. RIGOBERTO JUÁREZ SALAZAR

DR. VÍCTOR HUGO DÍAZ RAMÍREZ

JULIO 2024 TIJUANA, BAJA CALIFORNIA, MÉXICO

SIP-13
REP 2017

INSTITUTO POLITÉCNICO NACIONAL
SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS
Y DESIGNACIÓN DE DIRECTOR DE TESIS

Ciudad de México 21 de enero del 2021

El Colegio de Profesores de Posgrado de Centro de Investigación y Desarrollo de Tecnología Digital en su Sesión
(Unidad Académica)

Ordinaria No. 01 celebrada el día 21 del mes enero de 2021 conoció la solicitud presentada por el (la) alumno (a):

Apellido Paterno:	Zheng	Apellido Materno:	Wu	Nombre (s):	Juan

Número de registro: B 2 0 0 8 7 5

del Programa Académico de Posgrado: DCSD - Doctorado en Ciencias en Sistemas Digitales

Referente al registro de su tema de tesis; acordando lo siguiente:

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

Navegación por retroalimentación visual usando métodos de procesamiento multidimensional

Objetivo general del trabajo de tesis:

Desarrollar un algoritmo para la navegación de un robot móvil terrestre aplicando técnicas ópticas de localización y reconstrucción visual de la escena usando métodos de procesamiento multidimensional

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

Director: Dr. Víctor Hugo Díaz Ramírez 2° Director: Dr. Rigoberto Juárez Salazar

No aplica: ☐

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

Centro de Investigación y 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 aprobación de la versión completa de la tesis por parte de la Comisión Revisora correspondiente.

Director(a) de Tesis

Dr. Víctor Hugo Díaz Ramírez

Aspirante

M. en C. Juan Zheng Wu

2° Director de Tesis (en su caso)

Dr. Rigoberto Juárez Salazar

Presidente del Colegio

Dr. Julio César Rolón Garrido

S.E.P.
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGÍA DIGITAL
DIRECCIÓN

SIP-14
REP 2017

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

Dirección de Posgrado

ACTA DE REVISIÓN DE TESIS

En la Ciudad de Tijuana, B. C. siendo las 10:00 horas del día 05 del mes de JULIO del 2024 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado del: Centro de Investigación y Desarrollo de Tecnología Digital para examinar la tesis titulada: Navegación por retroalimentación visual usando métodos de procesamiento multidimensional del (la) alumno (a):

Apellido Paterno:	Zheng	Apellido Materno:	Wu	Nombre (s):	Juan

Número de boleta: B 2 0 0 8 7 5

Alumno del Programa Académico de Posgrado: Doctorado 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 25 % 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 INDICÓ 25% DE SIMILITUD GENERAL, CON UN DESGLOSE DE 3% EN LA DEFINICIÓN DE UN CONCEPTO DESDE UNA FUENTE, Y MENOS DEL 1% DE SIMILITUD DESDE 176 FUENTES DIVERSAS, POR LO QUE NO CONSTITUYE UN PLAGIO.

****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 LOGRÓ LOS OBJETIVOS DE LA TESIS.
* EL ESTUDIANTE ATENDIÓ LAS REVISIONES INDICADAS.

COMISIÓN REVISORA DE TESIS

Dr. Víctor Hugo Díaz Ramírez
Director de Tesis

Dr. Eduardo Javier Moreno Valenzuela

Dr. Luis Tupak Aguilar Bustos

Dr. Rigoberto Juárez Salazar
Director de Tesis Externo

Dr. Ricardo Ramón Pérez Alcocer
Miembro externo

Dr. Julio César Rolón Garrido
PRESIDENTE DEL COLEGIO DE PROFESORES

S E P
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN Y DESARROLLO DE TECNOLOGÍA DIGITAL
DIRECCIÓN

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE AUTORIZACIÓN DE USO DE OBRA PARA DIFUSIÓN

En la Ciudad de México el día 1 del mes de julio del año 2024, el (la) que suscribe Juan Zheng Wu alumno(a) del programa Doctorado en Ciencias en Sistemas Digitales con número de registro B200875, adscrito(a) al Centro de Investigación y Desarrollo de Tecnología Digital manifiesta que es autor(a) intelectual del presente trabajo de tesis bajo la dirección del Dr. Rigoberto Juárez Salazar y Dr. Víctor Hugo Díaz Ramírez y cede los derechos del trabajo intitulado Navegación por retroalimentación visual usando métodos de procesamiento multidimensional, 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 expresado del autor y/o director(es). Este puede ser obtenido escribiendo a las siguiente(s) dirección(es) de correo. jzheng@citedi.mx, rjuarez@citedi.mx, vhdiaz@citedi.mx. Si el permiso se otorga, al usuario deberá dar agradecimiento correspondiente y citar la fuente de este.

Juan Zheng Wu

Nombre completo y firma autografá del (de la) Estudiante

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mi familia por su invaluable apoyo y facilitar el tiempo y espacio necesarios para la dedicación a mi trabajo de tesis.

Además, debo reconocer con profunda gratitud a mis asesores, Dr. Rigoberto Juárez Salazar y Dr. Víctor Hugo Díaz Ramírez. Su atención incondicional, el compartir generosamente sus conocimientos y experiencias, así como la constante retroalimentación, han sido fundamentales en mi desarrollo académico.

También deseo agradecer al Dr. Eduardo Javier Moreno Valenzuela, al Dr. Ricardo Ramón Pérez Alcocer y al Dr. Luis Tupak Aguilar Bustos, miembros de mi comité tutorial, por su disposición a compartir su tiempo y sus valiosas retroalimentaciones que han sido cruciales para culminar este proyecto.

Un reconocimiento especial al Centro de Investigación y Desarrollo de Tecnología Digital del Instituto Politécnico Nacional por brindar un lugar para mi formación científica. Aprecio sinceramente a los profesores por su enseñanza, al personal administrativo por su amable gestión, al personal de intendencia por mantener un ambiente de trabajo limpio y agradable, y al personal de seguridad por asegurar nuestra integridad durante la estancia en el centro.

Asimismo, agradezco a mi equipo de trabajo y compañeros del laboratorio de Procesamiento Digital de Imágenes por sus consejos y motivación constante, que me impulsaron a ser un mejor investigador, así como a todos mis compañeros por la amistad y el compañerismo mostrados.

Finalmente, extiendo un agradecimiento especial al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por el apoyo de manutención proporcionado, que ha sido esencial para la realización de este trabajo de tesis.

Dedicatoria

A mi amada esposa, Andrea Pozos Soto.

Por llegar a mi vida en medio de mi camino profesional y transformarla con tu amor, tu apoyo incondicional y tu fe inquebrantable en mí. Has sido mi soporte y mi mayor fuente de inspiración. Sin ti, este logro no habría sido posible.

Con todo mi amor y gratitud, Juan Zheng Wu.

Navegación por retroalimentación visual usando métodos de procesamiento multidimensional

Resumen

Los sistemas de visión son un medio esencial de percepción tanto para los humanos como para los robots autónomos. Estos sistemas permiten la percepción de colores, reconocimiento de formas, medición de tamaños y distancias, clasificación de objetos y la interpretación de escenas. No obstante, explotar todo el potencial de un sistema de visión en una computadora digital para aplicaciones robóticas presenta desafíos significativos. Procesando toda la información visual de una escena para lograr un reconocimiento y localización robustos de objetos sigue siendo un problema abierto de gran interés. Las escenas naturales son datos multidimensionales que deben ser extraídos utilizando diversos tipos de sensores. Los sistemas opto-digitales facilitan la creación e interpretación de datos mediante algoritmos de alto rendimiento. Además, las cámaras digitales ofrecen ventajas significativas, como un amplio campo de visión, mediciones de alta resolución, bajo consumo de energía y costos reducidos. En esta tesis, se propone un algoritmo de navegación por retroalimentación visual usando métodos de procesamiento multidimensional. El vehículo terrestre realizará una rutina dentro de una plataforma digital, y se corregirá su trayectoria usando la información visual obtenida por las imágenes capturadas por el sistema. Se presentará el modelo de pinhole con distorsión radial para determinar los parámetros de las cámaras con lentes de campo visual amplio. Posteriormente, los puntos de correspondencia se rastrearán a través de distintos métodos como flujo óptico, detección de colores, y filtros de correlación. Después, se determinará la pose del vehículo en el espacio tridimensional. Finalmente, la información tridimensional obtenida se utilizará para proporcionar retroalimentación a un robot móvil terrestre.

Palabras clave: Navegación visual, detección de objetos, reconstrucción tridimensional, flujo óptico, filtros de correlación, corrección de distorsión, cámara pinhole con distorsión radial, calibración de cámara, visión computacional.

Visual feedback navigation using multidimensional processing methods

Abstract

The vision system is an important means of perception for both humans and autonomous robots. Vision systems allow perception of colors, recognition of shapes, measuring size and distances, classification of objects, and interpretation of scenes. Unfortunately, exploiting the full capability of a vision system in a digital computer for robot applications is not simple. Processing all the visual information of a scene for robust object recognition and location is still an open problem of great interest. Natural scenes are multidimensional data that need to be extracted using different types of sensors. Opto-digital systems allow data to be created and interpreted using high-performance algorithms. Additionally, digital cameras have significant advantages, such as a wide field of view, high-resolution measurements, low power consumption, and low cost. In this thesis, a visual feedback navigation algorithm using multidimensional processing methods is proposed. A grounded vehicle will perform a routine within a digital test-bench platform, and the trajectory will be corrected using the visual information obtained by the algorithms through the images captured by the visual system. The pinhole model with radial distortion is presented. This model determines the parameters of cameras with high field-of-view lenses. Next, point correspondences are tracked using various methods, such as optical flow, color detection, and correlation filters. Afterward, the location of the vehicle in three-dimensional space is determined. Finally, the obtained three-dimensional information is used to provide feedback on a land mobile robot.

Keywords: Visual navigation, object detection, three-dimensional reconstruction, optical flow, correlation filters, distortion correction, pinhole camera with radial distortion, camera calibration, computer vision.

Índice general

Índice de cuadros

Índice de figuras

Capítulo 1

Introducción

La navegación de robots móviles ha emergido como un campo de gran importancia tanto en el ámbito científica y comercial [1–3]. Está rama de la robótica, se enfoca en dirigir vehículos autónomos de manera eficiente y segura a través de sus respectivos entornos. Por esta razón, el desarrollo y mejoramiento de los sistemas de navegación es importante para incrementarla confiabilidad y seguridad en aplicaciones tales como logística automatizada, y asistencia personal en entornos domésticos, entre otras.

La navegación autónoma de un vehículo requiere varios sensores para detectar la orientación y posición del vehículo en el espacio, y desplazarse por una ruta que puede ser predeterminada. Los láseres y sonares han sido los sensores predominantes para estas aplicaciones, proporcionando datos precisos de pose y proximidad entre objetos de la escena [4]. No obstante, estos sensores presentan desafíos significativos en términos de costo y eficiencia energética, además, tiene un alcance de detección limitado que puede comprometer la efectividad de entornos complejos o no estructurados. Por otro lado, los avances recientes en visión computacional están marcando grandes logros en la navegación de robots móviles. Las cámaras modernas, particularmente los sistemas de visión estéreo y los sistemas de inteligencia artificial, extiende considerablemente las capacidades de visión de los vehículos autónomos. Además, se los sistemas de visión proporciona información visual abundante que puede ser aprovechada para una amplia gama de tareas, incluyendo la navegación y el reconocimiento de objetos.

El uso de un sistema de visión para la navegación de robots móviles no solo reduce los costos asociados, sino que también disminuye el consumo energético, un factor crítico para la operación prolongada de robots autónomos. Las técnicas de visión por computadora permiten la integración de algoritmos de inteligencia artificial y reconocimiento de patrones, los cuales pueden mejorar significativamente la autonomía y la adaptabilidad de los robots móviles en entornos dinámicos.

Los robots de navegación visual pueden realizar un mapeo y localización

simultáneos (SLAM) de forma más efectiva comparado en contra de sensores tradicionales [5]. Además, la integración de técnicas como inteligencia artificial permite que los robots móviles interpreten y respondan a su entorno de manera más natural y eficiente, imitando los aspectos del procesamiento visual humano.

Los sensores tradicionales siguen siendo prevalente en el área de investigación, sin embargo, la integración de tecnologías de visión computacional indica mejoras donde los robots móviles serán capaces de operar de manera más autónoma y eficiente, incluso en entornos complejos y cambiantes. Esto no solo mejora la capacidad para desempeñar tareas en distintos escenarios, sino que también abre nuevas aplicaciones de la robótica, desde la exploración de terrenos hasta la asistencia en tareas domésticas y urbanas [6–10].

Los sistemas de visión aprovechan máxima capacidad de las cámaras para capturar datos multidimensionales del entorno, los cuales son cruciales para la navegación en entornos complejos. Por esto mismo, la implementación de algoritmos para procesar los datos y extraer la información aún son desafiantes. Estos métodos incluyen la detección de objetos, reconocimiento de objetos, la estimación de la posición y orientación (pose) de la cámara, y el descarte de información irrelevante o ruidosa. Este problema y manejo de datos se vuelven intrínsecamente multidimensionales. Cada imagen capturada no solo representa una matriz bidimensional de píxeles, sino que también incluye información de color a través de los canales RGB, coordenadas espaciales de cada pixel, múltiples parámetros de la pose de la cámara, y marcas de tiempo que indican cuándo se capturó cada imagen.

Además de la complejidad inherente, los sistemas de navegación visual frecuentemente se complementan con sensores auxiliares como giróscopos y acelerómetros. Estos dispositivos ayudan a mejorar la precisión de la estimación de la pose de la cámara y del robot, proporcionando datos sobre el movimiento y la orientación que pueden ser difíciles de obtener solo a través de las imágenes. La integración de estos datos sensoriales múltiples en un marco coherente es crucial y se realiza a través de técnicas de fusión de datos.

A pesar de los avances en la navegación visual, existen varios desafíos críticos que se puede enfrentar. La complejidad computacional es uno de ellos, el procesamiento de grandes volúmenes de datos visuales y en tiempo real requiere algoritmos optimizados y equipo de alto rendimiento. Otro desafío importante es la robustez del sistema frente a la pérdida de características visuales críticas, que puede ocurrir debido a cambios de iluminación, obstrucciones temporales, o entornos poco estructurados. Por último, en el caso de emplear varios sensores, la integración o fusión de datos provenientes de múltiples fuentes es otro desafío a superar. Estos aspectos son cruciales para garantizar que los robots móviles puedan operar de manera eficiente y segura en entornos complejos y dinámicos. Aunque se han hecho progresos, la capacidad de los robots para navegar en entornos inciertos con precisión todavía está lejos de ser óptima [11, 12]. Estos desafíos son detonantes para la investigación científica dentro de la comunidad de visión por computadora, y resolverlos es clave para lograr sistemas de navegación prácticos y efectivos en una amplia variedad de aplicaciones.

Uno de los componentes fundamentales que requiere atención es la estima-

ción de la pose del vehículo, que es vital para cualquier sistema de navegación de robots móviles. La pose del vehículo, que incluye posición y orientación, debe determinarse con alta precisión para que el sistema de navegación funcione correctamente. Las inexactitudes en la estimación de la pose originan errores de navegación que conducen a la incapacidad de realizar las tareas de navegación asignadas e incluso colusiones. Por esta razón, es esencial mejorar las técnicas de estimación de pose. Esto incluye en desarrollar nuevos algoritmos que integren de manera más efectiva múltiples sensores, como cámaras, LIDAR, y sensores inerciales. La fusión de datos de estos diversos sensores, a través de técnicas como el filtro de Kalman, o el filtro de partículas, puede proporcionar una estimación más precisa y robusta de la pose del robot.

Por otro lado, la implementación de tecnologías de aprendizaje automático y aprendizaje profundo puede ofrecer mejoras significativas en la capacidad de los sistemas de navegación para adaptarse y responder a entornos no estructurados o desconocidos. Por ejemplo, los modelos de aprendizaje profundo pueden ser entrenados para identificar patrones complejos y adaptarse a variables del entorno. Así, los modelos de aprendizaje tienen el potencial de mejorar la precisión de la estimación de pose y toma de decisiones en tiempo real. Sin embargo, el proceso de entrenamiento de los sistemas de aprendizaje requiere grandes cantidades de imágenes y mucho tiempo de procesamiento. Los sistemas digitales permite procesar datos crudos en modelos o patrones simples que pueden ser utilizados para diversas aplicaciones, incluyendo la detección y localización de objetos [13–17]. Por ejemplo, los sistemas de visión permiten detectar la ubicación bidimensional de un objeto en múltiples imágenes y determinar por triangulación la posición del objeto y el robot en el espacio. Este proceso es fundamental para la navegación y manipulación precisa en el espacio [18].

En este trabajo de tesis se propone desarrollar un algoritmo para la navegación de un robot móvil terrestre mediante retroalimentación visual. El enfoque se centrará en el uso de un modelo de cámara pinhole, que incluye la corrección de distorsión radial, para representar adecuadamente la geometría y el campo de visión del sistema visual del vehículo. Este modelo es particularmente útil para estimar y corregir distorsión radial, comúnmente encontrada imágenes capturadas por cámaras digitales convencionales.

Se utilizarán técnicas de calibración para obtener los parámetros intrínsecos y extrínsecos de la cámara, así como su distorsión. Los parámetros intrínsecos se relacionan con las características ópticas de la cámara, como la distancia focal, tamaño de píxel, distorsión, y el centro óptico, mientras que los parámetros extrínsecos describen la posición y orientación de la cámara respecto a un sistema de referencia global. Estos parámetros son la parte fundamental para el sistema de visión interprete correctamente la información espacial del entorno y así facilitar una navegación precisa y efectiva. El enfoque utilizado fue concebido para lograr alta precisión en la reconstrucción 3D y aumentar la eficiencia del sistema de navegación móvil.

En esta tesis se propone un método de navegación basado en retroalimentación visual empleando procesamiento multidimensional. Este trabajo aborda específicamente los retos asociados con la estimación de la posición y orien-

tación del vehículo y la reconstrucción tridimensional del entorno a partir de información visual. La implementación del método propuesto utiliza una cámara previamente calibrado y los puntos de correspondencia entre el plano de la imagen y el espacio tridimensional son detectados. El método propuesto se basa en los conceptos fundamentales de formación de imágenes y la adaptación del modelo *pinhole* a cámaras con lentes de campo visual amplio (*fisheye*). Se incorporaron técnicas detección y seguimiento de puntos para localización y seguimiento de características en la escena, además de triangulación de puntos para la detección de características tridimensionales y ubicación espacial. Finalmente, se optimiza la estimación de la pose de la cámara, permitiendo una navegación precisa y el trazo de la trayectoria del vehículo.

El método propuesto fue evaluado experimentalmente usando secuencias de video. Los resultados obtenidos fueron analizados en términos de la precisión en la estimación de la trayectoria de la cámara. El método propuesto es esencial para la navegación autónoma y ha demostrado ser un área de interés creciente debido a su aplicación potencial en diversos campos de la investigación, como la robótica móvil y la realidad aumentada. En resumen, esta investigación propone mejorar la precisión y robustecer los sistemas de navegación visual, superando los desafíos actuales y ampliando su aplicabilidad en prácticas reales. La contribución de esta tesis no solo avanza en el campo académico, sino que también tiene el potencial de influir significativamente en las aplicaciones industriales y comerciales, mejorando la autonomía y la eficiencia de los sistemas de navegación visual en entornos complejos.

Este documento está organizado de la siguiente forma. En el capítulo 2 se analizan los principios teóricos para determinar los parámetros de la cámara pinhole con distorsión radial. Después, el capítulo 3 propone el método de estimación de pose usando la información tridimensional. Posteriormente, el capítulo 4 presenta las herramientas configuradas para la validación del algoritmo propuesto. En el capítulo 5 se presentan los resultados obtenidos de este trabajo de tesis. En el capítulo 6 se presentan las conclusiones del trabajo de investigación y el trabajo a futuro. Este documento de tesis es complementado por dos apéndices que facilitan la implementación del algoritmo propuesto. El apéndice A presenta conceptos para estimar la homografía usando puntos de correspondencia. Finalmente, el apéndice B presentan los métodos de detección utilizados en este trabajo de tesis.

1.1. Objetivos

1.1.1. Objetivo general

El objetivo general de esta tesis es desarrollar un algoritmo para la navegación de un robot móvil terrestre aplicando técnicas ópticas de localización y reconstrucción visual de la escena usando métodos de procesamiento multidimensional.

1.1.2. Objetivos específicos

Los objetivos específicos para el desarrollo de este trabajo de tesis son los siguientes.

- Análisis y evaluación del modelo de cámara pinhole con distorsión radial.
- Implementación y calibración de cámaras con lentes de campo de visión amplio.
- Análisis y evaluación de un método de detección de objetos.
- Análisis y evaluación de un método triangulación para sistemas con múltiples dispositivos.
- Diseño de una plataforma de proyección de escenas para la evaluación de la navegación de vehículos terrestres.

1.2. Contribuciones

Parte de los resultados derivados de este trabajo de tesis fueron publicados en dos artículos científicos y seis memorias de congreso internacional.

- Victor H. Diaz-Ramirez, Rigoberto Juarez-Salazar, Juan Zheng, Jose Enrique Hernandez-Beltran, and Andrés Márquez, "Homography estimation from a single-point correspondence using template matching and particle swarm optimization," Appl. Opt. 61, D63-D74 (2022).
  DOI: https://doi.org/10.1364/AO.444847.
- Rigoberto Juarez-Salazar, Juan Zheng, and Victor H. Diaz-Ramirez, "Distorted pinhole camera modeling and calibration," Appl. Opt. 59, 4828-4834 (2020).
  DOI: https://doi.org/10.1364/AO.412159.
- Juan Zheng, Rigoberto Juarez-Salazar, y Victor H. Diaz-Ramirez, "Vision-based pose estimation for robot navigation in an uncontrolled environment," Proc. SPIE 12673, Optics and Photonics for Information Processing XVII (2023).
  DOI: https://doi.org/10.1117/12.2677909.
- Rigoberto Juarez-Salazar, Sofia Esquivel-Hernandez, Juan Zheng, y Victor H. Diaz-Ramirez, "Fringe projection profilometry without explicit projector calibration," Proc. SPIE 12673, Optics and Photonics for Information Processing XVII (2023).
  DOI: https://doi.org/10.1117/12.2677133.
- Juan Zheng, Rigoberto Juarez-Salazar, y Victor H. Diaz-Ramirez, "Dynamic scene multi-projector platform for vehicle navigation evaluation," Proc. SPIE 12225 Optics and Photonics for Information Processing XVI

(2022).
DOI: https://doi.org/10.1117/12.2633690.

- Rigoberto Juarez-Salazar, Sofia Esquivel-Hernandez, Juan Zheng, y Victor H. Diaz-Ramirez, "Stereo vision-based 3D pointer for virtual object interaction," Proc. SPIE 12225, Optics and Photonics for Information Processing XVI (2022).
  DOI: https://doi.org/10.1117/12.2633736.

- Juan Zheng, Rigoberto Juarez-Salazar, y Victor H. Diaz-Ramirez, "Distorted pinhole model for image warping in lane detection applications," Proc. SPIE 11841, Optics and Photonics for Information Processing XV (2021).
  DOI: https://doi.org/10.1117/12.2594934.

- 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 11841, Optics and Photonics for Information Processing XIV (2020).
  DOI: https://doi.org/10.1117/12.2569737.

Capítulo 2

Modelo de cámara opto-digital

2.1. Proceso de formación de imágenes

El proceso de formación de imágenes usando el modelo cámara *pinhole* es la configuración más estudiada en la literatura debido a su simplicidad [19,20]. Para aplicaciones de visión por computadora es suficiente el modelo pinhole debido a que la distorsión radial en las cámaras modernas son despreciables. No obstante, cuando se requiere adquirir la información que abarque la mayor parte de la escena, es necesario utilizar una configuración con lentes de alto campo visual. Estas lentes generan una distorsión radial fuerte en las imágenes capturadas no reproducibles con el modelo pinhole, provocando discrepancias en los resultados estimados. Para evitar los errores de estimación, se debe corregir la distorsión radial de las imágenes de entrada. Por esto, es fundamental determinar los parámetros de distorsión la cámara. Usualmente se asume que los parámetros de distorsión de la cámara son aproximados, independientemente del proceso de formación de imagen [21–23]. Sin embargo, al analizar los parámetros de la cámara, se observa que la distorsión radial depende tanto del tamaño del pixel como de la coordenada del punto principal. Por este motivo, la distorsión radial se toma en cuenta dentro del proceso de formación de imágenes. Por esta razón, se propone un modelo matemático para el proceso de formación de imágenes que extiende el modelo de la cámara pinhole para incluir la distorsión radial. Este modelo propuesto se divide en tres principios: transformación rígida, trazo de rayos y muestreo.

2.1.1. Transformación rígida

La transformación rígida describe la posición y orientación de la cámara en el espacio tridimensional. Esta transformación determina la relación entre las coordenadas del mundo real (xyz) y las coordenadas de la cámara (uvw), como se muestra en la figura 2.1. En el sistema de coordenadas global, un punto en el

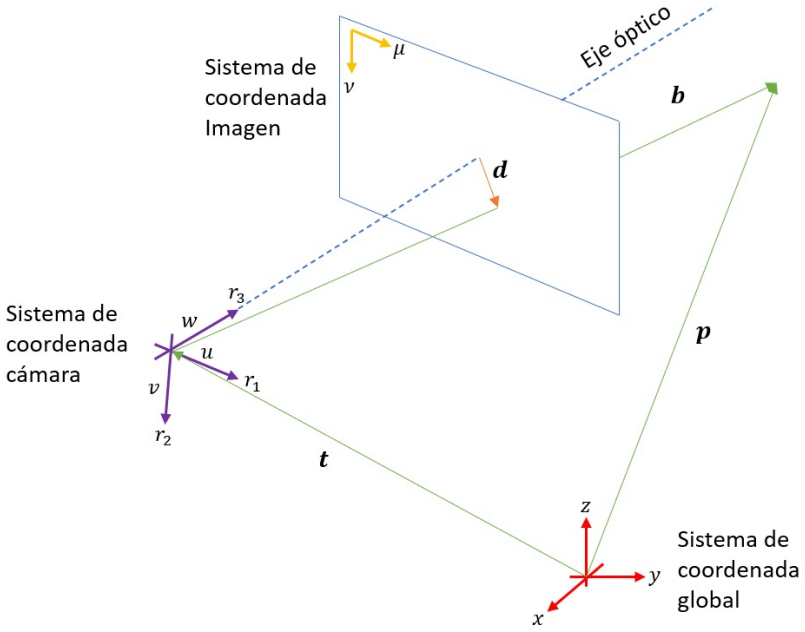

Figura 2.1: La pose de la cámara y los puntos en el espacio tridimensional son definidas por el sistema de coordenadas global xyz. Mientras, el trazo de rayos es descrita por el sistema de coordenadas de la cámara uvw. Por último, el plano imagen es descrita por su sistema de coordenadas pixel $\mu\nu$.

espacio tridimensional se representa como

$$\boldsymbol{p} = [p_x, \quad p_y, \quad p_z]^T. \tag{2.1}$$

Cuando el punto $\boldsymbol{p}$ es observado por la cámara, la coordenada local que representa el mismo punto se expresa como un vector $\boldsymbol{b} = [b_u, \quad b_v, \quad b_w]^T$ dado por

$$\boldsymbol{b} = R^T(\boldsymbol{p} - \boldsymbol{t}), \tag{2.2}$$

donde R es una matriz de rotación y $\boldsymbol{t}$ es un vector de traslación. Ambos parámetros, rotación y traslación, especifican la pose de la cámara. La ecuación (2.2) se puede expresar de manera más conveniente utilizando el operador de coordenadas homogéneas $\mathcal{H}$ como

$$\boldsymbol{b} = L\mathcal{H}[\boldsymbol{p}], \tag{2.3}$$

donde

$$L = [R^T, \quad -R^T\boldsymbol{t}] \tag{2.4}$$

es conocido en la literatura como la matriz de los parámetros extrínsecos de la cámara.

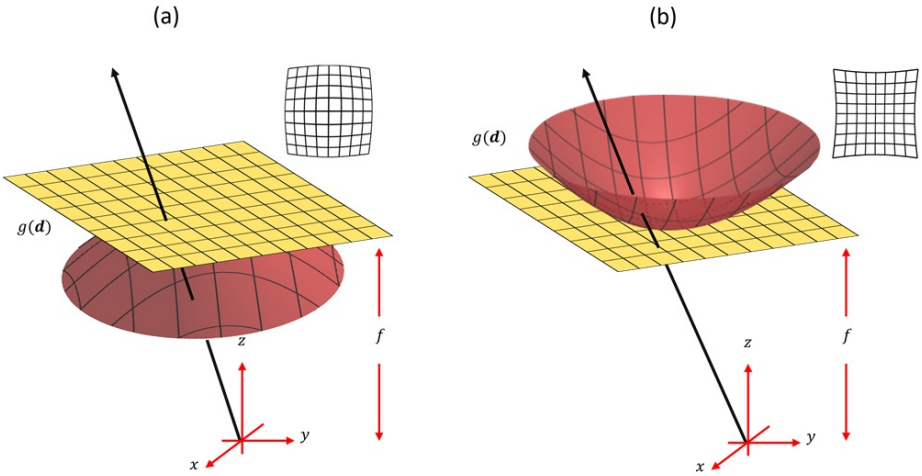

Figura 2.2: Superficie del plano imagen producida por la función $g(\boldsymbol{d})$ que reproduce distorsión radial de tipo (a) barril, y (b) cojín. Cuando la distorsión radial de la cámara es nula, la superficie es plana, como se ilustra con el plano amarillo en la figura.

2.1.2. Trazo de rayos

El trazo de rayos es una descripción abstracta para representar los rayos de luz que viajan desde la escena, cruzan el pinhole de la cámara, y alcanzan el plano imagen. Para realizar esta transformación, es conveniente definir el rayo de luz como aquella línea recta que pasa por el punto $\boldsymbol{b}$ y el origen del sistema de referencia de la cámara; esto es,

$$\boldsymbol{\ell} = \lambda \boldsymbol{b}. \tag{2.5}$$

Por lo tanto, el punto $\boldsymbol{b}$ es detectado como un punto $\boldsymbol{d} = [d_v, d_v]$ en la imagen que es la intersección de la función $g(\boldsymbol{d})$, que representa la imagen, y la línea $\boldsymbol{\ell}$. Es decir,

$$\begin{bmatrix} \boldsymbol{d} \\ g(\boldsymbol{d}) \end{bmatrix} = \lambda \boldsymbol{b}. \tag{2.6}$$

La distorsión radial es determinada por la distancia del punto $\boldsymbol{d}$ desde el punto principal. Dado este razonamiento, se puede modelar la superficie como un polinomio en potencias de la norma Euclidiana $\|\boldsymbol{d}\|$. Por lo tanto, la superficie de la imagen es representada como

$$g(\boldsymbol{d}) = f + \delta_2 \|\boldsymbol{d}\|^2 + \delta_3 \|\boldsymbol{d}\|^3 + \cdots + \delta_\omega \|\boldsymbol{d}\|^\omega, \tag{2.7}$$

donde $\delta_k, k = 1, 2, \ldots, w$ son los parámetros de distorsión. La figura 2.2 illustra dos tipos de superficie $g(\boldsymbol{d})$ que generan la distorsión radial barril y cojín, que se encuentra típicamente en la literatura.

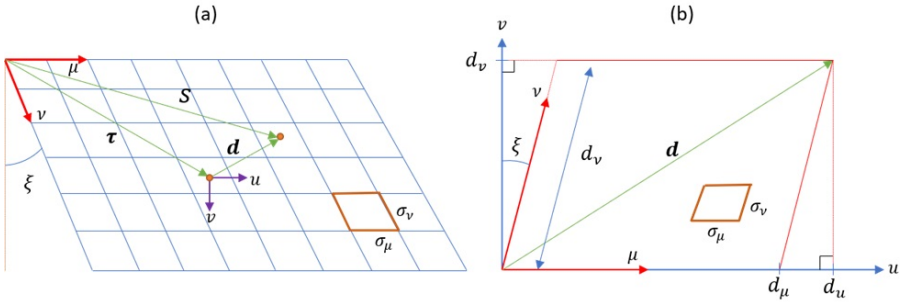

Figura 2.3: (a) Arreglo de píxeles fotosensibles donde su forma está dada por el ángulo de oblicuidad ξ. (b) Representación de puntos en coordenadas físicas a coordenadas píxel.

2.1.3. Muestreo

El muestreo es el proceso de capturar la imagen proyectada en el plano de la cámara, convirtiendo la información continua de la escena en un conjunto de datos discretos (pixeles). El sensor de la cámara contiene un arreglo de píxeles fotosensibles donde las imágenes son formadas a partir de la información generada en la superficie por el trazo de rayos. De este modo, los puntos $\boldsymbol{d}$ en coordenadas uv son transformadas a coordenadas pixel $\mu\nu$, como se muestra en la figura 2.3. Considerando que el sensor de la cámara tiene pixeles de tamaño $\sigma_\mu \times \sigma_\nu$ y un ángulo de oblicuidad ξ, las coordenadas del punto principal (d_u, d_v) se puede transformar a su respectivo coordenada píxel (d_μ, d_ν) como

$$d\mu = \frac{d_u - d_v \tan\xi}{\sigma_\mu}, \quad \text{y} \quad d_\nu = \frac{d_v \sec\xi}{\sigma_\nu}. \tag{2.8}$$

Esta transformación se puede generalizar para cualquier punto de observación. Por lo tanto, si $\boldsymbol{\tau} = [\tau_\mu, \tau_\nu]^T$ son las coordenadas pixel del punto principal. Entonces, la coordenada pixel $\boldsymbol{s}$ de un punto imagen $\boldsymbol{d}$ es dada por

$$\mathcal{H}[\boldsymbol{s}] = S\mathcal{H}[\boldsymbol{d}], \tag{2.9}$$

donde S es la matriz de muestreo definida como

$$S = \begin{bmatrix} 1/\sigma_\mu & -(\tan\xi)/\sigma_\mu & \tau_\mu \\ 0 & (\sec\xi)/\sigma_\nu & \tau_\nu \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.10}$$

Una vez definida la matriz de muestreo, el proceso del trazo de rayos y el muestreo se puede simplificar como

$$\mathcal{H}[\boldsymbol{s}] = K\mathcal{H}[\boldsymbol{b}], \tag{2.11}$$

donde K es la matriz de los parámetros intrínsecos de la cámara. La matriz K es definida como

$$K = S\Xi_f = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.12}$$

donde

$$\Xi_f = \begin{bmatrix} fI_2 & \mathbf{0}_2 \\ \mathbf{0}_2^T & 1 \end{bmatrix}, \tag{2.13}$$

es la matriz diagonal de escala, f es la distancia focal, I_2 es la matriz identidad de tamaño 2×2, y $\mathbf{0}_2$ es un vector de ceros de tamaño 2×1. Con estos conceptos en mente, se procede a examinar el modelo de cámara pinhole con distorsión radial.

2.2. Modelo pinhole con distorsión radial

Para el modelo de la cámara pinhole con distorsión radial, la matriz K de parámetros intrínsecos es usada para expresar la re-proyección de un punto en su coordenada pixel al plano superficie. Substituyendo la ecuación (2.6) en la (2.11), se obtiene

$$\boldsymbol{d}/f = \mathcal{H}^{-1}[K^{-1}\mathcal{H}[\boldsymbol{s}]]. \tag{2.14}$$

Considerando la estructura particular de la matriz de parámetros intrínsecos K (matríz triangular superior), su inversa tiene la forma

$$K^{-1} = \begin{bmatrix} A \\ \mathcal{H}[\mathbf{0}_2]^T \end{bmatrix}, \tag{2.15}$$

donde

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \end{bmatrix}. \tag{2.16}$$

Por lo tanto, la dirección del rayo de luz detectado en el pixel $\boldsymbol{s}$ se puede escribir usando las ecuaciones (2.6) y (2.7) como

$$\begin{bmatrix} \boldsymbol{d} \\ f + \delta_2\|\boldsymbol{d}\|^2 + \delta_3\|\boldsymbol{d}\|^3 + \cdots + \delta_\omega\|\boldsymbol{d}\|^\omega \end{bmatrix} = f\Lambda(\boldsymbol{s}, A, \boldsymbol{\delta}') \tag{2.17}$$

donde

$$\Lambda(\boldsymbol{s}, A, \boldsymbol{\delta}') = \begin{bmatrix} A\mathcal{H}[\boldsymbol{s}] \\ 1 + \delta_2'\|A\mathcal{H}[\boldsymbol{s}]\|^2 + \delta_3'\|A\mathcal{H}[\boldsymbol{s}]\|^3 + \cdots + \delta_\omega'\|A\mathcal{H}[\boldsymbol{s}]\|^\omega \end{bmatrix}, \tag{2.18}$$

y $\boldsymbol{\delta}'$ es un vector que contiene los parámetros de distorsión escaladas dadas como

$$\boldsymbol{\delta}' = \begin{bmatrix} \delta_2' \\ \delta_3' \\ \vdots \\ \delta_\omega' \end{bmatrix} = \begin{bmatrix} \delta_2 f \\ \delta_3 f^2 \\ \vdots \\ \delta_\omega f^{(\omega-1)} \end{bmatrix}. \tag{2.19}$$

Si los parámetros intrínsecos, extrínsecos y distorsión de la cámara están disponibles, entonces se puede calcular imágenes libres de distorsión radial utilizando las ecuaciones (2.3) y (2.17). Una vez comprendido el proceso de formación de imágenes, se busca determinar los puntos tridimensionales de la escena. En este contexto, se presentará un método de triangulación, el cual constituye un enfoque inverso, permitiendo la reconstrucción de la geometría tridimensional a partir de múltiples perspectivas.

2.3. Estimación de puntos tridimensionales

Para estimar puntos en el espacio tridimensional, se utiliza el principio conocido como triangulación. La triangulación determina la ubicación de un punto en el espacio cuando éste es observado simultáneamente desde múltiples perspectivas. En particular, un sistema cámara-proyector permite detectar puntos en el espacio desde la perspectiva de la cámara y del proyector. De esta forma, es posible aplicar el principio de triangulación y determinar la ubicación de puntos del espacio tridimensional. Las direcciones de observación son determinadas por cada dispositivo mediante el modelo pinhole. Por ejemplo, un punto $\boldsymbol{p}$ en el espacio 3D es detectado en el plano imagen como un punto $\boldsymbol{s}$ como

$$\boldsymbol{s} = \mathcal{H}^{-1}[C\mathcal{H}[\boldsymbol{p}]], \tag{2.20}$$

donde $\mathcal{H}$ es el operador de coordenadas homogéneas, y $C = K[R^T, -R^T\boldsymbol{t}]$ es la matriz que representa la cámara o proyector, e incluye los parámetros intrínsecos, K, la orientación dada por una matriz de rotación, R, y la posición dada por un vector de traslación, $\boldsymbol{t}$. Aplicando el operador inverso de las coordenadas homogéneas en la ecuación (2.20), se obtiene

$$\lambda\mathcal{H}[\boldsymbol{s}] = K[R^T\boldsymbol{p}, \quad -R^T\boldsymbol{t}], \tag{2.21}$$

donde λ es un escalar diferente de cero. Entonces, el punto $\boldsymbol{p}$ puede ser calculado como

$$\boldsymbol{p} = \boldsymbol{t} + \underbrace{\lambda RK^{-1}\mathcal{H}[\boldsymbol{s}]}_{\boldsymbol{d}}, \tag{2.22}$$

donde $\boldsymbol{d} = \lambda RK^{-1}\mathcal{H}[\boldsymbol{s}]$ es la dirección en la que el punto $\boldsymbol{p}$ fue observado, y λ es una incógnita escalar. Por lo tanto, si el punto $\boldsymbol{p}$ es observado por dos dispositivos, se obtienen un sistema de ecuaciones de la forma

$$\begin{aligned} \boldsymbol{p}_1 &= \boldsymbol{t}_1 + \lambda_1\boldsymbol{d}_1, \\ \boldsymbol{p}_2 &= \boldsymbol{t}_2 + \lambda_2\boldsymbol{d}_2. \end{aligned} \tag{2.23}$$

Como el el vector $\boldsymbol{p}$ representa al mismo punto observado por los dos dispositivos, entonces las ecuaciones anteriores se pueden igualar como

$$\boldsymbol{t}_1 + \lambda_1\boldsymbol{d}_1 = \boldsymbol{t}_2 + \lambda_2\boldsymbol{d}_2, \tag{2.24}$$

Despejando las variables incógnitas, la ecuación (2.24) es reescrita como

$$\lambda_1 \boldsymbol{d}_1 - \lambda_2 \boldsymbol{d}_2 = \boldsymbol{t}_2 - \boldsymbol{t}_1, \tag{2.25}$$

o en su forma matricial como

$$\underbrace{[\boldsymbol{d}_1, \quad -\boldsymbol{d}_2]}_{A} \underbrace{\begin{bmatrix}\lambda_1 \\ \lambda_2\end{bmatrix}}_{\Lambda} = \underbrace{\boldsymbol{t}_2 - \boldsymbol{t}_1}_{T}, \tag{2.26}$$

donde los constantes λ_1 y λ_2 son estimadas usando el método de mínimos cuadrados (o $\Lambda = (A^T A)^{-1} A^T T$). Con las constantes estimadas, se utilizan la ecuación (2.23) para determinar el punto observador $\boldsymbol{p}$ mediante

$$\boldsymbol{p} = \frac{1}{2}(\boldsymbol{p}_1 + \boldsymbol{p}_2). \tag{2.27}$$

Cabe señalar que la triangulación de puntos tridimensionales no se limita únicamente a un sistema cámara-proyector, sino que puede adaptarse a sistemas con múltiples cámaras y proyectores. En este enfoque, se consideran los puntos de correspondencia entre imágenes para determinar las coordenadas tridimensionales, lo que permite una mayor flexibilidad y precisión en la reconstrucción tridimensional. En la siguiente sección se presentarán resultados preliminares del procesamiento de una secuencia de vídeo capturada por una cámara con una lente *fisheye*, utilizando el modelo de cámara pinhole con distorsión radial analizado.

2.4. Validación del modelo de cámara pinhole con distorsión radial

Se llevaron a cabo tres experimentos distintos para validar el modelo propuesto para la cámara pinhole con distorsión radial. En el primer experimento, se calibró una cámara para obtener sus parámetros intrínsecos, extrínsecos, y distorsión. En el segundo experimento, se determinó la posición de los objetos en la imagen utilizando los parámetros obtenidos de la cámara para corregir la distorsión inicial. En el tercer experimento, se determinó la posición tridimensional de un vehículo terrestre en la escena utilizando el método de triangulación.

2.4.1. Calibración de cámara con distorsión radial

En este experimento se empleó un tablero de ajedrez de 7×10 cuadros, donde el lado de cada cuadro medía 23,7 mm. Este patrón de calibrción facilitó la detección de los puntos de correspondencia ($\boldsymbol{\rho}$, $\boldsymbol{s}$) necesarios para llevar a cabo la calibración de la cámara [24]. Se adquirieron 52 imágenes utilizando una cámara fisheye, como se muestra en la figura 2.4, las cuales fueron procesadas para determinar sus respectivas homografías. Estas homografías resultan fundamentales para obtener la calibración inicial mediante la aplicación del modelo

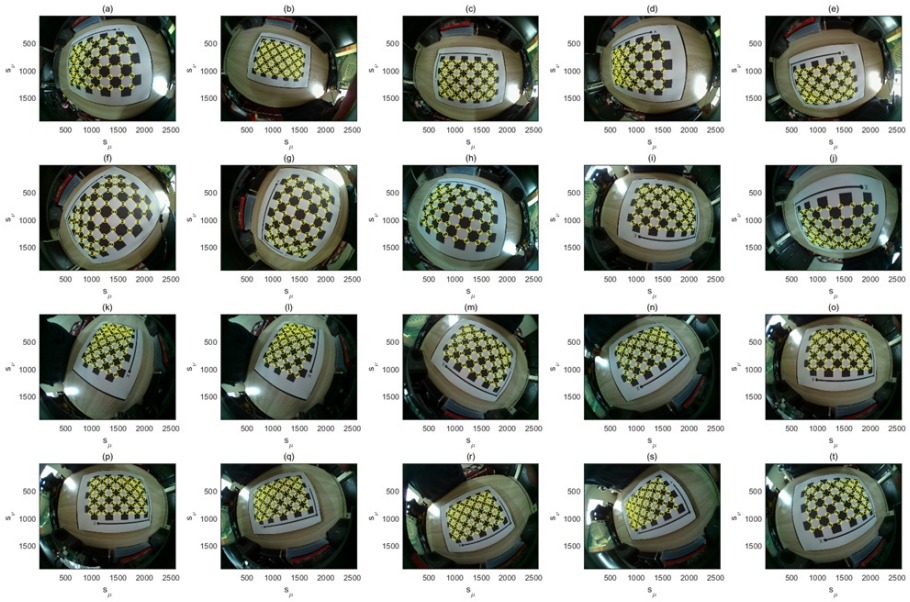

Figura 2.4: Imágenes de entrada (20 de 52) para detectar los puntos $\boldsymbol{s}_{i,j}$ del patrón de calibración.

de la cámara pinhole [20]. Los resultados de esta calibración se utilizan como entrada para la calibración del modelo de cámara pinhole con distorsión radial, utilizando el método de Gauss-Newton formulado por mínimos cuadrados [24]. Los parámetros intrínsecos, extrínsecos y de distorsión obtenidos fueron:

Cámara 1										
K				R				t		d
2.7090	-0.0032	0.2210		-0.7777	-0.3507	0.5217		-1.41E+03		-0.3475
0	2.7040	0.3318		-0.6261	0.5069	-0.5925		2.34E+03		0.3818
0	0	1		-0.0567	-0.7875	-0.6137		1.16E+03		

Cámara 2										
K				R				t		d
0.9441	0.0012	0.0347		0.8308	0.4221	-0.3628		8.72E+02		0.1736
0	0.9459	-0.0508		0.5462	-0.4929	0.6773		-4.35E+02		-0.1562
0	0	1		0.1071	-0.7608	-0.64003		1.19E+03		

Para verificar la capacidad de eliminar la distorsión radial en imágenes, se usaron los parámetros obtenidos en el proceso de calibración para procesar una secuencia de vídeo, como se ilustra en las figuras 2.5(a)-(d). Mediante el estudio del proceso de formación de imágenes, se corrigió la distorsión radial, como se muestra en las figuras 2.5(e)-(h).

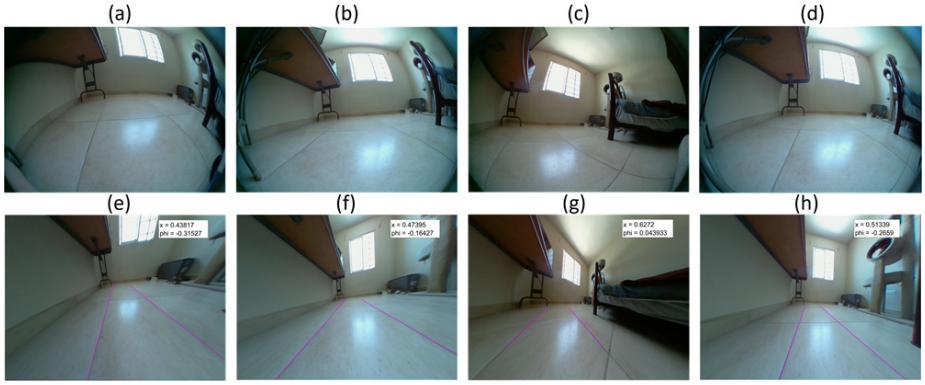

Figura 2.5: (a)-(d) Imágenes de entrada capturadas con una cámara fisheye. (e)-(h) Imágenes sin distorsión radial obtenidas usando el modelo de cámara pinhole con distorsión radial. Las imágenes sin distorsión permiten simplificar el proceso de detección de líneas de carril usando líneas rectas.

2.4.2. Detección de objetos

Adicionalmente, se empleó la secuencia de vídeo obtenida con la cámara fisheye para detectar objetos en movimiento mediante el uso del flujo óptico, como se muestra en las figuras 2.6(a)-(d). Posteriormente, se generaron máscaras binarias identificando las regiones con velocidades elevadas, como se muestra en las figuras 2.6(e)-(h). Las máscaras obtenidas se usaron para identificar objetos en movimiento en la escena. Finalmente, se procede a la identificación de los objetos en función de la cantidad de píxeles concentrados dentro de una ventana predefinida (se consideran al menos 300 píxeles para su clasificación como objeto), como se muestra en las figuras 2.6(i)-(l).

2.4.3. Estimación de posición tridimensional

En este experimento, se somete al vehículo terrestre a una prueba de navegación simple. En esta primera prueba, el vehículo debe desplazarse en la escena a lo largo de una línea recta. El sistema de visión captura la escena desde dos puntos de observación diferentes, como se ilustra en la figura 2.7. Las cámaras se calibran previamente para obtener los parámetros intrínsecos, extrínsecos y de distorsión. A través de la detección de objetos mediante un filtro de correlación, se localiza la posición del vehículo. La primera medición se muestra en la figura 2.8 , y se utiliza para determinar su posición tridimensional con el sistema de visión. Luego, los puntos detectados por cada dispositivo se procesan mediante el método de triangulación para calcular su posición real en el espacio tridimensional. La trayectoria recorrida por el vehículo se representa en la figura 2.9.

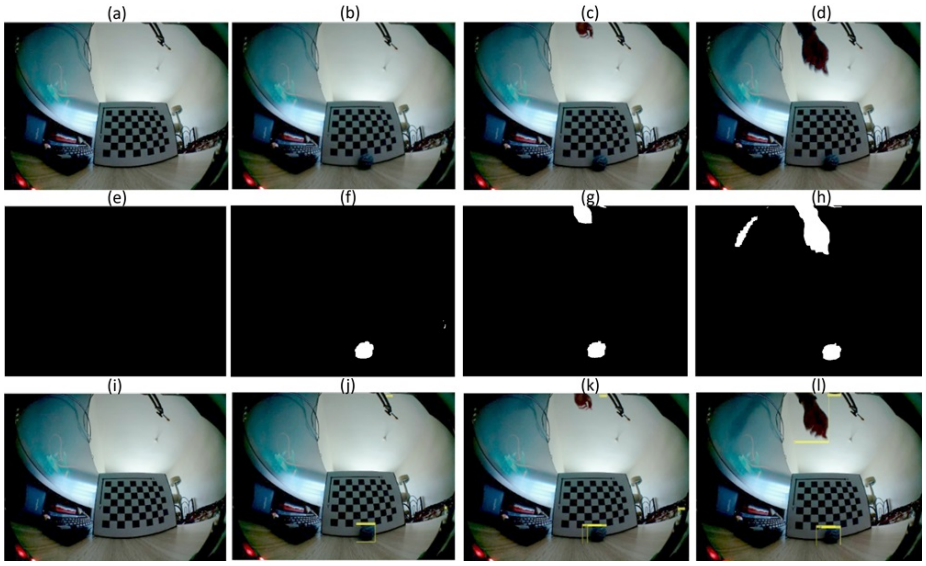

Figura 2.6: Detección de objetos usando flujo óptico. (a)-(d) Imágenes de entrada con objetos en movimiento. El flujo óptico se estimó usando el método de Horn-Schunck. (e)-(h) Máscaras binarias obtenidas al umbralizar los niveles de flujo óptio estimados. (i)-(l) Objetos en movimiento detectados en la escena.

Figura 2.7: Escena experimental para el vehículo terrestre.

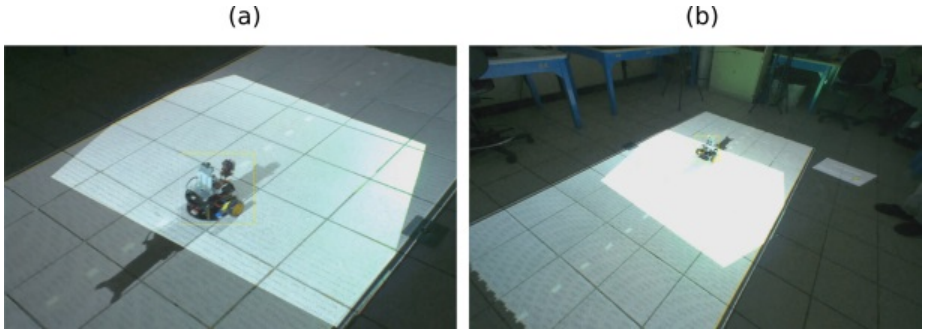

Figura 2.8: Detecciones del robot móvil terrestre usando filtro de correlación en ambos dispositivos.

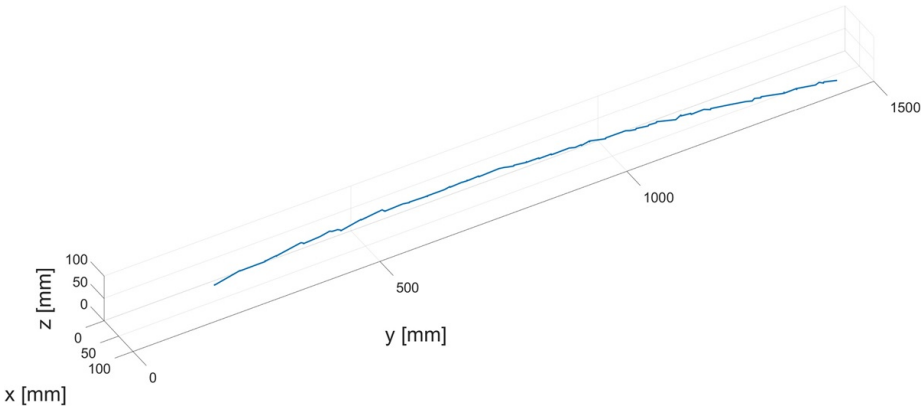

Figura 2.9: Resultado de la estimación de posición en el espacio tridimensional mediante el método de la triangulación.

En este capítulo, se presentó un algoritmo para determinar la posición de un vehículo terrestre en el espacio. Se analizó el sistema de formación de imágenes de la cámara usando el modelo pinhole con distorsión radial. El modelo empleado permitió incluir el efecto de distorsión inherente a los lentes ópticos de campo visual amplio (fisheye). Este enfoque fue útil para remover la distorsión radial y simplificar considerablemente las tareas de detección de objetos y estimación de posición.

Se probaron distintos métodos de detección de objetos, incluyendo detección por color, flujo óptico, y filtros de correlación. Cada método fue evaluado para determinar ventajas, desventajas, y simplicidad de implementación. Posteriormente, la detección del vehículo a partir de cada cámara del sistema permitió calcular la posición del vehículo en el espacio tridimensional usando triangulación. Es importante destacar que este método no abordó la estimación de orientación del vehículo. En el próximo capítulo, se presentará un método para estimar la pose del vehículo utilizando información tridimensional de la escena.

Capítulo 3

Estimación de pose usando información 3D

En este capítulo se aborda un enfoque que permite determinar la pose de la cámara mediante la teoría de la geometría epipolar. Previo a ello, se revisarán algunos conceptos fundamentales de la geometría epipolar y principios asociados considerando el modelo de cámara pinhole [19].

3.1. Geometría epipolar

En una configuración de sistema estéreo, las cámaras pueden ser diferentes y estar desalineadas. El punto de intersección entre la línea que une los centros de proyección de las cámaras y el plano imagen se denomina epípolo. Las líneas que pasan por el epípolo se conocen como líneas epipolares. Si $\boldsymbol{e}$ y $\boldsymbol{e'}$ son los epípolos en el plano imagen de la primera y segunda cámara, respectivamente, entonces las líneas epipolares son

$$\boldsymbol{\ell} = \mathcal{H}[\boldsymbol{e}] \times \mathcal{H}[\boldsymbol{x}], \tag{3.1}$$

$$\boldsymbol{\ell}' = \mathcal{H}[\boldsymbol{e'}] \times \mathcal{H}[\boldsymbol{x}'], \tag{3.2}$$

donde $\boldsymbol{x}$ y $\boldsymbol{x'}$ son puntos de correspondencia en la imagen de la primera y segunda cámara, respectivamente. La relación entre ambos puntos se puede definir como una transformación proyectiva G entre cámaras, dado que

$$\boldsymbol{x}' = \mathcal{H}^{-1}[G\mathcal{H}[\boldsymbol{x}]]. \tag{3.3}$$

Al sustituir la ecuación (3.1) en la ecuación (3.3), se obtiene

$$\boldsymbol{\ell}' = \mathcal{H}[\boldsymbol{e'}] \times G\mathcal{H}[\boldsymbol{x}]. \tag{3.4}$$

El producto vectorial se puede reescribir como una matriz antisimétrica $\mathcal{H}[\boldsymbol{e}']_\times$ como

$$\boldsymbol{\ell}' = \underbrace{\mathcal{H}[\boldsymbol{e}']_\times G}_{F}\mathcal{H}[\boldsymbol{x}], \tag{3.5}$$

donde $F = \mathcal{H}[\boldsymbol{e}']_\times G$ es la matriz que relaciona a los planos de la cámara mediante sus epípolos. La matriz F es conocida como *matriz fundamental* y permite relacionar los puntos del plano izquierdo con líneas que intersecta al epípolo del plano derecho o viceversa. Por lo tanto, se puede simplificar la estimación de $\boldsymbol{e}'$ al asumir que la pose de la primera cámara se encuentra en el origen. El epípolo $\boldsymbol{e}'$ se determina como

$$\boldsymbol{e}' = K'\boldsymbol{t}, \tag{3.6}$$

donde K' es la matriz de los parámetros intrínsecos de la segunda cámara, y $\boldsymbol{t}$ es la posición de la segunda cámara respecto al sistema de coordenadas global. La homografía del sistema se puede representar en su forma explícita en términos de dos cámaras como

$$G = G'G^+ = K'RK^{-1}. \tag{3.7}$$

Combinando las ecuaciones (3.6) y (3.7) en la definición de la matriz fundamental, se obtiene

$$F = [K'\boldsymbol{t}]_\times K'RK^{-1} = K'^{-T}[\boldsymbol{t}]_\times RK^{-1}. \tag{3.8}$$

Entonces, si se conocen los parámetros intrínsecos de las cámaras, obtenidos en calibraciones previas, la matriz fundamental se puede simplificar como

$$E = K'^{-T}FK^{-1} = [\boldsymbol{t}]_\times R, \tag{3.9}$$

donde E es conocida como *matriz esencial*. La matriz esencial encapsula la relación geométrica entre dos vistas de una escena tridimensional y se puede descomponer para recuperar los parámetros extrínsecos de la segunda cámara mediante descomposición en valores singulares. Una vez que se han recuperado los parámetros de la cámara a partir de la matriz esencial, se aplica el método de triangulación para determinar los puntos tridimensionales correspondientes para cada par de imágenes. Es importante considerar que los puntos tridimensionales estimados pueden estar afectadas por ruido introducido por errores de detección o desfase de puntos. Para mitigar este problema, se busca minimizar el error de reproyección de puntos tridimensionales para cada pose estimada de la cámara. Este proceso de refinamiento es conocido en la literatura como ajuste conjunto (en inglés: *bundle adjustment*) [25–28]. El ajuste conjunto es abordado en detalle en la siguiente sección.

3.2. Ajuste conjunto

El ajuste conjunto busca minimizar el error de reproyección de puntos que se encuentran en distintas vistas. Para esto, se define la re-proyección de un punto como

$$\boldsymbol{\mu}_j^i = \mathcal{H}^{-1}[C^i\mathcal{H}[\boldsymbol{P}_j]], \tag{3.10}$$

donde $\boldsymbol{\mu}_j^i$ indica el j-ésimo punto de correspondencia de la imagen I_i, C^i es la matriz de la cámara que relaciona la imagen I, y el punto $\boldsymbol{P}_j$ de la escena. Una medición convencional puede no satisfacer esa relación debido a las perturbaciones externas experimentadas durante el proceso de capturar de imágenes. Este problema produce errores de reproyección, es decir los puntos detectados en la imagen no coinciden con los puntos estimados a partir de la reproyección. Por lo tanto, se debe ajustar la matriz de la cámara y los puntos tridimensional de tal forma que los errores de reproyección se minimicen como

$$\text{mín} \sum_{ij} d(\hat{\boldsymbol{\mu}}_j^i, \boldsymbol{\mu}_j^i)^2, \tag{3.11}$$

donde $d(\cdot, \cdot)$ es la distancia geométrica entre dos puntos en el plano de la imagen. Existen varias maneras de resolver este problema de minimización, incluyendo métodos iterativos, mínimos cuadrados no lineas, y Levenberg-Marquardt, entre otras. En la siguiente sección se describen dos métricas de error usadas para evaluar la metodología propuesta en este trabajo de tesis.

3.3. Métricas de error

En este trabajo de tesis, se emplearán dos métricas de error específicas para evaluar los resultados obtenidos del algoritmo de estimación de pose usando información tridimensional. Estas métricas fueron elegidas debido a su habilidad para evaluar con precisión tanto la calidad como la precisión de la estimación de la pose.

3.3.1. Raíz del error cuadrático medio

La raíz del error cuadrático medio (RMSE, por las siglas en inglés: *Root Mean Square Error*) es una métrica utilizada para evaluar la precisión de un modelo de regresión. El método cuantifica las predicciones del modelo contra los valores reales o experimentales obtenidos del sistema. En este caso, cuando el valor RMSE es bajo, se considera que la capacidad predicativa del modelo es buena. La ecuación que caracteriza la raíz del error cuadrático medio es

$$E_{\text{RMSE}} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}, \tag{3.12}$$

donde n es el número total de observaciones, y_i es el valor real o experimental obtenido en la i-ésima observación en el sistema, y $\hat{y}_i$ es el valor predicho por el modelo. A diferencia de otras métricas de error, el RMSE es sensible a grandes desviaciones puesto que los errores son ponderados cuadráticamente.

3.3.2. Error absoluto medio

El error absoluto medio (MAE, por las siglas en inglés: *Mean Absolute Error*) es una métrica de evaluación de regresión que proporciona una medida simple

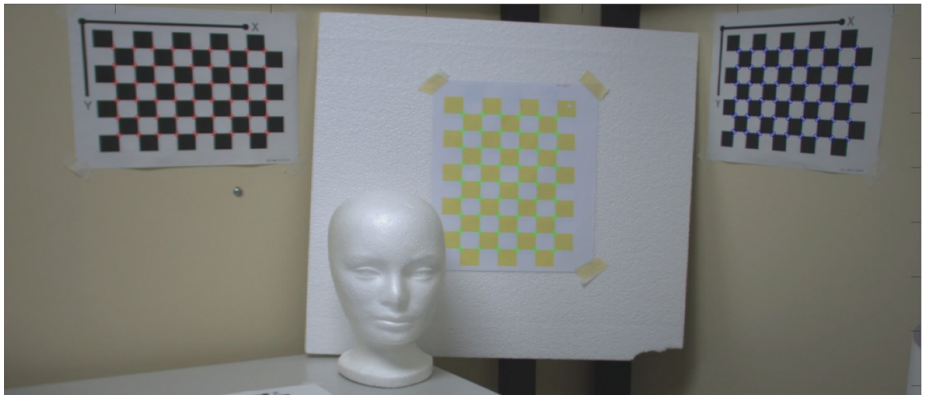

Figura 3.1: Escena experimental para estimar la pose de la cámara sujeta a desplazamiento lineal.

y fácil de interpretar. Esta métrica está definida como la magnitud promedio de los errores absolutos entre las predicciones del modelo y los valores reales; matemáticamente,

$$E_{\text{MAE}} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}. \tag{3.13}$$

A diferencia del RMSE, el MAE no penaliza severamente (cuadráticamente) los errores, sino de una forma lineal. Por esta razón, esta métrica es menos sensibles a grandes desviaciones de error. En la siguiente sección, se evaluará el método de estimación de pose propuesto, evaluando su desempeño en términos de las métricas RMSE y MAE.

3.4. Validación de estimación de pose

En esta sección, se llevará a cabo la validación del algoritmo propuesto para la estimación de la pose utilizando tanto datos simulados como experimentales. La evaluación se realizará mediante la comparación de los resultados obtenidos con las métricas de error analizadas previamente. Diferentes escenarios serán considerados para evaluar la eficiencia del método propuesto.

3.4.1. Estimación de movimiento lineal

La primera evaluación se realiza usando una escena con puntos conocidos definidos por patrones de calibración. Se registra la distancia de desplazamiento de una cámara utilizando la información tridimensional de la escena. Inicialmente, se estima la pose de una cámara y posteriormente es desplazada linealmente 33 cm a lo largo del eje z. El desplazamiento se efectúo mediante un tripié que permite cambiar la posición de la cámara de manera lineal. La figura 3.1 muestra

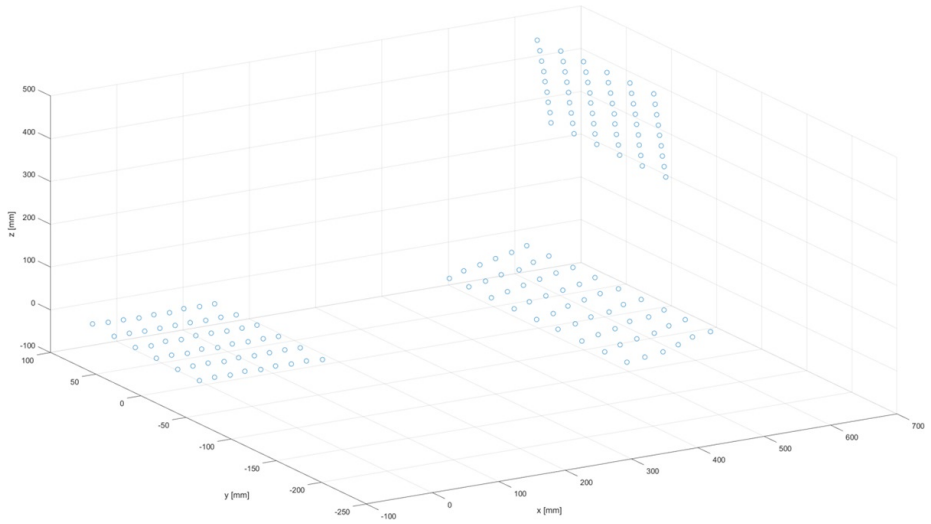

Figura 3.2: Resultado de la triangulación de puntos de los patrones de calibración en una escena.

la escena inicial con los patrones de calibración usados para detectar los puntos de interés. Los puntos fueron detectados usando las dimensiones reales de los patrones y fueron procesados para determinar su posición tridimensional [29]. En la figura 3.2 se presenta el resultado de la triangulación de los puntos detectados. La información tridimensional fue empleada para estimar la pose de la cámara obteniendo como resultado

$$R_1 = \begin{bmatrix} 0{,}8786 & -0{,}1442 & 0{,}4552 \\ -0{,}0646 & -0{,}9804 & -0{,}1859 \\ 0{,}4731 & 0{,}1339 & -0{,}8708 \end{bmatrix}, \quad \boldsymbol{t}_1 = \begin{bmatrix} -375{,}4 \\ 36{,}1 \\ 1322 \end{bmatrix}. \tag{3.14}$$

Posteriormente, la cámara se desplazó linealmente con ayuda del tripié y se capturó una segunda imágen, como se muestra en la figura 3.3. De la misma manera, los puntos de los patrones de calibración fueron detectados y procesados para estimar la posición tridimensional de la cámara, obteniendo

$$R_2 = \begin{bmatrix} 0{,}8773 & -0{,}0953 & 0{,}4704 \\ 0{,}0109 & -0{,}9759 & -0{,}2181 \\ 0{,}4798 & 0{,}1964 & -0{,}8551 \end{bmatrix}, \quad \boldsymbol{t}_2 = \begin{bmatrix} -561{,}4 \\ 48{,}5 \\ 1654{,}8 \end{bmatrix}. \tag{3.15}$$

Las orientaciones obtenidas son muy similares debido a que no se realizaron cambios de ángulo visual en la cámara. Por otro lado, las posiciones $\boldsymbol{t}_1$ y $\boldsymbol{t}_2$ exhiben correctamente el cambio de posición de la cámara de acuerdo con el desplazamiento realizado a lo largo del eje z. El desplazamiento estimado entre

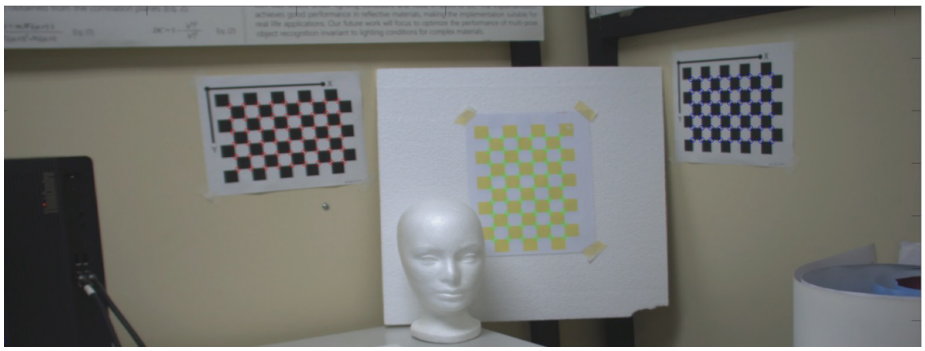

Figura 3.3: Escena capturada con un desplazamiento a lo largo del eje z para estimar la pose de la cámara y su distancia de traslación.

la posición inicial y final de la cámara se calculó como

$$\boldsymbol{d} = \boldsymbol{t}_2 - \boldsymbol{t}_1 = \begin{bmatrix} 186{,}0 \\ -12{,}4 \\ -332{,}8 \end{bmatrix} mm. \tag{3.16}$$

Observe que la diferencia a lo largo del eje z, ($-332{,}8$ mm), es consistente con el desplazamiento de 33 cm introducido usando el tripié de la cámara (error de 2,8 mm). Estos resultados muestran una aproximación suficiente para emplear el método de estimación de pose propuesto en tareas de navegación de un robot móvil terrestre.

3.4.2. Reconstrucción tridimensional

En este experimento, se realizó una reconstrucción tridimensional proyectiva a través de una secuencia de imágenes capturada por una cámara calibrada. Primero, se detecta los puntos característicos de la escena usando métodos como Características Robustas Aceleradas (SURF), Transformación de Características Invariante a Escala (SIFT), Puntos Claves Robusto-Binarios Invariante a Escala (BRISK), entre otros [30–33]. Las características sirven para determinar la correspondencia de puntos entre imágenes como se muestra en la figura 3.4. La correspondencia de puntos obtenida fue utilizada para estimar la matriz fundamental y extraer la pose de la cámara. La pose inicial estimada fue utilizada para la triangulación de los puntos característicos. En la figura 3.5, se muestra la trayectoria de la cámara y el resultado de la escena construida. Finalmente, se realiza un post-procesamiento para minimizar el error de re-proyección de los puntos estimados usando el ajuste conjunto. Adicionalmente, se calcula el valor de intensidad del pixel de los puntos obteniendo como resultado la figura 3.6.

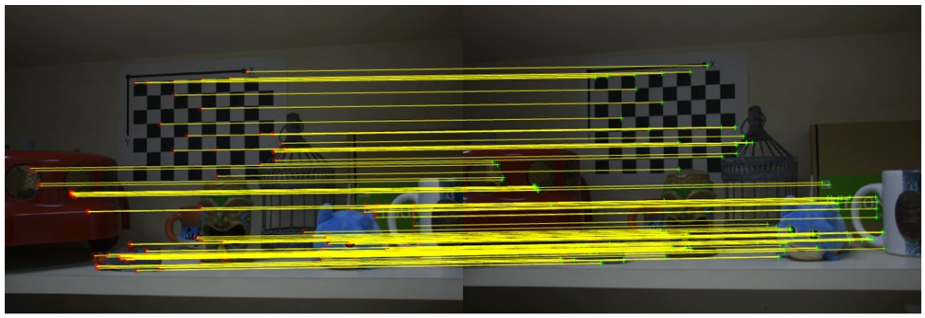

Figura 3.4: Imagen izquierda y derecha capturadas con una cámara en posición inicial y final. Las líneas conectan los puntos característicos detectados.

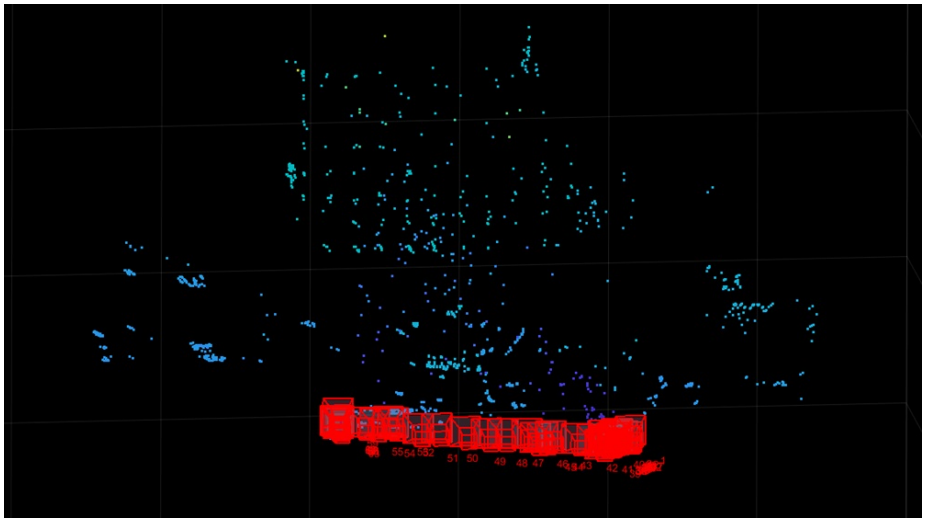

Figura 3.5: Escena reconstruida mediante una secuencia de imágenes capturadas por una cámara.

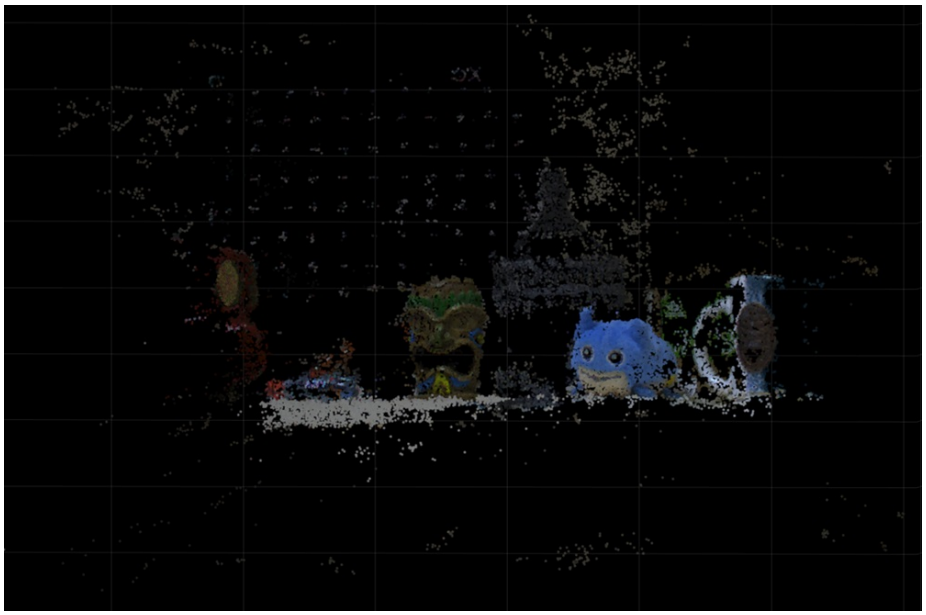

Figura 3.6: Resultado obtenido mediante ajuste conjunto en la reconstrucción tridimensional.

3.4.3. Evaluación de error de estimación de pose

En esta sección se presentan los resultados de estimación de pose obtenidos procesado la secuencia de video capturada en dos pruebas. En ambas pruebas la cámara realiza un recorrido dentro de la escena. En los dos experimentos se emplean cámaras calibradas; es decir, se conocen las matrices de parámetros intrínsecos. Las imágenes de cada video son procesadas para detectar puntos característicos y estimar la pose de la cámara.

En la primera prueba, la cámara realiza una trayectoria simple, que consistió en una traslación a lo largo del eje horizontal. El propósito de este experimento fue verificar el funcionamiento y rendimiento del algoritmo propuesto. En la segunda prueba, se usaron las imágenes de la base de datos *New Tsukaba*, debido a que proporciona la trayectoria exacta realizada por la cámara durante su nagevación dentro de la escena [34, 35]. Esta última prueba fue útil para disponer la trayectoria de referencia y así evaluar el rendimiento del algoritmo de estimación de pose. A continuación, se describe el algoritmo para estimar la pose de la cámara usando la información tridimensional.

En la primera etapa del experimento, se inicializa el sistema asignando la pose inicial de la cámara en la primera imagen. La segunda pose de la cámara se calculada usando la segunda imagen y estimando la geometría epipolar [19, 20]. Para esto, fue necesario detectar los puntos característicos de ambas imágenes

Figura 3.7: Escenas de prueba usadas para estimar la pose de la cámara. (a) Primera y última imagen de la secuencia de una trayectoria simple. (b) Primera y última imagen de la secuencia de una trayectoria conocida usando la base de datos *New Tsukaba*.

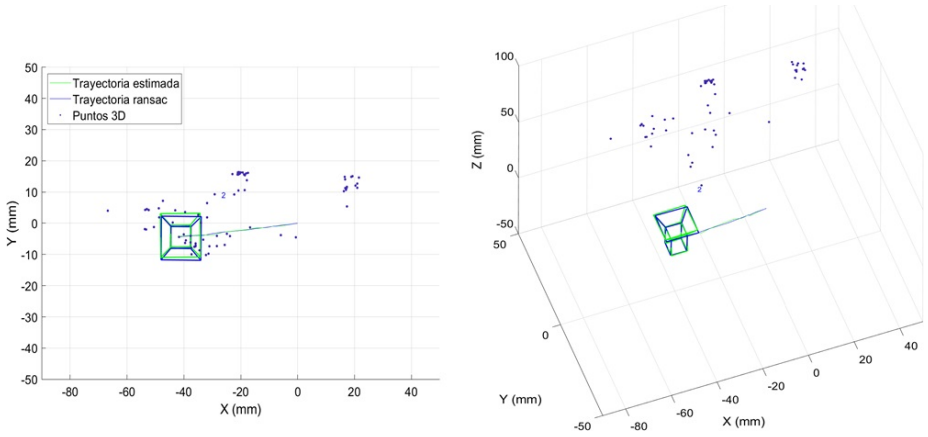

Figura 3.8: Resultados obtenidos mediante la estimación de pose usando la información tridimensional y estimación de pose usando consenso de muestra aleatoria (Random Sample Consensus o RANSAC, por sus siglas en inglés). La línea verde corresponde a la trayectoria estimada y la línea azul corresponde la trayectoria de referencia.

usando el método de Características Robustas Aceleradas (SURF, por las siglas en inglés: *Speeded-Up Robust Features*), aunque otros métodos también pueden ser usados [30–33]. Posteriormente, se determinan los puntos de correspondencia, por medio de algoritmos de comparación y se calcula la matriz esencial quien contiene la segunda pose de la cámara [36].

En la segunda etapa del experimento, se captura la siguiente imagen de la secuencia de video y se obtienen nuevos puntos característicos. Los puntos nuevos son comparados con los puntos característicos de la imagen previa en la secuencia de video para determinar correspondencias. Como resultado, se obtiene un nuevo par estéreo con una nueva geometría epipolar, que permite estimar la nueva pose de la cámara. Adicionalmente, se determinan los puntos 3D obteniendo una nube de puntos 3D y sus correspondientes puntos imagen (2D) en la cámara para la pose actual.

En la tercer etapa del experimento, se eliminan puntos de observación que tengan un error de reproyección mayor que un umbral prefijado. Después, se calcula la pose de la cámara mediante el método propuesto usando la información tridimensional obtenida. Las etapas dos y tres se repiten hasta alcanzar la última imagen de la secuencia de video del experimento.

Por último, se refinó el ajuste minimizando el error de reproyección y pose de cada vista observada [25–28]. Los resultados de cada experimento se puede observar en las figuras 3.8 y 3.9. En la figura 3.8 se observa una trayectoria simple que permitió verificar el funcionamiento correcto del método propuesto. Además, se implementaron métodos alternativos para realizar comparaciones

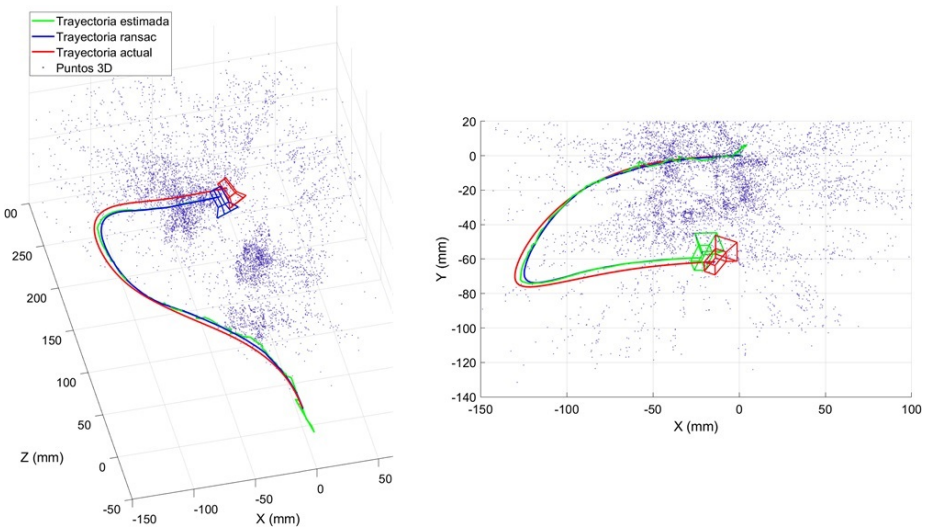

Figura 3.9: Distintas trayectorias estimadas fueron evaluadas para determinar la precisión y robustez de los algoritmos implementados. La línea verde corresponde a la trayectoria estimada, la línea azul corresponde la trayectoria aplicando RANSAC, y la línea roja corresponde a la trayectoria real.

de resultados [37, 38]. En la figura 3.9, se puede observar tres trayectorias. La primera es la trayectoria estimada marcada con color verde, la segunda es la trayectoria de comparación marcada con color azul, y la tercera es la trayectoria real marcada con color rojo. Haciendo uso de las métricas de error descrita en la Sección 3.3, se obtuvieron los niveles de error mostrados en la Tabla 3.1.

Los resultados obtenidos muestran que el método propuesto no logra altos niveles de precisión en cuanto a la estimación de posición. Por otro lado, la precisión en cuanto a la estimación de la orientación es aceptable para tareas de navegación planteadas en esta tesis. En resumen, el método propuesto es funcional debido a los resultados preliminares obtenidos, aunque se debe trabajar en mejorar la precisión de las estimaciones de posición.

Método	**Posición**			
	RSME	**MAE**	**Media**	**STD**
Propuesto	11.8942	8.5342	3.4066	11.2154
RANSAC	8.9516	7.1590	3.2805	8.1240
	Orientación			
Propuesto	1.1070	1.5434	0.2518	1.0742
RANSAC	0.6248	1.0843	0.1400	0.6023

Cuadro 3.1: Errores de estimación de pose de la cámara de la trayectoria conocida.

Capítulo 4

Plataformas experimentales

4.1. Sistema multiproyector de escenas dinámicas

Los sistemas de navegación terrestres se están volviendo cada vez más importantes en los últimos años [39–43]. La seguridad del usuario es un factor primordial en las aplicaciones de navegación. La falla de un sistema de navegación pueden tener consecuencias inaceptables, desde daños materiales hasta lesiones y pérdidas humanas [44–46]. Por lo tanto, la corrección y efectividad de los algoritmos son de vital importancia [47].

La validación de algoritmos requiere evaluación exhaustiva bajo una amplia variedad de posibles escenarios de casos reales para minimizar el riesgo de fallos. Diferentes estrategias de prueba han mostrado resultados aceptables en la evaluación de algoritmos de navegación, tales como pruebas en plataformas físicas, simuladores, y realidad virtual [48–51]. Sin embargo, estas estrategias no son prácticas en todos los casos. Las plataformas físicas son costosas y la evaluación exhaustiva requiere enormes recursos materiales, tiempo, y espacio. Por otro lado, los simuladores y la realidad virtual tienen un gran número de situaciones de prueba, pero están desconectados de la operatividad del vehículo físico.

En este trabajo, se propone la construcción de un sistema multiproyector para la generación de dinámica de escenas. El sistema multiproyector propuesto permite realizar pruebas de navegación vehicular en una gran cantidad de escenas con bajos costos de producción y de tiempo de preparación. Este sistema propuesto emplea cuatro proyectores para desplegar dinámicamente diferentes pistas o escenarios, diseñados para evaluar diferentes aspectos de navegación.

El sistema multiproyector propuesto fragmenta la escena a desplegar y compensa la distorsión generada por el ángulo de proyección de cada proyector. Esto se realiza considerando al proyector una “cámara inversa” que transforma de la imagen que se desea desplegar (escena) en una proyección sobre el plano de referencia (fragmento de imagen distorsionado). Este proceso requiere cono-

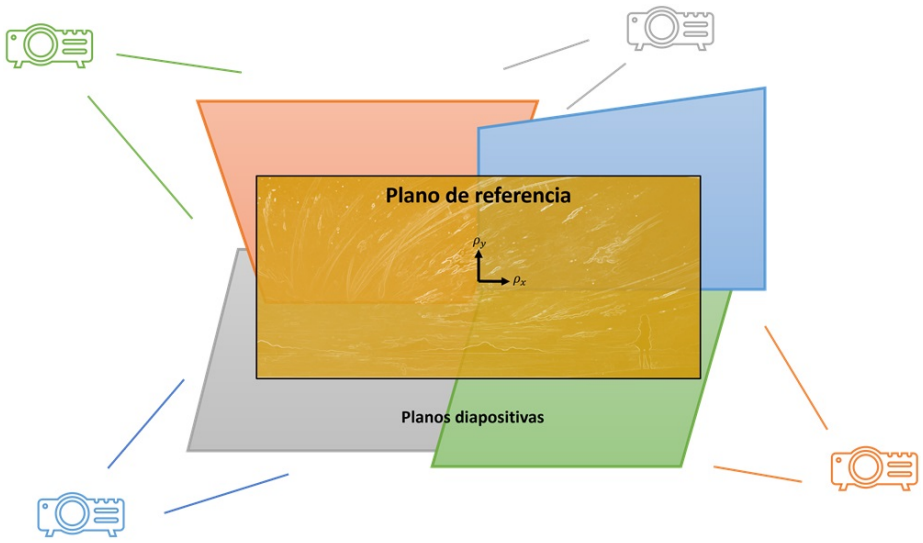

Figura 4.1: Generación dinámica de escenas usando un sistema multiproyector.

cer la posición y orientación entre el plano de referencia y el plano diapositiva del proyector, lo cual se puede describir matemáticamente mediante una matriz de homografía. Existen distintas metodologías para calcular dicha homografía. Por simplicidad, se utilizarán cuatro puntos de correspondencia entre el plano imagen y el proyector. Sin embargo, posteriormente se puede adaptar para un número arbitrario de puntos de correspondencia.

4.1.1. Generación de escenas usando multiproyección

La creación de escenas usando multiproyección (también conocido como imágenes mosaico) es una técnica de procesamiento de imágenes que genera una única imagen superponiendo múltiples fragmentos de la imagen total [52,53]. En un sistema multiproyector, los planos diapositiva se superponen para crear una única imagen de mosaico, como se muestra en la figura 4.1. Además, los proyectores deben pre-distorsionar apropiadamente el fragmento de imagen asociado para compensar la distorsión que introduce el ángulo de proyección. Para esto, es necesario determinar la relación entre los planos de diapositivas y el plano de referencia. En la figura 4.2 se muestra que la homografía relaciona un solo plano de diapositivas y el plano de referencia. Con este enfoque, cada plano diapositiva requiere una homografía para construir una imagen mosaico coherente. Sin embargo, los planos diapositiva son desconocidos debido a la posición y orientación de cada proyector del sistema. Sin embargo, es posible estimar las homografías necesarias usando los puntos esquina del plano diapositiva y relacionarlos con

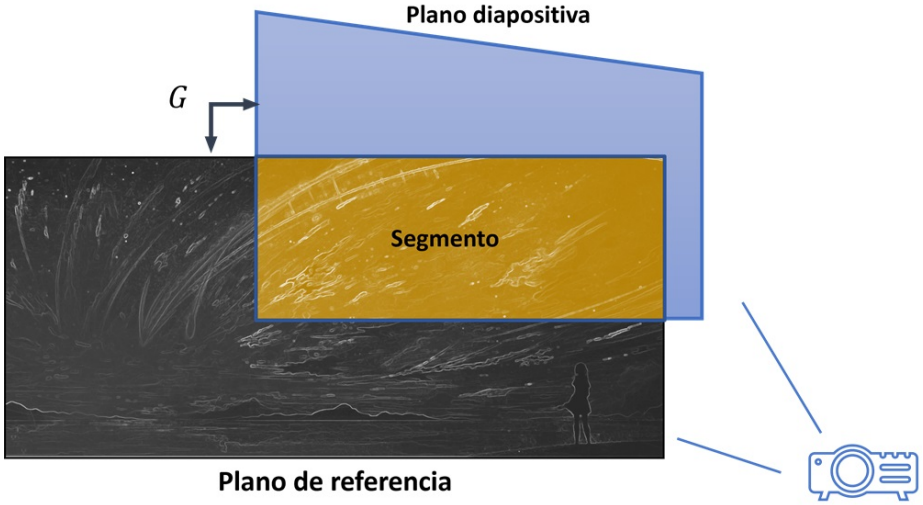

Figura 4.2: Proyector desplegando un segmento de imagen en el plano de referencia. La relación entre el plano diapositiva y el plano de referencia está dado por una matriz homografía G.

los puntos esquina del plano de referencia como se describe a continuación.

Primero, el plano diapositiva se visualiza usando una imagen auxiliar arbitraria y establecer una nueva relación con puntos conocidos del plano de referencia. Por ejemplo, la figura 4.3(a) muestra los puntos de selección en la imagen y su correspondencia es conocida porque el plano de referencia fue establecido previamente. El método de estimación usado en este trabajo se describe en el Apéndice A. Una vez que se conoce la homografía entre el plano diapositiva y el plano de referencia, se despliega la imagen de interés por re-proyección de puntos usando la homografía inversa como

$$\boldsymbol{\rho} = \mathcal{H}^{-1}[G^{-1}\mathcal{H}[\boldsymbol{\mu}]]. \tag{4.1}$$

Segundo, la re-proyección de la imagen recupera la posición del plano de diapositivas en el plano de referencia y se selecciona en sentido contrario a las agujas del reloj desde la esquina superior derecha, como se muestra en la fig. 4.3(b). Los puntos seleccionados son esquinas del plano de diapositivas en coordenadas del plano de referencia y sus puntos de correspondencia son

$$\boldsymbol{\mu}_1 = \begin{bmatrix} N \\ 1 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} 1 \\ M \end{bmatrix}, \quad \boldsymbol{\mu}_4 = \begin{bmatrix} N \\ M \end{bmatrix}, \tag{4.2}$$

donde M y N son la altura y el ancho en pixeles del plano diapositiva del proyector. De esta forma, la homografía que relaciona el plano de diapositivas y el plano de referencia se puede determinar. Tercero, dado la homografía

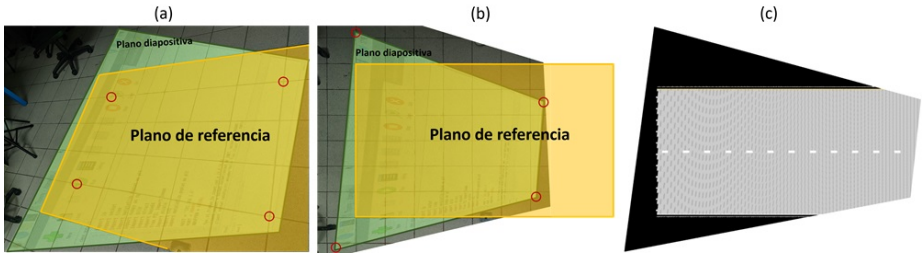

Figura 4.3: Algoritmo propuesto para mostrar una imagen mosaico coherente en el plano de diapositivas. (a) Se utiliza una imagen de referencia para determinar la relación entre los planos. (b) La imagen de referencia se vuelve a proyectar utilizando la homografía determinada para encontrar las coordenadas de la diapositiva en el plano de referencia. (c) Resultados de la imagen correspondiente a mostrar en el proyector.

se puede recuperar todas las coordenadas del plano de diapositivas aplicando la re-proyección usando la ecuación (4.1). Luego, se aplica una interpolación de imagen dividiendo el plano de referencia que corresponde al plano de diapositivas [54]. La figura 4.3(c) muestra la imagen resultante que el proyector desplegará. Finalmente, el proceso se repetirá para todos los proyectores del sistema obteniendo una superposición coherente de todos los planos de diapositiva. En la próxima subsección, se presentan los resultados experimentales para la plataforma digital de navegación de vehículos.

4.1.2. Validación de formación de imágenes dinámicas

La utilidad del método propuesto se verificó mediante la evaluación de un algoritmo de control de movimiento simple para el movimiento de un robot con ruedas. Para este experimento, la pista de prueba se generó utilizaron cuatro proyectores colocados en diferentes posiciones y orientaciones de tal forma que los planos diapositiva illuminaran el campo de prueba, como se muestra en la figura 4.4(a). Posteriormente, se estimaron las cuatro matrices homografía asociadas a cada proyector del sistema usando los puntos esquina de cada plano diapositiva y el plano de referencia. Después, se usó la inversa de cada homografía estimada para generar los fragmentos de imagen pre-distorsionados correspondientes a cada proyector. Finalmente, los fragmentos de imagen resultantes se enviaron hacia cada proyector, como se muestra en la figura 4.4(b).

La plataforma multiproyector construida se empleó para evaluar la detección de la posición del robot móvil, como se muestra en la figura 4.5(a). Se configuraron las cámaras para capturar la navegación del vehículo en la pista creada. La secuencia capturada se procesó para detectar las posiciones del vehículo utilizando filtros de correlación [55]. En sistemas opto-digital, la posición detectada en coordenadas píxel puede usarse para determinar la posición real del objeto

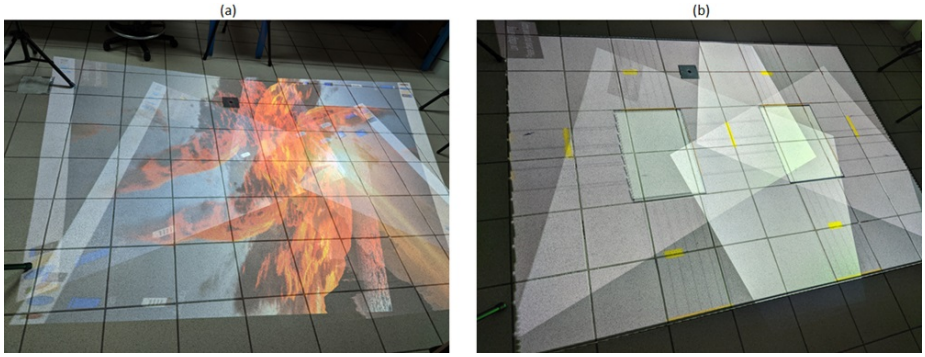

Figura 4.4: (a) Cuatro proyectores iluminando el campo de prueba. (b) Imagen mosaico construido por la superposición coherente de los planos diapositiva.

en el espacio tridimensional mediante triangulación. La figura 4.5(b) muestra la trayectoria hecha por el robot en el espacio tridimensional. De esta forma, los resultados mostraron que el método propuesto no interfiere con el método de detección de posición tridimensional para vehículos terrestres robotizados y es factible para otros algoritmos de navegación de vehículos.

4.2. NVIDIA Jetson Nano

Las pruebas experimentales realizadas en este trabajo se hicieron con un robot equipado con una tarjeta Jetson Nano como plataforma computacional a bordo. Esta plataforma computacional se eligió debido a su bajo costo energético y alto rendimiento computacional, ideal para aplicaciones prototipo. Esta sección proporciona una guía para configurar la tarjeta Jetson Nano de 4GB utilizando el lenguaje de programación Python 3.6.9 y los marcos de aprendizaje profundo TensorFlow, PyTorch y Torchvision, junto con los paquetes necesarios para el controlador PCA9685.

4.2.1. Requisitos preliminares

Los requisitos preliminares para habilitar la tarjeta Jetson Nano son los siguientes.

- Tarjeta microSD (mínimo 32GB).
- Fuente de alimentación (5V 4A recomendado).
- Teclado, ratón y monitor.
- Conexión a internet.

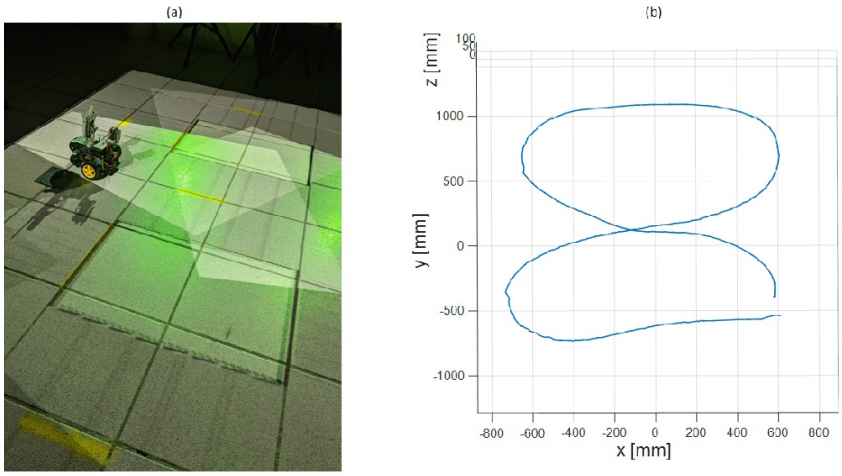

Figura 4.5: (a) y (b) Posiciones detectadas del robot móvil en diferentes fotogramas tomados por dos cámaras, respectivamente. (c) Se muestran las posiciones tridimensionales del robot móvil.

4.2.2. Instalación de la imagen Jetson Nano

1. Descargar la imagen de JetPack 4.5.2 desde el sitio web oficial de NVIDIA.
2. Usar la herramienta Etcher para cargar la imagen a la tarjeta microSD.
3. Insertar la tarjeta microSD en la Jetson Nano y encender el dispositivo.
4. Seguir las instrucciones en pantalla para completar la configuración inicial y crear una cuenta de usuario.

4.2.3. Instalación de TensorFlow

La instalación de TensorFlow en la tarjeta Jetson Nano requiere ejecutar manualmente un proceso debido a las versiones específicas de JetPack y Python que están instaladas. Estos pasos son descritas a continuación.

1. Abrir una terminal y actualizar el sistema.

```
sudo apt-get update
sudo apt-get upgrade
```

2. Instalar los paquetes necesarios para el sistema.

```
sudo apt-get install libhdf5-serial-dev hdf5-
    tools libhdf5-dev zlib1g-dev zip libjpeg8
    -dev liblapack-dev libblas-dev gfortran
```

3. Instalar y actualizar las bibliotecas de Python.

```
sudo pip3 install -U pip testresources
    setuptools==49.6.0
sudo pip3 install -U numpy==1.16.1 future
    ==0.18.2 mock==3.0.5 h5py==2.10.0
    keras_preprocessing==1.1.1
    keras_applications==1.0.8 gast==0.2.2
    futures protobuf pybind11
```

4. Instalar el TensorFlow.

```
sudo pip3 install --pre --extra-index-url
    https://developer.download.nvidia.com/
    compute/redist/jp/v45 tensorflow
```

4.2.4. Instalación de PyTorch y torchvision

La instalación manual de PyTorch y torchvision se realiza con las siguientes instrucciones.

1. Abrir una terminal y descargar el PyTorch y torchvision pre-compilada específicamente para la versión de JetPack utilizada.

```
wget https://nvidia.box.com/shared/static/
    p57jwntv436lfrd78inwl7iml6p13fzh.whl -O
    torch-1.10.0-cp36-cp36m-linux_aarch64.whl

git clone --branch v0.11.1 https://github.com
    /pytorch/vision torchvision
```

2. Instalar las dependencias requeridas para la PyTorch y torchvision.

```
sudo apt-get install libopenblas-base
    libopenmpi-dev libomp-dev

sudo apt-get install libjpeg-dev zlib1g-dev
    libpython3-dev libopenblas-dev libavcodec
    -dev libavformat-dev libswscale-dev
```

3. Instalar el PyTorch.

```
pip3 install 'Cython<3'

pip3 install numpy torch-1.10.0-cp36-cp36m-
    linux_aarch64.whl
```

4. Instalar el torchvision.

```
cd torchvision

export BUILD_VERSION=0.11.1

python3 setup.py install --user
```

4.2.5. Instalación del controlador PCA9685

La instalación del controlador PCA9685 es más sencilla, ya que se realiza con una sola instrucción. Sin embargo, es crucial asegurarse de tener las versiones específicas de las bibliotecas de Python para garantizar el funcionamiento correcto de la tarjeta.

```
sudo pip3 install -U \
    adafruit-circuitpython-busdevice==5.1.2 \
    adafruit-circuitpython-motor==3.3.5 \
    adafruit-circuitpython-pca9685==3.4.1 \
    adafruit-circuitpython-register==1.9.8 \
    adafruit-circuitpython-servokit==1.3.8 \
    Adafruit-Blinka==6.11.1 \
    Adafruit-GPIO==1.0.3 \
    Adafruit-MotorHAT==1.4.0 \
    Adafruit-PlatformDetect==3.19.6 \
    Adafruit-PureIO==1.1.9 \
    Adafruit-SSD1306==1.6.2
```

Cabe señalar que es muy importante verificar cuidadosamente la instalación correcta de los paquetes y la versión exacta de cada uno de ellos. Asimismo, se debe asegurar de instalar los marcos de aprendizaje profundo TensorFlow, PyTorch y torchvision, y los paquetes necesarios para trabajar con el controlador PCA9685. La instalación defectuosa, o instalación de versiones incompatibles, evitará desarrollar y ejecutar aplicaciones avanzadas de alto rendimiento en la Jetson Nano.

Capítulo 5

Resultados experimentales

La evaluación experimental del robot móvil en una prueba de navegación se realizó usando una pista dinámica generada con un sistema de cuatro proyectores. La homografía asociada a cada proyector fue estimada previamente relacionando el plano de referencia y el plano diapositiva de cada proyector. La figura 5.1(a) muestran los puntos usados para la estimación de homografías de un proyector. Estos puntos fueron detectados utilizando un patrón de calibración desplegado por el dispositivo, asegurando una alta precisión en la correspondencia de puntos. Por otro lado, los puntos del plano de referencia fueron marcados en las intersecciones de los azulejos del suelo, como se muestra en la figura 5.1(b). Esta calibración aseguró que la proyección y detección de puntos en el entorno fueran precisas, estableciendo como resultado la plataforma de una imagen coherente. Después, se realizan el mismo procedimiento para cada una de los proyectores empleados. La figura 5.1(c) muestra la re-proyección de la imagen de entrada de un proyector para verificar que la homografía asociada fue estimada correctamente.

Se implementó un sistema de transferencia inalámbrico para enviar a los proyectores los fragmentos de imagen correspondientes. La figura 5.2 muestra un ejemplo de dos fragmentos de imagen que deben enviarse a dos proyectores. Asimismo, la figura 5.3 muestra el resultado de proyectar las imágenes enviadas a los proyectores. Esta configuración permitió abarcar la mayor parte del campo de prueba, facilitando el uso de la pista como ruta de navegación para el vehículo terrestre.

Posteriormente, se capturó un video por medio una red neuronal especializada en la detección de líneas de carril, crítico para la navegación del robot móvil en tiempo real. La detección precisa de estas líneas permitió al controlador ajustar los parámetros necesarios para guiar el robot móvil con alta precisión. Debido a los movimientos del vehículo, el video fue sometido a un pre-procesamiento para mejorar la estabilidad de la grabación, eliminando la fluctuación o ruido que pudiera afectar la precisión del algoritmo propuesto.

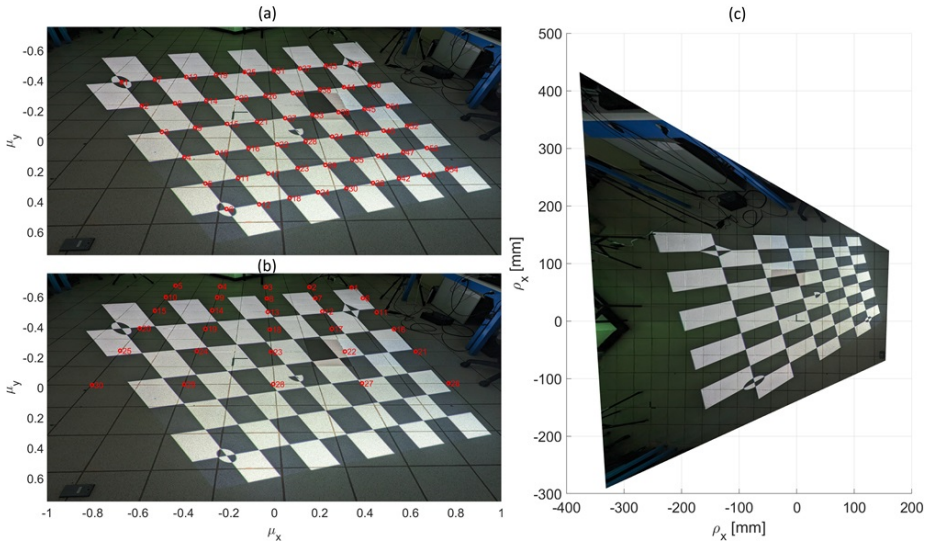

Figura 5.1: Etapas de la calibración de un proyector para desplegar imágenes superpuestas. (a) Puntos del patrón de calibración proyectado. (b) Puntos conocidos en el plano de referencia. (c) Re-proyección de la imagen de entrada para validar que se ha detectado correctamente el plano de referencia. Observe las lozas del suelo están alineadas en filas horizontales y verticales del mismo tamaño.

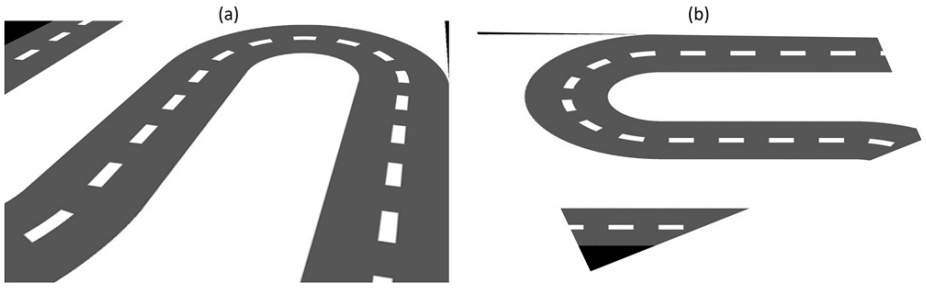

Figura 5.2: Ambas son imágenes de salida en distintas perspectivas. (a) Es la imagen o diapositiva que desplegará el proyector, y (b) es la imagen desplegado hacia el plano de referencia de la escena.

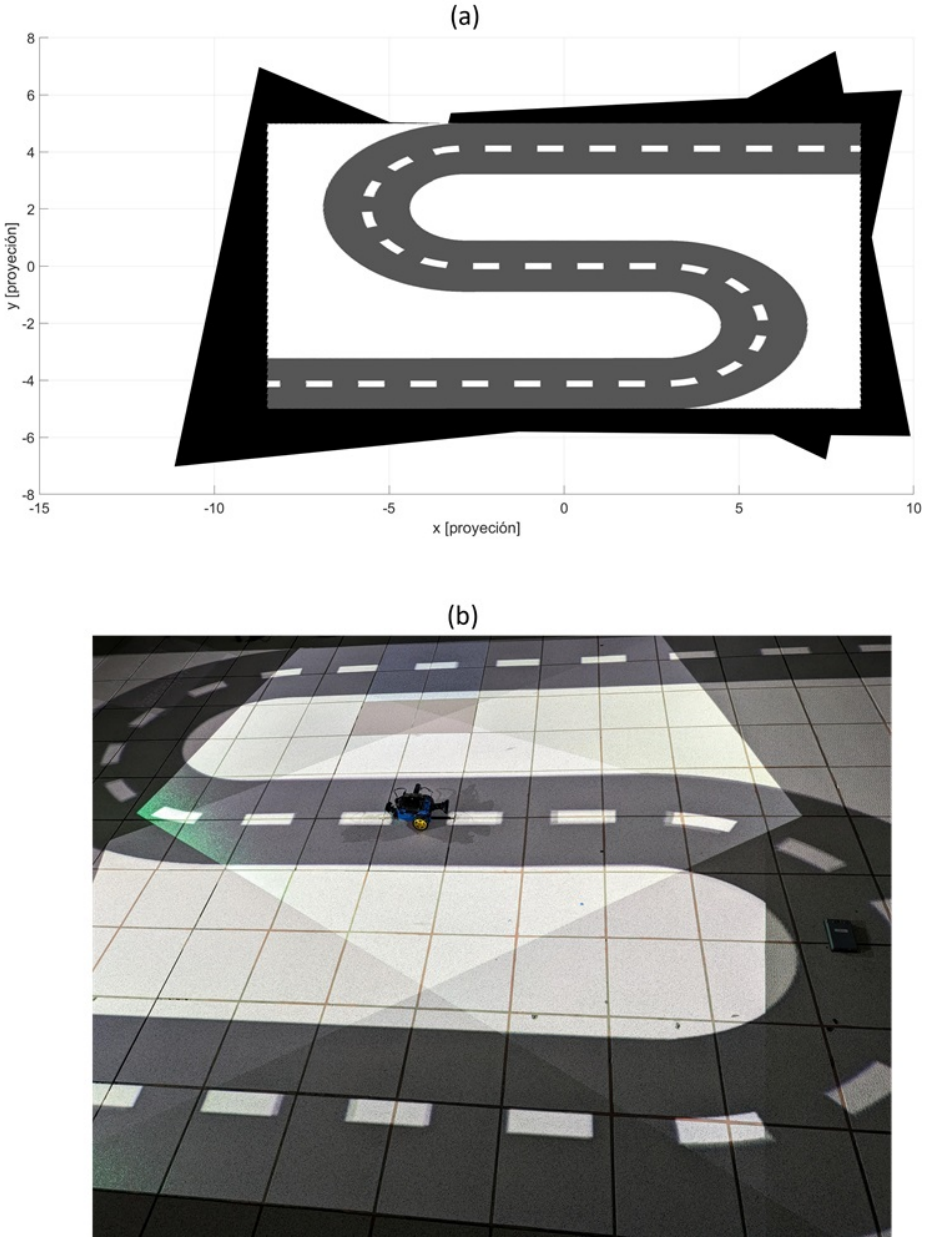

Figura 5.3: Resultado de calibración de proyectores. (a) Simulación de la plataforma calibrada para una configuración de cuatro proyectores. (b) Plataforma de la escena real usando los parámetros obtenidos para generar imágenes superpuestas por los sistemas de proyectores configuradas.

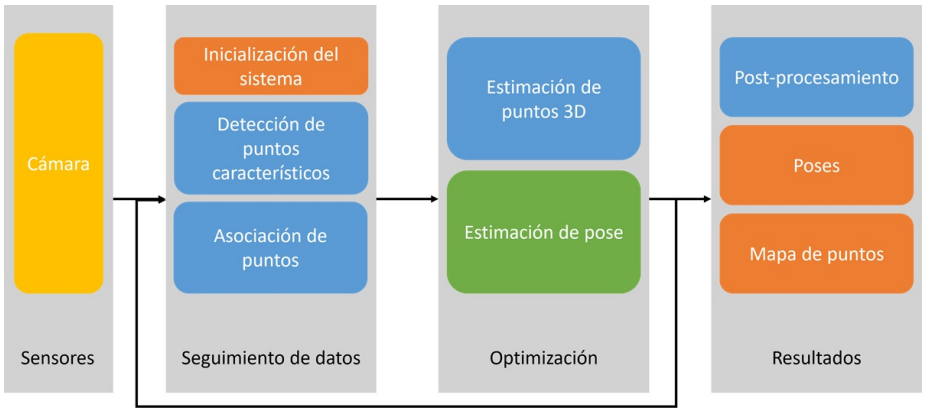

Figura 5.4: Diagrama de flujo del algoritmo propuesto para la estimación de pose usando la información tridimensional.

El algoritmo propuesto fue utilizado para estimar puntos tridimensionales y realizar el seguimiento de estos puntos con el fin de determinar la pose del robot móvil terrestre. Inicialmente, se llevó a cabo una estimación preliminar de los puntos en la escena. Después, se identificaron y rastrearon los puntos que podían ser detectados nuevamente en la escena a través de distintas vistas. Estos puntos de correspondencia permitieron obtener los trazos de rayos necesarios para realizar la triangulación, utilizando múltiples vistas procesadas. El proceso de triangulación fue fundamental para obtener una reconstrucción tridimensional precisa del entorno.

Al finalizar el procesamiento, se optimizaron las poses del vehículo mediante el método de ajuste conjunto. Este proceso implicó la estimación de las re-proyecciones de los puntos utilizando todas las vistas capturadas y los puntos de correspondencia detectados. La reducción de puntos redundantes y el ajuste preciso de las poses resultaron en una mejora significativa de los resultados, como se ilustra en la figura 5.5. Esta optimización se realizó utilizando el método iterativo de Gauss-Newton, que minimiza el error de re-proyección y mejora la precisión general del sistema. Finalmente, la figura 5.6 muestra las posiciones estimadas a lo largo de toda la secuencia de vídeo, optimizadas mediante la detección y el seguimiento de puntos de correspondencia.

Los resultados obtenidos validan la eficacia del algoritmo de estimación de pose, mostrando que puede lograr una precisión similar a la obtenida con técnicas basadas en inteligencia artificial. Sin embargo, a diferencia de los métodos basados en redes neuronales, que requieren un pre-entrenamiento extenso y costoso en términos computacionales, el algoritmo propuesto no requiere esta fase de pre-entrenamiento. Esta característica reduce significativamente el tiempo y los recursos necesarios para implementar el sistema, haciéndolo más accesible y práctico para diversas aplicaciones.

La precisión del algoritmo se comparó con varios enfoques de inteligencia artificial en términos de estimación de pose y seguimiento de características. Los resultados indicaron que, aunque los métodos basados en redes neuronales pueden ofrecer alta precisión, el algoritmo propuesto logra un rendimiento comparable sin la necesidad de entrenamiento previo. Esto se debe a su capacidad para procesar directamente la información tridimensional capturada por las cámaras y optimizar la pose del robot móvil terrestre.

Además, el algoritmo demostró ser robusto frente a variaciones en el entorno y cambios en las condiciones de iluminación, factores que a menudo afectan negativamente a los sistemas basados en inteligencia artificial. Esta robustez se debe en parte al uso de técnicas de procesamiento multidimensional y a la integración de múltiples vistas para la triangulación, lo que proporciona una estimación más precisa y confiable de la pose del robot.

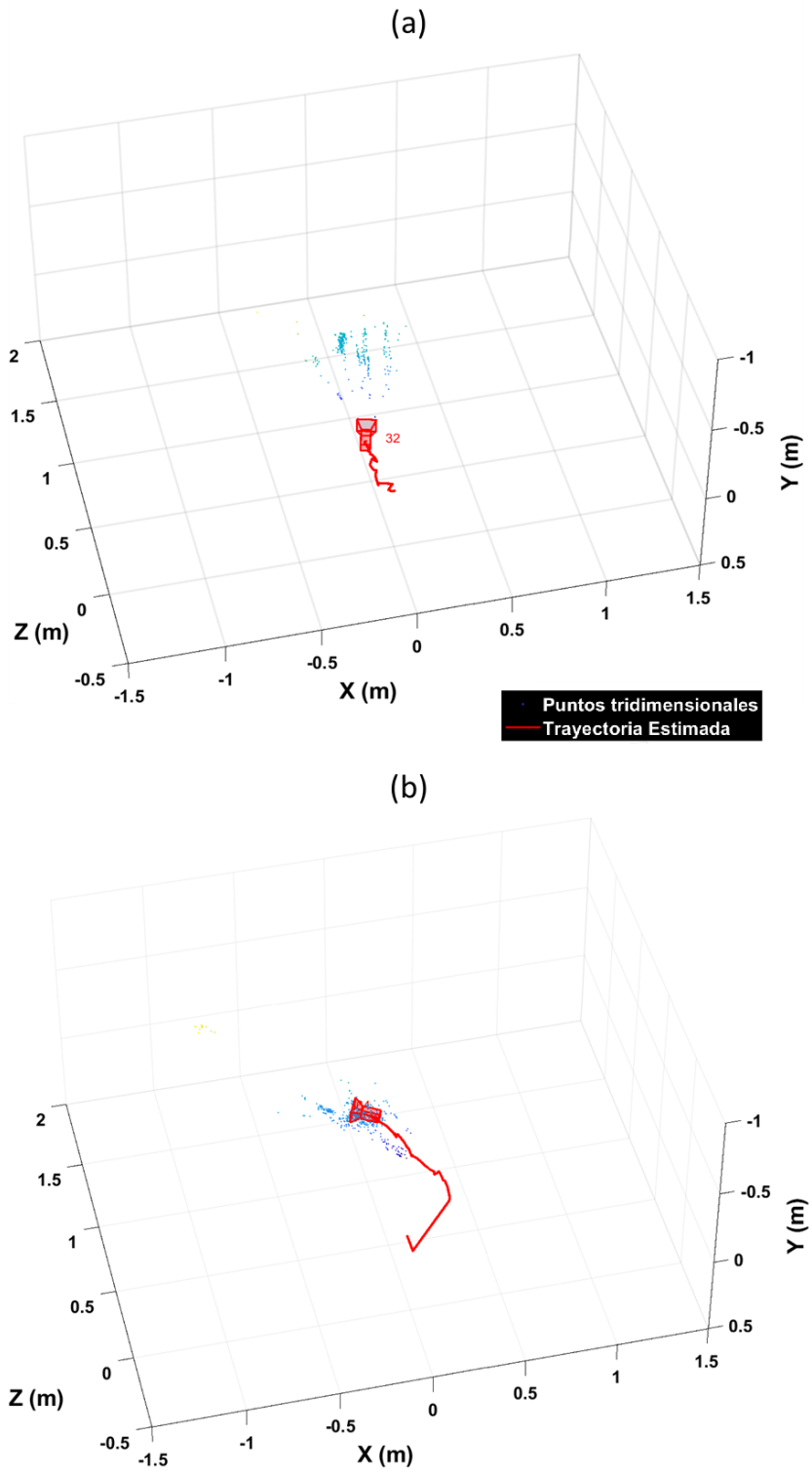

Figura 5.5: Las posiciones estimadas se derivan de puntos tridimensionales obtenidos. (a) El vehículo no puede continuar la navegación debido a la insuficiencia de puntos usados. (b) El vehículo logra completar la trayectoria planificada.

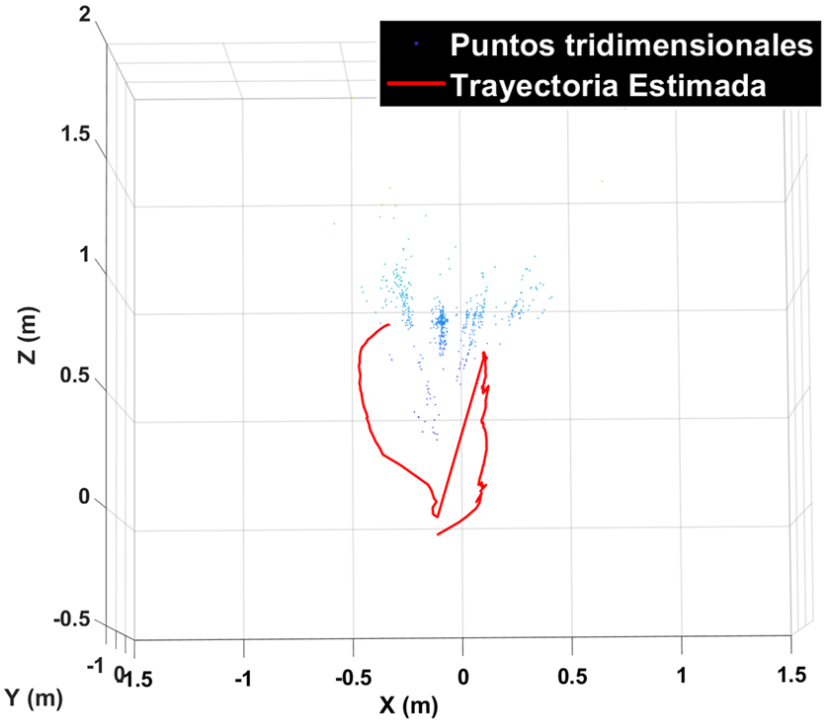

Figura 5.6: Los puntos en seguimientos y posiciones estimadas por la navegación del robot móvil terrestre en la pista dinámica generada.

Capítulo 6

Conclusiones

En esta tesis, se propuso un método de estimación de pose basado en información visual y una plataforma multi-proyector para la generación de pistas dinámica, útiles para pruebas de navegación. Se presentaron los fundamentos conceptuales de operación del sistema y se realizaron pruebas experimentales de operación. Los resultados obtenidos mostraron la factibilidad del método propuesto para desarrollar estrategias navegación para robots autónomos. Asimismo, el sistema multi-proyector construido mostró gran utilidad para evaluar otros sistemas de navegación en entornos dinámicos.

Durante el transcurso del presente trabajo de tesis se observó la utilidad de incluir sensores adicionales como guías láser, unidades inerciales, y codificadores rotativos, para complementar el sistema de visión. Por ejemplo, los sensores láser permiten conocer la distancia entre el robot y los objetos circundantes para realizar un seguimiento y evasión de obstáculos, una tarea crucial que complementa el problema de navegación visual. La evasión de obstáculos ha sido ampliamente documentada como esencial en robótica autónoma, mejorando la seguridad y la eficiencia del movimiento en entornos dinámicos. Los sistemas inerciales ayudan ayudan a determinar los cambios bruscos de elevación y mantener el vehículo en una trayectoria adecuada sobre superficies no planas, estabilizando la navegación en terrenos irregulares. Además, los codificadores rotativos complementan la estimación de pose utilizando información histórica y temporal del vehículo en movimiento, mejorando así la precisión en la navegación continua y proporcionando datos críticos para la corrección de trayectoria.

La inclusión de múltiples sensores permite una fusión sensorial más robusta, que es crucial para la navegación autónoma en entornos complejos. Esta fusión de sensores mejora la redundancia y la confiabilidad del sistema, permitiendo al robot definir trayectorias basándose en múltiples fuentes de datos. La fusión de datos ha sido explorada en diversas aplicaciones robóticas, mostrando cómo la combinación de diferentes tipos de sensores que puede superar las limitaciones individuales de cada uno de ellos.

El marco teórico desarrollando en este trabajo de investigación también tiene aplicaciones en otras disciplinas como cirugía asistida por computadora, ingeniería inversa, vigilancia, y control de calidad en líneas de producción. Por ejemplo, la realidad aumentada es una tecnología que permite simular escenarios críticos que son de alto riesgo de practicar sin tener conocimientos previos. En navegación aéreas, un estudiante puede realizar múltiples simulaciones antes de realizar vuelos en condiciones reales. En medicina, los sistemas de simulación quirúrgica permiten a los médicos practicar procedimientos complejos, minimizando los riesgos asociados a errores durante cirugías reales. La realidad aumentada también se utiliza en la formación de personal en industrias como nuclear y la petroquímica, donde los errores pueden tener consecuencias catastróficas.

El método propuesto no solo mejora la precisión de la localización del vehículo, sino también tiene el potencial de integrarse con sistemas de inteligencia artificial. La combinación de visión por computadora y aprendizaje profundo ha demostrado ser efectiva en diversas aplicaciones de navegación autónoma, ofreciendo soluciones adaptativas que mejoran el rendimiento en tiempo real. Además, el uso de técnicas de aprendizaje permite incrementar las capacidades de navegación a través de la evolución del algoritmo que optimiza el rendimiento en entornos dinámicos y desconocidos.

Los resultados experimentales obtenidos en este trabajo de tesis validaron la utilidad del método de visión propuesto para tareas de navegación autónoma. Complementar el método con un sistema de aprendizaje automático ofreció una alternativa práctica para la detección de pistas sin necesidad de preentrenamiento, lo que representa una solución prometedora para futuros desarrollos en el campo de la robótica móvil.

En conclusión, este trabajo de investigación abre nuevas oportunidades de investigación en robótica móvil, visión por computadora y otras áreas relacionadas, como la realidad aumentada y la robótica industrial. La integración de sensores adicionales y técnicas de aprendizaje automático mejora significativamente la capacidad de los sistemas autónomos para navegar en entornos complejos y dinámicos. La investigación futura podría centrarse en la mejora de los algoritmos de fusión sensorial y en la implementación de redes neuronales más avanzadas para optimizar la toma de decisiones en tiempo real, expandiendo así las aplicaciones y la eficiencia de los robots autónomos.

Apéndice A

Estimación de homografías

Las homografías son de gran utilidad cuando se emplea una cámara pinhole y se desea relacionar los puntos del plano de la imagen (cámara) o del plano de diapositivas (proyector) con puntos de un plano de referencia en el espacio tridimensional [20]. El sistema de transformación de puntos se puede describir como

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

donde $\boldsymbol{\mu}$ y $\boldsymbol{\rho}$ son puntos de correspondencia entre el plano de la imagen o diapositiva y el plano de referencia, respectivamente, G es la matriz de homografía, y $\mathcal{H}[\cdot]$ es el operador de coordenadas homogéneas [20]. La homografía G se define como una matriz de 3×3 y también puede representarse como

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{g}}_1^T \\ \bar{\boldsymbol{g}}_2^T \\ \bar{\boldsymbol{g}}_3^T \end{bmatrix}, \tag{A.2}$$

donde $\bar{\boldsymbol{g}}_1^T$, $\bar{\boldsymbol{g}}_2^T$ y $\bar{\boldsymbol{g}}_3^T$ son las filas de la matriz G. Usando esta definición, la ecuación (A.1) puede reescribirse como

$$\boldsymbol{\mu} = \mathcal{H}^{-1}\left[\begin{bmatrix} \bar{\boldsymbol{g}}_1^T \\ \bar{\boldsymbol{g}}_2^T \\ \bar{\boldsymbol{g}}_3^T \end{bmatrix} \mathcal{H}[\boldsymbol{\rho}]\right] = \mathcal{H}^{-1}\left[\begin{bmatrix} \bar{\boldsymbol{g}}_1^T\mathcal{H}[\boldsymbol{\rho}] \\ \bar{\boldsymbol{g}}_2^T\mathcal{H}[\boldsymbol{\rho}] \\ \bar{\boldsymbol{g}}_3^T\mathcal{H}[\boldsymbol{\rho}] \end{bmatrix}\right] = \frac{1}{\bar{\boldsymbol{g}}_3^T\mathcal{H}[\boldsymbol{\rho}]}\begin{bmatrix} \bar{\boldsymbol{g}}_1^T\mathcal{H}[\boldsymbol{\rho}] \\ \bar{\boldsymbol{g}}_2^T\mathcal{H}[\boldsymbol{\rho}] \end{bmatrix}. \tag{A.3}$$

Usando la ecuación (A.3), se puede escribir un sistema de ecuaciones lineales como

$$\begin{bmatrix} g_{31}\rho_x\mu_x + g_{32}\rho_y\mu_x + \mu_x \\ g_{31}\rho_x\mu_y + g_{32}\rho_y\mu_y + \mu_y \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{g}}_1^T\mathcal{H}[\boldsymbol{\rho}] \\ \bar{\boldsymbol{g}}_2^T\mathcal{H}[\boldsymbol{\rho}] \end{bmatrix}, \tag{A.4}$$

que se puede reescribir convenientemente colocando las incógnitas en el lado derecho de la igualdad como

$$\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} = \begin{bmatrix} \mathcal{H}[\boldsymbol{\rho}]^T\bar{\boldsymbol{g}}_1 \\ \mathcal{H}[\boldsymbol{\rho}]^T\bar{\boldsymbol{g}}_2 \end{bmatrix} - \begin{bmatrix} \rho_x\mu_x g_{31} + \rho_y\mu_x g_{32} \\ \rho_x\mu_y g_{31} + \rho_y\mu_y g_{32} \end{bmatrix}, \tag{A.5}$$

o, en forma matricial,

$$\underbrace{\begin{bmatrix} \mathcal{H}[\boldsymbol{\rho}]^T & \mathbf{0_3^T} & -\rho_x\mu_x & -\rho_y\mu_x \\ \mathbf{0}_3^T & \mathcal{H}[\boldsymbol{\rho}]^T & -\rho_x\mu_y & -\rho_y\mu_y \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \bar{\boldsymbol{g}}_1 \\ \bar{\boldsymbol{g}}_2 \\ g_{31} \\ g_{32} \end{bmatrix}}_{\boldsymbol{g}} = \boldsymbol{\mu}, \tag{A.6}$$

donde $\boldsymbol{g}$ es el vector de incógnitas, y contiene los elementos de la matriz homografía. Observe que la ecuación (A.6) corresponde al caso donde hay un punto de correspondencia $(\boldsymbol{\mu}, \boldsymbol{\rho})$, obteniendo dos ecuaciones. Sin embargo, esto no es suficiente para determinar el vector $\boldsymbol{g}$, que contiene ocho incógnitas. Por lo tanto, es necesario al menos cuatro puntos de correspondencia para estimar una matriz de homografía. Para el caso general, un sistema de ecuaciones matriciales para n puntos de correspondencia se representa como

$$\underbrace{\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_n \end{bmatrix}}_{\mathcal{A}} \boldsymbol{g} = \underbrace{\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_3 \\ \vdots \\ \boldsymbol{\mu}_n \end{bmatrix}}_{\mathcal{Y}}. \tag{A.7}$$

Este sistema se puede resolver utilizando el *método de mínimos cuadrados*, de modo que el vector $\boldsymbol{g}$ se puede calcular como

$$\boldsymbol{g} = (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T\mathcal{Y}. \tag{A.8}$$

En este trabajo de tesis, el método de estimación de homografía descrito se usó para diversas áreas, incluyendo calibración de cámaras, calibración de proyectores, y formación de imágenes mosaico para la generación dinámica de imágenes usando un sistema multi-proyector.

Apéndice B

Detección de objetos

En este trabajo de investigación, la detección de objetos a partir de imágenes fue una tarea que se desarrolló empleando diferentes métodos. A continuación se describen cuatro enfoques evaluados durante este trabajo de tesis; en particular, la detección de objetos empleando flujo óptico, detección por espacios de color, filtros de correlación, y redes neuronales convolucionales.

B.1. Flujo óptico

El flujo óptico refiere a los campos de velocidad de intensidad que provocan los objetos en movimiento en una imagen. La importancia del análisis del flujo óptico se debe a su utilidad en aplicaciones de visión por computadora, puesto que permite detectar y obtener el movimiento de los objetos en una escena. Este método se ha convertido en un línea de investigación importante por su eficiencia y utilidad en aplicaciones como segmentación, estructura tridimensional, estabilización, y compresión de vídeo, entre otras [56, 57].

La estimación clásica del flujo óptico inicia definiendo una imagen E donde su nivel de intensidad en el punto x, y en el instante de tiempo t, se representa como

$$E(x, y, t). \tag{B.1}$$

Esta intensidad permanece constante en un instante de tiempo t cualesquiera; es decir,

$$\frac{dE}{dt} = 0. \tag{B.2}$$

De igual manera, un punto en la imagen puede trasladarse a una cierta distancia en dirección de los ejes en un determinado tiempo, como

$$E(x, y, t) = E(x + \delta_x, y + \delta_y, t + \delta_t), \tag{B.3}$$

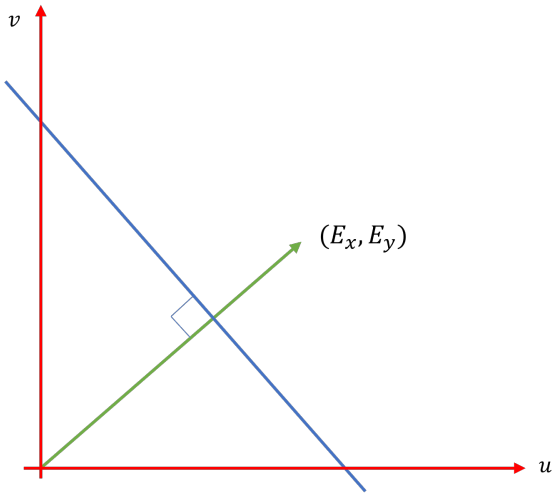

Figura B.1: La velocidad u, v forma parte de la línea recta perpendicular al vector de nivel de intensidad E_x, E_y.

que podemos reescribir usando series de Taylor como

$$E(x, y, t) = E(x, y, t) + \delta_x \frac{\partial E}{\partial x} + \delta_y \frac{\partial E}{\partial y} + \delta_t \frac{\partial E}{\partial t} + \epsilon, \tag{B.4}$$

donde ϵ son los términos de orden superior. Después, despreciando los términos de orden superior, dividiendo por δ_t y en el límite $\delta_t \to 0$ se obtiene la siguiente ecuación diferencial

$$\frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0, \tag{B.5}$$

o, simplificando la notación, llegamos a lo que se conoce como *ecuación lineal del flujo óptico* dada por

$$E_x u + E_y v + E_t = 0, \tag{B.6}$$

donde u y v son las velocidades del flujo en los ejes x e y, respectivamente. Utilizando la ecuación de la recta en el espacio u, v, se forma la figura B.1, donde la velocidad se encuentra dentro de la línea recta y el vector del nivel de intensidad (E_x, E_y) siempre permanece perpendicular.

En la ecuación (B.6) se tienen dos incógnitas y una sola restricción. Por lo tanto, la ecuación (B.6) no proporciona información suficiente para poder calcular el flujo óptico de forma única. El método de Horn-Schunck agrega una restricción de la variación de suavidad que toma en cuenta todos los píxeles [13]. Usando está restricción global se puede determinar el flujo óptico de cada píxel, sin embargo, esto puede causar problema al análisis de la detección debido a que es sensible al ruido. Por otro lado, el método de Lucas-Kanade asume

una restricción local para determinar el flujo óptico que ofrece una robustez relativamente alta en ambientes no controlados [14]. Está restricción asume un movimiento constante de un espacio determinado.

Dado el criterio de mínimos cuadrados, la ecuación (B.6) para las velocidades $\boldsymbol{v} = [u, v]$ en una pequeña vecindad espacial Ω se reescribe como

$$\sum_{\boldsymbol{x}\epsilon\Omega} W^2(\boldsymbol{x})[E_x u + E_y v + E_t]^2 = 0, \tag{B.7}$$

donde $W(\boldsymbol{x})$ es una función ventana donde el centro de la vecindad tiene un mayor peso que los que rodean. Desarrollando la ecuación (B.7), se obtiene

$$\begin{bmatrix} E_{x_1} & \dots & E_{x_n} \\ E_{y_1} & \dots & E_{y_n} \end{bmatrix} \begin{bmatrix} W_{\boldsymbol{x}_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & W_{\boldsymbol{x}_n} \end{bmatrix}^2 \begin{bmatrix} E_{x_1} & E_{y_1} \\ \vdots & \vdots \\ E_{x_n} & E_{y_n} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} E_{x_1} & \dots & E_{x_n} \\ E_{y_1} & \dots & E_{y_n} \end{bmatrix} \begin{bmatrix} W_{\boldsymbol{x}_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & W_{\boldsymbol{x}_n} \end{bmatrix}^2 + \begin{bmatrix} E_{t_1} & \dots & E_{t_n} \end{bmatrix} = 0, \tag{B.8}$$

y se puede simplificar como

$$A^T W^2 A \boldsymbol{v} = A^T W^2 \boldsymbol{b}, \tag{B.9}$$

donde

$$\begin{aligned} A &= [E_{x_1} + E_{y_1}, E_{x_2} + E_{y_2}, \dots, E_{x_n} + E_{y_n}]^T, \\ W &= \text{diag } [W(\boldsymbol{x}_1), W(\boldsymbol{x}_2), \dots, W(\boldsymbol{x}_n)], \\ \boldsymbol{b} &= -[E_{t_1}, E_{t_2}, \dots, E_{t_n}]. \end{aligned} \tag{B.10}$$

El flujo óptico $\boldsymbol{v}$ se calcula directamente utilizando la pseudoinversa de Moore-Penrose [58] para encontrar la solución de la ecuación (B.10) como

$$\boldsymbol{v} = [(A^T W^2)^T (A^T W^2)]^{-1} (A^T W^2)^T A^T W^2 \boldsymbol{b}. \tag{B.11}$$

Para validar el método estudiado, se realizó una simulación computacional para observar el flujo óptico estimado y su robustez cuando las condiciones son controladas. En esta simulación se generan cuatro círculos de diferentes patrones como objeto de interés. Los objetos de la escena se pueden manipular usando transformaciones geométrica como traslación y rotación. Cada imagen generada es procesada para determinar el flujo óptico usando una ventana espacial de 20×20. en En la figura B.2, se observan los resultados obtenidos, donde su pudo verificar que el proceso de estimación de flujo óptico se implementó correctamente.

En este trabajo se realizó una comparación entre el método de Horn-Schunck y Lucas-Kanade. Ambos son métodos para determinar el flujo óptico. Por un lado, el enfoque de Horn-Schunck es global debido a que utiliza el criterio de

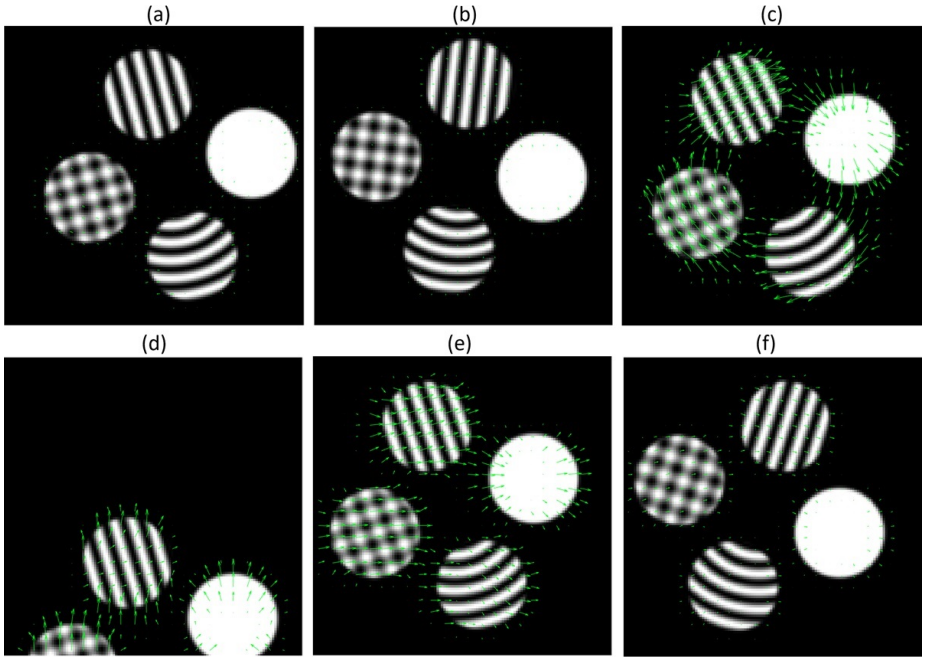

Figura B.2: Simulación para determinar el flujo óptico en una escena con objetos en movimiento.

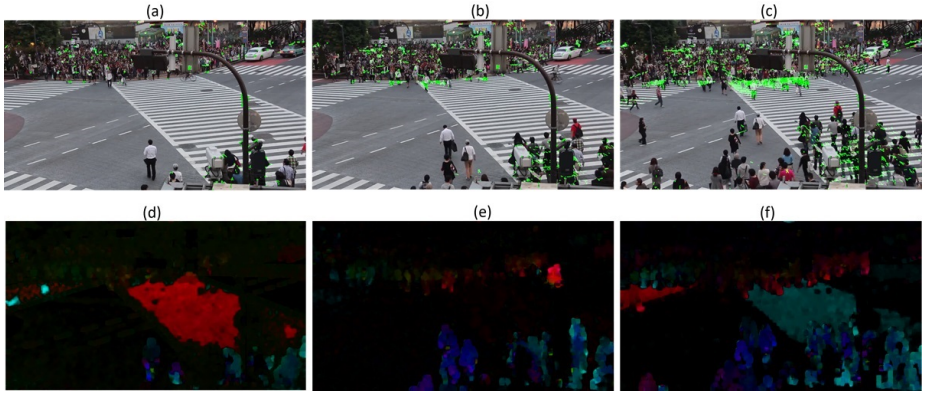

Figura B.3: Diferentes visualizaciones del flujo óptico. (a)-(c) Son resultados obtenidos mediante el método de Lucas-Kanade. Por otro lado, (d)-(f) son resultados obtenidos del método de Horn-Schunck.

suavidad en toda la imagen. Por otro lado, el método de Lucas-Kanade es un enfoque local ya que asume que la velocidad es constante solo dentro de una vecindad espacial. Para este experimento, se utiliza un video para obtener una secuencia de imágenes que será procesada usando ambos algoritmos. Los resultados del flujo óptico se pueden observar en la figura B.3. Se utilizó el método de Lucas-Kanade para obtener los resultados de la figura B.3(a)-(c). Como parámetro de entrada al algoritmo se utilizó una vecindad espacial de 15×15 para 200 características. Asimismo, la figura B.3(d)-(f) son los resultados obtenidos por el método de Horn-Schunck usando como parámetro de entrada una ventana de tamaño 3×3 para estimar la suavidad. Dado los resultados obtenidos, se puede observar que el método local procesa solamente ciertos puntos de interés, mientras el método global procesa todos los píxeles de la imagen. El método global implica bastante procesamiento computacional y susceptibilidad al ruido por resultados adicionales que no sean de interés.

B.2. Espacio de colores

La detección de color mediante el espacio HSV (por sus siglas en inglés Hue, Saturation, Value – Matiz, Saturación, Valor) es una técnica utilizada en procesamiento de imágenes y visión por computadora. El espacio de color HSV se descompone la información en tres componentes descritas a continuación. La

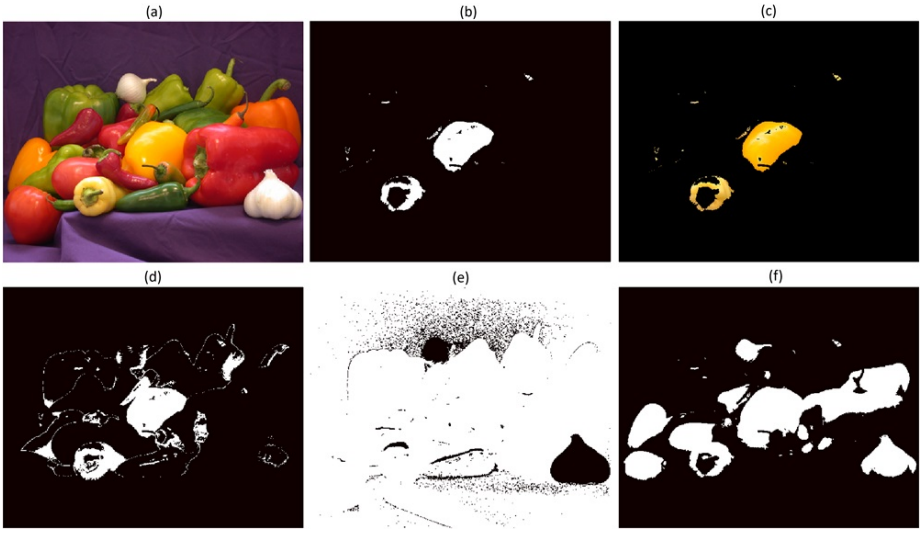

Figura B.4: Detección de colores mediante el uso de máscaras. (a) Imagen de entrada. (b) Máscara binaria. (c) Imagen de salida. (d)-(f) Máscaras binarias de los canales del espacio HSV.

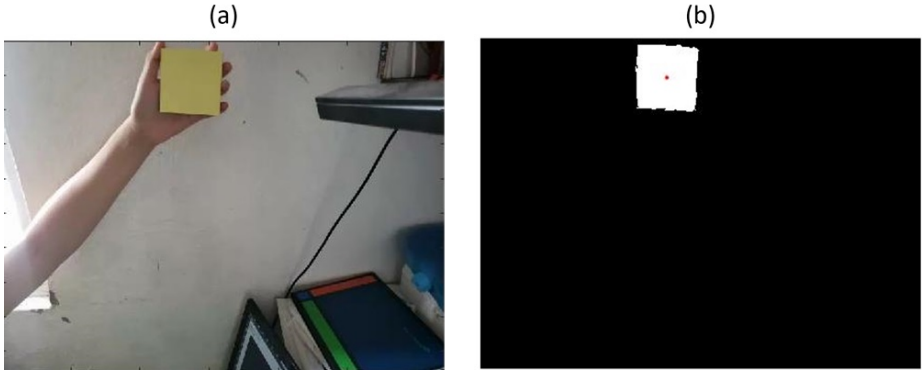

Figura B.5: Detector de objetos mediante la información obtenida por los colores.

matiz representa el tipo de color, y se mide en grados de 0 a 360. Por ejemplo, el grado 0 representa el color rojo, el grado 120 representa el color verde, y el grado 240 representa el color azul. Por otro lado, la saturación indica la percepción de la pureza en colorimetría, que varia entre los valores de 0 a 100 %. Un valor del 0 % agrega una tonalidad de gris, mientras que 100 % es el color más puro. Por último, el valor es conocido como brillo, tiene valores entre 0 a 100 % donde el valor mínimo equivale a colores oscuros y el valor máximo es el color más brillante equivalente al blanco.

El proceso de detección de objeto mediante el espacio de colores es lo siguiente. Primero, se selecciona un color para generar tres máscaras binarias mediante un valor constante. Cada máscara pertenece a un componente del espacio HSV. Después, se realiza una operación lógica *AND* con las tres máscaras para obtener una máscara binaria en el espacio RGB (por sus siglas en inglés Red, Green, Blue – Rojo, Verde, Azul). Finalmente, se aplica el filtro binario a la imagen de entrada para obtener el objeto de interés. En la figura B.4 se puede observar como ejemplo un experimento para la detección del color amarillo. Finalmente, la posición del objeto se obtiene determinando el centroide de los puntos detectados como se muestra en la figura B.5.

B.3. Filtros de correlación

El filtro de correlación construido por la mínima de la suma del error al cuadrado (del inglés *minimum output sum of squared error MOSSE*) es una técnica avanzada utilizada en visión por computadora para tareas de rastreo de objetos [55, 59, 60]. El filtro MOSSE se basa en la optimización de un filtro de correlación para minimizar el error cuadrático medio entre su salida y una imagen del objeto deseado. Es decir, el objetivo de este método es determinar un filtro H que, al correlacionarse con una imagen I, produzca una respuesta G

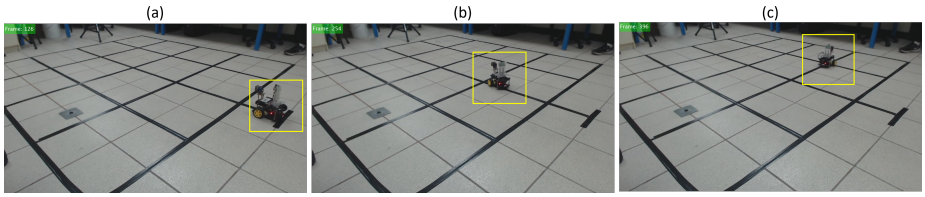

Figura B.6: Detección de un robot móvil terrestre usando filtros de correlación.

similar a una función delta centrada en la ubicación del objeto. Por lo tanto, es necesario minimizar el error del filtro usando múltiples imágenes del objetivos como

$$\min_H \sum_i^n |H \cdot I_i - G_i|^2, \tag{B.12}$$

donde I_i son las imágenes de entrenamiento y G_i es la función delta correspondiente. Por lo tanto, se propone generar el filtro de correlación de la siguiente manera. Primero, se selecciona el objeto de interés en la escena. A continuación, se genera un filtro mediante el entrenamiento con transformaciones geométricas del objeto seleccionado, utilizando el criterio de la ecuación (B.12). La imagen entrante se transforma al espacio de Fourier y se aplica una convolución con el filtro. Finalmente, se determina la posición del objeto de interés buscando el pico de intensidad máxima en el resultado de la convolución. En la Figura B.6 se pueden observar los resultados de la detección de un robot móvil utilizando filtros de correlación.

B.4. Redes neuronales convolucionales

La red neuronal convolucional ha demostrado ser altamente eficaz en diversas tareas de visión por computadora, incluyendo la detección de objetos [61–63]. Las redes neuronales profundas abordan el problema de la degradación, en el cual un aumento en el número de capas puede resultar en un mayor error y una disminución en la precisión del entrenamiento. Para mitigar este problema, se emplean bloques residuales que permiten el entrenamiento de redes profundas sin experimentar inconvenientes como el desvanecimiento del gradiente.

Para la detección de objetos, se utiliza la arquitectura Redes Residuales 18 (ResNet-18), que consta de 18 capas organizadas en un conjunto de bloques residuales, como se muestra en la figura B.7. Cada bloque incluye dos capas convolucionales seguidas por una conexión de atajo que omite estas dos capas, como se muestra en la figura B.8. Esta conexión facilita la propagación directa del gradiente a través de la red, permitiendo el entrenamiento eficiente de las capas más profundas.

Entre las ventajas de utilizar esta arquitectura se destacan su facilidad de implementación, adecuación para aplicaciones en tiempo real y dispositivos con

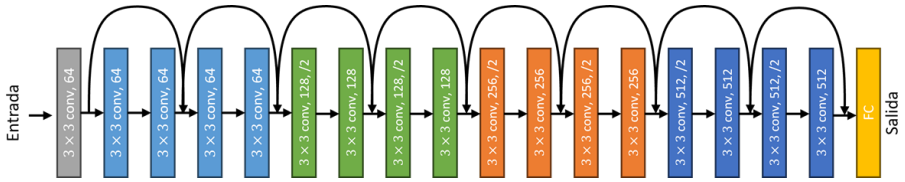

Figura B.7: La arquitectura de la red residual de 18 capas está organizada en un conjunto de bloques residuales, los cuales permiten omitir dos capas convolucionales. La última capa es crucial, ya que finaliza el proceso de aprendizaje y produce los resultados.

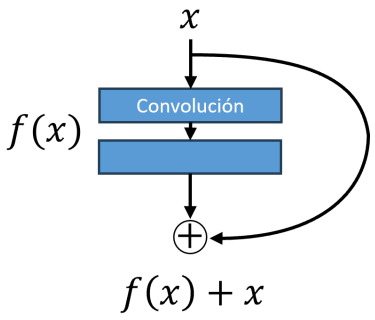

Figura B.8: Bloque de aprendizaje residual.

recursos limitados. La profundidad moderada de una ResNet-18 le permite generalizar bien en tareas de visión por computadora, manteniendo un equilibrio entre precisión y eficiencia. Debido a los bloques residuales, la ResNet-18 puede ser entrenada de manera eficiente, evitando problemas comunes en redes profundas, como el desvanecimiento del gradiente.

Bibliografía

[1] X. Zhang, C. Wang, L. Jiang, L. An, R. Yang, Collision-avoidance navigation systems for maritime autonomous surface ships: A state of the art survey, Ocean Engineering 235 (2021) 109380.

[2] P. Roy, C. Chowdhury, A survey of machine learning techniques for indoor localization and navigation systems, Journal of Intelligent & Robotic Systems 101 (3) (2021) 1–34.

[3] F. Gul, W. Rahiman, S. S. Nazli Alhady, A comprehensive study for robot navigation techniques, Cogent Engineering 6 (1) (2019) 1632046.

[4] F. Garcia, D. Martin, A. de la Escalera, J. M. Armingol, Sensor fusion methodology for vehicle detection, IEEE Intelligent Transportation Systems Magazine 9 (1) (2017) 123–133.

[5] R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza, Introduction to autonomous mobile robots, MIT press, 2011.

[6] F. Bonin-Font, A. Ortiz, G. Oliver, Visual navigation for mobile robots: A survey, Journal of Intelligent and Robotic Systems 53 (3) (2008) 263.

[7] K. Jo, K. Chu, M. Sunwoo, Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning, IEEE Transactions on Intelligent Transportation Systems 13 (1) (2012) 329–343.

[8] Q. Li, L. Chen, M. Li, S.-L. Shaw, A. Nüchter, A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios, IEEE Transactions on Vehicular Technology 63 (2) (2014) 540–555.

[9] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, J. Malik, Cognitive mapping and planning for visual navigation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7272–7281.

[10] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, A. Farhadi, Target-driven visual navigation in indoor scenes using deep reinforcement learning, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3357–3364.

[11] A. M. Hasan, K. Samsudin, A. R. Ramli, R. Azmir, S. Ismaeel, A review of navigation systems (integration and algorithms), Australian journal of basic and applied sciences 3 (2) (2009) 943–959.

[12] K. Zhang, F. Niroui, M. Ficocelli, G. Nejat, Robot navigation of environments with unknown rough terrain using deep reinforcement learning, in: 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), IEEE, 2018, pp. 1–7.

[13] B. K. Horn, B. G. Schunck, Determining optical flow, Artificial Intelligence 17 (1) (1981) 185 – 203.

[14] 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.

[15] N. Sharmin, R. Brad, Optimal filter estimation for lucas-kanade optical flow, Sensors 12 (9) (2012) 12694–12709.

[16] D. Fleet, Y. Weiss, Optical flow estimation, in: Handbook of mathematical models in computer vision, Springer, 2006, pp. 237–257.

[17] D. Sun, S. Roth, M. J. Black, Secrets of optical flow estimation and their principles, in: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, 2010, pp. 2432–2439.

[18] R. Juarez-Salazar, A. Giron, J. Zheng, V. H. Diaz-Ramirez, Key concepts for phase-to-coordinate conversion in fringe projection systems, Appl. Opt. 58 (2019) 4828–4834.

[19] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, 2004.

[20] R. Juarez-Salazar, V. H. Díaz-Ramírez, Operator-based homogeneous coordinates: application in camera document scanning, Optical Engineering 56 (7) (2017) 070801.

[21] L. N. Gaxiola, R. Juarez-Salazar, V. H. Diaz-Ramirez, Simple method for correction of distortion in images, in: Optics and Photonics for Information Processing X, Vol. 9970, 2016, pp. 147 – 153.

[22] S. Yoneyama, H. Kikuta, A. Kitagawa, K. Kitamura, Lens distortion correction for digital image correlation by measuring rigid body displacement, Optical Engineering 45 (2) (2006) 1 – 9.

[23] A. Wang, T. Qiu, L. Shao, A simple method of radial distortion correction with centre of distortion estimation, Journal of Mathematical Imaging and Vision 35 (3) (2009) 165–172.

[24] R. Juarez-Salazar, J. Zheng, V. H. Diaz-Ramirez, Distorted pinhole camera modeling and calibration, Applied Optics 59 (36) (2020) 11310–11318.

[25] A. Macario Barros, M. Michel, Y. Moline, G. Corre, F. Carrel, A comprehensive survey of visual slam algorithms, Robotics 11 (1) (2022) 24.

[26] H. Strasdat, J. M. Montiel, A. J. Davison, Visual slam: why filter?, Image and Vision Computing 30 (2) (2012) 65–77.

[27] I. A. Kazerouni, L. Fitzgerald, G. Dooly, D. Toal, A survey of state-of-the-art on visual slam, Expert Systems with Applications (2022) 117734.

[28] T. Taketomi, H. Uchiyama, S. Ikeda, Visual slam algorithms: A survey from 2010 to 2016, IPSJ Transactions on Computer Vision and Applications 9 (1) (2017) 1–11.

[29] R. Juarez-Salazar, G. A. Rodriguez-Reveles, S. Esquivel-Hernandez, V. H. Diaz-Ramirez, Three-dimensional spatial point computation in fringe projection profilometry, Optics and Lasers in Engineering 164 (2023) 107482.

[30] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, Lecture notes in computer science 3951 (2006) 404–417.

[31] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2004) 91–110.

[32] S. Leutenegger, M. Chli, R. Y. Siegwart, Brisk: Binary robust invariant scalable keypoints, in: 2011 International conference on computer vision, Ieee, 2011, pp. 2548–2555.

[33] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International conference on computer vision, Ieee, 2011, pp. 2564–2571.

[34] M. Peris, S. Martull, A. Maki, Y. Ohkawa, K. Fukui, Towards a simulation driven stereo vision system, in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE, 2012, pp. 1038–1042.

[35] S. Martull, M. Peris, K. Fukui, Realistic cg stereo image dataset with ground truth disparity maps, in: ICPR workshop TrakMark2012, Vol. 111, 2012, pp. 117–118.

[36] M. Muja, D. G. Lowe, Fast matching of binary features, in: 2012 Ninth conference on computer and robot vision, IEEE, 2012, pp. 404–410.

[37] J. Cheng, L. Zhang, Q. Chen, X. Hu, J. Cai, A review of visual slam methods for autonomous driving vehicles, Engineering Applications of Artificial Intelligence 114 (2022) 104992.

[38] R. Mur-Artal, J. M. M. Montiel, J. D. Tardos, Orb-slam: a versatile and accurate monocular slam system, IEEE transactions on robotics 31 (5) (2015) 1147–1163.

[39] A. Vu, A. Ramanandan, A. Chen, J. A. Farrell, M. Barth, Real-time computer vision/dgps-aided inertial navigation system for lane-level vehicle navigation, IEEE Transactions on Intelligent Transportation Systems 13 (2) (2012) 899–913. doi:10.1109/TITS.2012.2187641.

[40] N. Yang, W. F. Tian, Z. H. Jin, C. B. Zhang, Particle filter for sensor fusion in a land vehicle navigation system, Measurement science and technology 16 (3) (2005) 677.

[41] I. Cox, Blanche-an experiment in guidance and navigation of an autonomous robot vehicle, IEEE Transactions on Robotics and Automation 7 (2) (1991) 193–204. doi:10.1109/70.75902.

[42] G. Burnett, 'turn right at the traffic lights': The requirement for landmarks in vehicle navigation systems, Journal of Navigation 53 (3) (2000) 499–510. doi:10.1017/S0373463300001028.

[43] M. Ness, M. Herbert, A prototype low cost in-vehicle navigation system, in: Proceedings of VNIS '93 - Vehicle Navigation and Information Systems Conference, 1993, pp. 56–59. doi:10.1109/VNIS.1993.585584.

[44] Z. M. Kassas, P. Closas, J. Gross, Navigation systems panel report navigation systems for autonomous and semi-autonomous vehicles: Current trends and future challenges, IEEE Aerospace and Electronic Systems Magazine 34 (5) (2019).

[45] N. R. Velaga, M. A. Quddus, A. L. Bristow, Y. Zheng, Map-aided integrity monitoring of a land vehicle navigation system, IEEE Transactions on Intelligent Transportation Systems 13 (2) (2012) 848–858. doi:10.1109/TITS.2012.2187196.

[46] S. Julier, H. Durrant-Whyte, On the role of process models in autonomous land vehicle navigation systems, IEEE Transactions on Robotics and Automation 19 (1) (2003) 1–14. doi:10.1109/TRA.2002.805661.

[47] Q. Luo, Y. Cao, J. Liu, A. Benslimane, Localization and navigation in autonomous driving: Threats and countermeasures, IEEE Wireless Communications 26 (4) (2019) 38–45. doi:10.1109/MWC.2019.1800533.

[48] T. Sotiropoulos, G. Guiochet, I. Ingrand, W. Waeselynck, Virtual worlds for testing robot navigation: A study on the difficulty level, in: 2016 12th European Dependable Computing Conference (EDCC), 2016, pp. 153–160. doi:10.1109/EDCC.2016.14.

[49] J. Wan, H. Suo, H. Yan, J. Liu, A general test platform for cyber-physical systems: unmanned vehicle with wireless sensor network navigation, Procedia Engineering 24 (2011) 123–127.

[50] P. Trojanek, K. Eder, Verification and testing of mobile robot navigation algorithms: A case study in spark, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 1489–1494.

[51] J. B. Van Erp, H. A. Van Veen, Vibrotactile in-vehicle navigation system, Transportation Research Part F: Traffic Psychology and Behaviour 7 (4-5) (2004) 247–256.

[52] D. Capel, Image mosaicing, in: Image Mosaicing and super-resolution, Springer, 2004, pp. 47–79.

[53] J. Zheng, A. Giron, R. Juarez-Salazar, V. H. Diaz-Ramirez, Image stitching by projective transformations, in: Optics and Photonics for Information Processing XIII, International Society for Optics and Photonics, SPIE, 2019, pp. 51 – 55.

[54] H. Hou, H. Andrews, Cubic splines for image interpolation and digital filtering, IEEE Transactions on acoustics, speech, and signal processing 26 (6) (1978) 508–517.

[55] L. N. Gaxiola, V. H. Díaz-Ramírez, J. J. Tapia, A. Diaz-Ramirez, V. Kober, Robust face tracking with locally-adaptive correlation filtering, in: E. Bayro-Corrochano, E. Hancock (Eds.), Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer International Publishing, Cham, 2014, pp. 925–932.

[56] J. L. Barron, D. J. Fleet, S. S. Beauchemin, Performance of optical flow techniques, International Journal of Computer Vision 12 (1) (1994) 43–77.

[57] 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.

[58] R. Penrose, A generalized inverse for matrices, Mathematical Proceedings of the Cambridge Philosophical Society 51 (3) (1955) 406–413.

[59] D. S. Bolme, J. R. Beveridge, B. A. Draper, Y. M. Lui, Visual object tracking using adaptive correlation filters, in: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, 2010, pp. 2544–2550.

[60] V. H. Diaz-Ramirez, V. Contreras, V. Kober, K. Picos, Real-time tracking of multiple objects using adaptive correlation filters with complex constraints, Optics Communications 309 (2013) 265–278.

[61] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[62] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, Springer, 2016, pp. 630–645.

[63] M. Shafiq, Z. Gu, Deep residual learning for image recognition: A survey, Applied Sciences 12 (18) (2022) 8972.