

GRADO UNIVERSITARIO EN  
INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA  
2016/2017

*Trabajo Fin de Grado*

# **Aerial Panoramic Image Reconstruction for Inspection and Survey Purposes**

---

Autor:

Juan Camilo Soto Triviño

Tutor:

Abdulla Hussein Abdulrahman Al Kaff



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**



**Palabras clave:** Imagen panorámica, UAV, Inspección, Feature Matching, cámara monocular

## Resumen

Debido al aumento de la demanda de aplicaciones relacionadas con la inspección y el reconocimiento en las que se precisa el uso de UAVs, la reconstrucción de imágenes panorámicas se ha convertido en un campo en el que actualmente se investiga muy activamente por los expertos en visión por computador. Por lo tanto, este proyecto tiene como objetivo el desarrollo de un algoritmo capaz de crear imágenes panorámicas, que sea capaz de hacerlo de tal manera que sea un algoritmo: lo suficientemente rápido para trabajar en tiempo real, con el cual se pierda la menor cantidad de información posible, que se pueda integrar fácilmente en el sistema del UAV y que se pueda aprovechar para incorporar en el futuro otras técnicas relacionadas con la detección de objetos o la odometría visual. Para cumplir con los objetivos de trabajo en tiempo real y mínima pérdida de información, se propone un método de reconstrucción controlada, en el que se evalúa y se selecciona en todo momento las imágenes que van a formar parte de la panorámica. También, el algoritmo evalúa cuando no se puede continuar con la creación de la panorámica y se debe empezar con otra rápidamente sin perder información. Por último, para cumplir con el objetivo de fácil integración en el sistema, se propone el uso de la estructura ROS, que se basa en el intercambio de mensajes entre diferentes nodos (subsistemas).

**Keywords:** Panoramic image, UAV, Inspection, Feature Matching, Monocular camera

## **Abstract**

Due to the rising of application demand related to inspection and survey in which are required UAVs, panoramic image reconstruction has become a field where, currently, computer vision experts actively dive in it. Therefore, the objective of this project is to develop an algorithm capable to create panoramic images, so that it is an algorithm: fast enough to work in real time, in which lower possible data will be lost, simple to integrate within the whole UAV system and functional to include other techniques in the future such as object detection and visual odometry. To reach the objectives related to real time and lower data losing, it is proposed a method of checked reconstruction. In that method, an evaluation and a selection of the capture images is done constantly to find the best image to merge with the panorama. Besides, the algorithm decides the moment in that the reconstruction must stop and when a new panoramic quickly must be created before losing information. Finally, to reach the objectives related to simple integration, it is proposed using the framework ROS that is based on exchanging messages between different nodes.

# Índice general

<b>Lista de figuras</b>	<b>IV</b>
<b>Lista de tablas</b>	<b>VI</b>
<b>CAPÍTULO 1: INTRODUCCIÓN</b>	<b>2</b>
1.1. UAVS . . . . .	2
1.1.1. Clasificación por tamaño . . . . .	2
1.1.2. Clasificación por alcance y autonomía . . . . .	3
1.1.3. Clasificación por tipo de tecnología de sustentación . . . . .	4
1.1.4. Clasificación según la aplicación . . . . .	6
1.2. Descripción del problema . . . . .	7
1.3. Exposición de los Capítulos del proyecto . . . . .	9
<b>CAPÍTULO 2: ESTADO DEL ARTE</b>	<b>11</b>
2.3.1. Reconstrucción de imágenes Panorámicas . . . . .	11
2.3.2. Creación de imágenes panorámicas a partir de imágenes aéreas	14
2.3.3. Reconstrucción de imágenes Panorámicas en tiempo real . . . . .	15
<b>CAPÍTULO 3: ALGORITMO PROPUESTO</b>	<b>18</b>
3.4. Modelo matemático . . . . .	20
3.4.1. Modelo <i>Pin Hole</i> y matriz de Homografía . . . . .	20
3.4.2. Estimación de la matriz de Homografía . . . . .	23
3.4.3. Cambio de perspectiva en la imagen capturada . . . . .	23
3.4.4. Obtención de la matriz de Rotación . . . . .	24
3.5. Ecuación del histograma . . . . .	27
3.6. Extracción de características en las Imágenes . . . . .	29
3.7. <i>Matching</i> de las imágenes . . . . .	31
3.8. Búsqueda de “costuras” entre las imágenes . . . . .	32
3.9. Unión de las imágenes usando <i>Feather Blending</i> . . . . .	33
3.10. Selección de la mejor imagen para realizar el Blending . . . . .	35
3.10.1. Obtención del Porcentaje de Superposición entre imágenes . . . . .	36

3.10.2. Selección de la mejor imagen . . . . .	38
3.11. Creación de una nueva panorámica . . . . .	38
3.12. Uso de <i>Clustering</i> para segmentar fachadas . . . . .	39
<b>CAPÍTULO 4: EXPERIMENTOS Y RESULTADOS</b>	<b>42</b>
4.13. Resultado de los tiempos y el número de <i>inliers</i> según el número de imágenes . . . . .	44
4.14. Resultados obtenidos al modificar el factor de escala en algunos métodos . . . . .	50
4.15. Resultados obtenidos al modificar del Porcentaje de Superposición .	53
<b>CAPÍTULO 5: CONCLUSIÓN Y TRABAJOS FUTUROS</b>	<b>57</b>
5.16. Conclusión . . . . .	57
5.17. Trabajos futuros . . . . .	58
<b>CAPÍTULO 6: MARCO REGULADOR</b>	<b>61</b>
<b>CAPÍTULO 7: ANÁLISIS SOCIO-ECONÓMICO</b>	<b>64</b>
6.18. Presupuesto . . . . .	64
6.19. Aplicaciones propuestas en las que las imágenes panorámicas serían de utilidad . . . . .	67
<b>Bibliografía</b>	<b>74</b>
<b>Anexos</b>	<b>75</b>
<b>I. Pseudocódigo del algoritmo</b>	<b>77</b>
<b>II. Imágenes panorámicas creadas en los experimentos</b>	<b>78</b>

# Índice de figuras

3.1. Punto representado en dos <i>Frames</i> diferentes . . . . .	21
3.2. Estimación de la matriz de Homografía mediante RANSAC . . . . .	24
3.3. Cambio de perspectiva en la imagen capturada; <b>a</b> : Formas observadas desde la primera imagen, <b>b</b> : Formas observadas desde la primera imagen, y <b>c</b> : Segunda imagen transformada . . . . .	25
3.4. Dibujo del Histograma antes y después de ser Ecuilizado (eje horizontal normalizado de 0 a 255 y eje vertical normalizado de 0 al número máximo de píxeles) . . . . .	28
3.5. Ecuilización del Histograma usando el dataset del Lago como ejemplo	29
3.6. Keypoints encontrados en la region de interes de la panorámica y en la nueva imagen adquirida . . . . .	30
3.7. Good Fetures encontrados después del Matching de los descriptores	32
3.8. Inliers encontrados después de utilizar RANSAC . . . . .	32
3.9. Camino encontrado que determina la línea de “costura” entre las dos imágenes . . . . .	33
3.10. Valor de $\alpha$ a lo largo de la imagen al usar Feather Blending . . . . .	35
3.11. Calculo del area de superposición a partir del uso de máscaras. a) se corresponde con la máscara de la panorámica y b) con la de la nueva imagen capturada. El área de superposición se muestra en azul en c). . . . .	37
3.12. Área de superposición entre dos imagenes . . . . .	37
3.13. Bucle de evaluación de las nuevas imágenes capturadas . . . . .	39
3.14. Uso de clustering para segmentar la fachada. Imagen original de la fachada a), grupos después de usar K-Means b) e imagen binaria después de usar un threshold c). . . . .	40
4.15. Gráfica nº Imágenes - Tiempo en segundos (Matching, Matriz de rotación y Superposición), del dataset <i>Imágenes Urbanas 2</i> . . . . .	45
4.16. Gráfica nº Imágenes - Tiempo en segundos (Costuras y Union de imágenes), del dataset <i>Imágenes Urbanas 2</i> . . . . .	46

4.17. Gráfica nº Imágenes - Tiempo total(s), del dataset <i>Imágenes Urbanas 2</i> . . . . .	46
4.18. Gráfica nº Imágenes - Inliers, del dataset <i>Imágenes Urbanas 2</i> . . . . .	47
4.19. Gráfica nº Imágenes - Tiempo en segundos (Matching, Matriz de rotación y Superposición), del dataset <i>Imágenes Urbanas 1</i> . . . . .	48
4.20. Gráfica nº Imágenes - Tiempo en segundos (Costuras y Union de imágenes), del dataset <i>Imágenes Urbanas 1</i> . . . . .	49
4.21. Gráfica nº Imágenes - Tiempo total(s), del dataset <i>Imágenes Urbanas 1</i> . . . . .	49
4.22. Gráfica nº Imágenes - Inliers, del dataset <i>Imágenes Urbanas 2</i> . . . . .	50
6.23. Acumulación de caucho y marcas de la pista de aterrisaje del aeropuerto de seattle . . . . .	68
II.24.Dataset <i>Imágenes Urbanas 1, compuesto por 12 imágenes</i> . . . . .	78
II.25.Dataset <i>Imágenes Urbanas 2, compuesto por 20 imágenes</i> . . . . .	79
II.26.Imagen panorámica construida con el dataset <i>Imágenes Urbanas 1</i> . . . . .	80
II.27.Imagen panorámica construida con el dataset <i>Imágenes Urbanas 2</i> . . . . .	81
II.28.Imagén panorámica al 75 % de Superposición - dataset <i>Vídeo Lago</i> . . . . .	82
II.29.Imagén panorámica al 55 % de Superposición - dataset <i>Vídeo Mural</i> . . . . .	83
II.30.Dataset del aeropuerto de Seattle . . . . .	84
II.31.Panorámica del aeropuerto de Seattle (reconstrucción de la ortofoto) . . . . .	85



# Índice de tablas

1.1. Clasificación de las diferentes Aplicaciones Civiles que desempeñan los UAVs . . . . .	7
4.2. Resumen de los experimentos realizados . . . . .	43
4.3. Factores de escala aplicados en las 7 panorámicas . . . . .	52
4.4. Tiempos de reconstrucción de las 7 panorámicas (cambiando la escala de las imágenes) . . . . .	52
4.5. Tiempo de ejecución - dataset <i>Vídeo Lago</i> . . . . .	53
4.6. Tiempo de ejecución - dataset <i>Vídeo Mural</i> . . . . .	54
4.7. Información de los resultados - dataset <i>Vídeo Lago</i> . . . . .	54
4.8. Información de los resultados - dataset <i>Vídeo Mural</i> . . . . .	55
6.9. Capítulo 1 del presupuesto: estructura del UAV . . . . .	65
6.10. Capítulo 2 del presupuesto: sensores . . . . .	65
6.11. Capítulo 3 del presupuesto: sistemas . . . . .	65
6.12. Capítulo 4 del presupuesto: mano de obra . . . . .	66
6.13. Resumen de los capítulos del presupuesto . . . . .	66

# **CAPÍTULO 1: INTRODUCCIÓN**

Actualmente, existe una gran cantidad de aplicaciones, tanto militares como civiles, relacionadas con la toma de datos aéreos. El uso de Vehículos Aéreos no Tripulados (UAVs), se ha generalizado para llevar a cabo tareas importantes en dichas aplicaciones. En un principio, los UAVs solo se utilizaban en el ámbito militar, debido a su coste y la complejidad de su diseño. Su uso en aplicaciones civiles se ha extendido gracias a que, esta complejidad y este coste, se ha reducido considerablemente en los últimos años en parte gracias al uso de Multi-Rotores. Por otro lado, la adquisición de datos aéreos, ha mejorado mucho debido al gran incremento de la eficiencia de las unidades de computación.

## 1.1. UAVS

Aunque se puede categorizar a los UAVs de muchas maneras diferentes, en este trabajo se hará un resumen de las más relevantes y de esta manera, saber que alternativas se tienen en el momento de decidir que sistema utilizar. Se puede realizar una clasificación de estos basada en: el tipo de misión (destinados a aplicaciones de topografía, fotografía aérea, salvamento, combate, Anti incendios...), su tamaño, su alcance o autonomía y su forma de mantener la sustentación. Las clasificaciones se han llevado a cabo según [1], en el caso de tamaño y alcance o autonomía; según [2], en el caso del tipo de misión.

### 1.1.1. Clasificación por tamaño

Si se realiza una clasificación de los UAVs por tamaño, se pueden organizar en: UAVs Muy Pequeños, que a su vez se puede dividir en Micro o Nano UAVs; en Pequeños; en Medianos; y por último, en Grandes. Normalmente, la elección entre un tipo y otro de UAV estará relacionada con el peso de la carga de pago que se desee transportar, con el tiempo de autonomía requerido en la aplicación y el coste del sistema, entre otros muchos factores.

- **UAVs Muy Pequeños**

Dentro de esta clase, se encuentran los UAVs que tienen el tamaño de un gran insecto de 30 a 50 cm de longitud. Existen dos tipos de UAVs pequeños dentro de esta categoría. Uno de los tipos mencionados, son los UAVs que su diseño incorpora alas parecidas a las de un insecto o un pájaro y el otro tipo de UAVs son los que incorporan alas giratorias, más parecidas a las de una aeronave. Estos últimos suelen ser los más grandes de las dos categorías. La elección entre el primer tipo de alas y las alas giratorias esta basada en el tipo de maniobrabilidad que se desea a la hora de controlar el dron. Las

alas de insecto o pájaro, permiten un aterrizaje en superficies más pequeñas y menos estables. Esto último proporciona al UAV vigilancia continua sin tener que gastar energía en la sustentación.

- **UAVs Pequeños**

Dentro de los UAVs Pequeños, pueden ser categorizados los drones que al menos tengan un tamaño mayor a 50 cm y que pueden llegar a uno o dos metros. Muchos de estos UAVs son de ala fija y tienen que ser lanzados por la mano del operario para hacerlos despegar.

- **UAVs Medianos**

Los UAVs Medianos, son los UAVs demasiado grandes para ser transportados por una persona, pero además, son mucho más pequeños que una aeronave ligera. Estos normalmente tienen un ala que puede llegar a medir 10 m y pueden llevar cargas de peso de hasta 200 Kg. En esta categoría existen tanto drones de ala fija como de alas giratorias.

- **UAVs Grandes**

En esta categoría, principalmente se encuentran los UAVs que son destinados a operaciones de combate.

### 1.1.2. Clasificación por alcance y autonomía

Según el alcance y la autonomía de vuelo, se pueden diferenciar los siguientes tipos de UAVs:

- **UAVs de cercano alcance y bajo coste.** Esta clase incluye los UAVs con un alcance de 5 Km y una autonomía de 20 a 45 minutos.
- **UAVs de corto alcance.** En esta clase se incluyen los UAVs que tienen un alcance de 50 Km y un tiempo de autonomía de 1 a 6 horas. Estos drones son normalmente usados en aplicaciones de reconocimiento y vigilancia.
- **UAVs de medio alcance.** En este tipo de UAVs se incluyen los de alta velocidad que pueden trabajar en radios de 650 Km. Estos drones aparte de trabajar en aplicaciones de reconocimiento y vigilancia, también suelen hacerlo en aplicaciones relacionadas con la recogida de datos meteorológicos.
- **UAVs de largo alcance.** Estos UAVs también suelen trabajar en misiones de reconocimiento y vigilancia, con la diferencia de que tienen una autonomía de 36 horas y suelen trabajar en radios de 300 Km.

### 1.1.3. Clasificación por tipo de tecnología de sustentación

A continuación, se explicarán los diferentes tipos de drones según el tipo de tecnología que estos utilizan para mantenerse flotando en el aire. Además, se realizará un resumen de las desventajas y ventajas de cada uno.

#### ■ UAVs Multirotor

Los multirotor son lo tipos de drones más comunes usados por profesionales y aficionados. Estos drones son usados en aplicaciones comunes, como por ejemplo las aplicaciones relacionadas con fotografía aérea e inspección. Además, en la actualidad, son muy usados para propósitos relacionados con el ocio y las carreras.

Dentro de esta categoría, existen cuatro tipos de drones según el número de rotores que le proporcionan sustentación. Existen los *Tricópteros*, que contienen tres rotores; los *Cuadricópteros*, con cuatro rotores; los *Hexacópteros*, con seis rotores; y los *Octacópteros*, que poseen ocho rotores.

Los UAVs Multirotor, son la opción más fácil y barata de colocar una cámara en el aire, ya que otorgan un excelente control en posición. Por lo tanto, son perfectos para capturar imágenes. Por otro lado, tienen el inconveniente de que están muy limitados en autonomía y velocidad, haciéndolos una mala elección para adquirir datos en grandes superficies, para monitorización por largos periodos de tiempo y largas distancias de inspección (por ejemplo, inspección de carreteras o vías férreas). A pesar de que la tecnología esta mejorando continuamente, los UAVs Multirotor siguen siendo muy ineficientes, requiriendo una gran cantidad de energía solo para elevarse y mantenerse en el aire. Con las baterías actuales, siguen estando limitados a un tiempo de vuelo de 20 a 30 minutos llevando una cámara como carga de pago.

#### ■ UAVs de Ala Fija

Los drones de ala fija son totalmente diferentes en diseño y construcción que los multirotor. A diferencia de estos últimos, los UAVs de ala fija no utilizan energía cuando están planeando, necesitando únicamente de esta para moverse hacia delante y no para contrarrestar la fuerza de la gravedad como lo hacen los multirotor. Por esta razón, los UAVs de ala fija son capaz de cubrir largas distancias y estudiar grandes superficies. Además de su eficiencia, es posible usar motores de combustión como fuente de energía, pudiendo algunos UAVs estar en el aire durante 16 horas o más.

Obviamente, la desventaja de utilizar un dron de ala fija es que no se puede mantener flotando en una posición. Esto último hace que este tipo de drones sea excluido de algunas aplicaciones que requieran capturar imágenes aéreas. Otras desventajas de utilizar este tipo de UAVs es que, según el tamaño, pueden tener un sistema de despegue o aterrizaje complejos (adicionalmente se puede requerir el uso de catapultas para despegar, o también, paracaídas o redes para recuperarlo con seguridad). Solo los drones de ala fija pequeños, como se mencionó en la clasificación por tamaño, pueden ser lanzados con la mano por un operario para su despegue y pueden aterrizar con la panza (sin tren de aterrizaje).

Otras desventajas que se pueden relacionar con los UAVs de ala fija, son las habilidades necesarias para volarlos y el precio elevado que tienen. Como se menciono arriba, se requiere de otras tecnologías para despegar y aterrizar o de una pista de aterrizaje, lo que encarece más su uso. Al contrario que los cuadrópteros por ejemplo (el control de un cuadróptero se puede ir aprendiendo sobre la marcha), los UAVs de ala fija precisan de mayor destreza en el despegue, en el vuelo y en el aterrizaje.

#### ■ UAVs con un solo rotor

El diseño y la estructura de un UAV con un solo rotor es bastante similar al de los helicópteros. Al contrario que los multirrotor, un modelo de un solo rotor esta compuesto por un rotor de gran tamaño que le otorga sustentación, más un rotor más pequeño en la cola que controla el *heading* (o ángulo de navegación) de este.

El UAV con un solo rotor es mucho más eficiente que un multirrotor y también puede poseer un motor de combustión como en el caso de los UAV de ala fija, lo que le proporciona mayor tiempo de vuelo. En aerodinámica, cuanto mayor sea el tamaño de las hélices y cuanto más lenta sea su velocidad de giro, más eficiente suele ser el dron. Además, gracias al tamaño de sus hélices, el UAV de un solo rotor es capaz de transportar mayor carga de pago y sensores más pesados, como por ejemplo un escaner laser LIDAR.

Las desventajas de utilizar este tipo de UAVs, radican en la alta complejidad, su elevado costo y el alto riesgo de operación que poseen. También, aunque por un lado tienen la ventaja de mantenerse flotando en un posición en concreto, por otro lado no son tan estables como los drones multirrotor. Además, estos vehículos también demandan un entrenamiento especial para maniobrar el despegue y el aterrizaje, aunque no precisan de plataformas ni catapultas para hacerlo.

- **VTOL Híbrido: de Ala Fija + Multirotor**

Este tipo de UAV mezcla algunos beneficios de los drones de ala fija con la habilidad de poder despegar y aterrizar verticalmente, además de poder mantenerse flotando en una posición. El diseño más común para este tipo de vehículos, se basa en un ala fija con motores de elevación vertical fijados en la estructura. Otros diseños, incluyen una hélice en la cola que proporciona un empuje hacia delante al UAV. También, existen vehículos en los cuales sus rotores pueden pivotar, de tal manera que estos cumplen la función de despegue y aterrizaje vertical, más la función de propulsar el dron hacia delante.

Debido a la complejidad de muchas de las configuraciones de los VTOLs híbridos, hasta que no se desarrollaron los autopilotos, giróscopos y acelerómetros modernos, ha sido una tarea complicada mantener estables estos tipos de UAVs.

#### 1.1.4. Clasificación según la aplicación

En esta clasificación, se describen numerosas áreas en el sector civil en las cuales los UAVs pueden ser utilizados. En la tabla 1.1.4, se enseña un resumen de esta clasificación que divide los UAVs en seis tipos según [2].

El uso de UAVs para prevenir desastres o ayudar una vez ocurridos, es de gran importancia. Por ejemplo, los UAVs pueden ser utilizados en desastres naturales como incendios forestales, inundaciones, terremotos y tormentas para observar y analizar la situación a través de la toma de datos en tiempo real. De manera frecuente, el área donde ocurre un terremoto o un huracán, queda de tal manera que los oficiales y equipos encargados de actuar en dichas áreas son incapaces de acceder a ellas. En situaciones de emergencia, la monitorización en tiempo real de datos es esencial. Al mismo tiempo, los UAVs pueden servir de apoyo en las operaciones de búsqueda y rescate.

En [3], Ged F. Griffin estudia el uso de UAVs en misiones de gestión de desastres y explica el tipo de carga de pago necesaria, a parte de las capacidades y los beneficios que proporciona el uso de drones, clasificándolo todo por misión. Los sensores más utilizados en este tipo de misiones son: sensores Electroópticos (referidos normalmente a las cámaras), sensores Infrarrojos de barrido frontal (FLIR), escáneres LIDAR, radares de tipo SAR y detectores químicos. Las capacidades más relevantes de los UAVS pueden ser: rápida reconstrucción del mapa de la zona del desastre, proporcionar información en tiempo real y proporcionar soporte en la comunicación. Esto último conlleva algunos beneficios,

Aplicaciones Civiles	Ejemplos
Prevención y Gestión de Desastres:	<ul style="list-style-type: none"> <li>• Prevención de fuegos forestales</li> <li>• Monitorización de inundaciones</li> <li>• Evaluación de daños en terremotos</li> <li>• Búsqueda de supervivientes</li> <li>• Asegurar áreas golpeadas por huracanes</li> </ul>
Protección del Medio Ambiente:	<ul style="list-style-type: none"> <li>• Monitorización pesca ilegal</li> <li>• Control de fugas de petroleo</li> <li>• Renaturalización</li> <li>• Monitorización de emisiones</li> </ul>
Investigaciones científicas:	<ul style="list-style-type: none"> <li>• Estudio geológico</li> <li>• Estudio Atmosférico</li> <li>• Estudio de huracanes</li> <li>• Observación de volcanes</li> <li>• Agricultura y Silvicultura</li> </ul>
Protección de Infraestructuras:	<ul style="list-style-type: none"> <li>• Inspección de tuberías de gas y fuel</li> <li>• Inspección de líneas electricas</li> <li>• Inspección de vias férreas</li> </ul>
Seguridad:	<ul style="list-style-type: none"> <li>• Vigilancia costera</li> <li>• Vigilancia en eventos con publico</li> </ul>
Misiones de Comunicación:	<ul style="list-style-type: none"> <li>• UAVs como sustitutos de satélites</li> <li>• Comunicaciones de banda ancha</li> <li>• Rele de comunicaciones</li> </ul>

**Tabla 1.1:** Clasificación de las diferentes Aplicaciones Civiles que desempeñan los UAVs

como por ejemplo: reducción del riesgo de daño a los oficiales; identificar accesos seguros y rutas de salida, para el personal de emergencia y los evacuados; y reducción del coste de operaciones por hora de vuelo.

Otro uso de los UAVs, esta relacionado con proteger infraestructuras críticas. Esto último incluye la protección de tuberías de gas o fuel, de las líneas eléctricas, la observación de raíles, carreteras y rutas marítimas. Es necesario que la red de tuberías (por ejemplo de gas natural) se encuentre en constante monitorización para prevenir accidentes y fugas de gas incontrolables.

## 1.2. Descripción del problema

El objetivo principal de este proyecto, es desarrollar un software capaz de construir imágenes panorámicas destinado a aplicaciones de inspección y reconocimiento, principalmente aplicaciones donde el escenario sea la fachada de un edificio, una carretera o pista, una vía férrea o un campo de cultivo visto



desde una perspectiva aérea. Para tal propósito, se quiere utilizar un UAV (un Hexacóptero) y una cámara monocular, formando un sistema sencillo y asequible de captura de imágenes aéreas.

El primer problema que se presenta, es que solo se posee una cámara con un solo objetivo para crear la imagen panorámica, es decir, no se posee ningún otro sensor como por ejemplo: una cámara RGB-D (cámara estéreo), un radar, un barómetro, un sensor inercial o un sensor láser (el autopiloto si posee una IMU, un barómetro y un gps, pero no son sensores muy precisos). Solo con una cámara monocular, no se puede saber la distancia a la que se encuentra el dron del suelo o de una fachada. Por lo tanto, es necesario realizar la alineación de las imágenes (para unir las) sin tener ningún conocimiento de la posición del dron respecto al escenario, haciendolo de tal manera que no se note la transición entre ambas imágenes. Para esto, es necesario utilizar algún método que encuentre similitudes entre ellas y luego las deforme de alguna manera para poder realizar la alineación. Para este proyecto, se resolverá este problema proyectando todas las imágenes al plano de la imagen panorámica utilizando la matriz de rotación y la escala calculada a partir de la matriz de Homografía. Esto soluciona el problema porque se asume que la primera imagen es totalmente paralela al plano del escenario, pero tiene el inconveniente de que si se comete algo de error se irá acumulando a lo largo de la reconstrucción.

También, es muy importante que la cámara monocular se encuentre debidamente calibrada, teniendo conocimiento de la distancia focal y de la posición del centro óptico (en unidades de pixel) para realizar correctamente las proyecciones de las imágenes y, además, conocer el valor de la distorsión para corregirla antes de que esta afecte a los resultados.

Otro objetivo, es la realización de la construcción de la panorámica en tiempo real. Esto es muy útil para realizar acciones con los datos obtenidos en tiempo real, como por ejemplo: detectar objetos, hacer tracking de ellos y marcarlos o marcar su trayectoria en la panorámica etc. También, se ahorra tiempo y memoria al no tener que guardar todos los frames de las imágenes tomadas o al no tener que realizar un postprocesado de las imágenes. El inconveniente de trabajar en tiempo real, es que no es un problema fácil de resolver y se puede perder información en la construcción que luego no se puede recuperar, algo que no ocurriría con un postprocesado supervisado. Para resolver el problema de trabajar en tiempo real, primero, se evaluarán los mejores métodos para realizar el proceso de buscar características y hacer el Matching entre las imágenes. También, se modificará la escala de la imagen en diferentes pasos del algoritmo para ver como mejora la velocidad de este, siempre intentando perder la menor cantidad posible de información.

Debido a que trabajar en tiempo real, no permite una supervisión humana

de como se esta reconstruyendo la panorámica, es imprescindible tener un control de la realización de esta cuando se haga a bordo. Para esto, se implementará un sistema que evalúe las imágenes y elija la mejor de ellas para realizar la unión de la nueva imagen a la panorámica.

### **1.3. Exposición de los Capítulos del proyecto**

Este documento esta organizado en siete capítulos. En el Capítulo 2 se presenta el Estado del Arte, donde se exponen trabajos relacionados y previos a este, y donde se comparan algunas metodologías utilizadas con las propuestas en este proyecto. En el Capítulo 3, se explica en detalle el algoritmo propuesto para realizar la reconstrucción de la panorámica y el modelo matemático utilizado para entender mejor el algoritmo, a parte del funcionamiento de este.

Después, en el Capítulo 4, se estudian los resultados del algoritmo, a partir de tres experimentos. Estos experimentos determinan si se puede trabajar en tiempo real y bajo que condiciones. También, se determinan los cuellos de botella y los errores más típicos ocasionados en la reconstrucción, así como la duración de esta.

En el Capítulo 5, se concluye con todo lo expuesto en los capítulos anteriores a este y sobretodo con lo obtenido en los resultados de los experimentos. Además, en este capítulo se proponen trabajos futuros que mejorarían la eficiencia del algoritmo y la calidad de los resultados.

A continuación, en el Capítulo 6, se realiza un análisis de la normativa vigente sobre operaciones relacionadas con el pilotaje de aeronaves por control remoto.

En último lugar, en el Capítulo 7 se presenta un presupuesto detallado basado en el trabajo realizado en laboratorio y se propone el diseño de un Hexacóptoro para la realización de experimentos aéreos. Para terminar, en este capítulo se exponen dos aplicaciones en las que sería útil el uso de un UAV con el sistema propuesto en este proyecto y como mejorarían el trabajo que actualmente se realiza en ellas.

# **CAPÍTULO 2:**

## **ESTADO DEL ARTE**

En este capítulo, se presentarán y se analizarán algunos trabajos relacionados con el enfoque propuesto en este proyecto. Todo esto, con el fin de comparar las diferentes técnicas utilizadas. En primer lugar, se describirán los distintos métodos expuestos en los trabajos relacionados con la reconstrucción de panorámicas. Después, se centrará la atención en los proyectos relacionados con la creación de las panorámicas a partir de imágenes aéreas. Por último, se pasará a explicar los diferentes métodos utilizados para reconstruir panorámicas en tiempo real.

### 2.3.1. Reconstrucción de imágenes Panorámicas

En primer lugar, se va a dividir el proceso de crear una panorámica en diferentes partes, que servirán de apoyo para explicar los diferentes métodos que existen para ello. Shikha Arya en su trabajo [4] divide el procesos de adherir dos imágenes en tres partes: calibración de la imagen, registro de la imagen y *Blending* (Unir las imágenes de manera que la transición entre ellas se haga de forma suave). En este proyecto, se ha dividido este proceso, añadiendo una parte más que se considera importante: la forma en la que la imagen es adquirida (adquisición de la imagen).

- **Adquisición de la imagen**

La adquisición de la imagen, es el primer paso que hay que realizar para crear la panorámica. Adquirir una imagen se puede definir como el proceso de capturar una escena gracias a la luz. Para que la construcción de la panorámica se haga de forma satisfactoria, la imagen debe ser capturada correctamente.

Para adquirir la imagen, se debe tener en cuenta como va a ser el proceso de reconstrucción, ya que según el algoritmo utilizado se debe hacer la captura de la imagen de una manera específica. Por ejemplo, para este proyecto las capturas de las imágenes consecutivas, se deben realizar manteniendo el sensor lo más paralelo posible al escenario u objeto (el procedimiento exacto se explicará más adelante). Otros proyectos, como el de Murtadha Alomran y Douglas Chai [5] para adquirir la imágenes, utilizan un trípode de tal manera que la cámara queda restringida a un solo eje de rotación, de esta manera se puede capturar en un radio de 360 grados obteniendo así imágenes panorámicas del entorno que rodea la cámara. Este tipo de restricción de la cámara suele ser el más típico en la reconstrucción de panorámicas utilizando una sola cámara, para ellos es necesario tener localizado el centro de rotación correctamente [6] si no se quiere tener defectos en los resultados. Otros trabajos, como [7] exponen métodos con los que no hace falta controlar el movimiento de la cámara o restringir como las imágenes son tomadas.

Para este proyecto, referente a la captura de las imágenes, se propone un método rápido pero que comprende ciertas restricciones y asunciones. En primer lugar, el escenario u objeto que se va fotografiar, tiene que ser totalmente plano (como la fachada de un edificio) o se tiene que encontrar a una distancia suficiente alta para considerarse plano (como la tierra vista desde el UAV). Además, el resultado de la panorámica depende de como se ha capturado la primera imagen. Esto último, es debido a que el método deformará las imágenes consecutivas a la primera, de tal manera que se proyecten todas al plano que forma esta. Con esto, se permite que el algoritmo no necesite de un postprocesado para construir la panorámica, ya que esta se va construyendo a medida de que se van capturando las imágenes.

#### ■ Calibración de la imagen

La calibración de la imagen, es usada para minimizar la diferencia existente entre modelo ideal de la lente de la cámara y el modelo real. También algunos defectos como la distorsión o las diferentes exposiciones de la cámara pueden ser reducidas.

En este proyecto, se propone realizar una ecualización del histograma de las imágenes antes de buscar características en ellas. De esta manera, se mejora el contraste de las imágenes reduciendo los efectos producidos por la exposición. Como se ha mencionado, ecualizar la imagen ayudara a encontrar mejor las características de esta, pero no compensará la iluminación en la imagen resultante. Otros trabajos como [8], proponen compensar los defectos producidos por la exposición de la cámara. Con esto, se intenta no tener cambios bruscos de iluminación entre las imágenes y eliminar el *ghosting*. El problema de estos métodos, es que son métodos postproceso que comparan la intensidad de los píxeles a lo largo del dataset de imágenes capturadas. También, suelen ralentizar mucho el algoritmo, pero a cambio de imágenes panorámicas con mejor transición con respecto a la iluminación.

#### ■ Registro de la imagen

En este punto, se define la técnica que será utilizada para alinear las imágenes que van a ser utilizadas para realizar la panorámica. Se suelen diferenciar dos técnicas. La primera compara la intensidad de los píxeles de una imagen con la intensidad de los píxeles de la otra, utilizando para ello algún método de minimización y así conseguir la alineación de las imágenes [9]. Estas técnicas se suelen definir como técnicas directas. En este proyecto nos centraremos en el segundo tipo de técnicas, el cual se basa en la búsqueda de características

que puedan compartir ambas imágenes. Estas características son comparadas usando una serie de descriptores locales en cada imagen.

Los métodos basados en características suelen estar divididos en: detección de las características, generación de descriptores globales, Matching o emparejamiento de las características, obtención de una matriz de Homografía y alineación de las imágenes [10].

La detección de características es crucial para algunas aplicaciones como el SLAM (Simultaneous Location And Mapping) [11], la odometría visual [12] o la detección de objetos [13]. Algunos detectores de características están basados en la detección de corners, mientras que otros se basan en cambios de textura de la imagen. Para que la detección de las características se haga de la mejor manera posible, es necesario encontrar un método que sea invariante respecto a la traslación, la rotación, la escala, una transformación afín y la presencia de ruido.

Para alinear ambas imágenes, es necesario obtener alguna matriz de transformación que permita esto. Normalmente, se utiliza un método iterativo denominado RANSAC, bastante robusto, que ayuda a obtener la matriz de Homografía la cual permite alinear ambas imágenes (a partir de la deformación de una de ellas). La matriz de Homografía, además contiene información de la matriz de transformación homogénea entre la posición de una captura y otra. Algunos trabajos relacionados donde usan RANSAC son: [14] y [10]. En este proyecto se utilizará RANSAC después de aplicar un método de Matching entre las características.

La técnica de alineación global usada con más frecuencia es la técnica de *Bundle adjustment*, la cual es una técnica de fotogrametría usada para combinar múltiples imágenes en una misma escena de tres dimensiones. En este paso, primero se calcula una estimación inicial de las características en un entorno 3D y una estimación de las posiciones de la cámara. Después, se aplica el algoritmo de Bundle adjustment para obtener los valores óptimos de la reconstrucción 3D, para ello se minimiza la probabilidad logarítmica de los errores de proyección entre el conjunto de puntos predicho y el observado, usando para ello un algoritmo de mínimos cuadrados.

En este proyecto no se hará una reconstrucción del escenario 3D ni una alineación global, la alineación de las imágenes se realizará a partir de la deformación de una de ellas teniendo en cuenta la matriz de Rotación y la traslación en unidades de pixel.

- **Blending**

El último paso para crear la panorámica, es unir las imágenes. En un algoritmo de Blending, en primer lugar se debe elegir la superficie de composición (por ejemplo, una superficie plana, cilíndrica, esférica, etc.) y entonces decidir cómo unir estas imágenes para obtener la panorámica deseada. En este proyecto, la superficie de composición la determinará la primera imagen y será el plano de proyección de esta. Existen diferentes métodos de Blending, alguno de ellos basados en el gradiente y en descomponer la imagen piramidalmente. En este proyecto se utiliza un método basado en suavizar y difuminar los bordes de las imágenes, aunque los dos mencionados anteriormente son los métodos más utilizados debido a sus resultados, estos van difuminando la panorámica a medida que va creciendo debido al uso de gradientes [15].

Además, en este proyecto, se propone incluir diferentes métodos a lo largo del proceso de reconstrucción que ayuden a obtener un mejor resultado de la imagen panorámica. Uno de esos métodos es buscar de forma óptima (perdiendo la menor información posible) las costuras de las imágenes, es decir, la línea de unión entre una imagen y otra. También se propone un sistema de evaluación de las imágenes que van siendo adheridas a la panorámica, eligiendo la que el algoritmo considere mejor en cada momento. Esta elección, será basada en la correcta deformación de la imagen y en el porcentaje de superposición de la panorámica respecto a esta. Este sistema de evaluación, también decidirá si se debe detener la reconstrucción de la panorámica para empezar con una nueva.

### **2.3.2. Creación de imágenes panorámicas a partir de imágenes aéreas**

En aplicaciones aéreas que conllevan el uso de UAVs, se utilizan cámaras que no solo sufrirán un movimiento de traslación, sino que también sufrirán rotaciones debido a los movimientos del dron, haciendo la construcción de panorámicas una tarea más complicada. Comparado con otros métodos como [16], los cuales necesitan una velocidad constante y una orientación fija, el método propuesto contiene menos restricciones. El método propuesto en este proyecto, permite movimiento libre en el eje  $x$  e  $y$  de la cámara, así como movimientos de Yaw del dron. Además, este método tolera pequeños movimientos de Roll y Pitch que serán corregidos en la transformación de las imágenes, corrigiendo también los cambios de escala producidos por los movimientos en  $z$ . Sin embargo, unir muchas

imágenes puede conllevar que estos movimiento vayan generando cierto error que se va acumulando.

Algunos trabajos, proponen otros métodos para no acumular error en largas reconstrucciones. Uno de estos métodos, es georeferenciar las imágenes usando imágenes satélites como referencia ([17] y [18]). Otros enfoques, que impiden la acumulación de error son los métodos basados en posiciones [19], los cuales usan un sensor extra como un GPS/INS para obtener directamente posiciones de la cámara. Sin embargo, estos métodos necesitan sensores de alta precisión y alto coste para determinar correctamente las posición de la cámara.

Comparado con los métodos descritos arriba, el método propuesto en este proyecto, esta solo basado en la captura de imágenes, sin necesidad de obtener otro tipo de datos.

### **2.3.3. Reconstrucción de imágenes Panorámicas en tiempo real**

En [20], se explica que el uso de Bundle Adjustmet para minimizar el error de reproyección no es posible en aplicaciones de tiempo real, ya que el tiempo computacional es muy alto e incrementa cada vez que se toma una nueva imagen [21]. Por otro lado, el trabajo mencionado se basa en un algoritmo de extracción de características y estimación de una matriz de transformación. La matriz de transformación que obtiene es una matriz de Homografía y [20] utiliza esta matriz para deformar la imagen obtenida en el plano de la primera. A diferencia de [20], el método propuesto en este trabajo descompone la matriz de Homografía, corrigiendo el error producido por la rotación antes de alinear la imagen. Esto último permite una alineación más exacta de las características de las imágenes, ya que se puede realizar ajustando por una error de mínimos cuadrados la distancia entre los dos conjuntos de características en solo dos dimensiones.

En [22], se propone un método que se denomina “Separating Axis Theorem (SAT)”. Este método, es usado para detectar colisiones entre objetos [23]. En el caso de [22], utilizan una implementación de este método en 2D para no tener que buscar características en toda la imagen panorámica, ya que esto haría cada vez más lento el algoritmo de detección a medida de que la panorámica crece. Para esto último, en el algoritmo que se propone en este trabajo, se ha implementado un método muy sencillo que tiene en cuenta que la panorámica se va reconstruyendo a medida de que el dron avanza. Este método, crea una región de interés alrededor de la última zona actualizada de la panorámica y la va actualizando a medida que se van capturando imágenes. Visto de otra manera, es como hacer un tracking



del UAV dentro de la panorámica de tal manera que solo tenga que detectar características en la zona de interés y no a lo largo de toda la panorámica. En [24], se propone otro método para no tener que buscar características a lo largo de toda la panorámica cada vez que se captura una nueva imagen. Ellos denominan este método como Bag-of-Words, que utiliza un diccionario jerarquizado, permitiendo indexar rápidamente las imágenes y asociarlas a una palabra como muestra [25].

# **CAPÍTULO 3:**

## **ALGORITMO PROPUESTO**

En este capítulo, se explicará de forma detallada el funcionamiento de todo algoritmo propuesto. Primero se empezará con un breve resumen, para después desarrollar cada una de las partes en diferentes secciones.

El algoritmo de reconstrucción, empieza con un la adquisición de una nueva imagen. La primera imagen adquirida, debe cumplir con una serie de especificaciones (debe ser tomada con el sensor lo más paralelo posible al objeto). Las siguientes imágenes tomadas, se deben adquirir de tal manera que el sensor solo realice movimientos en el plano 2D definido por la posición de la cámara en el instante en la que obtiene el primer frame. Movimientos de Yaw del dron son permitidos, sin afectar al resultado de la panorámica. También, son permitidas pequeñas rotaciones en Pitch y en Roll, aparte de pequeños movimientos en el eje Z, pero serán considerados ruido a tratar. La segunda imagen adquirida, ya no tiene tanta importancia que la primera, ya que si es necesario el algoritmo la desechará. Una vez se ha realizado la unión de las dos primeras imágenes, la imagen resultante es considerada la imagen panorámica y a partir de este punto, las nuevas imágenes capturadas se adhieren a esta.

Después de realizar la captura de una nueva imagen, se buscan características con un algoritmo de detección (en este caso el algoritmo de detección de corners AGAST). Las características obtenidas de la imagen y de la panorámica con AGAST, serán denominadas a partir de este punto como *Keypoints*. Para mejorar la detección de los Keypoints, a priori se mejora la textura de la imagen aplicando una ecualización de su histograma. Una vez hallados estos Keypoints, se crean unos descriptores binarios que contienen la información local de estos, usando el algoritmo FREAK. Posteriormente, los Keypoints de una de las imágenes, se emparejan con las Keypoints de la otra, usando un algoritmo de Matching (emparejado) y la matriz de descriptores obtenida de ambas imágenes en el paso anterior. Los dos conjuntos de características obtenido después del Matching, serán denominado como *Good Features*.

Utilizando los Good Features, se utiliza el método iterativo RANSAC para obtener características de las imágenes que mejor se ajusten a una matriz de Homografía. Estas características serán denominadas *Inliers* y son utilizadas para calcular la matriz de Homografía a partir de mínimos cuadrados. Después de una correcta evaluación de la matriz de Homografía (si no es valida la matriz de Homografía, se capturará otra imagen), se descompone esta de tal manera que se obtiene la matriz de Rotación. Esta matriz de Rotación, será la utilizada para deformar la imagen actual a un plano paralelo de la panorámica. Después de deformar dicha imagen, y obtener la proyección de los Inliers de esta utilizando también la matriz de Rotación (que se denominarán *Inliers Proyectados*), se calcula

la escala necesaria para que la distancia entre dos Inliers de la panorámica y la distancia entre los dos Inliers Proyectados referentes a los dos primeros, sea la misma. Una vez calculada esta escala, se utiliza para deformar la nueva imagen capturada teniendo esta vez en cuenta tanto la matriz de Rotación como la escala. A partir de este punto, los Inliers Proyectados son los que se han obtenido en esta deformación. La escala mencionada también es utilizada para realizar una segunda evaluación, en la que se desecha la imagen como no se encuentre entre dos umbrales definidos.

En este punto, se alinea la imagen deformada con la imagen panorámica a partir de mínimos cuadrados, haciendo corresponder los Inliers de la panorámica con los Inliers Proyectados.

Una vez se han alineado ambas imágenes, se procede a calcular un valor aproximado del porcentaje de superposición entre ambas. En este proyecto, este porcentaje vendrá determinado por el área de superposición y el área de la imagen deformada. Para el cálculo del porcentaje de superposición, que puede comprender desde la deformación de la imagen hasta la obtención de este, se puede realizar con las imágenes reducidas para mejorar la eficiencia del algoritmo.

A continuación, otra evaluación es producida. Esta vez, se evalúa el porcentaje de superposición. Si este porcentaje es menor a un umbral (el 30%), se captura una nueva imagen. Si el porcentaje se encuentra entre dos umbrales definidos como Umbral Superior y Umbral Inferior, la imagen se guardará en la memoria y se evaluará posteriormente para elegir entre un conjunto de imágenes capturadas para seleccionar la mejor. Si el umbral está por debajo del Umbral Inferior, se procederá a la selección de la mejor imagen entre las guardadas en memoria.

Después de seleccionar la mejor imagen para formar parte de la panorámica, se calcula los píxeles que aportará la panorámica y esta en la nueva panorámica resultante. A este método se le denominará método de búsqueda de Costuras y se podrá procesar con las imágenes reducidas.

Por último, se utilizará *Feather Blending* para adherir la imagen seleccionada a la panorámica y hacerlo de forma que no se note la transición entre estas dos.

Si, dos imágenes capturadas, consecutivas, no pasan alguna de las evaluaciones propuestas, se detendrá la reconstrucción de la panorámica y se empezará con una nueva.

También, para este algoritmo, se propone que la búsqueda de Keypoints en la panorámica se realice en una región de interés, de esta manera no será necesaria la detección de características en toda la panorámica. Esta región, rodea la última zona actualizada de la panorámica, teniendo en cuenta que la nueva imagen capturada se superpondrá a dicha zona.

El pseudocódigo del algoritmo propuesto en este proyecto se puede observar en el Algoritmo 1 del Anexo I.

### 3.4. Modelo matemático

En esta sección, se expondrá el modelo matemático utilizado para dar una visión de como se proyectan los puntos tridimensionales (de los objetos capturados) en una imagen y de como se pueden transformar los píxeles de una imagen para cambiar de perspectiva entre las diferentes posiciones que ha tomado la cámara.

En primer lugar, se explicará que es una matriz de transformación homogénea, como funciona el modelo *Pin Hole* y como se define la matriz de Homografía. Después, se describirá como se obtiene la matriz de Homografía a partir de dos imágenes y como se puede descomponer esta para obtener una matriz de Rotación. Por último, se explicará como se puede utilizar esta matriz de Rotación para cambiar la perspectiva en la imagen capturada.

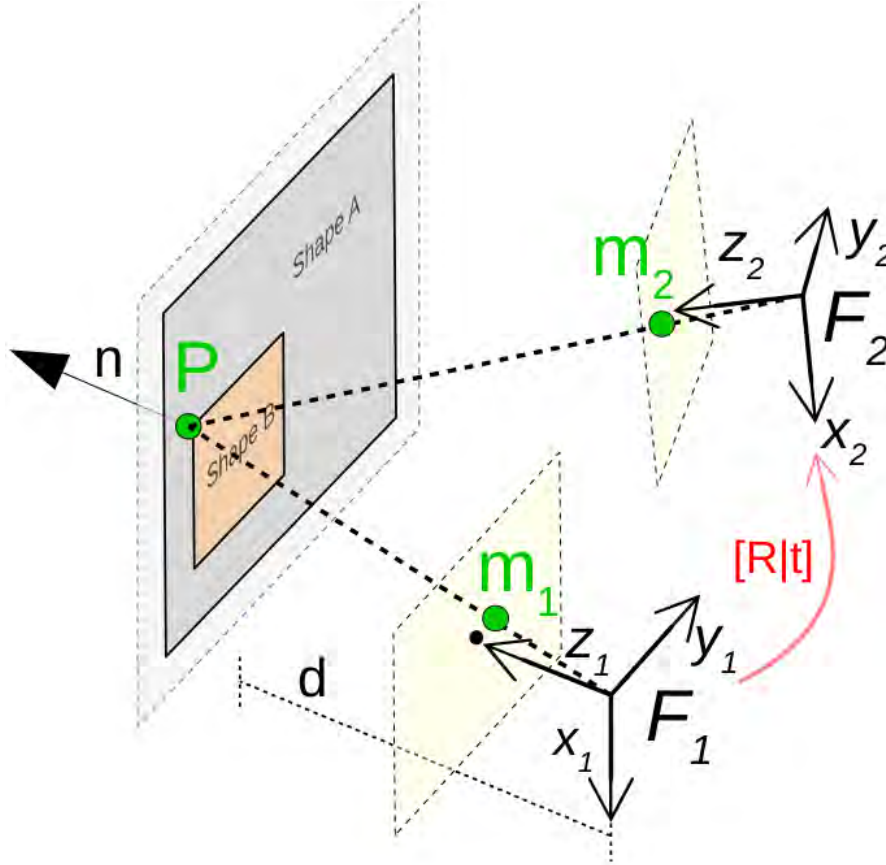
Para que este modelo funcione, es necesaria una correcta calibración de la cámara para evitar acumulación de errores por culpa de la distorsión. También, esta calibración es necesaria para obtener la matriz de parámetro Intrínsecos. Z.Zhang, propone en el siguiente trabajo [26] una manera sencilla y flexible de calibrar la cámara utilizando solo la cámara y un póster que muestra diferentes patrones.

#### 3.4.1. Modelo *Pin Hole* y matriz de Homografía

Para este proyecto, las diferentes posiciones que toma la cámara van a ser denominadas como *Frames*. En la Figura 3.1, se puede observar dos Frames consecutivos ( $F_1$  y  $F_2$ ) que se encuentran enfrente del plano del objeto.

En la Figura 3.1, se muestra como el Frame  $F_2$  ha sido rotado y trasladado con respecto al primer Frame. También se muestra, que el eje de coordenadas del sistema  $F_1$  esta configurado en un ángulo ideal, es decir, el eje  $z_1$  atraviesa de forma perpendicular el plano del objeto. La matriz, capaz de realizar un cambio de base a un vector tridimensional desde el primer Frame al segundo Frame, se denomina matriz de Transformación Homogénea y se puede observar en la Fórmula 3.1. Esta matriz esta compuesta por una matriz de Rotación  ${}^1R_2$  y un vector de Traslación  ${}^1t_2$ . Un dato a tener en cuenta es que si ambos Frames comparten el mismo centro óptico (centro del sistema de coordenadas), el vector de traslación tendrá todas sus componentes con un valor igual a cero. Por otro lado, si la matriz de Rotación es la matriz identidad, los ejes de coordenadas de ambos Frames deben de ser paralelos.

$${}^1T_2 = \begin{bmatrix} {}^1R_2 & {}^1t_2 \\ 0 & 1 \end{bmatrix} \quad (3.1)$$



**Figura 3.1:** Punto representado en dos *Frames* diferentes

Un punto en el espacio tridimensional, contenido en el plano del objeto, será denotado como  $P_{1i}$  y  $P_{2i}$  si está visto desde el sistema de coordenadas  $F_1$  y  $F_2$ , respectivamente. Este punto puede ser normalizado o proyectado, como se puede observar en la Ecuación 3.2 y 3.3.

$$m_{1i} = \left( \frac{x_{1i}}{z_{1i}}, \frac{y_{1i}}{z_{1i}}, \frac{z_{1i}}{z_{1i}} \right) = (x'_{1i}, y'_{1i}, 1) \quad (3.2)$$

$$m_{2i} = \left( \frac{x_{2i}}{z_{2i}}, \frac{y_{2i}}{z_{2i}}, \frac{z_{2i}}{z_{2i}} \right) = (x'_{2i}, y'_{2i}, 1) \quad (3.3)$$

Una vez el punto ha sido normalizado en alguno de los dos sistemas de

coordenadas, se puede obtener su posición como pixel en la imagen multiplicando dicho punto por la matriz de Parámetros Intrínsecos  $k$  3.4 como muestra en la Ecuación 3.5. Estando, la matriz de Parámetros Intrínsecos, formada por la distancia focal expresada en unidades de pixel ( $f_x$  y  $f_y$ ) y el centro de la imagen ( $o_x$  y  $o_y$ ).

$$k = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$p = Km \quad (3.5)$$

En las Ecuaciones 3.6 y 3.7, se puede observar como es el cambio de coordenadas del punto proyectado al sistema de la imagen en  $F_1$ . Como resultado de las operaciones, se obtienen los vectores  $(u_{1i}, v_{1i})$  y  $(u_{2i}, v_{2i})$ , que son píxeles pertenecientes a la primera y segunda imagen, respectivamente. Un ejemplo de la posición del sistema de coordenadas en la imagen, se puede observar en la Figura 3.3 a).

$$u_{1i} = f_x \cdot x'_{1i} + o_x \quad (3.6)$$

$$v_{1i} = f_y \cdot y'_{1i} + o_y \quad (3.7)$$

La Ecuación 3.8, permite transformar un pixel de la segunda imagen  $p_{2i}$  de tal manera que sufra un cambio de perspectiva, es decir, que pase de ser visto desde  $F_2$  a ser visto desde  $F_1$ . De esta manera, se podrá saber la posición de la proyección del punto tridimensional en la primera imagen.

$$s \cdot p_{1i} = \gamma k H k^{-1} p_{2i} \quad (3.8)$$

En la Ecuación 3.8, se define  $s$  como un factor de escala y  $H$  como la matriz de Homografía Euclidiana, compuesta por la matriz de Rotación, la matriz de Traslación y la normal al plano (ver Ecuación 3.9).

$$H = {}^1R_2 + \frac{{}^1t_2}{d_2}n^T \quad (3.9)$$

### 3.4.2. Estimación de la matriz de Homografía

Para el cálculo de la matriz de Homografía, se ha utilizado el método iterativo denominado RANSAC [27] (acrónimo inglés de RANdom SAMple Consensus) de la manera en la que se describe en el libro de Richard Hartley y Andrew Zisserman [28]. Los pasos seguidos en este proyecto para la estimación de la matriz de Homografía (ver Figura 3.2), desde que se obtienen los Good Features, son los siguientes:

- Primero, el algoritmo selecciona aleatoriamente cuatro parejas de características de los conjuntos de puntos denominados en este proyecto *Good Features*. Siendo  $GF$  el conjunto referente a la panorámica y  $GF'$  el referente a la nueva imagen capturada.
- En segundo lugar, se calcula una matriz de Homografía utilizando los cuatro pares de puntos y uno de los métodos básicos de estimación [29].
- Después, se determina que puntos del conjunto *Good Features* son *Inliers*, es decir, que puntos cumplen con la Formula 3.10. Donde  $gf_i$  es un Good Feature de la panorámica,  $gf'_i$  es su respectiva pareja de la imagen capturada y  $H$  la matriz de Homografía hallada en el punto anterior.

$$Distancia\ cuadrática(gf_i, H \cdot gf'_i) < 2,5 \quad (3.10)$$

- Los puntos que han sido determinados como *Inliers*, se van guardando en otros dos conjuntos de puntos creados. Obteniendo así los conjuntos  $Inl$  y el conjunto  $Inl'$  referidos a la imagen panorámica y la nueva imagen, respectivamente.
- Por último, después de un número determinado de iteraciones (dependiendo del número de *Inliers*), se recalcula la matriz de homografía  $H$  por mínimos cuadrado, utilizando los conjuntos  $Inl$  e  $Inl'$ .



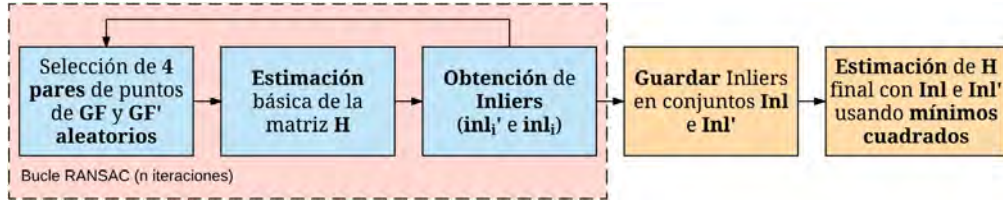


Figura 3.2: Estimación de la matriz de Homografía mediante RANSAC

### 3.4.3. Cambio de perspectiva en la imagen capturada

Para eliminar el ruido producido por la rotación de la cámara, se transforman todos los píxeles de la imagen con la matriz de Rotación obtenida (ver Ecuación 3.11). Esta transformación, es análoga a haber rotado sistema de coordenadas del Frame de la cámara, cambiando el punto de vista del escenario (ver Figura 3.3). Como resultado de la rotación, se obtiene una imagen paralela al escenario y a la primera imagen capturada.

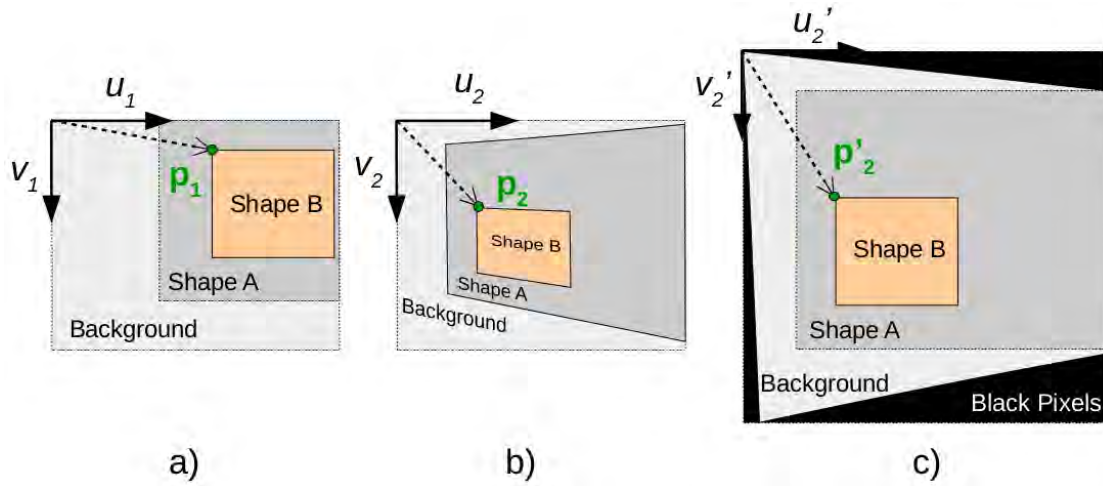
$$p'_{2i} = k({}^1R_2)k^{-1}p_{2i} \quad (3.11)$$

Esta rotación no es suficiente, ya que lo que se desea es que los ejes  $x$  e  $y$  de los Frames de la Figura 3.1 se encuentren en el mismo plano. Hasta este punto, los planos que forman dichos ejes, son paralelos entre sí. Para conseguir que además dichos ejes estén en el mismo plano, hay que escalar los puntos de la imagen. Por esto último, se aplica la Ecuación 3.12, dando como resultado la nueva posición de los píxeles de la segunda imagen en la nueva imagen.

$$p''_{2i} = s \cdot p'_{2i} \quad (3.12)$$

Para obtener el factor de escala  $s$ , primero se selecciona dos pares de puntos emparejados (puntos del mismo objeto que aparezcan en ambas imágenes) más lejanos. A continuación, los dos puntos de la segunda imagen son transformados utilizando la Ecuación 3.11. La distancia entre los dos puntos de la primera imagen  $l_1$  y los dos puntos de la segunda  $l_2$  son comparados como muestra la Ecuación 3.31 para obtener el valor de  $s$ .

$$s = \frac{l_1}{l_2} \quad (3.13)$$



**Figura 3.3:** Cambio de perspectiva en la imagen capturada; **a:** Formas observadas desde la primera imagen, **b:** Formas observadas desde la segunda imagen, y **c:** Segunda imagen transformada

#### 3.4.4. Obtención de la matriz de Rotación

El método utilizado para la descomposición de la matriz de Homografía, en una Matriz de Rotación y un Vector de Traslación, es el propuesto por Oliver Faugeras [30]. Este método está basado en la Descomposición en Valores Singulares de la matriz de Homografía. En [31], se obtiene que la matriz de Homografía está compuesta por otra matriz diagonal  $\Lambda$  que contiene los valores singulares de esta, y por otras dos matrices ortogonales, como muestra la Ecuación 3.14.

$$H = U\Lambda V^T \quad (3.14)$$

Esta matriz  $\Lambda$ , a su vez, se puede considerar también una matriz de Homografía, pudiendo adoptar la forma que se puede ver en la Ecuación 3.15. Obtener los componentes  $R_\Lambda$ ,  $t_\Lambda$  y  $n_\Lambda$ , es fácil cuando el resultado es una matriz diagonal. Primero que todo,  $t_\Lambda$  se puede eliminar fácilmente de las tres ecuaciones vectoriales que se obtienen de la Ecuación 3.15. Después se impone que  $R_\Lambda$  sea una matriz ortogonal, con lo que se consigue que se puedan calcular las componentes de  $n_\Lambda$  de forma lineal a partir de un nuevo conjunto de ecuaciones que relacionan únicamente las componentes con los tres valores singulares como se enseña en [30]. Como resultado, se obtiene 8 conjuntos de  $R_\Lambda$ ,  $t_\Lambda$  y  $n_\Lambda$ .

$$\Lambda = R_\Lambda + t_\Lambda \cdot n_\Lambda^T \quad (3.15)$$

A partir de este punto, con los valores obtenidos de  $\Lambda$  y sus componentes, se

puede aplicar las Ecuaciones 3.16, 3.17, y 3.18 para obtener las tres componentes  $R$ ,  $t$  y  $n$  (hallar el vector de traslación y la normal al plano, es necesario para los pasos que detallarán a continuación).

$$R = UR_{\Lambda}V^T \quad (3.16)$$

$$t = U \cdot t_{\Lambda} \quad (3.17)$$

$$n = V \cdot n_{\Lambda} \quad (3.18)$$

Como se puede observar, este método da como resultado ocho soluciones posibles, es decir, ocho conjuntos compuestos por: una matriz de Rotación, de un vector de traslación y de un vector normal al plano. Además, no todas las soluciones que nos da este método son físicamente posibles. El siguiente paso para obtener la matriz de Rotación es quedarse solo con una de las ocho soluciones.

El primer paso que hay que seguir, y que permite reducir el número de soluciones a cuatro, es imponer la restricción [32] de que ambos Frames  $F_1$  y  $F_2$ , representados en la Figura 3.1, deben estar en el mismo lado del plano del objeto; o lo que es lo mismo, que la distancia desde  $F_1$  a  $F_2$  no puede ser mayor a la distancia que hay entre  $F_1$  y el plano del objeto. La Ecuación 3.19 define matemáticamente la restricción que se impone y se puede reescribir como la Ecuación 3.20.

$${}^1t_2 \cdot n < d \quad (3.19)$$

$$1 + n^T R^T t < d \quad (3.20)$$

Esta restricción, da como resultado dos soluciones completamente diferentes y sus contrarios (componentes de  $t$  y  $n$  con signo cambiado). La siguiente restricción (que se basa en que todos los puntos que aparecen en la imagen, deben estar en frente de la cámara), dará como solución un conjunto y su contrario. Para esta imposición, es necesario tener un conjunto de puntos que aparezcan en las imágenes tomadas y la matriz  $K$  de parámetros intrínsecos. Los pasos a seguir, para eliminar dos de las soluciones, son los siguientes:

- Primero, como muestra la Ecuación 3.21, se debe re proyectar el punto de la

imagen al plano normalizado.

$$m_{1i} = k^{-1} \cdot p_{1i} \quad (3.21)$$

- Después, para que la solución sea válida, cada punto reproyectado y multiplicado por su normal (como muestra la Ecuación 3.22), debe dar como resultado un valor positivo. Lo mismo se puede comprobar utilizando el Frame  $F_2$  y las ecuación descrita en la Fórmula 3.23.

$$m_{1i}^T \cdot n_{1i} > 0 \quad (3.22)$$

$$m_{2i}^T (\mathbf{R} \cdot n_i) > 0 \quad (3.23)$$

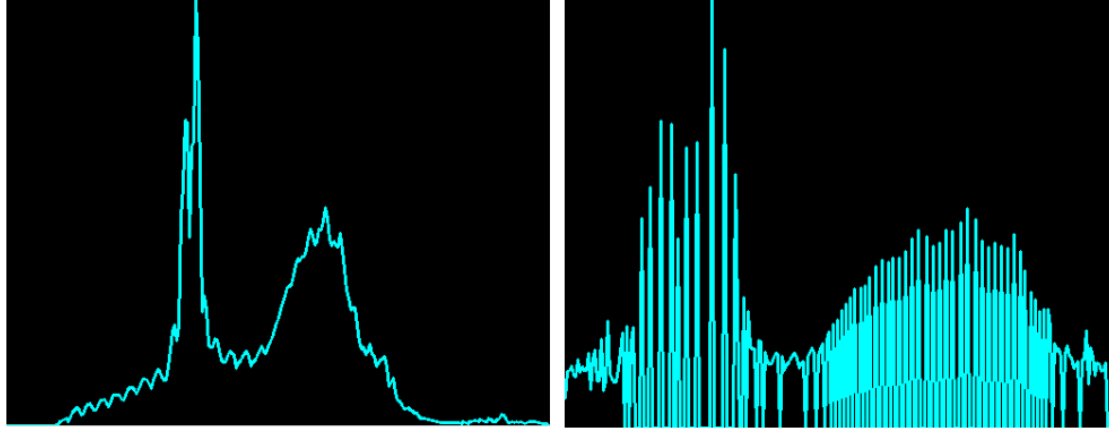
Después de esta última restricción, ya solo tenemos dos conjuntos (el conjunto  $\mathbf{R}$ ,  $t$  y  $n$ , y el conjunto  $\mathbf{R}$ ,  $-t$  y  $-n$ ). Para saber cual de los dos conjuntos es el correcto, hay que comparar el cambio de posición de un punto (de una imagen a otra) con el signo de  $t_x$  y  $t_y$ . Por ejemplo: si  $t_x$  es positivo, la componente  $p_{1x}$  tiene que ser mayor a  $p_{2x}$  por el movimiento y las coordenadas de la cámara.

### 3.5. Ecuación del histograma

Como se enseña en [33], el contraste de la imagen puede mejorar si se realiza una ecualización de su histograma. El funcionamiento de los algoritmos estudiados en la sección 3.6 (detectores y descriptores de características) se basa en la comparación local de la intensidad de los píxeles. Por lo tanto, si mejora el contraste, el resultado de la extracción de características de la imagen también mejorará, originando un emparejado entre los puntos de ambas imágenes (matching) más preciso.

La ecualización del histograma, implica transformar este histograma en otro con una distribución más uniforme, de tal manera que el número de píxeles de la imagen esté repartido a lo largo de todo el rango de intensidades (en este proyecto se trabajará con la imagen en escala de grises de 0 a 255). En la Figura 3.4 se puede observar a la izquierda el resultado del histograma en una de las imágenes aéreas estudiada en los experimentos (imágenes del Lago) y a la derecha, el resultado del histograma después de ser ecualizado.

El primer paso para obtener el histograma ecualizado, es calcular la función de distribución acumulativa de  $H(i)$ , obteniendo  $H'(i)$  de la ecuación 3.24.



**Figura 3.4:** Dibujo del Histograma antes y después de ser Ecualizado (eje horizontal normalizado de 0 a 255 y eje vertical normalizado de 0 al número máximo de píxeles)

$$H'(i) = \sum_{0 \leq j < i} H(j) \quad (3.24)$$

Si el histograma fuese ideal (una línea horizontal), el resultado de la distribución acumulativa sería una línea recta con cierta pendiente de la forma 3.25.

$$H(i') = (i' + 1) \frac{N^{\circ} \text{ de píxeles}}{256} \quad (3.25)$$

Si se igualan ambas distribuciones (calculada e ideal) y se despeja  $i'$ , se obtiene el nuevo valor de intensidad para cada nivel de gris como se muestra en la Ecuación 3.26. Por último, se crea una *Look-Up Table* (tabla de consulta donde se asocia cada  $i$  con la  $i'$  correspondiente) para que el algoritmo sea más eficiente computacionalmente.

$$i' = \text{parte\_entera}(H(i) \cdot \frac{256}{N^{\circ} \text{ de píxeles}}) \quad (3.26)$$

Por último, se sustituye cada píxel de la imagen con una intensidad  $i$  (que puede ir de 0 a 255) por el píxel  $i'$  correspondiente, utilizando para ello la *Look-Up Table* creada. En las imágenes de la Figura 3.5 (arriba una imagen del Lago en escala de grises y abajo esa misma imagen pero después de ecualizar su histograma), se puede observar como el uso esta técnica mejora considerablemente el contraste de la imagen.



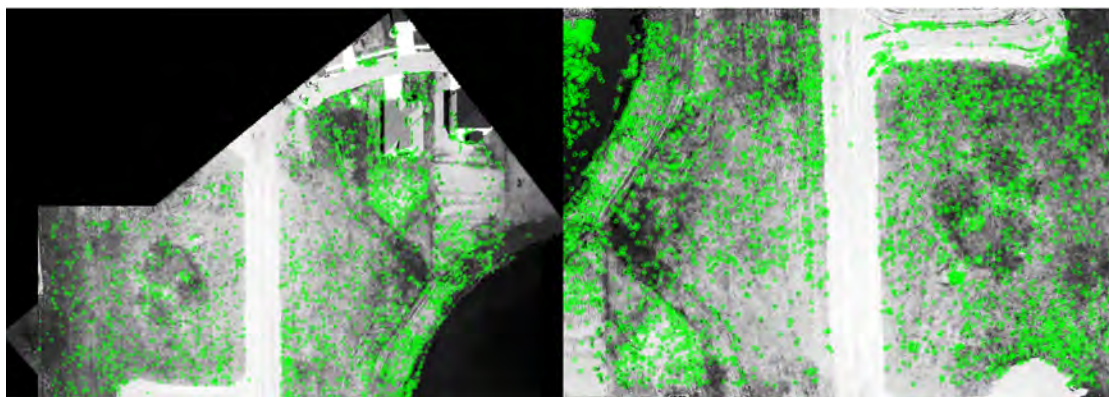
Figura 3.5: Ecualización del Histograma usando el dataset del Lago como ejemplo

### 3.6. Extracción de características en las Imágenes

Como ya se mencionó antes, en este proyecto el *Matching* entre las imágenes se realizará a partir de las características que comparten estas. En este apartado, se detectarán, a través de un algoritmo, las características más significativas de cada una de las imágenes. Los puntos donde se localizan estas características serán denominados como *Keypoints*. Además, se utilizará otro algoritmo para extraer información de los *Keypoints* y de los píxeles vecinos que se encuentren a un radio de estos. La información extraída se guardará en una matriz a la cual se le llamará

Matriz de Descriptores, donde cada una de las filas de esta matriz hace referencia a la información extraída de cada uno de los Keypoints.

Es necesario que el algoritmo trabaje a tiempo real para cumplir con uno de los objetivos propuestos y la detección de los Keypoints es la tarea que mayor coste computacional supone, por lo tanto se elegirá un algoritmo que pueda cumplir con el objetivo mencionado. El algoritmo elegido para la localización de Keypoints es AGAST (Adaptive and Generic AST) [34]. AGAST es un algoritmo de alta velocidad para la detección de corners basada en la intensidad de los puntos estudiados y sus vecinos. La velocidad de procesamiento de AGAST es bastante mayor comparada con la velocidad de otros detectores de características, pero tiene la desventaja de poseer baja repetibilidad al hacer zoom o cambiar el punto de vista de la imagen. Esto último se puede observar en el estudio hecho por A. Canclini y M. Cesana en [35], donde comparan AGAST con FAST, BRISK, CenSurE, ORB, SURF y SIFT. También, hay que tener en cuenta, que este algoritmo trabaja mejor en situaciones donde hay una cantidad suficiente de corners y que otros algoritmos trabajarían mejor en situaciones donde hay muchos cambios de contraste. En la Figura 3.6, se pueden observar en color verde los Keypoints obtenidos en una región de la panorámica (la última actualizada) y en la nueva imagen candidata a formar parte de la panorámica.



**Figura 3.6:** Keypoints encontrados en la region de interes de la panorámica y en la nueva imagen adquirida

El algoritmo utilizado para obtener la Matriz de Descriptores es FREAK [36], un “descriptor” binario que devuelve una fila de unos y ceros por cada descriptor (fila de la matriz). Esta basado en el funcionamiento de la retina y compara la intensidad de los píxeles vecinos, utilizando para esto círculos que se distribuyen con mayor intensidad en las zonas cercanas al pixel estudiado. Según [37], que recomienda el uso de BRISK como descriptor, FREAK es más rápido que este pero más eficiente computacionalmente hablando. Por otro lado, [38] menciona

que FREAK es mejor que BRISK y ORB cuando existen cambios en el punto de vista de la imagen. Por estas razones, se ha elegido este algoritmo entre los otros propuestos en los trabajos mencionados.

### 3.7. *Matching* de las imágenes

Los Keypoints y sus correspondientes descriptores extraídos en la sección 3.6, serán utilizados para emparejar puntos relacionados de la zona de superposición existente entre la panorámica y la imagen capturada. Por cada descriptor de cada imagen, se buscará el mejor descriptor de la otra imagen que mejor encaje. Para esto último, se utilizará un método basado en la búsqueda de los vecinos más cercanos (KNN) implementado en FLANN (Fast Library for Approximate Nearest Neighbors). FLANN es una librería con muchos algoritmos de búsqueda de los vecinos más cercanos, más rápidos que realizar una búsqueda exhaustiva utilizando KNN, pero que a costa de esto pierden algo de precisión. Para este proyecto, se ha utilizado el algoritmo KD-tree debido a la investigación realizada por Muja y Lowe [39].

KD-tree es un método basado en un árbol binario en el cual, cada nodo divide un subconjunto de datos en otros subconjuntos más pequeños. El conjunto de descriptores de la primera imagen (denominados datos de entrenamiento), crean esta estructura tipo árbol. Entonces, empezando por la raíz del árbol, se va descendiendo y seleccionando a que nodo pertenece un descriptor de la otra imagen (denominados datos de consulta). Cuando se llega al último nodo, este es seleccionado como el mejor candidato para el detector que esta consultando.

El Matching de descriptores, ayuda a eliminar gran cantidad de Keypoints teniendo en cuenta que no todos se encuentran en el área de superposición entre las imágenes, pero no filtra todos los puntos de tal manera que se pueda calcular una matriz de Homografía correcta con ellos. Por esta razón, se utiliza el método RANSAC para obtener los Keypoints que mejor se ajustan a una matriz de Homografía (Inliers), a partir de los Good Matches obtenidos al utilizar KD-tree. En la sección 3.4.2, se estudia como se usa RANSAC para obtener esta matriz y los Inliers. En la Figura 3.7, se puede observar los descriptores emparejados después de utilizar el método descrito en esta sección, y en la Figura 3.8 los puntos que han quedado como resultado al utilizar RANSAC.



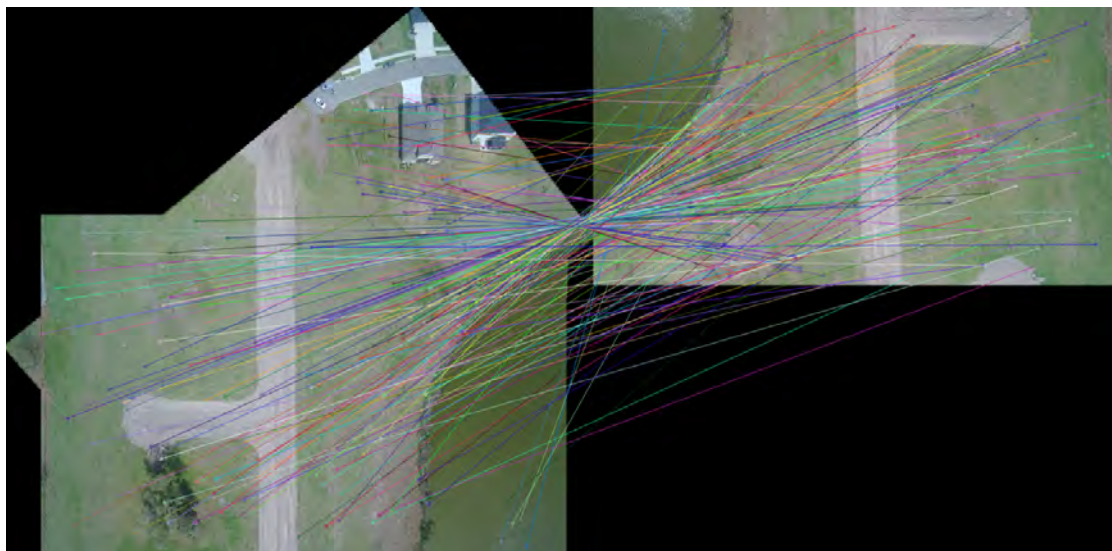


Figura 3.7: Good Fetures encontrados despúes del Matching de los descriptores

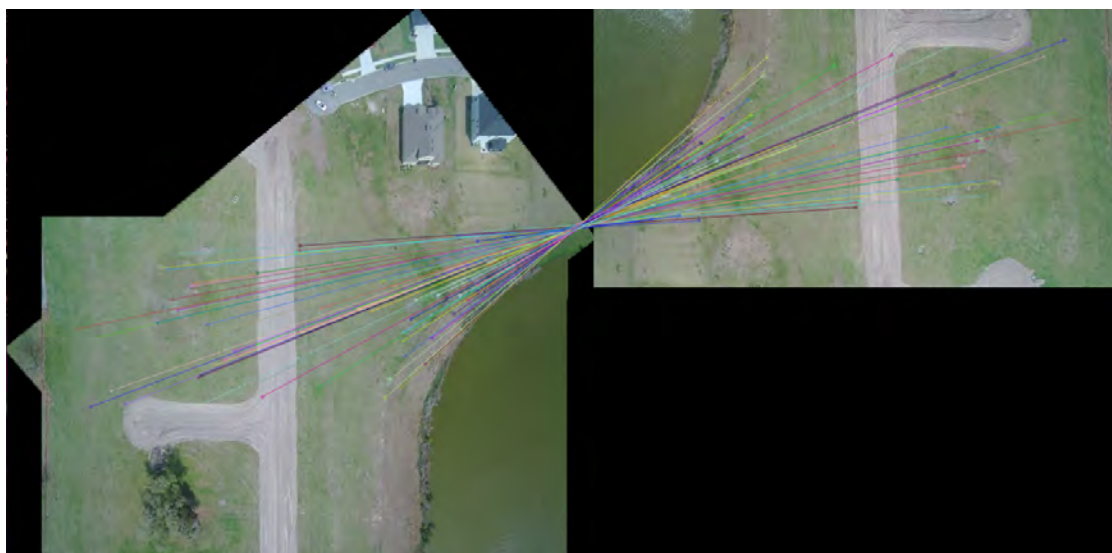


Figura 3.8: Inliers encontrados despúes de utilizar RANSAC

### 3.8. Búsqueda de “costuras” entre las imagenes

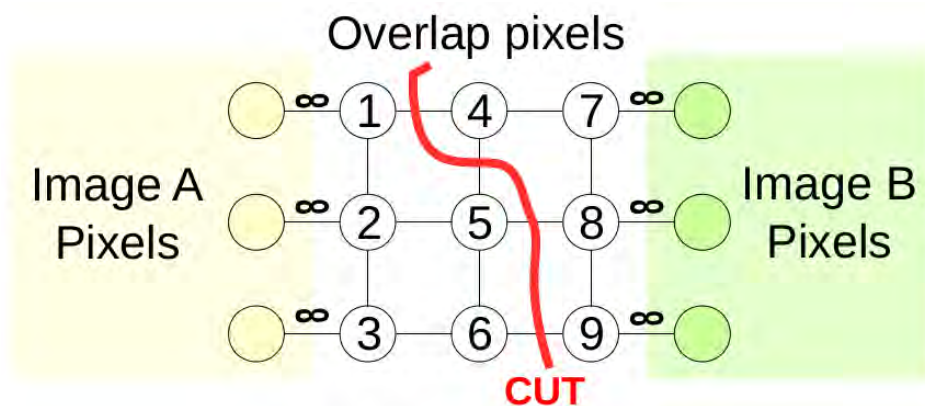
En este punto, se calcula la línea de “costura” entre la panorámica y la nueva imagen, es decir, en este punto se obtiene una línea de corte en cada imagen que separa los píxeles que se van a incluir en la panorámica resultante de los que no. Para calcular esta costura, se ha utilizado el método basado en recorte de grafos propuesto por Vivek Kwatra y Arno Schödl [40].

Este método, define cada píxel en la región de superposición como un nodo. También, define la primera y la segunda imagen como la región A y B, respectivamente. Por último, conecta los píxeles adyacentes mediante una línea que denomina arco, para después asignar una etiqueta a cada uno de los arcos.

El valor de las etiqueta se calcula a partir de la Ecuación 3.27 denominada por los creadores del método como *matching quality cost*. En dicha ecuación,  $s$  y  $t$  se refieren a la posición de dos nodos adyacentes, mientras que  $A$  y  $B$  son los colores de los píxeles a los que apuntan en las diferentes imágenes. Por ejemplo,  $B(s)$  es el color del pixel  $s$  en la imagen  $B$ . Los arcos conectados a píxeles que se encuentran fuera de la región de superposición, son etiquetados con un valor infinito.

$$M = \|A(s) - B(s)\| + \|A(t) - B(t)\| \quad (3.27)$$

Finalmente, el problema de recorte de grafos se resuelve dibujando un camino que divide las regiones  $A$  y  $B$ . Dicho camino, recorrerá la región de superposición pasando por los arcos que tengan una etiqueta con un valor menor que los demás. Esto se puede observar en la Figura 3.9, donde la línea roja es la que indica que píxeles son los que aportará cada imagen. Los píxeles que se encuentren a la izquierda de dicha línea, son los que aportará la imagen  $A$ ; mientras que los píxeles que se encuentren a la derecha, serán los que porte la imagen  $B$ .



**Figura 3.9:** Camino encontrado que determina la línea de “costura” entre las dos imágenes

### 3.9. Unión de las imágenes usando *Feather Blending*

Una vez elegidos los píxeles que aportará cada imagen, se une la imagen panorámica con la nueva imagen, realizando dicha unión por la zona de corte calculada en el apartado anterior. Para posicionar ambas imágenes en el plano 2D que conformará la nueva panorámica, se utiliza como referencia la coordenada del corner superior izquierdo de ambas imágenes. Además, para que la transición

en los puntos de costura sea suave, se utiliza el método Feather Blending para la unión de dichas imágenes.

Entre los métodos de Blending estudiados por Zhe Zhu y Jiaming Lu [41], para este proyecto, se estudiaron dos: *Multi-band Blending* y *Feather Blending*.

Feather Blending, es un método que simplemente combina linealmente las dos imágenes usando las Formulas 3.28,3.29,3.30. Donde,  $\alpha$  es un valor que se multiplica por la intensidad de las componentes RGB de cada pixel. Este valor, va de 0 a 1, siendo  $\alpha_c$  el valor de  $\alpha$  en la costura de la imagen. A medida que los píxeles se alejan localmente de la costura de la imagen hacia el interior de esta, el valor de  $\alpha$  va creciendo hasta se vale 1. En sentido contrario, a medida que los píxeles se alejan de la costura hacia fuera de la imagen,  $\alpha$  va decreciendo hasta valer 0. La Figura 3.10, muestra como cambia  $\alpha$  en la zona local del conjunto de puntos que forman la linea de costura.

$$I(x, y)_{izq} = (\alpha \cdot R, \alpha \cdot G, \alpha \cdot B) \quad (3.28)$$

$$I(x, y)_{drch} = ((1 - \alpha) \cdot R, (1 - \alpha) \cdot G, (1 - \alpha) \cdot B) \quad (3.29)$$

$$I_{blend} = I(x, y)_{izq} + I(x, y)_{drch} \quad (3.30)$$

Para el uso del método Multi-band Blending, primero se computa una Pirámide Gaussiana en cada una de las imágenes (hay que especificar el número de niveles que se quiera utilizar). En segundo lugar, a partir de las pirámides Gaussianas, se calculan las pirámides Laplacianas de cada una de las imágenes. Después, se une cada nivel de la pirámide Laplaciana de una imagen con el nivel correspondiente de la otra. Por último, a partir de todas las imágenes unidas (en todos los niveles) se reconstruye la nueva imagen panorámica. Este método, funciona muy bien sin buscar costuras en las imágenes, y además no produce ghosting.

En este proyecto, se ha propuesto el uso de Feather Blending, ya que la transformación de la intensidad de los píxeles solo se realiza localmente alrededor de las costuras. Por otro lado, métodos como Multi-band blending hacen uso de pirámides gaussianas que se aplican a lo largo de toda la panorámica una y otra vez en cada iteración, provocando que a medida que crece la panorámica esta se emborrone.

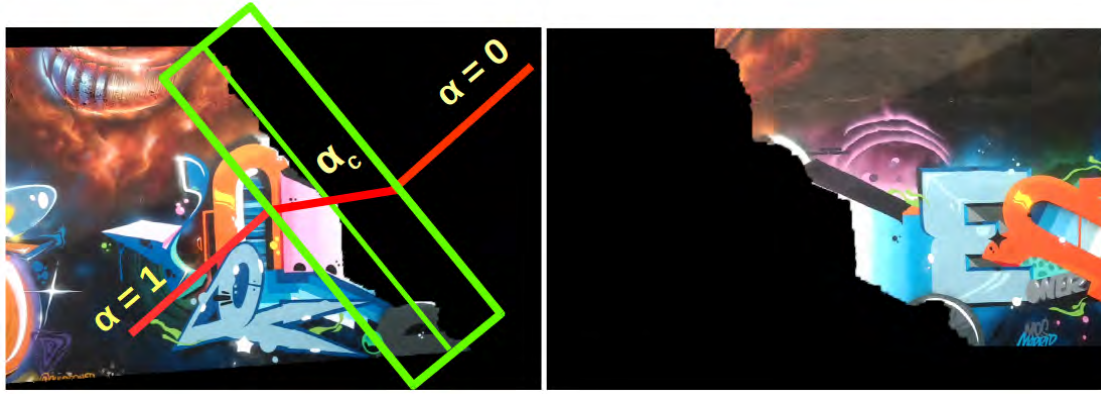


Figura 3.10: Valor de  $\alpha$  a lo largo de la imagen al usar Feather Blending

### 3.10. Selección de la mejor imagen para realizar el Blending

El método para encontrar las Líneas de Costura en las imágenes y el Feather Blending, requieren mucho coste computacional (en los resultados se puede observar el tiempo de ejecución en los dos puntos mencionados). Por lo tanto, no es posible unir las imágenes a la panorámica a la velocidad a la que se capturan. Además, no es conveniente que la panorámica final contenga mucha densidad de imágenes por el siguiente motivo:

- El método Feather Blending, como se mencionó arriba, emborrona el área alrededor de la costura para que no se note la transición de una imagen a otra.
- Si la panorámica resultante, esta compuesta por muchas imágenes, también lo estará por muchas costuras. Esto último ocasionará que la imagen panorámica pierda definición.

El caso mencionado arriba, ocurre cuando el porcentaje de superposición entre las imágenes es elevado. El caso contrario, en el que el tanto por ciento de superposición es bajo, podría originar que no se encuentre un número suficiente de características para obtener una matriz de Homografía correcta.

Para este proyecto, se ha creado un método que filtra las imágenes utilizando un umbral del tanto por ciento de superposición entre la imagen y la panorámica. Además, este método selecciona la mejor imagen observando cuánto ha cambiado el tamaño de cada una de ellas después de ser proyectada (la selección de la mejor imagen es llevada a cabo entre las imágenes que se encuentren dentro del umbral).

Como se observará en el apartado de los experimentos, es aconsejable cambiar el umbral según la aplicación y de esta manera llevar la reconstrucción de la panorámica a su punto más óptimo.

### 3.10.1. Obtención del Porcentaje de Superposición entre imágenes

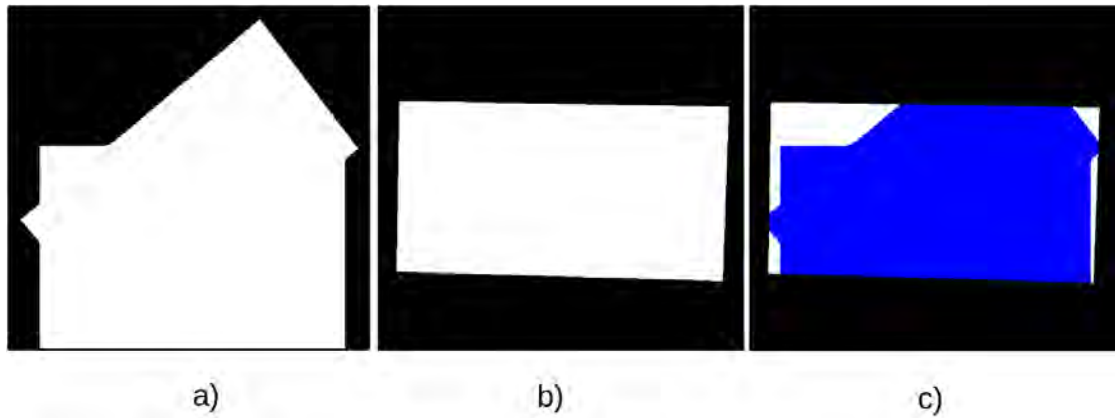
Cada vez que se captura una nueva imagen, como se ha comentado en los puntos anteriores, se buscan las características en común entre esa nueva imagen y la última zona actualizada de la panorámica. Después, se obtiene la Matriz de Homografía para posteriormente calcular la matriz de rotación que proyecta la nueva imagen.

En este punto del algoritmo, se proyecta la última Imagen capturada y se coloca (gracias a los Inliers) en el lugar correspondiente de la panorámica. En este momento, se tiene que ambas imágenes contienen píxeles que comparten la misma posición en el espacio. Esto último se puede observar en la Figura 3.12. Para determinar cual es el área de la región que comprende los píxeles que comparten ambas imágenes:

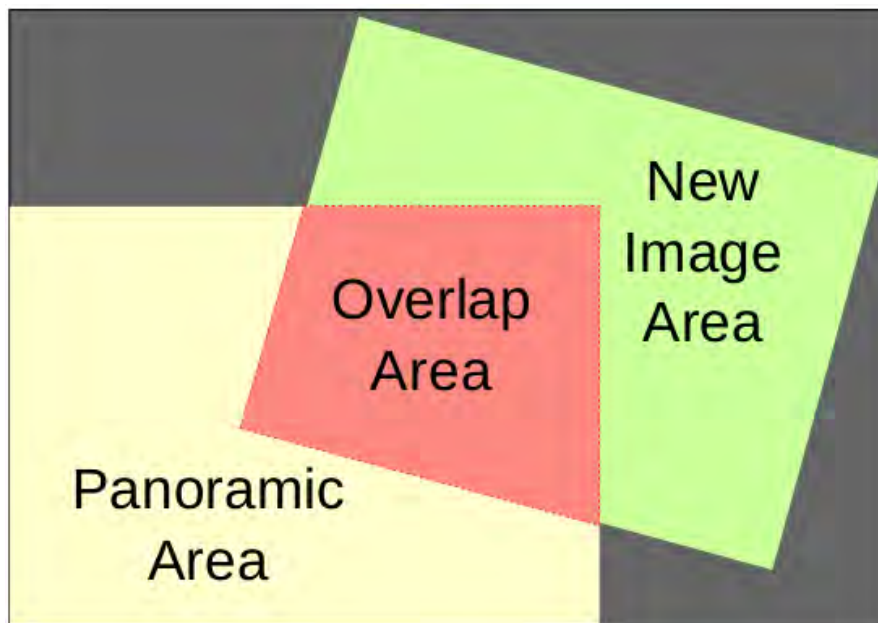
- Primero, se crean dos máscaras (dos imágenes binarias del mismo tamaño y posición que las originales). Estas máscaras, tienen siempre un valor en la escala de grises de 255. En la Figura 3.11, a) corresponde a la máscara de la panorámica, mientras que b) a la de la nueva imagen capturada.
- Después, se realiza una operación lógica AND, dando como resultado una tercera máscara donde todos los píxeles blancos, son píxeles que comparten tanto la máscara de la panorámica como la de la imagen proyectada. En la Figura 3.11.b) se pinta en azul el resultado de la operación AND encima de la segunda máscara.
- A continuación, utilizando el algoritmo [42] se obtiene un conjunto de puntos que define el contorno del área superpuesta.
- Con el conjunto de puntos obtenido, se calcula el área de superposición utilizando un método basado en el teorema de *Green* [43].

Para calcular el área de la imagen proyectada, se utiliza la máscara que hace referencia a esta imagen y se determina su área a partir de los dos métodos descritos en los dos últimos puntos de arriba.

A partir de la Fórmula 3.31 se obtiene el tanto por ciento de superposición. Como la imagen panorámica crece a la vez que se van añadiendo nuevas imágenes



**Figura 3.11:** Cálculo del área de superposición a partir del uso de máscaras. a) se corresponde con la máscara de la panorámica y b) con la de la nueva imagen capturada. El área de superposición se muestra en azul en c).



**Figura 3.12:** Área de superposición entre dos imágenes

a esta, en el divisor de la Ecuación 3.12 se debe colocar el área de la imagen proyectada. Si se utilizara el área de la imagen panorámica, a medida que creciese esta el valor del tanto por ciento iría disminuyendo hasta acercarse a cero.

$$\%Superposición = \frac{\text{Área}_{Superposición}}{\text{Área}_{Img.Proyectada}} \cdot 100 \quad (3.31)$$

### 3.10.2. Selección de la mejor imagen

Después de filtrar las imágenes usando el umbral del tanto por ciento de superposición, hay que determinar que imagen será añadida a la panorámica. Para determinar que imagen será la mejor candidata, primero se compara el área de cada una de estas imágenes con el área de la misma imagen pero proyectada (como se muestra en la Formula 3.32).

Después de la comparación, se elige como la mejor imagen la que de como resultado el valor absoluto más pequeño. Este método se basa en que se puede interpretar que la cámara se ha mantenido lo máximo posible dentro del plano de la primera captura, si no hay mucha diferencia entre el área original y el área de la imagen proyectada.

$$\Delta \text{Área} = | \text{Área}_{\text{Img.Capturada}} - \text{Área}_{\text{Img.Proyectada}} | \quad (3.32)$$

## 3.11. Creación de una nueva panorámica

Debido a que el algoritmo de este proyecto se basa en que la cámara del UAV debe de estar lo más paralela posible al plano, se ha diseñado un método que obliga al algoritmo a crear una panorámica adicional en el caso de que esto no se cumpla. El número de panorámicas que se cree, serán las necesarias hasta que la reconstrucción acabe.

Para empezar con la creación de una nueva imagen panorámica, hay que determinar que factores indica que es el momento preciso de hacerlo. En este proyecto, se realiza una serie de evaluaciones al lo largo del algoritmo para cada captura.

La primera evaluación, que se debe hacer y que descartaría la imagen que se esta evaluando, es comprobar que se han encontrado suficientes Good Features para poder estimar la matrix de Homografía. Una vez se ha calculado la matriz de Homografía, se realiza otra evaluación, poco compleja, de cuanto cambia el área de la imagen si esta es proyectada. Para esto último, se siguen los siguientes pasos:

- Se cogen los cuatro puntos correspondientes a las cuatro esquinas de la imagen original.
- A continuación, se multiplica cada uno de los puntos por la matriz de Homografía obteniendo las cuatro esquinas de la imagen proyectada.

- Se divide el área del polígono que resulta de unir las cuatro esquinas obtenidas en el apartado anterior (las de la imagen proyectada), con el área del polígono que se obtiene al unir las cuatro esquinas de la imagen original.
- Si el resultado de esa división resulta que es mayor a 2 o menor a 0,5, se descarta la imagen que se está evaluando.

El siguiente paso es proyectar la imagen y evaluar el tanto por ciento de superposición que existe entre ella y la panorámica. Si este porcentaje es menor al 30 % no se garantizaría que los Features encontrados estén bien distribuidos y que, por lo tanto, la matriz de transformación sea correcta. Como las dos anteriores, esta evaluación también descartaría la imagen actual que se está verificando.

Si dos imágenes capturadas consecutivas no cumplen, cada una, con alguna de las evaluaciones descritas arriba, se guarda la imagen panorámica actual y se empieza a reconstruir una nueva. Un diagrama de este apartado se puede ver en la Figura 3.13.

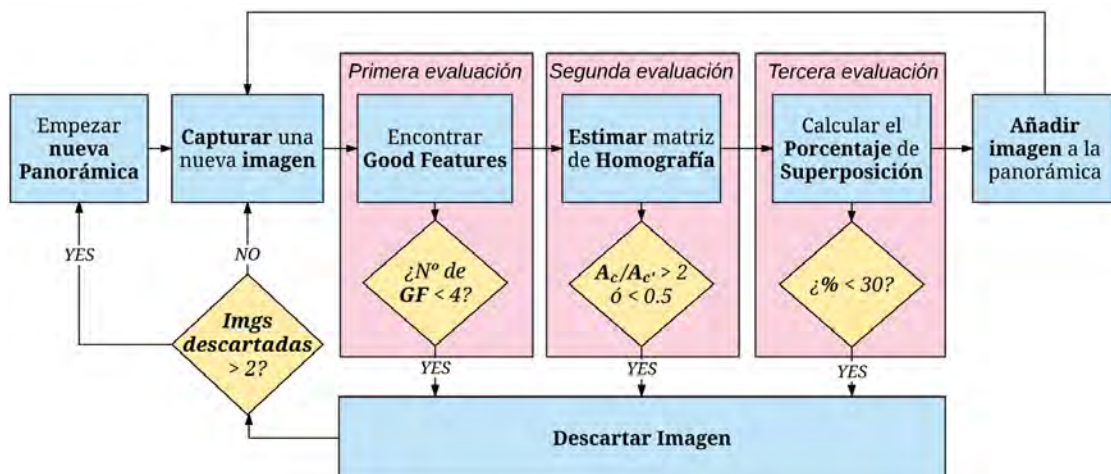


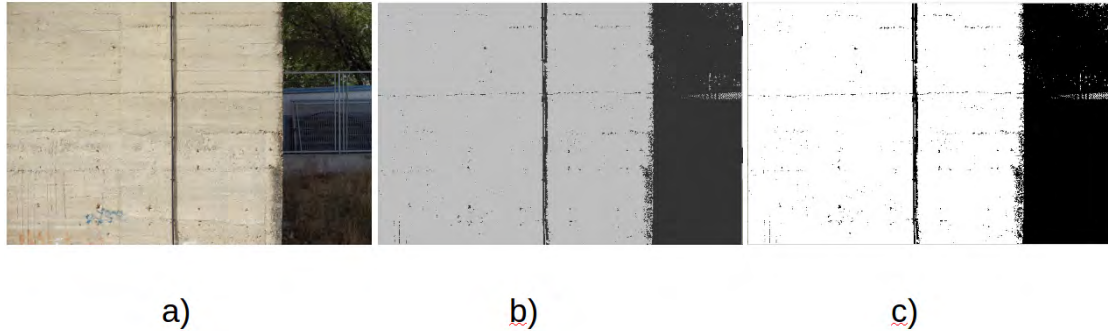
Figura 3.13: Bucle de evaluación de las nuevas imágenes capturadas

### 3.12. Uso de *Clustering* para segmentar fachadas

Debido a que el algoritmo propuesto en este proyecto tiene en cuenta que el objeto que se está observando es totalmente plano, puede surgir ciertos problemas si se quiere realizar la reconstrucción de una fachada, ya que esta contiene ventanas. También, el algoritmo puede coger características del fondo del escenario (por ejemplo en las esquinas de los edificios) llegando a ocasionar que se incumpla la



asunción mencionada. Por estas razones, se ha introducido un algoritmo basado en *K-Means* para crear una máscara que solo permita buscar características en el frontal de la fachada, como se muestra en la Figura 3.14. Una vez segmentada la fachada, se aplica una dilatación de la máscara para no perder detalles en las esquinas que puedan servir para obtener características.



**Figura 3.14:** Uso de clustering para segmentar la fachada. Imagen original de la fachada a), grupos después de usar K-Means b) e imagen binaria después de usar un threshold c).

K-Means es un algoritmo que detecta grupos en un conjunto de píxeles dado. Esto lo hace sin necesidad de supervisar o corregir los resultados de la agrupación [44].

# **CAPÍTULO 4: EXPERIMENTOS Y RESULTADOS**

En esta sección se realizará una serie de experimentos para poner a prueba el algoritmo propuesto. Se evaluará tanto la velocidad del método como la calidad de los resultados. También, se modificarán ciertos parámetros, como los umbrales de superposición o los factores de escala para ver como estos afectan al resultado.

Para tales propósitos, se utilizarán cuatro *datasets* diferentes. Dos de estos datasets están compuestos por imágenes sueltas con un porcentaje de superposición ya definido. Los dos datasets restantes son vídeos que ayudarán a comprobar el funcionamiento del algoritmo en tiempo real. Más información sobre estos datasets, y de como serán denominados en este proyecto, se puede ver a continuación:

- **Imágenes Urbanas 1.** El primer dataset, consiste en imágenes tomadas a 780 metros sobre un área urbana. Las imágenes fueron capturadas por un UAV que volaba en línea recta, consiguiendo un porcentaje de superposición entre imágenes del 60 % y cada imagen tiene un tamaño de  $1000 \times 642$  píxeles.
- **Imágenes Urbanas 2.** El segundo dataset, consiste también en imágenes aéreas pero con un tamaño de  $1500 \times 1001$  píxeles. Este junto al primer dataset se ha obtenido de [45].
- **Vídeo Mural.** El siguiente dataset, es un vídeo de un mural tomado con un Smart-phone generando imágenes de  $1920 \times 1080$  píxeles a 30 fps. Con este vídeo se podrá comprobar los resultados en estructuras verticales.
- **Vídeo Lago.** El último dataset, es un vídeo de los alrededores de un lago tomado con un UAV que mantiene una ruta aleatoria y genera imágenes de  $1280 \times 676$ .

Los experimentos se han llevado a cabo en un estación de tierra (el algoritmo no ha sido ejecutado en pleno vuelo), pero una configuración del Hardware del UAV para el propósito de este proyecto es propuesta en la sección 6.18. La unidad de procesamiento utilizada en los experimentos está compuesta por:

- Una CPU Intel i5-2410M a 2.3 GHz
- Una memoria RAM de 4GB
- Una Tarjeta Gráfica NVIDIA GEFORCE GT520M de 512 mb.

Para comprobar el funcionamiento en tiempo real, los frames de los datasets *Vídeo Lago* o *Vídeo Mural* son leídos en un hilo de procesamiento, como se haría con las imágenes de la cámara del UAV. Mientras las imágenes del vídeo van avanzando, en otro hilo está corriendo el algoritmo de reconstrucción. De esta

manera, si por ejemplo, han pasado 10 frames del vídeo mientras el programa esta ejecutando la unión de una imagen a la panorámica, estos frames se perderán, cogiendo el programa el frame número 11 cuando le corresponda coger una nueva imagen.

En el primer experimento, se estudiará como cambia el tiempo de ejecución en diferentes partes del algoritmo según se van uniendo cada vez más imágenes a la panorámica. Además

En la tabla 4.2, se muestra un resumen de los datasets utilizados en cada uno de los experimentos así como los parámetros que se modificarán o se medirán.

Experimento	Datasets	Parámetros	Datos estudiados
1	Conjunto de imágenes Urbanas 1 y 2.	Número de imágenes de la panorámica.	Tiempo de ejecución y número de Inliers.
2	Conjunto de imágenes Urbanas 2	Factores de escala.	Tiempo de ejecución y calidad de la panorámica resultante.
3	Vídeo del Mural y del Lago	Umbral del Porcentaje de Superposición entre imágenes.	Tiempo de ejecución, número de inliers, tamaño de la panorámica resultante y errores producidos.

**Tabla 4.2:** Resumen de los experimentos realizados

A partir de este punto, para resumir las gráficas y las tablas donde se quiera estudiar el tiempo de ejecución, se agruparán los diferentes pasos del algoritmo en cinco partes:

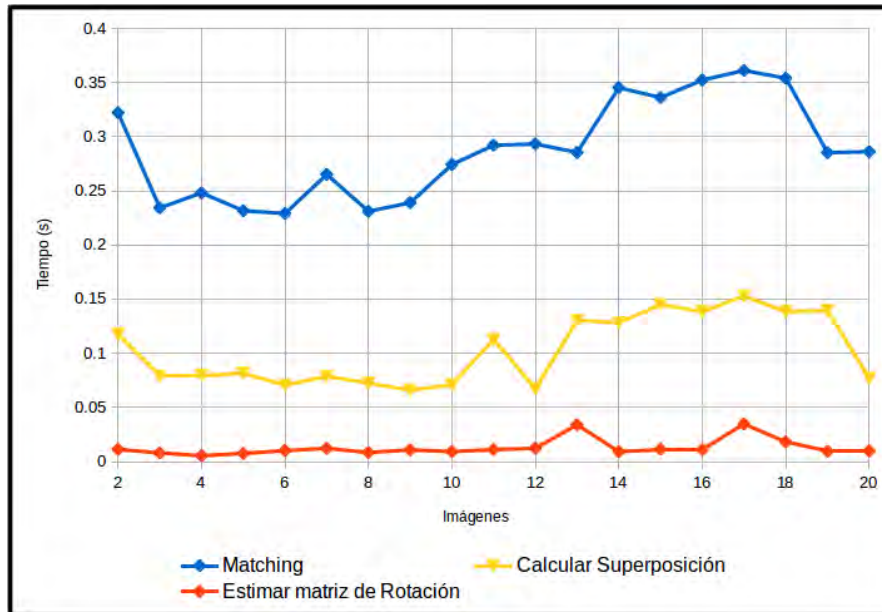
- El Matching: que agrupa desde la captura de una nueva imagen, hasta la obtención de las Good Features.
- La estimación de la matriz de Rotación: que contiene los pasos del cálculo de la matriz de Homografía, de la descomposición de esta y de la evaluación de dicha matriz.
- El cálculo del porcentaje de Superposición: que abarca la deformación de las imágenes, el cálculo del tanto por ciento de superposición y de la evaluación de este porcentaje.
- La búsqueda de las Costuras de las imágenes.
- La Unión de la imagen captura con la imagen panorámica.

### 4.13. Resultado de los tiempos y el número de *inliers* según el número de imágenes

Para este experimento se han construido dos panorámicas: una a partir de 20 imágenes (ver Figura II.25) del dataset *Imágenes Urbanas 2* y otra a partir de 12 imágenes del dataset *Imágenes Urbanas 1*, que se puede ver en la Figura II.24. El resultado de estas panorámicas se pueden ver en las Figuras II.26 y II.27 del Anexo II, respectivamente. El objetivo de este experimento, es estudiar como cambia el tiempo de ejecución en cada una de las partes del algoritmo, descritas en la introducción de capítulo y como cambia el número de *Inliers*, a medida que va creciendo la imagen panorámica. Para esto último, se han observado dos ejemplos diferentes, que son las dos panorámicas construidas con los datasets mencionados antes.

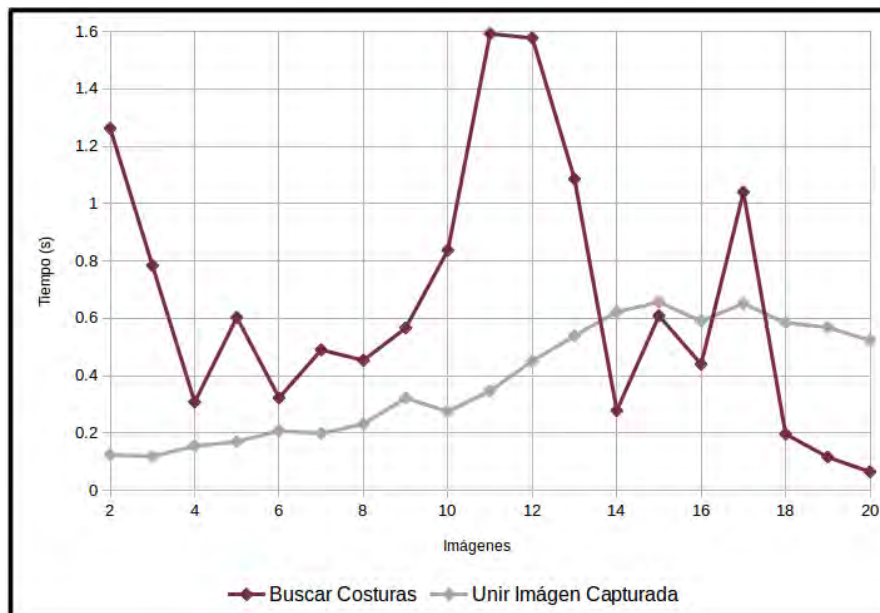
En este experimento, la reconstrucción de las panorámicas se han llevado a cabo con un factor de escala de 0.4 (en el dataset *Imágenes Urbanas 2*) para los pasos: búsqueda de las Costuras de las imágenes y el cálculo del porcentaje de Superposición. Mientras que, los pasos restantes se han realizado con las imágenes con resolución original. De igual manera, se ha procedido con el dataset *Imágenes Urbanas 1*), pero con un factor de escala de 0.6. Escalar la imagen en estos pasos, ayuda a que la ejecución del programa sea más fluida. Esto último, se verá en la sección 5.16.

En la gráfica 4.15 (donde se estudia como cambia los tiempos en función de las tres primeras partes del algoritmo) se puede observar como, aproximadamente desde que la panorámica contiene 14 imágenes hasta las 18 imágenes, hay un ligero aumento en los tres tiempos. También, se puede ver como estos tres tiempos decaen en la adición de las dos últimas imágenes a la panorámica. Por último, se puede observar que el tiempo en la unión de las dos primeras imágenes es alto en las líneas del Matching y de la Superposición. Esto último puede ser debido, en el caso del Matching, al gran número de características que tiene que encontrar y emparejar en la primera unión.



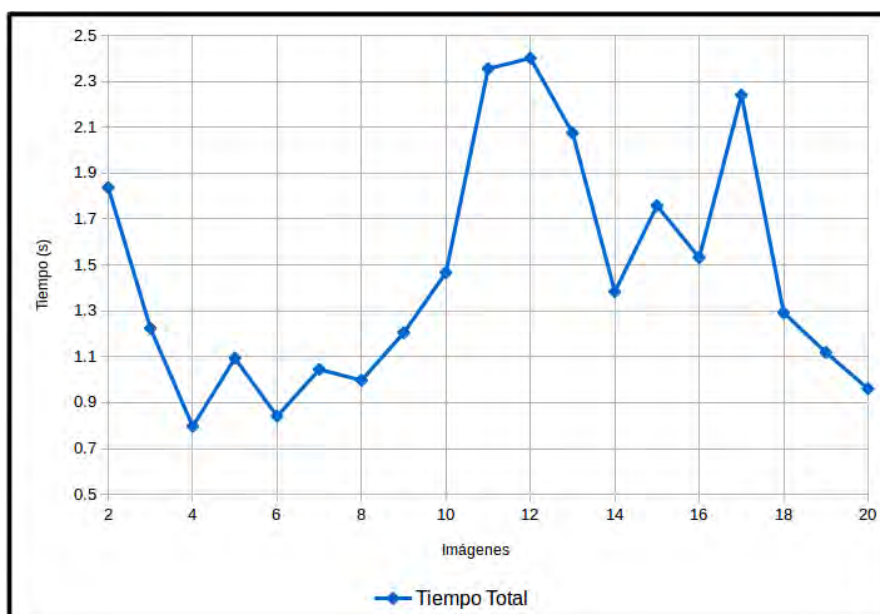
**Figura 4.15:** Gráfica nº Imágenes - Tiempo en segundos (Matching, Matriz de rotación y Superposición), del dataset *Imágenes Urbanas 2*

En la gráfica 4.16, se puede ver claramente como le cuesta cada vez más al algoritmo el proceso de unir la nueva imagen a la panorámica. En el caso de la búsqueda de Costuras, el tiempo no sigue ningún patrón a medida que aumentan las imágenes en la panorámica y es el proceso más lento del algoritmo. También cabe destacar, que la línea de buscar Costuras acaba muy abajo desde donde empezó. Esto último, puede ser causado por la deformación de las imágenes y como éstas van perdiendo textura a medida a la que avanza la reconstrucción de la panorámica.



**Figura 4.16:** Gráfica n° Imágenes - Tiempo en segundos (Costuras y Union de imágenes), del dataset *Imágenes Urbanas 2*

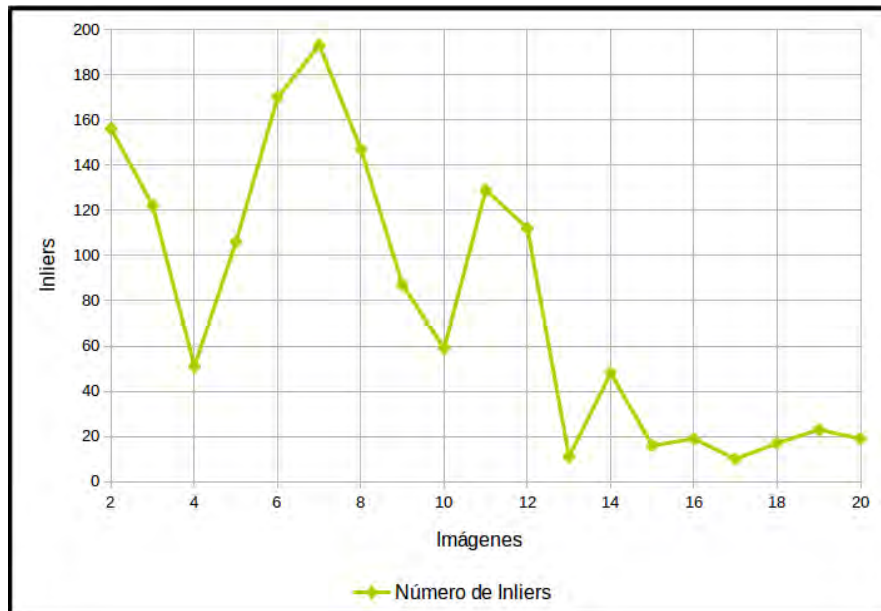
En el tiempo total (ver Gráfica 4.17), se puede observar como la línea que marca el tiempo de búsqueda de costuras es el cuello de botella del algoritmo, ya que la línea del tiempo total tiene la forma de esta (ver Gráfica 4.16) más un offset que se podría definir como la suma de los tiempos restantes.



**Figura 4.17:** Gráfica n° Imágenes - Tiempo total(s), del dataset *Imágenes Urbanas 2*

En la Gráfica 4.18, se puede observar la tendencia descendente que tiene la línea Imágenes - Inliers. También, se puede ver que el número de Inliers desciende

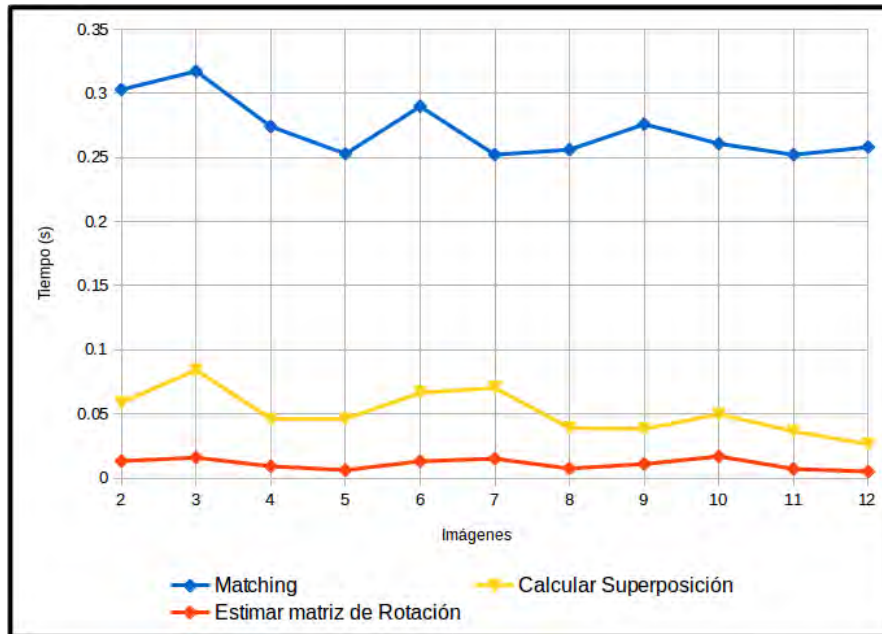
mucho (de 160 a 20), pudiéndose traducir este descenso como que al algoritmo cada vez le cuesta más encontrar característica en la imagen panorámica. Este descenso de características, puede ser producido por el error acumulado en cada iteración que va escalando las imágenes haciendo que se deformen demasiado.



**Figura 4.18:** Gráfica nº Imágenes - Inliers, del dataset *Imágenes Urbanas 2*

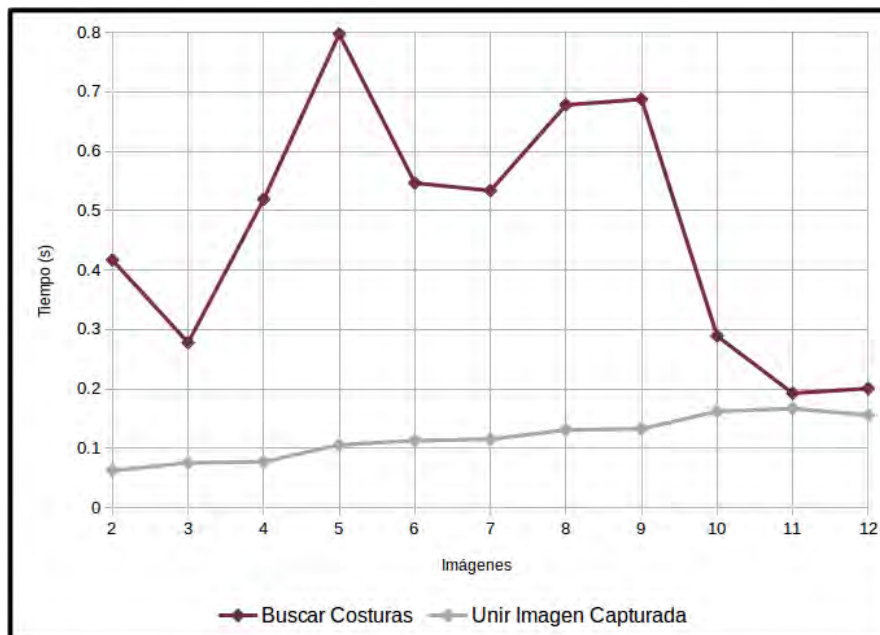
A continuación, se intentará contrastar las conclusiones con otro ejemplo (con el otro dataset). En la Gráfica 4.19, la única similitud que encontramos con la Gráfica 4.15, es el descenso de los tiempos en las últimas imágenes. Este descenso puede ser provocado por la disminución de características encontradas en la imagen panorámica.





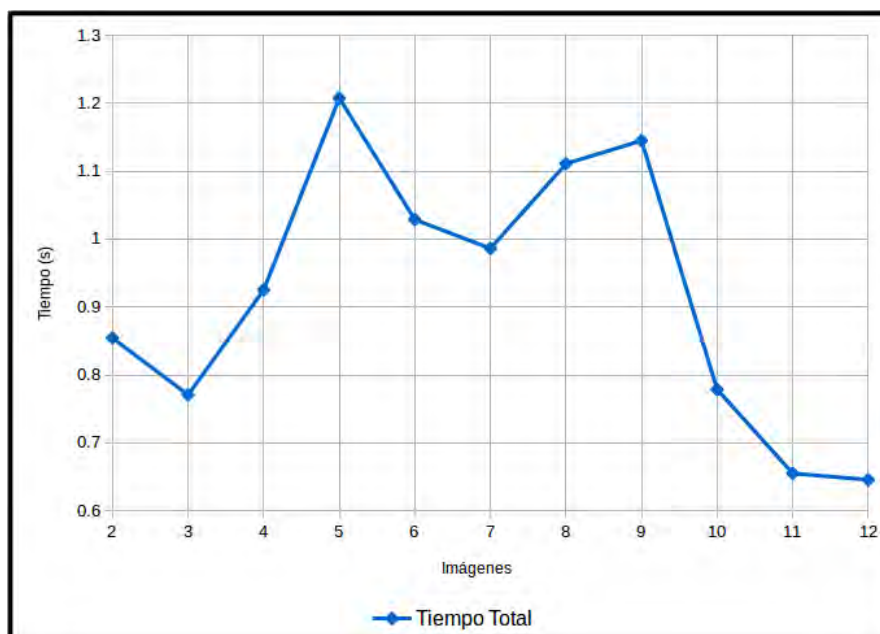
**Figura 4.19:** Gráfica nº Imágenes - Tiempo en segundos (Matching, Matriz de rotación y Superposición), del dataset *Imágenes Urbanas 1*

En la Gráfica 4.20, se puede seguir observando tres comportamientos parecidos a la Gráfica 4.16. El primero de ellos, es que el cálculo de las Costuras de la imágenes sigue siendo el proceso más lento de todos. El segundo comportamiento parecido, es que el proceso de Unir la nueva imagen capturada con la panorámica sigue una tendencia ascendente a medida que va creciendo la panorámica. Por último, se puede observar en la gráfica que el tiempo de búsqueda de Costuras acaba muy por debajo de donde empezó, igual que en el ejemplo anterior.



**Figura 4.20:** Gráfica n° Imágenes - Tiempo en segundos (Costuras y Union de imágenes), del dataset *Imágenes Urbanas 1*

En la Gráfica 4.21, se puede ver como el tiempo total sigue dependiendo mucho del tiempo de búsqueda de costuras y además del tiempo del Matching.



**Figura 4.21:** Gráfica n° Imágenes - Tiempo total(s), del dataset *Imágenes Urbanas 1*

En este experimento, también se observa que la Gráfica 4.22 n° Imágenes - Inliers, sufre un descenso a medida que las imágenes que constituyen la panorámica van aumentando, de igual manera que en el ejemplo anterior.

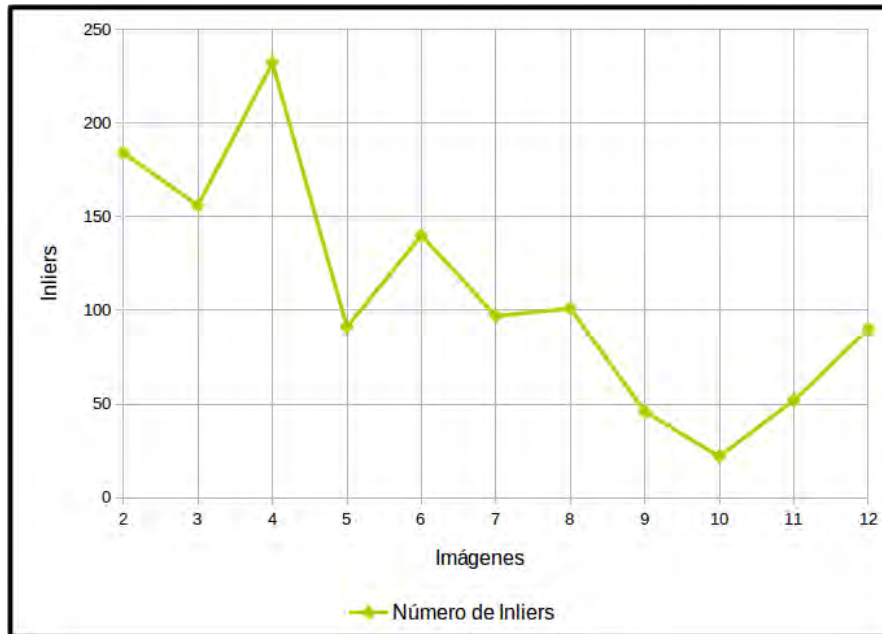


Figura 4.22: Gráfica nº Imágenes - Inliers, del dataset *Imágenes Urbanas 2*

En este experimento se puede concluir que, comparando los tiempos obtenidos en los dos ejemplos, estos guardan bastante similitud. En primer lugar, el cuello de botella del algoritmo se encuentra en el proceso que se encarga de encontrar las Costuras entre la imagen capturada y la imagen panorámica. En segundo lugar, el proceso de adherir una nueva imagen a la panorámica se vuelve cada vez más lento a medida de que esta crece. También, se puede observar que el tiempo del Matching, del cálculo del tanto por ciento de Superposición y de la estimación de la matriz de Rotación, suele ser muy constante, soliendo caer cuando la reconstrucción esta apunto de detenerse por alguno de los motivos evaluados. Por último, debido a la deformación que sufre las imágenes que se van uniendo a la panorámica y que se va acumulando más error a medida que esta crece, el número de Inliers (número característica de ambas imágenes de las que se obtiene la matriz de Homografía) va descendiendo con el crecimiento de esta.

#### 4.14. Resultados obtenidos al modificar el factor de escala en algunos métodos

Para este experimento solo se va a trabajar con un dataset (*Imágenes Urbanas 2*) y se va a estudiar como cambia el tiempo de ejecución del algoritmo, al aplicar un factor de escala en alguno de los procesos. También, se determinará como afecta este cambio de escala a la calidad de la imagen resultante. Para todo ello, se van a construir 7 panorámicas con solo 10 Imágenes (cada panorámica) del dataset.

Los tiempos comparados, será la media de cada uno de los tiempos de ejecución (Matching, calcular Superposición, estimar matriz de rotación, buscar Costuras, Unir imagen capturada, tiempo total) para cada imagen añadida a la panorámica. Las 7 panorámicas construidas tendrán las siguientes especificaciones respecto a la escala de la imágenes:

- Panorámica 1. Todo el proceso de creación se realizara con las imágenes a tamaño original ( $1500 \times 1001$ ).
- Panorámica 2. El proceso de buscar las Costuras (como se observo en la sección 5.16, es el proceso más lento) se hará con el tamaño de las imágenes por un factor de escala de 0,6. El resto de los procesos se realizarán a tamaño normal.
- Panorámica 3. El factor de escala en el proceso de buscar las Costuras se reducirá a 0,4. El resto de los procesos se realizarán a tamaño normal.
- Panorámica 4. El proceso de calcular el tanto por ciento de Superposición se hará con el tamaño de las imágenes por un factor de escala de 0,6. El resto de los procesos se realizarán a tamaño normal.
- Panorámica 5. El factor de escala en el proceso de calcular el tanto por ciento de Superposición se reducirá a 0,4. El resto de los procesos se realizarán a tamaño normal.
- Panorámica 6. Las imágenes, desde que se capturan se reducirán con un factor de escala de 0,6. Todos los procesos se realizarán con las imágenes reducidas.
- Panorámica 7. En este caso, el factor de escala es de 0,4 y se aplicará al tamaño de las imágenes desde que se capturen. Todos los procesos se realizarán con las imágenes reducidas.

En la Tabla 4.3, se puede ver un resumen de como se han construido las 7 panorámicas y en la Tabla 4.4, se puede observar los resultados de los tiempos de la reconstrucción.

Panorámica	Todos los procesos (FdE)	Búsqueda de las Costuras (FdE)	Cálculo del porcentaje de Superposición (FdE)
1	1	x	x
2	x	0.6	1
3	x	0.4	1
4	x	1	0.6
5	x	1	0.4
6	0.6	x	x
7	0.4	x	x

Tabla 4.3: Factores de escala aplicados en las 7 panorámicas

Panorámica	1	2	3	4	5	6	7
<b>Tiempo (s)</b>							
Matching	0.348	0.455	0.389	0.359	0.35	0.24	0.151
Estimar Rotación	0.019	0.032	0.017	0.015	0.017	0.013	0.006
Calcular Superposición	0.186	0.170	0.196	0.102	0.071	0.061	0.020
Buscar Costuras	5.389	1.312	0.207	5.694	5.406	1.519	0.196
Unir Imágenes	0.151	0.191	0.166	0.148	0.137	0.050	0.013
Total	6.093	2.159	0.976	6.319	5.987	1.884	0.386

Tabla 4.4: Tiempos de reconstrucción de las 7 panorámicas (cambiando la escala de las imágenes)

Si se echa un vistazo a la Tabla 4.4, los tiempos más pequeños se corresponden con la reducción del tamaño de las imágenes en el proceso de buscar las Costuras y con la reducción del tamaño de estas en todos los procesos. Cabe destacar, que el último caso incluye el primero más los procesos restantes y por eso es el caso más rápido. También, es importante darse cuenta que si se aplica el factor de escala desde que se captura la imagen, se perderá mucha información en la imagen resultante. Si se comparan, el tamaño de la tercera y la séptima panorámica (las panorámicas que se construyeron más rápido), como es lógico, el resultado del tamaño de la primera es mayor al doble que el de la última ( $1829 \times 2500$  frente a  $684 \times 783$ )

En conclusión, si se aplica un reescalado que reduzca el tamaño de las imágenes para calcular las Costuras de estas, se reducirá considerablemente el tiempo total de procesamiento del algoritmo y sin perder casi información en la panorámica resultante.

## 4.15. Resultados obtenidos al modificar del Porcentaje de Superposición

Ya que los vídeos permiten obtener diferentes porcentajes de superposición entre las imágenes, se utilizarán los datasets *Vídeo Mural* y *Vídeo Lago*, para estudiar como afecta este porcentaje a los tiempos de ejecución, al número de Inliers y al tamaño resultante de la panorámica. Además, también se examinará en este apartado que evaluación de las mencionadas en la sección 3.11 suele ser la que detiene el proceso de reconstrucción de la imagen panorámica para empezar con otro. Este porcentaje de superposición se modificará cambiando el umbral superior e inferior que permite que las imágenes capturadas sean posibles candidatas a formar parte de la panorámica. Las pruebas se realizarán con un umbral superior del 75 %, 70 %, 65 % y 55 % de Superposición, y el umbral inferior sera igual al superior menos el 15 %. Además, la simulación de este experimento se ha llevado a cabo de tal manera que el algoritmo piensa que recibe las imágenes desde una cámara y no desde los datasets, pudiéndose probar también el funcionamiento del programa en tiempo real.

Para este experimento, se utilizará un factor de escala para reducir la imagen en los procesos de búsqueda de las Costuras de las imágenes y para calcular el tanto por ciento de Superposición. Este escalado será de 0,33 en el caso del dataset *Vídeo Mural* y de 0,5 en el caso del dataset *Vídeo Lago*. Los resultados del experimento se pueden ver en las Figuras II.28 y II.29, donde se han representado las mejores panorámicas obtenidas en este experimento.

<b>Superposición (%)</b>	<b>75</b>	<b>70</b>	<b>65</b>	<b>55</b>
<b>Tiempo (s) para evaluar cada imagen capturada</b>				
Matching	0.707	0.709	0.802	0.988
Estimar Rotación	0.021	0.033	0.033	0.064
Calcular Superposición	0.073	0.087	0.095	0.115
<b>Tiempo (s) para añadir una nueva imagen a la panorámica</b>				
Buscar Costuras	0.487	0.670	0.542	0.533
Unir Imágenes	0.118	0.141	0.151	0.186

**Tabla 4.5:** Tiempo de ejecución - dataset *Vídeo Lago*

En los resultados de la Tabla 4.6, se puede observar que no existe mucha diferencia en el tiempo de ejecución al cambiar el porcentaje de Superposición. En este caso, se podría utilizar el menor porcentaje que se ha probado, ya que como la imagen tiene buena resolución casi siempre se va a encontrar un número

Superposición (%)	75	70	65	55
<b>Tiempo (s) para evaluar cada imagen capturada</b>				
Matching	0.610	0.624	0.626	0.665
Estimar Rotación	0.029	0.025	0.023	0.024
Calcular Superposición	0.081	0.085	0.078	0.085
<b>Tiempo (s) para añadir una nueva imagen a la panorámica</b>				
Buscar Costuras	0.752	0.754	0.795	0.481
Unir Imágenes	0.179	0.207	0.203	0.253

Tabla 4.6: Tiempo de ejecución - dataset *Vídeo Mural*

Superposición (%)	75	70	65	55
<b>RESULTADOS DE LA PANORÁMICA 1</b>				
Media de inliers	140	112	82	56
Tamaño resultante	2335x2484	1442x1494	2188x2247	1589x2083
Imágenes unidas	9	4	7	4
Tipo de error	Escala > 2 (2.001)	Escala > 2 (2.001)	Escala > 2 (0.424)	Escala > 2 (2.044)
<b>RESULTADOS DE LA PANORÁMICA 2</b>				
Media de inliers	49	87	50	66
Tamaño resultante	1737x4716	1887x2195	1653x403	1944x1972
Imágenes unidas	15	5	12	6
Tipo de error	Escala < 0.5 (0.242)	Escala > 2 (2.215)	Escala > 2 (2.0.13)	Escala > 2 (2.044)
<b>RESULTADOS DE LA PANORÁMICA 3</b>				
Media de inliers	60	50	34	36
Tamaño resultante	2344x2652	1807x3017	1788x2744	1733x2944
Imágenes unidas	9	8	5	6
Tipo de error	Escala > 2 (3.891)	Escala > 2 (2.018)	Escala > 2 (2.112)	Escala > 2 (2.508)

Tabla 4.7: Información de los resultados - dataset *Vídeo Lago*

suficiente de Inliers en las imágenes. En el caso del experimento del lago al 55% de Superposición, el número de Inliers medio de las tres panorámicas es bajo (ver Tabla 4.7). Esto último puede provocar que se creen matrices de transformación con cierto error y que este error se vaya acumulando.

Utilizar un porcentaje de superposición menor, permite crear panorámicas más grandes con un menor número de imágenes. Por ejemplo, en la Tabla 4.8 se puede

Superposición (%)	75	70	65	55
<b>RESULTADOS DE LA PANORÁMICA 1</b>				
Media de inliers	198	226	202	126
Tamaño resultante	9891x1807	10141x1950	10320x1435	11430x1431
Imágenes unidas	13	12	12	11
Tipo de error	Escala > 2 (2.254)	Escala > 2 (4.063)	Escala > 2 (2.086)	Escala > 2 (2.017)
<b>RESULTADOS DE LA PANORÁMICA 2</b>				
Media de inliers	433	321	339	214
Tamaño resultante	7286x1626	7462x1785	8897x1733	7015x1452
Imágenes unidas	9	8	9	6
Tipo de error	Escala > 2 (2.221)	Escala > 2 (4.920)	Escala > 2 (2.541)	Escala > 2 (2.240)
<b>RESULTADOS DE LA PANORÁMICA 3</b>				
Media de inliers	262	243	219	170
Tamaño resultante	5547x1597	3753x1420	6735x2094	6269x2257
Imágenes unidas	7	4	11	9
Tipo de error	Escala > 2 (2.341)	Escala > 2 (2.023)	No más imágenes	No más imágenes

**Tabla 4.8:** Información de los resultados - dataset *Vídeo Mural*

observar como con un 55% de Superposición se construye una panorámica más grande que usando un 65%, utilizando además dos imágenes menos. Como ya se mencionó, en el caso del experimento con el Mural, se obtiene un número alto de Inliers debido a la resolución, pero la resolución de las imágenes del Lago es menor y por lo tanto el porcentaje de superposición si afecta al número de Inliers encontrados.

También, en los experimentos (por ejemplo en las panorámicas creadas en el Lago (ver Figura II.28), se puede ver como un cambio en la altitud del UAV puede provocar cambios de escala y de percepción muy grandes.

Por último, se ha demostrado con este experimento que se puede trabajar a tiempo real como muestran los tiempos de las tablas 4.6 y 4.5.



# **CAPÍTULO 5: CONCLUSIÓN Y TRABAJOS FUTUROS**

## 5.16. Conclusión

Como conclusión, en este proyecto se ha propuesto un algoritmo basado en la detección de características para la reconstrucción de panorámicas destinado a aplicaciones que necesiten trabajar en tiempo real y sin perder mucha información a cambio. En primer lugar, se ha propuesto una serie de aplicaciones en las que este algoritmo sería gran utilidad. En segundo lugar, se ha explicado porque se ha elegido un método de creación de panorámicas basado en la detección de características y se han estudiados los pasos típicos a seguir en el proceso de reconstrucción. Después, teniendo estos pasos en cuenta, se ha determinado cuales formarían parte del algoritmo de reconstrucción planteado en este trabajo. A partir de estudios realizados en otros trabajos, se han elegido los métodos más convenientes teniendo en cuenta el objetivo de este. Por otro lado, se han propuestos cambios en el proceso de reconstrucción como mejorar la textura antes de detectar las características, alinear las imágenes teniendo en cuenta los Inliers, calcular el porcentaje de superposición entre las imágenes, y escalar las imágenes a conveniencia según el método que se este aplicando. Además, se ha propuesto un sistema de evaluación de imágenes que elige la mejor entre un conjunto seleccionado a priori. También, se ha utilizado este sistema para detectar errores de Matching entre imágenes y determinar cuando empezar una nueva reconstrucción. Por último, se ha realizado una serie de experimentos para comprobar si el algoritmo es capaz de trabajar en tiempo real, que configuración de los parámetros de este es la idónea y como afecta los diferentes escenarios, y la resolución de la imagen, a los resultados.

Referente a los resultados obtenidos en los experimentos, se puede concluir que el método propuesto cumple con los objetivos pero con ciertos requisitos. Por ejemplo, en los experimentos de la Sección se ha podido observar que el tiempo medio de adherir una imagen a la panorámica, (más el tiempo de evaluar una imagen) se puede reducir un 84 % con solo escalar la imagen para realizar el paso de buscar las líneas de costura, permitiendo pasar de un tiempo con el que sería imposible trabajar en tiempo real a otro con el que si sería posible. Esto último, como se observo en los experimentos, apenas añade errores en los resultados. Estos errores que añade son prácticamente imperceptibles, encontrándose solo en las zonas de unión entre las imágenes y sin provocar perdida de información.

También se pudo ver en los experimentos, que aunque el tiempo medio de ejecución, al reducir el tamaño de la imagen, es lo suficientemente bajo para trabajar en tiempo real, existen momentos en los que el tiempo total puede llegar a los 2,4 segundos (ver Gráfica 4.16). Esto último puede causar que haya que

limitar la velocidad de traslación del UAV para que le de tiempo a reconstruir la panorámica sin perder información, teniendo en cuenta que para la reconstrucción de fachadas hay que limitarla más esta velocidad, debido a que con menor movimiento se consigue un porcentaje de superposición mucho más pequeño que en el caso de una reconstrucción aérea.

Centrándose en los resultados de las imágenes panorámicas obtenidas, se puede concluir que el tamaño de estas y el número de imágenes que las componen, están bastante relacionados con: la resolución de las imágenes de entrada, la textura, como de bien el UVA se mantiene a la misma distancia del escenario y el porcentaje de superposición. El resultado de las panorámicas, se ve más afectado por este último, si las imágenes tienen menos textura y menos resolución, como se puede observar en los resultados de la Sección 4.15. Además, en esta sección también se pudo comprobar que se puede utilizar el algoritmo en tiempo real y que el método de evaluación funciona correctamente, empezando la construir de una nueva panorámica cuando: la anterior ya ha acumulado mucho error, hay cambios de nivel en el escenario que provoca que el UAV no se pueda mantener a la misma distancia de este y cuando se ha capturado la primera imagen con cierto ángulo respecto al plano.

## 5.17. Trabajos futuros

A lo largo de los experimentos se observó, que donde el algoritmo es más lento es para realizar el Matching y encontrar las costuras entre la panorámica y la imagen capturada. En esta sección, se propone algunas mejoras en estos y otros procesos para agilizar el algoritmo. También, se proponen otros cambios para obtener mejores resultados en el aspecto de la imagen panorámica.

Las primeras dos mejoras son relacionadas con agilizar el Matching de las imágenes. Sobre todo la parte de detección y descripción de características. En primer lugar, se propone el almacenamiento de los descriptores y los Keypoints de la imagen panorámica desde que se empieza a evaluar la primera imagen hasta que se selecciona una para el Blending. De esta manera, buscar Keypoints y extraer los descriptores solo se hará una vez por Blending y no cada vez que se capture una imagen para ser evaluada. Una vez estén estos datos guardados, se podrán comparar con los de las nuevas imágenes capturadas para hacer el Matching. La segunda mejora que se propone en estos dos procesos, es el uso de Multithreading (Multihilo), ya que así se puede dividir la imagen en dos, cuatro o más cuadrantes (según el número de hilos y núcleos que tenga el procesador), detectando los Keypoints y obteniendo los descriptores de cada cuadrante en paralelo. Después

de haber calculado cada grupo de descriptores por separado, se puede reconstruir la matriz de descriptores total para hacer el Matching con los descriptores de la otra imagen. También, se realizaría un estudio para saber como mejoraría estas dos nuevas implementaciones al tiempo de ejecución del algoritmo.

Por otro lado, para mejorar el aspecto de la panorámica y crear panorámicas con mayor número de imágenes, (y que por lo tanto abarquen mayor área) se propone para el futuro crear una función de coste para la elección de la mejor imagen. Para este trabajo, se ha implementado el proceso de selección de tal manera que se elije la mejor mejor imagen teniendo en cuenta solo como cambia la imagen deformada respecto a la original (ver sección 3.10). La función de coste que se propone para los trabajos futuros podría estar formada por: el número de Inliers encontrados para esa imagen, una variable que describa como están distribuidos estos Inliers en la imagen y como cambia la imagen deformada respecto a la original. Cada uno de estos parámetros se podrían multiplicar por un peso y se podría encontrar el mejor valor de cada uno de los pesos de forma experimental. Por último, la imagen sería seleccionada teniendo en cuenta el valor de esta función, por lo tanto estos pesos deben tener el signo correcto para que cuanto más pequeño sea el resultado de la función, mejor sea la imagen para ser adherida a la panorámica.

# **CAPÍTULO 6:**

## **MARCO REGULADOR**

De manera informativa y para el conocimiento de la ley que se aplica en el uso de aeronaves no tripuladas en entornos relacionados con la implementación realizada en este trabajo, en este capítulo se realizará un análisis de la normativa específica. En primer lugar, se expondrá cual es la agencia española que trabaja para que se cumplan las normas de seguridad de transporte aéreo en España. A continuación, se hablará del Real Decreto-ley 8/2014, de 4 de julio, en cuya sección 6ª se recoge el régimen temporal para las operaciones con aeronaves pilotadas por control remoto, enfatizando en que se puede hacer con un dron en la actual regulación española y los requisitos necesarios para grabar en exteriores.

En España, **AESA** (Agencia Estatal de Seguridad Aérea) es el organismo que vela por el cumplimiento de la normativa de aviación civil en todo el conjunto de la actividad aeronáutica. En la Ley 28/2006, de 18 de julio se autoriza la creación de la agencia para la ejecución de funciones de ordenación, supervisión e inspección de la seguridad del transporte aéreo. AESA, también evalúa los riesgos que se pueden producir en la seguridad del transporte aéreo mediante: la detección de amenazas, el análisis y la evaluación de los riesgos, y un proceso de control y mitigación del riesgo. Además, tiene la potestad de sancionar si se producen infracciones de las normas de aviación civil. En el Real Decreto 184/2008, de 8 de Febrero, se aprueba el Estatuto de AESA, por lo tanto a parte de regirse por la Ley de las Agencias estatales, esta agencia también lo hace por su propio estatuto.

La sección 6ª **Real Decreto-ley 8/2014**, de 4 de julio, se recoge el régimen temporal para las operaciones con drones de peso inferior a los 150 kg al despegue, en el que se establecen las condiciones de explotación de estas aeronaves para la realización de trabajos técnicos y científicos. En esta sección, se expondrán los puntos más significativos de esta ley teniendo en cuenta las posibles aplicaciones de este proyecto.

- En primer lugar, mencionar que lo que se establece en esta ley **no exime al responsable de la aeronave del resto de la normativa que se aplica**. Por ejemplo, en la protección de datos o la toma de imágenes aéreas, **sea el caso de vuelos fotogramétricos, hay que solicitar un permiso ante el Consejo Superior Geográfico**. Ni tampoco exime, de la responsabilidad del operador por los daños causados por la operación o la aeronave.
- Respecto al registro de la aeronave, las aeronaves civiles pilotadas por control remoto cuya masa máxima al despegar sea mayor de 25 kg deben estar **inscritas en el registro de matrícula de aeronaves** y deben tener un certificado de aeronavegabilidad. Quedando exentas de este requisito las aeronaves con un masa menor o igual a 25 kg.

- Actualmente se pueden utilizar drones para realización de trabajos aéreos como son: **actividades de investigación y desarrollo, levantamientos aéreos, observación y vigilancia aérea incluyendo filmación y actividades de vigilancia de incendios forestales, operaciones de emergencia, búsqueda y salvamento.** Pero hasta el momento, las operaciones que se pueden realizar **se limitan a zonas no pobladas y al espacio aéreo no controlado.**
- **Las actividades aéreas de trabajos técnicos o científicos** en condiciones meteorológicas visuales podrá ser realizadas bajo una serie de restricciones que se exponen en el Real Decreto-ley. Por ejemplo, la normativa permite a un UAV de hasta 25 Kg, grabar en exteriores, de día y en **condiciones meteorológicas favorables**, en zonas **fuera de aglomeraciones** de edificios en ciudades, pueblos o lugares habitados o de reuniones de personas al aire libre. También hay que tener en cuenta que se debe hacer en **espacio aéreo no controlado, dentro del alcance visual del piloto**, a una **distancia de éste no mayor de 500 m.** y a una **altura no mayor de 120 m** sobre el terreno. Además, el operador debe tener la documentación relativa en cima, debe también disponer de un manual de operaciones y haber realizado las operaciones de mantenimiento debidas.
- Para la realización de fotografías o filmaciones con cualquier tipo de aeronaves, tripuladas o no, es necesario cumplir con la Orden de Presidencia del Gobierno de 14 de marzo de 1957. Si va a operar **en Zonas Restringidas al Vuelo Fotográfico (RVF)** se **deberá contactar con el Estado Mayor del Aire.** En el resto de casos (Zonas No Restringidas al Vuelo Fotográfico (NRVF)) no es necesaria su consulta.
- Todos los pilotos de drones, indistintamente del tamaño de la aeronave, deberán **acreditar que poseen los conocimientos teóricos necesarios para obtener una licencia de piloto**, lo que se puede hacer de tres formas: tener o haber tenido en los últimos 5 años cualquier licencia de piloto, demostrar de forma fehaciente que disponen de los conocimientos teóricos para obtener la licencia o (si el peso máximo al despegue no es superior a 25 Kg) por medio de un certificado básico o avanzado emitido por una organización de formación aprobada (ATO) tras superar un curso.
- Para el realizar actividades aéreas de trabajos técnicos o científicos, también se deberá **comunicar a la Agencia Estatal de Seguridad Aérea** con una antelación mínima de cinco días al día del inicio de la operación.

# **CAPÍTULO 7:**

## **ANÁLISIS SOCIO-ECONÓMICO**



En este capítulo se propondrá una tabla con el presupuesto necesario para llevar a cabo un proyecto semejante al estudiado en este trabajo, y se expondrán algunas aplicaciones en las que el uso de un sistema UAV con el algoritmo de reconstrucción de panorámicas (y junto a una cámara monocular), podría generar un gran impacto en el sector referente a dichas aplicaciones.

## 6.18. Presupuesto

En esta sección se muestra una planificación aproximada del tiempo necesario para llevar a cabo el proyecto (teniendo en cuenta aproximadamente el tiempo personal que ha sido invertido en el), para realizar un cálculo del coste de ejecución. Esto, junto al coste del material y el los dispositivos referentes al sistema del UAV, formarán el presupuesto para llevar a cabo un proyecto de la misma envergadura.

Aunque los experimentos se han realizado en una estación de tierra, en esta sección se detalla el presupuesto para el montaje de un Hexacoptero, con una cámara monocular y una unidad de procesamiento como se muestra en el capítulo 1, 2 y 3 de las Tabla 6.9, 6.10 y 6.11, respectivamente. Con la configuración del montaje dada en este presupuesto, se pueden realizar pruebas tanto de inspección de fachadas como de fotografías aéreas con un plus de velocidad de procesamiento comparado comparado con el que se ha tenido en las pruebas realizadas en este proyecto.

CODIGO	UNIDAD	DESCRIPCIÓN	MEDICIÓN	PRECIO UNITARIO (€)	PRECIO TOTAL (€)
01		<b>CAPÍTULO 1. ESTRUCTURA DEL UAV</b>			
01.01	Ud	Frame de Hexacoptero plegable TAROT 680PRO	1,00	149,00	149,00
01.02	Ud	Motores brushless (sin escobillas) marca Leopard LC 2826-14T 1250KV	6,00	21,00	126,00
01.03	Ud	Variador (ESC) T-Motors Air 40A	6,00	33,00	198,00
01.04	Ud	Batería Gens ACE 4s 5000mAh 45C	1,00	89,00	89,00
01.05	Ud	Gimbal 2 ejes brushless Tarot Gopro 3/4	1,00	45,00	45,00
01.05	Ud	Paquete de Hélices	1,00	24,00	24,00
01.06	Ud	Sistema FPV: pantalla, camara y Video Tx/Rx	1,00	265,00	265,00
		<b>TOTAL CAPÍTULO</b>			896,00

Tabla 6.9: Capítulo 1 del presupuesto: estructura del UAV

CODIGO	UNIDAD	DESCRIPCIÓN	MEDICIÓN	PRECIO UNITARIO (€)	PRECIO TOTAL (€)
02		<b>CAPÍTULO 2. SENSORES</b>			
02.01	Ud	Cámara molecular SJCam SJ4000 WIFI, Full HD a 30 fps	1,00	98,00	98,00
		<b>TOTAL CAPÍTULO</b>			98,00

Tabla 6.10: Capítulo 2 del presupuesto: sensores

CODIGO	UNIDAD	DESCRIPCIÓN	MEDICIÓN	PRECIO UNITARIO (€)	PRECIO TOTAL (€)
03		<b>CAPÍTULO 3. SISTEMAS</b>			
03.01	Ud	Unidad de procesamiento Intel NUC 6i5SYK (Intel Core i5-6260U) + RAM 8GB DDR4 + SSD 250GB	1,00	688,00	688,00
		<b>TOTAL CAPÍTULO</b>			688,00

Tabla 6.11: Capítulo 3 del presupuesto: sistemas

El desarrollo del trabajo se ha dividido en cinco etapas diferentes para estimar

el tiempo total que se ha dedicado a la realización del proyecto. Las 5 fases o etapas en las que se ha dividido son: etapa de investigación, etapa de diseño y arquitectura, etapa de programación, etapa de pruebas o experimentos y etapa de documentación. El tiempo aproximado, teniendo en cuenta las fechas en las que se realizó el proyecto es:

- Tiempo en etapa de investigación = 40 días · 5 horas/día = 200 horas
- Tiempo en etapa de diseño y arquitectura = 10 días · 5 horas/día = 50 horas
- Tiempo en etapa de programación = 30 días · 5 horas/día = 150 horas
- Tiempo en etapa de pruebas = 20 días · 5 horas/día = 100 horas
- Tiempo en etapa de documentación = 20 días · 5 horas/día = 100 horas

En el capítulo 4º del presupuesto de la Tabla 6.12 se puede observar el total de horas y el precio total de la mano de obra. En este precio, no se ha incluido futuro mantenimiento del sistema, el cual debe incluir el precio de la mano de obra para futuras mejoras y correjimiento de errores en el programa.

CODIGO	UNIDAD	DESCRIPCIÓN	MEDICIÓN	PRECIO UNITARIO (€)	PRECIO TOTAL (€)
04		<b>CAPÍTULO 4. MANO DE OBRA</b>			
04.01	Horas	Ingeniero junior	600	18,00	10.800,00
		<b>TOTAL CAPÍTULO</b>			10.800,00

**Tabla 6.12:** Capítulo 4 del presupuesto: mano de obra

Finalmente, en la Tabla 6.13, se encuentra un resumen con el total de cada uno de los capítulos del presupuesto más el total del proyecto.

CODIGO	UNIDAD	DESCRIPCIÓN	MEDICIÓN	PRECIO UNITARIO (€)	PRECIO TOTAL (€)
		<b>RESUMEN CAPÍTULOS</b>			
		<b>CAPÍTULO 1. ESTRUCTURA DEL UAV</b>			896,00
		<b>CAPÍTULO 2. SENSORES</b>			98,00
		<b>CAPÍTULO 3. SISTEMAS</b>			688,00
		<b>CAPÍTULO 4. MANO DE OBRA</b>			10.800,00
		<b>TOTAL CAPITULOS</b>			12.482,00

**Tabla 6.13:** Resumen de los capítulos del presupuesto

## 6.19. Aplicaciones propuestas en las que las imágenes panorámicas serían de utilidad

La primera aplicación que se propone, sería de gran utilidad en el mantenimiento de las pistas de aterrizaje, así como de gran impacto en el sector y que ayudaría a agilizar la tarea de inspección en ellas. Según la guía para el mantenimiento de las pistas de acuerdo al Anexo 14 de la OACI (Organización de aviación civil internacional), es necesario el mantenimiento adecuado de la superficie de la pista asegurando una adecuada profundidad de textura, así como tipos y niveles de fricción adecuados. OACI promueve el drenaje de superficie, así como la eliminación del caucho y los contaminantes de la superficie de la pista como una práctica regular. También, Para mantener las marcas de pista en condiciones óptimas, los aeropuertos necesitan evaluar las marcas para determinar si se necesita remarcado o sólo se requiere mantenimiento (algunos criterios para determinar si es necesario este remarcado son mencionados en el informe del IPRF 01-G-002-05-1, Manual de Marcado de Aeródromos).

Por lo tanto, según lo descrito arriba, se utilizar el algoritmo desarrollado en este trabajo para la creación de una panorámica en la que se pueda visualizar de manera eficaz la acumulación de caucho en la pista y el estado de las marcas (en la imagen de la Figura 6.23, se puede observar tanto la acumulación de caucho, como las marcas de la pista). Para ello, se ha realiza un experimento en el que a partir de una ortofoto obtenida desde la base de datos de Bing Maps creada por Microsoft, se a obtenido un dataset de 20 imágenes de la pista de aterrizaje del aeropuerto de Seattle, con una resolución de 830x740 Las imágenes del dataset se pueden observar en la Figura II.30 y el resultado de la panorámica en la Figura II.31, ambos en el Anexo II.



**Figura 6.23:** Acumulación de caucho y marcas de la pista de aterrisaje del aeropuerto de seattle

En un sistema destinado a la comprobación del estado de las carreteras, se podría hacer uso de la creación de imágenes panorámicas junto a la odometría visual y la detección de objetos a partir de la detección de características. Estas técnicas, se mezclarían utilizándose de tal manera que:

- Mediante la detección de características, se localizarían las señales que marcan los puntos kilométricos de las carreteras.
- Usando el resultado de la imagen panorámica, se visualizarían posibles defectos en el tramo de la carretera.
- Si existe algún defecto o algo que se quiera remarcar en la carretera, con la odometría visual se sabrá la posición de esta marca, referente a los puntos kilométricos y lo hará con más exactitud que usando el gps.

Este método de marcado de los puntos kilométricos sería de gran ayuda para agilizar la labor de detección de imperfecciones en el marcado y en el pavimento de la carretera.

# Bibliografía

- [1] P. Fahlstrom and T. Gleason, *Introduction to UAV systems*. John Wiley & Sons, 2012.
- [2] T. Skrzypietz, “Unmanned aircraft systems for civilian missions, bigs policy paper, no. 1,” *Brandenburg Institute for Society and Security, Brandenburgisches Institute für Gesellschaft und Sicherheit*, 2012.
- [3] G. F. Griffin, “The use of unmanned aerial vehicles for disaster management,” *Geomatica*, vol. 68, no. 4, pp. 265–281, 2014.
- [4] S. Arya, “A review on image stitching and its different methods,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 5, 2015.
- [5] M. Alomran and D. Chai, “Feature-based panoramic image stitching,” in *Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on*. IEEE, 2016, pp. 1–6.
- [6] R. Littlefield, “Theory of the no-parallax point in panorama photography [-eb/ol].(2006—02 06).”
- [7] R. Szeliski and H.-Y. Shum, “Creating full view panoramic image mosaics and environment maps,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 251–258.
- [8] M. Uyttendaele, A. Eden, and R. Skeliski, “Eliminating ghosting and exposure artifacts in image mosaics,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, pp. II–II.
- [9] D. Khunt and Y. Makwana, “Image registration using intensity based technique,” *Journal of information, knowledge and research in electronics and communication engineering*, vol. 2, no. 2, pp. 607–609, 2012.

- [10] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [11] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, 2017.
- [12] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.
- [13] L. Wang, J. Shi, G. Song, and I.-F. Shen, "Object detection combining recognition and segmentation," in *Asian conference on computer vision*. Springer, 2007, pp. 189–199.
- [14] H. Chau and R. Karol, "Robust panoramic image stitching."
- [15] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless image stitching in the gradient domain," *Computer Vision-ECCV 2004*, pp. 377–389, 2004.
- [16] C.-S. Chen, Y.-T. Chen, and F. Huang, "Stitching and reconstruction of linear-pushbroom panoramic images for planar scenes," *Computer Vision-ECCV 2004*, pp. 190–201, 2004.
- [17] N. Heinze, M. Esswein, W. Krüger, and G. Saur, "Automatic image exploitation system for small uavs," in *Proc. SPIE*, vol. 6946, 2008, p. 69460G.
- [18] S. Se, P. Firoozfam, N. Goldstein, L. Wu, M. Dutkiewicz, P. Pace, and J. Naud, "Automated uav-based mapping for airborne reconnaissance and video exploitation," in *Proceedings of SPIE Vol*, vol. 7307, 2009.
- [19] S. Hsu, "Geocoded terrestrial mosaics using pose sensors and video registration," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I.
- [20] A. Kern, M. Bobbe, and U. Bestmann, "Towards a real-time aerial image mosaicing solution."
- [21] K. Mitra and R. Chellappa, "A scalable projective bundle adjustment algorithm using the l infinity norm," in *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*. IEEE, 2008, pp. 79–86.



- [22] T. Kekec, A. Yildirim, and M. Unel, “A new approach to real-time mosaicing of aerial images,” *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1755–1767, 2014.
- [23] P. Schneider and D. H. Eberly, *Geometric tools for computer graphics*. Morgan Kaufmann, 2002.
- [24] T. Botterill, S. Mills, and R. Green, “Real-time aerial image mosaicing,” in *Image and Vision Computing New Zealand (IVCNZ), 2010 25th International Conference of*. IEEE, 2010, pp. 1–8.
- [25] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. Ieee, 2006, pp. 2161–2168.
- [26] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov 2000.
- [27] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, Jun. 1981.
- [28] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2006.
- [29] S. Belongie and D. Kriegman, “Explanations on homography estimation,” *Department of Computer Science And Engineering, University of California, San Diego*, 2007.
- [30] O. Faugeras and F. Lustman, “Motion and structure from motion in a piecewise planar environment,” INRIA, Tech. Rep. RR-0856, Jun. 1988. [Online]. Available: <https://hal.inria.fr/inria-00075698>
- [31] E. Malis and M. Vargas, “Deeper understanding of the homography decomposition for vision-based control,” Ph.D. dissertation, INRIA, 2007.
- [32] T. Werner and T. Pajdla, “Cheirality in epipolar geometry,” in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 1, 2001, pp. 548–553 vol.1.
- [33] M. F. Khan, E. Khan, and Z. Abbasi, “Image contrast enhancement using normalized histogram equalization,” *Optik-International Journal for Light and Electron Optics*, vol. 126, no. 24, pp. 4868–4875, 2015.

- [34] E. Mair, G. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and generic corner detection based on the accelerated segment test,” *Computer Vision–ECCV 2010*, pp. 183–196, 2010.
- [35] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla, “Evaluation of low-complexity visual feature detectors and descriptors,” in *2013 18th International Conference on Digital Signal Processing (DSP)*, July 2013, pp. 1–7.
- [36] A. Alahi, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*. Ieee, 2012, pp. 510–517.
- [37] D. Bekele, M. Teutsch, and T. Schuchert, “Evaluation of binary keypoint descriptors,” in *2013 IEEE International Conference on Image Processing*, Sept 2013, pp. 3652–3656.
- [38] J. Figat, T. Kornuta, and W. Kasprzak, *Performance Evaluation of Binary Descriptors of Local Features*. Cham: Springer International Publishing, 2014, pp. 187–194. [Online]. Available: [https://doi.org/10.1007/978-3-319-11331-9\\_23](https://doi.org/10.1007/978-3-319-11331-9_23)
- [39] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration.” *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.
- [40] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: image and video synthesis using graph cuts,” in *ACM Transactions on Graphics (ToG)*, vol. 22, no. 3. ACM, 2003, pp. 277–286.
- [41] Z. Zhu, J. Lu, M. Wang, S. Zhang, R. Martin, H. Liu, and S. Hu, “A comparative study of algorithms for realtime panoramic video blending,” 2016.
- [42] S. Suzuki and K. be, “Topological structural analysis of digitized binary images by border following,” *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32 – 46, 1985.
- [43] L. Yang and F. Albrechtsen, “Fast and exact computation of cartesian geometric moments using discrete green’s theorem,” *Pattern Recognition*, vol. 29, no. 7, pp. 1061 – 1073, 1996.

- [44] W.-T. Lin, C.-H. Lin, T.-H. Wu, and Y.-K. Chan, “Image segmentation using the k-means algorithm for texture features,” *World Academy of Science, Engineering and Technology*, vol. 65, pp. 612–615, 2010.
- [45] M. Xia, J. Yao, R. Xie, L. Li, and W. Zhang, “Globally consistent alignment for planar mosaicking via topology analysis,” *Pattern Recognition*, vol. 66, pp. 239–252, 2017.

# Anexos



## Anexo I: Pseudocódigo del algoritmo

---

**Algorithm 1:** Creación de Imágenes Panorámicas
 

---

**Input:** Imágenes capturadas  $F$

**Output:** Imágenes Panorámicas  $Pano$

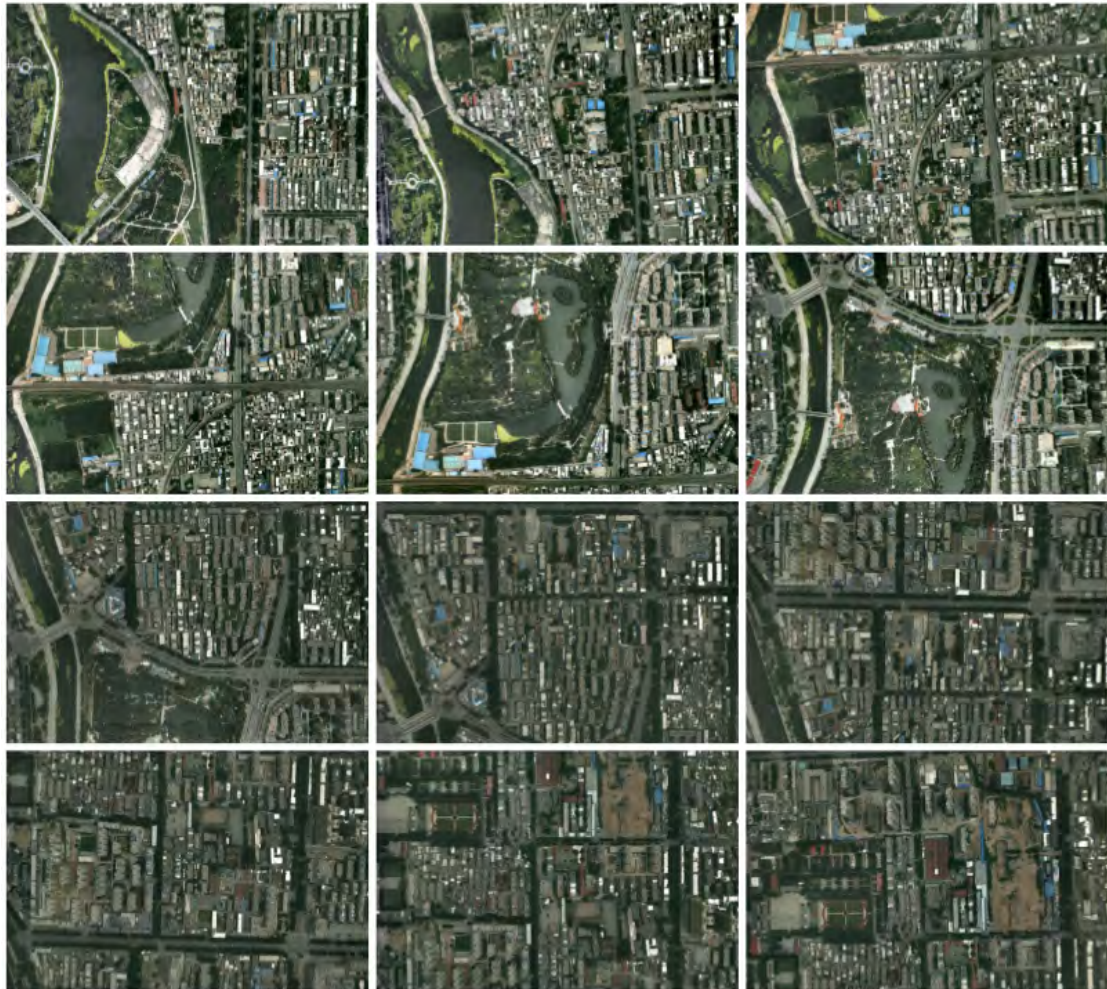
```

1 Define: Frame inicial  $F_i$ , Frame actual  $F_t$ , Keypoints iniciales  $kp_i[ ]$ ,
   keypoints actuales  $kp_t[ ]$ , Número de keypoints  $(N, M)$ ,  $n \in N$ ,  $m \in M$ ,
    $n = m$ ,  $x \in n$ , Inliers iniciales  $inlr_i$ , Inliers actuales  $inlr_t$ , Inliers
   Proyectados  $inlr_{tp}$ , Homografía  $H$ , Matriz de Rotación  $R$ , Estado de la
   Homografía  $state$ , Distancias  $d_i, d_t$ , Corners  $C_t, C_i$  y porcentaje de
   Superposición  $Perc_{VO}$ 
2 begin
3    $F_i \leftarrow$  ObtenerNuevoFrame( )
4   while  $Overlap_{precent} = OvMinThresh$  do
5      $F_t \leftarrow$  ObtenerNuevoFrame( )
6      $(kp_t(N), kp_i(M)) \leftarrow$  DetectarKeypoints( $F_t, F_i$ )
7      $(mp_t(n), mp_i(m)) \leftarrow$  MatchinKeypoints( $kp_t(N), kp_i(M), thresh$ )
8      $(inlr_t(n), inlr_i(m), H) \leftarrow$  RANSAC( $mp_t(n), mp_i(m)$ )
9      $state \leftarrow$  Validar( $H$ )
10    if  $state = noValido$  then
11      goto step 3
12     $R \leftarrow$  DescomponerHomografía( $H$ )
13     $inlr_{tp}(n) \leftarrow$  Proyectar( $inlr_t(n)$ )
14     $d_t \leftarrow$  calDist( $inlr_{tp}(n)$ )
15     $d_i \leftarrow$  calDist( $inlr_i(m)$ )
16     $scale \leftarrow \frac{d_t}{d_i}$ 
17    if  $0.5 < scale < 2$  then
18      goto step 3
19     $(C_t, C_i) \leftarrow$  CalcularCorners( $F_t, F_i$ )
20     $Perc_{VO} \leftarrow$  CalcularPorcentajeSuperposición( $C_t, C_i$ )
21    if  $55\% < Perc_{VO} < 75\%$  then
22       $F_t \leftarrow$  Guardar()
23    else
24      Continue
25     $Pano \leftarrow$  AdherirImagen( $F_t, F_i$ )
26     $F_i \leftarrow F_t$ 

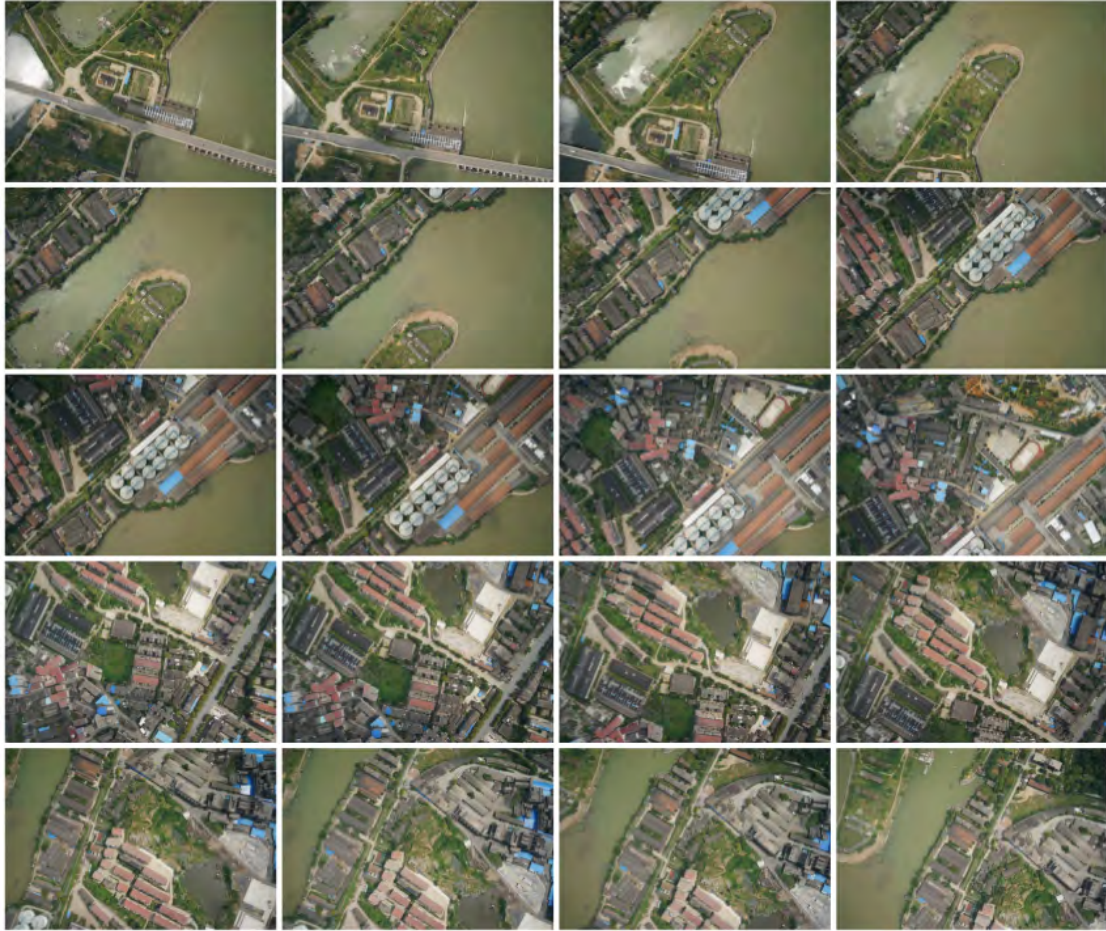
```

---

## Anexo II: imágenes panorámicas creadas en los experimentos



**Figura II.24:** Dataset *Imágenes Urbanas 1*, compuesto por 12 imágenes



**Figura II.25:** Dataset *Imágenes Urbanas 2*, compuesto por 20 imágenes





**Figura II.26:** Imagen panorámica construida con el dataset *Imágenes Urbanas 1*



**Figura II.27:** Imagen panorámica construida con el dataset *Imágenes Urbanas 2*

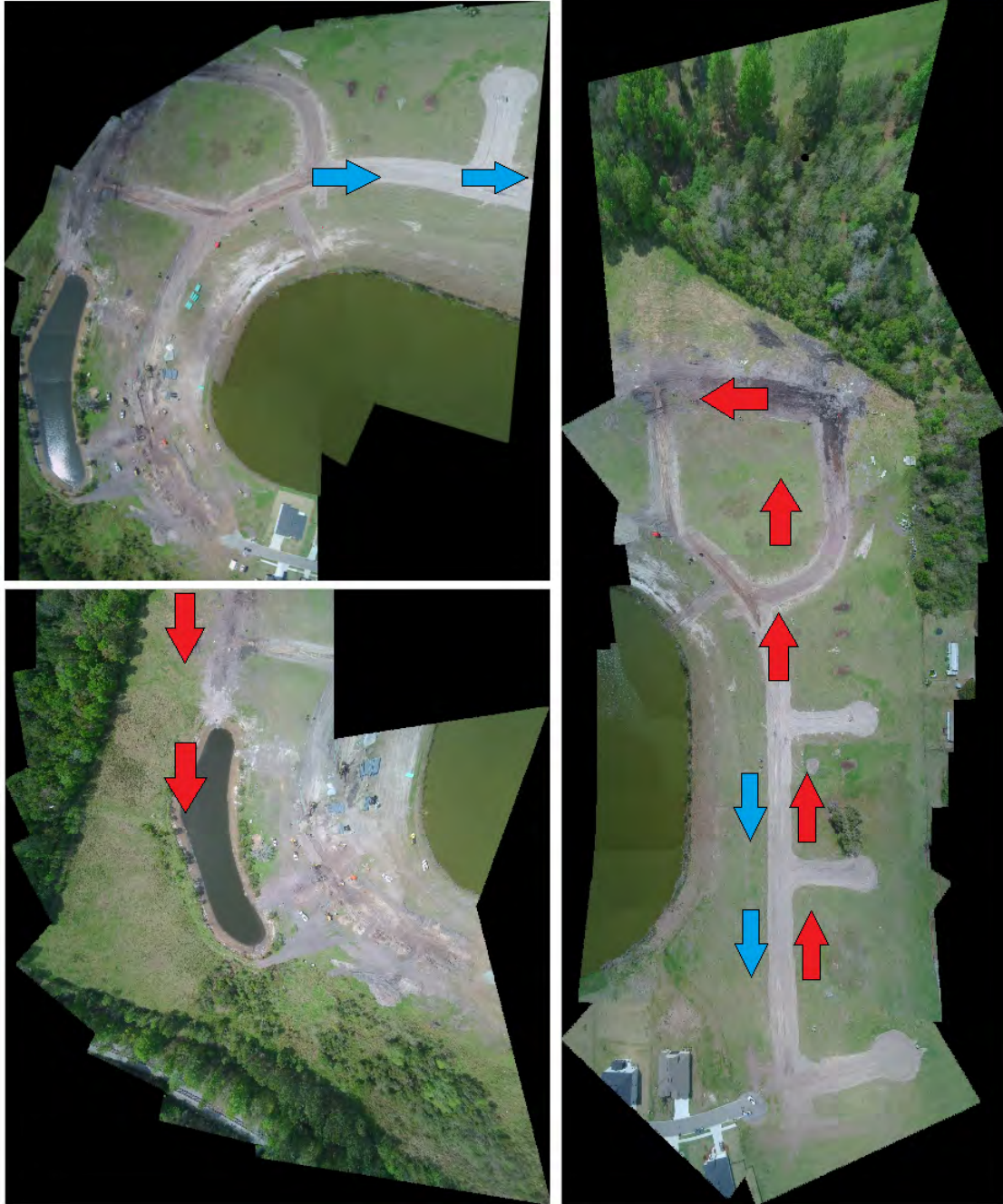
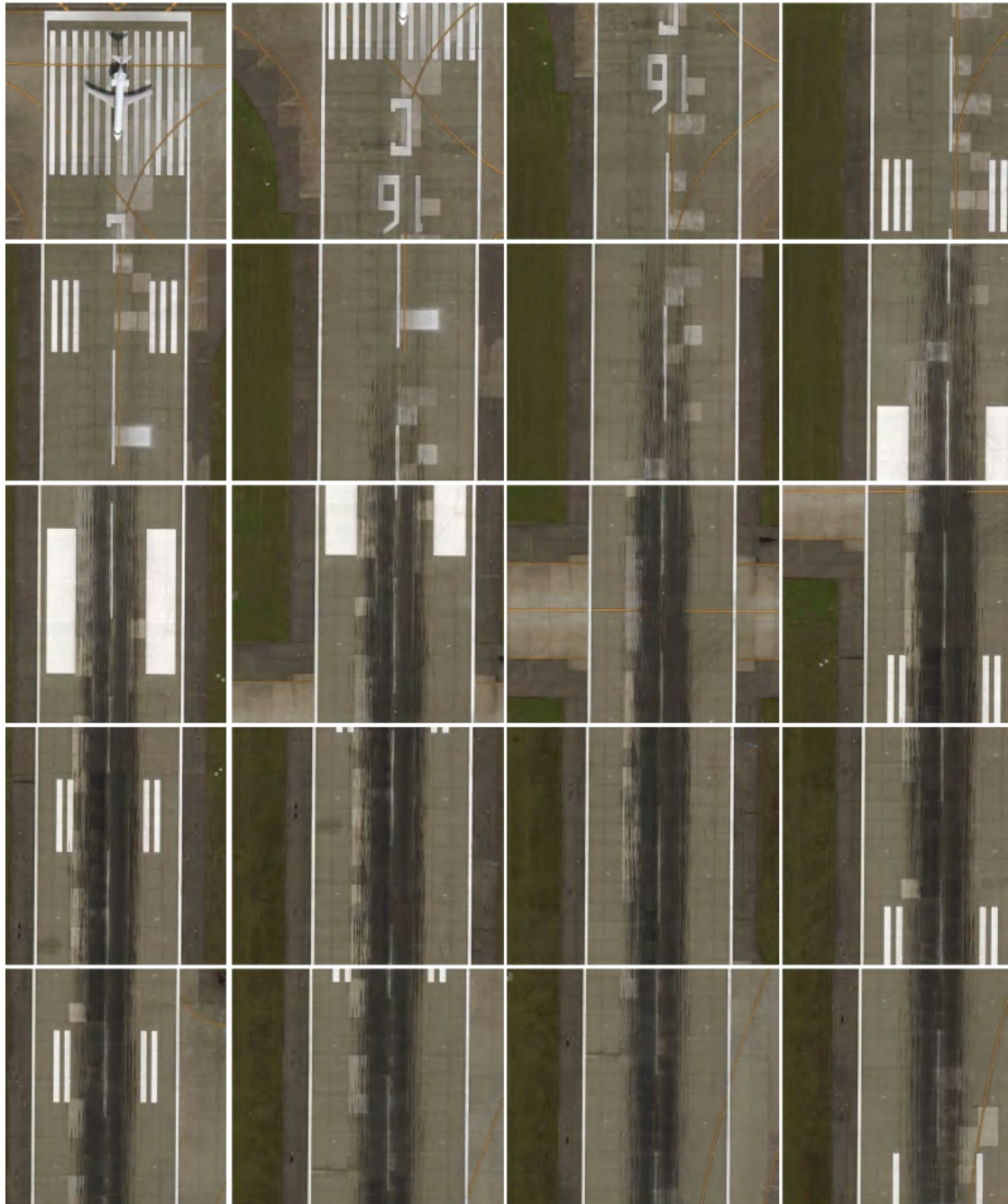


Figura II.28: Imagen panorámica al 75 % de Superposición - dataset *Vídeo Lago*



Figura II.29: Imagen panorámica al 55 % de Superposición - dataset *Vídeo Mural*



**Figura II.30:** Dataset del aeropuerto de Seattle



**Figura II.31:** Panorámica del aeropuerto de Seattle (reconstrucción de la ortofoto)