



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación
y Desarrollo Tecnológico

Tesis de Doctorado

Reconstrucción 3D de terrenos para agrocultivo

presentada por

M.C. Ariel Eliezer Vazquez Domínguez

como requisito para la obtención del grado de

Doctor en Ciencias de la Computación

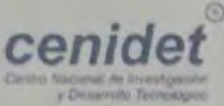
Director de tesis

Dra. Andrea Magadán Salazar

Codirector de tesis

Dr. José Ruiz Ascencio

Cuernavaca, Morelos, México. agosto de 2024

 Centro Nacional de Investigación y Desarrollo Tecnológico	ACEPTACIÓN DE IMPRESIÓN DEL DOCUMENTO DE TESIS DOCTORAL	Código: CENIDET-AC-006-D20
		Revisión: 0
	Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4	Página 1 de 1

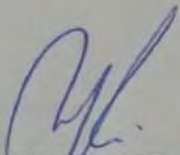
Cuernavaca, Mor., a 28 de junio de 2024

CARLOS MANUEL ASTORGA ZARAGOZA
 SUBDIRECTOR ACADÉMICO
 PRESENTE


AT'n: **JUAN GABRIEL GONZÁLEZ SERNA**
 PRESIDENTE DEL CLAUSTRO DOCTORAL

Los abajo firmantes, miembros del Comité Tutorial de la estudiante **ARIEL ELIEZER VÁZQUEZ DOMÍNGUEZ** manifiestan que después de haber revisado el documento de tesis titulado "RECONSTRUCCIÓN 3D DE TERENOS PARA AGROCULTIVO", realizado bajo la dirección de Andrea Magadan Salazar y codirección de José Ruíz Ascencio, el trabajo se **ACEPTA** para proceder a su impresión.

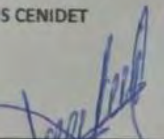
ATENTAMENTE



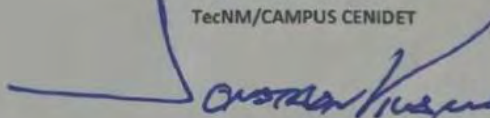
ANDREA MAGADÁN SALAZAR
 TecNM/CAMPUS CENIDET



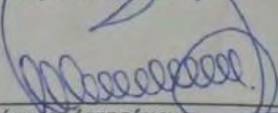
JOSÉ RUÍZ ASCENCIO
 TecNM/CAMPUS CENIDET



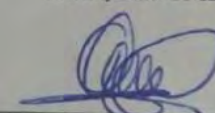
RAÚL PINTO ELÍAS
 TecNM/CAMPUS CENIDET



JONATHAN VILLANUEVA TAVIRA
 TecNM/CAMPUS CENIDET



MÁXIMO LÓPEZ SÁNCHEZ
 TecNM/CAMPUS CENIDET



JORGE ALBERTO FUENTES PACHECO
 IT CHILPANCINGO

c.c.p: María Elena Gómez Torres / Jefe del Departamento de Servicios Escolares
 c.c.p: Noé Alejandro Castro Sánchez / Jefe del Departamento C. Computacionales
 c.c.p: Expediente





Cuernavaca, Mor.,
No. De Oficio:
Asunto:

05/julio/2024
SAC/217/2024
Autorización de
impresión de tesis

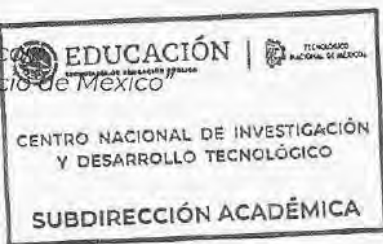
ARIEL ELIEZER VÁZQUEZ DOMÍNGUEZ
CANDIDATO AL GRADO DE DOCTOR
EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **“RECONSTRUCCIÓN 3D DE TERRENOS PARA AGRO CULTIVO”**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

*Excelencia en Educación Tecnológica
“Conocimiento y tecnología al servicio de México”*



CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

C. c. p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares

CMAZ/lmz



Dedicatoria

Dedico este trabajo a Dios, a mi esposa, a toda mi familia, a mis profesores y a todos mis compañeros, muchas gracias por su apoyo.

Agradecimiento

Al Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCYT) por el apoyo económico que me brindó durante mis estudios de doctorado.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET), por darme la oportunidad de estudiar y facilitarme sus instalaciones.

A mis asesores la Dra. Andrea Magadán Salazar y el Dr. José Ruiz Ascencio, por su tiempo, paciencia entre otras muchas cosas que me impulsaron mejorar cada día.

A mi comité revisor por sus comentarios y observaciones.

A mis compañeros por sus recomendaciones.

Resumen

El uso de la fotogrametría en los últimos años ha ido en aumento debido a que se ha popularizado el uso de drones por su bajo costo y gran versatilidad; sin embargo, para poder obtener una buena precisión es necesario contar con varios factores, como por ejemplo una buena cámara, un sensor de cinemática en tiempo real (Real Time Kinematic o RTK) y varios puntos de control terrestre bien colocados.

La presente tesis plantea una metodología novedosa para fotogrametría basada en técnicas de visión artificial para abordar el problema de reducir el error en el eje Z utilizando un dron con una cámara binocular.

El presente documento se integra de:

- La implementación de cinco diferentes calibraciones para la cámara binocular Zed 2.
- Una comparativa entre combinaciones de algoritmos para detectar, describir puntos característicos y algoritmos para su correspondencia.
- Una propuesta para generar co-correspondencias entre los puntos característicos presentes en la imagen monocular como en las imágenes binoculares, adquiridas en el mismo escenario.
- Propuesta para generar puntos de control virtuales mediante el uso de objetos con dimensiones conocidas dentro de la reconstrucción.
- También se presenta una metodología para poder generar los puntos utilizando la cámara binocular y evaluándolos a diferentes alturas y horarios, en entornos reales.

La propuesta de solución se evaluó inicialmente con escenarios simulados y posteriormente con imágenes adquiridas en tres escenarios reales. Los resultados muestran que la propuesta de solución es eficiente y el error disminuye en la mayoría de los casos; pero, el resultado está ligado a la posición geográfica de los puntos de control virtuales dentro de la reconstrucción, tal y como pasa con los puntos de control terrestre clásicos.

Abstract

Photogrammetry increased as drones gain popularity due to their affordability and versatility. However, achieving high accuracy requires several factors, including a quality camera, a real-time kinematic sensor (RTK), and strategically placed ground control points.

This thesis introduces a novel computer vision-based methodology to reduce the Z-axis error with a binocular camera.

The present document comprises the following sections:

- The implementation of five different calibrations for the Zed 2 binocular camera.
- A comparison between combinations of algorithms for detecting, describing minutiae, and matching algorithms.
- A proposal to generate co-correspondences between minutiae in the monocular image and in the binocular images acquired in the same scenario.
- A proposal to generate virtual control points by using objects with known dimensions within the reconstruction.
- A methodology is also presented for generating the points using the binocular camera and evaluating them at different heights and times in real environments.

The proposed solution was initially evaluated with simulated scenarios and subsequently with images acquired in three real scenarios. The results demonstrate that the proposed solution is efficient and that the error decreases in most cases. However, the result is linked to the geographical position of the virtual control points within the reconstruction, as is the case with the classic terrestrial control points.

Contenido

Resumen	v
Abstract	vi
Índice de figuras	x
Índice de tablas.....	xiii
Índice de ecuaciones	xiv
Glosario	xv
1. Introducción	1
1.1 Objetivos	1
1.2 Alcances y limitaciones.....	2
1.3 Estructura del documento.....	2
2. Trabajos relacionados.....	4
2.1 Antecedentes.....	4
Discusión.....	5
2.2 Estado del arte	5
2.2.1 Modelos digitales de elevación	5
2.2.2 Puntos característicos.....	7
2.2.3 Estado de la práctica.....	8
2.3 Discusión.....	13
3. Marco teórico	15
3.1 Algoritmos para detección y descripción de puntos característicos.....	15
3.1.1 SIFT [53]	15
3.1.2 ORB [52]	16
3.1.3 SuperPoint [59]	17
3.2 Algoritmos para correspondencia de puntos característicos	17
3.2.1 SuperGlue [68].....	17
3.2.2 Brute Force (BF) [65]	18
3.2.3 K-Nearest-Neighbours (KNN) [66]	18
3.2.4 KD-Tree-based nearest-neighbor (KD-TREE) [67]	19
3.3 Conceptos de fotogrametría	20
3.4 Structure From Motion (SFM)	21
3.4.1 Modelo proyectivo de la cámara	22
3.4.2 Geometría de los sistemas estéreo.....	25
3.4.3 Matriz Fundamental.....	26
3.4.4 Cálculo GSD ZED 2	29

3.5	Sistemas complementarios.....	29
3.6	Calibración de una cámara.....	32
3.7	Métricas.....	33
3.7.1	Raíz del error cuadrático medio (RMSE).....	33
3.7.2	CE90 y LE90.....	34
4.	Metodología de solución.....	35
4.1	Fotogrametría.....	35
4.2	Generación de puntos de control terrestre virtuales con cartulinas.....	36
4.3	Preprocesamiento con COLMAP.....	37
4.4	Re-proyección de nube de puntos a imagen.....	37
4.5	Algoritmo propuesto para generar puntos de control virtuales para la cámara binocular.....	38
4.6	Herramientas utilizadas (software y equipo).....	42
4.7	Equipo para adquisición de imágenes.....	42
5.	Pruebas y Resultados.....	45
5.1	Adquisición de Bancos de imágenes.....	45
5.1.1	Banco de imágenes 1.....	46
5.1.2	Banco de imágenes 2.....	46
5.1.3	Banco de imágenes 3.....	47
5.1.4	Banco de imágenes 4.....	48
5.1.5	Banco de imágenes 5.....	49
5.1.6	Banco de imágenes de ODM [93].....	51
5.1.7	Banco de imágenes simulado.....	51
5.2	Calibraciones primer caso de experimentación.....	53
5.2.1	Calibración con ROS.....	53
5.2.2	Calibración Stereolabs.....	55
5.2.3	Calibración OpenCV tamaño carta.....	56
5.2.4	Calibración OpenCV 13 cm.....	57
5.2.5	Comparación de las diferentes calibraciones.....	57
5.3	Caso 2 experimentación con puntos característicos (las 3 etapas).....	59
5.4	Caso 3 Co-correspondencias.....	67
5.5	Caso 4: Generación de puntos de control terrestre virtuales con cartulinas.....	70
5.5.1	Lectura de las tablas de resultados.....	70
5.5.2	Resultados con el banco de imágenes 1.....	70
5.5.3	Resultados con el banco de imágenes 2.....	71
5.5.4	Discusión.....	72

5.6	Caso 5: Preprocesamiento con COLMAP	72
5.7	Caso 6 Estabilidad de los puntos de control ante vibraciones y vegetación	73
5.8	Caso 7 Re-proyección de nube de puntos a imagen.....	74
6.	Conclusiones.....	77
6.1	Conclusiones	77
6.2	Objetivos logrados	78
6.3	Aportaciones.....	79
6.4	Trabajo futuro	80
6.5	Productos	80
	Referencias	81
	Anexos	93
	Anexo A: Resultados de correspondencias en una parcela con caña	93
	Anexo B: Resultados de re-proyección	96

Índice de figuras

Figura 2.1 Resultado de METASHAPE [72] en el banco de imágenes 4.....	8
Figura 2.2 Resultado de REALITYCAPTURE [73] en el banco de imágenes 4.	9
Figura 2.3 Resultado de MESHROOM [21] en el banco de imágenes 4.	10
Figura 2.4 Resultado PIX4D [71] en el banco de imágenes 4.	11
Figura 2.5 Resultado de OpenDroneMap [69] en el banco de imágenes 4.	12
Figura 3.1 Resultado de aplicar SIFT [53] en una imagen de ODM [93].....	16
Figura 3.2 Resultado de aplicar ORB [52] en una imagen de ODM [93].....	16
Figura 3.3 Resultado de aplicar SuperPoint [59] en una imagen de ODM [93].....	17
Figura 3.4 Flujo de trabajo de SuperGlue [4].	18
Figura 3.5 Ejemplo KD – TREE [97].....	20
Figura 3.6 GCP [75].....	21
Figura 3.7 Esquema que representa el proceso de SFM.	22
Figura 3.8 Sistemas de referencia [116].....	23
Figura 3.9 P y sistema D [18].....	24
Figura 3.10 Visión estéreo convergente.....	25
Figura 3.11 Geometría de cámara estéreo paralela o canónica [117].....	26
Figura 3.12 Imágenes creadas en UnrealEngine [119] con Airsim [120].	27
Figura 3.13 Ejemplo matriz F.....	27
Figura 3.14 Ejemplo matriz E.....	28
Figura 3.15 Ejemplo del cálculo de la posición.....	29
Figura 3.16 Ejemplo de un escenario AirSim [120] en modo Multirotor.	31
Figura 3.17 Ejemplo de un escenario en AirSim [120] en modo Computer Vision.	32
Figura 3.18 Tablero de ajedrez [130].	32
Figura 3.19 Mala precisión absoluta pero buena relativa.	33
Figura 3.20 Ejemplo de CE90 [131].	34
Figura 4.1 Esquema que representa la metodología de un proceso típico de fotogrametría, las figuras geométricas con borde en color azul son las fases que se proponen.	36
Figura 4.2 Preprocesamiento de las imágenes.	37
Figura 4.3 Esquema que representa el proceso de conversión de nube de puntos a imagen y a texto.	38
Figura 4.4 Esquema que representa el proceso de la cámara monocular para generar el TXT de la nube de puntos.....	40
Figura 4.5 Esquema que representa el proceso de la cámara binocular para generar los TXT de las nubes de puntos.	40

Figura 4.6 Esquema que representa el proceso para realizar las correspondencias entre la imagen monocular y las binoculares.	41
Figura 4.7 DJI MAVIC PRO PLATINUM [151].	43
Figura 4.8 DJI AGRAS T10 [152].	43
Figura 4.9 ZED 2 [153].	43
Figura 4.10 Jetson Nano [154].	44
Figura 5.1 Ejemplo de punto de control del primer vuelo.	46
Figura 5.2 Tres imágenes de muestra del conjunto de datos propio del primer vuelo adquirido con la cámara monocular.	46
Figura 5.3 Ejemplo de punto de control del segundo vuelo.	47
Figura 5.4 Tres imágenes de muestra del conjunto de datos propio del segundo vuelo adquirido con la cámara monocular.	47
Figura 5.5 Ejemplo de la imagen del lente izquierdo del tercer vuelo adquirido con la cámara binocular a 10 m.	48
Figura 5.6 Ejemplo de punto de control en el tercer vuelo.	48
Figura 5.7 Tres imágenes de muestra del conjunto de datos propio del tercer vuelo adquirido con la cámara monocular.	48
Figura 5.8 Cuatro imágenes de muestra del conjunto de datos propio del cuarto vuelo adquirido con la cámara monocular.	49
Figura 5.9 Ejemplo de punto de control [155].	49
Figura 5.10 Ejemplo de la imagen del lente izquierdo del quinto vuelo adquirido con la cámara binocular a 10 m.	50
Figura 5.11 Ejemplo de punto de control en el quinto vuelo.	50
Figura 5.12 Tres imágenes de muestra del conjunto de datos propio del quinto vuelo adquirido con la cámara monocular.	50
Figura 5.13 Dataset Quantix Mapper [93].	51
Figura 5.14 Mapa de Multi-biome [156].	52
Figura 5.15 Mapa de City Park SilverTM [157].	52
Figura 5.16 Escenario simulado propio.	53
Figura 5.17 Ventana del programa de calibración.	54
Figura 5.18 Calibración ZED.	54
Figura 5.19 Valores de calibración para cada lente.	55
Figura 5.20 Programa Zed Calibration.	56
Figura 5.21 Patrón tipo tablero de ajedrez tomado por la ZED 2.	56
Figura 5.22 Detección de esquinas de ajedrez OpenCV.	57
Figura 5.23 Patrón tipo tablero de ajedrez tomado por la ZED 2.	57
Figura 5.24 Error Zed 2.	59

Figura 5.25 Ejemplo de aplicación de ORB-BF en una imagen de ODM dataset.	60
Figura 5.26 Ejemplo de aplicación de ORB-KNN en una imagen de ODM dataset.	60
Figura 5.27 Ejemplo de aplicación de SIFT-BF en una imagen de ODM dataset.	60
Figura 5.28 Ejemplo de aplicación de SIFT-KNN en una imagen de ODM dataset.	61
Figura 5.29 Ejemplo de aplicación de SIFT-KD-TREE en una imagen de ODM dataset. .	61
Figura 5.30 Ejemplo de aplicación de SuperPoint en una imagen de ODM dataset.	62
Figura 5.31 Ejemplo de aplicación de ORB-BF en una imagen del banco de imágenes 4.	63
Figura 5.32 Ejemplo de aplicación de ORB-KNN en una imagen del banco de imágenes 4.	63
Figura 5.33 Ejemplo de aplicación de SIFT- BF en una imagen del banco de imágenes 4.	64
Figura 5.34 Ejemplo de aplicación de SIFT- KNN en una imagen del banco de imágenes 4.	64
Figura 5.35 Ejemplo de aplicación de SIFT-KD-TREE en una imagen del banco de imágenes 4.	65
Figura 5.36 Ejemplo de aplicación de SuperPoint-SuperGlue 1/4 en una imagen del banco de imágenes 4.	65
Figura 5.37 Ejemplo de aplicación de SuperPoint-SuperGlue en una imagen del banco de imágenes 4.	66
Figura 5.38 La imagen izquierda corresponde al lente izquierdo del primer par binocular, la imagen central y derecha corresponden al segundo par binocular para cada escenario, el primer renglón presenta al escenario de Multi-biome, el segundo renglón muestra a City Park y el último renglón contiene el escenario propio.	68
Figura 5.39 Correspondencias de dos pares de imágenes para cada escenario, el primer renglón presenta el escenario de Multi-biome, el segundo renglón muestra a City Park y el último renglón contiene el escenario propio simulado.	69
Figura 5.40 Correspondencias en común entre las 3 imágenes para cada escenario, primer escenario es el de Multi-biome, seguido por City Park y al último el escenario propio simulado todos del banco de imágenes simulado.	69
Figura A.1 Resultado promedio de correspondencia binocular monocular.	93
Figura A.2 Uno de los mejores resultados de correspondencia binocular monocular.	94
Figura A.3 Ejemplo de la fuerza del viento generado por el DJI T10.	95
Figura B.1 Ejemplo de re-proyección 1.a 0.4.	96
Figura B.2 Ejemplo de re-proyección 2. Figura 0.5.	97
Figura B.3 Ejemplo de re-proyección 3. Figura 0.6.	98

Índice de tablas

Tabla 2.1 Comparación de algoritmos de correspondencia [7].....	4
Tabla 2.2 Artículos que reportan el uso de técnicas de fotogrametría.....	6
Tabla 2.3 Sistemas para fotogrametría.	13
Tabla 2.4 Costos de sistemas para fotogrametría.....	13
Tabla 5.1 Medición de distancias con la ZED 2 utilizando diferentes calibraciones.....	58
Tabla 5.2 Resultados promedio de los algoritmos de detección y correspondencia con el dataset ODM.	62
Tabla 5.3 Resultados promedio de los algoritmos de detección y correspondencia con el banco de imágenes 4.....	66
Tabla 5.4 Resultados de utilizar puntos de control terrestre en banco de imágenes 1, DONDE S significa sin puntos de control, C significa considerando puntos de control y la columna DIF presenta la diferencia entre ambos.	71
Tabla 5.5 Resultados de utilizar puntos de control terrestre en banco de imágenes 2.	71
Tabla 5.6 Resultados de las reconstrucciones con preprocesamiento de COLMAP en el banco de imágenes 1.....	73
Tabla 5.7 Resultados de las reconstrucciones con preprocesamiento de COLMAP en el banco de imágenes 2.....	73
Tabla 5.8 Resultados de correspondencias con y sin estabilizador mecánico.....	74
Tabla 5.9 Resultados VGCP'S conjunto de imágenes 3.....	75
Tabla 5.10 Resultados VGCP'S conjunto de imágenes 5.....	75
Tabla 6.1 Solución por objetivo.....	78

Índice de ecuaciones

Ecuación 3.1	22
Ecuación 3.2	22
Ecuación 3.3	23
Ecuación 3.4	23
Ecuación 3.5	24
Ecuación 3.6	24
Ecuación 3.7	24
Ecuación 3.8	24
Ecuación 3.9	24
Ecuación 3.10	25
Ecuación 3.11	26
Ecuación 3.12	26
Ecuación 3.13	27
Ecuación 3.14	27
Ecuación 3.15	27
Ecuación 3.16	27
Ecuación 3.17	28
Ecuación 3.18	28
Ecuación 3.19	29
Ecuación 3.20	29
Ecuación 3.21	33
Ecuación 3.22	34
Ecuación 5.1	70

Glosario

Acrónimo	Original	Español
DEM	Digital Elevation Model.	Modelo digital de elevación .
2D	2 Dimensiones.	2 dimensiones.
3D	3 Dimensiones.	3 dimensiones.
BF	Brute Force.	Fuerza bruta.
BRIEF	Binary Robust Independent Elementary Features.	Características elementales independientes robustas binarias.
BRISK	Binary Robust invariant scalable keypoints.	Puntos clave binarios robustas invariantes escalables.
CENIDET	Centro Nacional de Investigación y Desarrollo Tecnológico.	Centro Nacional de Investigación y Desarrollo Tecnológico.
cm	Centímetros.	Centímetros.
DEM	Digital Elevation Model.	Modelo digital de elevación .
FAST	Features From Accelerated Segment Test.	Características de la prueba de segmentos acelerada.
GCP	Ground Control Point.	Punto de control de tierra.
GNSS	Global navigation satellite system.	Sistema global de navegación por satélite.
GPS	Global Positioning System.	Sistema de posicionamiento global.
IMU	Inertial Measurement Unit.	Unidad de medición inercial.
JCR	Journal Citation Reports.	Informes de citas de revistas.
KD-TREE	KD-Tree-based-Nearest-neighbor.	KD-Tree-based-Nearest-neighbor.
KNN	K-Nearest-Neighbours.	K-Nearest-Neighbours.
m	Metros.	Metros.
mm	Milímetros.	Milímetros.
MVS	Multi-View Stereo.	Múltiples vistas estéreo.
ODM	Open Drone Map.	Mapeo con drones libre.
OpenCV	Open Source Computer Vision Library.	Librería de visión artificial.
ORB	Oriented FAST and Rotated BRIEF.	FAST orientado y BRIEF girado.
RMSE	Root Mean Square Error.	Raíz del error medio cuadrático.
RPAS	Remotely Piloted Aircraft System.	Sistema de aeronaves pilotadas por control remoto.
RTK	Real Time Kinematic.	Navegación cinética satelital en tiempo real.
SFM	Structure from Motion.	Estructura desde el movimiento.
SIFT	Scale-Invariant Feature Transform.	Transformación de característica invariante de escala.
SIFT	Scale-Invariant Feature Transform.	Transformación de características invariantes de escala.
SURF	Speeded up robust features.	Características robustas aceleradas.
TLS	Terrestre laser sensor.	Sensor laser terrestre.
UAV	Unmanned Aerial Vehicle.	Vehículos aéreos no tripulados.
VGCP	Virtual Ground Control Point.	Punto de control de terrestre virtual.

Capítulo 1: Introducción

Existen diversas formas para poder generar un modelo de elevación de un terreno, técnicas como las clásicas (un topógrafo con su equipo especial) hasta técnicas más recientes que realizan este proceso de manera digital y con mayor precisión como es el uso de drones, vehículos aéreos no tripulados (UAV por sus siglas en inglés), avionetas, sensores laser terrestres, etc., [1-5].

El uso de aviones adicionados con cámaras para generar un modelo digital de elevación (DEM por sus siglas en inglés) es unos de los procesos de fotogrametría.

En tareas como la construcción o como en la irrigación de agrocultivos, es necesario tener la mayor precisión posible, especialmente en el eje Z, debido a que un pequeño cambio puede generar resultados completamente diferentes, para lograr esto con un dron existen soluciones, pero son muy costosas y aun así no son capaces de mantener una buena precisión en todos los casos.

Aunque los drones se han vuelto cada vez más populares debido a su accesibilidad y versatilidad, es posible que los modelos de elevación obtenidos con drones no alcancen consistentemente niveles satisfactorios de precisión. Con la finalidad de aumentar la precisión existen técnicas tales como el uso de puntos de control terrestres (GCP) o sensores extras tales como un sensor de cinemática en tiempo real (Real Time Kinematic o RTK).

En esta tesis se propone el uso de un dron equipado con una cámara binocular, un sistema de posicionamiento global (GPS) en combinación con técnicas de estructura a partir del movimiento (Structure from Motion o SFM) tanto monoculares y binoculares, para generar DEM con un error cuadrático medio (RMSE) menor a 10 cm a 10 m de altura, esto debido a que un error menor a 10 cm es aceptable, tal como lo menciona [6], que un buen rango es de 5-10 cm en los ejes X, Y, en el eje Z es de 10-15 cm.

1.1 Objetivos

A. Objetivo general

Proponer, aplicar y evaluar un algoritmo para la generación de modelos digitales de elevación, mediante el uso de una cámara binocular montada en un dron, con la finalidad

de tener una precisión menor a los 10 cm de RMSE en el eje z, a una altura aproximada de 10 m.

B. Objetivos específicos

- Estudiar las técnicas de estructura a partir del movimiento, visión binocular y múltiples vistas estéreo más utilizadas.
- Seleccionar y aplicar varias de estas técnicas para conocer su rendimiento.
- Proponer y/o modificar un algoritmo para la generación de un modelo digital de elevación.
- Combinar la información de la una cámara binocular (montada en un dron) con la proporcionada por sensores GPS, RTK y el uso de técnicas SFM para generar modelos digitales de elevación.
- Evaluar el sistema propuesto.
- Publicar un artículo índice JCR.

1.2 Alcances y limitaciones

A continuación, se describen los alcances y las limitaciones identificadas para el desarrollo de esta tesis:

A. Alcances

- Se revisan al menos 4 diferentes configuraciones que consideren técnicas de estructura a partir del movimiento (SFM), múltiples vistas estéreo (MVS) y visión estéreo para la generación de modelos digitales de elevación (DEM).
- Se compara el modelo propuesto contra las técnicas que mejor rendimiento reporten en el estado del arte.
- El modelo propuesto es capaz de generar un DEM con un error cuadrático medio (RMSE) menor a 10 cm.

B. Limitaciones:

- No se consideran límites de tiempo para el procesamiento.
- Las imágenes de trabajo son adquiridas mediante una cámara binocular y una monocular.
- La altura máxima del vuelo del dron binocular estará controlada en un rango de 5 m a 10 m y la altura del dron monocular entre 50 m y 100 m.
- Las imágenes son tomadas dentro de terrenos de cultivos en los cuales no se encuentran vegetación de gran tamaño, sin embargo, en algunos de ellos si contienen árboles.

1.3 Estructura del documento

En los próximos capítulos se da una descripción detallada de los procedimientos empleados para la realización del presente trabajo de investigación. La estructura del documento está conformada de la siguiente manera:

- Capítulo 1: Contiene la introducción y el análisis del presente trabajo.

- Capítulo 2: Contiene los antecedentes del presente trabajo, la revisión del estado del arte en temas como modelos digitales de elevaciones y puntos característicos; además en el capítulo se presenta el estado de la práctica.
- Capítulo 3: Contiene el marco teórico en temas como algoritmos para detección y correspondencia de puntos característicos, conceptos de fotogrametría, calibración de la cámara y las métricas utilizadas.
- Capítulo 4: Contiene el diseño e implementación de la solución del problema de tesis.
- Capítulo 5: Contiene el diseño y las pruebas que se realizaron para cada módulo del desarrollo de la tesis, además del análisis de los resultados obtenidos.
- Capítulo 6: Contiene las conclusiones generales, la discusión, trabajos futuros y las aportaciones.
- Anexos: Contiene las experimentaciones adiciones realizadas en el transcurso del doctorado, por ejemplo, los pasos para instalar Zed en ROS todo en el sistema operativo de Ubuntu; además de los resultados de re-proyección de puntos de la nube de puntos monocular.

Capítulo 2: Trabajos Relacionados

El capítulo presenta la revisión de tesis desarrolladas en el CENIDET, relacionadas al tema de tesis; se presenta un análisis del estado del arte relacionado a los conceptos de modelos digitales de elevación y puntos característicos y, por último, se presenta el estado de la práctica y una discusión del capítulo.

2.1 Antecedentes

En el grupo de inteligencia artificial del CENIDET, se han desarrollado algunas tesis sobre visión robótica utilizando cámaras binoculares y por lo tanto conforman los antecedentes para esta tesis. A continuación, se mencionan los trabajos más relevantes.

- Visión binocular para robot móvil en exteriores simulados [7]

En esta tesis, el autor desarrolló un robot móvil con piezas de baja precisión, capaz de detectar objetos y esquivarlos, en un ambiente de interiores con objetos de exteriores (exteriores simulados). Utilizó dos cámaras Logitech para generar una visión binocular. Estudia tres algoritmos de correspondencia: BM (*Block Matching*) [8], [9], SGBM (*Semi Global Block Matching*) [8], [10] y BTM (*Birchfield-Tomasi sampling-insensitive intensity local matcher*) [11], [12]. Menciona que el mejor algoritmo de correspondencia es el BTM, como se muestra en la Tabla 2.1. Para el análisis de las imágenes, realiza la calibración de la cámara, la rectifica y calcula la triangulación.

Tabla 2.1 Comparación de algoritmos de correspondencia [7].

Algoritmo	Rapidez	Falsos positivos	Falsos negativos	Precisión
BM [8], [9]	94 ms	Desaparecen con un bloque de 41 píxeles	En obstáculos mucho más pequeños que el robot (por ejemplo: las patas delgadas de las sillas)	Mínima precisión
SGBM [8], [10]	469 ms	Son pocos, aparecen y desaparecen, no se pudieron eliminar	En obstáculos mucho más pequeños que el robot (por ejemplo: las patas delgadas de las sillas)	Precisión media
BTM [11], [12]	5922 ms	Son muchos, no se intentaron eliminar	No se detectaron	Máxima precisión

- Fusión de Información de Pose en Robots Móviles con Visión [13]

En esta tesis el autor presenta un sistema capaz de calcular la pose de robots móviles, fusionando la odometría visual con la odometría mecánica. La parte relevante de esta tesis para el presente proyecto son sus experimentaciones sobre visión binocular y calibración.

- Navegación, Localización y Mapeo de Robots Móviles para Trayectorias Preespecificadas por Imágenes [14].

En esta tesis se presenta un sistema capaz de generar una trayectoria desde un video de entrenamiento, para posteriormente desplazarse dentro de esa trayectoria de modo autónomo. Utiliza dos cámaras monoculares montadas en un arreglo binocular como única fuente de información y el robot Robotino. El sistema consta de tres fases: entrenamiento, localización y navegación autónoma. Realiza una calibración tanto monocular como binocular.

Discusión

Se realizó un estudio sobre los trabajos de tesis ya presentados y se nota que no existe un antecedente directo para este tema de tesis. Las tesis detallan información sobre visión binocular, apoyando el proceso de entender los conceptos necesarios, el cómo funcionan este tipo de cámaras, y cómo llevar a cabo procesos tales como; las calibraciones de las cámaras; aspectos principales en el presente trabajo.

2.2 Estado del arte

La fotogrametría es un conjunto de técnicas que han ido aumentando su demanda en los últimos años, debido al auge de los drones comerciales equipados con cámaras 4k a costos accesibles.

En la presente sección, se describen los conceptos básicos de fotogrametría con drones, así como las metodologías que son mayormente empleadas para dar solución a la reconstrucción 3D con imágenes obtenidas de drones. El capítulo se encuentra dividido en dos secciones que son: Modelos digitales de elevación y sistemas para detección y correspondencia de puntos característicos, en los cuales se incluyen artículos que fueron relevantes en el transcurso de la investigación y una recopilación de sistemas que actualmente están en el mercado, los cuales están dedicados a la fotogrametría.

2.2.1 Modelos digitales de elevación

Para generar un modelo de elevación de un terreno (DEM por sus siglas en inglés), existen dos métodos [15]: a) los métodos directos que generalmente se llevan a cabo mediante un topógrafo con su equipo especial [16] (estación total, por ejemplo). Y b) los métodos indirectos, que son herramientas más recientes que realizan este proceso de manera digital y con mayor precisión mediante el uso de sensores láser terrestres, drones, vehículos aéreos no tripulados, avionetas, satélites, etc., [2], [4], [15], [17].

Para la presente investigación se utilizaron drones para generar los modelos digitales de elevación, con una técnica llamada fotogrametría, que significa literalmente 'medir de fotos' y se trata de una técnica para determinar las propiedades geométricas de objetos en escenas tridimensionales a partir de imágenes fotográficas [18], la cual permite medir con

cierta precisión las distancias entre dos puntos dentro de la zona a muestrear. La fotogrametría se puede categorizar en 2, las que ocupan técnicas clásicas como estructura a partir del movimiento [19-21] y las que ocupan redes neuronales [22-30]. En esta investigación, se trabajó con la metodología clásica.

Para generar un modelo digital de elevación existen 5 variables físicas esenciales y 2 computaciones, las cuales son: a) altura, b) cámara, c) el uso, cantidad y posición de GCPs, d) sensores como RTK o Lidar, e) traslape entre fotos f) métricas y g) el algoritmo o programa utilizado.

Para una correcta configuración de las variables antes mencionadas, se realizó una búsqueda en el estado del arte. Los resultados se muestran en la Tabla 2.2, donde el “*” señala que el autor no reportó ese dato.

Tabla 2.2 Artículos que reportan el uso de técnicas de fotogrametría.

Ref.	Altura	Cámara	GCP	Sensores	Traslape	Métricas	Algoritmo	RMSE
[31]	100-200 m	4000X3000	4	PPK Y RTK	*	RMSE	Pix4D	6.3-13.9 cm
[32]	900-4500 m	6600 X4400	0	*	*	RMSE - MAX E	PIX4D Metashape	1.08-9.77 m
[33]	170-180 m	5472 x 3648	9	PPK Y RTK	80%	RMSE	Metashape Photoscan	4-15 cm
[34]	120-300 m	7360 x 4912	43	*	90%	RMSE	Varios	0.2-2.7 m
[35]	*	DJI Phantom 3 Professional		*	*	RMSE	OpenSFM	*
[36]	80 m	DJI® spark	0	*	90%	RMSE	Pix4D	97 cm
[37]	50 m	DJI Phantom 4 Pro	0	*	85%	RMSE	Pix4D	*
[6]	61 m	DJI Mavic 2 Pro	9	*	75%	RMSE	Pix4D	21-96 cm
[38]	50 m	4 diferentes cámaras	7	*	*	RMSE	Pix4D	3-88 cm
[39]	70 m	DJI Mavic Pro Platinum	21	*	70%	RMSE	Pix4D	4.2-6.2 cm
[40]	50-100 m	DJI Phantom IV y DJI Phantom IV RTK	0	RTK	80%	RMSE	Pix4D	2.8-15.6 cm
[41]	120 m	Sensefly eBee Plus	34	*	75%	RMSE	Pix4D	0.2-1.1 m
[42]	4-6 m	DJI Mavic Air	15	*	65%	MBE RMSE MAPE	Pix4D	*
[43]	75 m	DJI Phantom4	15-20	*	80%	RMSE y otras para el pasto	PIX4D	8.1-14.6 cm
[44]	75-120 m	(DJI Mavic 2 Pro)	9	*	80%	RMSE	PIX4D	5.5-31.6 cm
[45]	75 m	Phantom 4 Pro	8	*	80%	RMSE, MEAN MEDIAN STD MIN Y MAX	Metashape	12.8-43cm
[46]	40-50-60 m	DJI Phantom 4 RTK	12	RTK	80%	RMSE	Metashape	0.16-0.76 cm
[47]	60, 70, 80 m	DJI Phantom 4 RTK UAV	10	PPK, RTK y SonarMite BTX SBES	80%	Diferencia entre medido por el sonar y lo calculado por SFM	Pix4D	*
[48]	40-50 m	DJI Phantom 3 Advanced	20		80%	RMSE	Pix4D	18.2-11.3 cm
[49]	110 m	FEIMA D2000 Multirotor UAV SONY a6000	16	PPK Y RTK	*	RMSE and M3C2	Photoscan	1.5-8.7 cm

La Tabla 2.2 muestra que los parámetros más comunes son: una altura menor a 100 m, con un dron comercial de la marca DJI, utilizando 9 puntos de control terrestre, sin algún

sensor extra a menos que el ambiente lo exija (ejemplo [47]), un traslape del 80%, con métrica principal RMSE y con el algoritmo de Pix4D.

2.2.2 Puntos característicos

Los puntos característicos son una parte fundamental en el proceso de fotogrametría, debido a que estos son la nube de puntos escasa, en otras palabras, son la primera etapa del proceso. Estos se dividen en 3 etapas, las cuales son: a) detección, b) descripción y c) correspondencia.

a) detección

Es la etapa inicial para localizar los puntos característicos. A pesar de que los métodos modernos combinan la detección y la descripción en un solo algoritmo, como por ejemplo FAST [50] y Harris [51] los cuales son específicos para esta labor. Esta actividad consiste en, en como su nombre lo indica, detectar los puntos característicos, los cuales pueden ser por ejemplo una esquina o un cambio de intensidad, ubicándolos dentro de la imagen.

b) descripción

Esta etapa consiste en describir o codificar los puntos detectados en la etapa anterior, este proceso varía según el tipo del algoritmo utilizado, por ejemplo, ORB [52] el cual es binario y tiene un descriptor con un tamaño de 32 bytes, o utilizando el algoritmo SIFT [53] que tiene un descriptor de tamaño 128 bytes. De la misma manera que la etapa anterior, existen algoritmos especializados en la descripción como, por ejemplo, BRIEF [54].

Sin embargo, la tendencia son algoritmos que realizan las dos etapas, algunos de los algoritmos más conocidos son: SIFT [53], ORB [52], SURF [55], BRISK [56] y KAZE [57].

En los últimos años se han desarrollado algoritmos para detectar y describir puntos característicos en una sola pasada utilizando redes neuronales, algunos de ellos son: DetNet [58], SuperPoint [59], TILDE [60], LIFT [61], LF-NET [62], SIP'S [63] y por último la primera red neuronal para detectar puntos característicos "A NEURAL NETWORK BASED CORNER DETECTION METHOD" [64].

c) correspondencia

Es la última etapa de los puntos característicos. Su objetivo es a partir de la imagen 1 encontrar dentro de la otra imagen, el punto característico de consulta. El proceso es un reto debido a que no siempre la imagen tiene un movimiento lineal, sino que en muchos casos se presenta una combinación de traslación, rotación y cambio de escala, lo que aumenta significativamente la complejidad del problema. Los algoritmos para correspondencia se pueden dividir en dos subsecciones, los clásicos como Brute Force (BF) [65], K-Nearest-Neighbours (KNN) [66] y KD-Tree-based-Nearest-neighbor (KD-TREE) [67] y los que utilizan redes neuronales.

Los que utilizan redes neuronales, han estado reportando mejores resultados que los clásicos, un ejemplo es SuperGlue [68], el cual no solamente toma en cuenta un punto característico y lo busca dentro de la otra imagen, sino que, también incluye información de sus vecinos para que de esta manera obtener mayor precisión.

2.2.3 Estado de la práctica

En el estado de la práctica se localizaron varios sistemas gratuitos que realizan la reconstrucción de imágenes adquiridas con drones. Estos son: MESHROOM [21], OpenDroneMap [69] y OPENSFM [70]. De los sistemas comerciales más utilizados se encuentran: PIX4D [71], METASHAPE [72] y REALITYCAPTURE [73].

Como entrada mínima a cada sistema se le tienen que dar 3 imágenes en RGB, que tengan una superposición deseable del 65-85%. Una superposición más cercana al 85% aumenta la precisión, pero aumenta el número de imágenes necesarias para cubrir toda la zona a muestrear. A continuación, se detalla brevemente cada sistema:

METASHAPE [72]

Agisoft Metashape [72] (anteriormente conocido como Agisoft Photoscan) es un sistema de fotogrametría de pago, el cual está disponible para Windows, MacOS y Linux. Tiene una interfaz amigable, de fácil uso y muy completa. Es decir, con una gran cantidad de funciones. Sin embargo, no proporciona información sobre qué algoritmo utiliza en cada una de sus etapas del procesamiento. La Figura 2.1 muestra un ejemplo de una reconstrucción utilizando METASHAPE [72] en el banco de imágenes 4, el cual es presentado en la sección 5.1.4.

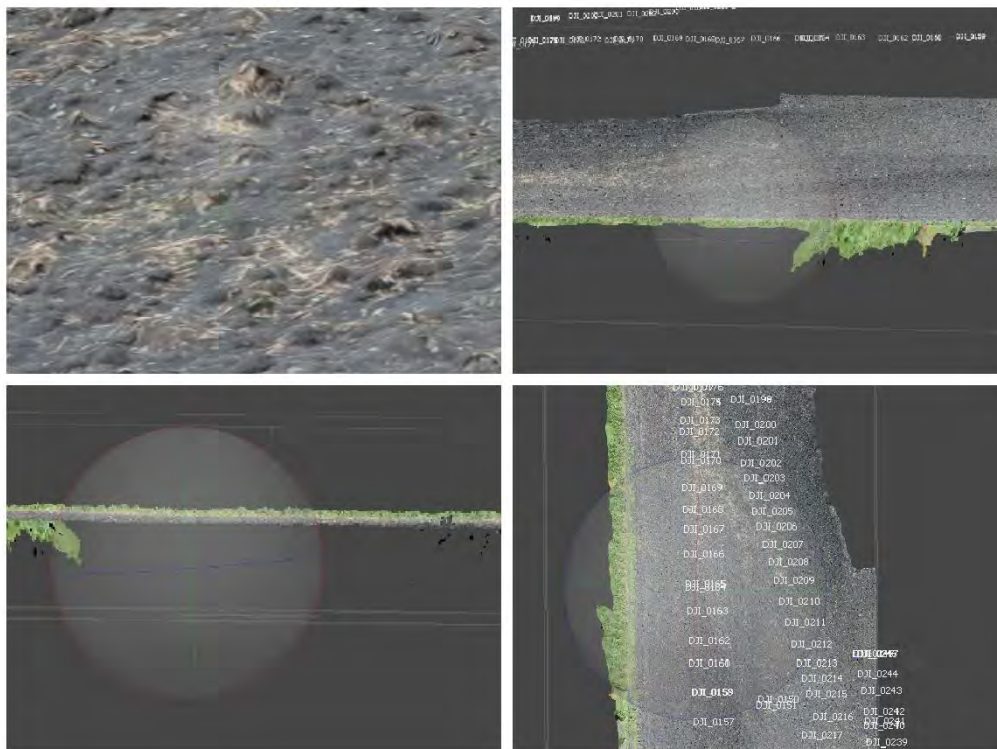


Figura 2.1 Resultado de METASHAPE [72] en el banco de imágenes 4.

REALITYCAPTURE [73]

RealityCapture es un software de escaneo 3D/fotogrametría para Windows que permite crear modelos 3D hiperrealistas para proyectos a partir de un conjunto de fotos o escáneres láser. Es un sistema de fotogrametría de pago, que contiene una interfaz con una mayor cantidad de opciones para configurar, pero más complicada que la proporcionada por METASHAPE [2] y de la misma forma no comparte información de los algoritmos que utiliza en cada momento. La Figura 2.2 muestra un ejemplo de una reconstrucción utilizando REALITYCAPTURE [73] en el banco de imágenes 4.

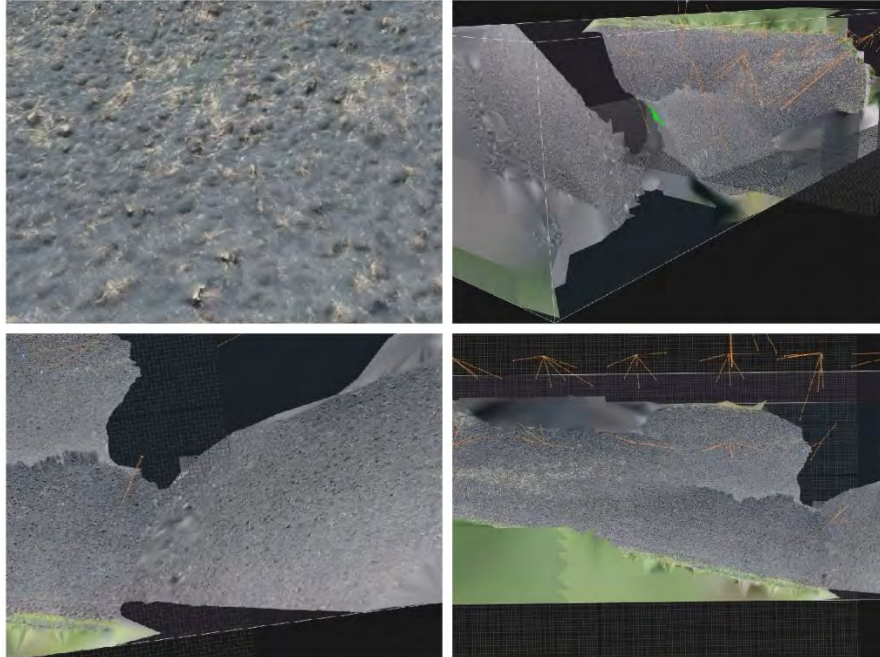


Figura 2.2 Resultado de REALITYCAPTURE [73] en el banco de imágenes 4.

MESHROOM [21]

Es un sistema de fotogrametría libre, el cual permite modificar una gran cantidad de parámetros. Entre ellos, permite seleccionar los algoritmos a utilizar, sin embargo, no deja realizarles modificaciones o añadir más algoritmos. Su interfaz es amigable, no obstante, no acepta el uso de GCP, lo cual limita mucho su precisión y por lo tanto su uso en fotogrametría. La Figura 2.3 muestra un ejemplo de una reconstrucción utilizando MESHROOM [21] en el banco de imágenes 4.

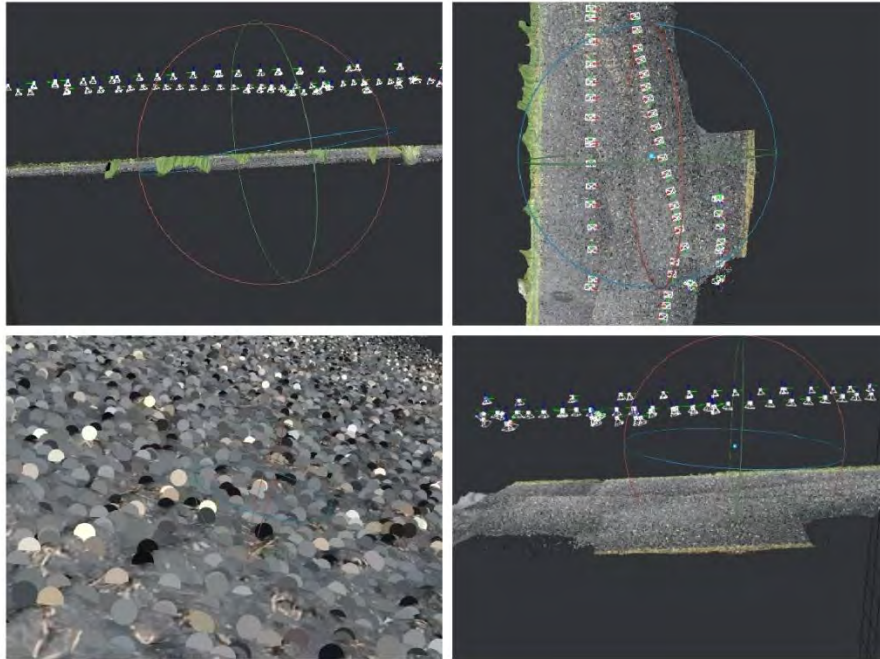


Figura 2.3 Resultado de MESHROOM [21] en el banco de imágenes 4.

PIX4D [71]

En el estado del arte PIX4D es el software más citado además de mencionar que es la herramienta que mejor precisión tiene de acuerdo con [74-81]. PIX4D es fácil de utilizar, tiene una gran cantidad de herramientas para el usuario final, además de estar optimizado para los drones DJI. El sistema PIX4D localiza correctamente el modelo del dron y añade todos los parámetros necesarios de la cámara al proceso. La Figura 2.4 muestra un ejemplo de una reconstrucción utilizando PIX4D [71] en el banco de imágenes 4.

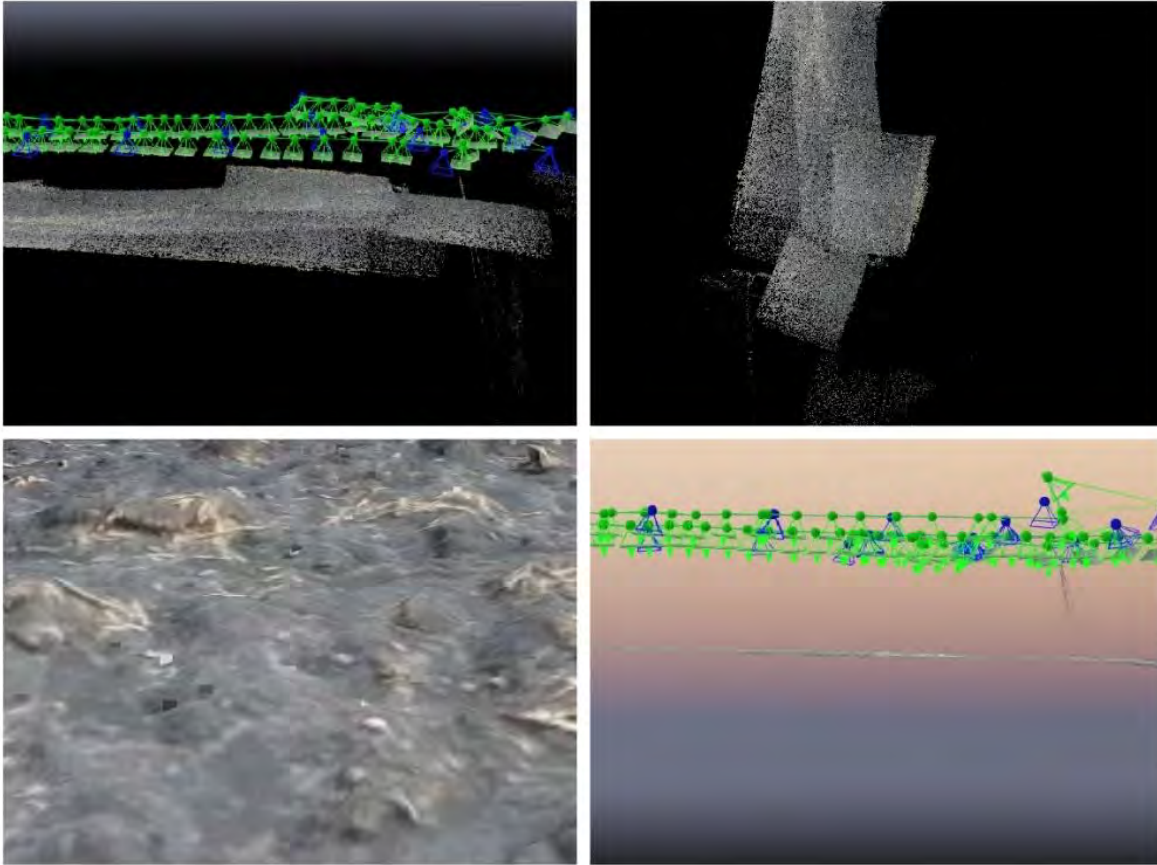


Figura 2.4 Resultado PIX4D [71] en el banco de imágenes 4.

OpenDroneMap [69]

Es uno de los sistemas de fotogrametría “*opensource*” más utilizados, ya que acepta el uso de GCP. Tiene una interfaz amigable, pero con muchas opciones y configuraciones de los algoritmos a utilizar, es de código abierto y recibe actualizaciones constantemente. Además, tiene una opción de realizar todo el proceso por consola. La Figura 2.5 muestra un ejemplo de una reconstrucción utilizando *OpenDroneMap* (ODM) [69] en el banco de imágenes 4.



Figura 2.5 Resultado de OpenDroneMap [69] en el banco de imágenes 4.

OPENSFM [70]

OPENSFM [70] es un sistema de fotogrametría muy popular de uso libre. Es la base para OpenDroneMap (ODM por sus siglas en inglés), sin embargo, no cuenta con tantas herramientas visuales y en algunas ocasiones no es capaz de lograr realizar una reconstrucción fotogramétrica, como es el caso con en el banco de imágenes 4.

Comentarios del estado de la práctica

Los sistemas de pago no proporcionan información alguna en sus versiones de prueba. La información que dejan guardar en otro formato y que ellos comparten es mínima, y no mencionan respecto a los algoritmos en cada etapa, tampoco permiten guardar los modelos digitales sino se tiene una versión de pago, solo dejan visualizarlo.

En el mercado, se pueden encontrar ya sistemas comerciales para realizar fotogrametría. Como se puede observar en la Tabla 2.3, actualmente Pix4D es el mayor referente al momento de elegir un programa para realizar fotogrametría con drones. Tiene una precisión aproximada de 7.7 cm de RMSE [76] pero este valor varía dependiendo de factores como la cámara utilizada, cuantos GCP se utilizaron, la precisión de los GCP, la altura de vuelo, la precisión del resto de sensores, etc.

Tabla 2.3 Sistemas para fotogrametría.

Artículo	Programa
3D mapping efficacy of a drone... [1]	Pix4D Mapper Pro
Accuracy of UAS... [75]	Pix4Dmapper Pro
Aplicación de algoritmos Structure... [82]	Photoscan Pro
Optimising UAV topographic surveys... [83]	Photoscan
A Statistical Examination of Image Stitching... [76]	Pix4D
Suas, sfm-mvs photogrammetry... [77]	Pix4D
Assessing suitability of modified... [78]	Pix4D
3-D uncertainty-based topographic... [84]	Photoscan Pro
Evaluating the accuracy of dem... [79]	Pix4UAV Desktop
The accuracy of automatic... [80]	Pix4D
Multi-temporal UAV data... [81]	Pix4D
Quantifying submerged fluvial... [85]	Photoscan
Accuracy analysis of a 3D model... [4]	Photoscan
Automated extraction of... [86]	Photo Modeler Scanner

En la Tabla 2.4 se muestran los precios de algunos programas para fotogrametría, el último es ODM, el cual es de licencia libre pero aún está en estado beta [3], [69].

Tabla 2.4 Costos de sistemas para fotogrametría.

Programa	Mensualidad	Anualidad
Pix4D Mapper [87] (Sin complementos)	350 USD	3500 USD
DroneDeploy [88] (Limitada)	499 y 599 USD	3948 USD
3DF Zephyr Aerial [89]	250 € + IVA	4200 € + IVA
Photoscan / Metashape [72]	179 USD	3499 USD
MESHROOM [21]	Licencia libre	Licencia libre
REALITYCAPTURE [73]	Lo que se use (créditos) o 3700 USD por licencia perpetua.	
OpenDroneMap [69]	Licencia libre	Licencia libre

2.3 Discusión

En las secciones anteriores se revisaron metodologías para la generación de modelos digitales de elevación, así como algoritmos para detectar, describir y los que realizan correspondencia de puntos característicos. Sin embargo, no se abordó el tema de fotogrametría utilizando una cámara binocular como una monocular y menos se explican los pasos necesarios para generar puntos de control terrestre con la cámara binocular, para añadir información extra a la reconstrucción. Eso se debe a que no se encontró información en el estado del arte, lo que más se aproxima es la información proporcionada en los artículos [90], [91].

En [90] se detalla el trabajo de un robot móvil, que utiliza una cámara binocular para generar una nube de puntos en tiempo real, pero con una profundidad x máxima y cada cierto

número de imágenes las procesa con estructura a partir del movimiento para obtener una profundidad o distancia mayor.

En el trabajo [91], el autor combina las ventajas de usar una cámara de luz estructurada con un algoritmo de SFM, logrando detectar zonas lejanas, cercanas y la escala. Además de explicar los problemas que surgen al unir estas dos técnicas, especialmente, que una tiene escala y la otra no, así que se tiene que realizar una transformación.

A pesar de que los dos artículos mencionados no tienen una relación completamente con el tema de la presente tesis, son una prueba de que se ha trabajado con la idea de fusionar algoritmos para corto y largo alcance. En la presente tesis se trabaja con la idea de añadirle información extra, proveniente de la cámara binocular la cual tiene una distancia corta de detección, a los algoritmos como estructura a partir del movimiento (SFM por sus siglas en inglés) que es un algoritmo cuya distancia de detección es mayor.

Capítulo 3:

Marco Teórico

En el presente capítulo se describen los conceptos básicos, técnicas y algoritmos necesarios para la comprensión de la presente tesis; además de las métricas necesarias para la evaluación de los resultados.

3.1 Algoritmos para detección y descripción de puntos característicos

3.1.1 SIFT [53]

The Scale Invariant Feature Transform (SIFT) fue introducido por Lowe [53] en 1999 y la idea principal es poder encontrar puntos invariantes a la traslación, el escalado y la rotación, dentro del espacio-escala de la imagen. Se realiza en cuatro etapas, la primera es la detección de puntos extremos en el espacio escala, la segunda es la localización exacta de los puntos destacados, la tercera es la asignación de la orientación y la cuarta es la descripción de puntos destacados. En otras palabras, SIFT realiza una diferencia de gaussianas (DOG) para la detección de los puntos característicos y realiza la descripción de estos puntos mediante un vector de 128 características. Sin embargo, se pueden utilizar por separado ambas fases.

Un ejemplo de su uso se puede encontrar en la tesis titulada “Comparación de Algoritmos de Extracción y Asociación de Rasgos Para Visión Robótica” [92] donde comparan el desempeño de Hessian affine, Harris affine, MSER y ASIFT para la detección de puntos y el algoritmo de SIFT para la descripción de dichos puntos característicos. En la Figura 3.1 se observa el resultado del algoritmo SIFT aplicado a una imagen del dataset de ODM.

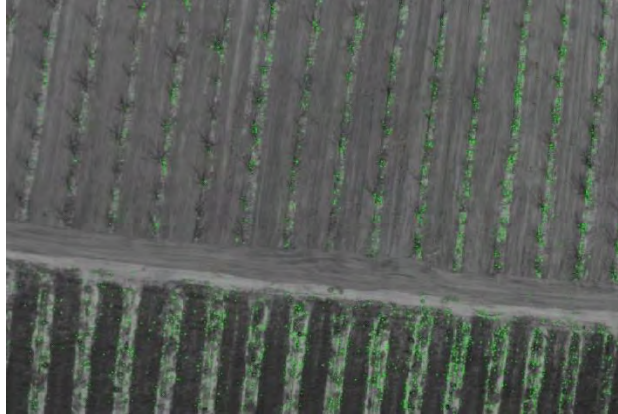


Figura 3.1 Resultado de aplicar SIFT [53] en una imagen de ODM [93].

3.1.2 ORB [52]

Este algoritmo fue propuesto por Rublee en el artículo “ORB: An efficient alternative to SIFT or SURF” [52] en el 2011. ORB es una alternativa a los algoritmos de SIFT y SURF. ORB básicamente es la combinación de los algoritmos de FAST para la detección y BRIEF para la descripción de los puntos. Pero, ORB ofrece mejoras a las limitaciones de sus algoritmos base. Agregando la invarianza a la escala que FAST no tiene y considerando también la invarianza a la orientación que BRIEF no considera.

También es importante resaltar que ORB es que es de licencia libre, mientras que si se desea utilizar SIFT o SURF para usos comerciales se tiene que pagar por una licencia. En la Figura 3.2 se muestra el resultado de la aplicar ORB a una imagen del conjunto ODM. La implementación se realizó utilizando OpenCV y Python.

En el CENIDET se han desarrollado numerosas comparaciones de detectores y descriptores de puntos característicos, en las cuales han seleccionado a ORB, debido a que tiene un buen desempeño y con un menor tiempo de procesamiento [94], [95].

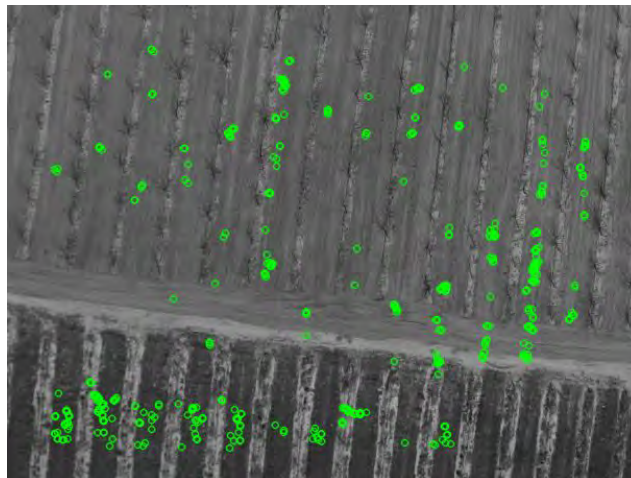


Figura 3.2 Resultado de aplicar ORB [52] en una imagen de ODM [93].

3.1.3 SuperPoint [59]

SuperPoint [59] es un algoritmo que utiliza redes neuronales convolucionales para detectar y describir puntos de interés (a nivel píxel) en una sola pasada.

Los autores crean un gran conjunto de datos de ubicaciones de puntos de interés en las imágenes. Para generar estos puntos primero entrenan una red neuronal convolucional con millones de ejemplos de datos sintéticos, creando Syntetic shapes. El conjunto de datos sintéticos consta de formas geométricas simples.

Al detector resultante le denominan *MagicPoint*, y mencionan que obtiene mejores resultados que los algoritmos clásicos en zonas donde hay muchas esquinas como en los objetos geométricos. Sin embargo, cuando se utiliza en un ambiente real (como en zonas de árboles) el rendimiento baja considerablemente.

Para reducir estas diferencias los autores proponen una técnica de múltiples escalas y transformaciones. A diferencia de los detectores clásicos que solo se basan en encontrar puntos característicos, busca que sean repetibles. Esta Adaptación Homográfica está diseñada para permitir el entrenamiento auto-supervisado del detector de puntos de interés. Para ello deforman la imagen varias veces para tener muchos puntos de vista y escalas para entrenar al modelo. Combinando estas dos técnicas (*magic point* y la adaptación homográfica) se entrena una red convolucional que extrae conjuntamente puntos de interés y descriptores de la imagen, logrando mejorar el rendimiento; al algoritmo final de detección le llaman *SuperPoint*. La Figura 3.3 presenta el resultado de aplicar el algoritmo SuperPoint en una imagen del dataset ODM.

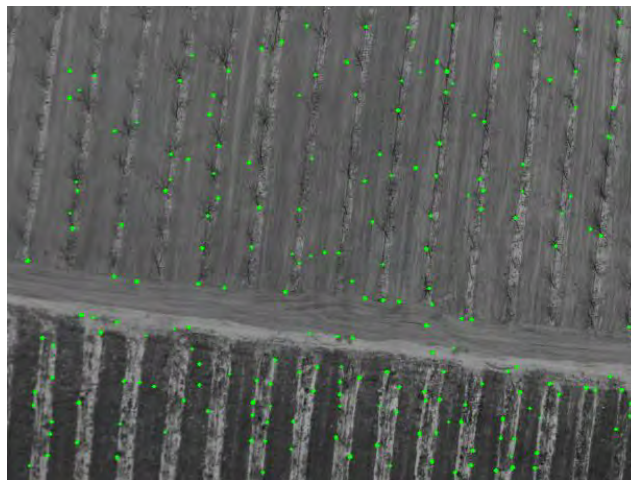


Figura 3.3 Resultado de aplicar SuperPoint [59] en una imagen de ODM [93].

3.2 Algoritmos para correspondencia de puntos característicos

3.2.1 SuperGlue [68]

SuperGlue [68] es un algoritmo para detectar correspondencias entre puntos característicos. Es destacable porque además de buscar los puntos característicos para hacer la asociación, como normalmente lo hacen otros algoritmos clásicos que solamente consideran la posición y la descripción de un punto característico, SuperGlue busca ese

punto característico inicial en el siguiente cuadro y considera a los vecinos del punto. Las asignaciones se estiman resolviendo un problema de transporte óptimo diferenciables y posteriormente calcula los descriptores coincidentes.

SuperGlue logra una mejora significativa con respecto a los enfoques existentes, lo que permite una estimación de pose relativa altamente precisa en pares de imágenes interiores y exteriores de línea de base extremadamente amplia. Además, SuperGlue se ejecuta en tiempo real, a una tasa de 10 FPS. El esquema de la propuesta y sus etapas se puede observar en la Figura 3.4.

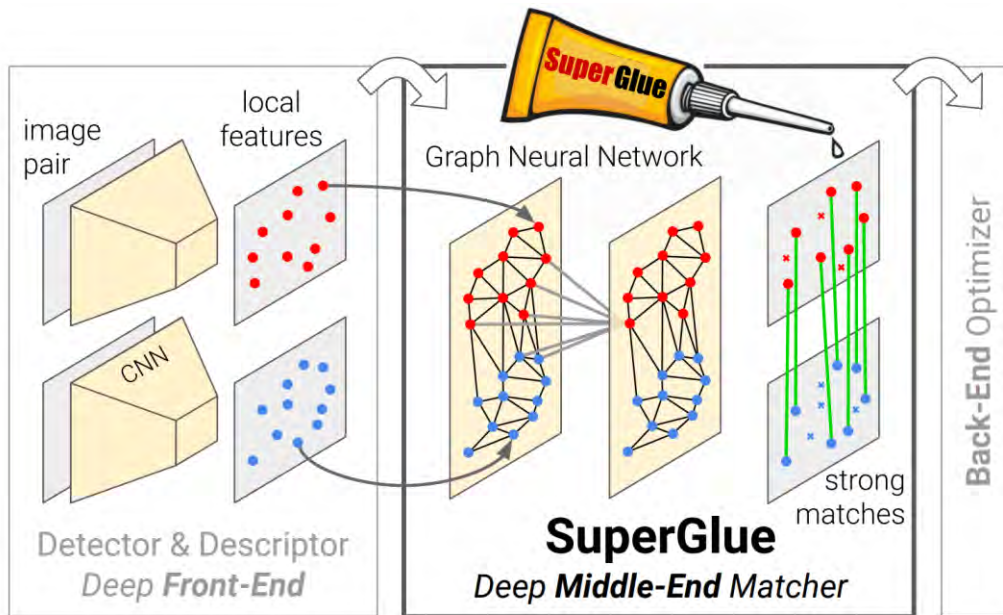


Figura 3.4 Flujo de trabajo de SuperGlue [4].

3.2.2 Brute Force (BF) [65]

Los algoritmos de fuerza bruta son exactamente lo que parecen, métodos sencillos para resolver un problema, que se basan en poder de cómputo y en probar todas las posibilidades, en lugar de técnicas avanzadas para mejorar la eficiencia.

Un ejemplo clásico en ciencias computacionales es el del comerciante, que tiene que visitar 10 ciudades en el menor tiempo posible, ¿Cómo puede calcular el orden en el que tiene que visitar las ciudades?, cuya solución es: primero calcular la distancia entre todos los pares de las ciudades y posteriormente seleccionar el par con la distancia más corta.

En el uso de puntos característicos, compara el descriptor de un punto contra todos los de la otra imagen y selecciona el que menor diferencia presente, lo cual lo hace propenso a falsos positivos, además de lento.

3.2.3 K-Nearest-Neighbours (KNN) [66]

El K-Nearest-Neighbours es un algoritmo de clasificación cuyo mecanismo de trabajo es muy simple: dadas las muestras de prueba, se encuentran las k muestras de entrenamiento más cercanas a ellas en el conjunto de entrenamiento, basándose en una cierta métrica de

distancia. Luego, con base en el número de clases de los K vecinos, se hacen las predicciones. Una opción para determinar la pertenencia es considerar la decisión por votación; es decir, si la mayoría de las K instancias pertenecen a una determinada clase, la instancia de entrada se clasifica en esa misma clase.

Puede entenderse aproximadamente como si una muestra de entrada en el espacio de características fuera más similar entre las K muestras en el entrenamiento (es decir, el vecino más cercano en el espacio de características) perteneciera a una determinada categoría, entonces la muestra también pertenece a esta categoría. El entendimiento es que los puntos K más cercanos a ellos votan para decidir en qué categoría de datos se clasifican.

3.2.4 KD-Tree-based nearest-neighbor (KD-TREE) [67]

Un árbol KD (k-dimensional) es una estructura de datos utilizada principalmente para la búsqueda de puntos en un espacio multidimensional, ampliamente utilizada en el aprendizaje automático [96].

Nodo del árbol: Cada nodo del árbol KD representa un punto en el espacio k-dimensional. El nodo tiene dos hijos: uno representa un punto "a la izquierda" con respecto al plano de corte que divide el espacio en dos subespacios, y el otro representa un punto "a la derecha" con respecto al plano de corte.

Estructura de árbol:

Parte de un conjunto de puntos y elige un punto como nodo raíz del árbol. A continuación, se elige una dimensión (o eje) para dividir el espacio en dos subespacios. La selección puede ser rotacional o considerando algún criterio como la varianza de los puntos en esa dimensión. Divide los puntos en dos grupos, uno para cada subespacio, dependiendo de si sus coordenadas en la dimensión elegida son menores o mayores que las coordenadas del punto de corte. Repite este proceso recursivamente para cada subespacio hasta que solo queden unos pocos puntos en cada subdivisión, o hasta que se alcance algún otro criterio de parada.

Búsqueda en el árbol: Para buscar un punto en el árbol, comienza en la raíz y desciende por el árbol de acuerdo con las divisiones. En cada nodo, compara la coordenada relevante del punto buscado con la coordenada de corte del nodo actual para determinar en qué subárbol debemos buscar.

Repite este proceso hasta llegar a un nodo hoja, que contiene el punto buscado o está lo suficientemente cerca según algún criterio de proximidad.

Aplicaciones: Los árboles KD son útiles en aplicaciones que implican la búsqueda de puntos en espacios multidimensionales, como la búsqueda de vecinos más cercanos, búsqueda de rango, *clustering*, entre otros. Son ampliamente utilizados en campos como la computación gráfica, minería de datos, reconocimiento de patrones y aprendizaje automático.

Un ejemplo se puede observar en la Figura 3.5.

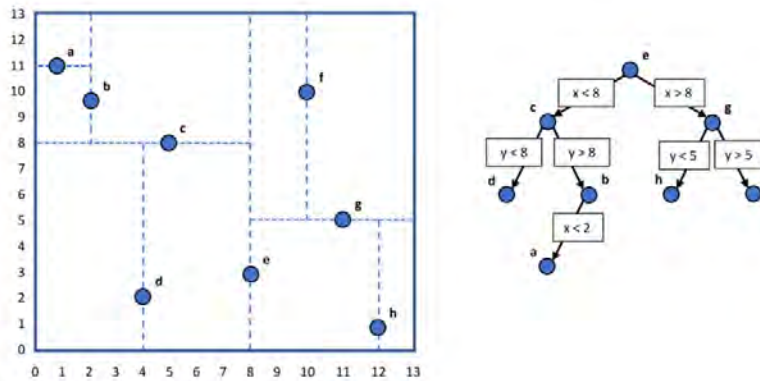


Figura 3.5 Ejemplo KD – TREE [97].

3.3 Conceptos de fotogrametría

Modelo digital de elevación (DEM por sus siglas en inglés):

Como su nombre lo indica, un DEM es un modelo digital de elevación de un terreno, un DEM se puede obtener de diferentes formas, por ejemplo, fotogrametría satelital [98], fotogrametría con drones [3], el uso de sensores laser terrestres [99], [100], etc.

Punto de control terrestre (GCP)

Son puntos del mapa de los cuales se conoce, con precisión, su ubicación global. Para generar un GCP, típicamente se utilizan GPS de doble banda. Utilizando algoritmos como OPUS [101] se pueden generar GCP con una precisión de 1 cm (Solamente en donde está disponible). Algunos GCP apoyan en el proceso de generar DEM ya que proporcionan información sobre la verdadera posición de un punto y con esa información, corregir los puntos aledaños y, por otro lado, el resto de los GCP son utilizados en el proceso de verificación de la precisión final del DEM [75].

Existen 2 tipos de GCPs, los estandarizados y los no estandarizados. Los primeros utilizan colores contrastantes contra el suelo, como por ejemplo una cuadrícula con dos cuadros negros y dos blancos intercalados [102], o en algunos otros casos patrones tipo QR diferentes para cada GCP [103] los cuales pueden ser detectados mediante software especializado. Los no estandarizados se pueden generar utilizando otro tipo de marcas, por ejemplo, una barda, un señalamiento de tránsito, etc.

El enfoque tradicional para detectar los puntos es manual, en donde un usuario busca el punto central del GCP en la fotografía y lo señala, para posteriormente buscar ese mismo punto en todas las fotografías en las que aparece; el proceso se repite para todos los GCPs. En reconstrucciones pequeñas el proceso tiene tiempos aceptables; sin embargo, en proyectos de mayor tamaño, los cuales tienen muchos GCPs y miles de imágenes, el proceso se vuelve tedioso y propenso a errores humanos por fatiga.

Existen enfoques automáticos para los GCPs estandarizados ya que tienen menor complejidad al momento de buscar patrones [102-104], debido a que ya existe algo predeterminado a buscar. Sin embargo, también existen enfoques automáticos para GCPs

no estandarizados, los cuales tienen la ventaja de necesitar una menor logística ya que objetos estáticos dentro de la misma zona pueden servir como GCPs. Por ejemplo [105] utilizó un enfoque de detección de bordes para extraer la línea costera e identificó GCP mediante coincidencia de línea. En [106], [107] utilizaron la coincidencia de plantillas para lograr la búsqueda automatizada de postes de luz y utilizarlos como GCP. En [108] utilizan 4 pasos que permiten recuperar la posición de un punto de interés en todas las imágenes en las que aparece y la aplica a señales de tránsito tales como pasos peatonales. En [109] se utilizó el algoritmo de Harris (el cual es un detector de esquinas de uso general) para llevar a cabo una detección automatizada de modo que objetos aislados brillantes arbitrarios puedan servir como GCP.

En la Figura 3.6 se muestra un terreno con puntos rojos y azules, los azules son puntos de control terrestre utilizados para mejorar la precisión del DEM y los rojos son utilizados para evaluar la precisión.



Figura 3.6 GCP [75].

Error de re-proyección

El error de re-proyección es una estimación de cuán exactos son los parámetros encontrados en la calibración de una cámara, cuanto más cerca esté del cero, es mayor su precisión. Es la distancia entre un punto característico detectado dentro del patrón de calibración y el mismo punto del mundo proyectado en la imagen [110]. En el caso de SIFT se calcula utilizando la *norma L2* para cada punto 2D -3D y se deduce el promedio de todos los errores.

3.4 Structure From Motion (SFM)

Estructura desde el movimiento (SFM por sus siglas en inglés) es una técnica que permite recuperar, tanto la pose de la cámara como el movimiento de esta [111]. Es ampliamente utilizada para reconstrucción 3D ya que se puede usar en bancos de imágenes muy grandes y no necesariamente ordenadas o capturadas con la misma cámara [112-114] y entrega una nube de puntos dispersa [114]. De manera general, los pasos para realizar SFM son los siguientes:

- Preprocesamiento de datos: Se calcula la covarianza de un par de imágenes superpuestas, calcula un pequeño esqueleto de imágenes y agrega el resto de las imágenes que no tienen mucha redundancia (Una distancia mínima).
- Extracción de características: Se detectan y describen los puntos característicos en las imágenes, utilizando por ejemplo ORB, SIFT, SURF.
- Correspondencia entre imágenes: Crea un árbol de vocabulario, convierte el vector de características en una palabra visual y con KNN hace esta correspondencia.
- Estimación de la matriz fundamental.
- Calibración de la cámara.
- Realizar ajuste de paquete para reducir el error de re-proyección.

En la Figura 3.7 se puede observar el esquema que representa el proceso del algoritmo de SFM.

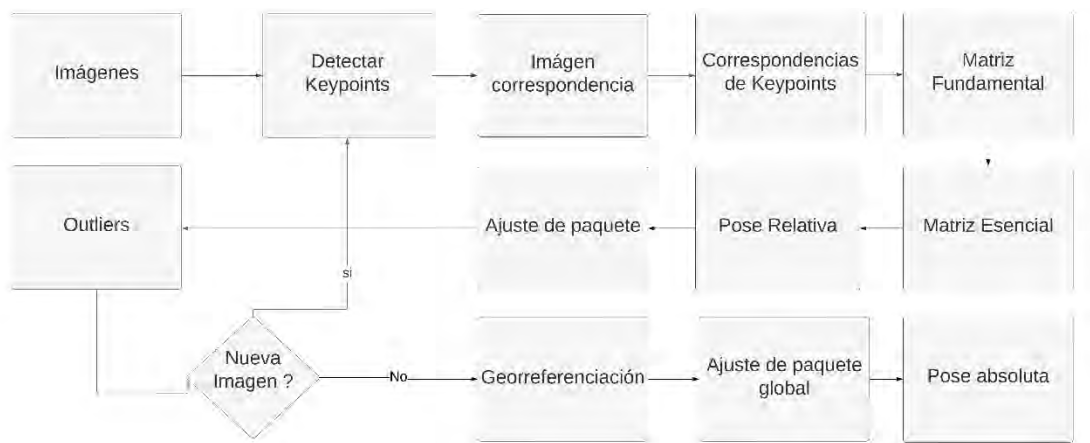


Figura 3.7 Esquema que representa el proceso de SFM.

A continuación, se detallan algunos conceptos claves para SFM:

3.4.1 Modelo proyectivo de la cámara

Dado un mundo de referencia 3D $W = O_W + X_W + Y_W + Z_W$ y un sistema de referencia de imagen 2D $D = o_d + u_d + v_d$, el cual tiene su origen o_d en la parte superior izquierda de la imagen, la matriz de proyección perspectiva M que tiene dimensiones de 3×4 , es la que relaciona las coordenadas de los puntos del mundo 3D ${}^wq = [X, Y, Z, 1]^T$ los cuales están relacionados al sistema W , con las coordenadas 2D ${}^d p = [u, v, 1]^T$ los cuales están asociados al sistema de referencia D , estos son los píxeles de la imagen, ver Ecuación 3.1.

$${}^d q = {}^d M_w * {}^w p \quad \text{Ecuación 3.1}$$

Como se puede observar en la Ecuación 3.2, la matriz de proyección perspectiva M se puede descomponer en el producto de 3 matrices, las cuales son: la matriz de parámetros intrínsecos de la cámara ${}^d K_p$, la matriz perspectiva ${}^p P_c$ y la matriz de valores extrínsecos de la cámara ${}^c T_w$.

$${}^d M_w = {}^d K_p * {}^p P_c * {}^c T_w \quad \text{Ecuación 3.2}$$

En esta ecuación aparecen 4 sistemas de referencia W, C, P y D :

El sistema

- $C = o + X_c + Y_c + Z_c$ es el sistema de referencia de la cámara, tiene su origen en el centro óptico O y su eje Z_c coincide con el eje óptico de la cámara.
- $P = O + u_p + v_p$ es el sistema de referencia asociado al plano de imagen, cuyo origen está situado en el punto principal O de la imagen.
- $D = o_d + u_d + v_d$ sistema de referencia cuya orientación de sus ejes coinciden con la orientación de los ejes de P y tiene su origen en la parte superior izquierda de la imagen.
- El sistema $W = O_W + X_W + Y_W + Z_W$ es del mundo 3D.

En la Figura 3.8 se ejemplifica en qué parte se encuentra cada sistema.

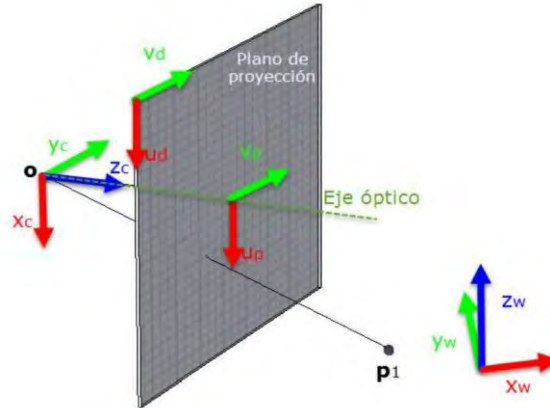


Figura 3.8 Sistemas de referencia [116].

Matriz T

La matriz de valores extrínsecos de la cámara cT_w que tiene una dimensión de 4×4 , es la que relaciona las coordenadas 3D del mundo con las coordenadas 3D de la cámara, como la Ecuación 3.3. En otras palabras, obtiene las coordenadas del punto ${}^c q$ expresado con respecto a C a partir del mismo punto ${}^w p$, expresado con respecto a W , aplicando la matriz de transformación cT_w .

$${}^c q = {}^cT_w * {}^w p \quad \text{Ecuación 3.3}$$

La Ecuación 3.4 muestra la forma de la matriz T

$${}^cT_w = \begin{pmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Ecuación 3.4}$$

Donde R , es la matriz de rotación 3×3 , que define la orientación de W con respecto a C ; t es el vector columna de traslación 3×1 que define dónde está el origen de W con respecto a C y 0 es el vector fila 1×3 nulo, el cual sirve para poder colocar el valor de 1 y volver la matriz homogénea.

Matriz P

La matriz de proyección perspectiva pP_c , es una matriz de 3×4 que transforma coordenadas del punto 3D ${}^c p = [X, Y, Z, w]^T$ del sistema de referencia de la cámara C , en

coordenadas de este punto 2D ${}^p q = [u', v', z]^T$ o ${}^p q = [u, v, 1]^T$ proyectado en el sistema de referencia del plano de la imagen, el cual tiene su origen en el centro de la imagen, la matriz P es una transformación afín de escalamiento por el factor f .

$${}^p q = {}^p P_c * {}^c p \tag{Ecuación 3.5}$$

La Ecuación 3.6 muestra la forma de la matriz ${}^p P_c$

$$\begin{bmatrix} u' \\ v' \\ z \end{bmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{Ecuación 3.6}$$

$$u = \frac{u'}{z}; v = \frac{v'}{z} \tag{Ecuación 3.7}$$

Donde f es la distancia focal de la cámara.

Matriz K

La matriz de parámetros intrínsecos de la cámara ${}^d K_p$ relaciona las coordenadas del punto 2D ${}^p q = [u, v, 1]^T$ del sistema de referencia del plano de la imagen P con las coordenadas 2D del punto ${}^d q = [x, y, 1]^T$ con respecto al sistema de coordenadas del sensor D . La Figura 3.9 ejemplifica esta relación.

$${}^d q = {}^d K_p * {}^p p \tag{Ecuación 3.8}$$

La Ecuación 3.9 muestra la forma de la matriz ${}^d K_p$. Algunos autores en lugar de utilizar dx , colocan k_x haciendo referencia a la cantidad de píxeles por unidad de medida, que es lo mismo.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{pmatrix} 1/dx & s & x_0 \\ 0 & 1/dy & y_0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{Ecuación 3.9}$$

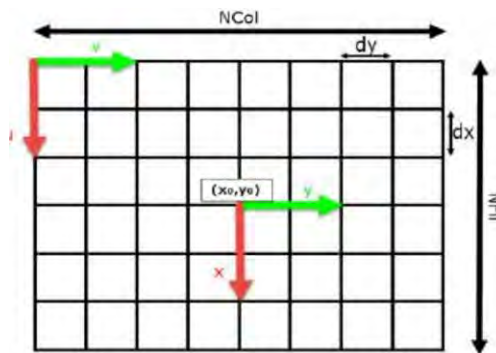


Figura 3.9 P y sistema D [18].

Donde $NCol$ es el número de columnas del sensor de la cámara, $NFil$ es el número de filas del sensor de la cámara, dx y dy son las dimensiones de un píxel en milímetros y $[x_0, y_0]^T$ son las coordenadas del punto principal en píxeles y S es el factor de "skew".

Este valor está relacionado con el ángulo que toma el eje Y del sistema de coordenadas del plano de la imagen cuando los ejes no son perpendiculares. En los modelos estándar los ejes siempre son perpendiculares por lo que $S=0$.

Finalmente, la matriz M queda con la forma de la Ecuación 3.10. En la Figura 3.9 se puede observar un ejemplo visual del sistema P y el sistema D .

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{pmatrix} 1/dx & s & x_0 \\ 0 & 1/dy & y_0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} * \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{Ecuación 3.10}$$

3.4.2 Geometría de los sistemas estéreo

Se define visión estéreo o estereoscópica como aquella en la que se emplea más de una imagen para obtener una idea de tridimensionalidad [18]. Cuando se captura una imagen utilizando una cámara convencional se transforma una escena 3D a 2D, perdiendo en este proceso la profundidad. Existen varias formas de poder recuperar esa profundidad, como lo es un LIDAR, una sola cámara utilizando SFM o el uso de una cámara binocular [117]. En el mercado venden varias cámaras binoculares como la ZED 2 que son de bajo costo, la cual consta de dos cámaras separadas entre sí por una distancia aproximada de 155 mm. Es esta distancia la que permite capturar dos imágenes del mismo objetivo, pero con un desplazamiento, así como lo hace la visión humana. En la Figura 3.10 se puede observar la geometría de una cámara estéreo convergente y algunos de sus componentes, como lo son sus centros ópticos de cada cámara.

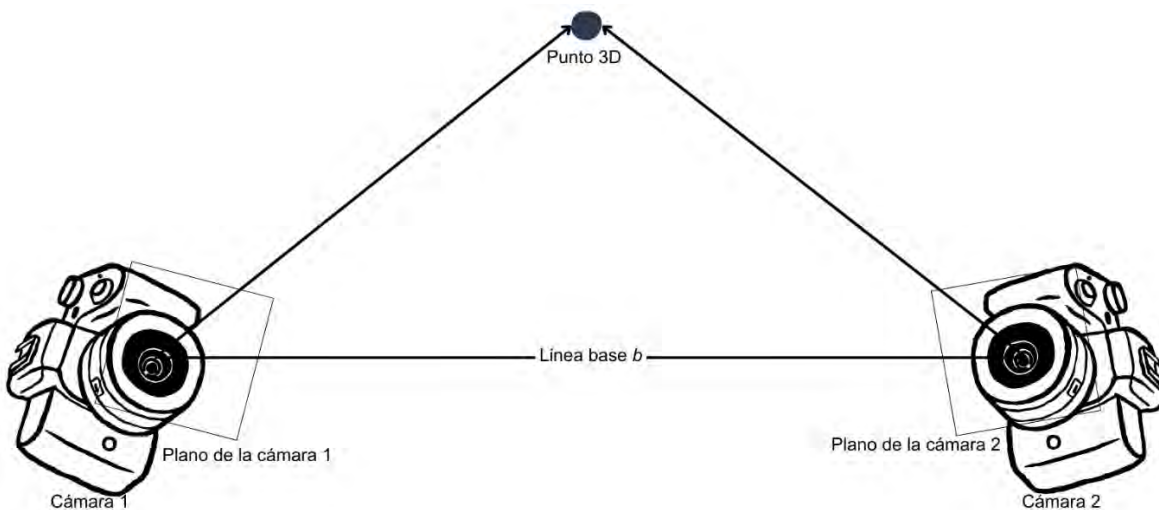


Figura 3.10 Visión estéreo convergente.

Dos imágenes de una misma escena, pero una de ellas con ligero movimiento, por ejemplo, con una rotación x , están relacionadas porque son proyecciones de una misma información 3D. La geometría que relaciona ambas vistas se conoce como geometría epipolar y las relaciones de dependencia que se establecen entre las proyecciones de un punto de la escena, en ambas imágenes, se denominan restricciones epipolares [118].

Las entidades geométricas que intervienen están representadas en la Figura 3.11 y son:

- *Línea base*: es el segmento 3D que une los centros de proyección de ambas cámaras.
- *Plano epipolar*: es el plano que queda definido por un punto de la escena y los centros de proyección de ambas cámaras.
- *Epipolo*: es el punto en el que la línea base corta al plano imagen asociado a dicha vista. Es la proyección en una cámara, del centro de proyección de la otra cámara que forma el par estéreo. El epipolo puede caer fuera del trozo del plano imagen que devuelve la cámara.
- *Línea epipolar*: es la línea de intersección entre el plano de la imagen y el plano epipolar.

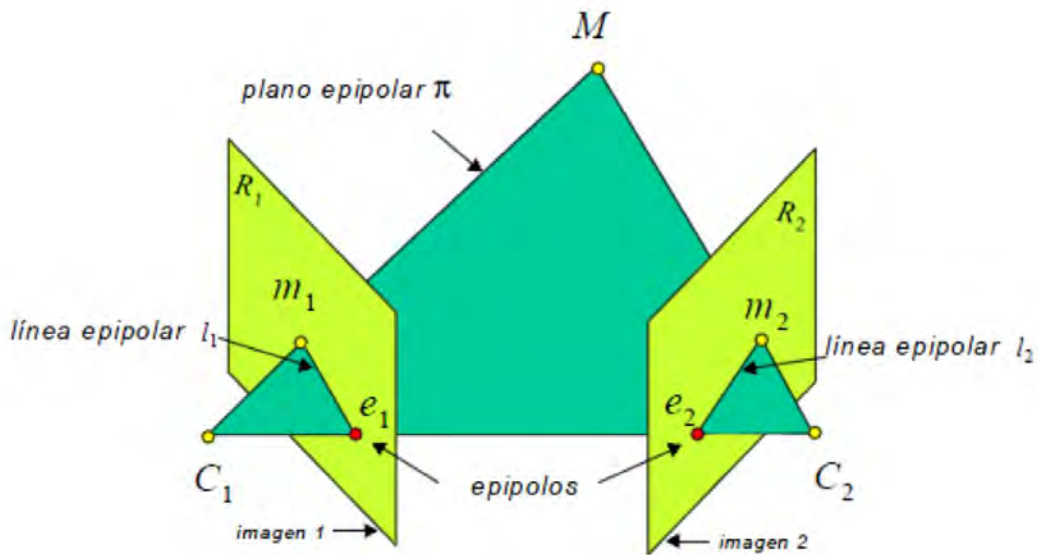


Figura 3.11 Geometría de cámara estéreo paralela o canónica [117].

3.4.3 Matriz Fundamental

La matriz fundamental mapea el plano proyectivo de la imagen 1 (de la Figura 3.11) a la recta epipolar de la imagen 2 (de la misma Figura 3.11) considerando que las matrices de valores intrínsecos dK_p no están calibradas. En otras palabras, una matriz fundamental es la representación algebraica de la geometría epipolar. Solamente depende de los parámetros de la cámara [18].

Sea X un punto 3D del mundo cuyas proyecciones son x_1 y x_2 , la restricción epipolar determina que el punto x_1 define a la recta l_2 en la segunda imagen en la cual se encuentra el punto x_2 . En el plano proyectivo, la pertenencia de un punto a una recta se expresa como:

$$x_2^T l_2 = 0 \tag{Ecuación 3.11}$$

$$X(\lambda) = \begin{bmatrix} K_1^{-1} x_1 \\ 0 \end{bmatrix} + \lambda \begin{bmatrix} 0_{3 \times 1} \\ 1 \end{bmatrix} = D + \lambda O_1 \tag{Ecuación 3.12}$$

Donde D es el punto infinito de la recta y O_1 es el centro de proyección de la cámara 1. La proyección de dicha recta en la imagen 2 es la recta definida por los puntos resultantes de proyectar D y O_1 sobre la imagen 2, con lo que resulta:

$$e_2 = P_2 O_1 = K_2 [R/t] \begin{bmatrix} O_{3X1} \\ 1 \end{bmatrix} K_2^{-1} \quad \text{Ecuación 3.13}$$

$$d_2 = P_2 D = K_2 [R/t] = K_2 R K^{-1} X_1 \quad \text{Ecuación 3.14}$$

$$l_2 = e_2 x d_2 = (e_2)_x d_2 = (k_2^{-t}(t)_x R K^{-1}) X_1 = F_{X1} \quad \text{Ecuación 3.15}$$

Sustituyendo en la ecuación $x_2^T l_2 = 0$, resulta que la restricción epipolar puede expresarse algebraicamente como:

$$x_2^T F x_1 = 0 \quad \text{Ecuación 3.16}$$

La matriz F se conoce como matriz fundamental y recoge todas las restricciones asociadas a la geometría epipolar.

La Figura 3.12 muestra dos imágenes capturadas dentro del motor gráfico *UnrealEngine* [119] utilizando el simulador de drones *AirSim* [120].

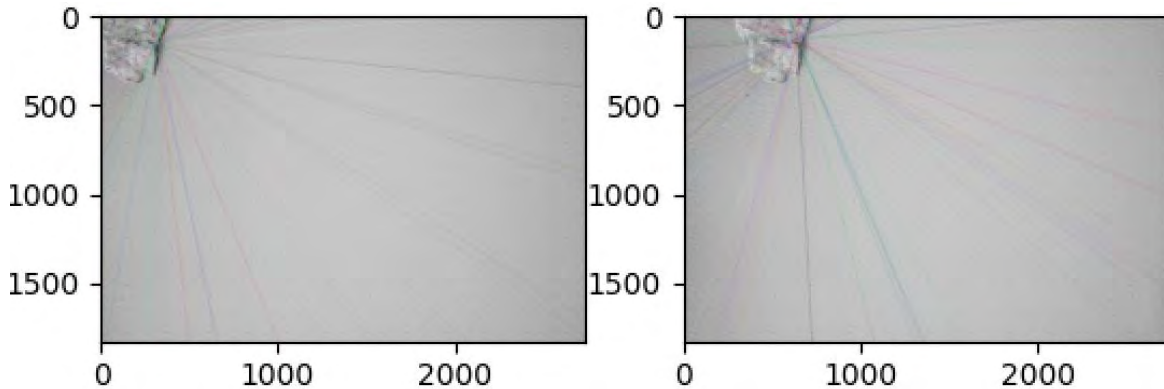


Figura 3.12 Imágenes creadas en *UnrealEngine* [119] con *AirSim* [120].

Un ejemplo del cálculo de la matriz F se muestra en la Figura 3.13, utilizando las dos imágenes de la Figura 3.12 y el algoritmo de 8 puntos.

Matriz fundamental F :

$$\begin{bmatrix} 9.93075023e-06 & -6.32510904e-06 & -2.18615660e-03 \\ -3.85977488e-06 & -1.21801220e-05 & 3.04666173e-03 \\ -5.81535596e-03 & 5.57086335e-03 & 1.00000000e+00 \end{bmatrix}$$

Figura 3.13 Ejemplo matriz F .

Matriz Esencial

La matriz fundamental establece una restricción entre dos imágenes sin calibrar debido a que relaciona puntos expresados en píxeles [18]. Por lo tanto, incorpora la información tanto de la posición relativa de cada cámara como de los parámetros intrínsecos de cada una de ellas. Si las cámaras están calibradas se puede trabajar con coordenadas normalizadas:

$$x = K^{-1}x \quad \text{Ecuación 3.17}$$

Sustituyendo en la ecuación 3.15 queda expresada como

$$\hat{x}_2^T ([t]_x R) \hat{x}_1 = \hat{x}_2^T E \hat{x}_1, E = K_2^T F K_1 = [t]_x R \quad \text{Ecuación 3.18}$$

La matriz E se denomina matriz esencial y también define la restricción epipolar, en unas coordenadas de imagen diferentes [18]. Al haber eliminado el efecto de los parámetros intrínsecos de las cámaras, la matriz esencial solo depende de la posición relativa de una cámara respecto a la otra. Un ejemplo del cálculo de la matriz esencial se muestra en la Figura 3.14, utilizando la información de las dos imágenes de la Figura 3.12.

E :

$$\begin{bmatrix} 85.66035859 & -54.40762112 & 8.93737309 \\ -33.20119354 & -104.48098321 & -63.35121459 \\ 9.47913328 & -66.87449558 & -28.03378968 \end{bmatrix}$$

Figura 3.14 Ejemplo matriz E

Es posible obtener la posición relativa de una cámara respecto de la otra a partir de la factorización de la matriz esencial. Se divide en 4 pasos [18]:

- 1 Calcular la descomposición en valores singulares de E : $E = U \text{diag}(\sigma, \sigma, 0) V^T$
- 2 La matriz de proyección de la primera cámara es: $P_1 = [I_{3 \times 3} | 0_{3 \times 3}]$
- 3 Entonces las cuatro posibles descomposiciones en R y t son:
 - $P_2 = [UWV^T | +u_3]$
 - $P_2 = [UWV^T | -u_3]$
 - $P_2 = [UW^T V^T | +u_3]$
 - $P_2 = [UW^T V^T | -u_3]$
- Con $W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

4 De las cuatro posibles soluciones se debe seleccionar aquella que sitúe los puntos reconstruidos delante de ambas cámaras. En ausencia de información adicional, la escala de la traslación no puede ser determinada. Con la solución propuesta, la traslación queda normalizada a 1, por lo que todas las medidas posteriores quedarían escaladas por la distancia base real [18].

En la Figura 3.15 se pueden observar las 4 posibles soluciones del cálculo de la posición de las dos imágenes de la Figura 3.12, para este caso en específico la solución es la descomposición $P_2 = [UWV^T | +u_3]$, debido a que la mayoría de los valores son positivos.

Cámaras estimadas:

P = UWV^T:

```
[[ 1.67026298e+03  2.59382069e+03  1.18055939e+03  5.52008420e+02]
 [ 2.92633058e+03 -1.33931684e+03  7.83426212e+02  1.10839823e+02]
 [ 8.03474337e-01  2.99618292e-01 -5.14449092e-01  8.69085403e-01]]
```

P = -UWV^T:

```
[[ 1.67026298e+03  2.59382069e+03  1.18055939e+03 -5.52008420e+02]
 [ 2.92633058e+03 -1.33931684e+03  7.83426212e+02 -1.10839823e+02]
 [ 8.03474337e-01  2.99618292e-01 -5.14449092e-01 -8.69085403e-01]]
```

P = UW^TV^T:

```
[[ -1.21103094e+03 -2.22889005e+03 -2.11585867e+03  5.52008420e+02]
 [ -2.83411968e+03  1.41259262e+03 -9.71228431e+02  1.10839823e+02]
 [ -8.04566310e-02  2.74930730e-01 -9.58091762e-01  8.69085403e-01]]
```

P = -UW^TV^T:

```
[[ -1.21103094e+03 -2.22889005e+03 -2.11585867e+03 -5.52008420e+02]
 [ -2.83411968e+03  1.41259262e+03 -9.71228431e+02 -1.10839823e+02]
 [ -8.04566310e-02  2.74930730e-01 -9.58091762e-01 -8.69085403e-01]]
```

Figura 3.15 Ejemplo del cálculo de la posición.

3.4.4 Cálculo GSD ZED 2

El Ground “Sampling Distance” (GSD) es la distancia entre dos puntos centrales de muestras, en el caso de la fotogrametría, como se utilizan cámaras, las muestras son los píxeles; en otras palabras, el GSD es la distancia medida en tierra de dos centros de píxeles, este valor varía de cámara en cámara, así como de la altura de vuelo.

Es importante conocer este valor debido a que es la precisión máxima que puede alcanzar la reconstrucción. El GSD es el valor máximo calculado con las Ecuaciones 3.19 y 3.20

$$GSD_A = \frac{\text{Altura de vuelo} * \text{Ancho del sensor}}{\text{Longitud focal} * \text{Ancho de la imagen}} \quad \text{Ecuación 3.19}$$

$$GSD_B = \frac{\text{Altura de vuelo} * \text{Largo del sensor}}{\text{Longitud focal} * \text{Largo de la imagen}} \quad \text{Ecuación 3.20}$$

La cámara ZED 2 tiene los siguientes valores: Largo y ancho del sensor = 5.98 mm, altura de vuelo = 10 m, ancho de la imagen = 2688 px, largo de la imagen = 1520 px, la ecuación 2 da como resultado GSD= 1.05 cm y la ecuación 3 GSD= 1.86 cm, por lo tanto, en el caso de la ZED 2 el GSD mayor esperado es de 1.86 cm.

3.5 Sistemas complementarios

COLMAP [121], [122]

COLMAP [121], [122] es un sistema para la reconstrucción 3D, capaz de realizar desde una nube de puntos escasa hasta la texturización con técnicas como POISSON [123]. Fue

diseñado por Johannes L. Schönberger en el 2016 y es uno de los más citados en el estado del arte [124-127]. Además de su capacidad para generar modelos 3D a partir de nubes de puntos dispersas, COLMAP ofrece una serie de funciones y técnicas avanzadas que lo hacen destacar en el campo de la reconstrucción 3D:

Reconstrucción incremental: COLMAP permite la reconstrucción incremental, lo que significa que puede agregar nuevas imágenes y ampliar los modelos 3D existentes sin tener que reconstruir desde cero cada vez.

Fusionar reconstrucciones parciales: el sistema es capaz de fusionar múltiples reconstrucciones parciales en un único modelo coherente. Esto resulta útil cuando se trabaja con conjuntos de datos grandes o fragmentados.

Densificación de nubes de puntos: COLMAP puede densificar nubes de puntos, es decir, agregar puntos adicionales entre nubes de puntos existentes, para aumentar la resolución del modelo 3D y mejorar su precisión.

Texturización de Modelos 3D: Además de la reconstrucción geométrica, COLMAP ofrece herramientas para aplicar texturas a los modelos 3D, lo que los hace visualmente más realistas y detallados.

Compatibilidad y Flexibilidad: COLMAP es compatible con una amplia gama de formatos de imagen y modelos 3D, lo que facilita su integración en diferentes flujos de trabajo y aplicaciones. Además, su código abierto permite a los usuarios personalizar y extender sus funcionalidades según sea necesario.

En conjunto, estas características hacen de COLMAP una herramienta versátil y potente para la reconstrucción 3D, adecuada para una variedad de aplicaciones que van desde la investigación académica hasta el desarrollo comercial en campos como la arqueología, la ingeniería, la cinematografía y la realidad virtual.

PyCOLMAP [20]

PyCOLMAP es la versión compilada de COLMAP para Python, la única desventaja que posee es su incapacidad de realizar tanto una nube de puntos densa como tampoco la texturización, pero debido a que está compilado para funcionar con Python, es capaz de utilizar SuperPoint y SuperGlue ya que ambos están diseñados para dicho lenguaje.

UnrealEngine

UnrealEngine es uno de los motores gráficos más reconocidos actualmente creado por la compañía de Epic Games y que en la actualidad cuenta con 5 versiones. Es utilizado en su mayoría para la creación de videojuegos, pero no es exclusivo de este mundo, también es utilizado para generar animaciones, arquitectura, ingeniería, simulaciones de robots, estudio de sonido, entre otras muchas funciones [128].

AirSim [120]

Es un simulador para drones, autos y más, fue diseñado para funcionar en UnrealEngine, pero actualmente también puede funcionar en Unity. Es de código abierto y es capaz de simular algunos de los controladores de vuelo más famosos tal como PX4, ArduPilot, etc., para una simulación más realista.

AirSim tiene 3 modos de simulación, “Car”, “Multitrotor” y “Computer Vision Mode”, los utilizados en este reporte son “Multitrotor” y “Computer Vision Mode”, debido a que “Car” es utilizado para vehículos de cuatro ruedas.

- AirSim en modo Multitrotor

Permite la simulación de un dron, el cual consta de 5 cámaras (3 inferiores y dos laterales), las cuales son completamente modificables, desde su posición, orientación, distancia entre ellas, distancia focal, etc. Además, cuenta con los siguientes sensores, una IMU, un barómetro, un GPS, un magnetómetro, un LIDAR y una cámara infrarroja. El dron simulado es afectado por la gravedad, por los valores del controlador de vuelo el viento, etc. La Figura 3.16 muestra un ejemplo de la simulación de Multitrotor.



Figura 3.16 Ejemplo de un escenario AirSim [120] en modo Multitrotor.

- AirSim en modo Computer Vision Mode

El modo de *Computer Vision Mode*, cuenta solamente con las cámaras y deshabilita todos los demás sistemas, pero tiene la ventaja de no ser afectado por la gravedad t por lo tanto tiene una gran repetibilidad. La Figura 3.17 muestra un ejemplo de la simulación en el modo de *Computer Vision Mode*.

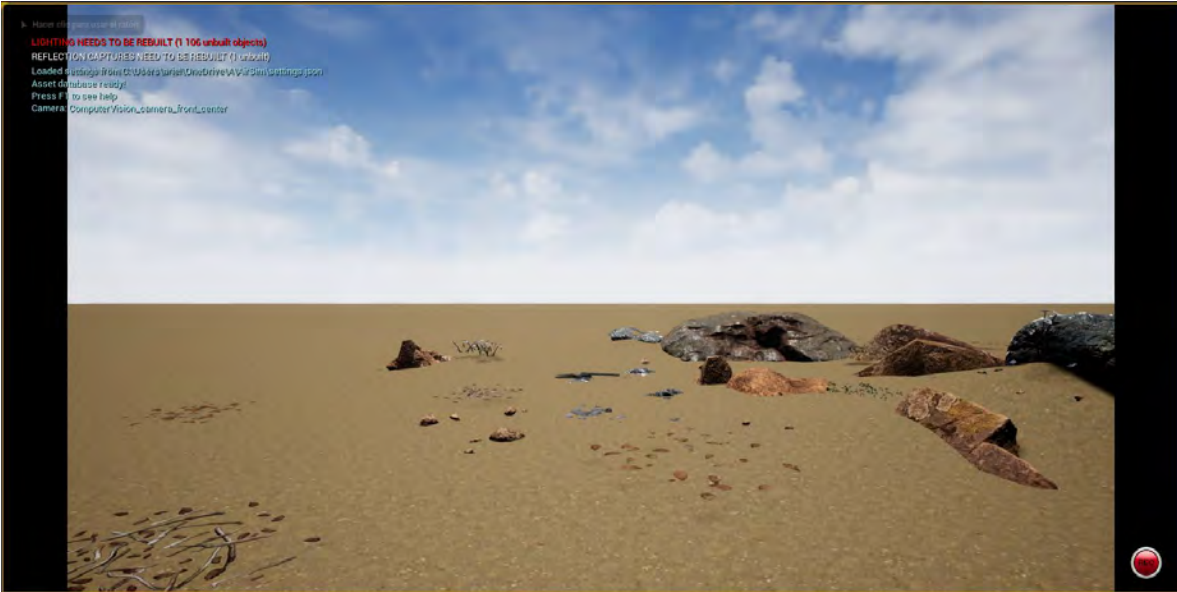


Figura 3.17 Ejemplo de un escenario en AirSim [120] en modo Computer Vision.

3.6 Calibración de una cámara

La calibración de una cámara es un proceso fundamental, en el cual se determinan sus valores intrínsecos de la misma mediante escenarios conocidos llamados patrones de calibración y algoritmos especializados de visión por computador [129].

Cualquier objeto debidamente caracterizado podría ser utilizado como un objeto de calibración, sin embargo, la opción más práctica es el de un patrón regular, como un tablero de ajedrez. En la Figura 3.18 se puede observar un patrón de calibración de tipo tablero de ajedrez.

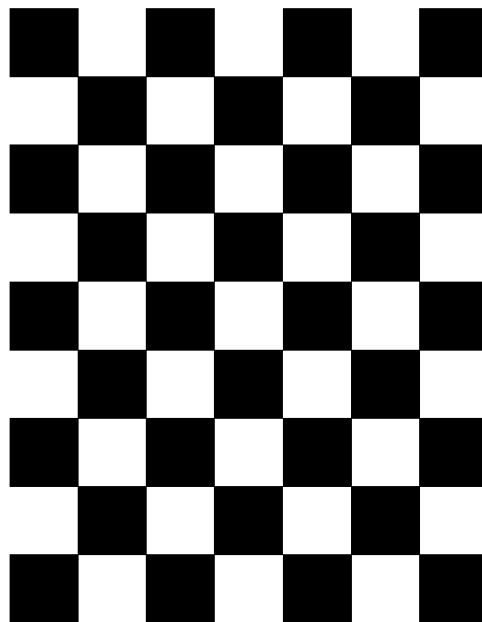


Figura 3.18 Tablero de ajedrez [130].

El procedimiento típico consiste en tener un objeto fijo ya sea la cámara o el tablero y el otro realizando movimientos tanto de rotación como traslación, obteniendo fotografías del patrón de calibración desde diferentes ángulos, realizando movimientos suaves con el objeto móvil, a una distancia entre los 45 cm y 1.5 m.

3.7 Métricas

Un sistema de fotogrametría se enfrenta al problema de que, su reconstrucción debe tener precisión tanto relativa como absoluta. Es decir, la re-proyección de sus puntos debe tener consistencia entre sus puntos locales como coherencia geográfica. Por ejemplo, en la Figura 3.19 se ilustra una casa con una buena precisión relativa, porque las dimensiones de una casa pueden medirse con exactitud en un sistema de coordenadas arbitrario. Sin embargo, la precisión absoluta es escasa o desconocida, porque la casa no está representada en el sistema de coordenadas geográficas de referencia correcto.

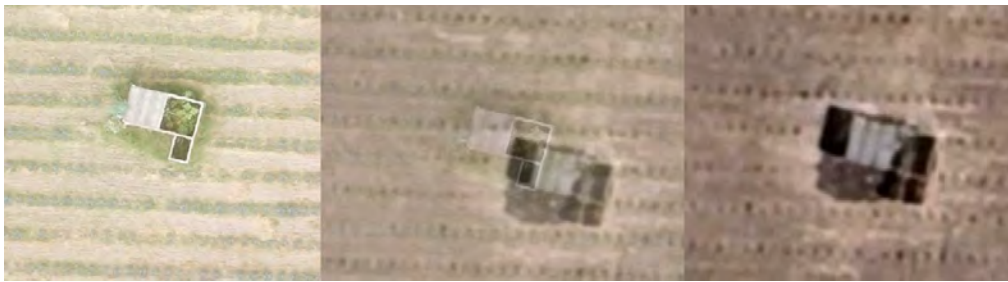


Figura 3.19 Mala precisión absoluta pero buena relativa.

3.7.1 Raíz del error cuadrático medio (RMSE)

RMSE (*Root Mean Square Error* o Raíz del Error Cuadrático Medio) realiza una diferencia entre los valores estimados y los valores reales, estas diferencias se elevan al cuadrado y se calcula el promedio de todas ellas. Como su nombre lo dice, a este promedio se le debe calcular su raíz cuadrada [131-133] RMSE mide la magnitud de error. En fotogrametría, los GCP proporcionan la posición mundial con unos pocos centímetros de error. Algunos se utilizan para mejorar la precisión del DEM, mientras que otros sirven para medir el error. Estos últimos se calculan mediante SFM y se realiza una comparación entre el valor medido del GCP sobre el terreno y el valor calculado para cada eje, utilizando la Ecuación 3.21.

$$RMSE_x = \sqrt{\frac{\sum_{i=1}^n (X_{O_i} - X_{GNSS_i})^2}{n}} \quad \text{Ecuación 3.21}$$

Donde:

- n es el número de GCP evaluados.
- X_{O_i} es la coordenada del eje X, para el i -ésimo GCP dentro del DEM.
- X_{GNSS_i} es la coordenada del eje X para el i -ésimo valor del GCP medido por el GNSS

3.7.2 CE90 y LE90

Otras métricas utilizadas son el CE90 y el LE90. El CE90 es el error circular en el percentil 90. Esto significa que un mínimo del 90 por ciento de los puntos medidos tiene un error horizontal inferior al valor CE90 establecido. LE90 es el error lineal del percentil 90, lo que significa que un mínimo del 90 por ciento, de los errores verticales se encuentran dentro del valor LE90 establecido. En otras palabras, CE90 y LE90 es lo mismo, pero el primero es para el eje X y el segundo es para el eje Y [131] (Ecuación 3.22).

$$CE90 = \frac{90 * N}{100}$$

Ecuación 3.22

- Donde: N = Cantidad de datos.

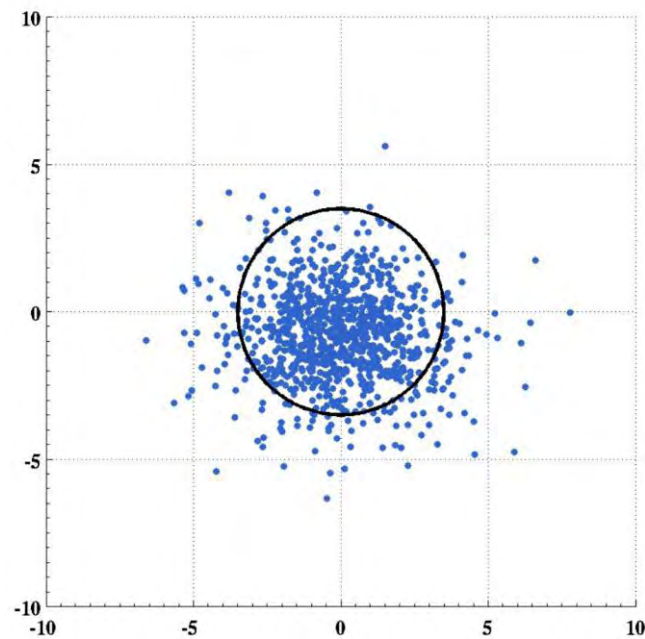


Figura 3.20 Ejemplo de CE90 [131].

Capítulo 4: Metodología de solución

En el presente capítulo se presenta la metodología empleada para el mejoramiento de la precisión en reconstrucciones fotogramétricas. Se divide en las siguientes etapas de desarrollo:

- Fotogrametría.
- Generación de puntos de control terrestre virtuales con cartulinas.
- Preprocesamiento con COLMAP.
- Generación de puntos de control terrestre virtuales con la cámara ZED 2.

4.1 Fotogrametría

En la Figura 4.1 se muestra la metodología empleada para fotogrametría en el presente trabajo, la cual se integra de las etapas típicas de la fotogrametría. Las figuras con borde en color azul son las fases que se proponen, así como la generación de puntos de control virtuales (pentágono de fondo amarillo).

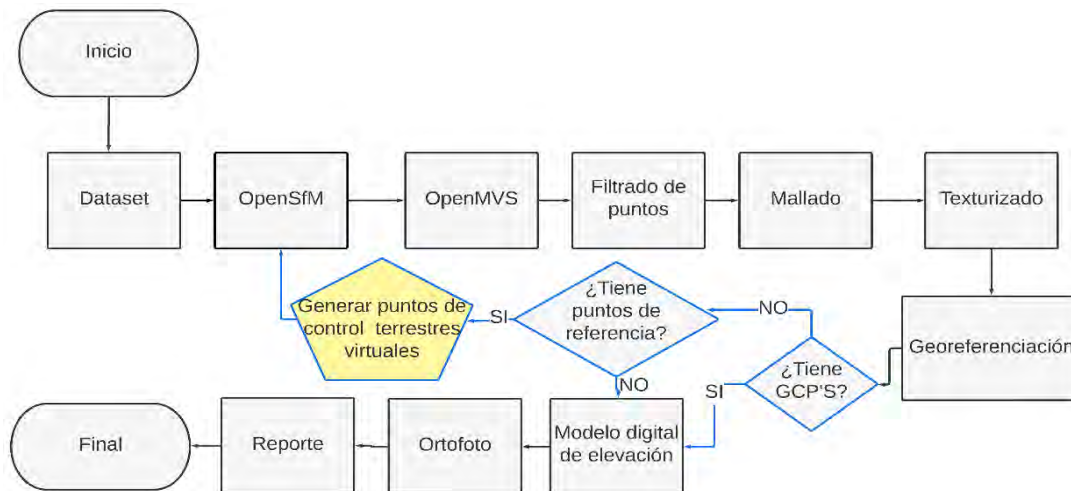


Figura 4.1 Esquema que representa la metodología de un proceso típico de fotogrametría, las figuras geométricas con borde en color azul son las fases que se proponen.

- A) El proceso inicia con un banco de imágenes del cual se genera una nube de puntos escasa mediante el algoritmo llamado OpenSfM [70].
- B) Dicha nube se densifica utilizando OpenMVS [35], [112].
- C) Se eliminan los puntos atípicos utilizando un filtro pasa medias.
- D) Se realiza el proceso de *Meshing* para obtener la malla de la nube.
- E) Se texturiza utilizando la herramienta poisson [132] para que la reconstrucción tenga mayor realismo.
- F) Se realiza la georreferenciación en coordenadas globales de acuerdo con el estándar EPSG:4326 WGS 84 [133] (en grados).
- G) Se verifica si la reconstrucción tiene puntos de control terrestre. Si la respuesta es afirmativa se genera el modelo digital de elevación, se obtiene una ortofoto de la reconstrucción y se realiza un reporte de calidad (proceso clásico). Sin embargo, si la reconstrucción no tiene puntos de control terrestre, se verifica si se conoce la distancia entre algunos puntos de la imagen, para poder mapearlos dentro de ella y con ello generar los puntos de control terrestre virtuales (VGCP por sus siglas en inglés) con el método propuesto en la siguiente sección.
- H) Una vez que se tienen los puntos de control terrestre, el proceso se repite para la siguiente reconstrucción.

4.2 Generación de puntos de control terrestre virtuales con cartulinas

El algoritmo que se propone para generar puntos de control virtuales utiliza: a) las distancias conocidas de cuatro puntos de los objetos de referencia en las imágenes (en este caso cartulinas blancas con un tamaño de 65 cm × 50 cm) y las distancias calculadas entre los mismos cuatro puntos, pero dentro de la reconstrucción en coordenadas geodésicas. Los pasos son los siguientes:

- A) Se genera una reconstrucción a partir de un conjunto de imágenes utilizando Web OpenDroneMap [69], metodología explicada anteriormente. Las imágenes deben contener un objeto visible desde la altura de trabajo, de preferencia un objeto geométrico a el cual se le conoce su tamaño, cómo una cartulina. Al objeto se le denominará “objeto de distancias conocidas”.

- B) En la interfaz gráfica de OpenDroneMap [69], seleccionar las 4 esquinas del objeto de referencia; además, se conoce la distancia entre los cuatro puntos. La posición de las 4 coordenadas se guarda y exportan en formato json. Dicho formato almacena los puntos en coordenadas geodésicas globales.
- C) Se realiza la conversión de coordenadas globales EPSG:4326 WGS 84 [133] las cuales están dadas en grados, a coordenadas de zona EPSG:32614 WGS 84 / UTM zone 14N [134] las cuales están en metros. La conversión se lleva a cabo mediante la librería pyproj [135].
- D) Se calculan las distancias de las 4 esquinas del objeto de referencia a partir de las posiciones obtenidas en la imagen.
- E) Se calcula el error con respecto a las distancias físicas del objeto.
- F) Se corrige el error de distancia moviendo los puntos geográficamente manteniendo la forma del objeto de referencia. Este proceso se lleva a cabo alineando dos pares de puntos de manera vertical y después diagonalmente.
- G) Se calcula el centro del objeto de referencia a partir de las coordenadas rectificadas.
- H) Las coordenadas rectificadas se convierten al estándar EPSG:4326 WGS 84 [133] de coordenadas globales en modo grados.
- I) Se guarda la información en el siguiente orden: 1) las coordenadas geodésicas del punto central, 2) la posición del punto central en términos espaciales en la imagen, 3) el nombre de la imagen dentro del banco de imágenes y 4) finalmente, el número de punto de control terrestre virtual que represente, considerando a todas las imágenes. Este proceso de almacenamiento se realiza para cada imagen que contiene al objeto de referencia y para todos los objetos de referencia que se tengan.

4.3 Preprocesamiento con COLMAP

En la Figura 4.2 se puede observar el esquema que representa la metodología para el preprocesamiento utilizando tanto COLMAP como la librería de Libimage-exiftool-perl [136], la cual permite modificar y o agregar los parámetros EXIF de la imagen. La nube de puntos binocular se genera directamente a partir de las dos imágenes de entrada, por lo que no requiere ningún tratamiento adicional.

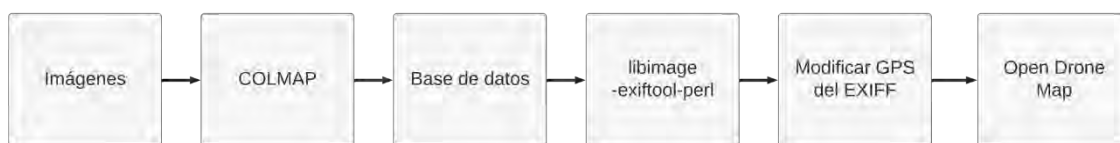


Figura 4.2 Preprocesamiento de las imágenes.

Como se puede observar en la Figura 4.2 :

- A) El proceso inicia con un banco de imágenes con el cual se genera una nube de puntos escasa utilizando COLMAP.
- B) Se genera una base de datos, con la información de todos los puntos de la nube, de puntos, así como la información de las cámaras calculada por COLMAP.
- C) Utilizando la librería Libimage-exiftool-perl [136] se modifica la información EXIF de cada imagen, en específico la información GPS de la imagen.

4.4 Re-proyección de nube de puntos a imagen

En la Figura 4.3 se puede observar el proceso para convertir los puntos (de la nube de puntos) en coordenadas de una imagen en específico, proceso que sirve para recuperar

información de la nube de puntos generada por la cámara monocular ya que ODM no permite proyectar una imagen dentro de la nube de puntos nativamente, por lo tanto, se tiene que hacer el proceso inverso. En el Anexo B: Resultados de re-proyección se pueden observar imágenes que sirven de referencia para visualizar los puntos ya re-proyectados.

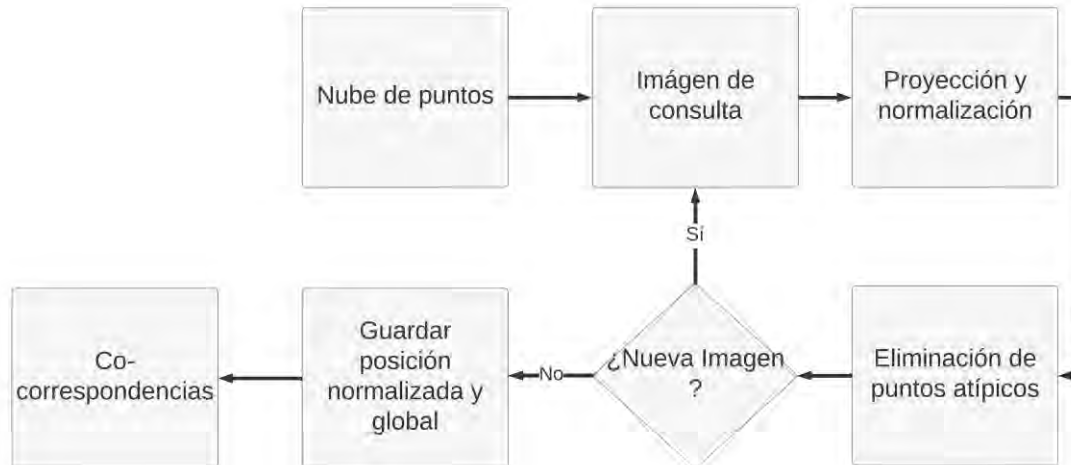


Figura 4.3 Esquema que representa el proceso de conversión de nube de puntos a imagen y a texto.

Como se puede observar en la Figura 4.3:

- A) El proceso inicia con una nube de puntos ya sea binocular o monocular.
- B) Cada punto de la nube de puntos se re-proyecta sobre la imagen de consulta y se normaliza.
- C) Se eliminan los puntos que caen fuera de las dimensiones de la imagen, ejemplo como al momento de realizar la re-proyección de un punto se puede calcular que para la imagen de consulta x se obtiene que las coordenadas deberían ser (5454, 8799). Sin embargo, la imagen tiene una resolución de 3400×2500 , por lo tanto, ese punto será eliminado de los puntos encontrados dentro de la imagen de consulta ya que no se encuentra dentro de las dimensiones de la imagen.
- D) Se repite el proceso por cada imagen del banco de imágenes.
- E) Se almacenan los datos en el siguiente formato: **a)** si es monocular: primero el nombre de imagen, seguido por el número de punto y al último las coordenadas tanto locales de la imagen como globales; el proceso se realiza para cada imagen en un solo archivo. **b)** En el caso de ser binocular se genera un archivo por cada imagen con el siguiente formato, coordenada local seguido por coordenada global. La diferencia de formato está dada porque la monocular entrega una menor cantidad de puntos que la binocular.

4.5 Algoritmo propuesto para generar puntos de control virtuales para la cámara binocular

Se propone un algoritmo para generar puntos de control virtuales utilizando las distancias calculadas por la cámara binocular y por SFM, de los mismos puntos dentro de la reconstrucción en coordenadas geodésicas. Los pasos para generar los puntos de control virtuales son los siguientes:

- A) Realizar la adquisición de un banco de imágenes, tanto con la cámara binocular como la monocular. La adquisición monocular se deberá adquirir de un vuelo de mayor altura entre los 50 m y los 100 m de toda la zona a reconstruir, teniendo un *overlap* del 80% (altura y traslape fijado considerando la Tabla 2.2) y, la adquisición con la cámara binocular se deberá adquirir en un vuelo de menor altura entre los 5 m y los 10 m obteniendo unos cuantos puntos centrados de la zona a reconstruir, por ejemplo, unos 4 puntos para un terreno plano de 20 mil metros, cada punto separado 20 m entre sí, intentado que los puntos tengan conexiones con varias imágenes de la monocular, los datos del vuelo binocular se guardan en formato SVO, el cual se descomprime posterior al vuelo.
- B) Aplicar un preprocesamiento con COLMAP para las imágenes monoculares.
- C) Generar una reconstrucción escasa a partir de un conjunto de imágenes monoculares utilizando el módulo OpenSFM de Web OpenDroneMap.
- D) Realizar una conversión de coordenadas globales *EPSG:4326 WGS 84* [133] (las cuales están dadas en grados) a coordenadas de zona *EPSG:32614 WGS 84 / UTM zone 14N* [134], las cuales están en metros. La conversión se lleva a cabo mediante la librería *pyproj* [135].
- E) Conversión de nube de puntos a puntos por imagen de la cámara monocular.
- F) Generar una nube de puntos por imagen de consulta de la cámara binocular.
- G) Conversión de nube de puntos a puntos por imagen de la cámara binocular.
- H) Para cada par de imágenes en las cuales se quiere generar un punto de control virtual, se realiza el cálculo de las correspondencias.
- I) Eliminar de la lista de correspondencias, todas las que no contengan un punto en la nube de puntos tanto binocular como monocular, esto se realiza tomando en cuenta *queryIdx* y *trainIdx*, que son parámetros del algoritmo de correspondencia, así como la matriz Fundamental determinada por el algoritmo RANSAC (Random Sample Consensus).
- J) Buscar un punto inicial, considerando la posición del punto y en qué número de imágenes parece ese punto.
- K) Buscar los puntos circulares al punto inicial en un radio de 500 px.
- L) Obtener la diferencia en los ejes X, Y, Z entre las coordenadas binoculares y monoculares para cada punto y calcular el error promedio.
- M) Corregir el error de distancia moviendo las coordenadas del punto inicial, considerando el error promedio por eje.
- N) Convertirlo a formato *EPSG:4326 WGS 84* [133] (coordenadas globales en grados).
- O) Se guarda la información en el siguiente orden: 1) las coordenadas geodésicas del punto central, 2) la posición del punto central en términos espaciales en la imagen, 3) el nombre de la imagen dentro del banco de imágenes y 4) finalmente, el número de punto de control terrestre virtual que represente, considerando a todas las imágenes. Este proceso de almacenamiento se realiza para cada imagen que contenga dicho objeto de referencia y para todos los objetos de referencia que se tengan.

En la Figura 4.4 se muestra el preproceso llevado por la cámara monocular para poder generar las re-proyecciones guardándolas todas en un solo archivo de texto. En la Figura 4.5 se muestra el proceso realizado por la cámara binocular para poder generar las re-proyecciones y guardarlas en un archivo de texto para cada imagen. En la Figura 4.6 se puede observar un diagrama de flujo con el algoritmo simplificado para generar correspondencias, tomando en cuenta la información generada en las etapas anteriores.

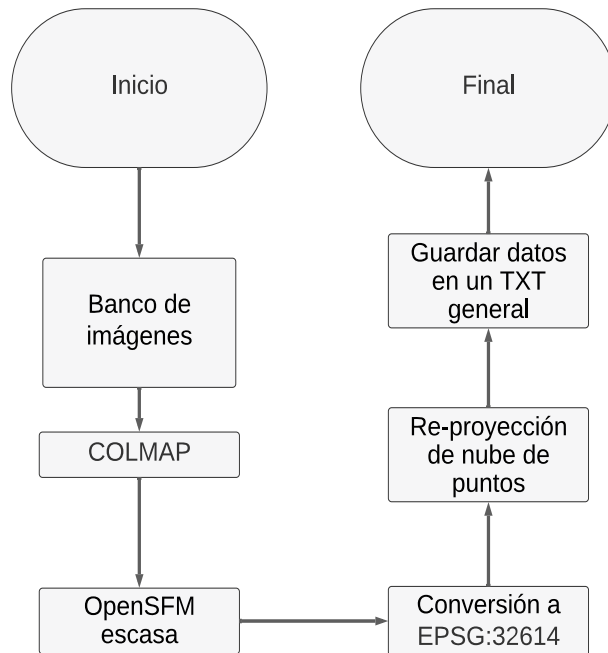


Figura 4.4 Esquema que representa el proceso de la cámara monocular para generar el TXT de la nube de puntos.

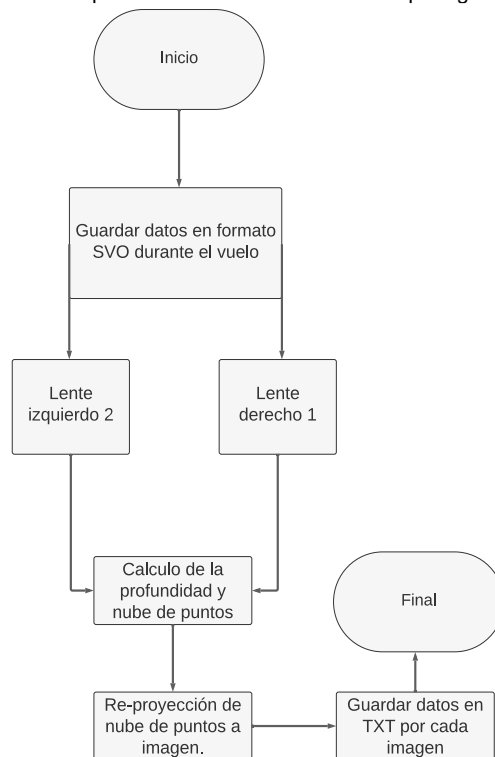


Figura 4.5 Esquema que representa el proceso de la cámara binocular para generar los TXT de las nubes de puntos.

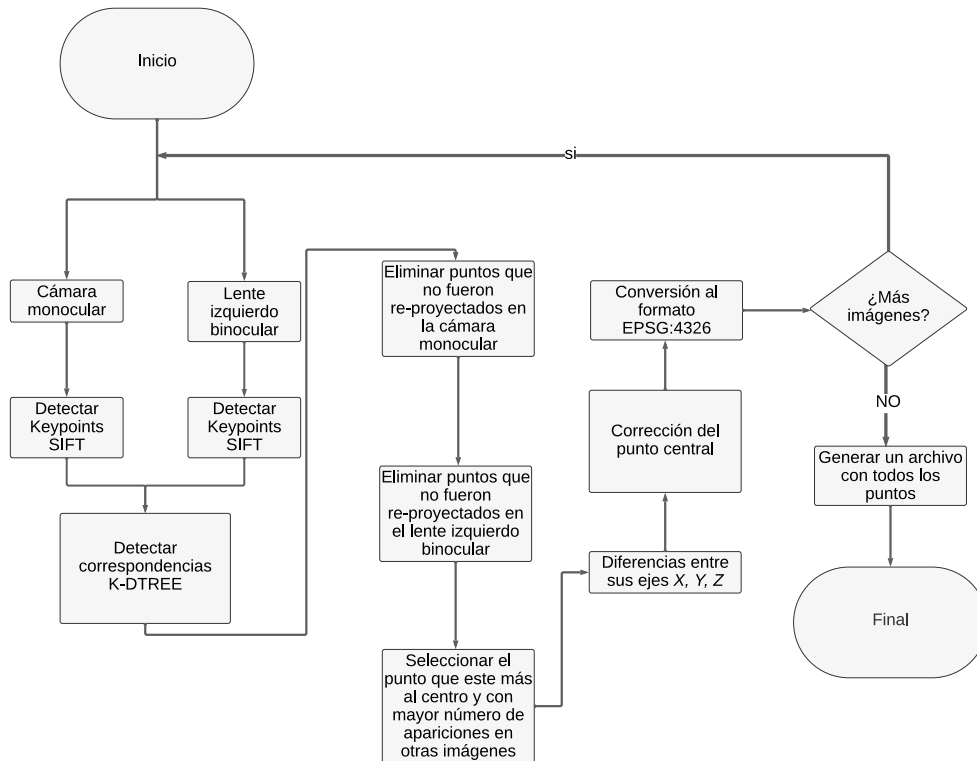


Figura 4.6 Esquema que representa el proceso para realizar las correspondencias entre la imagen monocular y las binoculares.

En resumen, para generar un punto de control terrestre virtual utilizando una cámara binocular y una monocular, se realizan los siguientes pasos:

- 1.- Generar una reconstrucción escasa de las imágenes monoculares.
- 2.- Re-proyectar esos puntos dentro de las imágenes originales, guardando la información en un archivo de texto.
- 3.- Calcular la profundidad para cada par de las imágenes binoculares, utilizando técnicas estéreo (SDK de ZED) y guardándolas individualmente.
- 4.- Búsqueda de correspondencias entre las imágenes monoculares y binoculares, seleccionado las imágenes que más correspondencia tengan.
- 5.- Eliminación de puntos que no se encuentren en la nube de puntos monoculares y en la binocular (la que fue calculada por la disparidad).
- 6.- Selección de un punto que esté en el centro y que se encuentre en la mayor cantidad de imágenes monoculares posibles (usando el archivo de texto del paso 2).
- 7.- Corregir la posición del punto seleccionado tomando en cuenta la posición de los otros puntos encontrados a su alrededor (medidos por la cámara binocular contra la monocular).
- 8.- Guardar la información en formato de GCP.

4.6 Herramientas utilizadas (software y equipo)

Para el desarrollo del proyecto se contó con las siguientes herramientas:

- Una computadora con los siguientes componentes: INTEL I7 8700k, NVIDIA RTX 3060, XPG D40 64GB RAM 3200 MHz, XPG S40G 512GBm.2.
- Docker [137]: Docker es una plataforma de software que permite crear, probar e implementar aplicaciones rápidamente. Docker empaqueta software en contenedores que incluyen todo lo necesario para que el software se ejecute, incluidas bibliotecas, herramientas de sistema, código y tiempo de ejecución [138].
- OpenCV [139]: Es una biblioteca de código abierto que contiene implementaciones que abarcan más de 2500 algoritmos. Además, está especializada en el sistema de visión artificial y machine Learning [140].
- PyTorch versión 1.13.1 [141]: PyTorch proporciona un paquete de Python para funciones de alto nivel como el cálculo de tensor (como NumPy) con una fuerte aceleración de GPU y TorchScript [142].
- ODM [19]: OpenDroneMap (ODM) es un kit de herramientas de código abierto para procesar imágenes capturadas por drones. OpenDroneMap convierte las imágenes simples obtenidas de las cámaras en datos geográficos tridimensionales, fotografías que se pueden utilizar, posteriormente, en combinación con otros conjuntos de datos geográficos [143].
- Ubuntu 20.04 [144]: El sistema operativo Ubuntu es más que una distribución de código abierto basada en Debian, otro sistema operativo, cuyo punto común es Linux [145].
- Ros Noetic [146]: El sistema operativo de robot (ROS) es un conjunto de bibliotecas de software y herramientas de código abierto que ayudan a crear aplicaciones de robots. Desde controladores hasta algoritmos de última generación y con potentes herramientas de desarrollo. ROS nació en 2007 [147].
- CUDA 11.1 [148]: CUDA es una plataforma informática paralela y un modelo de programación creado por NVIDIA. CUDA ayuda a los desarrolladores a acelerar sus aplicaciones aprovechando el poder de los aceleradores GPU. CUDA fue lanzada en el 2006 por NVIDIA [149].
- ZED SDK 3.2.2 [150]: Es un SDK el cual permite controlar las cámaras de la compañía Stereolabs, entre las cuales se incluye la ZED 2.

4.7 Equipo para adquisición de imágenes

La etapa de adquisición de las imágenes se utilizaron los siguientes dispositivos:

- Dron Monocular
- Dron Binocular

Dron Monocular

El dron monocular es el DJI Mavic pro platinum, el cual tiene una cámara equipada con un sensor de 12.35 M CMOS de 1/2.3" con un tamaño de imagen de 4000 × 3000, con información EXIF. Se utilizó junto con la aplicación para celulares de DroneDeploy [88], el

cual tiene la función de generar las rutas de vuelo, tomando en cuenta la altura y traslape deseadas. En la Figura 4.7 se muestra una foto del dron.



Figura 4.7 DJI MAVIC PRO PLATINUM [151].

Dron Binocular

El dron binocular es un DJI AGRAS T10 (ver Figura 4.8) modificado con una cámara binocular (la cual está montada junto con un estabilizador mecánico en la parte inferior a la altura del actuador). La cámara binocular es la ZED 2 (ver la Figura 4.9) la cual tiene una resolución de 2208×1242 por lente, en formato JPG, con un GSD de 0.5 a 1 cm. La cámara ZED 2 no cuenta con información EXIF. Como tarjeta controladora de la ZED 2 se utiliza la JETSON NANO (ver la Figura 4.10), la cual se encarga únicamente de adquirir las imágenes y guardarlas en un formato SVO, el cual es original de Stereolabs.



Figura 4.8 DJI AGRAS T10 [152].



Figura 4.9 ZED 2 [153].



Figura 4.10 Jetson Nano [154].

Capítulo 5: Pruebas y Resultados

En el presente capítulo se presentan los 7 casos de experimentación realizadas para evaluar los métodos descritos en el capítulo 4, junto con los bancos de imágenes.

5.1 Adquisición de Bancos de imágenes

Se presentan los 7 bancos de imágenes utilizados en el desarrollo de la presente tesis:

- Banco de imágenes 1.
- Banco de imágenes 2.
- Banco de imágenes 3.
- Banco de imágenes 4.
- Banco de imágenes 5.
- Banco de imágenes de ODM [93].
- Banco de imágenes simulado.

Los bancos de imágenes 1, 2 y 4 son bancos de imágenes capturados con el dron monocular, cada banco fue capturado en un terreno diferente.

Los bancos de imágenes 3 y 5 son binoculares, ya que fueron capturados tanto con una cámara monocular como una binocular y a diferentes alturas, cada banco fue capturado en un terreno diferente.

El banco de imágenes de ODM [93] es un banco de imágenes publicado en los foros de ODM, fue seleccionado ya que la zona muestreada es una parcela agrícola.

El último banco de imágenes fue una simulación generada en el entorno de UnrealEngine, junto con AirSim.

5.1.1 Banco de imágenes 1

Se realizaron 3 vuelos sobre una parcela ubicada en el campo San Pablo en el municipio de Jojutla, estado de Morelos, México; variando la altura de cada vuelo, a 50 m, 75 m y a 100 m (alturas establecidas tomando en cuenta la información de la Tabla 2.2), obteniendo 147 fotos, 86 fotos y 58 fotos respectivamente. Las fotografías fueron tomadas el día 19 de noviembre del 2022, a partir de las 7:00 y hasta las 7:40 a. m. Dentro del terreno se distribuyeron 5 cartulinas blancas que tiene un tamaño de 65 cm \times 50 cm, en la Figura 5.1 se puede observar un ejemplo. Las fotografías tienen una resolución de 4000 \times 3000 píxeles, en formato JPG, con un GSD variado por la altura que va desde los 1.6 a 3.0 cm y con su geolocalización en el EXIF. Ejemplos de este banco de imágenes se muestran en la Figura 5.2.



Figura 5.1 Ejemplo de punto de control del primer vuelo.



Figura 5.2 Tres imágenes de muestra del conjunto de datos propio del primer vuelo adquirido con la cámara monocular.

5.1.2 Banco de imágenes 2

Se realizaron 3 vuelos sobre una parcela ubicada en el campo Higuera Viejo en el municipio de Jojutla, estado de Morelos, México; variando la altura de cada vuelo, a 50 m, 75 m y a 100 m (alturas establecidas tomando en cuenta la información de la Tabla 2.2), obteniendo 142 fotos, 81 fotos y 62 fotos respectivamente. Las fotografías fueron tomadas el día 1 de diciembre del 2022, a partir de las 4:00 p. m. y hasta las 4:40 p. m. Dentro del terreno se distribuyeron 5 cartulinas verdes que tiene un tamaño de 65 cm \times 50 cm con un punto central de color negro, en la Figura 5.3 se puede observar un ejemplo. Las fotografías tienen una resolución de 4000 \times 3000, en formato JPG, con un GSD de variado por la altura que va desde los 1.6 a 3.0 cm y con su geolocalización en el EXIF, ejemplos de este banco de imágenes se muestran en la Figura 5.4.



Figura 5.3 Ejemplo de punto de control del segundo vuelo.



Figura 5.4 Tres imágenes de muestra del conjunto de datos propio del segundo vuelo adquirido con la cámara monocular.

5.1.3 Banco de imágenes 3

Se realizaron 3 vuelos sobre una parcela con caña de dos meses aproximadamente, ubicada en el municipio de Tlaltizapán, estado de Morelos, México.

- Los dos primeros vuelos fueron realizados con la cámara binocular. En los vuelos el dron iba a 4 puntos determinados a una altura de 10 m y poco a poco iba bajando la altura hasta los 2 m para posteriormente subir a 10 m y proseguir.
- La diferencia principal entre los dos primeros vuelos es que, en el primero se ocupó un estabilizador mecánico y en el otro no. Es decir, en el primer vuelo se tiene la cámara colocada directamente sobre el chasis y el otro vuelo la cámara está sujeta al estabilizador mecánico, esto con la finalidad de ver si las altas frecuencias del dron afectan negativamente al estabilizador (ver Figura 5.5).
- El tercer vuelo fue realizado por el dron con la cámara monocular a una altura de 100 m, (el dron fue atacado por aves por esta razón no se pudo volar más bajo).

Las fotografías fueron tomadas el día 20 de mayo del 2023, a partir de las 4:00 p. m. y hasta las 4:40 p. m. Dentro del terreno se distribuyeron 4 cartulinas verdes que tienen un tamaño de 65 cm × 50 cm con un punto central de color negro, en la Figura 5.6 se puede observar un ejemplo de estas. Las fotografías tienen una resolución de 4000 × 3000, en formato JPG, con un GSD de 3 cm y con su geolocalización en el EXIF. Se tienen 46 imágenes para el vuelo con la cámara monocular y para cada vuelo con la cámara binocular se tienen 2 mil imágenes, esto debido a que la cámara binocular no cuenta con una opción de capturar la imagen cada cierto tiempo, entonces va adquiriendo fotos de manera constante. La razón principal de la diferencia de formato es que la cámara monocular tiene una resolución 4K (4000 × 3000), y la cámara Zed está limitada a 2K (2208 × 1242 por objetivo). Ejemplos de este banco de imágenes se muestran en la Figura 5.7.



Figura 5.5 Ejemplo de la imagen del lente izquierdo del tercer vuelo adquirido con la cámara binocular a 10 m.



Figura 5.6 Ejemplo de punto de control en el tercer vuelo.



Figura 5.7 Tres imágenes de muestra del conjunto de datos propio del tercer vuelo adquirido con la cámara monocular.

5.1.4 Banco de imágenes 4

Se adquirieron 96 imágenes con dron DJI mini SPARK de un terreno ubicado en el municipio de Jojutla, estado de Morelos, México a una altura de 35 metros. Las fotografías fueron tomadas el día 17 de junio del 2020, a partir de las 10:00 a. m. y hasta las 11:00 a. m. En el terreno se distribuyeron 12 puntos de control terrestres (GCP por sus siglas en inglés) en hojas oficio, en la Figura 5.9 se puede observar un ejemplo de un GCP. Las fotografías tienen una resolución de 3968×2976 , en formato JPG, con un GSD de 2.60 cm/píxel y con su geolocalización en el EXIF, ejemplos de este banco de imágenes se muestran en la Figura 5.8. El dron DJI mini SPARK tiene una cámara con un sensor de 12 MP CMOS de 1/2.3" y con una distancia focal de 2.6.



Figura 5.8 Cuatro imágenes de muestra del conjunto de datos propio del cuarto vuelo adquirido con la cámara monocular.



Figura 5.9 Ejemplo de punto de control [155].

5.1.5 Banco de imágenes 5

Con el apoyo del Tecnológico de Monterrey campus Guadalajara, se realizaron 3 vuelos sobre una parcela con agave, ubicada en el municipio de Zapopan, estado de Jalisco, México.

- Los dos primeros vuelos fueron realizados con la cámara monocular. Las rutas de vuelo fueron definidas por la aplicación de DroneDeploy, donde se fijó un 80% de traslape, además de configurar la altura a 100 m y 50 m respectivamente.
- El tercer vuelo fue realizado con el dron con la cámara binocular, el cual iba siguiendo una ruta de vuelo parecida al del dron monocular, pero de forma manual y a una altura de 10 m, esto con la finalidad de poder escoger en pos-procesado imágenes donde se obtuvieran mejores correspondencias entre imágenes monocular y binocular. En la Figura 5.10 se observa un ejemplo.

Las fotografías fueron adquiridas el día 1 de diciembre del 2023, a partir de la 1:00 p. m. y hasta las 1:30 p. m. Dentro del terreno se distribuyeron 4 cartulinas verdes que tiene un tamaño de 65 cm × 50 cm con un punto central de color negro, en la Figura 5.11 se puede observar un ejemplo. Las fotografías tienen una resolución de 4000 × 3000, en formato JPG, con un GSD de 3 cm y con su geolocalización en el EXIF

Se obtuvieron 138 imágenes para los vuelos monoculares (algunos ejemplos se pueden ver en la Figura 5.12) y para cada vuelo binocular se tienen 2 mil imágenes igual que en el banco de imágenes 3.



Figura 5.10 Ejemplo de la imagen del lente izquierdo del quinto vuelo adquirido con la cámara binocular a 10 m.



Figura 5.11 Ejemplo de punto de control en el quinto vuelo.



Figura 5.12 Tres imágenes de muestra del conjunto de datos propio del quinto vuelo adquirido con la cámara monocular.

5.1.6 Banco de imágenes de ODM [93]

OpenDroneMap dataset es una página web donde las personas publican los recorridos que hacen con sus drones, guardando información que va desde el tipo de cámara, el tipo de dron, sensores y el lugar en donde se realizó el recorrido. OpenDroneMap dataset forma parte de la comunidad OpenDroneMap, que se especializa en realizar fotogrametría con drones de forma gratuita. En la Figura 5.13 se pueden observar 4 imágenes capturadas durante el recorrido.

De los foros de OpenDroneMap/dataset [93]. Se seleccionó un conjunto de 96 imágenes multiespectrales de un vuelo con el dron Quantix Mapper con una resolución de 608×456 píxeles, de un terreno de 4 acres en la ciudad de Lathrop, California. Las imágenes tienen la información EXIF de la cámara, sin GCP ni RTK y con su geolocalización.

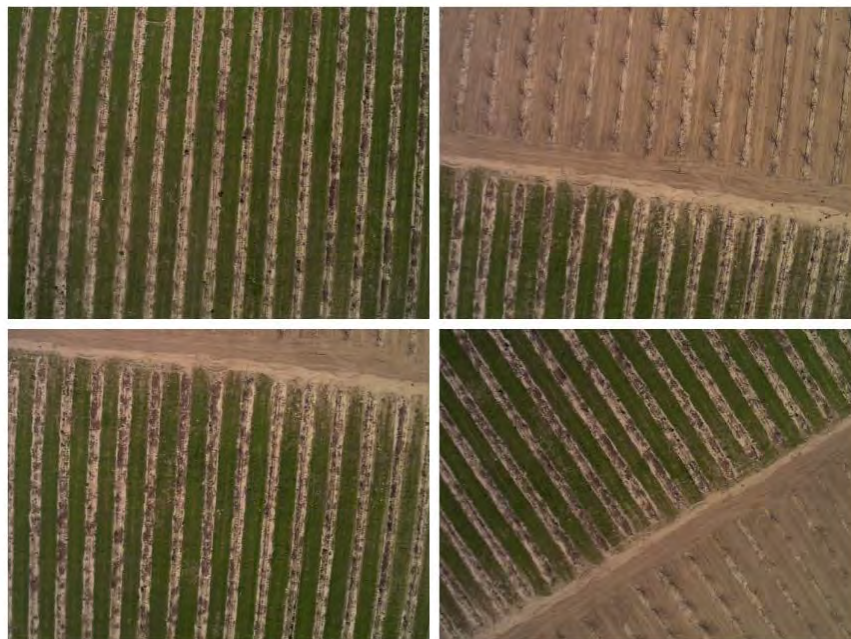


Figura 5.13 Dataset Quantix Mapper [93].

5.1.7 Banco de imágenes simulado

Utilizando UnrealEngine junto con Airsim, se realizaron varios recorridos de 3 diferentes escenarios, variando su altura, el modo de simulación. En AirSim se realizó el cambio del modo Computer Vision para adquirir las imágenes, al modo de Multirotor para adquirir los datos provenientes del LIDAR, para almacenarlos en un archivo de texto.

El primer escenario es Multi-biome de Brushify Environment Shaders Pack [156], el cual está compuesto por 3 diferentes montañas y un lago en la parte central. Del escenario se seleccionó una parte de 225 m^2 de la montaña de color rojo, a una altura variable debido a que la montaña tiene una inclinación. En la Figura 5.14 se puede observar todo el escenario.



Figura 5.14 Mapa de Multi-biome [156].

El segundo escenario considerado es el de City Park de SilverTM [157], el cual es un parque que consta de varias zonas, como por ejemplo un campo de fútbol, de béisbol, etc. Se seleccionó una parte de 225 m² del campo de béisbol, a una altura de 9.45 m y a 4.45 m para el caso estéreo. En la Figura 5.15 se puede observar todo el escenario.



Figura 5.15 Mapa de City Park SilverTM [157].

El último escenario es uno propio, el cual consta de una zona de 225 m² de desierto con piedras de diferente tamaño y diseños; además de algunas ramas para darle más textura al escenario, con una altura de 7.5 m y a 5 m para el caso estéreo. En la Figura 5.16 se puede observar todo el escenario.



Figura 5.16 Escenario simulado propio.

Los bancos de imágenes resultantes cuentan con 100 imágenes de cada una de las 3 cámaras inferiores del dron, con una resolución de 2560×1440 (simulando la resolución de la cámara estéreo Zed 2). Se crean 10 hileras con 10 fotografías por cada una. La distancia entre imágenes es de 1.5 m entre una y otra; la imagen estéreo es generada una vez por cada hilera (de 10 fotos) de manera aleatoria.

5.2 Calibraciones primer caso de experimentación

En esta sección se revisa el proceso de calibración de la cámara ZED 2, la cual es una cámara binocular, por lo tanto, se tienen que calibrar las dos lentes de manera individual. Existen algunos algoritmos que son capaces de calibrar considerando las dos lentes, tal como ROS CALIBRATION, el cual calibra la cámara binocular en un solo paso.

Para la presente investigación se contó con una calibración de fábrica, la cual se puede obtener de la página oficial de Stereolabs, solamente con el identificador de la cámara. Además de la calibración de fábrica se generaron 4 calibraciones adicionales para la cámara ZED 2, las cuales son:

- I. Calibración con ROS y contar con un objetivo de tamaño de hoja carta (2 cm entre cuadros).
- II. Calibración con el sistema de Stereolabs, utilizando un monitor.
- III. Calibración con OpenCV y contar con un objetivo de tamaño de hoja carta (2 cm entre cuadros).
- IV. Calibración con OpenCV y contar con un objetivo de 90 cm (13 cm entre cuadros).

5.2.1 Calibración con ROS

Para llevar a cabo la calibración de la cámara ZED 2 en el ambiente de ROS, se realizaron los siguientes pasos:

- A. Se imprimió un patrón tipo tablero de ajedrez con unas medidas de 8×6 cm [130], el cual se colocó en una superficie plana.
- B. Se instaló el sistema de ROS.

- C. Se ejecutó el programa llamado ROS *binocular camera calibration* (Figura 5.17).
- D. Se realizaron movimientos lentos en todos los sentidos hasta que se active el botón de calibrar como se muestra en la Figura 5.18.
- E. El programa realiza la calibración y entrega un resultado como el de la Figura 5.19.

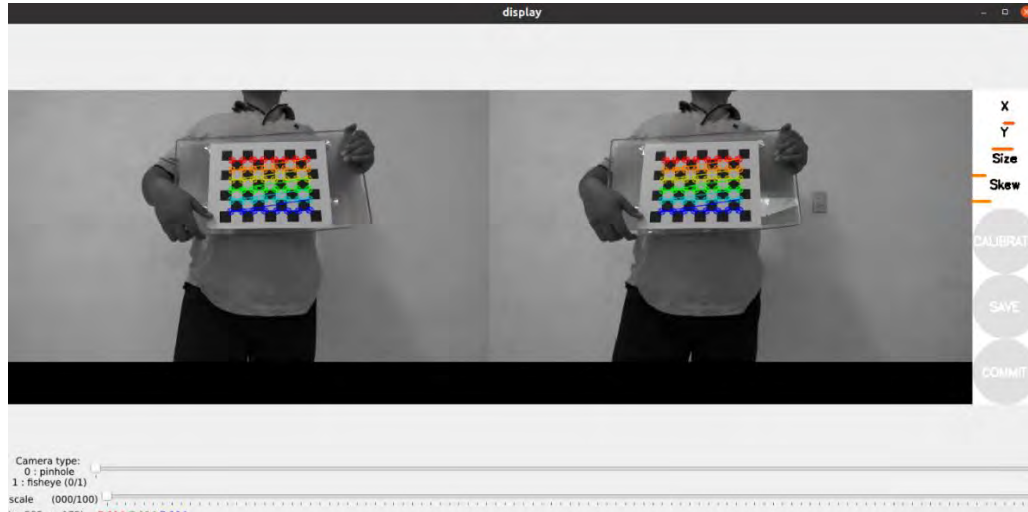


Figura 5.17 Ventana del programa de calibración.



Figura 5.18 Calibración ZED.

```

[narrow_stereo/left]
camera matrix
1090.097289 0.000000 1087.155779
0.000000 1095.523334 624.929572
0.000000 0.000000 1.000000

distortion
-0.033473 0.031981 0.002764 -0.002348 0.000000

rectification
0.999956 -0.001187 0.009327
0.001159 0.999995 0.002981
-0.009331 -0.002970 0.999952

projection
1124.657287 0.000000 1058.949371 0.000000
0.000000 1124.657287 625.039284 0.000000
0.000000 0.000000 1.000000 0.000000
# oSf version 5.0 parameters

[image]

width
2208

height
1242

[narrow_stereo/right]
camera matrix
1084.901127 0.000000 1082.991226
0.000000 1091.195831 618.519897
0.000000 0.000000 1.000000

distortion
-0.038104 0.023485 0.000592 -0.005544 0.000000

rectification
0.999996 0.000347 0.002953
-0.000338 0.999996 -0.002976
-0.002954 0.002975 0.999991

projection
1124.657287 0.000000 1058.949371 -580.211409
0.000000 1124.657287 625.039284 0.000000
0.000000 0.000000 1.000000 0.000000

```

Figura 5.19 Valores de calibración para cada lente.

5.2.2 Calibración Stereolabs

Se calibró utilizando el software especializado de Zed, llamado “Zed Calibration”, el proceso fue el siguiente: Con la cámara Zed 2 conectada al PC y un monitor plano, se inicializa el programa Zed Calibration el cual muestra en la pantalla, un tablero tipo ajedrez de 22×10 cuadros con un objetivo en el centro (como se muestra en la Figura 5.20). Entonces, se sigue el objetivo mostrado por toda la pantalla alejando la cámara Zed tal como el programa lo requiere, al finalizar se genera un archivo de calibración ubicado en “C:\ProgramData\Stereolabs\settings\SNXXXXXX.conf”.

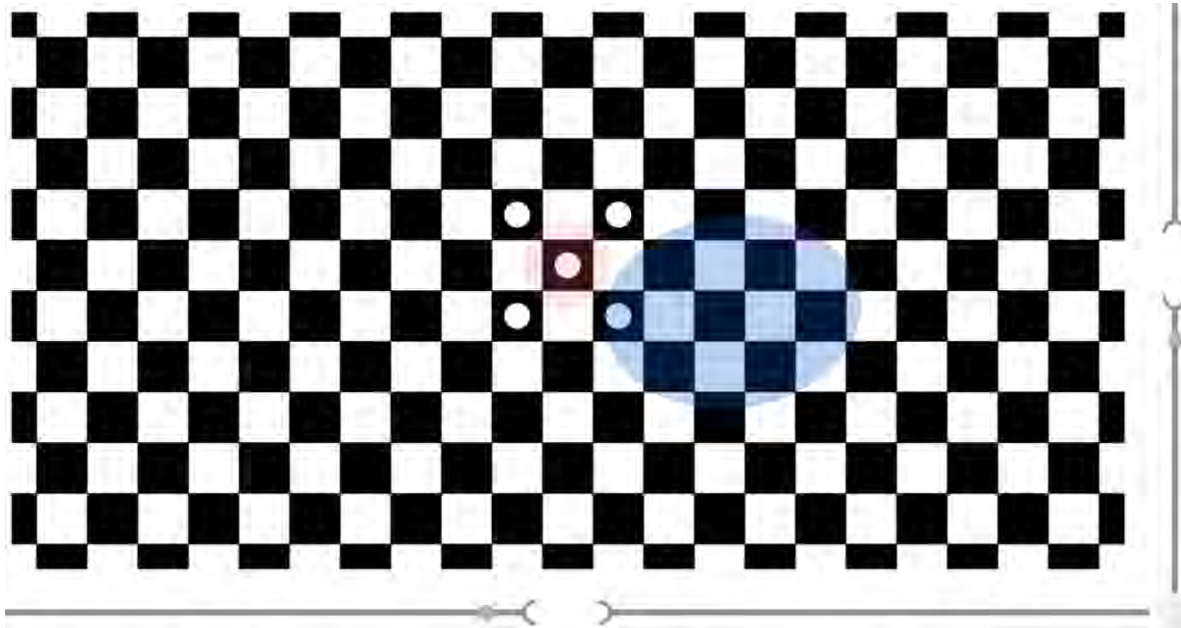


Figura 5.20 Programa Zed Calibration.

5.2.3 Calibración OpenCV tamaño carta

Se calibró utilizando las librerías de OpenCV 4.5.2 que siguen el algoritmo del artículo “A flexible new technique for camera calibration” [158]. El proceso fue el siguiente: se adquirieron 10 fotografías con cada lente de un patrón tipo tablero de ajedrez, variando unos 10 cm de distancia y 10° el ángulo de visión. La Figura 5.21 muestra un ejemplo.

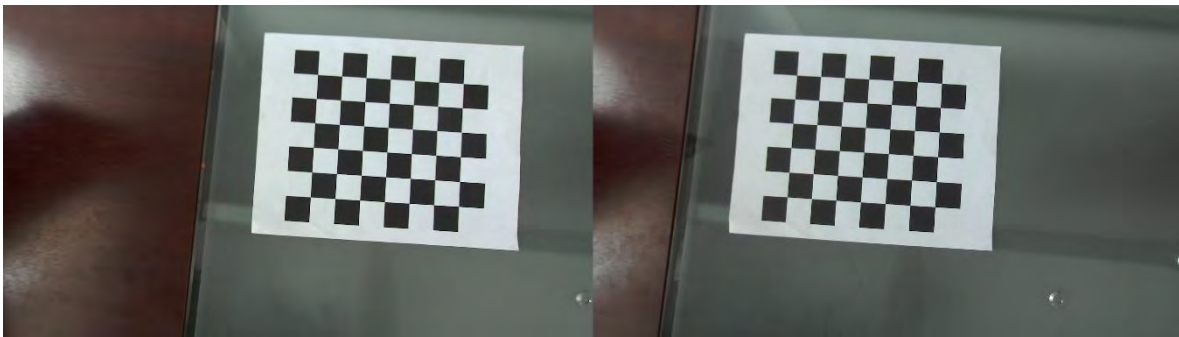


Figura 5.21 Patrón tipo tablero de ajedrez tomado por la ZED 2.

Posteriormente, se detectaron las esquinas del tablero en todas las imágenes, como se observa en la Figura 5.22, a continuación, con todos los datos obtenidos, se prosiguió con la calibración utilizando las librerías de OpenCV 4.5.2. Y se obtuvo un error de re-proyección de 0.064 px para el lente izquierdo y de 0.084 px para el derecho.

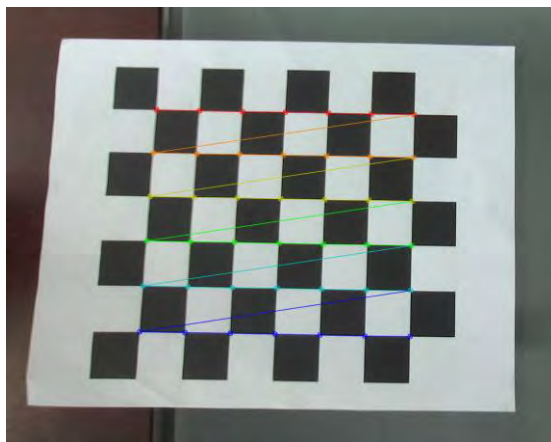


Figura 5.22 Detección de esquinas de ajedrez OpenCV.

5.2.4 Calibración OpenCV 13 cm

De la misma forma que en la calibración de tamaño carta, se calibró utilizando librerías de OpenCV 4.5.2 descritas en el algoritmo del artículo “A flexible new technique for camera calibration” [158], con la modificación en la distancia de captura y el tamaño del patrón de calibración. El proceso seguido fue el siguiente: se adquirieron 20 fotografías con cada lente de un patrón tipo tablero de ajedrez con una cuadrícula de 8×6 y de tamaño de 13 centímetros por cada cuadrado, a una distancia de 3 m, variando el ángulo de visión. La Figura 5.23 muestra un ejemplo.



Figura 5.23 Patrón tipo tablero de ajedrez tomado por la ZED 2.

Posteriormente, se detectaron las esquinas del tablero en todas las imágenes, a continuación, con todos los datos obtenidos, se prosiguió con la calibración utilizando las librerías de OpenCV 4.5.2, se obtuvo un error de re-proyección de 0.024 px para el lente izquierdo y de 0.027 px para el derecho.

5.2.5 Comparación de las diferentes calibraciones

Con la finalidad de medir la presión de la cámara Zed 2, con las 5 diferentes calibraciones con las que se cuenta, se realizó una prueba la cual consiste en colocar la cámara a 10 diferentes distancias (de 1 m a 10 m), midiendo tanto físicamente como utilizando la cámara binocular, pero variando la calibración utilizada.

El cambio en las calibraciones de la cámara ZED 2 se realiza modificando los valores de calibración originales de la cámara, los cuales se encuentran por defecto en el archivo "C:\ProgramData\Stereolabs\settings\SNXXXXXX.conf".

Los valores fueron reemplazados con los arrojados por las diferentes calibraciones y se probaron utilizando el programa ZED Depth Viewer que es nativo de ZED, el cual permite conocer la distancia a la cual se encuentra un punto seleccionado.

Las 10 mediciones fueron realizadas variando un metro entre ellas, partiendo de 1 metro hasta aproximadamente 10 m. Posteriormente, se compararon los resultados obtenidos con las diferentes calibraciones, ver Tabla 5.1, donde los nombres de las columnas indican: Distancia medida entre la cámara y el objetivo, ROS menciona la calibración realizada por ROS, Original es la calibración de fábrica, PC es la calibración generada por *Zed calibration*, Carta es la calibración de OpenCV con la hoja tamaño carta y 13 cm es la calibración de OpenCV con los cuadrados de 13 cm por lado.

Para una mejor visualización de como el error de la medición crece conforme se aleja el objetivo de la cámara, en la Figura 5.24 se muestra una gráfica comparativa, donde el eje X es la distancia entre el objetivo y la cámara, el eje Y es el error entre la distancia real y la medida en metros.

Tabla 5.1 Medición de distancias con la ZED 2 utilizando diferentes calibraciones.

Distancia	ROS	Original	PC	Carta	13 cm
1.12	1.12	1.08	0.97	1.12	1.13
2	2.01	1.95	1.7	2.01	2.05
3.02	2.98	2.87	2.3	3.01	3.07
4	3.94	3.75	2.84	3.98	4.05
5	4.9	4.7	3.3	4.89	5.05
6	5.8	5.56	3.7	5.8	6.06
7	6.7	6.3	4.04	6.7	7
8	7.6	7.25	4.46	7.7	7.98
9	8.55	8	4.7	8.5	9
9.77	9.47	8.5	4.8	9.2	9.72

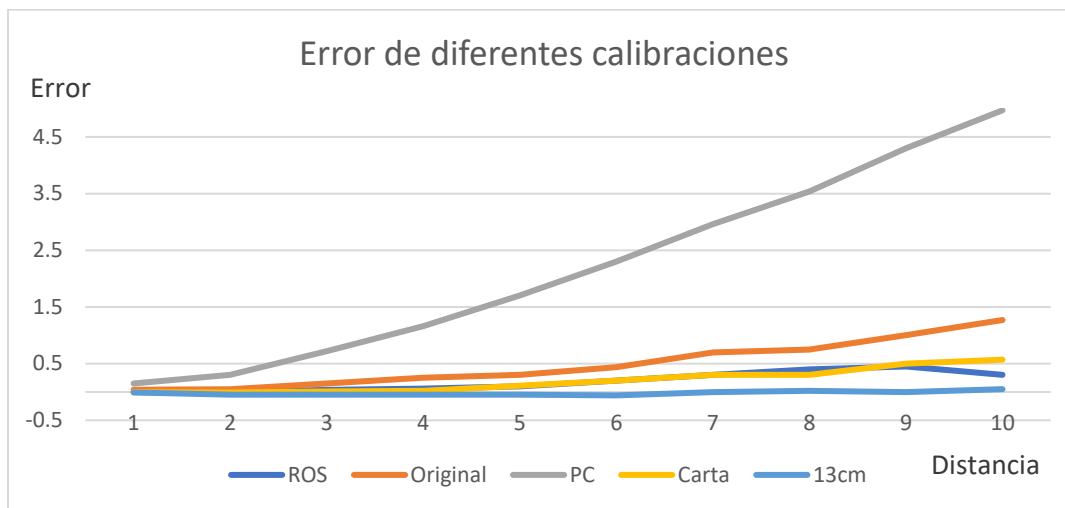


Figura 5.24 Error Zed 2.

En la Figura 5.24 se puede observar gráficamente el error medido en metros para cada calibración (eje Y), dando como resultado que la mejor calibración es la de 13 cm, seguida por ROS, Carta, original y la peor calibración fue la realizada por el programa de Zed calibration, cabe resaltar que mientras más alejado se encuentre la cámara es menos estable su medida y que los valores reportados son valores promedios.

5.3 Caso 2 experimentación con puntos característicos (las 3 etapas)

El objetivo del caso es evaluar el desempeño de las técnicas de detección, descripción de puntos característicos combinados con algoritmos de correspondencia. Se realizaron 6 diferentes combinaciones de algoritmos de la siguiente forma, donde el primero se utiliza para detectar y el segundo para realizar las correspondencias entre los puntos: 1) ORB-BF, 2) ORB-KNN 3) SIFT-BF, 4) SIFT-KNN, 5) SIFT-KD-TREE y 6) SuperPoint-SuperGlue. Se evaluaron en dos Datasets.

Las Figuras 5.25, 5.26, 5.27, 5.28, 5.29 y 5.30 muestran un ejemplo del resultado de cada combinación en el banco de imágenes de ODM y en las Figuras 5.31, 5.32, 5.33, 5.34, 5.35, 5.36 y 5.37 se muestra el resultado en Banco de imágenes 4. Para las pruebas se toman dos imágenes de cada conjunto y los resultados son específicos para ese par de imágenes, sin embargo, en las tablas de resultados se muestra el promedio de todas las imágenes. Las fotografías tienen una ligera diferencia entre ellas.

a) Dataset ODM :

- Combinación **ORB-BF**: ORB detectó 500 puntos característicos en cada imagen de los cuales 145 consiguieron su correspondencia en la siguiente imagen (lo que equivale al 29%), en un tiempo de 0.058 segundos en promedio. Entre las correspondencias existen un gran número de falsos positivos (80%) como se observa en la Figura 5.25. Todas las líneas deberían estar en la misma dirección.

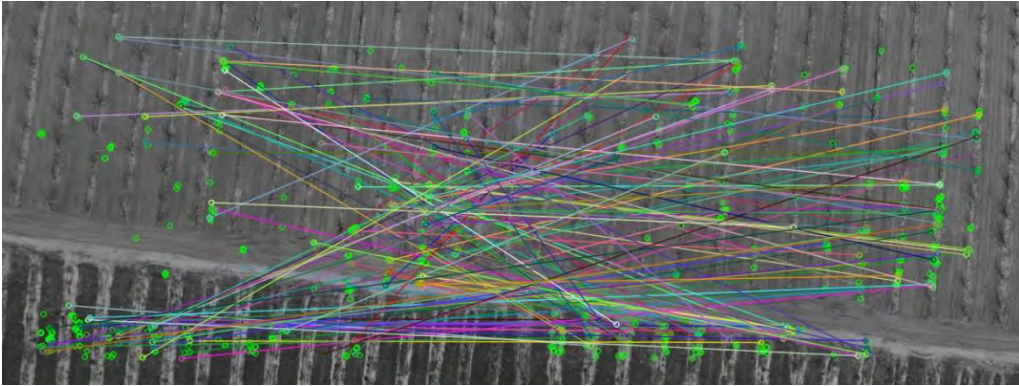


Figura 5.25 Ejemplo de aplicación de ORB-BF en una imagen de ODM dataset.

- Combinación **ORB-KNN**: ORB detecta 500 puntos característicos en ambas imágenes en zonas pertenecientes a bordes, de los cuales KNN sólo logra realizar 19 correspondencias entre ellas (lo que equivale al 3.8%), en un tiempo de 0.056 segundos en promedio. En este caso en específico se puede observar una precisión del 0%, ya que ningún punto tiene una correspondencia correcta debido a que el terreno contiene muchas zonas con apariencia visual muy similar, ver Figura 5.26.

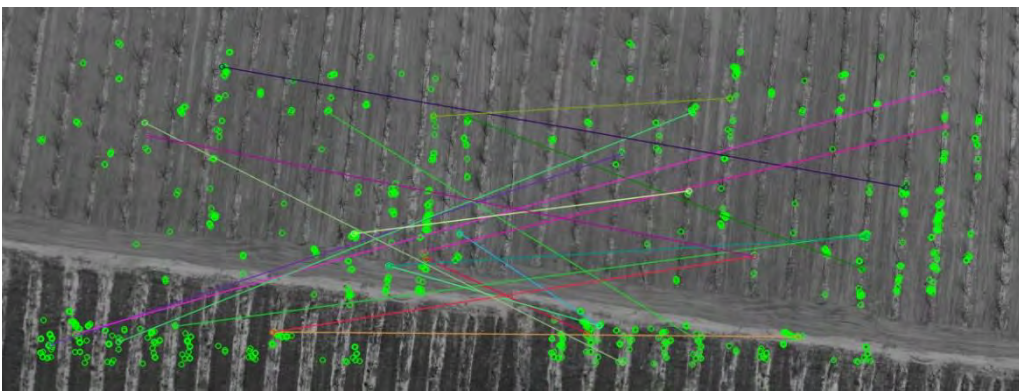


Figura 5.26 Ejemplo de aplicación de ORB-KNN en una imagen de ODM dataset.

- Combinación **SIFT-BF**: en este caso, SIFT detectó en la imagen izquierda 3,673 y en la derecha 2,785 puntos característicos; de los cuales se encontraron 1,566 correspondencias (lo que equivale al 42.63%), en un tiempo de 2.261 segundos en promedio. En la Figura 5.27 se observa el resultado visual y como en el caso anterior se observan un gran número de falsos positivos (21.5%).

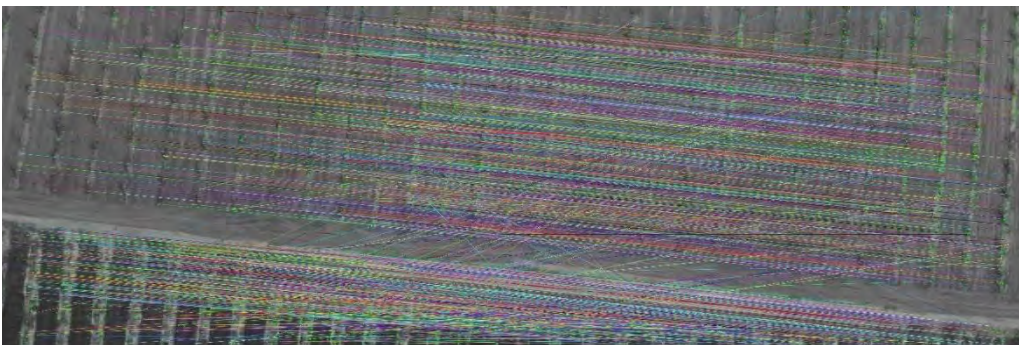


Figura 5.27 Ejemplo de aplicación de SIFT-BF en una imagen de ODM dataset.

- Combinación **SIFT-KNN**: en este caso, SIFT detectó en la imagen izquierda 3,673 y en la derecha 2,785 puntos característicos; de los cuales se encontraron 1,536 correspondencias (lo que equivale al 41.82%), en un tiempo de 2 segundos en promedio. En la Figura 5.28 se observa el resultado visual y como en el caso anterior se observan un gran número de falsos positivos (23%).

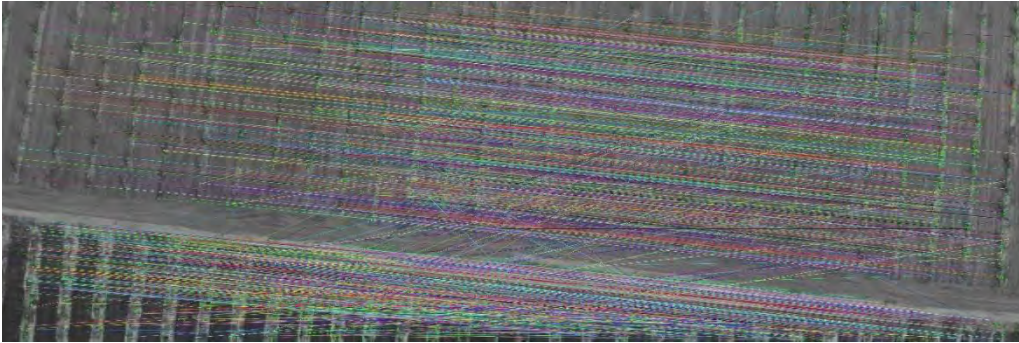


Figura 5.28 Ejemplo de aplicación de SIFT-KNN en una imagen de ODM dataset.

- Combinación **SIFT-KD-TREE**: SIFT detectó en la imagen izquierda 3,673 y en la derecha 2,785 puntos característicos; de los cuales se encontraron 953 correspondencias (lo que equivale al 25.94%), en un tiempo de 1.947 segundos en promedio. En la Figura 5.29 se observa el resultado visual, no obstante, al contrario de los casos anteriores, este caso en específico no tiene falsos positivos.

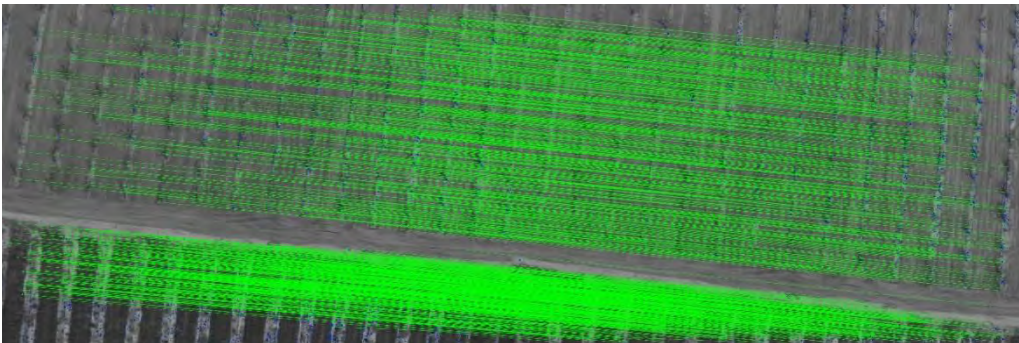


Figura 5.29 Ejemplo de aplicación de SIFT-KD-TREE en una imagen de ODM dataset.

- Combinación **SuperPoint-SuperGlue**: SuperPoint detectó en la imagen izquierda 690 y en la derecha 644 puntos característicos; de los cuales se encontraron 421 correspondencias (lo que equivale al 61.01%), en un tiempo de 0.062 segundos en promedio. En la Figura 5.30 se observa el resultado visual, no tiene falsos positivos.

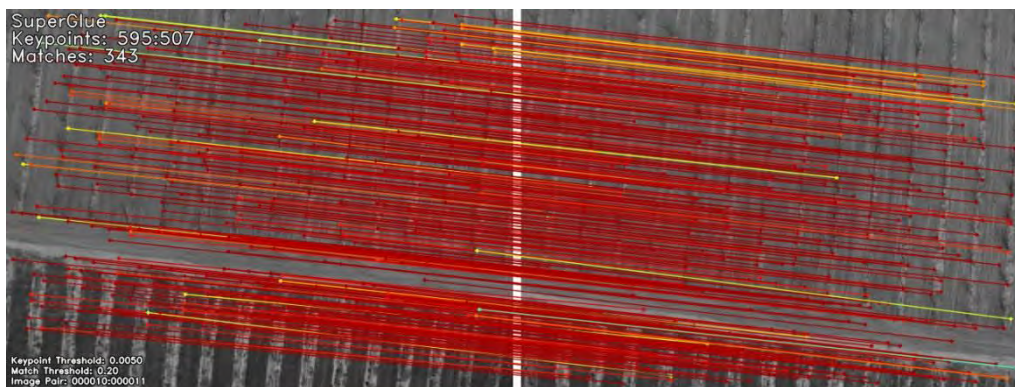


Figura 5.30 Ejemplo de aplicación de SuperPoint en una imagen de ODM dataset.

Resultados

En la Tabla 5.2 se muestran los resultados promedio obtenidos de las 49 imágenes para cada combinación de algoritmos antes mostrados, con todas las imágenes del dataset de ODM. En este caso:

- El detector que encuentra un mayor número de puntos característicos es SIFT; sin embargo, es bastante lento, le toma 600 milisegundos en promedio para cada imagen.
- La combinación de SuperPoint-SuperGlue obtiene un promedio de 66.91% de correspondencias. Además, lleva a cabo el proceso de detectar los puntos y la correspondencia de estos en un tiempo de 61 milisegundos lo cual lo hace bueno para aplicaciones en tiempo real, asimismo es estable.
- ORB tanto con BF como en combinación con KNN no tiene un buen desempeño. El detector ORB es el más rápido de todos los evaluados, y también el que reporta un menor número de puntos; pero no logra conseguir una buena estabilidad, detectando solamente un 6.36% de correspondencias, contando los falsos positivos.

Tabla 5.2 Resultados promedio de los algoritmos de detección y correspondencia con el dataset ODM.

Nombre	puntos detectados	correspondencias	porcentaje	tiempo
ORB-BF	495.36	121.96	24.64%	0.058 s
ORB-KNN	495.36	31.51	6.36%	0.056 s
SIFT-KNN	7262.87	3391.55	46.69%	2.000 s
SIFT-KD-TREE	7262.87	2227.45	30.20%	1.947 s
SIFT-BF	7262.87	2688.60	50.78%	2.261 s
SP-SG	602.73	407.83	66.91%	0.620 s

b) Banco de imágenes 4

Se realizó la evaluación de las combinaciones de las técnicas ya vistas pero ahora con el banco de imágenes 4. Es importante hacer notar que se realizaron 2 modificaciones.

- El primer cambio consistió en solo considerar los primeros 50 mil puntos que ORB encuentra.
- El segundo cambio consistió en realizar nuevamente el experimento con **SuperPoint-SuperGlue**, pero modificando las dimensiones de las imágenes, ya que no fue posible aplicar dichas técnicas con la resolución de origen debido a limitaciones de Hardware.

Entonces, se redimensionaron a una cuarta parte ($1,984 \times 1488$) y a 608×456 que son los valores de defecto de SuperPoint.

- Combinación **ORB-BF**: Se obtuvieron los siguientes resultados, se configuró ORB para que detectara 50 mil puntos como máximo. Sin embargo, que a pesar de detectar 10,584 correspondencias (lo que equivale al 21.168%), en un tiempo de 23.505 segundos en promedio, el algoritmo BF que no tiene una buena precisión (5% de correspondencias verdaderas), como se muestra en la Figura 5.31.

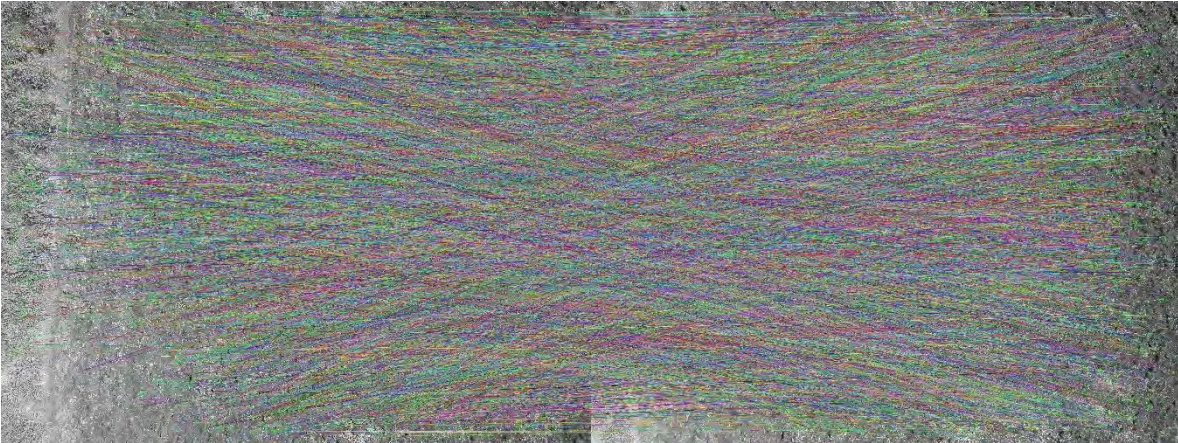


Figura 5.31 Ejemplo de aplicación de ORB-BF en una imagen del banco de imágenes 4.

- Combinación **ORB-KNN**: En este caso, se obtuvieron 50 mil puntos detectados por ORB; sin embargo, KNN no obtiene buenos resultados, al contrario, detecta un menor número de correspondencias (1,816 correspondencias, lo que equivale al 3.63%, en un tiempo de 15.783 segundos en promedio), como se puede observar en la Figura 5.32.

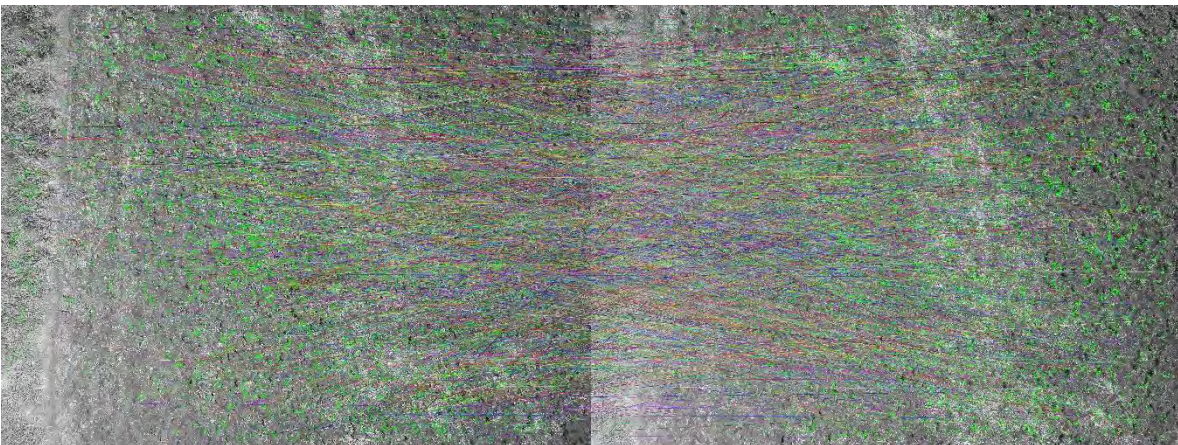


Figura 5.32 Ejemplo de aplicación de ORB-KNN en una imagen del banco de imágenes 4.

- Combinación **SIFT-BF**: SIFT detectó en la imagen izquierda 71,111 y en la derecha 74,417 puntos característicos; de los cuales se encontraron 30,708 correspondencias (lo que equivale al 41.26%), en un tiempo de 46.16 segundos en promedio. En la Figura 5.33 se observa el resultado visual, se observan un gran número de falsos positivos (21.5% de correspondencias verdaderas).



Figura 5.33 Ejemplo de aplicación de SIFT- BF en una imagen del banco de imágenes 4.

- Combinación **SIFT-KNN**: en este caso, SIFT detectó en la imagen izquierda 71,111 y en la derecha 74,417 puntos característicos; de los cuales se encontraron 21,425 correspondencias (lo que equivale al 28.79%), en un tiempo de 31.626 segundos en promedio. En la Figura 5.34 se observa el resultado visual y como en el caso anterior se observan un gran número de falsos positivos (12% de correspondencias verdaderas).

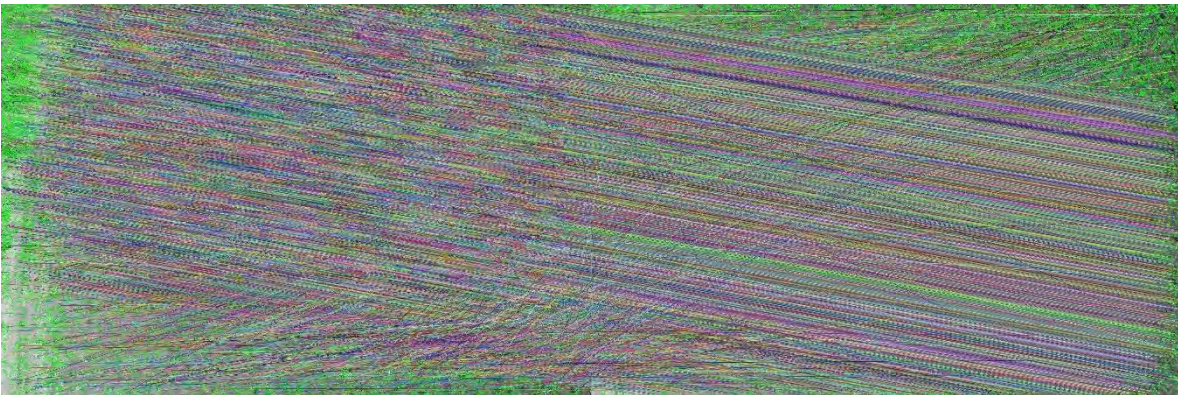


Figura 5.34 Ejemplo de aplicación de SIFT- KNN en una imagen del banco de imágenes 4.

- Combinación **SIFT-KD-TREE**: SIFT detectó en la imagen izquierda 71,111 y en la derecha 74,417 puntos característicos; de los cuales se encontraron 11,986 correspondencias (lo que equivale al 16.10%), en un tiempo de 10.3 segundos en promedio. En la Figura 5.35 se observa el resultado visual, el cual no tiene falsos positivos visibles.

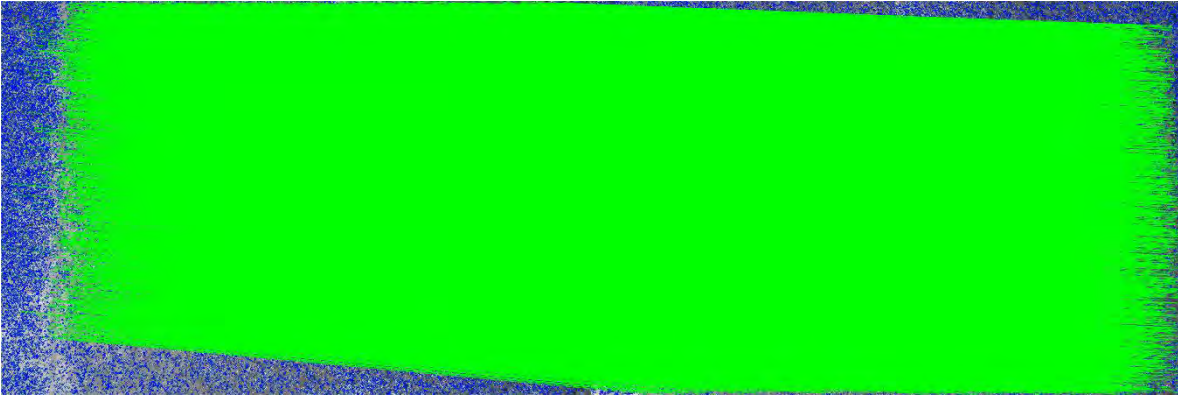


Figura 5.35 Ejemplo de aplicación de SIFT-KD-TREE en una imagen del banco de imágenes 4.

- Combinación **SuperPoint-SuperGlue 1/4 tamaño de la imagen**: SuperPoint detectó en la imagen izquierda 8,540 y en la derecha 8,559 puntos característicos; de los cuales se encontraron 7,121 correspondencias (lo que equivale al 83.19%), en un tiempo promedio de 3.011 segundos. En la Figura 5.36 se observa el resultado visual, el cual no tiene falsos positivos visibles.

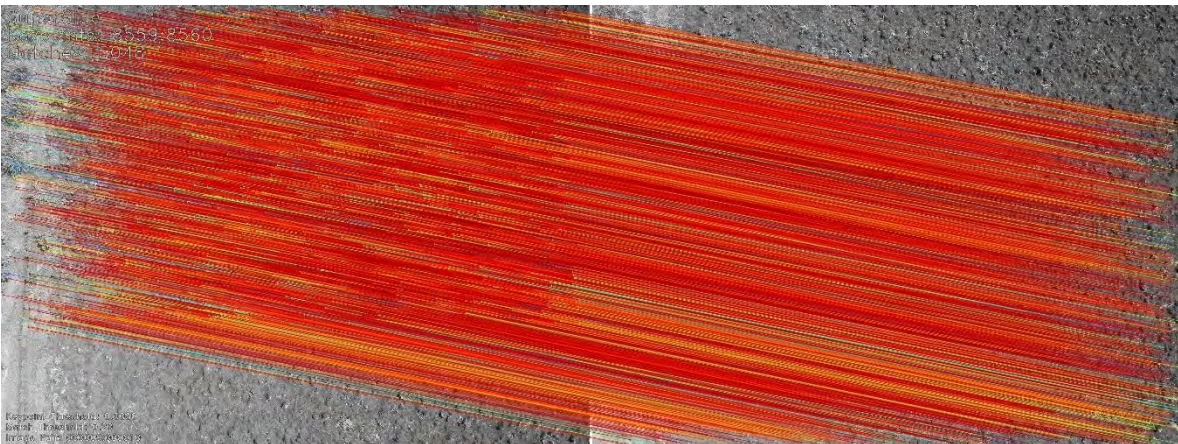


Figura 5.36 Ejemplo de aplicación de SuperPoint-SuperGlue 1/4 en una imagen del banco de imágenes 4.

- Combinación **SuperPoint-SuperGlue tamaño de imagen de 608 × 456**: SuperPoint detectó en la imagen izquierda 690 y en la derecha 644 puntos característicos; de los cuales se encontraron 421 correspondencias (lo que equivale al 61.01%), en un tiempo de 0.062 segundos en promedio. En la Figura 5.37 se observa el resultado visual, el cual no tiene falsos positivos.

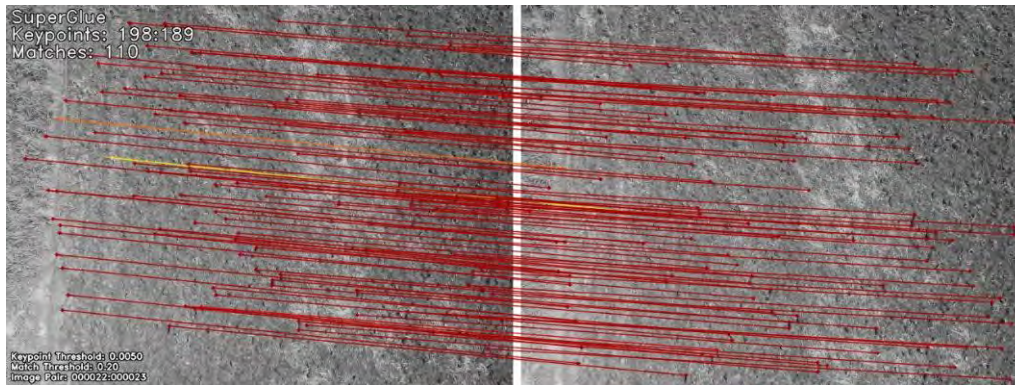


Figura 5.37 Ejemplo de aplicación de SuperPoint-SuperGlue en una imagen del banco de imágenes 4.

Resultados

Considerando las 96 imágenes del banco de imágenes 4, los resultados fueron semejantes a los obtenidos en el caso anterior.

- A pesar de que ORB obtiene un mayor número de puntos característicos, los resultados en la correspondencia con ambos algoritmos son malos al relacionar pocos de ellos.
- El detector SIFT es el detector que mayor número de puntos encuentra y en combinación con KD-TREE tienen una buena relación entre el tiempo de detección y el número de correspondencias realizadas.
- **SuperPoint-SuperGlue** esta combinación entrega el mejor porcentaje de correspondencias, pero el detector localiza una menor cantidad de puntos característicos en comparación con ORB o SIFT.

La Tabla 5.3 muestra el detalle de los puntos detectados, las correspondencias logradas y el tiempo para llevar a cabo ambos procesos.

Tabla 5.3 Resultados promedio de los algoritmos de detección y correspondencia con el banco de imágenes 4.

Nombre	puntos detectados	correspondencias	porcentaje	tiempo
ORB-BF	50000	11629.75	23.26%	23.5050 s
ORB-KNN	50000	1913.61	3.83%	15.7830 s
SIFT-KNN	75307.22	28945.32	38.45%	31.6260 s
SIFT-KD-TREE	75307.22	20667.38	26.88%	10.3040 s
SIFT-BF	75307.22	32512.80	47.30%	46.1660 s
SP-SG r	602.7292	407.83	66.91%	0.0620 s
SP-SG 1/4	8131.649	5194.02	62.76%	3.0110 s

De los resultados obtenidos, podemos sacar las siguientes conclusiones:

- SIFT es un detector que encuentra un alto número de puntos característicos, pero es lento en comparación con otros detectores.
- Aunque SIFT es eficaz en la detección de puntos, su velocidad puede limitar su utilidad en aplicaciones que requieren procesamiento en tiempo real.

- La combinación de SuperPoint-SuperGlue es destacable por su buen rendimiento en la correspondencia, con un promedio del 66.91% de correspondencias. Además, logra este rendimiento en un tiempo relativamente corto de 61 milisegundos por imagen, lo que lo hace adecuado para aplicaciones en tiempo real. Sin embargo, el número de puntos característicos detectados por SuperPoint es menor en comparación con ORB o SIFT.
- ORB, aunque es el detector más rápido evaluado y encuentra un mayor número de puntos característicos que SuperPoint, su desempeño en la correspondencia es pobre, con solo un 6.36% de correspondencias, incluyendo falsos positivos. Esto sugiere que ORB puede no ser adecuado para aplicaciones que requieren una correspondencia precisa de puntos.
- En resumen, cada detector tiene sus ventajas y desventajas. SIFT destaca en la detección de puntos, pero es lento, SuperPoint-SuperGlue ofrece una buena combinación de velocidad y rendimiento en correspondencias, mientras que ORB es rápido, pero tiene un rendimiento deficiente en la correspondencia. La elección del detector depende de las necesidades específicas de la aplicación, priorizando entre velocidad, precisión y cantidad de puntos detectados.

5.4 Caso 3 Co-correspondencias

Con el objetivo de reducir el error de SFM y de acuerdo con [159] se utiliza uno de los beneficios de las cámaras binoculares el cual consiste en que; se puede estimar el error en la medición tomando en consideración la distancia medida.

Por lo tanto, conociendo el error de medición a varias alturas, es favorable el realizar una doble búsqueda de puntos característicos y correspondencias entre 2 pares de imágenes binoculares, de las cuales en el primer par de imágenes se utiliza la imagen del lente izquierdo (como una cámara monocular) y las dos imágenes del segundo par, las cuales son adquiridas a una altura menor, para poder comparar las mediciones entre puntos.

La Figura 5.38 muestra las 3 imágenes utilizadas para este proceso para cada escenario, la imagen izquierda corresponde al lente izquierdo del primer par binocular, la imagen central y derecha corresponden al segundo par binocular para cada escenario, el primer escenario es el de Multi-biome, seguido por City Park y al último el escenario propio.

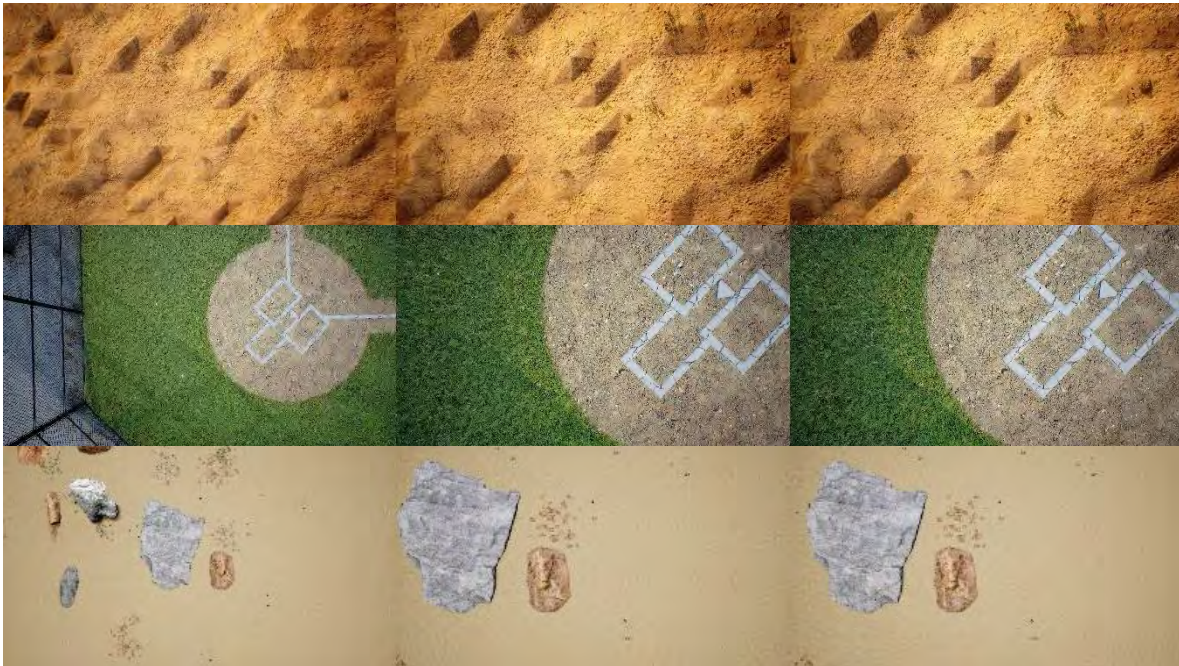


Figura 5.38 La imagen izquierda corresponde al lente izquierdo del primer par binocular, la imagen central y derecha corresponden al segundo par binocular para cada escenario, el primer renglón presenta al escenario de Multi-biome, el segundo renglón muestra a City Park y el último renglón contiene el escenario propio.

Entonces, el proceso que se sigue es el siguiente:

- Se detectan y describen los puntos característicos utilizando la transformada SIFT para cada una de las imágenes;
- Se realiza la primera búsqueda de correspondencia entre dos pares de imágenes utilizando el algoritmo de KD-TREE [67]. El primer par de imágenes este compuesto por la imagen izquierda y la imagen central de la Figura 5.38, el segundo par está compuesto por la imagen central con la imagen derecha de la Figura 5.38, tal y como se observa en la Figura 5.39.
- Finalmente se realiza una segunda búsqueda de correspondencias entre los dos vectores de correspondencias, utilizando la distancia Euclidiana junto con el resultado obtenido del par de imágenes: la imagen izquierda y la imagen derecha de la Figura 5.38, obteniendo los resultados mostrados en la Figura 5.40.

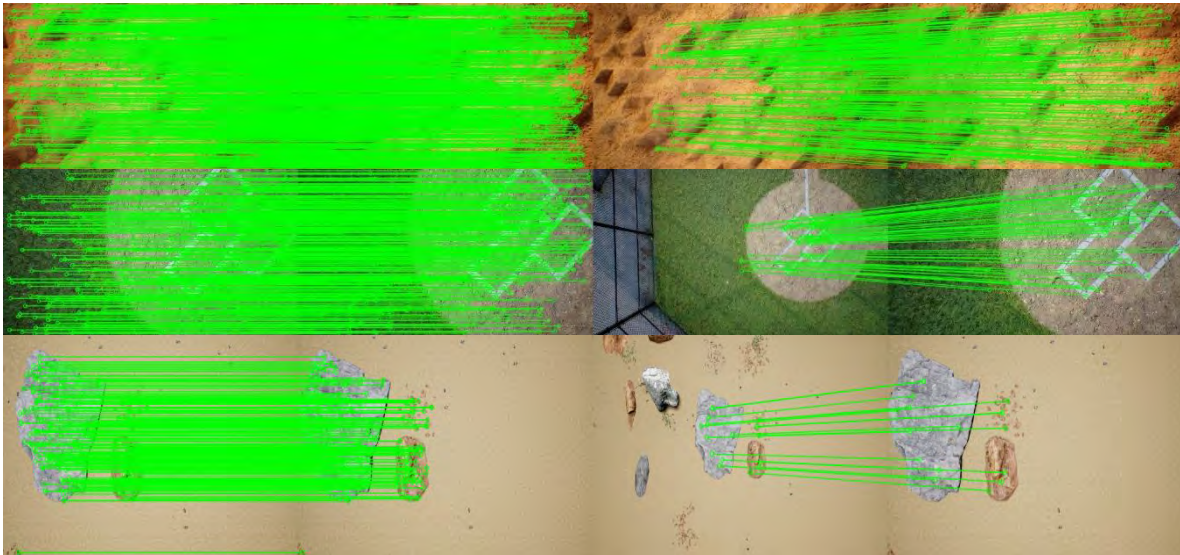


Figura 5.39 Correspondencias de dos pares de imágenes para cada escenario, el primer renglón presenta el escenario de Multi-biome, el segundo renglón muestra a City Park y el último renglón contiene el escenario propio simulado.

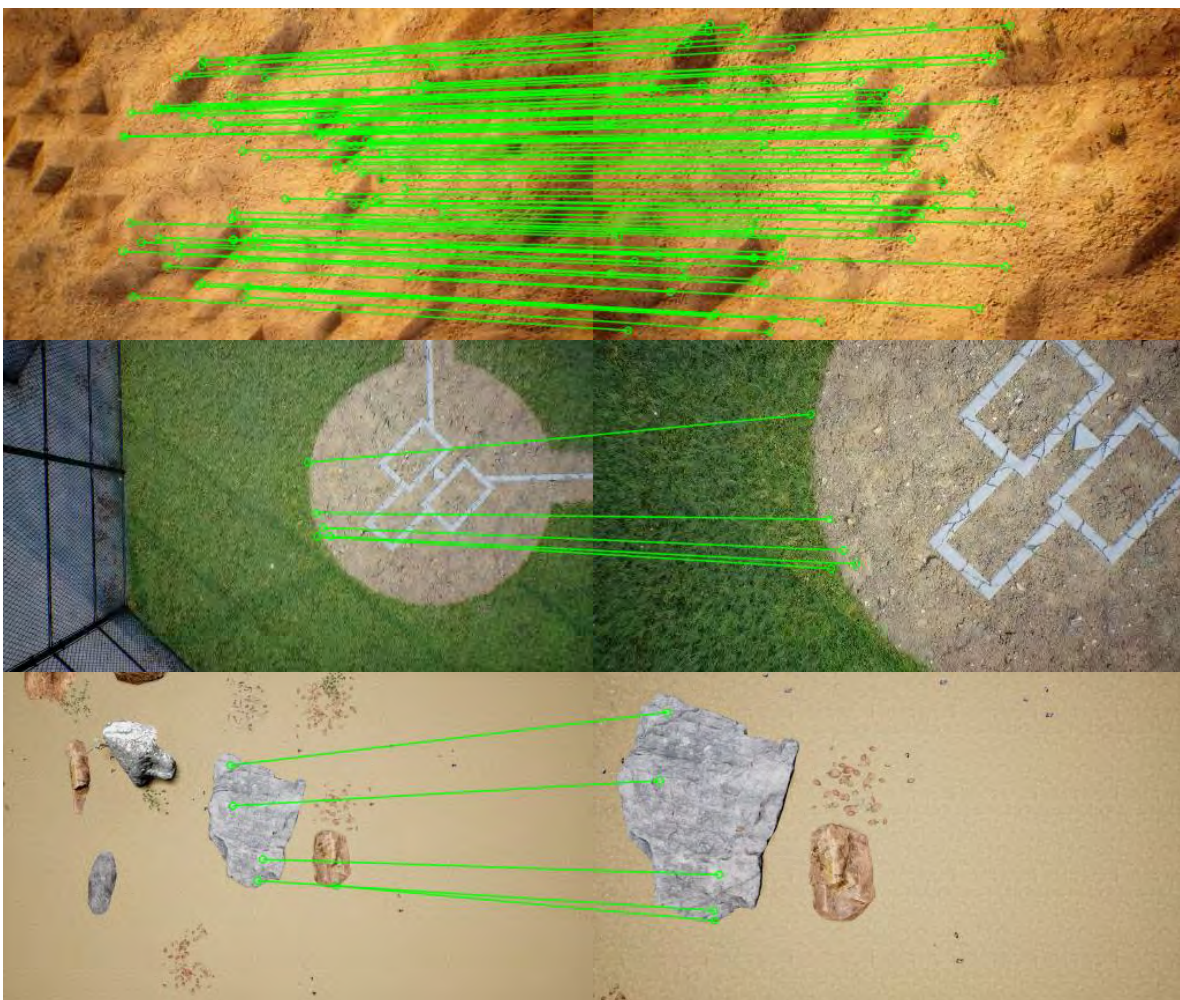


Figura 5.40 Correspondencias en común entre las 3 imágenes para cada escenario, primer escenario es el de Multi-biome, seguido por City Park y al último el escenario propio simulado todos del banco de imágenes simulado.

Esto permite el obtener la posición calculada por la cámara binocular de un punto x a una altura menor y utilizarla como un punto de control terrestre para mejorar la precisión de SFM, disminuyendo la necesidad de colocar físicamente puntos de control terrestres.

5.5 Caso 4: Generación de puntos de control terrestre virtuales con cartulinas

El objetivo de este caso fue evaluar la precisión obtenida al incorporar los puntos de control terrestre virtuales, para ello se realizaron dos casos de prueba. Se generó una reconstrucción para cada una de las tres alturas de los bancos de imágenes 1 y 2: a) considerando los puntos de control terrestres virtuales y, b) sin ellos. Para poder calcular una diferencia en porcentaje y facilitar la comprensión de los datos, se utilizó la Ecuación 5.1, donde un resultado positivo significa que la precisión de la reconstrucción aumentó con el uso de los puntos de control terrestre virtuales.

$$Diferencia = \left(\frac{original - modificado}{original} \right) * 100 \quad \text{Ecuación 5.1}$$

Donde:

Original= Valor de la métrica en la reconstrucción sin puntos de control terrestre virtuales.

Modificado= Valor de la métrica en la reconstrucción con puntos de control terrestre virtuales.

5.5.1 Lectura de las tablas de resultados

Una parte crucial en los próximos puntos es entender las tablas de resultados, estas tablas fueron generadas del informe de calidad proporcionado por OpenDroneMap, los datos son los siguientes:

- RMSE Absoluto: Se refiere al valor resultante de utilizar la Ecuación 3.21 sobre todos los ejes de la reconstrucción en formato global.
- RMSE Relativo Se refiere al valor resultante de utilizar la Ecuación 3.21 sobre todos los ejes de la reconstrucción en formato local.
- CE90: Se refiere al valor resultante de la métrica CE90 sobre los valores Relativos y Absolutos.
- LE90: Se refiere al valor resultante de la métrica LE90 sobre los valores Relativos y Absolutos.

5.5.2 Resultados con el banco de imágenes 1

Los resultados obtenidos se presentan en la Tabla 5.4, donde RMSE Absoluto se refiere al RMSE del GPS y RMSE relativo a la re-proyección local. La Tabla 5.4 presenta la información de los vuelos a sus diferentes alturas; donde S = sin puntos de control, C = con puntos de control y Dif= diferencia en porcentajes. Los resultados están dados en metros y el signo negativo significa que el utilizar puntos de control terrestre virtuales añade ruido a la reconstrucción resultante.

Los valores del RMSE relativo para el caso de 100 m pasa de tener 12.5 cm a 3.6 cm con el uso de puntos de control terrestres virtuales, lo que se traduce a una mejora del 71.277%. Con 75 m pasa de tener 3.9 cm a 3.6 cm lo que equivale al 6.89% de mejora, sin embargo,

a 50 m pasa de 2.8 a 3.1 cm lo que significa un aumento en el error del 9.5%. Concluyendo, evaluando la metodología con el banco de imágenes 1 se obtuvo que el utilizar puntos de control virtuales mejora los resultados en 2 de los 3 casos de prueba. las mejoras van desde el 16.5% hasta el 35.8%.

Tabla 5.4 Resultados de utilizar puntos de control terrestre en banco de imágenes 1, DONDE S significa sin puntos de control, C significa considerando puntos de control y la columna DIF presenta la diferencia entre ambos.

	100 m			75 m			50 m		
	S	C	Dif	S	C	Dif	S	C	Dif
RMSE Absoluto	0.637 m	0.637 m	0%	0.544 m	0.377 m	30.656%	0.462 m	0.619 m	-33.95%
RMSE Relativo	0.125 m	0.036 m	71.277%	0.039 m	0.036 m	6.897%	0.028 m	0.031 m	-9.524%
CE90	1.087 m	1.05 m	3.359%	0.963 m	0.642 m	33.385%	0.964 m	1.261 m	-30.8%
LE90	0.232 m	0.073 m	68.683%	0.061 m	0.064 m	-4.918%	0.044 m	0.057 m	-29.54%
Promedio			35.887%			16.505%			-25.95%

5.5.3 Resultados con el banco de imágenes 2

Como ya se mencionó, el segundo banco de imágenes se integra de 3 vuelos a diferentes alturas (100, 75 y 50 metros). Primero, sin utilizar ningún punto de control virtual (resultados en la primera columna de la Tabla 5.5) y posteriormente, considerando en la reconstrucción los puntos de control virtuales (resultados en la segunda columna de la Tabla 5.5).

Como se puede observar en el Tabla 5.5, los valores del RMSE relativo para el caso de 100 m pasa de tener 4.4 cm a 4.6 cm con el uso de puntos de control terrestres virtuales, lo que se traduce en un ligero aumento del error en 5.34%. A 75 m pasa de tener 5.3 cm a 3.6 cm lo que equivale al 34.44% de mejora y a 50 m pasa de 1.9 a 2.1 cm lo que significa un aumento en el error del 10.5%. El utilizar puntos de control virtuales mejora los resultados el RMSE absoluto en 3 de los vuelos, logrando en promedio mejoras que van desde el 0.56% hasta el 10.407%.

Tabla 5.5 Resultados de utilizar puntos de control terrestre en banco de imágenes 2.

	100 m			75 m			50 m		
	S	C	Dif	S	C	Dif	S	C	Dif
RMSE Absoluto	0.704 m	0.691 m	1.847%	0.558 m	0.485 m	13.09%	0.517 m	0.383 m	25.983%
RMSE Relativo	0.044 m	0.046 m	-5.34%	0.053 m	0.036 m	31.447%	0.019 m	0.021 m	-10.52%
CE90	1.222 m	1.111 m	9.124%	0.975 m	0.887 m	9.03%	0.98 m	0.644 m	34.252%
LE90	0.075 m	0.077 m	-3.35%	0.067 m	0.075 m	-11.94%	0.038 m	0.043 m	-13.15%
Promedio			0.568%			10.407%			9.138%

Los tiempos de procesamiento para las reconstrucciones, varían dependiendo del número de imágenes que contenga cada una, ya que cada una participa en procesos tanto individuales como grupales, tales como detectar puntos característicos hasta la reconstrucción densa. Los tiempos totales de la reconstrucción van desde los 30 minutos hasta 1 hora y media sin el uso de los puntos de control terrestres virtuales y con su uso, el tiempo de procesamiento sube desde los 50 minutos hasta las dos horas.

Sin embargo, las diferencias en el tiempo de procesamiento se ven compensadas considerando los tiempos de trabajo de campo para los puntos de control terrestre clásicos, los cuales necesitan de un proceso especial para cada uno de ellos. Es decir, cada punto se tiene que realizar con una estación total o en su defecto al no contar con el equipo

especial o experiencia, se necesitaría el apoyo de un topógrafo. La diferencia principal entre el trabajo de campo y utilizar los puntos de control virtuales radica en que, el poner objetos con distancias conocidas (en este caso las cartulinas) solo se colocan en el lugar planeado para un punto de control terrestre y ya se puede proceder a la captura de fotografías, no es necesario ningún proceso adicional.

5.5.4 Discusión

Después de la revisión de los resultados se puede concluir lo siguiente:

- Los experimentos realizados en las diferentes condiciones (altura, hora y uso de diferente número de puntos de control virtuales) muestran que el utilizar puntos de control virtuales puede ayudar a reducir el error medio cuadrático tanto local como global en la mayoría de los casos.
- El experimento en el cual no se logra una mejoría podría deberse a errores humanos tales como en el etiquetado de los puntos en los objetos de referencia en las imágenes y la distribución de las cartulinas en el cultivo. Los aspectos mencionados son manuales e introducen ruido a la metodología propuesta, por tal motivo se propone el generar tanto la detección de los puntos dentro de las imágenes como el cálculo de las distancias conocidas utilizando un dispositivo para medir distancias como puede ser un Lidar o una cámara binocular.
- Los resultados a las diferentes alturas señalan que se obtiene un menor RMSE conforme el dron vuela más bajo; sin embargo, esto también implica un mayor número de fotografías y tiempo de procesamiento, por lo tanto, la altura óptima para un trabajo está dada por la precisión requerida y el tiempo disponible.

Recapitulando el trabajo tiene como objetivo el reducir el RMSE calculado al realizar una reconstrucción 3D utilizando fotogrametría con drones, sin utilizar un GPS profesional en los puntos de control terrestres, manteniendo un RMSE relativo menor a 10 cm a 100 m de altura. Como puede observarse, el objetivo se cumple en casi todos los casos que se utilizan puntos de control terrestre virtuales. En la experimentación se puede observar la mejora al utilizarlos, sobre todo en el banco de imágenes 1 a 100 m, donde al utilizarlos pasa de tener un RMSE de 12.5 cm a 3.6 cm.

5.6 Caso 5: Preprocesamiento con COLMAP

El objetivo de este caso es evaluar si existen diferencias entre utilizar el preprocesamiento de las imágenes con COLMAP o no para mejorar la precisión de las reconstrucciones. Se generaron 12 reconstrucciones, de los bancos de imágenes 1 y 2 detallados en la sección 5.2: 6 experimentos sin el preprocesamiento y 6 con el preprocesamiento que consiste en generar una reconstrucción escasa con COLMAP para posteriormente, modificar los valores EXIF de las imágenes con las posiciones calculadas por COLMAP. Para estas pruebas no se generan puntos de control virtuales.

Como ya se mencionó, para cada altura (a 100, 75 y 50 m) de los bancos de imágenes 1 y 2 (que son 269 imágenes en total).

La Tabla 5.6 y la Tabla 5.7 presentan los resultados obtenidos de los experimentos mencionados en el banco de imágenes 1 y el banco de imágenes 2 respectivamente.

Tabla 5.6 Resultados de las reconstrucciones con preprocesamiento de COLMAP en el banco de imágenes 1.

	100 m			75 m			50 m		
	S	C	Dif	S	C	Dif	S	C	Dif
RMSE Absoluto	0.636 m	0.655 m	-2.984%	0.550 m	0.547 m	0.665%	0.463 m	0.466 m	-0.575%
RMSE Relativo	0.125 m	0.121 m	3.191%	0.0433 m	0.0277 m	36.153%	0.0253 m	0.022 m	13.157%
CE90	0.659 m	0.695 m	-5.53%	0.633 m	0.453 m	28.436%	0.345 m	0.356 m	-3.188%
LE90	0.659 m	0.665 m	-0.986%	0.578 m	0.567 m	1.901%	0.345 m	0.356 m	4.484%
Promedio			-1.579%			16.78%			3.469%

Tabla 5.7 Resultados de las reconstrucciones con preprocesamiento de COLMAP en el banco de imágenes 2.

	100 m			75 m			50 m		
	S	C	Dif	S	C	Dif	S	C	Dif
RMSE Absoluto	0.689 m	0.691 m	-0.241%	0.557667 m	0.566 m	-1.550%	0.514 m	0.530 m	-3.044%
RMSE Relativo	0.107 m	0.0416 m	61.059%	0.053 m	0.0523 m	1.252%	0.027 m	0.025 m	8.536%
CE90	0.709 m	0.826 m	-16.572%	0.3755 m	0.384 m	-2.265%	0.358 m	0.376 m	-5.020%
LE90	0.539 m	0.413 m	23.447%	0.666 m	0.6785 m	-1.876%	0.67 m	0.64 m	4.690%
Promedio			16.923%			-1.10%			1.296%

El utilizar COLMAP para un preprocesamiento de los datos, en la mayoría de los casos si generó un aumento en la precisión. Sin embargo, si se desea realizar una reconstrucción de manera rápida sin importar tanto la calidad no sería recomendada ya que el aumento es del 5.96% en promedio, por otro lado, si el tiempo de procesamiento no es una limitante, el uso del preprocesamiento con COLMAP es eficaz.

5.7 Caso 6 Estabilidad de los puntos de control ante vibraciones y vegetación

El objetivo de la prueba es verificar el funcionamiento del estabilizador mecánico con el banco de imágenes 3 que se integra de 3 vuelos: dos con el dron con cámara binocular (uno con la cámara Zed colocada directamente sobre el chasis y la otra sobre un estabilizador mecánico) y el tercero con el dron con cámara monocular; sobre una parcela que tiene caña de azúcar de aproximante 2 meses de edad.

Sin embargo, no fue posible realizar la correspondencia de puntos en las imágenes que tienen mayor presencia de vegetación debido a que el dron es grande y potente, ocasionando que las hojas se muevan, y el movimiento es mayor conforme el dron va descendiendo, incluso volando las cartulinas. Los resultados obtenidos se pueden observar en el Anexo A: Resultados de correspondencias en una parcela con caña en el cual se encuentran los resultados de correspondencias en la parcela con caña, la Figura A.1 muestra un resultado promedio, la Figura A.2 detalla uno de los mejores resultados de correspondencias entre monocular y binocular y, la Figura A.3 tiene un ejemplo de cómo el viento generado por el dron movía los objetos.

Continuando con la prueba para medir si el estabilizador es funcional en un dron de este tipo (con motores excesivamente grandes que podrían generar una especie de vibración que desestabilizara el estabilizador), se realizó una prueba tomando en cuenta la etapa de correspondencias entre imágenes, comparando cada imagen binocular con todas las monoculares.

Cada imagen tiene un re-escalado a una cuarta parte de su tamaño original para agilizar el proceso ya que la cámara binocular por vuelo tiene más de 2 mil imágenes, y reportando la imagen que tenga mayor número de correspondencias contra una imagen de referencia

monocular (proceso repetido para cada imagen monocular), esto para los dos vuelos, además de almacenar la cantidad de correspondencias encontradas por imágenes. En la Tabla 5.8 se pueden observar los resultados obtenidos de promediar los resultados para cada vuelo, con y sin el uso del estabilizador mecánico.

Tabla 5.8 Resultados de correspondencias con y sin estabilizador mecánico.

Correspondencias	Sin estabilizador	Con estabilizador
Promedio	43.26	45
Mínimo	7	18
Máximo	63	72

El uso del estabilizador mecánico no se ve afectado por las altas frecuencias del dron y es capaz de ayudar a estabilizar la cámara ante movimientos bruscos, esto se ve reflejado en los resultados de la Tabla 5.8.

5.8 Caso 7 Re-proyección de nube de puntos a imagen

La finalidad del caso es probar que es posible la generación de puntos de control virtuales a partir de las imágenes de la cámara binocular. Para ello, se realizaron a) 7 reconstrucciones utilizando el conjunto de imágenes 3 y b) 4 reconstrucciones con el conjunto de imágenes 5.

Las reconstrucciones para el conjunto de imágenes 3 son las siguientes:

- Reconstrucción con las imágenes puras.
- Reconstrucción con las imágenes de preprocesamiento de COLMAP.
- Reconstrucción con las imágenes puras utilizando los puntos de control virtuales.
- Reconstrucción con las imágenes de preprocesamiento de COLMAP, utilizando los puntos de control virtuales.
- Reconstrucción con las imágenes puras, pero con GCPs generados con el GPS de un celular.
- Reconstrucción con las imágenes puras, pero con GCPs generados con las cartulinas.
- Reconstrucción con las imágenes de preprocesamiento de COLMAP, pero con GCPs generados con las cartulinas.

En la Tabla 5.9 se pueden observar los resultados obtenidos con las experimentaciones antes mencionadas del conjunto de imágenes 3.

Las reconstrucciones para el conjunto de imágenes 5 son las siguientes:

- Reconstrucción con las imágenes a 50 m.
- Reconstrucción con las imágenes utilizando los puntos de control virtuales a 50 m.
- Reconstrucción con las imágenes a 100 m.
- Reconstrucción con las imágenes utilizando los puntos de control virtuales a 100 m.

En la Tabla 5.10 se pueden observar los resultados obtenidos con las experimentaciones del conjunto de imágenes 5.

De los experimentos anteriores se obtiene el resultado de que es posible utilizar la cámara binocular para poder generar puntos de control terrestre virtuales, los cuales tienen múltiples beneficios, entre los cuales destacan:

- No requieren que se pongan marcas sobre la tierra como típicamente se realiza, por lo tanto, conlleva un ahorro de tiempo, ya que cada marca para que sean de calidad (10-1 cm de error), se necesita que se deje el receptor GPS en un rango de 15 minutos a 48 horas por punto [101], esto dependiendo del clima, si esta despejado o nublado y de la precisión deseada.
- No requiere que, para cada GCP se le localice dentro de todas las imágenes del banco de imágenes manualmente. El proceso lo realiza de forma automática, entregando un archivo donde indica en que imágenes se encuentra el VGCP y sus coordenadas dentro de la misma.
- Evita poner en riesgo al usuario, en zonas de difícil acceso como una pendiente muy pronunciada o un risco, donde sería necesario colocar un punto de control terrestre típico, realizando todo el proceso que con el conlleva, el dron lo genera de forma automática, sin que el usuario tenga que ir a esas zonas.

Tabla 5.9 Resultados VGCP'S conjunto de imágenes 3.

Proceso	Original	Original-VGCP	COLMAP	COLMAP-VGCP	GCPs CEL	VGCP CAR ORIGINAL	VGCP CAR MODIFI
RMSE Absoluto	0.7226 m	0.5223 m	0.7450 m	0.7106 m	6.4741 m	0.6936 m	0.6066 m
RMSE Relativo	0.0966 m	0.0983 m	0.0966 m	0.0936 m	0.2953 m	0.0996 m	0.1060 m
CE90	0.7930 m	0.5905 m	0.7950 m	0.5720 m	2.7320 m	0.8195 m	0.7340 m
LE90	0.7930 m	0.5905 m	0.4390 m	0.2255 m	1.8095 m	0.4820 m	0.5015 m
Diferencia promedio		21.9204%	-0.5541%	20.4611%	-388.3620%	-2.7477%	0.2215%

Tabla 5.10 Resultados VGCP'S conjunto de imágenes 5.

Proceso	Sin VGCP	Con VGCP	Diferencia	Sin VGCP	Con VGCP	Diferencia
RMSE Absoluto	0.4123 m	0.4356 m	-5.6588%	0.4260 m	0.2686 m	36.9327 %
RMSE Relativo	0.0440 m	0.0393 m	10.6060 %	0.1036 m	0.0646 m	37.6205 %
CE90	0.4200 m	0.2105 m	49.8809 %	0.1990 m	0.0390 m	80.4020 %
LE90	0.4100 m	0.1350 m	67.0731 %	0.8870 m	0.0940 m	89.4024 %
Diferencia promedio			30.4753 %			61.0894 %

Se puede observar en la Tabla 5.9 en este caso el uso de los GCPs generados con las cartulinas ya sean con el GPS del celular o de manera virtual no fue el mejor y esto junto con los experimentos de la sección 5.5 llevan a la conclusión de que el buen funcionamiento de los puntos de control terrestres ya sean reales o virtuales esta dado por la correcta colocación en tierra depende de varios factores, como la forma de la parcela a reconstruir, así como sus proporciones, además de la cantidad de puntos utilizados.

Existen algunos estudios, que buscan encontrar el mejor patrón de colocación de los puntos [49], [83], [160], [161], [162], [163], no obstante, sería interesante realizar una experimentación utilizando redes neuronales, un sensor Lidar, fotografías de la zona a trabajar, además del modelo generado por Google Earth.

El objetivo se cumple ya que en todos los casos en donde se utilizan los VGCP generados por la cámara binocular se ve una reducción del error de 33.45% en promedio.

Capítulo 6: Conclusiones

6.1 Conclusiones

En este trabajo se presentó un método para generar puntos de control terrestre mediante el uso de una cámara binocular montada en un dron, el cual vuela a una altura de 10 m.

Se presentan cuatro formas para poder realizar la calibración para la cámara binocular Zed 2. El estudio comparativo de esas diferentes formas da como resultado que, para el fin deseado de la cámara, la mejor forma de calibrarla es mediante un patrón de calibración y una distancia mayor a las comunes (de 30 cm se pasó a 3 m), esto utilizando las librerías de OpenCV.

Se presenta una comparativa entre diferentes combinaciones de algoritmos para detectar, describir puntos característicos y algoritmos para realizar las correspondencias. De la experimentación realizada se puede concluir que dependiendo del objetivo que se tenga se debe seleccionar la combinación a utilizar. Por ejemplo, el uso de la combinación de SIFT-KD-TREE, es la que mayor número de puntos característicos detecta, la segunda en precisión y una que tiene los mayores tiempos de procesamiento. Estas características vuelven ideal la combinación en trabajos donde es necesaria una gran cantidad de puntos, con buena precisión, como, por ejemplo, en la tarea de generar la nube de puntos, ya que el mayor número de puntos es mejor, claro sin sacrificar la precisión.

Por otra parte, la combinación de SuperPoint con SuperGlue, es la que mejor precisión tiene para el problema aquí presentado. Sin embargo, no logra detectar una gran cantidad de puntos característicos, esto la vuelve ideal para trabajos donde la precisión es fundamental, como, por ejemplo, en la tarea de calcular las posiciones de las cámaras.

Se presenta una forma de realizar la correspondencia entre puntos característicos presentes en 3 imágenes: las imágenes de la cámara binocular y la imagen monocular, utilizando una doble búsqueda de los puntos, tomando como base sus vectores de correspondencias de KD-TREE.

Se presenta una forma de generar puntos de control terrestres virtuales, utilizando distancias conocidas. En la presente tesis se utilizaron cartulinas para obtener esas distancias dentro de la reconstrucción, dando como resultado una mejora del 7.7% en promedio.

Se presenta un método para generar puntos de control terrestres de forma virtual tomando en cuenta la información proveniente de un dron binocular junto con la información

generada por SFM de las imágenes monoculares de un dron comercial, esto con la intención de reducir el uso de los GCPs típicos. Se propuso una metodología de co-correspondencias entre dos nubes de puntos una binocular y la otra monoculares, las cuales contienen diferentes sistemas de medidas, utilizando puntos característicos que tienen en común. Adicionalmente se propuso un preprocesamiento con COLMAP para refinar las posiciones calculadas por el GPS.

Se muestra que el uso de COLMAP para preprocesamiento en la mayoría de los casos mejora la precisión hasta en un 16%. Sin embargo, en algunos casos tiene un efecto contrario, pero solamente de hasta un 1.5%, por lo que el utilizar COLMAP como preprocesamiento es factible.

Se muestra que el uso de co-correspondencias es viable para generar GCPs virtuales los cuales se pueden añadir a las reconstrucciones como GCPs típicos y estos ayudan a disminuir el RMSE, dando como resultado una mejora en la precisión en 21.92%.

Además, el uso de GPS virtuales apoyan a reducir tiempos de adquisición de datos en campo, ya que no es necesario colocar marcas en el suelo ni medirlas globalmente, así como el tiempo de preprocesamiento humano de las imágenes, ya que no se tienen que buscar los GPS en todas las imágenes y marcarlos uno por uno, esto lo realiza el sistema de forma automática.

En conclusión, en el presente trabajo se presenta una metodología para mejorar la precisión al momento de generar una reconstrucción fotogramétrica, desde el proceso de calibración de la cámara binocular hasta el punto de generar puntos de control terrestres virtuales, mediante una cámara binocular.

6.2 Objetivos logrados

En la Tabla 6.1 se muestra una breve descripción de cómo se cumplieron con los objetivos tanto el general como los específicos del proyecto de tesis.

Tabla 6.1 Solución por objetivo.

Objetivo general	Solución del objetivo
Proponer, aplicar y evaluar un algoritmo para la generación de modelos digitales de elevación, mediante el uso de una cámara binocular montada en un dron, con la finalidad de tener una precisión menor a los 10 cm de RMSE en el eje Z, a una altura aproximada de 10 m.	Se implementó una forma para generar VGCP, utilizando una cámara binocular montada en un dron que vuela a 10 m de altura, lo cual ayuda a reducir el error de la reconstrucción, sin importar a que altura vuele el dron monocular (puede volar desde los 7 m hasta los 100 m), en los vuelos menores a 100 m siempre obtiene un RMSE en el eje Z menor a los 10 cm.
Estudiar las técnicas de estructura a partir del movimiento, visión binocular y múltiples vistas estéreo más utilizadas.	Se estudiaron los sistemas más populares en el estado del arte, dando como resultado que OpenDroneMap y COLMAP, son los sistemas más completos que incluyen técnicas de estructura a partir del

	movimiento, visión binocular y múltiples vistas estéreo.
Objetivo general	Solución del objetivo
Seleccionar y aplicar varias de estas técnicas para conocer su rendimiento.	Se implementaron varias técnicas, dando como resultado que COLMAP tiene mayor precisión; sin embargo, OpenDroneMap tiene un mayor número de herramientas, además de permitir la incorporación de puntos de control terrestre.
Proponer y/o modificar un algoritmo para la generación de un modelo digital de elevación.	Se seleccionó a COLMAP junto con SuperPoint y SuperGlue para la etapa de detectar y calcular las posiciones de las cámaras, añadiendo la información resultante a la reconstrucción de OpenDroneMap.
Combinar la información de la una cámara binocular (montada en un dron) con la proporcionada por sensores GPS, RTK y el uso de técnicas SFM para generar modelos digitales de elevación.	Se le incorpora a la reconstrucción de OpenDroneMap (SFM) puntos de control terrestres virtuales, generados por la cámara binocular, además de los datos del GPS. No se utilizó un sensor RTK, el sensor aumenta mucho el precio del dispositivo.
Evaluar el sistema propuesto.	Se evaluó el sistema propuesto en diferentes ambientes, así como en diferentes alturas, de los cuales en 11 de los 12 casos presentados existe una disminución del RMSE.
Publicar un artículo índice JCR.	Se publicó un artículo titulado "Generation of Virtual Ground Control Points Using a Binocular Camera" en Drones 2024, Volume 8, Issue 5, 195.

6.3 Aportaciones

Las aportaciones obtenidas con esta investigación son:

- Bancos de imágenes:
Se presentan 6 diferentes bancos de imágenes, los cuales van desde entornos reales hasta simulados. Lo destacable es que se utiliza un dron fumigador modificado con una cámara Zed 2 para poder recopilar información extra proveniente de esa cámara, además proponemos que idealmente sería generar un dron binocular, con una cámara monocular con gimbal y la otra cámara la cual estaría ubicada en la parte trasera del dron pero siempre viendo a nadir, esto con la finalidad de utilizar un solo dron, el cual pueda realizar la ruta de vuelo completa, tanto la monocular que va entre 50 m a 100 m, hasta la binocular que descendería entre 5 m a 10 m, para más información revisar la sección 5.1.
- Se implementó un sistema para detección y co-correspondencias de puntos característicos entre una cámara binocular y una monocular:

Se propone un algoritmo que busca puntos característicos que se encuentren presentes tanto en una imagen monocular, como en la binocular, tomando en cuenta las dos nubes de puntos, esto con la finalidad de encontrar puntos de referencia en la nube binocular que se puedan incorporar en la reconstrucción monocular y de esta forma se reduce el error relativo.

- Se implementó un sistema para generar puntos de control virtuales utilizando distancias conocidas:

Se propone el uso de una cartulina o de una cámara binocular para introducir información de comprobación a la reconstrucción monocular, utilizando el sistema del punto anterior.

- Preprocesamiento con COLMAP:

Se propone el uso de COLMAP para rectificar las coordenadas GPS de las imágenes, el uso de COLMAP para el preprocesamiento de datos generalmente aumenta la precisión. Sin embargo, si la velocidad de reconstrucción es prioridad sobre la calidad no es recomendable, ya que el aumento medio es sólo del 5,96%. Por el contrario, si el tiempo de procesamiento no es una preocupación, el preprocesamiento con COLMAP resulta eficaz.

6.4 Trabajo futuro

Como trabajo futuro se contempla el siguiente listado:

- Realizar una comparativa contra el uso de sensores RTK y Lidar.
- Entrenar una red neuronal para predecir, las mejores ubicaciones para generar los puntos de control terrestres virtuales.
- Probar la super resolución binocular (se intentó, pero es necesario una tarjeta gráfica de mayor poder computacional).
- Realizar pruebas en entornos de mayor tamaño, como pudiera ser un bosque o un cañón.

6.5 Productos

- Poster en “Escuela de Inteligencia Computacional y Robótica 2023” realizada del 10 al 14 de octubre en la UTEZ (producto del octavo semestre).
- Poster en “2022 IEEE School on Computational Intelligence and Robotics” el 16 de agosto en la UTEZ (producto del quinto semestre).
- Publicación de un artículo titulado “Comparison of detection and matching Techniques for aerial photogrammetry” en el congreso “2021 IEEE International Autumn meeting on Power, Electronics and Computing (ROPEC 2021) con índice SCOPUS en Ixtapa, México” (producto del cuarto semestre).
- Publicación de un artículo titulado “Generation of Virtual Ground Control Points Using a Binocular Camera” en Drones 2024, Volume 8, Issue 5, 195.

Referencias

- [1] E. Guisado-Pintado, D. W. T. Jackson, and D. Rogers, "3D mapping efficacy of a drone and terrestrial laser scanner over a temperate beach-dune zone," *Geomorphology*, vol. 328, pp. 157–172, Mar. 2019, doi: 10.1016/j.geomorph.2018.12.013.
- [2] M. R. James, S. Robson, and M. W. Smith, "3-D uncertainty-based topographic change detection with structure-from-motion photogrammetry: precision maps for ground control and directly georeferenced surveys," *Earth Surf Process Landf*, vol. 42, no. 12, pp. 1769–1788, Sep. 2017, doi: 10.1002/esp.4125.
- [3] P. Burdziakowski, "EVALUATION OF OPEN DRONE MAP TOOLKIT FOR GEODETIC GRADE AERIAL DRONE MAPPING □ CASE STUDY.," in *International Multidisciplinary Scientific GeoConference Surveying Geology and Mining Ecology Management, SGEM*, Jun. 2017, pp. 101–110. doi: 10.5593/sgem2017/23/S10.013.
- [4] D. Wierzbicki and M. Nienaltowski, "Accuracy Analysis of a 3D Model of Excavation, Created from Images Acquired with an Action Camera from Low Altitudes," *ISPRS Int J Geoinf*, vol. 8, no. 2, p. 83, Feb. 2019, doi: 10.3390/ijgi8020083.
- [5] Teng Cao, Z.-Y. Xiang, X.-J. Gong, and J.-L. Liu, "Terrain reconstruction of lunar surface based on binocular fisheye camera," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, IEEE, Jun. 2014, pp. 2463–2468. doi: 10.1109/WCICA.2014.7053109.
- [6] C. Iheaturu, C. Okolie, E. Ayodele, A. Egogo-Stanley, S. Musa, and C. Ifejika Speranza, "A simplified structure-from-motion photogrammetry approach for urban development analysis," *Remote Sens Appl*, vol. 28, p. 100850, Nov. 2022, doi: 10.1016/j.rsase.2022.100850.
- [7] S. A. González Segura, "Visión binocular para robot móvil en exteriores simulados," CENIDET, 2011.
- [8] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*, First Edit. O'Reilly Media, Inc., 2017. doi: 10.5555/2523356.
- [9] K. Konolige, "Small Vision Systems: Hardware and Implementation," in *Robotics Research*, London: Springer London, 1998, pp. 203–212. doi: 10.1007/978-1-4471-1580-9_19.
- [10] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Trans Pattern Anal Mach Intell*, vol. 30, no. 2, pp. 328–341, Feb. 2008, doi: 10.1109/TPAMI.2007.1166.

- [11] O. Abhijit and D. Justin, "OpenVis3D, librería para correspondencia estéreo densa de código abierto en C++." Accessed: Dec. 29, 2020. [Online]. Available: <https://code.google.com/archive/p/openvis3d/>
- [12] A. S. Ogale and Y. Aloimonos, "Shape and the Stereo Correspondence Problem," *Int J Comput Vis*, vol. 65, no. 3, pp. 147–162, Dec. 2005, doi: 10.1007/s11263-005-3672-3.
- [13] C. L. Gómez Olán, "Fusión de Información de Pose en Robots Móviles con Visión," CENIDET, 2013.
- [14] A. Vergara Bahena, "Navegación, Localización y Mapeo de Robots Móviles para Trayectorias Pre-especificadas por Imágenes," CENIDET, 2015.
- [15] INEGI, "Metadatos de: Modelo Digital de Elevación de México," México, Mar. 2015.
- [16] L. Berdasco Menendez, "Cómo hacer un levantamiento topográfico con estación total paso a paso," <https://www.certicalia.com/blog/levantamiento-topografico-estacion-total-paso-a-paso>.
- [17] E. Guisado-Pintado, D. W. T. Jackson, and D. Rogers, "3D mapping efficacy of a drone and terrestrial laser scanner over a temperate beach-dune zone," *Geomorphology*, vol. 328, pp. 157–172, 2019, doi: 10.1016/j.geomorph.2018.12.013.
- [18] E. Alegre, G. Pajares, and A. De la Escalera, *Conceptos y Métodos en Visión por Computador*, vol. 1. 2016.
- [19] OpenDroneMap, "OpenDroneMap." Accessed: Jun. 13, 2020. [Online]. Available: <https://www.opendronemap.org/>
- [20] COLMAP, "PyColmap," github. Accessed: Jun. 09, 2022. [Online]. Available: <https://github.com/colmap/pycolmap>
- [21] ALICEVISION, "ALICEVISION." Accessed: Jun. 13, 2021. [Online]. Available: <https://alicevision.org/>
- [22] S. Cai, A. Obukhov, D. Dai, and L. Van Gool, "Pix2NeRF: Unsupervised Conditional π -GAN for Single Image to Neural Radiance Fields Translation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2022, pp. 3971–3980. doi: 10.1109/CVPR52688.2022.00395.
- [23] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans Graph*, vol. 41, no. 4, pp. 1–15, Jul. 2022, doi: 10.1145/3528223.3530127.
- [24] A. Boulch and R. Marlet, "POCO: Point Convolution for Surface Reconstruction," Jan. 2022, [Online]. Available: <http://arxiv.org/abs/2201.01831>
- [25] Z. Li *et al.*, "Neuralangelo: High-Fidelity Neural Surface Reconstruction," Jun. 2023, [Online]. Available: <http://arxiv.org/abs/2306.03092>
- [26] F. Darmon, B. Bascle, J.-C. Devaux, P. Monasse, and M. Aubry, "Improving neural implicit surfaces geometry with patch warping," Dec. 2021, [Online]. Available: <http://arxiv.org/abs/2112.09648>
- [27] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction," Jun. 2021, [Online]. Available: <http://arxiv.org/abs/2106.10689>

- [28] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "SfM-Net: Learning of Structure and Motion from Video," Apr. 2017, [Online]. Available: <http://arxiv.org/abs/1704.07804>
- [29] C. Pinard, L. Chevalley, A. Manzanera, and D. Filliat, "Learning Structure-from-Motion from Motion," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11131 LNCS, 2019, pp. 363–376. doi: 10.1007/978-3-030-11015-4_27.
- [30] M. Klodt and A. Vedaldi, "Supervising the New with the Old: Learning SFM from SFM," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11214 LNCS, 2018, pp. 713–728. doi: 10.1007/978-3-030-01249-6_43.
- [31] E. Cledat, L. V. Jospin, D. A. Cucci, and J. Skaloud, "Mapping quality prediction for RTK/PPK-equipped micro-drones operating in complex natural environment," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 24–38, Sep. 2020, doi: 10.1016/j.isprsjprs.2020.05.015.
- [32] K. Gao *et al.*, "Local Feature Performance Evaluation for Structure-From-Motion and Multi-View Stereo Using Simulated City-Scale Aerial Imagery," *IEEE Sens J*, vol. 21, no. 10, pp. 11615–11627, May 2021, doi: 10.1109/JSEN.2020.3042810.
- [33] J. Tomaščík, M. Mokroš, P. Surový, A. Grznárová, and J. Merganič, "UAV RTK/PPK Method—An Optimal Solution for Mapping Inaccessible Forested Areas?," *Remote Sens (Basel)*, vol. 11, no. 6, p. 721, Mar. 2019, doi: 10.3390/rs11060721.
- [34] S. Jiang, C. Jiang, and W. Jiang, "Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 230–251, Sep. 2020, doi: 10.1016/j.isprsjprs.2020.04.016.
- [35] J. Meza, A. G. Marrugo, E. Sierra, M. Guerrero, J. Meneses, and L. A. Romero, "A Structure-from-Motion Pipeline for Topographic Reconstructions Using Unmanned Aerial Vehicles and Open Source Software," in *Communications in Computer and Information Science*, vol. 885, Springer Verlag, 2018, pp. 213–225. doi: 10.1007/978-3-319-98998-3_17.
- [36] M. Bossoukpe *et al.*, "Low-cost drones help measure tree characteristics in the Sahelian savanna," *J Arid Environ*, vol. 187, p. 104449, Apr. 2021, doi: 10.1016/j.jaridenv.2021.104449.
- [37] S. Chand and B. Bollard, "Low altitude spatial assessment and monitoring of intertidal seagrass meadows beyond the visible spectrum using a remotely piloted aircraft system," *Estuar Coast Shelf Sci*, vol. 255, p. 107299, Jul. 2021, doi: 10.1016/j.ecss.2021.107299.
- [38] J. Vautherin *et al.*, "PHOTOGRAMMETRIC ACCURACY AND MODELING OF ROLLING SHUTTER CAMERAS," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III–3, pp. 139–146, Jun. 2016, doi: 10.5194/isprannals-III-3-139-2016.
- [39] I. Elkharchy, "Accuracy Assessment of Low-Cost Unmanned Aerial Vehicle (UAV) Photogrammetry," *Alexandria Engineering Journal*, vol. 60, no. 6, pp. 5579–5590, Dec. 2021, doi: 10.1016/j.aej.2021.04.011.

- [40] M. T. Page *et al.*, “Evaluating Mesquite Distribution Using Unpiloted Aerial Vehicles and Satellite Imagery,” *Rangel Ecol Manag*, vol. 83, pp. 91–101, Jul. 2022, doi: 10.1016/j.rama.2022.03.007.
- [41] A. R. Bankert, E. H. Strasser, C. G. Burch, and M. D. Correll, “An open-source approach to characterizing Chihuahuan Desert vegetation communities using object-based image analysis,” *J Arid Environ*, vol. 188, p. 104383, May 2021, doi: 10.1016/j.jaridenv.2020.104383.
- [42] R. Greco, E. Barca, P. Raunonen, M. Persia, and P. Tartarino, “Methodology for measuring dendrometric parameters in a mediterranean forest with UAVs flying inside forest,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 122, p. 103426, Aug. 2023, doi: 10.1016/j.jag.2023.103426.
- [43] S. Adar *et al.*, “Testing a novel pasture quality index using remote sensing tools in semiarid and Mediterranean grasslands,” *Agric Ecosyst Environ*, vol. 357, p. 108674, Nov. 2023, doi: 10.1016/j.agee.2023.108674.
- [44] L. Techapinyawat, I. Goulden-Brady, H. Garcia, and H. Zhang, “Aerial characterization of surface depressions in urban watersheds,” *J Hydrol (Amst)*, vol. 625, p. 129954, Oct. 2023, doi: 10.1016/j.jhydrol.2023.129954.
- [45] C. C. de S. Mello, D. H. C. Salim, and G. F. Simões, “UAV-based landfill operation monitoring: A year of volume and topographic measurements,” *Waste Management*, vol. 137, pp. 253–263, Jan. 2022, doi: 10.1016/j.wasman.2021.11.020.
- [46] S. Varbla, A. Ellmann, and R. Puust, “Centimetre-range deformations of built environment revealed by drone-based photogrammetry,” *Autom Constr*, vol. 128, p. 103787, Aug. 2021, doi: 10.1016/j.autcon.2021.103787.
- [47] M. Specht, B. Szostak, O. Lewicka, A. Stateczny, and C. Specht, “Method for determining of shallow water depths based on data recorded by UAV/USV vehicles and processed using the SVR algorithm,” *Measurement*, vol. 221, p. 113437, Nov. 2023, doi: 10.1016/j.measurement.2023.113437.
- [48] B. Awasthi, S. Karki, P. Regmi, D. S. Dhimi, S. Thapa, and U. S. Panday, “Analyzing the Effect of Distribution Pattern and Number of GCPs on Overall Accuracy of UAV Photogrammetric Results,” in *Lecture Notes in Civil Engineering*, vol. 51, Springer, 2020, pp. 339–354. doi: 10.1007/978-3-030-37393-1_29.
- [49] X. Liu, X. Lian, W. Yang, F. Wang, Y. Han, and Y. Zhang, “Accuracy Assessment of a UAV Direct Georeferencing Method and Impact of the Configuration of Ground Control Points,” *Drones*, vol. 6, no. 2, p. 30, Jan. 2022, doi: 10.3390/drones6020030.
- [50] T. Rosten Edward and Drummond, “Machine Learning for High-Speed Corner Detection,” in *Computer Vision – ECCV 2006*, H. and P. A. Leonardis Aleš and Bischof, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.
- [51] C. Harris and M. Stephens, “A Combined Corner and Edge Detector,” in *Proceedings of the Alvey Vision Conference 1988*, Alvey Vision Club, 1988, pp. 23.1-23.6. doi: 10.5244/C.2.23.
- [52] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571, 2011, doi: 10.1109/ICCV.2011.6126544.

- [53] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE, 1999, pp. 1150–1157 vol.2. doi: 10.1109/ICCV.1999.790410.
- [54] V. and S. C. and F. P. Calonder Michael and Lepetit, "BRIEF: Binary Robust Independent Elementary Features," in *Computer Vision – ECCV 2010*, P. and P. N. Daniilidis Kostas and Maragos, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792.
- [55] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Lecture Notes in Computer Science*, vol. 3951 LNCS, pp. 404–417, 2006, doi: 10.1007/11744023_32.
- [56] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, 2011, pp. 2548–2555. doi: 10.1109/ICCV.2011.6126542.
- [57] A. and D. A. J. Alcantarilla Pablo Fernández and Bartoli, "KAZE Features," in *Computer Vision – ECCV 2012*, S. and P. P. and S. Y. and S. C. Fitzgibbon Andrew and Lazebnik, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 214–227.
- [58] A. Lenc Karel and Vedaldi, "Learning Covariant Feature Detectors," in *Computer Vision – ECCV 2016 Workshops*, H. Hua Gang and Jégou, Ed., Cham: Springer International Publishing, 2016, pp. 100–117.
- [59] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. 2018-June, pp. 337–33712, Dec. 2017, [Online]. Available: <http://arxiv.org/abs/1712.07629>
- [60] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "TILDE: A Temporally Invariant Learned DETector," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5279–5288. doi: 10.1109/CVPR.2015.7299165.
- [61] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9910 LNCS, pp. 467–483, Mar. 2016, doi: 10.1007/978-3-319-46466-4_28.
- [62] Y. Ono, P. Fua, E. Trulls, and K. M. Yi, "LF-Net: Learning local features from images," *Adv Neural Inf Process Syst*, vol. 2018-Decem, no. Nips, pp. 6234–6244, 2018.
- [63] T. Cieslewski, K. G. Derpanis, and D. Scaramuzza, "SIPs: Succinct Interest Points from Unsupervised Inlieressness Probability Learning," in *2019 International Conference on 3D Vision (3DV)*, IEEE, Sep. 2019, pp. 604–613. doi: 10.1109/3DV.2019.00072.
- [64] P. G. T. Dias, A. A. Kassim, and V. Srinivasan, "A neural network based corner detection method," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, IEEE, pp. 2116–2120. doi: 10.1109/ICNN.1995.489004.
- [65] K. Jałowiecki, M. M. Rams, and B. Gardas, "Brute-forcing spin-glass problems with CUDA," *Comput Phys Commun*, vol. 260, p. 107728, Apr. 2019, doi: 10.1016/j.cpc.2020.107728.
- [66] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN Model-Based Approach in Classification," in *Lecture Notes in Computer Science (including subseries Lecture*

- Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*), vol. 2888, 2003, pp. 986–996. doi: 10.1007/978-3-540-39964-3_62.
- [67] L. Qiang, H. Hao, W. Yongmin, L. Xu, G. Liqian, and H. Hao, “The KD-Tree-based nearest-neighbor search algorithm in GRID interpolation,” in *2012 International Conference on Image Analysis and Signal Processing*, IEEE, Nov. 2012, pp. 1–6. doi: 10.1109/IASP.2012.6425061.
- [68] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperGlue: Learning Feature Matching With Graph Neural Networks,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2020, pp. 4937–4946. doi: 10.1109/CVPR42600.2020.00499.
- [69] OpenDroneMap, “Open Drone Map.” Accessed: Dec. 30, 2020. [Online]. Available: <https://www.opendronemap.org/>
- [70] M. Adorjan, “OpenSfM A collaborative Structure-from-Motion System,” 2015.
- [71] Pix4D, “Pix4D.” Accessed: Dec. 11, 2020. [Online]. Available: <https://www.pix4d.com>
- [72] Agisoft, “Metashape.” Accessed: Jun. 14, 2020. [Online]. Available: <https://www.agisoft.com/>
- [73] RealityCapture, “RealityCapture.” Accessed: Dec. 11, 2020. [Online]. Available: <https://www.capturingreality.com>
- [74] E. Guisado-Pintado, D. W. T. Jackson, and D. Rogers, “3D mapping efficacy of a drone and terrestrial laser scanner over a temperate beach-dune zone,” *Geomorphology*, vol. 328, pp. 157–172, 2019, doi: 10.1016/j.geomorph.2018.12.013.
- [75] D. Day, W. Weaver, and L. Wilsing, “Accuracy of uas photogrammetry: A comparative evaluation,” *Photogramm Eng Remote Sensing*, vol. 82, no. 12, pp. 909–914, 2016, doi: 10.14358/PERS.82.12.909.
- [76] J. W. Gross and B. W. Heumann, “A Statistical Examination of Image Stitching Software Packages For Use With Unmanned Aerial Systems,” *Photogramm Eng Remote Sensing*, vol. 82, no. 6, pp. 419–425, Jun. 2016, doi: 10.14358/PERS.82.6.419.
- [77] A. Alfonso-Torreño, Á. Gómez-Gutiérrez, S. Schnabel, J. F. Lavado Contador, J. J. de San José Blasco, and M. Sánchez Fernández, “sUAS, SfM-MVS photogrammetry and a topographic algorithm method to quantify the volume of sediments retained in check-dams,” *Science of The Total Environment*, vol. 678, pp. 369–382, Aug. 2019, doi: 10.1016/j.scitotenv.2019.04.332.
- [78] M. Chakraborty, L. R. Khot, and R. T. Peters, “Assessing suitability of modified center pivot irrigation systems in corn production using low altitude aerial imaging techniques,” *Information Processing in Agriculture*, vol. 7, no. 1, pp. 41–49, Mar. 2020, doi: 10.1016/j.inpa.2019.06.001.
- [79] J. J. Ruiz, L. Diaz-Mas, F. Perez, and A. Viguria, “EVALUATING THE ACCURACY OF DEM GENERATION ALGORITHMS FROM UAV IMAGERY,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-1/W2, no. September, pp. 333–337, Aug. 2013, doi: 10.5194/isprsarchives-XL-1-W2-333-2013.
- [80] O. Küng *et al.*, “THE ACCURACY OF AUTOMATIC PHOTOGRAMMETRIC TECHNIQUES ON ULTRA-LIGHT UAV IMAGERY,” *ISPRS - International Archives*

- of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-1/, no. 3, pp. 125–130, Sep. 2012, doi: 10.5194/isprsarchives-XXXVIII-1-C22-125-2011.
- [81] A. Eltner, P. Baumgart, H.-G. Maas, and D. Faust, “Multi-temporal UAV data for automatic measurement of rill and interrill erosion on loess soil,” *Earth Surf Process Landf*, vol. 40, no. 6, pp. 741–755, May 2015, doi: 10.1002/esp.3673.
- [82] M. Llena, D. Vericat, and J. A. Martínez Casasnovas, “Aplicación de algoritmos Structure from Motion (SfM) para el análisis histórico de cambios en la geomorfología fluvial,” *Cuaternario y Geomorfología*, vol. 32, no. 1–2, p. 53, 2018, doi: 10.17735/cyg.v32i1-2.60410.
- [83] M. R. James, S. Robson, S. D’Oleire-Oltmanns, and U. Niethammer, “Optimising UAV topographic surveys processed with structure-from-motion: Ground control quality, quantity and bundle adjustment,” *Geomorphology*, vol. 280, pp. 51–66, Mar. 2017, doi: 10.1016/j.geomorph.2016.11.021.
- [84] M. R. James, S. Robson, and M. W. Smith, “3-D uncertainty-based topographic change detection with structure-from-motion photogrammetry: precision maps for ground control and directly georeferenced surveys,” *Earth Surf Process Landf*, vol. 42, no. 12, pp. 1769–1788, Sep. 2017, doi: 10.1002/esp.4125.
- [85] A. S. Woodget, P. E. Carbonneau, F. Visser, and I. P. Maddock, “Quantifying submerged fluvial topography using hyperspatial resolution UAS imagery and structure from motion photogrammetry,” *Earth Surf Process Landf*, vol. 40, no. 1, pp. 47–64, Jan. 2015, doi: 10.1002/esp.3613.
- [86] E. Ferreira, J. Chandler, R. Wackrow, and K. Shiono, “Automated extraction of free surface topography using SfM-MVS photogrammetry,” *Flow Measurement and Instrumentation*, vol. 54, no. April 2016, pp. 243–249, Apr. 2017, doi: 10.1016/j.flowmeasinst.2017.02.001.
- [87] Pix4D, “Pix4D.” Accessed: Jun. 14, 2020. [Online]. Available: <https://www.pix4d.com/es>
- [88] Dronedeploy, “dronedeploy.” Accessed: Jun. 13, 2020. [Online]. Available: <https://www.dronedeploy.com/>
- [89] 3dflow, “3DF ZEPHYR.” Accessed: Jun. 14, 2020. [Online]. Available: <https://www.3dflow.net/>
- [90] J. Engel, J. Stuckler, and D. Cremers, “Large-scale direct SLAM with stereo cameras,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Sep. 2015, pp. 1935–1942. doi: 10.1109/IROS.2015.7353631.
- [91] A. Chandrashekar, J. Papadakis, A. Willis, and J. Gantert, “Structure-From-Motion and RGBD Depth Fusion,” in *SoutheastCon 2018*, IEEE, Apr. 2018, pp. 1–8. doi: 10.1109/SECON.2018.8478927.
- [92] J. B. Fernández Roblero, “Comparación de Algoritmos de Extracción y Asociación de Rasgos Para Visión Robótica,” Cenidet, 2012.
- [93] Samdrone, “Quantix Mapper.” Accessed: Jun. 13, 2021. [Online]. Available: <https://community.opendronemap.org/t/multispectral-data-for-4-acreage-orchard-in-quantix-mapper/6840>

- [94] K. Peñaloza, "Navegación Visual en Trayectorias Cerradas," CENIDET, 2013.
- [95] S. B. Miguel, "Puntos destacados sobre bordes en imágenes para visión robótica," CENIDET, 2015.
- [96] GeeksforGeeks, "K Dimensional Tree." Accessed: Jun. 29, 2021. [Online]. Available: <https://www.geeksforgeeks.org/k-dimensional-tree/>
- [97] P. A. Ullrich and C. M. Zarzycki, "TempestExtremes: a framework for scale-insensitive pointwise feature tracking on unstructured grids," *Geosci Model Dev*, vol. 10, no. 3, pp. 1069–1090, Mar. 2017, doi: 10.5194/gmd-10-1069-2017.
- [98] M. Barbarella, M. Fiani, and C. Zollo, "Assessment of DEM derived from very high-resolution stereo satellite imagery for geomorphometric analysis," *Eur J Remote Sens*, vol. 50, no. 1, pp. 534–549, Jan. 2017, doi: 10.1080/22797254.2017.1372084.
- [99] C. Strecha *et al.*, "QUALITY ASSESSMENT OF 3D RECONSTRUCTION USING FISHEYE AND PERSPECTIVE SENSORS," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3/W4, pp. 215–222, Mar. 2015, doi: 10.5194/isprsannals-II-3-W4-215-2015.
- [100] K. Liu, C. Song, L. Ke, L. Jiang, and R. Ma, "Automatic watershed delineation in the Tibetan endorheic basin: A lake-oriented approach based on digital elevation models," *Geomorphology*, vol. 358, p. 107127, Jun. 2020, doi: 10.1016/j.geomorph.2020.107127.
- [101] D. T. Gillins, D. Kerr, and B. Weaver, "Evaluation of the Online Positioning User Service for Processing Static GPS Surveys: OPUS-Projects, OPUS-S, OPUS-Net, and OPUS-RS," *Journal of Surveying Engineering*, vol. 145, no. 3, p. 05019002, Aug. 2019, doi: 10.1061/(ASCE)SU.1943-5428.0000280.
- [102] G. Muradás Odriozola, K. Pauly, S. Oswald, and D. Raymaekers, "Automating Ground Control Point Detection in Drone Imagery: From Computer Vision to Deep Learning," *Remote Sens (Basel)*, vol. 16, no. 5, p. 794, Feb. 2024, doi: 10.3390/rs16050794.
- [103] W. Li, G. Liu, L. Zhu, X. Li, Y. Zhang, and S. Shan, "Efficient detection and recognition algorithm of reference points in photogrammetry," in *Optics, Photonics and Digital Technologies for Imaging Applications IV*, P. Schelkens, T. Ebrahimi, G. Cristóbal, F. Truchetet, and P. Saarikko, Eds., SPIE, Apr. 2016, p. 989612. doi: 10.1117/12.2225416.
- [104] A. Jain, M. Mahajan, and R. Saraf, "Standardization of the Shape of Ground Control Point (GCP) and the Methodology for Its Detection in Images for UAV-Based Mapping Applications," in *Advances in Intelligent Systems and Computing*, vol. 943, Springer Verlag, 2020, pp. 459–476. doi: 10.1007/978-3-030-17795-9_34.
- [105] Ts. Purevdorj and R. Yokoyama, "An approach to automatic detection of GCP for AVHRR imagery.," *Journal of the Japan society of photogrammetry and remote sensing*, vol. 41, no. 1, pp. 28–38, 2002, doi: 10.4287/jsprs.41.28.
- [106] S. Montazeri, C. Gisinger, M. Eineder, and X. xiang Zhu, "Automatic Detection and Positioning of Ground Control Points Using TerraSAR-X Multiaspect Acquisitions," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 5, pp. 2613–2632, May 2018, doi: 10.1109/TGRS.2017.2769078.

- [107] S. Montazeri *et al.*, “SAR ground control point identification with the aid of high resolution optical data,” in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, Jul. 2016, pp. 3205–3208. doi: 10.1109/IGARSS.2016.7729829.
- [108] Z. Zhu, T. Bao, Y. Hu, and J. Gong, “A Novel Method for Fast Positioning of Non-Standardized Ground Control Points in Drone Images,” *Remote Sens (Basel)*, vol. 13, no. 15, p. 2849, Jul. 2021, doi: 10.3390/rs13152849.
- [109] D. O. Nitti *et al.*, “Automatic GCP extraction with high resolution COSMO-SkyMed products,” in *SAR Image Analysis, Modeling, and Techniques XVI*, C. Notarnicola, S. Paloscia, N. Pierdicca, and E. Mitchard, Eds., SPIE, Oct. 2016, p. 1000302. doi: 10.1117/12.2241281.
- [110] MathWorks, “Evaluating the Accuracy of Single Camera Calibration.” Accessed: Jun. 29, 2021. [Online]. Available: <https://www.mathworks.com/help/vision/ug/evaluating-the-accuracy-of-single-camera-calibration.html>
- [111] Gao Yi, Luo Jianxin, Qiu Hangping, and Wu Bo, “Survey of structure from motion,” in *Proceedings of 2014 International Conference on Cloud Computing and Internet of Things*, IEEE, Dec. 2014, pp. 72–76. doi: 10.1109/CCIOT.2014.7062508.
- [112] S. Agarwal *et al.*, “Building rome in a day,” *Commun ACM*, vol. 54, no. 10, pp. 105–112, 2011, doi: 10.1145/2001269.2001293.
- [113] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, “Towards Internet-scale multi-view stereo,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2010, pp. 1434–1441. doi: 10.1109/CVPR.2010.5539802.
- [114] P.-H. Lee, J.-W. Huang, and H.-Y. Lin, “3D model reconstruction based on multiple view image capture,” in *2012 International Symposium on Intelligent Signal Processing and Communications Systems*, IEEE, Nov. 2012, pp. 58–63. doi: 10.1109/ISPACS.2012.6473453.
- [115] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, “A Survey of Structure from Motion,” *Acta Numerica*, vol. 26, pp. 305–364, Jan. 2017, doi: 10.1017/S096249291700006X.
- [116] J. M. Lopez vallez, A. Fernandes Caballero, and M. A. Fernández, “Conceptos y Técnicas de Estereovisión por Computador,” *REVISTA IBEROAMERICANA DE INTELIGENCIA ARTIFICIAL*, vol. 27, no. Inteligencia Artificial, pp. 35–62, 2005.
- [117] F. Valdivielso Miranda, “Reconocimiento y registro 3D de objetos conocidos en una escena,” 2010.
- [118] J. A. Cancelas, R. C. González, I. Álvarez, and J. M. Enguita, *Procesamiento Morfológico, Visión 3D: Estereoscopia, Álgebra lineal básica para visión por computador, Geometría Proyectiva para Visión 3D*, vol. 1. 2016.
- [119] EPICGAMES, “UnrealEngine.” Accessed: Jun. 19, 2022. [Online]. Available: <https://www.unrealengine.com/en-US/?sessionInvalidated=true>
- [120] Microsoft Research, “AirSim,” <https://microsoft.github.io/AirSim/>.
- [121] J. L. Schonberger and J.-M. Frahm, “Structure-from-Motion Revisited,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 4104–4113. doi: 10.1109/CVPR.2016.445.

- [122] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise View Selection for Unstructured Multi-View Stereo," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9907 LNCS, 2016, pp. 501–518. doi: 10.1007/978-3-319-46487-9_31.
- [123] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson Surface Reconstruction," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, in SGP '06. Goslar, DEU: Eurographics Association, 2006, pp. 61–70.
- [124] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys, "Pixel-Perfect Structure-from-Motion with Featuremetric Refinement," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2021, pp. 5967–5977. doi: 10.1109/ICCV48922.2021.00593.
- [125] J. Degol, J. Y. Lee, R. Kataria, D. Yuan, T. Bretl, and D. Hoiem, "FEATS: Synthetic Feature Tracks for Structure from Motion Evaluation," in *2018 International Conference on 3D Vision (3DV)*, IEEE, Sep. 2018, pp. 352–361. doi: 10.1109/3DV.2018.00048.
- [126] V. G. D. M. Lyra *et al.*, "Development of an Efficient 3D Reconstruction Solution from Permissive Open-Source Code," in *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, IEEE, Nov. 2020, pp. 232–241. doi: 10.1109/SVR51698.2020.00044.
- [127] R. Maxence, H. Uchiyama, H. Kawasaki, D. Thomas, V. Nozick, and H. Saito, "Mobile Photometric Stereo with Keypoint-Based SLAM for Dense 3D Reconstruction," in *2019 International Conference on 3D Vision (3DV)*, IEEE, Sep. 2019, pp. 574–582. doi: 10.1109/3DV.2019.00069.
- [128] A. Chirivella González, "nreal engine, todo lo que necesitas saber," profesionalreview. Accessed: Jun. 06, 2022. [Online]. Available: <https://www.profesionalreview.com/2022/04/17/unreal-engine-todo-lo-que-necesitas-saber/>
- [129] A. de la Escalera, J. María Armingol, J. Luis Pech, and J. Julián Gómez, "Detección Automática de un Patrón para la Calibración de Cámaras," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 7, no. 4, pp. 83–94, Oct. 2010, doi: 10.1016/S1697-7912(10)70063-7.
- [130] ROS, "How to Calibrate a Stereo Camera." Accessed: Jan. 12, 2021. [Online]. Available: http://wiki.ros.org/camera_calibration/Tutorials/StereoCalibration
- [131] AMAZON and DIGITALGLOBE, "ACCURACY OF WORLDVIEW PRODUCTS," https://dg-cms-uploads-production.s3.amazonaws.com/uploads/document/file/38/DG_ACCURACY_WP_V3.pdf.
- [132] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson Surface Reconstruction," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, in SGP '06. Goslar, DEU: Eurographics Association, 2006, pp. 61–70.
- [133] European Petroleum Survey Group, "EPSG:4326 WGS 84," <https://epsg.io/4326>. Accessed: Aug. 29, 2023. [Online]. Available: <https://epsg.io/4326>
- [134] EPSG, "EPSG:32614 WGS 84 / UTM zone 14N," <https://epsg.io/32614>.
- [135] PROJ, "pyproj," <https://proj.org/>.

- [136] P. Harvey, “Libimage-exiftool-perl,” <https://github.com/pandastream/libimage-exiftool-perl-9.27>.
- [137] Docker, “Docker,” <https://www.docker.com/>.
- [138] amazon, “Docker,” <https://aws.amazon.com/es/docker/>.
- [139] A. Kaehler and G. Bradski, *OpenCV 3*. O’Reilly Media, 2017.
- [140] crehana, “OpenCV,” <https://www.crehana.com/blog/transformacion-digital/que-es-opencv/>.
- [141] Pytorch, “Pytorch 1.13,” <https://pytorch.org/get-started/locally/>.
- [142] itelligent, “PyTorch,” <https://itelligent.es/es/conoces-pytorch-herramienta-open-source-la-puedes-crear-redes-neuronales/>.
- [143] mappinggis, “OpenDroneMap,” <https://mappinggis.com/2019/08/opendronemap-crea-mapas-con-imagenes-capturadas-por-drones/>.
- [144] Ubuntu, “Ubuntu.” Accessed: Jan. 18, 2021. [Online]. Available: <https://ubuntu.com/download/desktop>
- [145] godaddy, “Ubuntu,” <https://es.godaddy.com/blog/que-es-ubuntu-y-para-que-sirve/>.
- [146] Ros, “Ros Noetic.” Accessed: Jan. 19, 2021. [Online]. Available: <http://wiki.ros.org/noetic/Installation>
- [147] ros, “ros,” <https://www.ros.org/>.
- [148] Nvidia, “Cuda.” Accessed: Jan. 19, 2021. [Online]. Available: <https://developer.nvidia.com/cuda-11.1.0-download-archive>
- [149] NVIDIA, “What is cuda,” <https://blogs.nvidia.com/blog/2012/09/10/what-is-cuda-2/>.
- [150] ZED, “ZED SDK.” Accessed: Jan. 19, 2021. [Online]. Available: <https://www.stereolabs.com/developers/release/3.2/>
- [151] DJI and AMAZON, “DJI MAVIC PRO PLATINUM,” <https://www.amazon.com/-/es/CP-PT-00000071-01-DJI-Mavic-Pro-Platinum/dp/B07568RYPH>.
- [152] OmegaDrone, “DJI Agras T10,” https://omegadrone.com.mx/producto/dji-t10_2/.
- [153] Stereolabs, “Zed 2.” Accessed: Dec. 29, 2020. [Online]. Available: <https://www.stereolabs.com/zed-2/>
- [154] NVIDIA, “Jetson Nano,” <https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/jetson-nano/>.
- [155] Skycatch, “What Are Control Points (GCPs And CHPs) And How Do I Lay Them?” Accessed: Jul. 01, 2021. [Online]. Available: <https://support.skycatch.com/hc/en-us/articles/219924107-What-are-Control-Points-GCPs-and-CHPs-and-How-do-I-Lay-Them->
- [156] J. Garth, “BRUSHIFY.” Accessed: Jun. 14, 2022. [Online]. Available: <https://www.brushify.io/>
- [157] SilverTM, “City Park Environment Collection.”
- [158] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans Pattern Anal Mach Intell*, vol. 22, no. 11, pp. 1330–1334, 2000, doi: 10.1109/34.888718.

- [159] L. E. Ortiz, V. E. Cabrera, and L. M. G. Goncalves, "Depth Data Error Modeling of the ZED 3D Vision Sensor from Stereolabs," *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, vol. 17, no. 1, p. 1, Jun. 2018, doi: 10.5565/rev/elcvia.1084.
- [160] 3DSurvey, "How to set up Ground Control Points (GCP)," 3dsurvey. Accessed: Jan. 12, 2020. [Online]. Available: https://www.3dsurvey.si/upload/files/001_GCP_Setup_v002.pdf
- [161] K. Zhang, H. Okazawa, K. Hayashi, T. Hayashi, L. Fiwa, and S. Maskey, "Optimization of Ground Control Point Distribution for Unmanned Aerial Vehicle Photogrammetry for Inaccessible Fields," *Sustainability*, vol. 14, no. 15, p. 9505, Aug. 2022, doi: 10.3390/su14159505.
- [162] P. Martínez-Carricondo, F. Agüera-Vega, F. Carvajal-Ramírez, F.-J. Mesas-Carrascosa, A. García-Ferrer, and F.-J. Pérez-Porras, "Assessment of UAV-photogrammetric mapping accuracy based on variation of ground control points," *International Journal of Applied Earth Observation and Geoinformation*, vol. 72, no. February, pp. 1–10, Oct. 2018, doi: 10.1016/j.jag.2018.05.015.
- [163] A. Ulvi, "The effect of the distribution and numbers of ground control points on the precision of producing orthophoto maps with an unmanned aerial vehicle," *Journal of Asian Architecture and Building Engineering*, vol. 20, no. 6, pp. 806–817, Nov. 2021, doi: 10.1080/13467581.2021.1973479.

Anexo A: Resultados de correspondencias en una parcela con caña

Resultados de búsqueda de correspondencias entre dos imágenes de diferentes cámaras, utilizando SIFT como detector y descriptor de puntos característicos y KD-TREE como algoritmo de correspondencia.



Figura A.1 Resultado promedio de correspondencia binocular monocular



Figura A.2 Uno de los mejores resultados de correspondencia binocular monocular



Figura A.3 Ejemplo de la fuerza del viento generado por el DJI T10

Anexo B: Resultados de re-proyección

En la Figura B.1, la Figura B.2 y en la Figura B.2 se pueden observar los puntos de la nube de puntos re-proyectados en cada imagen en el banco de imágenes 1, estos puntos tienen la misma ubicación espacial entre imágenes (en el terreno). En las zonas de solapamiento se observan las mismas características, y es interesante observar cómo pasan de una imagen a otra. Las zonas de solapamiento permiten al sistema detectar automáticamente características comunes (puntos clave) mediante el análisis del sistema, que selecciona las características óptimas. En cambio, los métodos manuales requieren que el usuario proponga los puntos de interés en las imágenes.

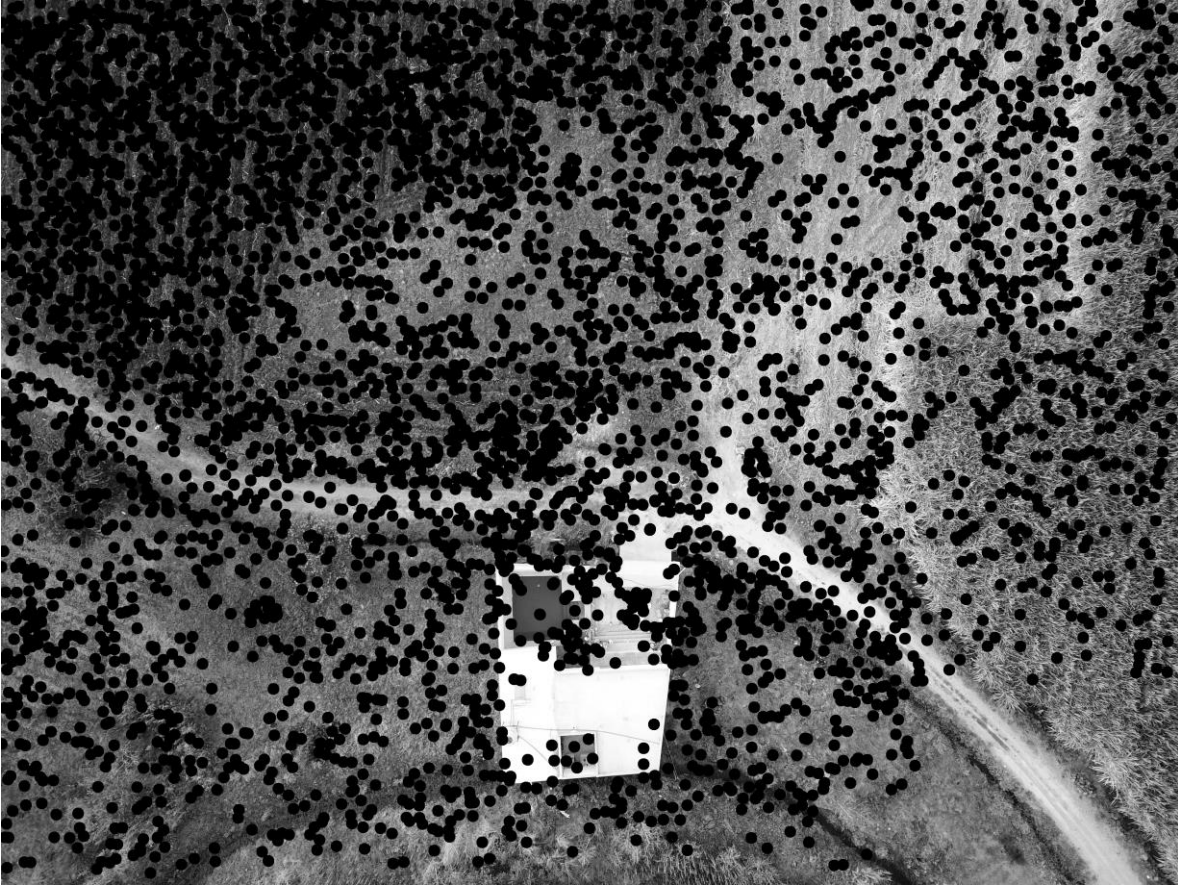


Figura B.1 Ejemplo de re-proyección 1.

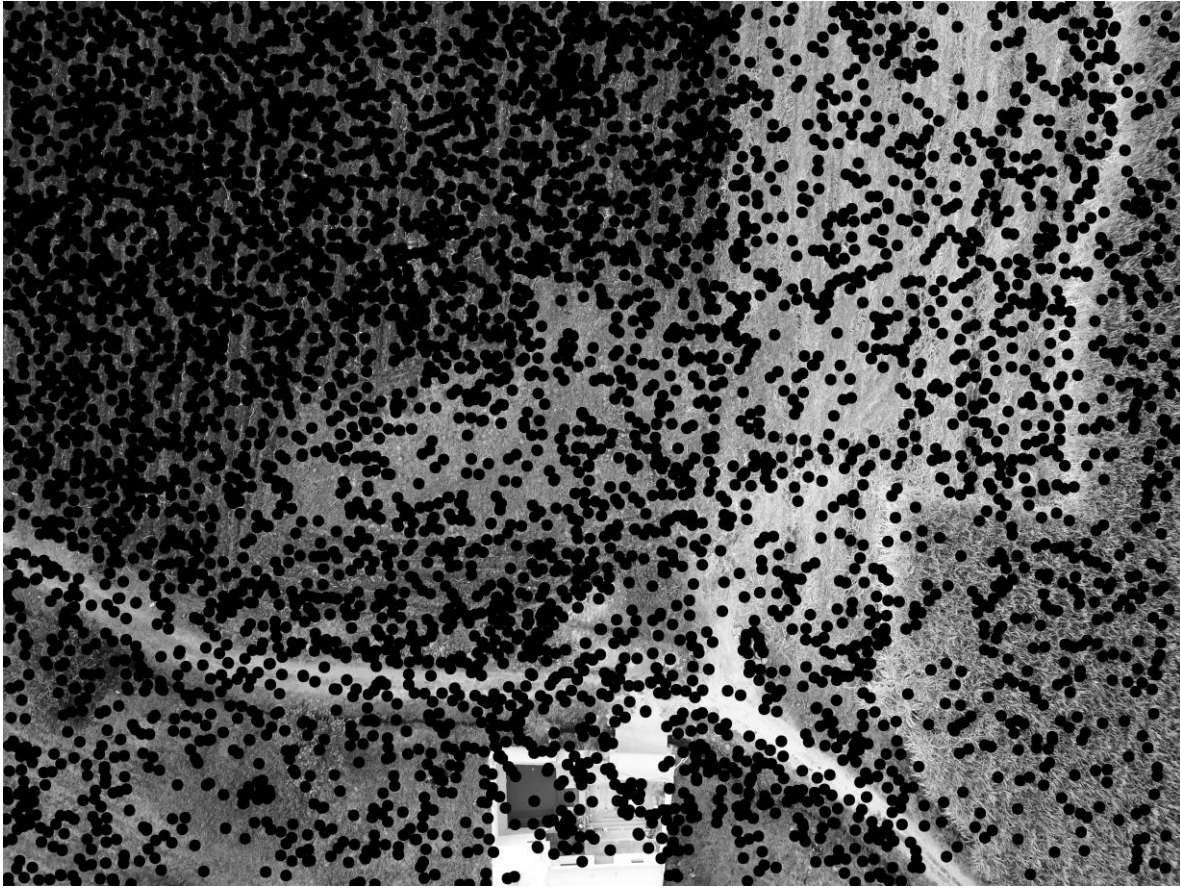


Figura B.2 Ejemplo de re-proyección 2.

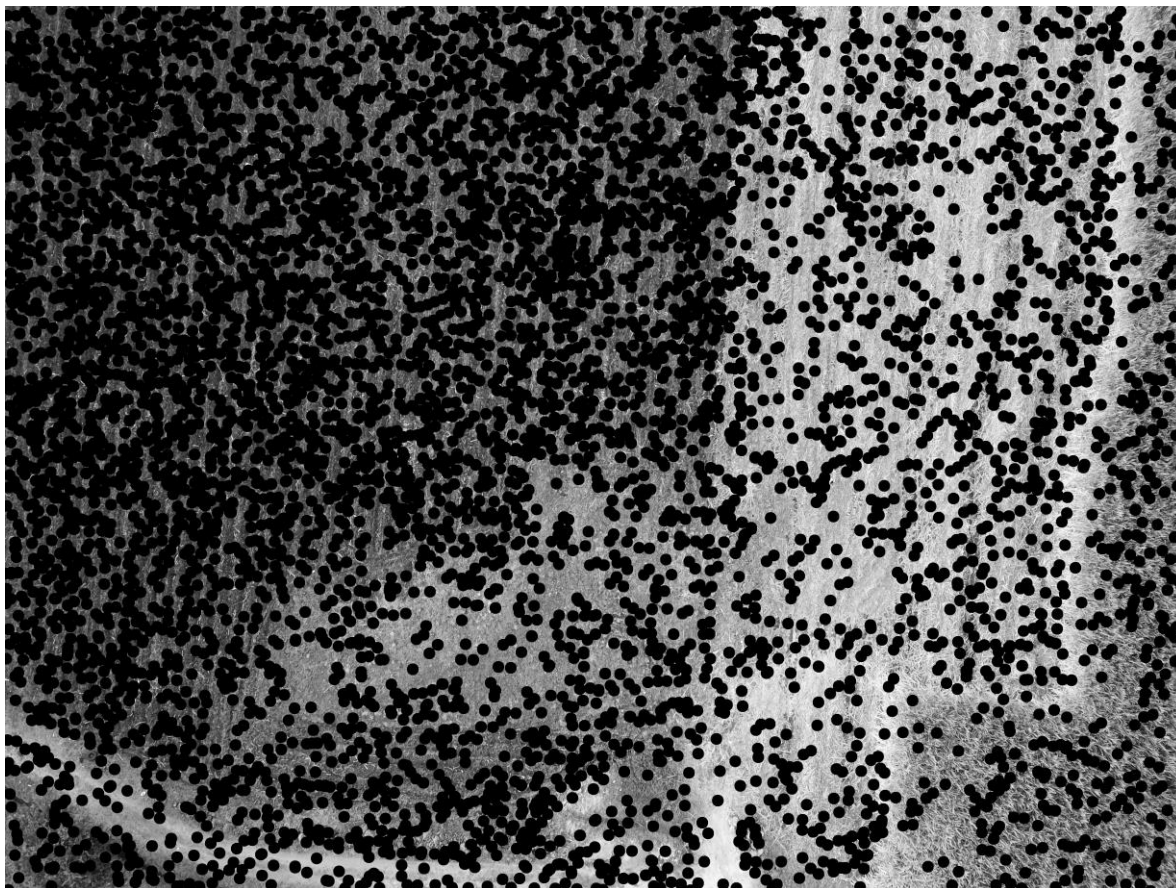


Figura B.3 Ejemplo de re-proyección 3.