

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías
Industriales

Posicionamiento de Drones para Captura de Áreas de
Interés y Formación de Panorámicas

Autor: Álvaro Raya Navarro

Tutor: José M^a Maestre Torreblanca

Dep. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Posicionamiento de Drones para Captura de Áreas de Interés y Formación de Panorámicas

Autor:

Álvaro Raya Navarro

Tutor:

José M^a Maestre Torreblanca

Profesor titular

Dep. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Proyecto Fin de Carrera: Posicionamiento de Drones para Captura de Áreas de Interés y Formación de Panorámicas

Autor: Álvaro Raya Navarro

Tutor: Jose M^a Maestre Torreblanca

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2013

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

A las personas que han creído en mí y me han apoyado en todo momento, haciendo más llevaderos los baches en el camino. Por ellos he llegado hasta aquí y por ellos seguiré hacia donde me lleve la Ingeniería.

Resumen

El objetivo de este proyecto es la monitorización de un entorno de interés mediante el uso de un grupo de dispositivos con cámara incorporada. El trabajo incluye desarrollar función de distribución que determina la posición de los mismos y el tratamiento de las imágenes necesario para unir sus capturas formando la imagen panorámica.

Al no disponer de las herramientas necesarias, el alcance de este estudio se limita a un entorno simulado, donde se supone un movimiento tridimensional de las cámaras y con posibilidad de giro en su orientación horizontal.

El código usado para implementar los algoritmos ha sido desarrollado en el entorno de MATLAB, usando algunas de las funciones que tiene incorporadas.

Abstract

The purpose of this project is to monitor a region of interest using a group of devices equipped with a camera. The work includes the development of a distribution function that position the agents for taking the picture, and the image processing methods needed to overlap all the pictures taken in order to make the panoramic image.

In the absence of devices to make a real implementation, the project was performed with a simulation where the agents move in three dimensions and can change their horizontal orientation.

The code used has been developed in MATLAB, using some of its built-in functions.

Índice

Agradecimientos	i
Resumen	iii
Abstract	v
Índice	vii
Índice de Tablas	ix
Índice de Figuras	xi
Acrónimos y siglas	xiii
1 Introducción	1
2 Unión de Imágenes	3
2.1. <i>Extracción de puntos singulares y su descriptor</i>	4
2.1.1 SUSAN y Fast	5
2.1.2 SIFT	5
2.1.3 SURF	6
2.1.4 Conclusión	6
2.2. <i>Búsqueda de puntos semejantes</i>	6
2.3. <i>Estimación de la transformación</i>	7
2.3.1 RANSAC	7
2.3.2 MSAC	7
2.4. <i>Superposición de imágenes</i>	7
2.5. <i>Guía de funciones y elementos de MATLAB</i>	8
2.6. <i>Prueba de funcionamiento</i>	9
3 Distribución	13
3.1. <i>Elementos</i>	14
3.2. <i>Método de optimización</i>	15
3.2.1 Primera parte	15
3.2.2 Segunda parte	15
3.3. <i>Algoritmo</i>	16
3.3.1 Elementos externos	16
3.3.2 Inicialización	16
3.3.3 Primer agente	17
3.3.4 Distribución agentes	17
3.3.5 Presentación del resultado	19
3.4. <i>Otros métodos de optimización</i>	20
3.4.1 Optimización en un paso	20
3.4.2 Búsqueda por contorno	20
4 Simulación	21
4.1. <i>Función de captura</i>	21
4.1.1 Entradas	22
4.1.2 Funcionamiento	22

4.1.3	Ejemplos de funcionamiento	22
4.2.	<i>Consideraciones en un escenario real</i>	27
4.2.1	Iluminación	27
4.2.2	Objetos a distancia intermedia	27
4.2.3	Distintos niveles de profundidad	28
4.2.4	Ruido y enfoque	28
5	Análisis de resultados	29
5.1.	<i>Primera prueba</i>	29
5.1.1	Entradas	30
5.1.2	Resultados	31
5.2.	<i>Segunda prueba</i>	34
5.3.	<i>Tercera prueba</i>	36
5.3.1	600x600	37
5.3.2	300x300	38
5.3.3	150x150	40
5.3.4	Comparación	41
6	Conclusión	43
6.1.	<i>Solapamiento</i>	43
6.2.	<i>Distribución</i>	43
6.3.	<i>Posibles continuaciones del trabajo</i>	44
	Referencias	45
	Anexo A: Código	47

Índice de Tablas

Tabla 4-1. Posición de las cámaras en las pruebas.	23
Tabla 4-2. Tiempos de ejecución variando la calidad.	26
Tabla 4-3. Entradas para la prueba.	27
Tabla 5-1. Parámetros de optimización.	29
Tabla 5-2. Parámetros entrada de la prueba 1.	30
Tabla 5-3. Relaciones de los elementos	30
Tabla 5-4. Configuración de los dispositivos de la primera prueba.	34
Tabla 5-5. Configuración de los dispositivos de la segunda prueba.	36
Tabla 5-6. Tiempos de ejecución de la tercera prueba en función de la calidad.	41

Índice de Figuras

Figura 2-1. Representación modelo <i>pinhole</i> tridimensional (a) y bidimensional (b).	3
Figura 2-2. Imágenes a solapar.	9
Figura 2-3. Imagen 1 incluida en el panorama.	10
Figura 2-4. 500 puntos más singulares de la panorámica.	10
Figura 2-5. 500 puntos más singulares de la imagen 2.	11
Figura 2-6. Relación de puntos semejantes.	11
Figura 2-7. Imagen 2 transformada.	12
Figura 2-8. Resultado de la unión.	12
Figura 3-1. Diagrama de bloques del algoritmo de distribución.	16
Figura 3-2. Esquema del modelo.	18
Figura 3-3. Posición horizontal de la cámara.	18
Figura 3-4. Posición horizontal de los extremos y distancia al foco.	18
Figura 3-5. Proyección de la distancia a los extremos de la imagen en el eje de la cámara.	19
Figura 3-6. Longitud vertical.	19
Figura 4-1. Imagen usada como referencia y puntos donde se centran las pruebas	23
Figura 4-2. Resultados para la prueba variando el alejamiento	24
Figura 4-3. Resultados de la prueba variando el ángulo y la imagen transformada tomando como referencia la original.	25
Figura 4-4. Resultados para la prueba variando la calidad.	26
Figura 4-5. Resultado de la prueba.	27
Figura 5-1. Imagen tomada como panorama de estudio.	30
Figura 5-2. Mapa de interés empleado en la primera y segunda prueba.	30
Figura 5-3. Captura del primer agente y actualización de la panorámica y el mapa de interés.	31
Figura 5-4. Captura del segundo agente y actualización de la panorámica y el mapa de interés.	31
Figura 5-5. Captura del tercer agente y actualización de la panorámica y el mapa de interés.	32
Figura 5-6. Captura del cuarto agente y actualización de la panorámica y el mapa de interés.	32
Figura 5-7. Captura del quinto agente y actualización de la panorámica y el mapa de interés.	33
Figura 5-8. Resultado de la primera prueba.	33
Figura 5-9. Capturas de los nueve agentes de la segunda prueba.	34
Figura 5-10. Resultado de la segunda prueba.	35
Figura 5-11. Mapa de interés resultante tras la segunda prueba.	35
Figura 5-12. Mapa de interés para la tercera prueba.	36
Figura 5-13. Capturas para la tercera prueba con un tamaño de 600x600.	37

Figura 5-14. Resultado para la tercera prueba con un tamaño de 600x600.	37
Figura 5-15. Mapa de interés resultante para la tercera prueba con tamaño de 600x600.	38
Figura 5-16. Capturas para la tercera prueba con un tamaño de 300x300.	38
Figura 5-17. Resultado para la tercera prueba con un tamaño de 300x300.	39
Figura 5-18. Mapa de interés resultante para la tercera prueba con tamaño de 300x300.	39
Figura 5-19. Capturas para la tercera prueba con un tamaño de 150x150.	40
Figura 5-20. Resultado para la tercera prueba con un tamaño de 150x150.	40
Figura 5-21. Mapa de interés resultante para la tercera prueba con tamaño de 300x300.	41

Acrónimos y siglas

UAV	Unmanned Aerial Vehicle
SUSAN	Smallest Univalve Segment Assimilating Nucleus
USAN	Univalve Segment Assimilating Nucleus
SIFT	Scale Invariant Feature Transform
DoG	Difference-of-Gaussian
SURF	Speeded-Up Robust
RANSAC	RANdom SAmple Consensus
M-SAC	M-estimator SAmple Consensus
u.l.	Unidad de longitud genérica

1 INTRODUCCIÓN

El uso de robots autónomos para monitorizar una zona señalada es una función útil para distintas tareas, por ejemplo el control de zonas de cultivo, vigilar incendios o incluso fines militares. En el presente, el método más empleado para conseguir dicho fin consiste en usar un único dispositivo, generalmente un UAV, que realiza un recorrido preestablecido tomando capturas en puntos clave que posteriormente se unen en una panorámica. El principal problema de este método es que se limita a tomar imágenes, sin tener la capacidad de mantener la cámara fija para tomar vídeos para poder implementar esta función se precisan más de un dispositivo con cámara que mantengan su posición fija.

Este proyecto se apoya en el realizado por T. Arauz [1], extendiendo sus resultados al caso tridimensional e incluyendo la respuesta ante posibles cambios de tamaño en las imágenes. El trabajo aquí desarrollado se divide en dos bloques principales: la formación de las panorámicas y la elección de la posición de los agentes.

La formación de panorámicas ha sido objeto de interés desde hace un tiempo. Actualmente muchos de los dispositivos que permiten realizar fotografías incluyen una función de montaje de panorámicas. No obstante, los algoritmos que emplean no son muy robustos, exigiendo unos requisitos en la posición de la cámara para tomar las capturas, como mantener la posición de la cámara fija y pivotar sobre un eje. Para este trabajo se necesita uno más complejo, pues se considera que las cámaras no tienen que tener un patrón en la relación entre su disposición. Este algoritmo se centra en la búsqueda de semejanzas entre distintas imágenes que capturan el mismo área, proceso que tiene que tomar características invariantes frente a cambios de escala y giros.

Como enlace con la distribución, se obliga a la cámara que se va a colocar a capturar una zona del área ya capturada para realizar la correcta unión de las imágenes. El grado mínimo de solapamiento entre imágenes será estudiado más adelante, y dependerá del método que se emplee.

Dentro de la distribución existen diversas configuraciones, desarrolladas en el capítulo correspondiente, para las cuales se ha estudiado la relación entre el grado de optimización de su respuesta y el tiempo de ejecución. Las decisiones previas que se han de tomar para diseñarla incluyen determinar un mapeado de la zona, que indique el interés de cada área, y los grados de libertad del movimiento de las cámaras, que se ha limitado a una traslación tridimensional y un único giro en la orientación horizontal de la misma. Estas características son fácilmente alcanzables en los dispositivos, ya que limitar a un único giro minimiza el número de motores necesario para mover la cámara, incluso suprimirse si el propio giro del dispositivo equivale al giro horizontal de la cámara.

Finalmente se ha tenido que diseñar una función que simule un entorno de trabajo, puesto que no se disponía de dispositivos suficientes para probarlo en una situación real. Esta simulación está programada de tal modo que, introduciendo los datos de la posición de las cámaras, devuelva la captura que obtendría cada una. Se ha definido un panorama plano, para evitar elevar la complejidad de esta función. Esta simplificación hace que no se consideren ciertos cambios de profundidad que podrían darse en una situación real.

El programa escogido para desarrollar el proyecto ha sido MATLAB. Este entorno destaca por ser cómodo e intuitivo tanto para la programación del funcionamiento como para la propia simulación del mismo. Además, presenta una amplia biblioteca de funciones de tratamiento de imagen que son muy útiles para el desarrollo de todo el proceso de unión de fotografías. Como añadido tiene la opción de medir el tiempo de ejecución de cada parte, lo que permite estudiar cómo afectan los cambios realizados al coste computacional.

Al trabajar con una simulación, no se ha dedicado ningún apartado al método de control de los dispositivos, pero hay artículos que desarrollan unos controladores que pueden ser empleados para el propósito de este trabajo, que precisa de una colocación en la que la exactitud es muy influyente a la hora de tomar la captura

deseada. También tiene que determinarse el tipo de comunicación que permita el envío de órdenes e imágenes con dispositivos ampliamente separados entre ellos. Esto puede suponer una restricción del movimiento de los dispositivos, evitando que se alejen demasiado entre ellos.

El contenido se distribuye en cinco capítulos, sin contar la introducción, y un anexo:

- **Unión de imágenes:** se explica el tratamiento de imagen empleado para la superposición de las capturas.
- **Distribución:** se diseña el algoritmo de posicionamiento de los agentes, construyendo para ello una función de coste.
- **Simulación:** se explica la función creada para simular el proceso, y las consideraciones que toma frente a un escenario real.
- **Análisis de resultados:** se llevan a cabo varias pruebas para comprobar el funcionamiento de las funciones diseñadas.
- **Conclusión:** se exponen las ideas extraídas de los resultados y las posibles ampliaciones que podrían realizarse en un futuro.
- **Anexo:** contiene el código comentado empleado para implementar el proyecto.

- **Locales:** sirven como descriptor de una zona en concreto de la imagen. Dichas áreas están centradas en los puntos de interés de la imagen, que normalmente equivalen a los puntos singulares, por lo que ambos procesos se consideran complementarios. De hecho, muchos métodos de extracción de puntos singulares incluyen la elaboración del vector de características. Estos vectores son usados principalmente para determinar puntos equivalentes entre distintas imágenes y suelen clasificarse según su variabilidad frente a cambios en la iluminación, escala o giro.

Una vez se han definido estos conceptos básicos, se expone el esquema del proceso de solapamiento de imágenes que se ha implementado para cumplir los objetivos de este proyecto. Aunque en los distintos estudios consultados hay variaciones entre los distintos métodos utilizados para afrontar cada parte, todos ellos comparten el mismo esquema de trabajo. Los puntos que lo componen son:

- Lectura de las capturas y su tratamiento antes de la realización del proceso.** La necesidad de aplicar un procesado previo a la imagen depende de los métodos que se vayan a emplear para los siguientes pasos. Esto se debe a que, en función del que se utilice, pueden ser necesarios algunos ajustes, como, por ejemplo, aplicar un filtro de ruido o que la imagen a estudiar esté en escala de grises.
- Localización de los puntos singulares de las imágenes.** Se determinan las coordenadas de los puntos que destaquen de su entorno en aquellas características que considere la función usada.
- Extracción de los descriptores.** Se construye un vector de características por cada punto singular que contenga la información del entorno del mismo. Como se ha indicado antes, este paso y el anterior se realizan de forma simultánea en la mayoría de los métodos que se han estudiado.
- Búsqueda de puntos similares entre las imágenes.** Se comparan los puntos de las imágenes en función de sus descriptores asociados y se marcan como semejantes aquellos cuya diferencia entre características no supere un umbral.
- Estimación de la relación de transformación.** Empleando la relación entre las parejas de puntos semejantes, se calcula la transformada que requiere una de las imágenes para que su perspectiva coincida con la otra.
- Unión de las imágenes.** Una vez que las imágenes coinciden se superponen, dando lugar a la imagen panorámica. Este proceso puede incluir un filtro de intensidad para reducir el contraste en el límite entre ambas imágenes.

Por último cabe destacar que estos procesos tienen otras aplicaciones aparte de la formación de imágenes panorámicas. Por ejemplo, en [4] se explica un método que permite transcribir un discurso sin importar el audio. Esto lo consigue comparando los puntos singulares y sus descriptores con una plantilla para localizar los labios de la persona que esté dando el discurso. Una vez que se tienen localizados se emplearía un algoritmo de identificación de palabras mediante la lectura de labios. También tiene aplicación en el ámbito de la medicina, sirviendo de apoyo en el diagnóstico de radiografías o ecografías. Un ejemplo se expone en [5], donde se emplea una base de datos que recoge las características que presentan las enfermedades en las radiografías para poder seleccionar, de forma automática, aquellas radiografías que tengan semejanzas con alguna de ellas.

2.1. Extracción de puntos singulares y su descriptor

Como se ha comentado, es necesario determinar los puntos característicos en las imágenes que permitan relacionarlas. Para ello se debe localizar los puntos en la imagen y asignarle unos parámetros que sirvan para definirlos y diferenciarlos entre ellos. Dado el objetivo del proyecto, donde las cámaras captarán un entorno desde distintas posiciones y orientaciones, se requiere que los parámetros empleados sean invariantes frente a giros, desplazamientos y cambios de escala. También deberá responder correctamente a variaciones de intensidad luminica entre imágenes, ya que en función de la separación de las cámaras puede que la iluminación varíe.

Existen estudios que comparan distintos métodos para realizar estas funciones, como [6], [7] y el mismo proyecto del que parte este [1]. Aun así, al tener más variaciones de ángulo y escala entre las capturas, se ha realizado una breve comparación entre las características de los métodos más recomendados, para comprobar si las conclusiones son aplicables a este proyecto.

2.1.1 SUSAN y Fast

El método SUSAN (*Smallest Univalued Segment Assimilating Nucleus*) sirve para localizar los bordes y esquinas de una imagen. Para lo necesario en el proyecto se estudia especialmente la respuesta buscando esquinas. El procedimiento a seguir viene detallado en [8], y, en resumen, consiste en:

- A. Colocar una máscara redonda alrededor de cada píxel, determinados núcleos.
- B. Seleccionar, dentro de la máscara, los píxeles que tienen la misma intensidad que el centro. Este área se denomina USAN (*Univalued Segment Assimilating Nucleus*).
- C. Calcular el centro de masas de dicha área.
- D. Comprobar si el USAN ocupa menos que un porcentaje, diferente si quieren identificarse bordes (~50%) o esquinas (~25%). Se establece un mínimo para evitar el ruido.
- E. Para las esquinas, extraer los mínimos locales hasta tener el número de esquinas deseado. Para ello se puede atender al valor de la intensidad o al tamaño de la USAN, dependiendo de si se pretende dar más peso a la precisión o a la velocidad.
- F. Se descartan los puntos cuyo centro de masas esté muy próximo al núcleo.

Este método es efectivo a la hora de determinar las esquinas, pues proporciona una localización bastante exacta. Además no necesita un filtro previo para eliminar el ruido, pues en el requerir un mínimo de USAN ya descarta los efectos del ruido. Otra cualidad positiva es la rapidez en la ejecución, dado la simpleza de las operaciones realizadas.

No obstante, este proceso solo extrae como información del punto la distribución del USAN y la ubicación del centro de masas. Estas características no son fiables a la hora de buscar puntos en común, además de no ser invariantes al cambio de escala, por lo que se requeriría de un proceso complementario para determinar otras características de las esquinas ya localizadas.

El método Fast [9] emplea los mismos principios que SUSAN, pero añadiendo un descriptor a cada punto en base a un vector formado por el borde de la máscara y relacionando puntos semejantes. Sin embargo, al depender del radio de la circunferencia, tampoco es invariante frente a cambios de escala, por lo que no es útil para el proyecto.

2.1.2 SIFT

El método SIFT (*Scale Invariant Feature Transform*) es un método más complejo para la extracción de puntos singulares basado en la diferencia de convoluciones con una matriz gaussiana que sea invariante frente a escalas y orientación. Este proceso, desarrollado por Lowe en [10], incluye tanto la localización de los puntos clave como la definición del descriptor invariante asociado.

Primero se aplica la convolución por cada punto, se reduce la escala de la matriz gaussiana, y se vuelve a aplicar. Este paso se repite tantas veces como escalas de la imagen se quieran comprobar, generando un equivalente de la imagen para cada escala. A continuación se calcula la diferencia del resultado obtenido en una escala con la inmediatamente superior e inferior, dando lugar así a un número más reducido de datos que permite un menor tiempo de muestreo al compararlas. A este proceso se le denomina DoG (*Difference-of-Gaussian*). Una vez obtenidos los datos por punto se calculan los mínimos y máximos para determinar los puntos característicos. Para ello se compara su valor de DoG, en la escala inferior, con el de los puntos de su vecindad, y con los puntos de vecindad de las escalas superiores. Con esto se consigue determinar los extremos y la escala que le corresponde. Finalmente para seleccionar el punto se eliminan aquellos que no presenten estabilidad y se emplea el hessiano para eliminar los puntos que se encuentran en bordes en lugar de esquinas.

A continuación se determina la orientación invariante de los puntos seleccionados. Para ello se aplica el filtro de la matriz gaussiana escalada acorde con el punto y se calcula los gradientes direccionales de los puntos en torno a este. El mayor de estos gradientes es el que determina la dirección.

Por último se construye el descriptor basado en los gradientes locales calculados tomando la orientación hallada como referencia. Además se adaptan para que tengan una respuesta aceptable a pesar de posible ruido o cambios leves en la imagen.

Este proceso presenta una exactitud elevada y cumple el requisito indispensable de presentar invariancia frente a cambios de escala y giros. No obstante presenta un coste computacional elevado debido a los múltiples

muestreos que hay que realizar para calcular las convoluciones y, especialmente, para buscar los puntos extremos.

2.1.3 SURF

El método SURF (*Speeded-Up Robust*) fue desarrollado por Bay en [11] como una posible mejora del método SIFT. Con él se busca definir un descriptor que sea invariante frente a cambios de escala y orientación, como el método anterior, pero mejorando el tiempo de respuesta. Para ello emplea imágenes integrales, que resultan más económicas en cuanto al tiempo. Estas se calculan formando una imagen de las mismas dimensiones que la que se está estudiando y colocando en cada píxel la suma de intensidades de los píxeles contenidos en el rectángulo de la imagen original comprendido entre el origen de coordenadas y el píxel en cuestión.

El detector de puntos (denominado *Fast-Hessian*) emplea la misma matriz hessiana para determinar tanto la localización como la escala de los puntos singulares. Los filtros se aplican sobre las imágenes integrales y, para variar la escala de estudio, basta con cambiar la escala del filtro que se use, sin necesidad de crear imágenes equivalentes con escalas reducidas, lo que supone una reducción notable en el coste computacional. Hay que añadir que el paso entre escalas debe estar medido acorde a ellas.

Para el descriptor primero se tiene que determinar la orientación del punto. Se estima estudiando la respuesta a la transformación Haar-wavelet para la dirección horizontal y vertical aplicada en un círculo alrededor del punto de diámetro proporcional a la escala que se haya asignado al mismo. Tomando el peso de la respuesta en ambas direcciones, se forman vectores y se extrae la orientación del vector mayor. Hay que destacar que para escalas muy reducidas y muy elevadas se obtienen orientaciones inestables.

Una vez se tiene la orientación, se procede a extraer el descriptor. Para ello se construye un cuadrado alineado con la dirección seleccionada y de tamaño proporcional a la escala. Este cuadrado se divide en 16 regiones y, para cada una, se calcula la suma de las respuesta a la función wavelet en las direcciones perpendicular y paralela de la principal. Como resultado se tiene para cada subregión un vector de cuatro componentes que son el sumatorio en cada dirección y el sumatorio de los valores absolutos en cada dirección. Con ello se consigue un resultado invariante frente a cambios de intensidad.

Finalmente, en el artículo se incluye una comparación del funcionamiento frente al método SIFT en el que se puede observar que el SURF presenta un funcionamiento más exacto y con un coste computacional menor.

2.1.4 Conclusión

Una vez realizada la comparativa se ha comprobado que el método que mejor se adapta a las exigencias de este proyecto es el SURF. Esta decisión, compartida por la mayoría de los artículos consultados, ha sido tomada en base a la capacidad de extraer puntos singulares y asociarle descriptores invariantes frente a distintas perspectivas de las cámaras de manera más efectiva y rápida que los demás.

2.2. Búsqueda de puntos semejantes

Una vez se han extraído los puntos de interés y las características locales, se realiza el proceso de búsqueda de puntos similares. Para ello se toman los puntos de ambas imágenes y se estudia su concordancia con los demás. Para determinar las parejas se calcula la diferencia entre los vectores descriptores de cada punto de una imagen y cada punto de la otra, y se seleccionan aquellas parejas cuyo valor de la diferencia sea inferior a un límite. Este umbral se selecciona de tal manera que se eviten los falsos positivos.

Cabe destacar que aplicando este proceso puede que un punto se asemeje a más de uno en la otra imagen. Para corregir esto las funciones incluyen una opción que establece como restricción que solo pueda haber un punto semejante por cada punto, seleccionando aquel cuya diferencia entre descriptores sea menor.

2.3. Estimación de la transformación

Una vez se tiene la relación entre puntos, se procede al cálculo de una relación de transformación que, aplicándose a la imagen que queremos editar, se consiga que sus puntos se correspondan a los de la imagen base. Para que dicha transformación de un resultado aceptable se requiere un mínimo de puntos relacionados entre las imágenes, para así determinar con exactitud los cambios de escala y de orientación, factor a tener en cuenta a la hora de determinar el grado de solapamiento de las capturas.

La función elegida para realizar la estimación emplea el algoritmo MSAC (*M-estimator Sample Consensus*), que es una variación del algoritmo RANSAC (*RANdom SAMple Consensus*). Se va a explicar brevemente el funcionamiento de ambos.

2.3.1 RANSAC

El funcionamiento de este algoritmo viene detallado por Martin A. Fischler y Robert C. Bolles en [12]. Aquí se va a hacer un resume para comprenderlo. Para empezar se parte de una situación en la que se posee un conjunto de puntos y se busca calcular los parámetros del modelo que permita relacionarlos, en este caso sería un conjunto de parejas de puntos y el modelo sería la transformada. Los pasos son:

- A. Determinar el mínimo número de puntos necesario para determinar los parámetros del modelo.
- B. Formar, tomando puntos de manera aleatoria, un subconjunto que tenga ese mínimo número de componentes.
- C. Calcular los parámetros del modelo con dicho subconjunto.
- D. Aplicar el modelo al resto de puntos del conjunto y calcular cuántos de ellos dan un resultado válido, dentro de un margen de error preseleccionado.
- E. Comprobar si el número de resultados válidos supera un límite determinado.
 - a. Si lo supera: termina el algoritmo y el resultado es el modelo obtenido.
 - b. Si está por debajo: se vuelve al paso B, formando un nuevo subconjunto con otros puntos al azar. Este proceso tiene un máximo de iteraciones, si se supera, se marca como resultado fallido.

2.3.2 MSAC

Como se expone en [13], el método RANSAC presenta una deficiencia en su función de coste. Esto se debe a que sólo cuenta aquellos puntos que quedan fuera de la transformada. Para mejorar esto, el algoritmo MSAC incluye en el coste el error que tengan los puntos que si se consideran válidos, es decir, valora de forma positiva el grado de exactitud que presenten dichos puntos. Con esto se consigue una función de coste más completa sin aumentar el coste computacional.

2.4. Superposición de imágenes

Una vez se ha obtenido la relación de transformación se aplica a la captura que se quiere unir a la imagen panorámica. Para poder determinar la posición que tendría la imagen dentro de la panorámica, se le añade a esta transformación un fondo de las mismas dimensiones que esta. Una vez se tienen ambas imágenes del mismo tamaño, se estudia la forma de unir las.

Un objeto de MATLAB permite aplicar distintos tipos de solapamiento. Uno de los métodos que ofrece combina la intensidad de ambas con un grado de opacidad seleccionado, quedando solo con la intensidad normal la parte común. En este caso no es de utilidad pues las dos imágenes tienen un fondo negro que oscurecería las zonas que no fuesen coincidentes.

Dentro del mismo objeto está otra opción que se adapta mejor a las necesidades del proyecto. Consiste en colocar la captura que se quiere unir encima de la imagen panorámica, reemplazando todos los píxeles que ocupe. Para determinar qué área es reemplazada, se incluye en las entradas de la función una máscara binaria que determina el área que ocupa la nueva imagen. Esta máscara se extrae de forma trivial, aplicando la misma transformada que se ha empleado con la imagen a una matriz de las mismas dimensiones con todos los componentes a uno. Este método es el que se ha escogido.

No obstante el método seleccionado tiene el inconveniente de poder presentar cambios bruscos de intensidad en los límites de las imágenes. Esto se puede producir porque las capturas, al tener las cámaras distintas posiciones y orientaciones, pueden captar distintos niveles de intensidad para la misma zona del plano. La mejor forma de corregir esto es aplicar un filtro de suavizado a la imagen final, cuidando que la pérdida de calidad no sea elevada.

2.5. Guía de funciones y elementos de MATLAB

En este apartado se van a explicar brevemente los elementos y funciones propios del repertorio de MATLAB que se han empleado para realizar la función de solapamiento de imagen. Aquí se va a exponer su utilidad, entradas, salidas y opciones que se han modificado, las opciones que no se nombren tendrán su valor por defecto. En la página es.mathworks.com se puede encontrar la explicación, junto con ejemplos.

- ***SURFPoints***: objeto que recoge la localización de los puntos y las características necesarias para poder extraer el descriptor. Sirve de puente entre las dos partes del método SURF.
- ***detectSURFFeatures***: aplica el método SURF para obtener los puntos característicos.
 - Entradas: la imagen, en escala de grises, en la que se quieren localizar los puntos.
 - Salidas: elemento tipo *SURFpoint* con la información obtenida.
 - Opciones:
 - *MetricThreshold*: valor que determina el grado de singularidad que debe tener un punto para considerarse como característico. Con un valor menor se consiguen más puntos pero con un descriptor menos preciso.
 - *NumScaleLevels*: determina los niveles de escala que se consideran en el descriptor. Se le da un valor alto dado que las cámaras tendrán distinta escala, aunque aumente el coste computacional.
- ***extractFeatures***: extrae información necesaria de los objetos *SURFpoints*.
 - Entradas: el objeto *SURFpoints* del que se quiere conocer la información y la imagen a la que pertenece dicho objeto.
 - Salidas: vector de características y la localización de los puntos asociados. Estos puntos pueden no ser todos los incluidos en la entrada, porque puede que no sea posible construir el descriptor en algunas ocasiones, como cuando se encuentran muy cerca del límite de la imagen.
 - Opciones: se dejan las opciones por defecto, para que sean invariantes frente a rotación y tenga una velocidad aceptable.
- ***matchFeatures***: encuentra los puntos semejantes entre las imágenes.
 - Entradas: Los descriptores de ambas imágenes.
 - Salidas: matriz bidimensional con los índices de los puntos similares en la correspondiente lista de puntos.
 - Opciones:
 - *MatchThreshold*: determina el porcentaje máximo de distancia para que dos puntos se consideren similares. Se busca un valor reducido, para evitar falsos positivos, pero que sea capaz de extraer suficientes parejas para estimar la transformada.
 - *MaxRatio*: elemento que sirve para eliminar parejas de puntos similares que no siguen el patrón del resto. Su valor se disminuirá con forme aumente el valor del anterior, para eliminar falsos positivos disonantes.
 - *Unique*: se activa esta opción para que un punto solo pueda tener un único semejante en la otra imagen.
- ***estimateGeometricTransform***: estima la transformación que hay que aplicar a una de las imágenes para que tenga igual distribución que la otra.
 - Entradas: coordenadas de los puntos similares de ambas imágenes. Primero las de la imagen a transformar y luego las de la que se considerará la base. También se introduce el tipo de transformada que se desea, y que influye en el mínimo número de puntos similares que se necesitan.
 - Salidas: transformación geométrica.

- Opciones:
 - *MaxNumTrials*: número máximo de pruebas a realizar. Si se supera este límite el proceso se interrumpe dando error. Se le da un valor superior al predeterminado, con cuidado de no elevar mucho el coste computacional.
 - *Confidence*: grado de correspondencia que tiene que tener la transformada con las parejas de puntos similares. Se le da un valor próximo al 100%.
 - *MaxDistance*: máxima distancia en píxeles entre un punto de la imagen que se usa como base y la proyección del punto similar en la imagen transformada. Se le da un valor pequeño pero evitando que se formen singularidades por culpa de este.
- ***imwarp***: aplica la transformada calculada a la imagen.
 - Entradas: la imagen y la relación de transformación.
 - Salidas: imagen transformada.
 - Opciones:
 - *OutputView*: se utiliza para tener un marco global del tamaño de la panorámica.
- ***visionAlphaBlender***: objeto que determina la forma de solapar las imágenes. Las opciones que se le asignan son:
 - *Operation*: selecciona el método de solapamiento. Se marca como *Binary mask*.
 - *MaskSource*: se marca como *Input port* para poder incluir la máscara como entrada de la función.
- ***step***: llama al objeto tipo *vision.AlphaBlender* para ejecutar su función asociada.
 - Entradas: el objeto, las dos imágenes que se quieren superponer y la máscara de solapamiento.
 - Salida: la imagen resultante.

2.6. Prueba de funcionamiento

Para comprobar el funcionamiento de la función se han tomado fotografías de un entorno real, desde distintos puntos y alejamiento. A la que se use como base se le ha añadido un marco de fondo negro, para representar la imagen en la que se está construyendo la panorámica. Experimentalmente se han comprobado los efectos que tienen las variaciones en las opciones de cada función, determinando los valores que funcionan mejor para las condiciones del proyecto.

Primero se introducen las imágenes que se quieren unir (Figura 2-2), y a una de ellas se le añade un marco negro para emplearla como base de la imagen panorámica (Figura 2-3).

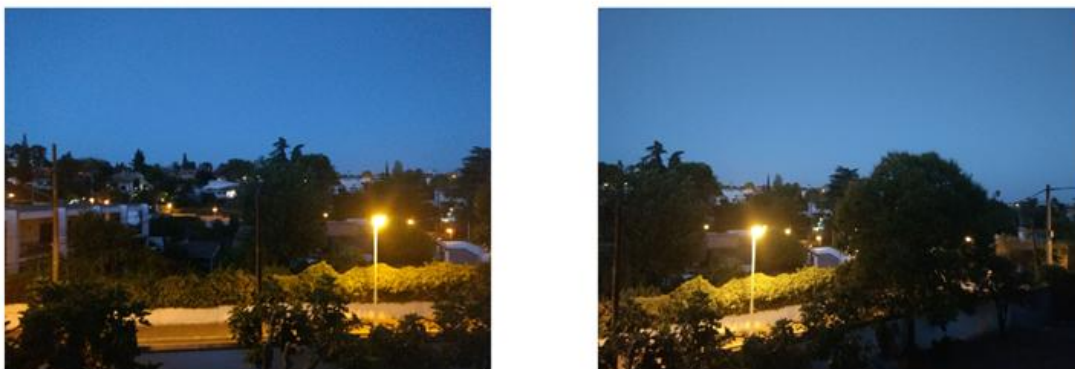


Figura 2-2. Imágenes a solapar.

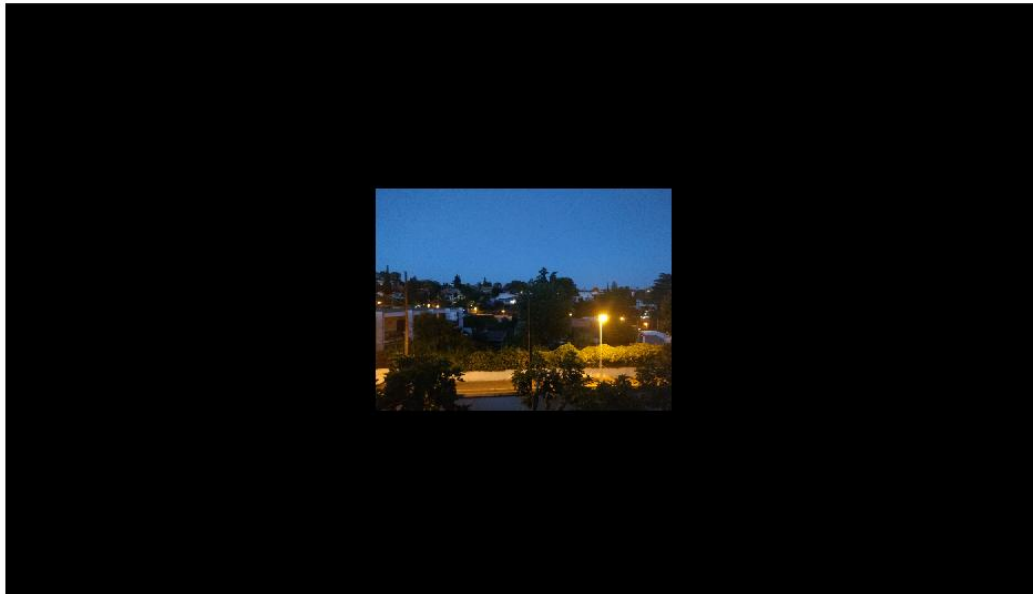


Figura 2-3. Imagen 1 incluida en el panorama.

A continuación se extraen los puntos característicos y sus descriptores. En la Figura 2-4 y Figura 2-5 se muestran los puntos que se han hallado junto con una circunferencia que indica su escala. Aunque solo se marquen los 500 puntos que presentan mayor singularidad, para el proceso se ha extraído un número mayor, del orden de los 2000.

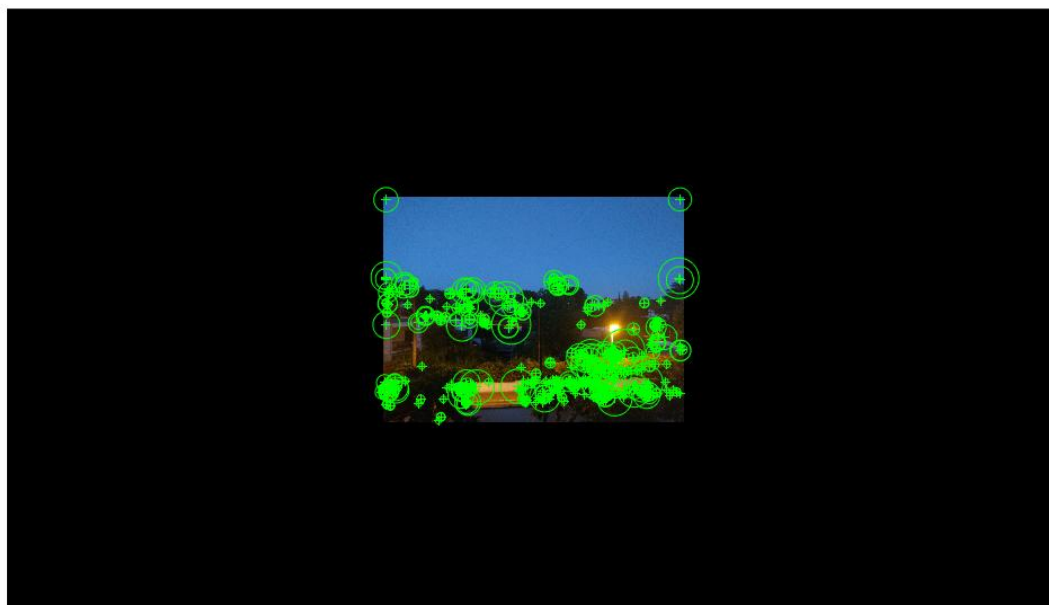


Figura 2-4. 500 puntos más singulares de la panorámica.



Figura 2-5. 500 puntos más singulares de la imagen 2.

Una vez se tienen los puntos, se procede a buscar semejantes. En la Figura 2-6 se puede ver como se emparejan dichos puntos. Con estas parejas se estima la relación de transformación.

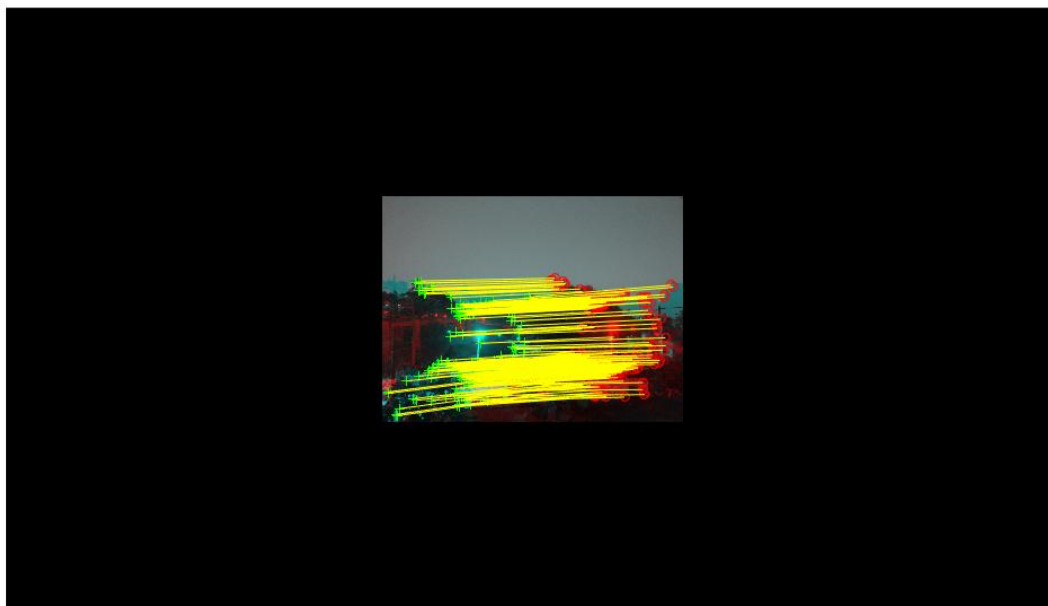


Figura 2-6. Relación de puntos semejantes.

Finalmente se aplica la transformación estimada a la imagen a solapar, añadiendo un marco de las mismas dimensiones que la imagen panorámica (Figura 2-7), y se solapan ambas, dando el resultado que se muestra en la Figura 2-8.



Figura 2-7. Imagen 2 transformada.

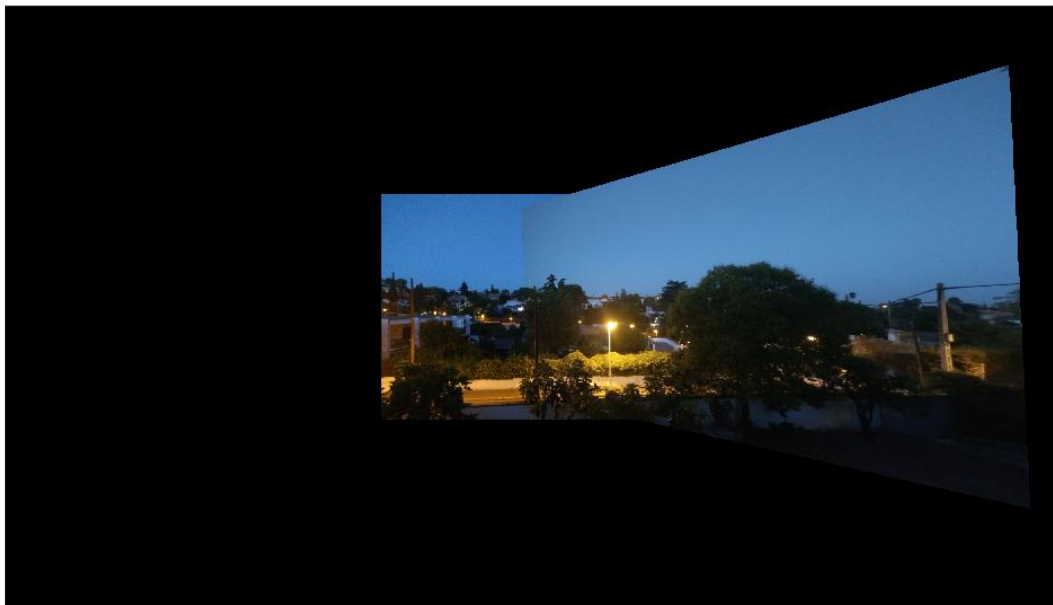


Figura 2-8. Resultado de la unión.

3 DISTRIBUCIÓN

Existen numerosas formas de afrontar una distribución de este tipo dependiendo de las preferencias que se quieran tener en cuenta. Un factor importante que debe considerarse es el tiempo de ejecución del proceso, crucial en muchos casos de actuación en tiempo real. Otro no menos importante es la aproximación al resultado óptimo, que en escasas ocasiones se consigue alcanzar en su totalidad, en el que se busca maximizar el área capturada manteniendo una calidad de imagen aceptable o un desplazamiento mínimo de los dispositivos. Por tanto se estudia la relación entre tiempo y resultado óptimo para los distintos modelos.

Atendiendo a estos dos parámetros, pueden definirse dos modelos generales de optimización: uno que trabaje con todos los agentes como conjunto, explorando todas las posibles combinaciones entre ellos; y otro que los distribuya uno por uno, buscando un óptimo local por cada agente. Se exponen a continuación las características de cada uno que se han considerado para hacer una elección.

Global.

Consiste en desarrollar un sistema a optimizar que considere todos los agentes de forma simultánea, estudiando la combinación de cada una de las posibles posiciones de todos los agentes al mismo tiempo.

La ventaja es que se consigue la distribución más óptima alcanzable en función de los parámetros que se seleccionen en la función de coste, ya que se consideran todos los casos posibles.

La principal desventaja que tiene este método es que presenta un elevado coste computacional, que conlleva un elevado tiempo de ejecución. Esto se debe a las múltiples configuraciones posibles que tiene el sistema al tener cuatro grados de libertad por cada dispositivo, que son movimiento tridimensional y giro horizontal. Este elevado coste se aprecia con mayor claridad cuando se aumenta el número de agentes, que provoca un aumento exponencial del tiempo, llegando a tener tantas combinaciones posibles que el excesivo coste computacional sea inviable para un correcto funcionamiento.

Local.

Este es un método heurístico que posiciona los agentes uno a uno, considerando únicamente los que ya están colocados para determinar la posición óptima. Consiste en realizar una función de optimización por cámara que solo presenta como variables los grados de libertad de dicho agente.

Este proceso, al tener una optimización fraccionada, no explora todas las variaciones de distribución. Esto provoca que el resultado final no sea el óptimo en una consideración global, por lo que la distribución es menos acertada que la del método conjunto.

Respecto al coste computacional, cada optimización local solo tiene cuatro grados de libertad, y aumentar el número de agentes solo aumenta el número de funciones de optimización a realizar, lo que aporta menos carga computacional que añadir más variables en una función global. Por tanto, este método garantiza más rapidez, especialmente cuando se aumenta el número de agentes involucrados.

Se escoge el método heurístico, dado que, para el objetivo del proyecto, interesa más un tiempo de ejecución menor, aunque el resultado esté más alejado del óptimo. Esta rapidez es de especial interés en el caso de que un agente falle, que se solucionaría reemplazando dicho dispositivo por el último que se hubiera colocado en el sistema. No obstante convendría realizar nuevamente las funciones de solapamiento de imagen para asegurar el correcto reemplazo.

Otro punto necesario para realizar la distribución es determinar cómo se indica el objetivo que se desea

capturar. Para ello se ha establecido que se proporcione como entrada del sistema un mapa que distribuya el nivel de importancia que tenga el área a estudiar. Por simplificar la distribución y evitar trabajar con imágenes en 3D, se considera que el área a estudiar va a estar contenida en un plano, cuya perpendicular corresponde a la dirección de las cámaras con ángulo cero. Dentro de este mapa existen varios grados de interés y, al incluir una nueva captura, el área a la que corresponda reduce su nivel de interés. Con ello se consigue que, para zonas con un grado alto, más de un agente busque capturarla, mejorando así la respuesta a errores, ya que si falla uno de los agentes el resto seguiría controlando ese área.

A la hora de colocar un dispositivo se tiene que valorar si la posición es alcanzable. La principal restricción viene determinada por el nivel de solapamiento de las capturas. Principalmente se busca que la cámara que se coloque capture un mínimo de puntos singulares para poder unir la imagen a la panorámica.

3.1. Elementos

- **Características de la cámara:** son los parámetros dados por el modelo de cámara. Tienen que ser facilitados como entrada del sistema:
 - **Tamaño de la imagen:** número de píxeles horizontales y verticales.
 - **Ángulos focales:** ángulos horizontal y vertical.
- **Definición de los agentes:** datos necesarios para definir la posición de los agentes. Son las salidas que se deben mandar a los dispositivos para que procedan a su desplazamiento. Incluyen:
 - **Coordenadas cartesianas (X, Y, Z):** las coordenadas X e Y corresponden a las direcciones del plano del área de interés y Z corresponde a la perpendicular. Se han empleado coordenadas relativas a un agente para facilitar el control de la posición, pues se ha supuesto que extraer las coordenadas directamente de los datos del área de interés implica un error de distancia entre agentes que dificulta la superposición de las imágenes. Por tanto el programa sitúa el primer agente utilizando las coordenadas absolutas del mapa de interés y establece el origen de coordenadas centrado en la imagen capturada por este agente. Es decir, el primer agente tendría coordenadas $[0, 0, d]$, donde d es la distancia al plano de interés.
 - **Ángulo:** En este proyecto solo se contempla el giro respecto a un eje paralelo a la vertical del mapa de interés.
- **Mapa de interés:** matriz que representa la distribución de la importancia en el área objetivo. Para este programa se ha supuesto que dicho objetivo corresponde a un plano, por lo que solo es necesario conocer la distancia al objetivo y las coordenadas absolutas del área, estas últimas solo son necesarias para determinar la posición del primer agente.
- **Panorámica:** imagen inicialmente en negro donde van añadiéndose las capturas. Inicialmente se dimensiona para que abarque todo el mapa de interés. Al finalizar el proceso se recortan los márgenes sobrantes.
- **Mapa de puntos característicos:** matriz binaria de igual tamaño que la imagen panorámica. Representa la localización de los puntos singulares extraídos de la panorámica. Es necesaria para determinar si el área a capturar posee un número mínimo de puntos para poder superponerse con la imagen total.
- **Relaciones píxel por unidad de longitud:** al alternar el algoritmo entre imagen y coordenadas continuamente, es necesario determinar la relación que existe entre la distancia entre píxeles de la imagen (o elementos de la matriz) y la distancia real en metros que representa para poder relacionar ambas magnitudes de forma automática. Estas relaciones tienen dos componentes, la horizontal y la vertical, y vienen expresadas en [píxel / u.l.].
 - **Del mapa de interés:** se proporciona como entrada junto con los datos del mapa. Se emplea para determinar dónde colocar la cámara para captar el punto de interés deseado.
 - **De las cámaras:** se calcula tomando la imagen capturada por el primer agente, es decir, corresponde a la relación para una cámara con el máximo acercamiento permitido y sin girar. Al ser esta primera imagen la base de la imagen panorámica, esta relación es la misma que tendría la imagen final.
 - **Relación entre ambas:** relación entre las anteriores. Se utiliza para determinar el tamaño de la imagen panorámica inicial y el tamaño de las imágenes equivalente en el mapa de interés.

3.2. Método de optimización

Al emplear el método heurístico, estudiar cada posible ubicación y orientación que la cámara puede alcanzar puede elevar el tiempo de ejecución, aunque no alcance valores tan elevados como el método conjunto. Para poder simplificar este proceso, y así dar un tiempo aceptable, se divide el problema de optimización en dos partes independientes entre sí. La primera se encargará de reducir el espacio de búsqueda y la segunda explorará la configuración más óptima en ese espacio.

Este método se aplica a cada agente, excepto el primero, que, al colocarse sin ángulo ni alejamiento, solo utiliza la primera parte. Este agente es usado como guía, siendo sus coordenadas el sistema referencia de los demás y su imagen la base de la imagen panorámica y la referencia de transformación del resto de imágenes.

3.2.1 Primera parte

Consiste en un método sencillo y de poco coste computacional para explorar todo el área de interés y seleccionar una sección muy reducida de estudio de esta. Con esto se evita comprobar todas las posibles configuraciones en cada punto del mapa de interés.

Para recorrer el mapa de interés se busca el centro de la imagen sin ángulo ni alejamiento que presente el máximo interés. Así se consigue una operación rápida de la que se extrae un único punto objetivo, que será al que apunte la cámara. El punto seleccionado es válido si el número de puntos singulares que capturaría esa imagen es superior al mínimo necesario para realizar el solapamiento. Al no tener alejamiento ni giro se supone que comprende el mínimo área de captura, por lo que, manteniendo ese punto como objetivo, siempre se captura ese área independientemente de la configuración del agente.

Este proceso tiene la desventaja de depender directamente de la fiabilidad con la que las imágenes transformadas ocupen el área en la imagen panorámica que les corresponde en el espacio real, ya que, si se salen de esta, se colocarían puntos singulares donde realmente no habría y podría imposibilitar la unión de imágenes. Una solución de compromiso es aumentar el mínimo de puntos para forzar una mayor superposición de capturas.

3.2.2 Segunda parte

Esta parte corresponde a la función '*optimo.m*' del código. Se encarga de probar las combinaciones de alejamiento (coordenada z) y ángulo para extraer la solución más óptima, dejando siempre la cámara apuntando al punto seleccionado. El rango de variación del alejamiento y el giro se ha seleccionado de tal forma que permita la correcta formación de la panorámica. Para esta optimización se emplea la siguiente función de coste:

$$\text{Coste} = -\text{Int} + (\text{Kint} * \text{IntPre})^2 * [\text{Kz} * ((Z - d)/Z\text{dmx})^2 + \text{Kang} * (|\text{Ang}|/\text{Angmx})^2]$$

- **Int: interés.** Se calcula la suma del interés de la imagen que se capturaría en esa configuración. A continuación se detalla el método de cálculo.
- **(Z-d): alejamiento respecto al del primer agente.** Al usarse esta primera imagen como base, se penaliza en el coste que difiera en alejamiento, ya que perdería definición el entorno capturado.
- **Ang: ángulo de giro de la cámara.** Al igual que ocurre con el alejamiento, se busca que sea igual al del primer agente, en este caso 0° . La transformación de giro influye más en la continuidad de la imagen, por lo que la ganancia asociada se ha diseñado para que le conceda más peso que al alejamiento.
- **IntPre: interés previo.** Se calcula el interés del área centrada en el punto seleccionado cuando la cámara no tiene giro ni alejamiento. Este valor se usa para determinar el peso que tiene la pérdida de calidad, haciendo que, si la zona es de elevado interés, se busque la mejor calidad de imagen.
- **Zdmx, Angmx.** Rango de alejamiento y máximo ángulo a evaluar. Se utilizan para reducir el valor de las variables a un rango entre 0 y 1.
- **Kz, Kang, Kint: ganancias de las variables.** Se emplean para distribuir el peso de cada componente de la función.

3.3. Algoritmo

En esta sección se detalla el funcionamiento del código empleado para llevar a cabo el proceso de distribución óptima explicado, siguiendo el esquema de la Figura 3-1. Aunque este proyecto tenga una implementación simulada, este algoritmo está diseñado para que pueda aplicarse en un caso real. Las modificaciones necesarias para realizar la prueba serán explicadas en el capítulo dedicado a la simulación.

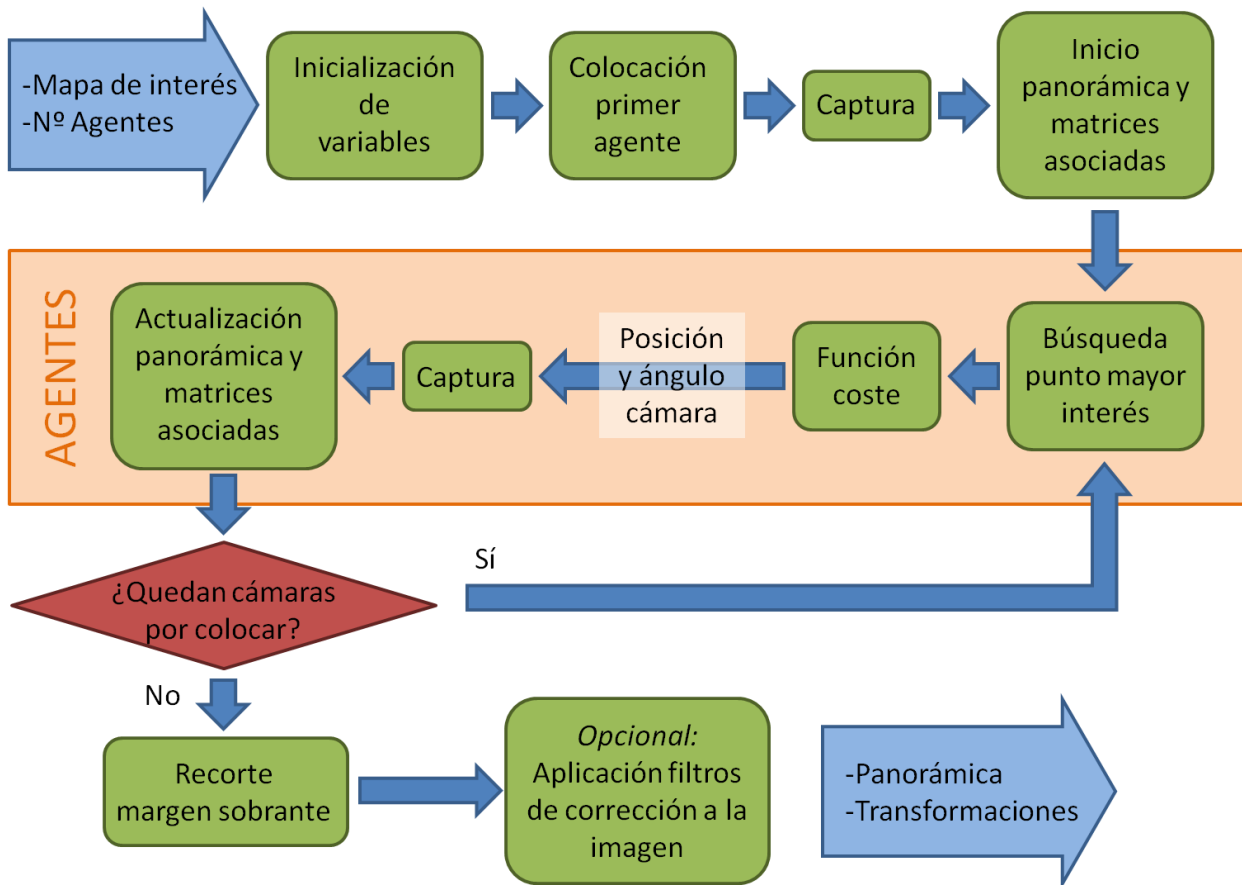


Figura 3-1. Diagrama de bloques del algoritmo de distribución.

3.3.1 Elementos externos

El primer paso es definir los elementos que son considerados entradas del sistema, y cuyos valores determinan el desarrollo del mismo:

- **Parámetros característicos de las cámaras.** Son los ángulos focales y el tamaño de imagen.
- **Mapa de interés.** Se define con:
 - Una matriz que distribuye el nivel de interés de cada punto del área de estudio.
 - La relación píxel por unidad de longitud de la matriz.
 - La distancia entre el área de interés y la zona de trabajo. Equivale al mínimo alejamiento de las cámaras.
- **Número de agentes.**

3.3.2 Inicialización

Aquí se calculan los parámetros dependientes y se declaran las variables y matrices necesarias para el desarrollo del código, determinando su valor inicial:

- **Vectores que determinan el estado de los agentes.** Recogen las coordenadas, el ángulo de giro, la imagen capturada y la relación de transformación de cada agente.

- **Relación píxel por unidad de longitud de las cámaras.** Esta relación, como se ha dicho antes, corresponde a la de una cámara con mínimo alejamiento y sin ángulo. Para hallarla basta con calcular las dimensiones en metros del área capturada, aplicando una relación trigonométrica tomando el ángulo focal y el alejamiento, y dividir el tamaño en píxeles entre estas dimensiones.
- **Relación entre la relación píxel por unidad de longitud de las cámaras y del mapa de interés.** Se dividen ambas relaciones. Se obtienen dos componentes (vertical y horizontal) que, si ambas matrices son proporcionales, son iguales.
- **Dimensiones de la imagen panorámica.** Se transforman las dimensiones del mapa de interés con la relación anterior para determinar las que tendría con una relación píxel por unidad de longitud igual a la de las cámaras. Se declaran las siguientes matrices con estas dimensiones:
 - **Imagen panorámica:** matriz imagen que abarque el mismo área que el mapa de interés. Inicialmente está a cero.
 - **Mapa de puntos característicos:** matriz binaria en la que son 1 los puntos singulares que se hayan extraído de la imagen panorámica.
 - **Base:** matriz de ceros que se utiliza como marco para que la captura transformada tenga las mismas dimensiones que la imagen panorámica, necesario para poder solaparlas.
- **Captura mínima.** Tamaño que tendría la captura a mínima distancia y sin ángulo en el mapa de interés. Se calcula para poder aplicar el primer paso de la optimización de forma inmediata.

3.3.3 Primer agente

Como se ha indicado previamente, el primer agente sirve como base de todo el proceso. Para colocarlo se realiza el siguiente proceso:

- A. Dos bucles anidados recorren cada componente del mapa de interés. Se busca la sección del mapa del tamaño de la captura mínima cuya suma de interés sea mayor, y se guarda su centro.
- B. Mediante el control de los dispositivos se coloca el dispositivo centrado en este punto, y se utiliza su posición como referencia. Para ello se da valor cero a X e Y , y a Z el alejamiento mínimo.
- C. Se calcula el equivalente de las coordenadas del punto en la imagen panorámica, se espera a que se realice la captura y se incluye centrada en dicho punto.
- D. Se extraen los puntos característicos como se ha explicado en la sección anterior. La estructura resultante contiene una matriz con las coordenadas de los puntos singulares. Para mapear los puntos se realiza un bucle que recorre cada fila de dicha matriz, y utiliza las coordenadas de esta para poner a 1 el punto correspondiente del mapa de puntos característicos.

3.3.4 Distribución agentes

En este bloque se implementan la función de optimización y las de tratamiento de imagen en conjunto para la realización de la distribución una vez colocado el primer agente. El orden a seguir es el siguiente:

- A. Se busca el punto con mayor interés de la misma forma que con el primer agente, pero añadiendo en el bucle la condición de que el equivalente de ese área en el mapa de puntos singulares contenga un mínimo de puntos. Este valor se ha extraído experimentalmente y depende de la calidad de la imagen.
- B. Se transforman las coordenadas del punto a cartesianas respecto al origen antes definido.
- C. Al no estudiarse el caso de inclinación vertical, la coordenada Y es la misma que la del punto.
- D. Para la coordenada horizontal, el alejamiento y el ángulo se aplica la función *optimo.m*. El cuerpo de la función son dos bucles anidados que aplican la función de coste a cada combinación de ángulo y alejamiento, buscando el mínimo. Para calcular el interés se ha de seleccionar primero la sección del mapa que capturaría la cámara, así como las coordenadas que esta tendría. Se parte de la representación de la cámara en el plano horizontal (Figura 3-2 donde α es el ángulo focal horizontal y φ el ángulo de giro de la cámara):

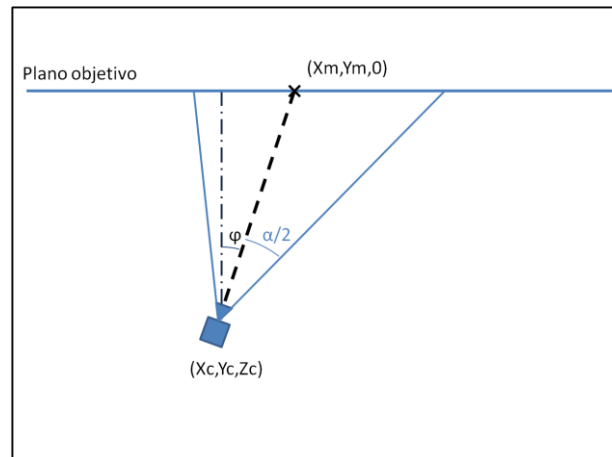


Figura 3-2. Esquema del modelo.

- a. Se calcula la coordenada X_c de la cámara para que el punto objetivo quede en el centro. Para ello se construye un triángulo rectángulo con el ángulo de giro y el alejamiento de la cámara (Figura 3-3) y se extrae la coordenada de la cámara respecto a la del punto. Debido a que no se estudia el caso de giros verticales la coordenada vertical será la misma que la del punto.

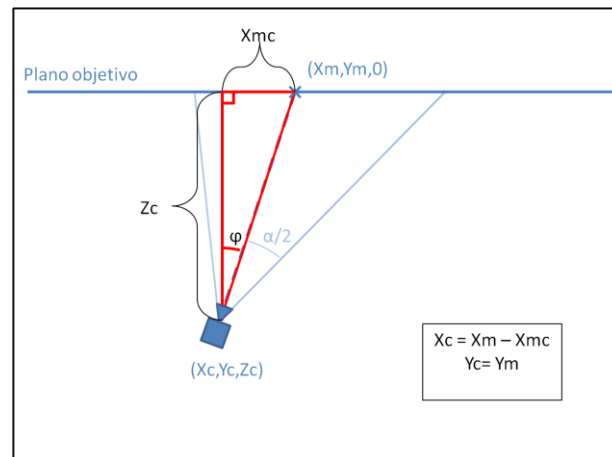


Figura 3-3. Posición horizontal de la cámara.

- b. Con este valor se calcula la coordenada X de los extremos laterales. Para ello se calcula la coordenada relativa a la cámara, construyendo los triángulos correspondientes (valores X_{dc} y X_{ic} en la Figura 3-4) y se le suma la coordenada de la cámara.

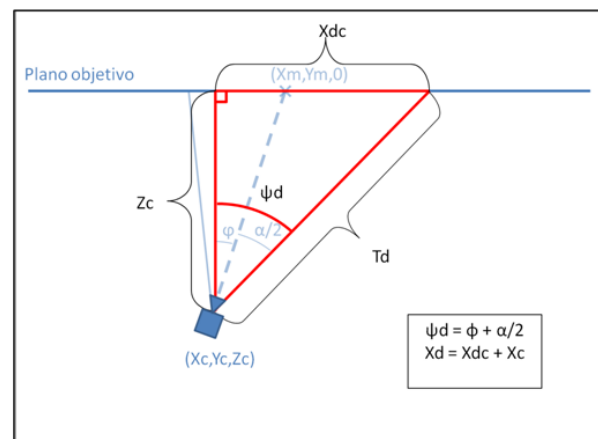
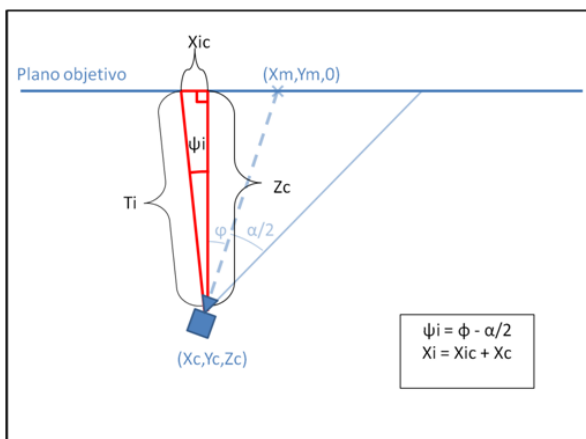


Figura 3-4. Posición horizontal de los extremos y distancia al foco.

- c. Se extrae en dichos extremos la distancia al punto (valores T_d y T_i en la Figura 3-4) y se proyecta respecto al eje de la cámara (valores T_{dp} y T_{ip} en la Figura 3-5).

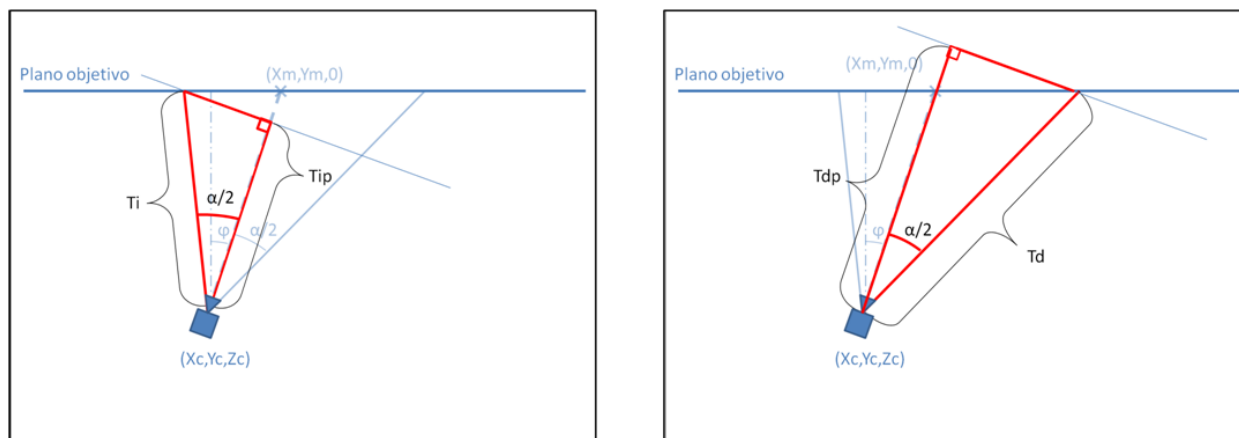


Figura 3-5. Proyección de la distancia a los extremos de la imagen en el eje de la cámara.

- d. Con esta distancia se calcula la altura de la imagen en dichos extremos con la relación que se expresa en la Figura 3-6 (donde β es el ángulo focal vertical). Estas serán la altura máxima y mínima de la imagen, y la progresión entre ellas será lineal.

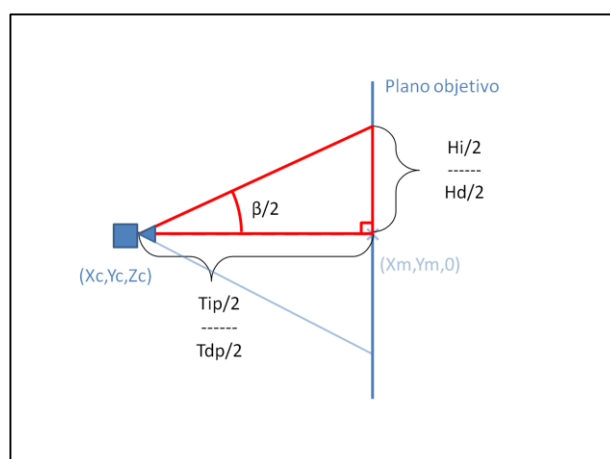


Figura 3-6. Longitud vertical.

- e. Se pasan los valores a unidades del mapa de interés mediante la relación correspondiente, y se extrae la suma del interés de los puntos comprendidos en el área correspondiente. Si algún punto queda fuera de los límites del mapa se le da un valor de 0.
- f. Si la cámara gira hacia la izquierda se sigue el mismo procedimiento y usando las mismas ecuaciones, con la única diferencia de que el ángulo de giro es negativo.
- E. Se mandan las coordenadas y giro al controlador del dispositivo y se espera hasta recibir su captura.
- F. La función de tratamiento de imagen incluye esta captura en la imagen panorámica, y actualiza las matrices del mapa de interés y los puntos singulares. Para ello:
- Se escala la máscara de la captura transformada al tamaño del mapa de interés. Se disminuye el interés de esa área.
 - Se mapean los puntos de interés como se ha realizado con el primer agente.

3.3.5 Presentación del resultado

Finalmente se ajusta la imagen para su correcta presentación. Primero se eliminan los márgenes sobrantes de la panorámica, donde no se ha colocado ninguna imagen. Es un código simple compuesto de cuatro bucles que recorren la imagen panorámica desde cada extremo hasta encontrar un píxel no negro, marcando ese punto como nuevo extremo. Se recorta la imagen en función de los nuevos límites.

En este punto también se aplica el filtro para suavizar los límites entre las diferentes capturas, como se ha explicado en la sección anterior. También pueden añadirse otros filtros para mejorar el resultado dependiendo de la utilidad que se le vaya a dar.

También hay que señalar que se han almacenado las relaciones de transformación, que pueden ser de utilidad para superponer posibles vídeos que se capturen. En este caso sería recomendable repetir el proceso cada cierto intervalo para asegurar la correcta superposición durante todo el proceso ante variaciones del entorno.

3.4. Otros métodos de optimización

Como se ha comentado previamente, la optimización puede realizarse de múltiples formas. Aquí se exponen alternativas al método usado, aunque algunas pueden no ser implementadas con las condiciones actuales, debido al alto coste computacional.

3.4.1 Optimización en un paso

Consiste en juntar las dos partes de la optimización en una única función. Este sistema comprueba todas las posibles configuraciones del agente para cada elemento del mapa, seleccionando aquel punto y configuración que minimice la función de coste.

Este proceso presenta un resultado más cercano al óptimo que el seleccionado, pero su coste computacional es notablemente más elevado, lo que hace que sea menos recomendable.

3.4.2 Búsqueda por contorno

Este método presenta una forma de reducir el número de posiciones disponibles, estudiando únicamente los puntos de una línea de contorno. Si se quiere trabajar así es necesario cambiar la forma de actualizar el mapa de interés. Esto se debe a que, para poder definir un contorno, las imágenes que ya hayan sido situadas deben bajar a cero el interés del área correspondiente.

El primer paso del proceso a realizar es formar una máscara binaria, del mismo tamaño que la imagen panorámica, que representaría las zonas donde ya se han colocado imágenes. A esta máscara se le aplica una reducción del área ocupada para dejar un margen de solapamiento de las imágenes necesario para la unión. Se calcula el contorno del área reducida.

A continuación se aplica a cada punto del contorno la función de optimización diseñada (*optimo.m*), incluyendo dentro de la misma que el área comprendida tenga el mínimo de puntos singulares para la correcta asociación. Se selecciona el punto y la configuración de la cámara que minimice el valor de la función de coste. Si en el contorno estudiado no existe ningún punto con el mínimo necesario de puntos característicos, hay que ampliar el margen de reducción de la máscara y repetir el proceso hasta que se tenga un punto con el número ideal para unir las imágenes.

Con este método se mejora considerablemente el coste computacional frente al anterior, pero aun así se tiene que ejecutar la función *optimo.m* en reiteradas ocasiones, más aún si se ha de reducir nuevamente el contorno. Esto hace que siga siendo más lento que la distribución que se ha seleccionado. Además, por su tratamiento especial con el mapa de interés, se pierde la capacidad de tener varios agentes capturando una zona de interés elevado, por lo que no se seguiría capturando si el agente falla.

4 SIMULACIÓN

Como se trata de un proyecto previo a la puesta en práctica, y debido a la ausencia de material para ello, es necesario desarrollar un método para simular el funcionamiento del código en una situación próxima a la real. Para ello se seleccionan los parámetros externos al código que dependan de la interacción con el entorno, y se estudia su comportamiento frente a los cambios internos. Una vez se ha realizado la comprobación, se determinan los elementos que hay que definir para representar el espacio real y la función que los relaciona con el código de forma lógica.

En este caso solo se necesita simular la imagen que capturaría una cámara dada su posición y giro. Para ello se necesita disponer de una imagen que represente el área de interés que se quiere observar. Una vez se tiene el plano objetivo, se establece la relación de los puntos con las coordenadas de las cámaras para delimitar el área de captura. Hay que aclarar que las imágenes capturadas son siempre rectangulares pero el área que representan en el plano objetivo puede no serlo, esto se produce cuando la cámara está girada, por lo que se debe realizar la asociación de píxeles considerando dicha deformación.

El primer paso para diseñar la función que devuelva la captura es escoger el modelo de cámara con el que se va a trabajar. Por similitud con el usado para el proyecto, se va a realizar la simulación con el modelo *pinhole*. Gracias a esto podemos determinar el punto que captaría cada píxel de la imagen usando las coordenadas puntuales de la cámara y trazando haces con la dirección correspondiente. Estos haces tendrán ángulos de separación entre ellos iguales, pero la separación entre los píxeles que les corresponda solo será constante si el eje de la cámara es perpendicular al plano.

El otro elemento que se tiene que incluir como entrada es la imagen global u objetivo. Para evitar que la captura quede fuera de la imagen global, se ha asumido que el área que representa equivale al del mapa de interés, así se tendrá siempre imagen disponible. Asumiendo esto no hay que proporcionar la posición donde se encuentra, pues será la misma que se ha usado para el mapa. Lo que sí hay que indicar es la relación píxel por unidad de longitud, para poder conocer las coordenadas de cada punto. Para la correcta equivalencia con el mapa de interés se escala el tamaño de este para que sea proporcional a la imagen del panorama. Para ello se introduce una matriz que represente dicho mapa, sin importar sus dimensiones, y se iguala su tamaño a uno proporcional al paisaje con la escala reducida. Esta reducción determina el tiempo de ejecución que se invierte en recorrer el mapa.

Por último hay que destacar que en esta simulación se ha supuesto un funcionamiento ideal, sin atender a los posibles fallos que puedan ocurrir en una implementación en un entorno real. Se ha realizado así porque se ha considerado que en unas condiciones de entorno óptimas estos fallos no ocurrirían. Con ello se limita la utilidad del proyecto a escenarios que presenten dichas cualidades, pero se mantiene su validez para comprobar su funcionamiento principal. Como apunte se ha incluido al final de esta sección una lista de algunos errores que pueden ocurrir en dicho entorno y posibles medidas que los corrijan. Estas medidas se consideran parte de los objetivos si se realiza una ampliación de este trabajo.

4.1. Función de captura

Esta es la función encargada de, dadas las coordenadas de una cámara y su ángulo, devolver la imagen que captaría del panorama. Para lograr esto se requiere que se introduzcan los parámetros externos que, como se ha visto antes, simularán la situación de la cámara y el paisaje. La salida de la función será una imagen que representa la captura que obtendría el dispositivo que se está simulando y cuyas dimensiones vienen determinadas por características de la cámara.

4.1.1 Entradas

- **Imagen original.** Es aquella que representa el entorno de estudio. Se presupone que todo el panorama entra dentro del área de captura alcanzable por las cámaras. También se considera que el área que abarca equivale al mismo que abarca el mapa de interés proporcionado, aunque con distinta relación de píxel por distancia.
- **Origen de coordenadas.** Posición del píxel de la imagen original donde está situado dicho origen.
- **Relación píxel por unidad de longitud en la imagen original.** Es necesaria para conocer el equivalente en coordenadas matriciales de las posiciones expresadas en la unidad de medida que se esté trabajando.
- **Características de la cámara.** Incluye los dos ángulos focales y la resolución de la misma (dimensiones de la imagen en píxel). Variando la resolución se capta el mismo área pero con un tamaño distinto de imagen. La máxima resolución alcanzable corresponde con la resolución de la imagen original.
- **Coordenadas de la cámara.** La posición respecto al origen de coordenadas. Las unidades deben ser las mismas con las que se trabaje en todo el proceso.
- **Giro.** El ángulo que tenga la cámara respecto al eje paralelo a la vertical del panorama a estudiar.

4.1.2 Funcionamiento

La función consiste en dos bucles anidados que recorren cada componente de una matriz del tamaño definido por la resolución de la cámara. Para cada componente se calcula el punto de la imagen de referencia que se capturaría, hallando sus coordenadas reales y determinando el equivalente en las coordenadas de la imagen.

El primer bucle recorre la horizontal, cuya separación entre píxeles capturados no es igual. En vez de calcular la separación se calcula la coordenada horizontal que tiene la línea vertical a estudiar. Para ello primero se divide el ángulo focal horizontal en M partes iguales. Por cada uno de estos ángulos se realiza una relación trigonométrica se extrae la posición relativa respecto al origen de coordenadas como se hizo con los extremos en la Figura 3-4. Se calcula el valor en coordenadas matriciales y se le suma la coordenada del origen proporcionada en las entradas.

Antes de entrar al bucle de las verticales se calcula la distancia al píxel horizontal y se extrae su proyección respecto al ángulo de la cámara. Con esta distancia y el ángulo focal vertical se calcula la longitud del tramo que capturaría esa vertical. Este es el mismo procedimiento empleado en la función de optimización y que viene representado en la Figura 3-5 y Figura 3-6.

En el bucle anidado se recorre la vertical de la longitud calculada centrada en el punto horizontal de estudio. Al no tener ángulo vertical, la separación entre todos píxeles capturados de la misma línea vertical se mantiene constante, siendo esta la longitud del tramo entre la dimensión vertical de la imagen. Finalmente se calcula el valor en coordenadas matriciales.

Si al calcular las coordenadas se sale de los límites de la imagen de referencia se colocará un píxel negro, aunque en este caso se ha cogido una imagen lo suficientemente grande para que no ocurra.

4.1.3 Ejemplos de funcionamiento

Se han realizado varios experimentos para comprobar el funcionamiento. Para las pruebas se ha usado la Figura 4-1. Para esta imagen se ha determinado una relación píxel por unidad de distancia equivalente a la que tendría una cámara con tamaño de imagen de 600x600 píxeles, sin giro y con alejamiento mínimo. Por tanto, esta es la calidad máxima que se le puede dar a las capturas. Los resultados que se muestran a continuación corresponden a dichas pruebas, donde se estudia la respuesta ante cambios en posición, ángulo y calidad de imagen. En la Tabla 4-1 se indica las coordenadas vertical y horizontal respecto al origen y las coordenadas matriciales correspondientes. Para ambas viene dada primero la horizontal y luego la vertical. Los ángulos focales para estas pruebas miden ambos 30°.

Tabla 4-1. Posición de las cámaras en las pruebas.

Prueba	Coordenadas relativas [u.l.]	Coordenadas matriciales [píxel]
Origen coordenadas	(0,0)	(1500,2000)
Alejamiento	(200,-600)	(1611,1664)
Giro	(2200,-1800)	(2732,992)
Calidad	(1200,-1500)	(2172,1160)

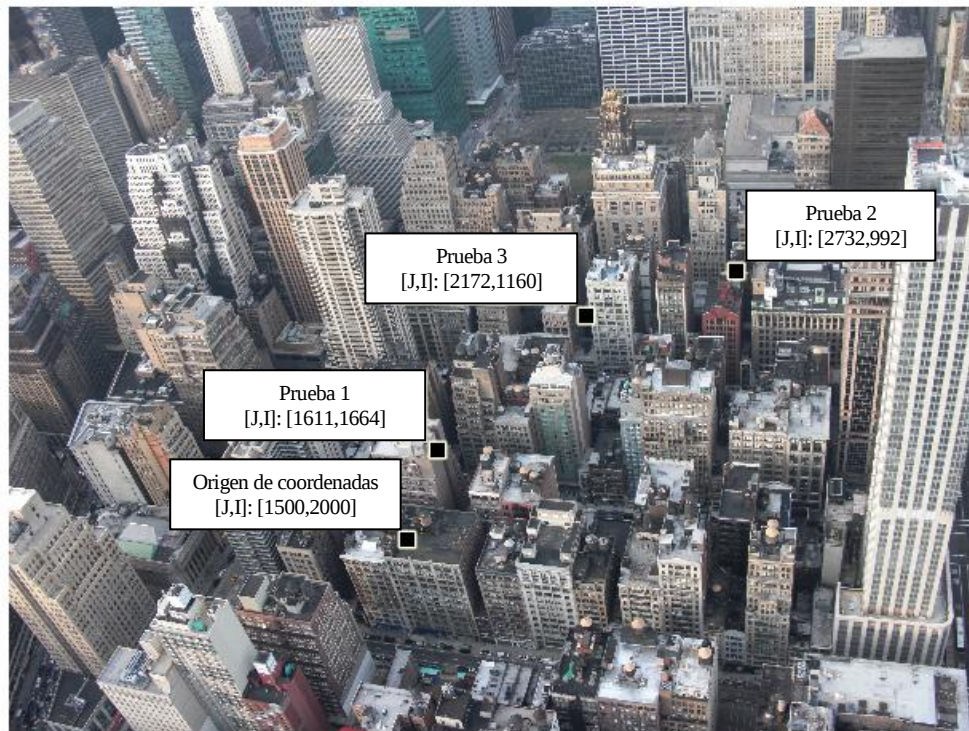
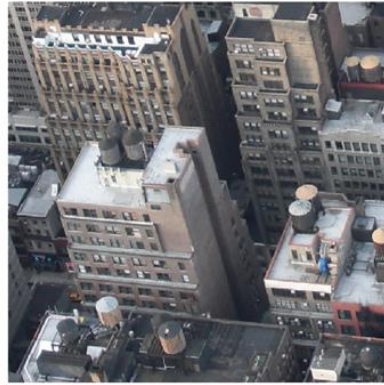


Figura 4-1. Imagen usada como referencia y puntos donde se centran las pruebas

- **Alejamiento** (Figura 4-2): cuatro pruebas variando el alejamiento y manteniendo las coordenadas X e Y constantes. Se comprueba que, con el mismo tamaño de imagen, aumenta el área capturada al aumentar el alejamiento. Para ellas se ha considerado sin giro y tamaño de imagen de 400x400.
- **Giro** (Figura 4-3): cuatro pruebas variando el ángulo. Una vez obtenida la captura, se aplica la función de tratamiento de imagen, tomando como base la imagen total, que correspondería a girar la imagen para que tenga el mismo ángulo que la base. A continuación se muestran las capturas y su transformación tomando como base la imagen original. Para ellas se ha considerado alejamiento de 2000 u.l. y tamaño de imagen de 400x400.
- **Calidad** (Figura 4-4): cuatro pruebas variando la calidad y manteniendo constantes las posición y ángulo de la cámara, para observar que se capturaría el mismo área pero con distinto tamaño de imagen. Para ellas se ha considerado sin giro y con alejamiento de 2000 u.l.. En la Tabla 4-2 se recogen los tiempos de ejecución para cada calidad.



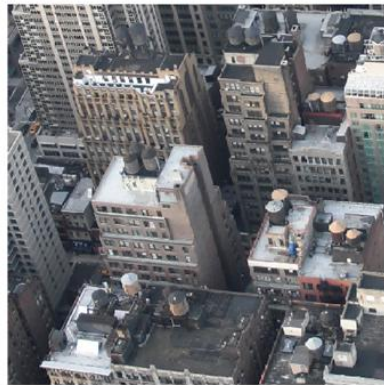
(a) 2000 u.l.



(b) 2250 u.l.



(c) 2500 u.l.



(d) 3000 u.l.

Figura 4-2. Resultados para la prueba variando el alejamiento

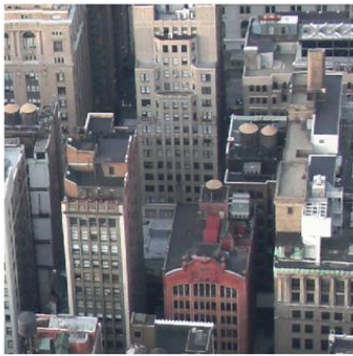
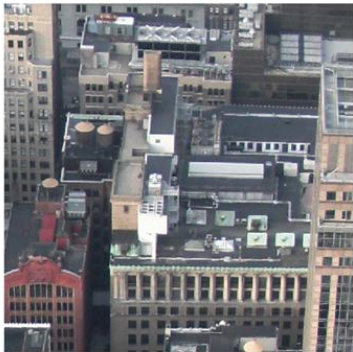
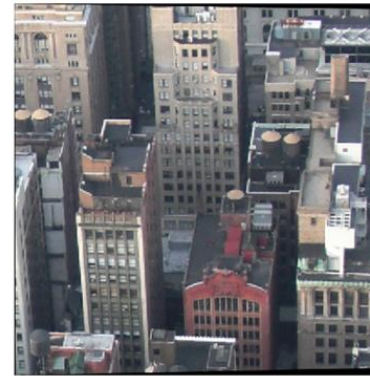
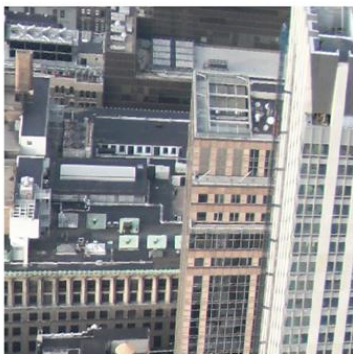
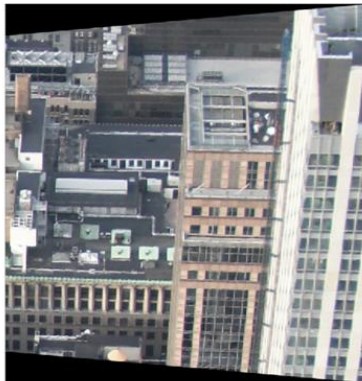
(a) -15° (b) -5° (c) 9° (d) 20° 

Figura 4-3. Resultados de la prueba variando el ángulo y la imagen transformada tomando como referencia la original.



(a) 600x600 píxeles



(b) 500x500 píxeles



(c) 300x300 píxeles



(d) 100x100 píxeles

Figura 4-4. Resultados para la prueba variando la calidad.

Tabla 4-2. Tiempos de ejecución variando la calidad.

Tamaño (píxel)	Tiempo de ejecución [s]
600x600	1.857
500x500	1.276
300x300	0.468
200x200	0.207

Con estos experimentos se puede observar que la función permite alcanzar los objetivos requeridos de la simulación, concediendo además la posibilidad de variar la calidad de las imágenes para probar la relación entre el tamaño de las capturas y la velocidad del programa.

Se ha realizado una prueba extra variando el tamaño y los ángulos focales para comprobar que se pueden extraer capturas que no sean cuadradas.

Tabla 4-3. Entradas para la prueba.

Posición	Ángulo	Ángulos focales	Tamaño
(-800,-700,2100)	7	(45,30)	(460,320)



Figura 4-5. Resultado de la prueba.

4.2. Consideraciones en un escenario real

Este método de simulación no cubre con toda fidelidad un caso real. Ello se debe a que, al extraer todas las capturas de la misma imagen, no se consideran las variaciones que se pueden producir en el entorno al observarse desde distintas posiciones. Si el entorno cambia de forma brusca entre dos capturas pueden dificultar el proceso de búsqueda de puntos similares entre ambas. A continuación se exponen algunos casos que pueden provocar variaciones.

4.2.1 Iluminación

Si poseen distinto ángulo puede haber diferencia en la relación de intensidad de las imágenes. Aun así no se contemplan grandes variaciones de ángulo, por lo que, con las cámaras a suficiente distancia, no supondría un gran impedimento. De todas formas sería recomendable estudiar el caso con agentes reales, y diseñar en caso necesario una función que iguale la iluminación entre las imágenes.

4.2.2 Objetos a distancia intermedia

Si existen objetos que se encuentre más cerca que el área de estudio, se dará el caso de que dicho objeto cambiará su posición relativa con el fondo entre imágenes, o incluso no ser captado por otras. Esto supone un problema para la asociación entre fotos principalmente por dos motivos:

- El objeto aporta puntos de interés que no deberían tenerse en consideración. Incluso si el objeto se ha captado bien en dos imágenes, la diferencia entre perspectivas supondría que, al unirlos a raíz de estos puntos, el área de interés se viese distorsionada o no encajase.
- El área que el objeto impide capturar contiene puntos de interés que pueden ser necesarios para hallar la relación de transformación entre las imágenes.

En definitiva, tener objetos en un plano intermedio entre las cámaras y el área objetivo puede provocar transformaciones erróneas entre las imágenes o incluso impedir que se obtenga alguna, imposibilitando la formación de una imagen panorámica coherente. La forma de evitar esto sería diseñar una función que permita identificar si hay objetos e intentar evitarlos, dando más peso a las imágenes que no los hayan capturado. Esta función no es inmediata y requiere un conocimiento del tratamiento de imágenes 3D que no entra dentro del alcance de este proyecto.

4.2.3 Distintos niveles de profundidad

El área de estudio puede no ser plana, presentando distintos niveles de profundidad. Esto, al igual que ocurre con los objetos intermedios, puede provocar cambios en las posiciones relativas de los distintos niveles, imposibilitando así la capacidad de establecer una transformada para solapar las imágenes.

La mejor forma de evitar variaciones bruscas es trabajar con áreas de interés que sean planas y perpendiculares al eje de las cámaras cuando no tienen presentan giro. Si existen distintos niveles de profundidad habría que limitar el ángulo máximo de giro y la distancia entre cámaras, para que las diferencias entre las perspectivas de cada cámara no influyan en la correspondencia entre las capturas.

4.2.4 Ruido y enfoque

Estas dos distorsiones de la imagen se incluyen en esta lista pese a que no vienen ligadas estrictamente a la diferencia de perspectivas entre las cámaras. Si bien el enfoque puede verse influido, el ruido se puede considerar independiente de la posición, ya que, al ser aleatorio, puede provocar que dos imágenes se vean distintas aunque sean tomadas desde la misma perspectiva.

Para el enfoque puede darse el caso de que las cámaras no enfoquen bien el objetivo. Para compensarlo existen filtros que pueden emplearse para enfocar la imagen.

En relación al ruido el método de extracción de características usado tiene la ventaja de ser capaz de ignorarlo, lo que hace que no sea necesario un tratamiento. Aun así, se puede aplicar un filtro a la imagen para mejorar la calidad a la hora de mostrarla. Si al realizar el proceso de solapamiento da error por culpa del exceso ruido y los filtros no lo mejoran, una posible solución es repetir la captura desde la misma posición y ver si se reduce el nivel de ruido. Si tras varios intentos se mantiene el ruido se marcaría la cámara como defectuosa y se retiraría el agente.

5 ANÁLISIS DE RESULTADOS

En esta sección se incluyen los resultados que se han obtenido al realizar varias pruebas con distinto número de agentes y calidad de imagen. Antes de realizar las pruebas definitivas se han realizado pruebas experimentales para determinar los parámetros necesarios para obtener los resultados deseados, tanto de la función de optimización como de las funciones del tratamiento de imagen. Para ello, primero hay que definir cómo se van a interpretar los niveles de interés y cuál va a ser su distribución.

El diseño del mapa de interés con el que va a trabajar la función se ha llevado a cabo con una matriz con capas de valores de 1 a 18, como se muestra en la Figura 5-2. Para que el mapa se corresponda con la imagen de la simulación, se ha definido una matriz de dimensiones fijas con la distribución deseada y se ha escalado a un tamaño proporcional al de la imagen, en este caso cinco veces menor. Hay que asegurarse de que al escalar la matriz las dimensiones proporcionadas sean valores enteros. La relación píxel por distancia es, por tanto, cinco veces menor que la de la imagen base. Cabe destacar que en una implementación real tanto el mapa como la relación vendrían dados y no habría que aplicarles ninguna modificación.

En la Tabla 5-1 se indican los valores que se otorgan a los distintos parámetros de la función de optimización. Estos han sido sintonizados de forma que, para las zonas de interés medio superior 15 se busque la máxima calidad, sin giro ni alejamiento; para un interés inferior a 5 la calidad sea prescindible y, si se da un interés intermedio, vaya cambiando gradualmente. También indica el rango de valores del ángulo del giro y el alejamiento máximo que se consideran para buscar el mínimo en la función de coste para un correcto funcionamiento del proceso de solapamiento de imágenes.

Tabla 5-1. Parámetros de optimización.

Parámetro	Valor
Ganancia alejamiento	1.3
Ganancia giro	1.56
Ganancia interés	0.001
Rango ángulo de giro	$[-18,18]^{\circ}$
Rango alejamiento	$[2000,2200]$ u.l.

A la disminución de interés que experimenta el mapa de interés al añadir una captura se le ha dado un valor de 5 por cada elemento del área ocupada. Con esto se consigue que las zonas de grado de interés superior a 10 puedan ser capturadas por varios agentes, permitiendo continuar capturándolas si uno de las cámaras falla.

La imagen que se ha usado como base para estas pruebas es la misma usada para la simulación, con la misma relación píxel por unidad de medida que se empleaba en dicha sección. Esto implica que, como ocurría en esas pruebas, el tamaño máximo que pueden tener las capturas es de 600x600 píxeles.

5.1. Primera prueba

Para esta prueba se han considerado cinco dispositivos, con una calidad intermedia.

5.1.1 Entradas

Tabla 5-2. Parámetros entrada de la prueba 1.

Parámetro	Valor
Nº píxeles horizontal	400 píxeles
Nº píxeles vertical	400 píxeles
Ángulo focal horizontal	30°
Ángulo focal vertical	30°
Distancia al mapa de interés	2000 u.l.



Figura 5-1. Imagen tomada como panorama de estudio.

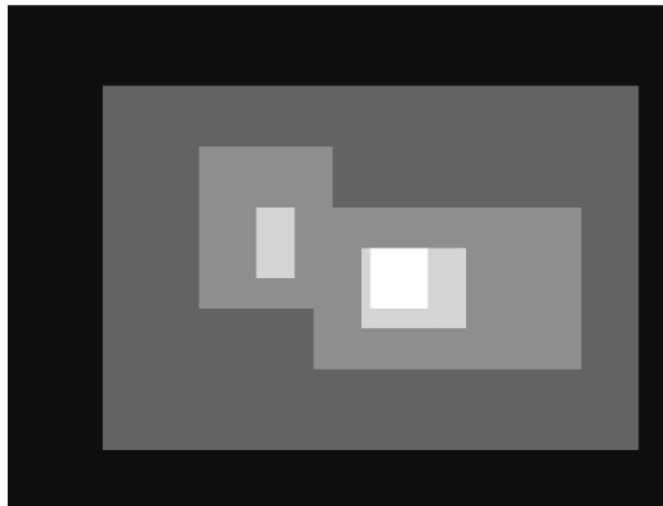


Figura 5-2. Mapa de interés empleado en la primera y segunda prueba.

Tabla 5-3. Relaciones de los elementos

Elemento	Relación [píxel/u.l.]
Imagen simulación	0.56
Mapa de interés	0.12
Capturas	0.37

5.1.2 Resultados

Para el análisis se incluyen las capturas sin transformar que toma cada dispositivo y la actualización tanto de la panorámica y del mapa de interés por cada imagen solapada.

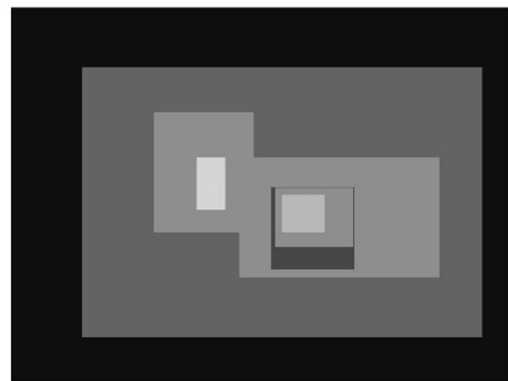
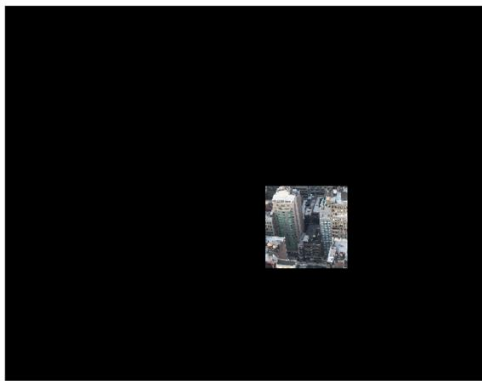


Figura 5-3. Captura del primer agente y actualización de la panorámica y el mapa de interés.

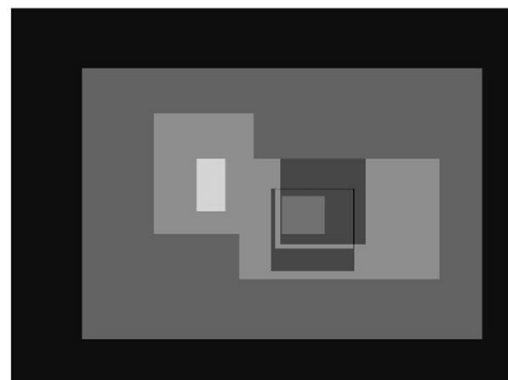
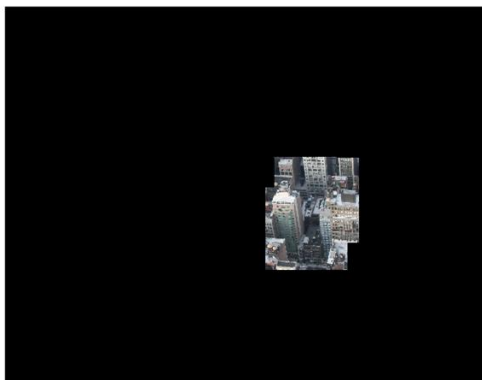


Figura 5-4. Captura del segundo agente y actualización de la panorámica y el mapa de interés.

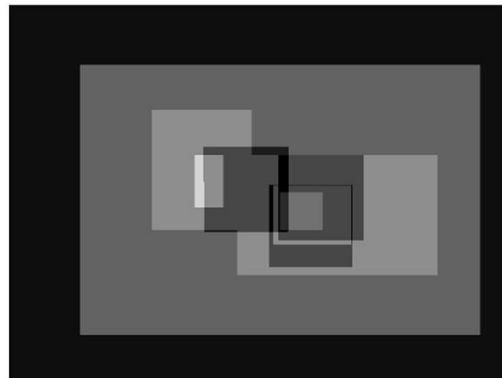


Figura 5-5. Captura del tercer agente y actualización de la panorámica y el mapa de interés.

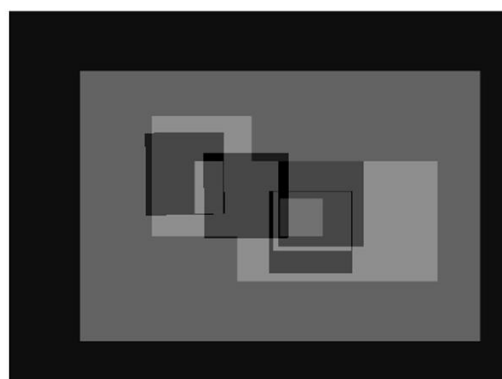
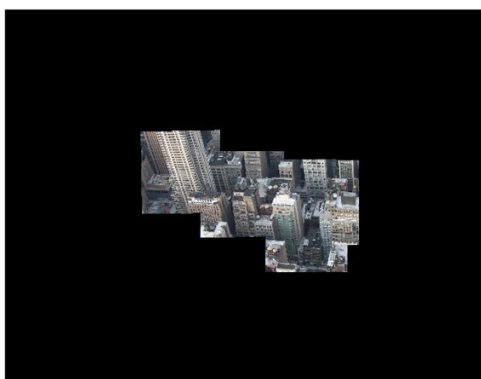
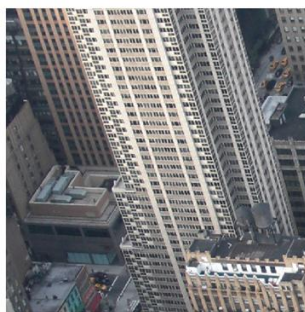


Figura 5-6. Captura del cuarto agente y actualización de la panorámica y el mapa de interés.

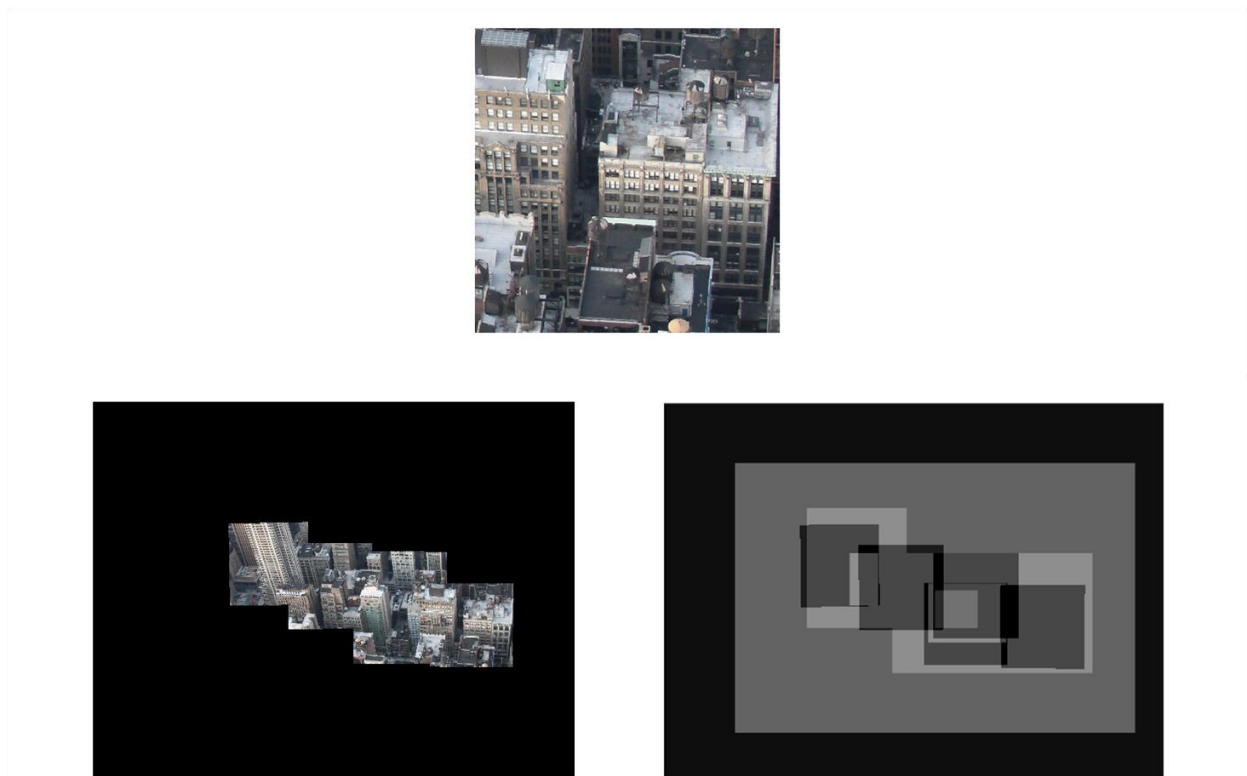


Figura 5-7. Captura del quinto agente y actualización de la panorámica y el mapa de interés.

Y finalmente se expone el resultado final, guardando en la memoria del programa la posición de las cámaras y la transformada que se le ha aplicado a las capturas. Las coordenadas matriciales del punto tomado como origen son (323,441) y el tiempo de ejecución es de 52.599s .

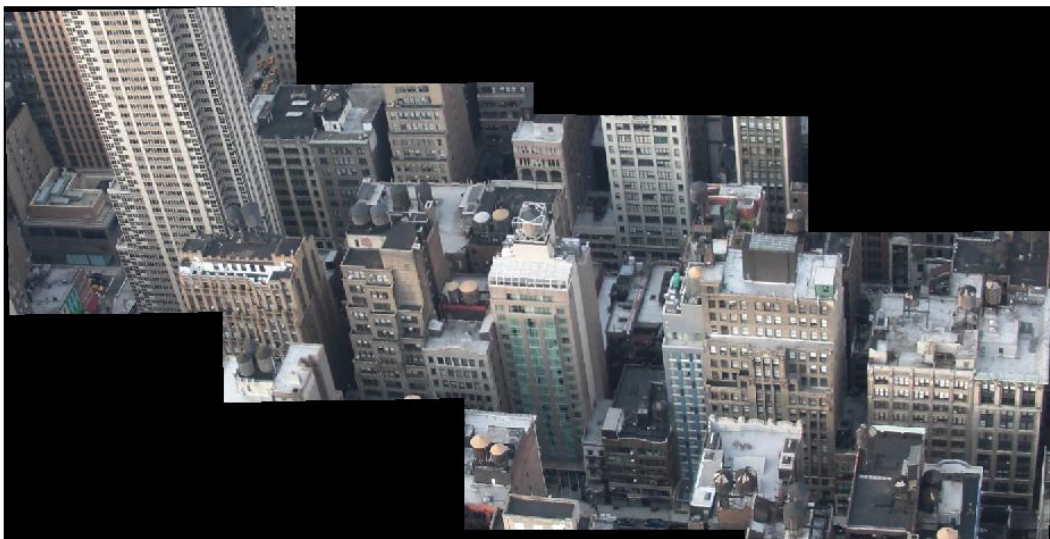


Figura 5-8. Resultado de la primera prueba.

Tabla 5-4. Configuración de los dispositivos de la primera prueba.

Dispositivo	Coordenadas (X,Y,Z) [u.l.]	Giro [°]
1°	(0, 0, 2000)	0
2°	(52.8, -375.1, 2070)	2
3°	(-618.8, -482.3, 2100)	-6
4°	(-1606.9, -785.9, 2070)	-2
5°	(779.6, 35.7, 2100)	6

5.2. Segunda prueba

Para esta prueba se han mantenido las mismas entradas que la anterior y se ha aumentado el número de agentes a 9.

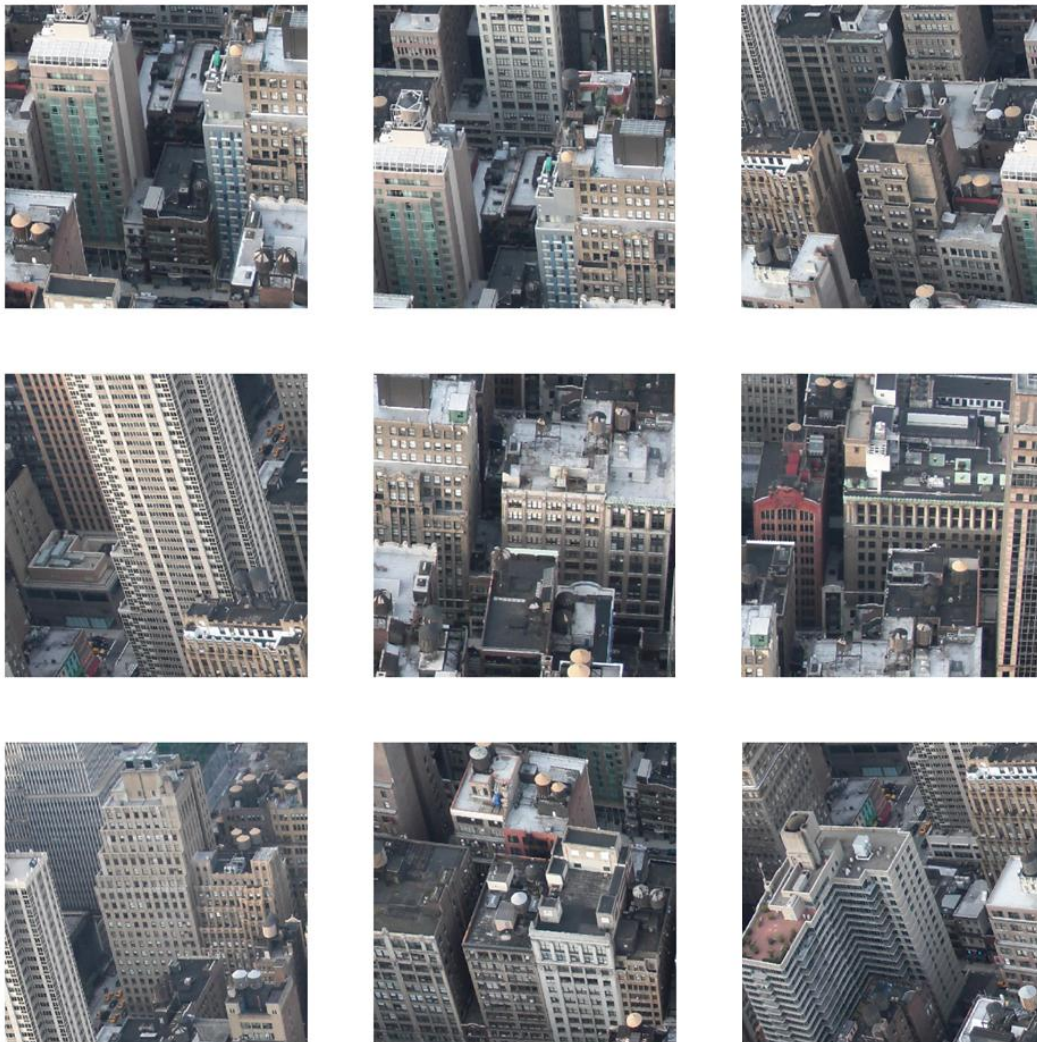


Figura 5-9. Capturas de los nueve agentes de la segunda prueba.



Figura 5-10. Resultado de la segunda prueba.

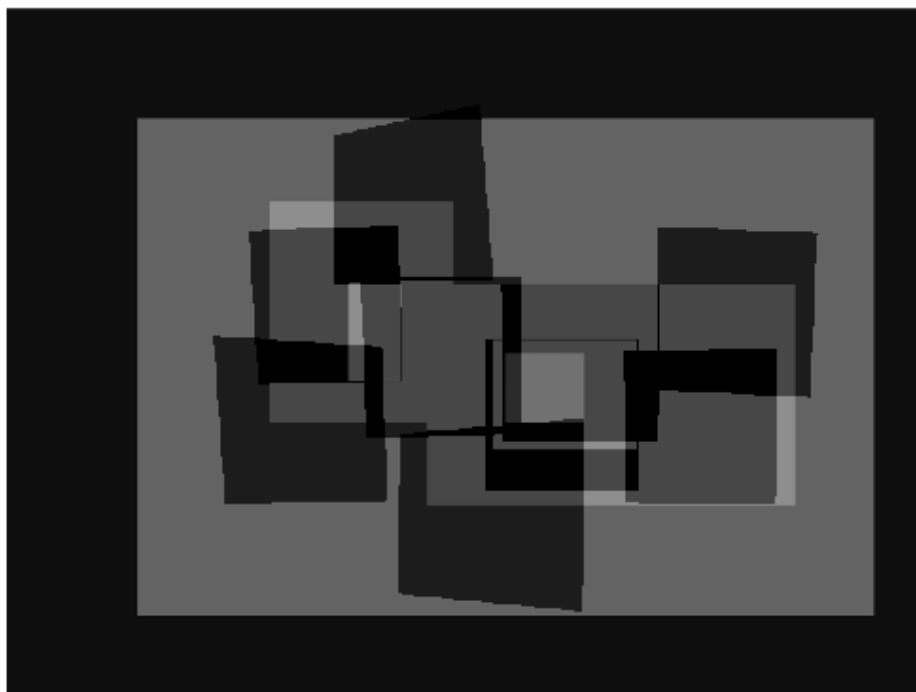


Figura 5-11. Mapa de interés resultante tras la segunda prueba.

Tabla 5-5. Configuración de los dispositivos de la segunda prueba.

Dispositivo	Coordenadas (X,Y,Z) [u.l.]	Giro [°]
1°	(0, 0, 2000)	0
2°	(52.8, -375.1, 2070)	2
3°	(-618.8, -410.8, 2100)	-6
4°	(-1606.9, -785.9, 2070)	-2
5°	(761.8, 89.3, 2100)	6
6°	(858.7, -732.4, 2120)	10
7°	(-1089.7, -1518.4, 2120)	0
8°	(-1265.3, 696.7, 2190)	18
9°	(-1164.1, 71.4, 2190)	-18

El origen de coordenadas es el mismo que el anterior y el tiempo de ejecución es de 125.573s.

Debido a la distribución agente por agente se observa que, hasta el quinto dispositivo, el comportamiento es similar al de la prueba anterior, con las diferencias que produce la aleatoriedad de la estimación de la transformada. A partir de ese punto se añaden los cuatro agentes restantes ampliando el rango de la panorámica. Se puede apreciar el error que se acumula al basar transformaciones tomando como base imágenes ya transformadas.

5.3. Tercera prueba

Para esta prueba se han seleccionado 6 cámaras y se ha modificado el mapa de interés. Se ha repetido tres veces variando la calidad de la imagen para comprobar el límite de funcionamiento y la diferencia entre el tiempo de ejecución.

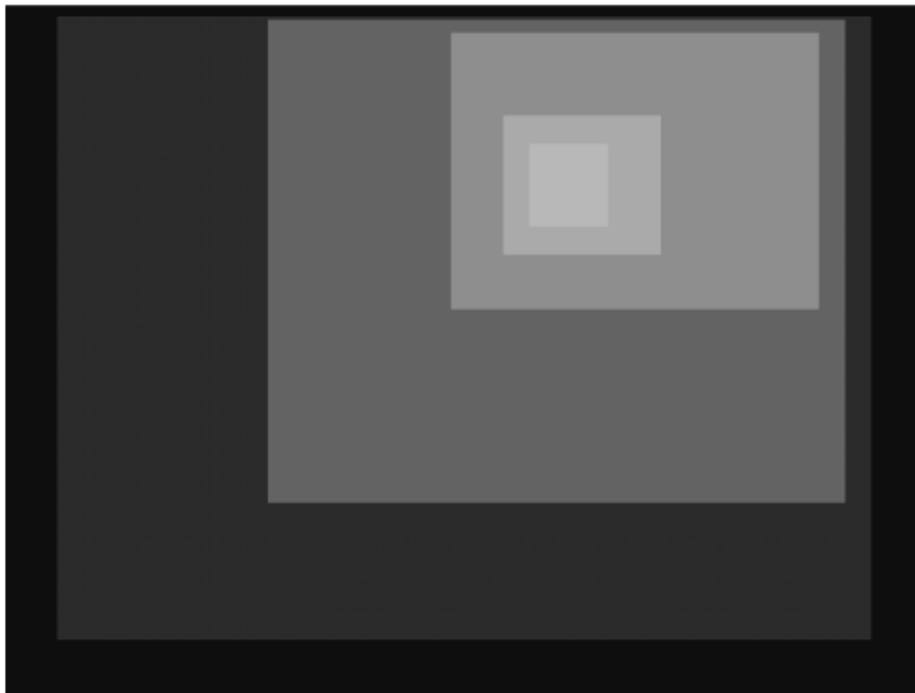


Figura 5-12. Mapa de interés para la tercera prueba.

5.3.1 600x600

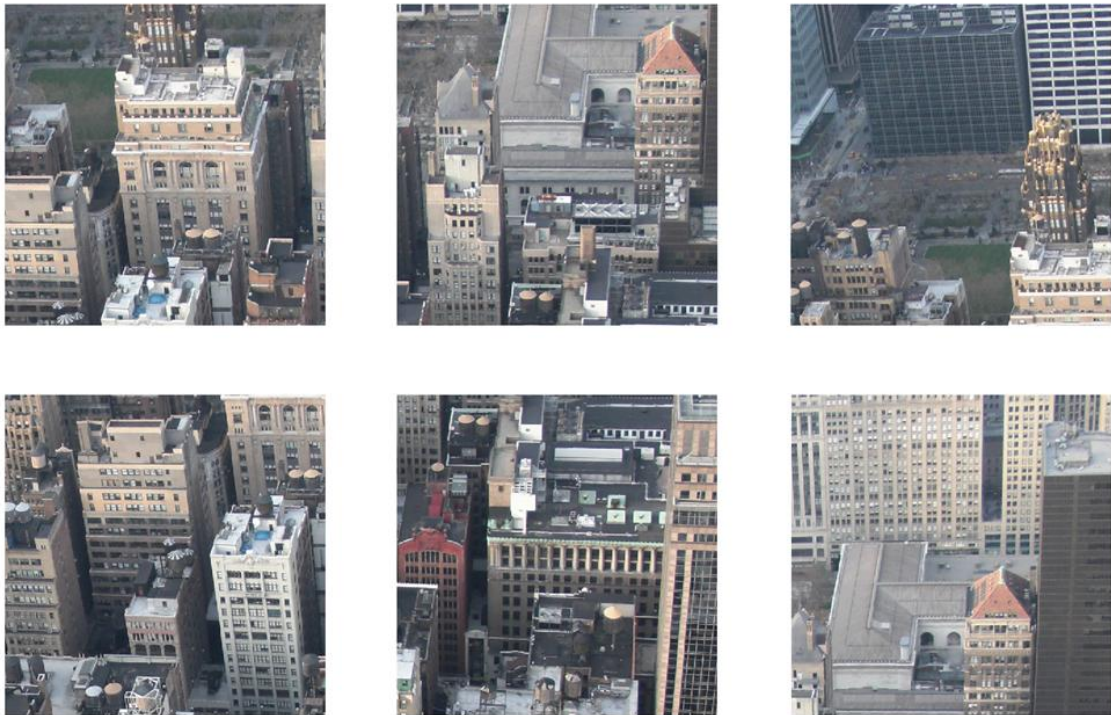


Figura 5-13. Capturas para la tercera prueba con un tamaño de 600x600.

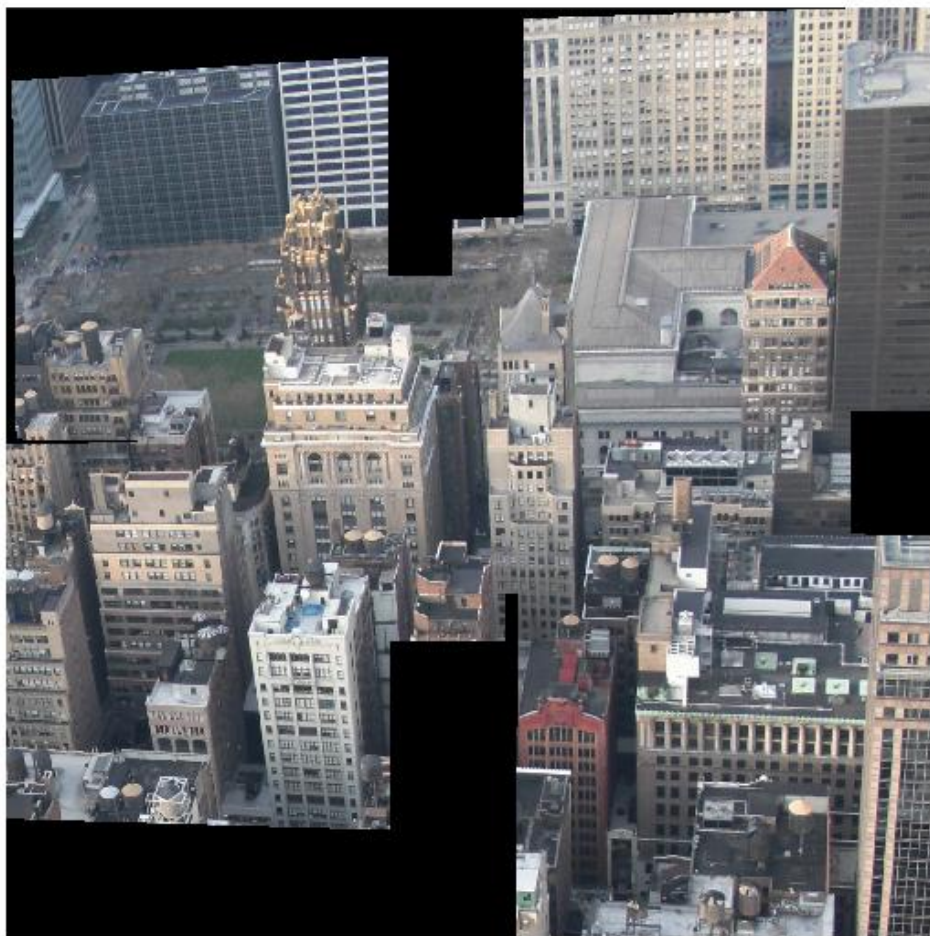


Figura 5-14. Resultado para la tercera prueba con un tamaño de 600x600.

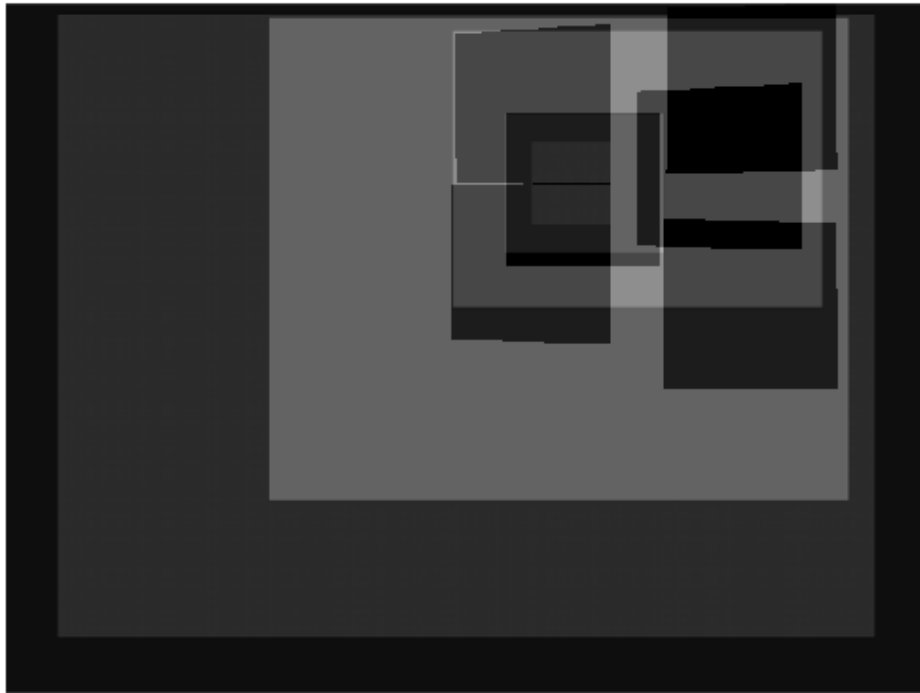


Figura 5-15. Mapa de interés resultante para la tercera prueba con tamaño de 600x600.

5.3.2 300x300

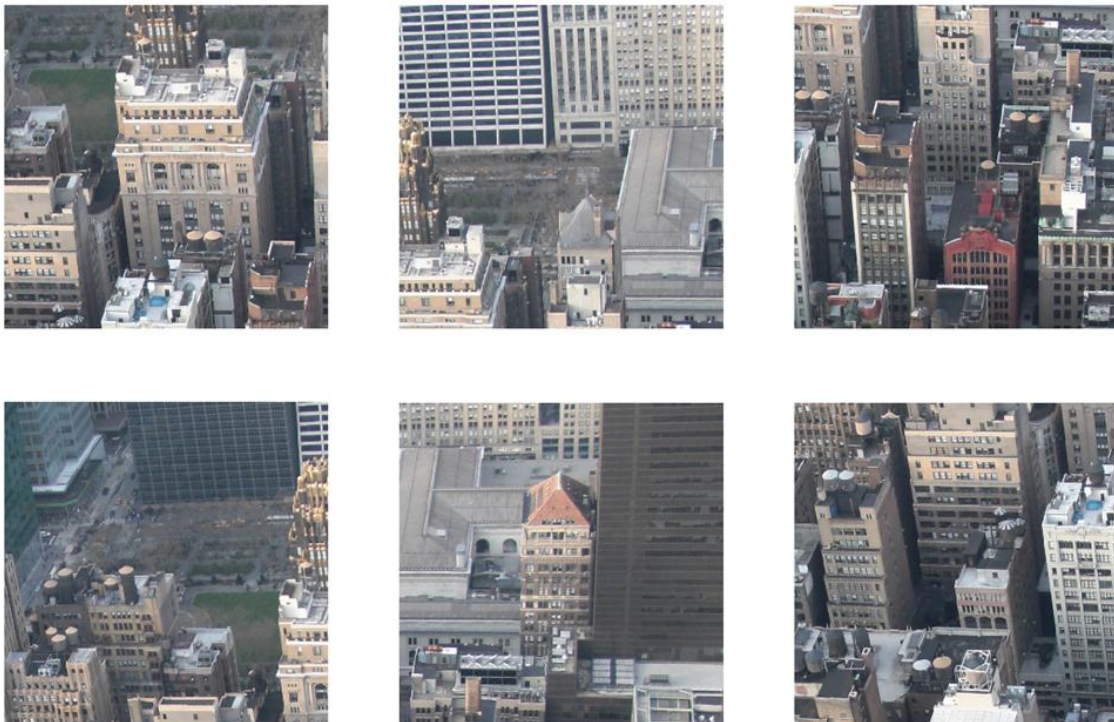


Figura 5-16. Capturas para la tercera prueba con un tamaño de 300x300.

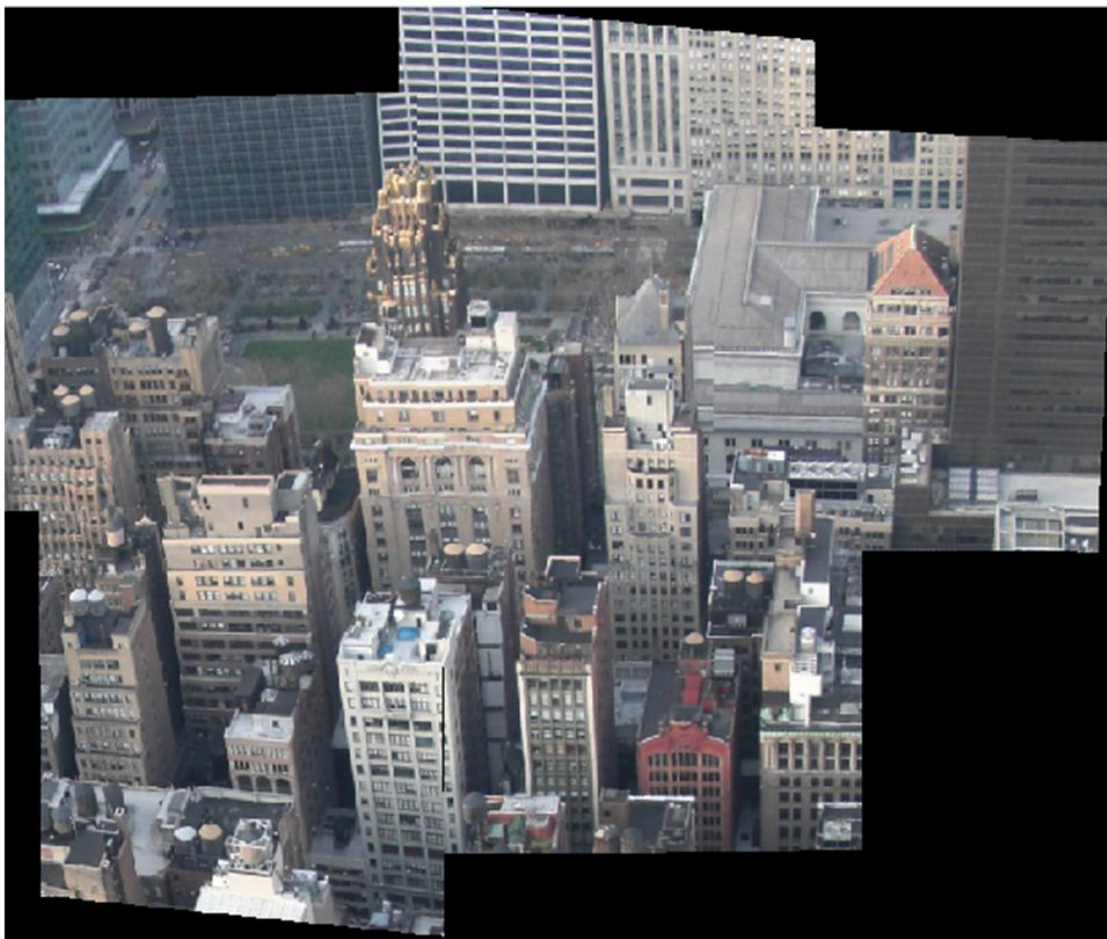


Figura 5-17. Resultado para la tercera prueba con un tamaño de 300x300.



Figura 5-18. Mapa de interés resultante para la tercera prueba con tamaño de 300x300.

5.3.3 150x150

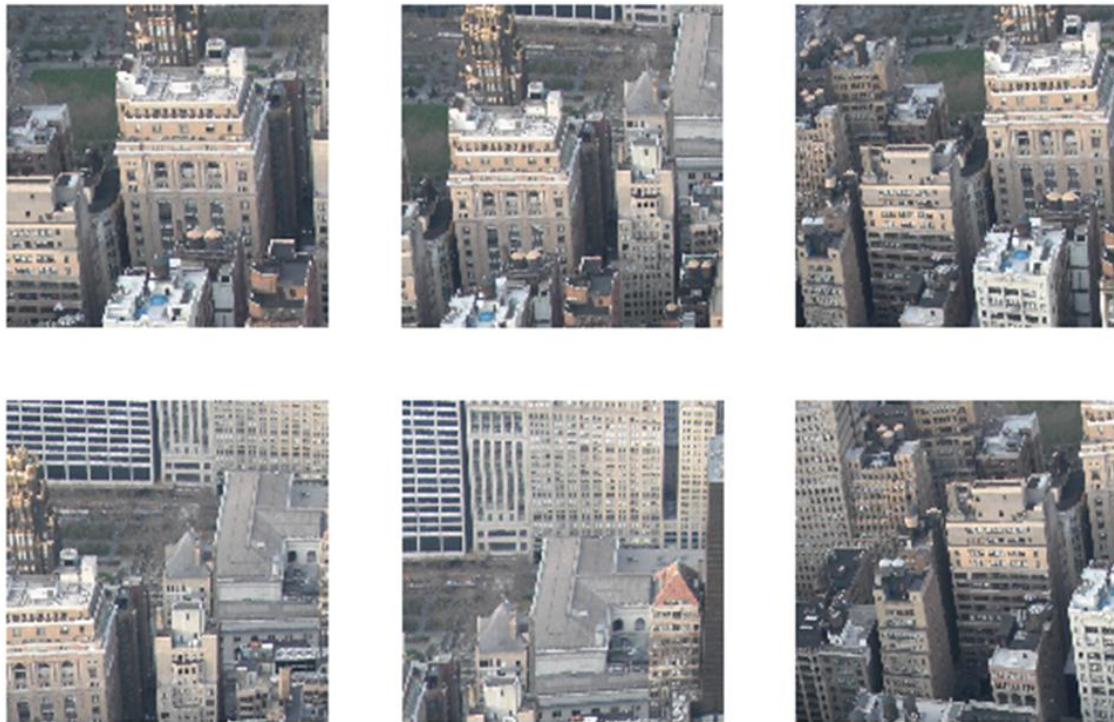


Figura 5-19. Capturas para la tercera prueba con un tamaño de 150x150.



Figura 5-20. Resultado para la tercera prueba con un tamaño de 150x150.

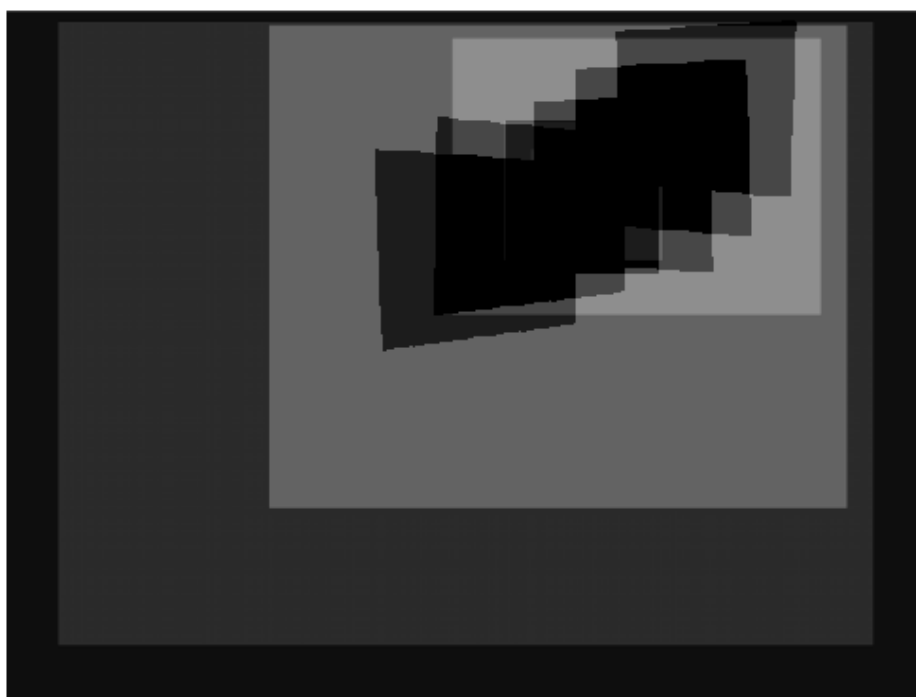


Figura 5-21. Mapa de interés resultante para la tercera prueba con tamaño de 300x300.

5.3.4 Comparación

En la Tabla 5-6 se muestran los tiempos para cada distinto nivel de calidad. Se comprueba que a menor tamaño de imagen más rápido es el proceso. Esta variación de tiempo sólo afecta al bloque del tratamiento de imagen, ya que el resto del coste computacional depende del tamaño del mapa de interés y del salto que se considere al recorrer sus elementos.

Tabla 5-6. Tiempos de ejecución de la tercera prueba en función de la calidad.

Tamaño imagen	Tiempo ejecución [s]
600x600	274.494
300x300	58.481
150x150	33.616

También se observa que, al disminuir la calidad, la estimación de la transformada es menos fiable, debido a que se obtienen un menor número de parejas de puntos similares. Esto se debe a que, al tener menor calidad, la densidad de puntos característicos disminuye, razón por la cual las capturas presentan mayor solapamiento entre ellas. Podría evitarse este mayor solapamiento reduciendo el número mínimo de puntos singulares en el área solapada, pero ello podría llevar a fallos en la estimación de la transformación. Con calidades altas ocurre lo contrario ya que, al extraerse más puntos, el área de solape para incluir dicho mínimo puede ser demasiado reducida y perder detalles de inclinación necesarios para la transformación. Por tanto, habría que modificar ligeramente este mínimo en función de la calidad con la que se trabaje.

Para las características de la prueba que estamos realizando se ha comprobado que el límite de tamaño para que se pueda formar la panorámica es de 150x150, es decir, un 25% de la calidad de la imagen original. Este mínimo es de interés pues, cuanto menor sea el tamaño de la imagen más rápido es el proceso de solapamiento. Además, suponiendo el caso en que los dispositivos no dispongan de un sistema operativo complejo, el almacenamiento y envío imágenes de gran tamaño puede ser algo poco recomendable.

6 CONCLUSIÓN

Para cumplir con los objetivos planteados en este proyecto se ha dividido el trabajo en dos bloques. El primero, un estudio de las herramientas de tratamiento de imagen disponibles en MATLAB, para determinar la correspondencia entre imágenes. El segundo, el diseño de la función de distribución que permita colocar los agentes de forma rápida y efectiva. Como añadido, la simulación diseñada para comprobar el funcionamiento puede ser usada en otros proyectos que necesiten el comportamiento de cámaras apuntando un área plana, permitiendo además poder variar las características de la cámara.

6.1. Solapamiento

Primero se han estudiado los métodos más usados para extraer puntos singulares de la imagen y las características que los definen. El escogido es el método SURF por eficacia, invariancia frente a cambios de escala y giros, y reducido tiempo de ejecución

A continuación se ha seleccionado el proceso MSAC para la estimación de la transformada a aplicar, por estar implementado en las funciones de MATLAB y adaptarse bien a las características asociadas a los puntos extraídos. Este método es el más fiable, pero su exactitud depende de la cantidad y exactitud de los puntos marcados como semejantes, lo que puede provocar variación entre el área que ocupa la imagen transformada y la que corresponde en realidad. Estas diferencias son mínimas consideradas de forma independiente, pero pueden provocar errores por acumulación al transformar capturas tomando como base imágenes mal transformadas. Se han ajustado los parámetros de las funciones para reducir todo lo posible este error sin afectar demasiado al coste computacional

Se ha comprobado que la pérdida de calidad hace que se extraigan menos parejas de puntos semejantes, empeorando la estimación de la transformada. Por tanto, el valor mínimo de calidad está ligado a las características del entorno y al número de cámaras con que se trabaje, ya que, como se ha comprobado, el error por acumulación de inexactitud en las transformaciones es mayor cuanto mayor es el número de capturas realizadas.

6.2. Distribución

De cara a la colocación de cámaras para la composición de imágenes panorámicas lo más económico en coste computacional es un método heurístico que las coloque una a una. Esto es debido a que, considerando los cuatro grados de libertad elegidos, las combinaciones son demasiadas para comprobar todas, provocando un incremento exponencial del tiempo al aumentar el número de cámaras.

En cuanto a la referencia para distribuirlos, se ha definido una matriz como un mapa de interés que representa la distribución de importancia de las distintas partes del área que se quiere estudiar. Como tiene que estudiarse dicho mapa, las dimensiones del mismo determinan el tiempo de ejecución que se emplea en buscar la posición óptima, por lo que si el proceso es lento puede reducirse el tamaño del mismo o, como se ha hecho en el código del proyecto, seleccionar saltos mayores que uno en los bucles que recorren la matriz.

6.3. Posibles continuaciones del trabajo

Por último añadir los campos en los que se puede profundizar para ampliar este proyecto:

- **Implementación real.** Si se dispone de los dispositivos, comprobar el desarrollo en una situación real. Para ello tendría que diseñarse el controlador de los dispositivos y el método de transmisión de datos. Esto podría requerir también que se reajustaran los parámetros o que se incluyesen filtros para poder solapar las imágenes.
- **Respuesta frente a fallos de la cámara o comunicación.** Incluir un código sencillo que, frente a fallos en los dispositivos o las imágenes tomadas, repita las capturas un número determinado de veces para ver si es problema temporal y, en caso contrario, continuar la distribución sin tener en cuenta dicho dispositivo.
- **Giros verticales.** Incluir este grado de libertad para ampliar las posibilidades de la imagen panorámica.
- **Límites de movimiento.** Establecer unos límites en las posiciones que puede ocupar un dispositivo, ya sea por temas de espacio o límite de transmisión de datos. Esta restricción debe incluirse en la función de optimización a la hora de seleccionar una posición válida.
- **Cambiar el mapa de interés.** Modificar las funciones para poder trabajar con zonas que no sean planas, como por ejemplo, superficies cilíndricas.

REFERENCIAS

- [1] M. T. Arauz Pisón, *Robot coordination to create collaborative panorama images*, Sevilla, España, 2016.
- [2] A. Criminisi, I. Reid y A. Zisserman, *A Plane Measuring Device*, Parks Road, Oxford, UK, 1997.
- [3] M. Hassaballah, A. A. Abdelmgeid y H. A. Alshazly, «Image features detection, description and matching,» de *Image Feature Detectors and Descriptors; Foundations and Applications*, Springer, 2016, pp. 11-45.
- [4] A. G. Chitu, L. J. M. Rothkrantz, J. Wojdel y P. Wiggers, *Comparison Between Different Feature Extraction Techniques for Audio-Visual Speech Recognition*, Delft University of Technology, Delft, Holanda.
- [5] E. L. Hall, R. P. Kruger, S. J. Dwyer, D. I. Hall, R. W. McLaren y G. S. Lodwick, «A survey of preprocessing and feature extraction techniques for radiographic images,» *IEEE Transactions on Computers*, vol. 20, n° 9, 1971.
- [6] O. Boullosa, *Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas*, Madrod, España, 2011.
- [7] J. Chao, A. Al-Nuaimi, G. Schroth y E. Steinbach, «Performance comparison of various feature detector-descriptor combinations for content-based image retrieval with JPEG-encoded query images,» de *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, Pula, 2013, pp. 29-34.
- [8] S. Smith y J. M. Brady, «SUSAN - a new approach to low level image processing,» *Int. Journal of Computer Vision*, vol. 23(1), pp. 45-78, 1997.
- [9] J. T. Arnfred y S. Winkler, «Fast-match: Fast and robust feature matching on large images,» *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3000-3003, 2015.
- [10] D. G. Lowe, «Distinctive images features from scale-invariant keypoints,» *Int. Journal of Computer Vision*, vol. 60(2), pp. 91-110, 2004.
- [11] H. Bay, T. Tuytelaars y L. Van Gool, «SURF: Speeded Up Robust Features,» *European Conference on Computer Vision*, pp. 404-417, 2006.
- [12] M. A. Fischler y R. C. Bolles, «Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,» *Communications of the ACM*, vol. 24(6), pp. 381-395, 1981.
- [13] P. H. S. Torr y A. Zisserman, «MLESAAC: A new robust estimator with application to estimating image geometry,» *Computer Vision and Image Understanding*, vol. 78(1), pp. 138-156, 2000.

Anexo A: Código

Distribución

```
%%%CÁMARA%%%
N=400;                %Nº pixels altura foto
M=400;                %Nº pixels base foto
Alpha=30*pi/180;      %Angulo lente horizontal
Beta=30*pi/180;       %Angulo lente vertical

%%%AGENTES%%%
Nagent=9;             %%Número de agentes
Ax=zeros(1,Nagent);
Ay=zeros(1,Nagent);
Az=zeros(1,Nagent);
Aang=zeros(1,Nagent);
Acap=zeros(N,M,3,Nagent); %%capturas

%%%IMAGEN ORIGINAL SIMULACIÓN%%%
imgtotal=imread('NY3.JPG');
f=double(imgtotal);
[IMRi, IMRj, n]=size(imgtotal);

%%%CREACIÓN MAPA INTERÉS%%%
d=2000;               %distancia al mapa objetivo
MapInter=crearmapinteres(IMRi,IMRj);
figure; imshow(uint8(255/18*MapInter));

%%%CÁLCULO RELACIONES NECESARIAS%%%
relpixcam=[(M/2)/(d*tan(Alpha/2)); (N/2)/(d*tan(Beta/2))];
relpixsim=[(600/2)/(d*tan(Alpha/2)); (600/2)/(d*tan(Beta/2))];
relpixint=relpixsim/5;
%Relaciones cruzadas
rel=relpixcam(1,1)/relpixint(1,1);
rel2=relpixsim(1,1)/relpixint(1,1);

%%dimensiones mapa interés en base a la relacion de la cámara
[imap, jmap]=size(MapInter);
iTot=round(imap*rel);
jTot=round(jmap*rel);
%%Tamaño captura en mapa de interés
Nm=round(N/(2*rel));
Mm=round(M/(2*rel));

%%%MATRICES NECESARIAS
%Mapa puntos característicos
SURFmap=zeros(iTot,jTot);
```

```

%panorama
Panor=uint8(zeros(iTot,jTot,n));
%panoramaView
View = imref2d([iTot jTot]);

%%FOTO INICIAL
%buscar foto de más interes sin girar y a máximo acercamiento
interesmax=0;
for i=Nm+1:imap-Nm
    for j=Mm+1:jmap-Mm
        interes=sum(sum(MapInter(i-Nm:i+Nm,j-Mm:j+Mm)));
        if interes>interesmax
            interesmax=interes;    i1=i;    j1=j;
        end
    end
end
%%Características primer agente
%se pone el centro de la foto del agente nº1 como origen de coord
Ax(1,1)=0;
Ay(1,1)=0;
Az(1,1)=d;
Aang(1,1)=0;
%%Captura
Acap(:, :, :, 1)=captura(Ax(1,1),Ay(1,1), Az(1,1),Aang(1,1),i1*rel2, ...
j1*rel2,relpixsim,Alpha,Beta,M,N,imgtotal);

%%Colocación de la captura en la panorámica
Panor(round(i1*rel-N/2):round(i1*rel+N/2-1),round(j1*rel-M/2) ...
:round(j1*rel+M/2-1),:)=Acap(:, :, :, 1);
Panorg=rgb2gray(Panor);
%%Actualización del mapa de interés
MapInter(i1-Nm:i1+Nm,j1-Mm:j1+Mm)=MpInter(i1-Nm:i1+Nm,j1-Mm:j1+Mm)-5;
%%Actualización mapa de puntos SURF
points1 = detectSURFFeatures(Panorg, 'MetricThreshold', 600, ...
'NumscaleLevels', 6);
punt=points1.Location;
[n,m]=size(punt);
for i=1:n
    SURFmap(round(punt(i,2)),round(punt(i,1)))=1;
end
% figure; imshow(Panor);
% figure; imshow(uint8(255/18*MapInter));

for a=2:Nagent
    interesmax=0;
%%Búsqueda del área mínima de mayor interés
    for i=Nm+1:3:imap-Nm
        for j=Mm+1:3:jmap-Mm
            puntos=sum(sum(SURFmap(round(i*rel-N/2*0.9): ...
round(i*rel+N/2*0.9),round(j*rel-M/2*0.9):round(j*rel+M/2*0.9))));
            if puntos>190 %%estudiar el mínimo de puntos para unir fotos
                interes=sum(sum(MapInter(i-Nm:i+Nm,j-Mm:j+Mm)));
                if interes>interesmax
                    interesmax=interes;
                    iint=i;    jint=j;
                end
            end
        end
    end
end

```

```

    end
end

    %%Cálculo de la posicion optima
yint=(iint-11)/relpixint(2,1);  xint=(jint-j1)/relpixint(1,1);
Ay(1,a)=yint;
[Ax(1,a), Az(1,a),Aang(1,a)]=optimoxint,yint,relpixint, ...
    Alpha,Beta,MapInter,d,i1,j1,interesmax);

    %%Simulación de la captura de la imagen
imagen=captura(Ax(1,a),Ay(1,a), Az(1,a),Aang(1,a),i1*rel2, ...
    j1*rel2,relpixsim,Alpha,Beta,M,N,imgtotal);
Acap(:, :, :, a)=imagen;
%figure();imshow(imagen);

%Unión de la captura y actualización de matrices
[MapInter,Panor,SURFmap,points1,Panorg,trans]=compose(MapInter,...
    Panor,Panorg,SURFmap,imagen,View,points1);
%     figure; imshow(uint8(255/18*MapInter));
%     figure; imshow(Panor);
end

%%RECORTE DEL MARGEN VACÍO%%
isup=0; iinf=0; jizq=0; jder=0;
for i=1:iTot
    for j=1:jTot
        if Panorg(i,j)~=0
            isup=i;
            break
        end
    end
    if isup~=0
        break
    end
end
for i=iTot:-1:1
    for j=1:jTot
        if Panorg(i,j)~=0
            iinf=i;
            break
        end
    end
    if iinf~=0
        break
    end
end
for j=1:jTot
    for i=1:iTot
        if Panorg(i,j)~=0
            jizq=j;
            break
        end
    end
    if jizq~=0
        break
    end
end
end

```

```

for j=jTot:-1:1
    for i=1:iTot
        if Panorg(i,j)~=0
            jder=j;
            break
        end
    end
    if jder~=0
        break
    end
end

PanorC=Panor(isup:iinf,jizq:jder,:);
figure;
imshow(Panor);

```

Óptimo

```

function [xopt,zopt,angopt]=optimo(xm,ym,relpix,Alpha,Beta, ...
MapInter,d,in,jn,intmax)
    %ENTRADAS: coord. xy del punto, relación pixel/u.l. del mapa,
    % ángulos focales, mapa de interés, y la distancia a este, coord.
    % matriciales origen,ángulos focales e interés mínimo.
    %SALIDAS: coord. xz y ángulo óptimo del agente

    [imax,jmax]=size(MapInter);
    costemin=inf;

    for phi=-18:2:18
        for z=d:10:d+200
            phir=phi*pi/180;                %Ángulo en radianes
            Yp=round(ym*relpix(2,1));        %Coord y de la cámara en pixel

            %%Cálculo coord horizontales extremos
            xa=xm-z*tan(phir);                %Coord x de la cámara
            xi=z*tan(phir-Alpha/2)+xa;         %Coord x del extremo izquierdo
            xd=z*tan(Alpha/2+phir)+xa;         %Coord x del extremo derecho
            Xip=round(xi*relpix(1,1));         %Coord x del izquierdo en pixel
            Xdp=round(xd*relpix(1,1));         %Coord x del derecho en pixel

            %%Cálculo dimensiones de la captura
            % Anchura en píxeles
            Wp=(Xdp-Xip);
            % Altura extremos
            Hi=(z/cos(Alpha/2-phir))*cos(Alpha/2)*tan(Beta/2);
            Hd=(z/cos(Alpha/2+phir))*cos(Alpha/2)*tan(Beta/2);
            % Media altura extremos en pixel
            Hipm=round(Hi*relpix(2,1));
            Hdpm=round(Hd*relpix(2,1));

            %%Parámetros límite trapecio
            m=(Hdpm-Hipm)/Wp;
            n=min(Hipm,Hdpm);
            Hmmax=max(Hipm,Hdpm);
            %%Cálculo interes

```

```

m=(Hdpm-Hipm)/Wp;
n=abs(Hipm-Hdpm);
Hmmax=max(Hipm,Hdpm);
Plantilla=zeros(2*Hmmax,Wp);
np=2*Hmmax*Wp;
    for i=1:2*Hmmax
        for j=1:Wp
            if Yp+in-Hmmax>=1 && Yp+in-Hmmax<=imax && ...
                Xip+jn>=1 && Xip+jn<=jmax
                    Plantilla(i,j)=MapInter(Yp+in-Hmmax+i,Xip+jn+j);
                end
            end
        end
    for j=1:Wp
        for i=1:abs(Hipm-Hdpm)-1
            if i-m*j-n>0
                Plantilla(abs(Hipm-Hdpm)-i,j)=0;
                Plantilla(2*Hmmax-abs(Hipm-Hdpm)+i,j)=0;
            end
        end
    end
Inter=sum(sum(Plantilla));

    %% FUNCIÓN DE COSTE %%
%%Parámetros
    kz=1.3;
    kang=1.7;
    K=1/1000;

    coste=-Inter+(K*intmax)^2*(kz*((z-d)/200)^2+ ...
        kang*(abs(phi)/18)^2);
    if coste<costemin
%%Salidas
        costemin=coste;
        zopt=z;
        xopt=xa;
        angopt=phi;
    end
end
end

```

Crear mapa de interés

```

function MapInter=crearmapinteres(IMRi,IMRj)
MapInter=ones(2500,3500);

%%MAPA PRUEBA 1 y 2
MapInter(400:2200,500:3300)=7;
MapInter(700:1500,1000:1700)=10;
MapInter(1000:1800,1600:3000)=10;
MapInter(1200:1600,1850:2400)=15;
MapInter(1000:1350,1300:1500)=15;
MapInter(1250:1500,1900:2200)=18;

```

```

%%MAPA PRUEBA 3
% MapInter(40:2300,200:3300)=3;
% MapInter(50:1800,1000:3200)=7;
% MapInter(100:1100,1700:3100)=10;
% MapInter(400:900,1900:2500)=12;
% MapInter(500:800,2000:2300)=13;

%%Reescalado
h=floor(IMRi/5); v=floor(IMRj/5);
MapInter=imresize(MapInter,[h v]);

```

Unión imágenes

```

function [MapInter,Panor,Panorg,SURFmap,points1, trans]=compose(...
    MapInter, Panor,Panorg,SURFmap,imagen,View,points1)
    %%ENTRADAS: Mapa de interés, panorámica, panorámica en escala de
    % grises,mapa puntos SURF, captura a unir, marco para transformación,
    % puntos característicos de la panorámica
    %%SALIDAS: Mapa de interés, panorámica, panorámica en escala de
    % grises,mapa puntos SURF y puntos característicos de la panorámica
    % actualizados; y la transformada aplicada

    %%Imagen a unir
    i2 = imagen;
    I2 = rgb2gray(i2);

    %%Detectar puntos singulares
    points2 = detectSURFFeatures(I2,'MetricThreshold',600, ...
    'NumscaleLevels',6);

    %%Extraer características
    [features1, valid_points1] = extractFeatures(Panorg, points1);
    [features2, valid_points2] = extractFeatures(I2, points2);

    %%Selección de parejas de puntos semejantes
    index_pairs = matchFeatures(features1, features2, ...
    'MatchThreshold',1.1,'MaxRatio',0.3,'Unique',true);
    matched_points1 = valid_points1(index_pairs(:, 1), :);
    matched_points2 = valid_points2(index_pairs(:, 2), :);
    %figure; showMatchedFeatures(Panorg , I2, matched_points1, ...
    matched_points2);

    %%Estimación de la transformación
    trans=estimateGeometricTransform(matched_points2, ...
    matched_points1,'projective', 'Confidence',99.95, ...
    'MaxNumTrials', 2600, 'MaxDistance',1.5);

    %%Transformación de la imagen
    warpedImage = imwarp(i2, trans,'OutputView',View);
    m=ones(size(I2));
    warpedm = imwarp(m, trans,'OutputView',View);
    warpedm = warpedm >= 1;

```

```

%%ACTUALIZACIÓN DE LAS MATRICES%%
%%Inclusión en la panorámica
%Objeto de unión
blender = vision.AlphaBlender('Operation', 'Binary mask', ...
'MaskSource', 'Input port');
Panor = step(blender, Panor, warpedImage, warpedm);

%%Reducción del mapa de interés
warpedmr=imresize(warpedm,size(MapInter));
MapInter=MapInter-5*warpedmr;

%%Estracción de puntos singulares y descriptor
Panorg=rgb2gray(Panor);
points1 = detectSURFFeatures(Panorg,'MetricThreshold',600, ...
'NumscaleLevels',6);

%%Mapeado de los puntos característicos
M=points1.Location;
SURFmap=0*SURFmap;
[n,m]=size(M);
for i=1:n
    SURFmap(round(M(i,2)),round(M(i,1)))=1;
end

```

Captura

```

function imagen=captura(xa,ya,za,ang,i1,j1,rpix,Alfa,Beta,M,N,imorig)
%ENTRADAS: coord. cartesianas, angulo, coord. matriciales origen,
%relación pixel/u.l., ángulos focales, imagen original.

imagen=uint8(zeros(N,M,3));
angr=ang*pi/180; %angulo en radianes
[i1imt, j1imt,n]=size(imorig);
for a=1:M
%%Cálculo coordenada horizontal
    al=(angr-Alfa/2)+a*Alfa/M;
    x=za*tan(al)+xa;
    t=za/cos(al); %Distancia al punto
    ti=t*cos((0.5-a/M)*Alfa); %Proyección en el eje

%%Cálculo coordenada vertical
    h=2*ti*tan(Beta/2); %Altura
    for b=1:N
        y=-h/2+b*h/N; %División altura
        %%Coordenadas matriciales
        i=floor((y+ya)*rpix(2,1))+i1;
        j=floor(x*rpix(1,1))+j1;
        %%Se comprueba que el punto esté en la imagen original
        if i<i1imt && j<j1imt && i>0 && j>0
            imagen(b,a,:)=imorig(i,j,:);
        end
    end
end
end

```