

Jornadas de Automática

Composición fotográfica mediante el uso de un dron

Sánchez García, J.M.^{a,*}, Sánchez-Moreno, J.^a, Moreno-Salinas, D.^a

^a Dpto. de Informática y Automática, E.T.S. de Ingeniería Informática, UNED, C/ Juan del Rosal, nº 16, 28040, Madrid, España.

To cite this article: Sánchez García, J.M., Sánchez-Moreno, J., Moreno-Salinas, D., 2024. Photo composition using a drone. *Jornadas de Automática*, 45. <https://doi.org/10.17979/ja-cea.2024.45.10702>

Resumen

La composición fotográfica, conocida como mosaicos, es crucial en aplicaciones donde no es posible capturar toda la extensión de grandes superficies en una sola toma. Por ende, se requiere fotografiar secciones más pequeñas para luego componerlas y lograr una reproducción lo más precisa posible de la realidad. En este trabajo se presenta el resultado de aplicar los principios de las distintas etapas necesarias para crear un mosaico, complementado con el uso de un dron para la captura de las imágenes. La creación del mosaico implica técnicas avanzadas de procesamiento de imágenes que facilitan la detección de características, la transformación geométrica y la alineación de píxeles. Sin embargo, la experimentación con diferentes algoritmos ha revelado que no siempre es viable encontrar una transformación geométrica que produzca un mosaico de calidad, especialmente cuando las características de la fotografía no son óptimas, lo cual puede ser atribuible, en parte, a la resolución de los dispositivos fotográficos utilizados.

Palabras clave: Robots voladores, Planificación de trayectorias y trayectorias, Detección, Programación y Visión.

Photo composition by using a drone

Abstract

The photographic composition, commonly known as mosaics, holds particular significance in applications where capturing the entirety of large surfaces in a single frame is impractical. Thus, it necessitates taking photographs of smaller sections and subsequently composing them to achieve a faithful reproduction of reality. This work presents the outcome of applying the principles of the various stages required to create a mosaic, augmented using a drone for image capture. Creating a mosaic involves advanced image processing techniques that enable feature detection, geometric transformation, and pixel alignment. However, experimentation with different algorithms has revealed that finding a geometric transformation that yields a quality mosaic is not always feasible, particularly when the characteristics of the photographs are suboptimal, partly due to the resolution of the photographic devices used.

Keywords: Flying robots, Trajectory and path planning, Sensing, Programming and Vision.

1. Introducción

En la actualidad, el empleo de vehículos aéreos no tripulados (UAV), se ha vuelto omnipresente en muchos sectores industriales, desempeñando una amplia variedad de tareas. En muchas de sus aplicaciones tanto civiles como militares, la visión artificial desempeña un papel crucial. Un caso concreto es la aplicación de las técnicas de mosaico.

El uso del mosaico se ha extendido ampliamente a

diversos campos, como la edición fotográfica, la arquitectura, la medicina, la biometría, el análisis de texturas, la inspección de fachadas, la robótica, las imágenes satelitales y el seguimiento de catástrofes naturales, entre otros ejemplos. Esta aplicación concreta de la visión artificial recurre a la aplicación de distintas técnicas en varias etapas: registro, reproyección y combinación (*registering, warping y blending*, respectivamente).

En este trabajo se recurre al cuadricóptero de bajo coste Crazyflie 2.0 de la firma Bitcraze. El Crazyflie 2.0 ofrece, a pesar de las limitaciones impuestas por su tamaño, muchas posibilidades gracias a su hardware y a su arquitectura *open source*. Procesadores como el STM32, el ESP32 y el GAP8, ofrecen muchas posibilidades en los desarrollos. Así, mediante la planificación de una trayectoria y los algoritmos de visión artificial adecuados, el Crazyflie es capaz de realizar un fotografiado de su entorno de manera autónoma para, posteriormente, realizar el correspondiente mosaico final.

El trabajo se organiza de la siguiente forma. En las secciones 2 y 3 se describen las arquitecturas hardware y software del dron. La Sección 4 explica las técnicas empleadas en la creación de los mosaicos, presentando algunos resultados. La Sección 5 aborda las trayectorias del dron para la toma de imágenes. Finalmente, la Sección 6 presenta conclusiones y algunas líneas de continuación.

2. Arquitectura hardware

La arquitectura hardware mínima necesaria es la formada por el dron Crazyflie 2.0 (Figura 1) junto a los módulos AI-deck 1.1(Figura 2.a) y Flowdeck v2 (Figura 2.b). El Crazyflie 2.0 tiene instalados dos microcontroladores: el STM32 para el control del vuelo, lectura de sensores y telemetría, y el NRF51 que maneja las comunicaciones (radio a 2.4 GHz y el Bluetooth LE), y la gestión de la energía y de los módulos de expansión instalados.



Figura 1: Crazyflie 2.0. Fuente: Bitcraze AB.

El módulo AI-deck 1.1 expande las capacidades de computación, permitiendo cálculos basados en algoritmos de inteligencia artificial gracias a su procesador GAP8 de arquitectura RISC-V, además de permitir comunicaciones Wi-Fi mediante un chip NINA-W102 dotado de un procesador ESP32. Adicionalmente, tiene instalada una cámara monocroma de ultra bajo consumo de 320x320 píxeles. El módulo Flowdeck v2 permite detectar los movimientos en cualquier dirección.

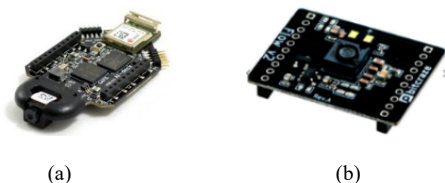


Figura 2: Módulos AI-deck (a) y Flow deck (b). Fuente: Bitcraze AB.

El dron ofrece poca potencia y autonomía por lo que las pruebas en exteriores no han resultado satisfactorias incluso con corrientes de aire suaves. Además, los cambios de iluminación y texturas de la superficie han influido en el módulo Flowdeck, presentado problemas en la precisión de la navegación.

3. Arquitectura software

El microprocesador STM32 del Crazyflie 2.0 presenta una capa de software para el control del vuelo. Por otra parte, el módulo AI-deck 1.1 consta de distintas capas de software para gestionar sus dos microprocesadores, el ESP32 y el GAP8. La ejecución del código para la captura de la imagen se efectúa en el GAP8, realizándose una retransmisión continua de la imagen captada por la cámara y enviándose por medio de un canal Wi-Fi a un host a través del ESP32. A lo largo de la trayectoria seguida durante el vuelo, se capturan y almacenan los fotogramas que formarán parte del mosaico.

4. Técnica de mosaico

Las técnicas de mosaico tienen como finalidad ensamblar una serie de imágenes que presentan regiones de superposición. Así, el mosaico permite conseguir una única imagen con un campo de visión más amplio sin comprometer la resolución. Gracias a estas técnicas es posible salvar las limitaciones inherentes a las cámaras de no poder capturar una única imagen de un área muy extensa sin necesidad de alejarse para así ampliar el campo de visión del dispositivo de captura.

Los algoritmos de mosaico no están exentos de puntos débiles para conseguir imágenes de calidad. La iluminación, los objetos en movimiento, el ruido en las imágenes, entre otros, son detalles para tener en cuenta ya que pueden dar lugar a imágenes con aspectos visuales de mala calidad (Pandey *et al.*, 2019).

La técnica de mosaico consta de tres etapas: registro de las imágenes (*register*), reproyección (*warping*) y combinación (*blending*).

4.1. Registro de las imágenes

En esta primera etapa se trata de establecer una correspondencia geométrica entre pares de imágenes de la misma escena.

La mayoría de los algoritmos están basados en el dominio espacial. De las técnicas disponibles en esta categoría se hace uso de las que se basan en la extracción de características. Se debe tener en cuenta que las imágenes pueden tomarse desde diferentes perspectivas y, posiblemente, distintas exposiciones, por lo que es necesario plantear una solución para conseguir su alineación. La extracción de características consiste en extraer los puntos de interés de la imagen, que son aquellos puntos en los que la señal cambia en dos dimensiones, como resultado de discontinuidades geométricas que se producen, por ejemplo, en bordes, esquinas, píxeles, fronteras entre regiones o contornos (Rosten *et al.*, 2008; Sakharkar and Gupta, 2013). Las características deseables de un punto de interés son: que tenga un contenido relevante y rico en color, brillo y contraste; que su posición y representación estén bien definidas; que sea invariante a la rotación y al escalado; y que sea insensible a los cambios de la iluminación.

Dentro de la taxonomía de los métodos usados para el proceso de registro, los métodos basados en características son los que presentan mayor robustez y mejores resultados a expensas de su elevado coste computacional (Ghosh and

Kaabouch, 2015). A continuación, se describen las técnicas analizadas:

a) Detector de esquinas de Harris: Busca la diferencia de intensidades para un desplazamiento (u,v) en las ocho direcciones de la vecindad de un píxel.

b) Transformada de características de escala invariante (SIFT - *Scale Invariant Feature Transform*): Detecta características a bajo nivel y es invariante a las traslaciones, las rotaciones y el escalado. Busca las características invariantes a través de múltiples escalas usando la diferencia de gaussianas (DoG) (1) entre las distintas escalas (k) , evitando el coste computacional de aplicar la laplaciana de la gaussiana L (Lowe, 2004).

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (1)$$

c) Características robustas aceleradas (SURF - *Speeded Up Robust Features*): Presenta mejoras respecto a la velocidad de cálculo, rotaciones, cambios de escala y desenfoques. Hace uso de una matriz hessiana (2) calculando su determinante para cada punto de la imagen:

$$\mathcal{H}(x, y, \sigma) = \begin{pmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{pmatrix} \quad (2)$$

Las derivadas segundas de cada uno de los términos de la matriz se corresponden con las aproximaciones discretas de las máscaras de la Figura 3.

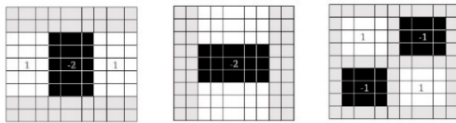


Figura 3: Aproximación gaussiana de segundo orden: L_{xx} , L_{yy} y L_{xy} .

El valor del determinante de (2) viene dado por (3), donde w es un peso relativo para balancear la expresión y D la DoG:

$$Det(\mathcal{H}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (3)$$

La finalidad de (3) es calcular los valores máximos que determinarán los puntos característicos. Un aspecto clave de este método es que hace convolucionar una imagen integral, en la que el valor de cada píxel es la suma de sus píxeles vecinos superiores y sus píxeles vecinos a la izquierda (Figura 4), con el filtro (Mistry and Banerjee, 2017).

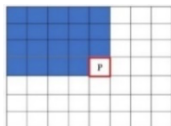


Figura 4: Imagen integral.

Las imágenes creadas a diferentes escalas se comparan entre sí en bloques de 3x3, seleccionando el píxel de mayor determinante según (3) como punto característico. Alrededor del punto característico, se seleccionan los puntos espaciados una distancia igual al tamaño del filtro y se aplican dos filtros de Haar para el cálculo de las derivadas parciales. Finalmente, se va formando un vector de dirección con la suma de todos los puntos que están en cada arco de longitud

$\pi/3$, siendo el vector de mayor norma el asignado como orientación del punto característico.

d) Características de las pruebas de segmento aceleradas (FAST - *Features from Accelerated Segment Test*): Es computacionalmente más eficiente y rápido que otros métodos, haciéndolo muy adecuado para el procesamiento de imágenes en aplicaciones de visión por computador en tiempo real y robótica móvil. Inicialmente se considera un círculo de radio 3 píxeles que abarca 16 píxeles alrededor de cada píxel candidato a ser una esquina. El candidato será una esquina si existe un conjunto de n píxeles contiguos dentro del círculo que sean más brillantes que el píxel candidato más un umbral, o son todos más oscuros que el píxel candidato menos un cierto umbral (Rosten and Drummond, 2006; Ghosh and Kaabouch, 2015).

Lo que hace al algoritmo más rápido es que sólo se comparan los píxeles 1, 5, 9 y 13 de los 16 posibles (Figura 5) antes de decidir si se continúa con la comparación de los píxeles restantes.

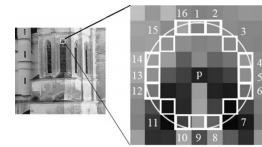


Figura 5: Candidato a punto de interés (Ghosh and Kaabouch, 2015, p. 5).

En (Rosten and Drummond, 2006) se mencionan los problemas que puede presentar este método, como son la influencia en el rendimiento del orden de evaluación de los píxeles o la detección de puntos de interés adyacentes.

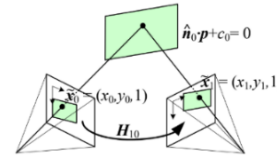


Figura 6: Homografía. Fuente: (Szeliski, 2022, p. 62).

4.1.1. Homografías

Es posible establecer una relación geométrica entre imágenes tomadas desde distintas posiciones si los puntos están en el mismo plano 3D. Como existe una relación geométrica entre el mundo real y el plano de la imagen es posible establecer una relación entre dos imágenes del mismo plano. La homografía es una transformación proyectiva que permite determinar la correspondencia entre dos formas geométricas, de forma que a cada uno de los puntos y las rectas les corresponden, respectivamente, un punto y una recta de la otra. En la Figura 6 se deduce que la homografía H permite la correspondencia entre puntos, concretamente, $\tilde{x}_1 \approx H_{10}\tilde{x}_0$.

En la Figura 7 se representa una cámara del tipo *pinhole* que toma imágenes desde diferentes ángulos a través del mismo centro óptico y cuyos planos se van superponiendo parcialmente. En ella se puede establecer que existe una homografía H_{01} que proyecta el plano P_1 en P_0 . La homografía en el plano P_2 se calcula mediante composición de la homografía anterior H_{01} y la homografía H_{12} que proyecta P_2 en P_1 . Por tanto, se puede establecer que, mediante la composición de homografías $H_n = H_{n-1} \cdot H_{n-1,n}$, las distintas imágenes quedan proyectadas en el mismo plano estando

relacionadas entre sí por una homografía (Hartley and Zisserman, 2003).

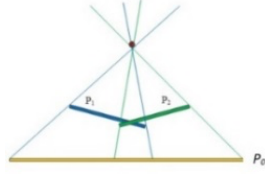


Figura 7: Homografía de los planos P_1 y P_2 sobre el plano P_0 .

Mediante el uso de coordenadas homogéneas se obtiene la matriz de 3×3 de acuerdo a (3). De acuerdo con (4), la coordenada proyectada en \mathbb{R}^2 es (x_d, y_d) , mientras que $(\tilde{x}_d, \tilde{y}_d, \tilde{z}_d)$ y $(\tilde{x}_s, \tilde{y}_s, \tilde{z}_s)$ son sus coordenadas homogéneas proyectadas y de origen, respectivamente.

$$p = \begin{bmatrix} \tilde{x} \\ \tilde{z} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} \text{ con } \tilde{z} \neq 0 \quad (3)$$

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_s \\ \tilde{y}_s \\ \tilde{z}_s \end{bmatrix} \quad (4)$$

Si se elimina la restricción de la última línea de la matriz en (4) y sabiendo que $\tilde{z}_s = 1$ se obtiene la matriz de homografía (5), consiguiendo así una transformación proyectiva.

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_s \\ \tilde{y}_s \\ 1 \end{bmatrix} \quad (5)$$

La proyectividad es un mapeo lineal de coordenadas homogéneas a través de una matriz de 3×3 no singular. Tanto en el caso de imágenes de una escena planar tomadas desde distintas posiciones como desde la misma posición la transformación entre dos imágenes es una transformación en coordenadas homogéneas (Zagrouba et al., 2009).

Una vez realizada la extracción de características y creados los vectores descriptores que almacenan sus propiedades como, por ejemplo, la magnitud y la orientación, se deben encontrar coincidencias entre pares de puntos de las características para poder crear la homografía, y establecer así la relación geométrica entre las imágenes.

Para un par de puntos característicos en los que existe coincidencia y a partir de (3) y (5), se obtienen los coeficientes de la matriz de transformación (6) y (7) para cada par de puntos coincidentes $x_d^{(i)}$ e $y_d^{(i)}$, siendo i el índice correspondiente al par de puntos coincidentes.

$$x_d^{(i)} = \frac{\tilde{x}_d^{(i)}}{\tilde{z}_d^{(i)}} = \frac{h_{11}x_s^{(i)} + h_{12}y_s^{(i)} + h_{13}}{h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}} \quad (6)$$

$$y_d^{(i)} = \frac{\tilde{y}_d^{(i)}}{\tilde{z}_d^{(i)}} = \frac{h_{21}x_s^{(i)} + h_{22}y_s^{(i)} + h_{23}}{h_{31}x_s^{(i)} + h_{32}y_s^{(i)} + h_{33}} \quad (7)$$

siendo $\tilde{z}_d^{(i)} = 1$, expresándolo en forma matricial y generalizando para todos los n pares de puntos de las imágenes se obtiene (8):

$$Ah = 0 \quad (8)$$

donde A y h se corresponden con (9) y (10), respectivamente.

$$A = \begin{bmatrix} x_s^{(1)} y_s^{(1)} 1 0 0 0 -x_d^{(1)} x_s^{(1)} -x_d^{(1)} y_s^{(1)} -x_d^{(1)} \\ 0 0 0 x_s^{(1)} y_s^{(1)} 1 -y_d^{(1)} x_s^{(1)} -y_d^{(1)} y_s^{(1)} -y_d^{(1)} \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ x_s^{(i)} y_s^{(i)} 1 0 0 0 -x_d^{(i)} x_s^{(i)} -x_d^{(i)} y_s^{(i)} -x_d^{(i)} \\ 0 0 0 x_s^{(i)} y_s^{(i)} 1 -y_d^{(i)} x_s^{(i)} -y_d^{(i)} y_s^{(i)} -y_d^{(i)} \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ x_s^{(n)} y_s^{(n)} 1 0 0 0 -x_d^{(n)} x_s^{(n)} -x_d^{(n)} y_s^{(n)} -x_d^{(n)} \\ 0 0 0 x_s^{(n)} y_s^{(n)} 1 -y_d^{(n)} x_s^{(n)} -y_d^{(n)} y_s^{(n)} -y_d^{(n)} \end{bmatrix} \quad (9)$$

$$h = \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} \quad (10)$$

La ecuación (11) es una ecuación homogénea que resulta en problema de vectores propios en los que la solución es el vector propio con el menor valor propio de la matriz $A^T A$.

$$A^T A h = \lambda h \quad (11)$$

Su resolución se puede realizar mediante la descomposición en valores singulares SVD. Si $A = U \Sigma V^T$, entonces U son los vectores propios de la matriz AA^T , V los vectores propios de la matriz $A^T A$, siendo V el resultado de la ecuación, y Σ la matriz diagonal de valores propios.

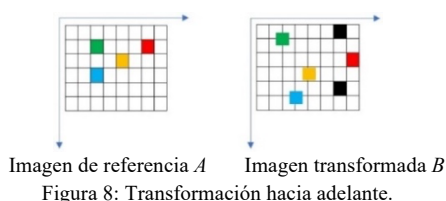
El principal problema en la unión de imágenes para la creación de los mosaicos es encontrar un modelo que relacione los píxeles de una imagen con los de otra a partir de coincidencias entre puntos característicos (Szeliski, 2022).

En el caso de un dron desplazándose a lo largo de una superficie y que toma imágenes desde distintas posiciones sobre el mismo plano, se podría pensar en aplicar una simple transformación euclídea ya que los píxeles de las imágenes deberían presentar una buena alineación. Esta transformación incluye la traslación y la rotación, pero no trata la distorsión de la perspectiva (Nagabhushan and Jathanna, 2015). De acuerdo con (Hartley and Zisserman, 2003) y con (Zagrouba et al., 2009), cuando todos los puntos de la imagen son coplanares, las imágenes siguen estando relacionadas por una homografía.

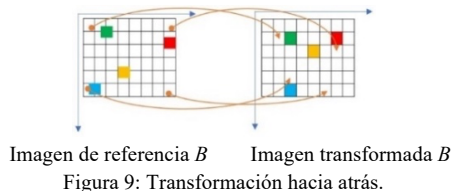
Durante el cálculo de la homografía se pueden encontrar coincidencias no válidas entre puntos de interés, ya que pueden coincidir en valor, pero no ser el mismo punto. Para el tratamiento de estos valores anómalos se hace uso de un algoritmo muy común en el campo del tratamiento de imágenes, el *RANdom SAmple Consensus* (RANSAC).

4.2. Proyección o warping

En esta segunda etapa se construye el mosaico mediante la alineación de las imágenes. A partir de una secuencia ordenada de imágenes se calculan las homografías por pares y se va componiendo el mosaico. Se debe aplicar una primera transformación hacia adelante (*forward warping*) a los píxeles de la imagen. Durante esta primera transformación pueden aparecer problemas de localización, donde los centros de los píxeles podrían no coincidir y, por lo tanto, tener que aplicar alguna técnica de interpolación, o problemas de no existencia de correspondencia entre píxeles, quedando píxeles vacíos. Para evitar estos problemas, esta transformación hacia adelante se aplica únicamente a las esquinas de la imagen. Las imágenes A y B de la Figura 8 están relacionadas por medio de una función de transformación $A(x, y) = B(T(x, y))$.



A la imagen B con nuevas dimensiones se le aplica una transformación inversa (*backward warping*), mediante homografía $f(x, y) = g(T^{-1}(x, y))$ (Figura 9), resolviéndose así el problema de los píxeles vacíos.



En este caso también se puede repetir el problema de la localización, que puede ser solucionado una vez más mediante el vecino más cercano o mediante interpolación.

4.3. Blending

La finalidad de esta última etapa es minimizar las inconsistencias en la imagen combinada que se producen en las zonas de solapamiento por las uniones de las imágenes.

De acuerdo con (Sakharkar and Gupta, 2013) y (Pandey and Pati, 2019) se puede realizar la siguiente clasificación de métodos: a) métodos de suavizado de transición, entre los que se encuentran el *alpha blending*, el *feathering* y el *blending* piramidal, b) métodos de búsqueda óptima y c) métodos de mezcla híbrida. Los métodos analizados corresponden a los de suavizado de transición:

a) El método *alpha blending* hace referencia a la opacidad del píxel, aplicando una media simple de las imágenes en el área de solapamiento.

b) El método de *feathering* asigna un peso a cada píxel mediante el cálculo de la transformada de distancia de una imagen binaria. Esta transformada de distancia asigna un número a cada píxel de la imagen binaria que es la distancia

entre ese píxel y el píxel de la imagen binaria distinto de cero más cercano. La Figura 11.a muestra el resultado de la unión de dos imágenes tomadas con el dron mediante este método.



Figura 10: Ejemplos de aplicación de los métodos de *feathering* (a) y de *blending* piramidal (b).

c) El método de *blending* piramidal (Burt and Adelson, 1983) realiza el suavizado a distintos niveles de resolución mediante el uso de pirámides paso-banda. La imagen piramidal no es más que una representación jerárquica formada por distintas resoluciones de la imagen, correspondiendo el nivel más bajo a la resolución más alta. La técnica consiste en una etapa de reducción de la imagen a distintos tamaños mediante gaussiana, y otra de expansión de la gaussiana en el nivel más bajo y restarlo de la imagen en el nivel en el que se esté. La Figura 10.b presenta el resultado de la unión de dos imágenes del dron con este método.

4.4 Ejemplos de aplicación de las técnicas

Las figuras 11, 12 y 13 muestran resultados de varios mosaicos obtenidos después de aplicar los métodos SIFT, SURF y FAST, respectivamente.

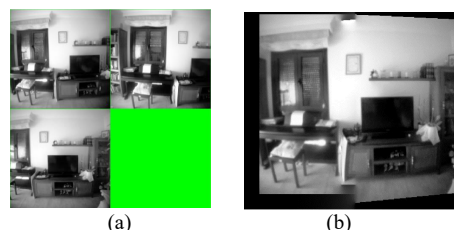


Figura 11: Conjunto de imágenes para crear un mosaico (a) y resultado del método SIFT con *blending* piramidal (b).

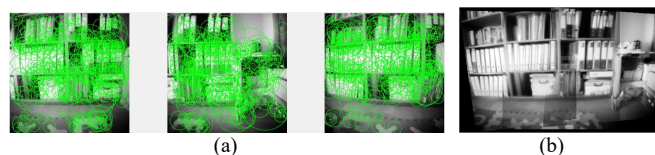


Figura 12: Imágenes y sus puntos característicos (a) y resultado del método SURF con *alpha blending* (b).

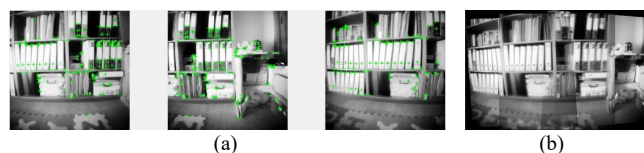


Figura 13: Imágenes y sus puntos característicos (a) y resultado del método FAST con *alpha blending* (b).

La Tabla 1 resume los tiempos y las homografías válidas que se han ido calculando a lo largo del proceso de generación del mosaico en base al número de imágenes. Se aprecia que para un número pequeño de imágenes los tiempos son muy similares, incrementándose en gran medida cuando el número

de imágenes que forman parte del mosaico final aumenta. En el caso de FAST, a partir de 6 imágenes, los resultados de los mosaicos obtenidos no han sido satisfactorios, debido a la baja detección de características que permitan realizar una asociación entre puntos

Imágenes	SIFT		SURF		FAST	
	3	6	3	6	3	6
Homografías	7	28	3	27	9	-
Iteraciones	2876	9095	2876	9095	2876	-
Tiempo (s)	3.76	42.14	3.68	38.62	3.37	-

Tabla 1: Resumen de homografías y tiempos de los ejemplos ejecutados.

5. Trayectoria de vuelo

Para poder aplicar una transformación geométrica entre pares de características extraídas de las imágenes, las fotografías tomadas deben presentar zonas de solapamiento.

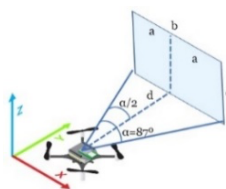


Figura 14: Campo de visión de la cámara.

De acuerdo con las especificaciones de la cámara del AI-deck, la resolución es de 320x320 píxeles y el campo de visión, tanto horizontal como vertical es $\alpha=87^\circ$. A partir de las relaciones mostradas en la Figura 14, se obtienen las dimensiones del marco de la fotografía según (12) y (13).

$$b = 2d * \tan(\alpha/2) \tag{12}$$

$$c = b \tag{13}$$

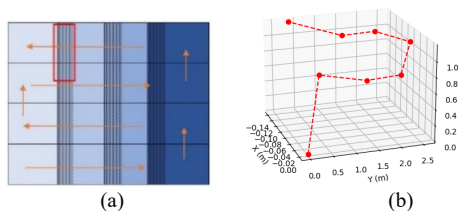


Figura 15: Ejemplo de trayectoria de vuelo seguida para capturar imágenes y formar un mosaico. Vista frontal (a) y vista tridimensional (b).

Así, a partir de (12) y (13) se pueden obtener los desplazamientos necesarios que se deben realizar para permitir la superposición de las imágenes. La Figura 15.a muestra la trayectoria buscada. Para una mejor visualización, solo se muestra el solapamiento horizontal entre imágenes. La Figura 15.b muestra una trayectoria tridimensional, en la que puede apreciarse que no se consigue toda la precisión deseada en los movimientos.

6. Conclusiones

Las técnicas de mosaico se pueden entender como la aplicación, de una manera ordenada, de técnicas que son

utilizadas en el tratamiento digital de imágenes. La baja resolución de las imágenes obtenidas por la cámara monocroma del módulo AI-deck 1.1, influye en la calidad final de los mosaicos. En este aspecto se abre una línea de trabajo en la que reemplazar la cámara monocroma por una RGB del propio ecosistema de Bitcraze o utilizar alguna cámara compatible con el hardware y que ofrezca mayor resolución.

El uso del módulo Flow deck v2 durante las pruebas en exteriores pone de manifiesto algunas de sus limitaciones en la navegación y el control. Si bien es cierto que en interiores está recomendado una buena iluminación, superficies poco brillantes y con texturas, en ocasiones puede producirse mucho reflejo de la superficie en el sensor provocando movimientos erráticos. De igual manera sucede cuando el dron entra en zonas de sombra, al no ser capaz el sensor de detectar la textura.

Como línea de investigación futura se pueden añadir módulos al Crazyflie 2.0 para mejorar su posicionamiento en interiores y utilizar el potencial del módulo de AI-deck 1.1, en el que poder implementar las técnicas para realizar el mosaico. En cuanto a la planificación de trayectorias se pueden aplicar técnicas heurísticas o redes neuronales. En el plano de las técnicas de mosaico se dejan abiertas líneas de trabajo futuro, como, por ejemplo, hacer uso de FPGA (*Field Programmable Gate Array*) para estudiar su contribución en la aceleración de la ejecución de los distintos algoritmos.

Referencias

Burt, P. J., Adelson, E. H., 1983. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics* 2, 217-236. DOI: 10.1145/245.247

Ghosh, D., Kaabouch, N. 2015. A survey on image mosaicing techniques. *Journal of Visual Communication and Image Representation* 34, 1-11. DOI: 10.1016/j.jvcir.2015.10.014

Hartley, R., Zisserman, A., 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York.

Lowe, D.G., 2004. Distinctive image features from scale-invariant key points. *International Journal of Computer Vision* 60, 91-110. DOI: 10.1023/B:VISI.0000029664.99615.94

Pandey, A., Pati, U.C., 2019. Image mosaicking: a deeper insight. *Image and Vision Computing* 89, 236-257. DOI: 10.1016/j.imavis.2019.07.002

Mistry, D., Banerjee, A., 2017. Comparison of feature detection and matching approaches: SIFT and SURF. *Global Research and Development Journal for Engineering* 2, 7-13.

Nagabhushan, P., Jathana, V.E., 2015. Mosaicing of text contents from consecutive frames in pedestal shot videos. *International Journal of Engineering Research and Technology* 4, 1006-1013. DOI: 10.17577/ijertv4is070808

Rosten, E., Drummond, T., 2006. Machine learning for high-speed corner detection. *European Conference on Computer Vision* 1, 430-443. DOI: 10.1007/11744023_34

Rosten, E., Porter, R., Drummond, T., 2008. Faster and better: a machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 105-119. DOI: 10.1109/TPAMI.2008.275

Sakharkar, V., Gupta, S.R., 2013. Image Stitching Techniques-An overview. *Int. J. Comput. Sci. Appl.* 6(2), 324-330

Szeliski, R., 2022. *Computer Vision: Algorithms and Applications*. Springer.

Zagrouba, E., Barhoumi, W., Amri, S. 2009. An efficient image-mosaicing method based on multifeatured matching. *Machine and Vision Applications* 20, 139-162. DOI: 10.1007/s00138-007-011.