

Mapas mediante imágenes tomadas por un UAV

Rubén Pérez Conte

Resumen— El propósito de este trabajo es investigar sobre los procesos necesarios para crear mapas a partir de vistas aéreas, también conocidos como ortofotos, sacando estas vistas de un vídeo de una zona sobrevolada. En este trabajo se estudian los principales algoritmos de detección de puntos, la extracción de características de estos puntos y el cálculo necesario para transformar y concatenar las diferentes imágenes, evaluando diferentes opciones. Se describe el proceso necesario para la obtención de estos mapas desde la lectura del vídeo hasta la obtención de los primeros resultados sobre los que aplicaremos diferentes mejoras con el fin de mejorar este primer resultado. La evaluación de los resultados obtenidos se hace utilizando una ortofoto real sacada de un SIG y poniéndola en contraposición con el software comercial que ha dado los mejores resultados. Se concluye que nuestro proceso obtiene resultados similares a los del estado del arte si añadimos una información manual extra. La mejora futura vendría por la eliminación de esta entrada manual.

Palabras clave—detector de puntos característicos, emparejamiento, transformación geométrica, ortofoto.

Abstract— The purpose of this paper is to investigate the processes needed to create maps from aerial views, also known as orthophotos, taking these views from a video of the overflown area. In this paper we tackle the main algorithms related to point detection, feature extraction and the calculations of the required to transform and concatenate the different images. In this sense, we have studied and evaluated several options. The necessary process to obtain these maps are described from the video load to the first results on which we apply different enhancements to improve the result. The evaluation of the results is done a real orthophoto taken from a GIS and compared with the commercial software that gives the best result. We conclude that our process achieves a similar performance to the state of the art if we add an extra manual information. Future improvements would come from the suppression of this manual help.

Index Terms—key points detector, matching, geometrical transformation, orthophoto.



1 INTRODUCCIÓN

EN la actualidad existen múltiples usos para mapas obtenidos a partir de vistas aéreas. Tras su inclusión en un Sistema de Información Geográfica (SIG) podemos pensar en usos tan diversos como la planificación de usos del suelo: agricultura, minería, infraestructuras. Estos mapas son generados con imágenes tomadas por satélites y aviones que han sobrevolado la zona en cuestión a gran altitud. Estas imágenes tienen la característica de abarcar una gran área del suelo proporcionando una vista cenital sin apenas deformación por la perspectiva facilitando en gran medida su uso. No obstante tienen la desventaja de ser muy costosas.

Los UAV son pequeños aviones no tripulados con una gran versatilidad. Entre los usos que se pueden dar nos interesa su capacidad para sobrevolar y filmar zonas con bajo coste. Esta versatilidad, agilidad y bajo coste permite plantear un nuevo conjunto de usos para los mapas generados con esta tecnología: agricultura de precisión, control y gestión de catástrofes,... El vídeo generado tiene como característica principal la alta resolución, al ser

tomado a baja altura; pero como contrapartida presenta una gran deformación por la distancia y el ángulo desde el que se adquiere.

Debido a la gran deformación que sufren estas imágenes se requiere trabajar con ellas para la obtención de un mapa útil. Este proyecto trata sobre los diversos procesos necesarios para la obtención de estos mapas a partir de un vídeo. Los objetivos de este trabajo serán el estudio de las técnicas necesarias para la creación de estas ortofotos así como su evaluación. También se buscará determinar la correcta secuencia de utilización de las técnicas así como ofrecer estrategias en este proceso que mejoren los resultados.

Inicialmente se realizó un estudio de los diversos procesos necesarios para la realización de un mapa a partir de un vídeo. En este estudio se determinó que los pasos necesarios para la realización son: extracción de fotogramas del vídeo, detección de puntos característicos en las imágenes, emparejamiento de los puntos, estimación de la transformación y unión de las imágenes. Otro trabajo realizado ha consistido en la investigación de posibles lenguajes de programación con el que realizar el trabajo. Uno de los caminos abiertos consistía en usar la librería OpenCV que da soporte necesario para trabajar con dife-

-
- E-mail de contacto: ruben8668@gmail.com
 - Menció n realizada: Computació.
 - Trabajo tutorizado por: Felipe Lumbreras Ruiz (Ciències de la Computació)
 - Curso 2013/14

rentes lenguajes como C++ o Java permitiendo generar una solución fácil de distribuir y de usar. No obstante como se trata de un trabajo introductorio a este problema se decidió trabajar con Matlab por su comodidad de uso y facilidad para prototipar una solución.

2 ESTADO DEL ARTE

En el mercado actual existe una gran cantidad de empresas que obtienen beneficios a través de la creación de mapas, ortofotos y demás productos derivados, a partir de imágenes tomadas por un UAV. Estas empresas, como por ejemplo MosaicMill [1] y CATUAV [2], basan su negocio en ofrecer el servicio de captura y tratamiento para la obtención de los mapas. Los clientes pueden usar posteriormente los mapas proporcionados en diferentes áreas como la agricultura, conservación de la naturaleza, trazados de carreteras, minería, etc.

Si bien la mayoría del software es de pago, existen algunos gratuitos. Algunos de ellos no están pensados para la creación de mapas a partir de vídeos y aun así, pueden ser usados para obtener resultados preliminares. Unos ejemplos de estos son Hugin [3], PTGui [4] y AirPhotoSE [5] los cuales analizaremos a continuación.

Hugin es el primero de los programas que he utilizado, y puedo decir que tiene un uso complejo. Requiere la carga de imágenes individualmente. A posteriori la aplicación se encarga de unir las dependiendo de como parametrices el programa, lo cual es sumamente complicado para un usuario novel. Hace uso de diferentes ventanas para su control. Estas ventanas no son intuitivas con lo cual requiere dedicarle bastante tiempo. También su resultado, en gran medida por la dificultad de utilización y el no estar pensado para este fin directamente no es muy bueno.



Fig.1 – Resultado Hugin.

El segundo programa probado, PTGui, ha dado unos buenos resultados. Este programa se caracteriza por ser realmente muy intuitivo sin descartar las opciones de ajuste avanzado. Al igual que el anterior programa requiere de la carga individual de imágenes pero la disposición de los controles y su sencillez lo hace agradable al uso. En cuanto al resultado dado, siendo un programa para mosaico de panoramas, es bueno. En éste, a simple vista, únicamente se aprecia una discontinuidad en un camino de la zona boscosa (de la que se pueden obtener

menos puntos característicos).

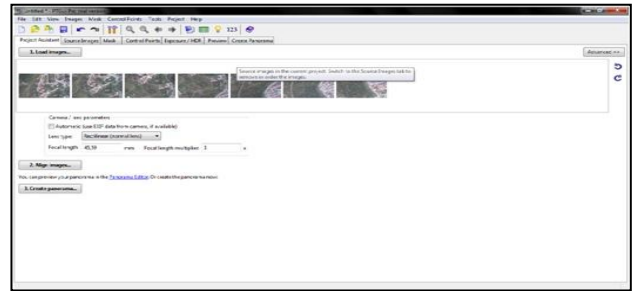


Fig.2 – Interfaz PTGui.



Fig.3 – Resultado PTGui

En los dos anteriores programas con su uso no se obtenía información sobre el proceso que era llevado a cabo por el programa para la realización de la ortofoto. En el último programa que analizaremos, AirPhotoSE, por el contrario y manteniendo la facilidad de uso si presenta pistas. Estas pistas se encuentran en el lado izquierdo de su interfaz que nos informa de las fases como la cantidad de puntos característicos que encuentra en cada imagen usando el algoritmo SIFT. También indica los procesos de emparejamiento de estos puntos así como el ajuste posterior de las imágenes para mejorar los resultados.

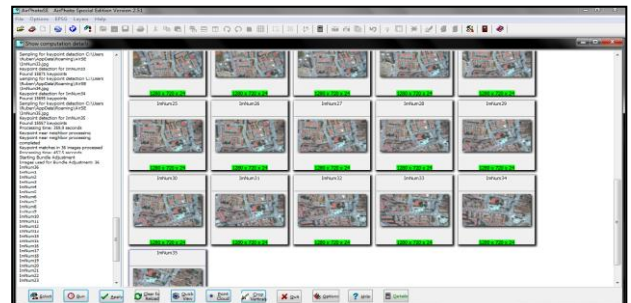


Fig.4 – Interfaz AirPhotoSE.



Fig.5 – Resultado AirPhotoSE.

3 METODOLOGÍA

Sin tener en cuenta la investigación inicial requerida para obtener el conocimiento sobre la materia necesario para determinar la dirección a tomar, el trabajo ha consistido de múltiples fases. Estas fases son mencionadas y explicadas a continuación:

3.1 Selección de zonas de interés

El material proporcionado consiste en un vídeo de 35min y 1 Gbyte aproximadamente, desde el despegue hasta el aterrizaje pasando por el sobrevuelo de diversas zonas. Los periodos de despegue y aterrizaje no son deseados. Asimismo un vídeo de 1 Gbyte se considera demasiado pesado para su trabajo por ello se procedió a la extracción de pequeños periodos de un minuto con los que trabajar.

Las zonas grabadas son tanto bosque y campos de cultivo como ciudad. Estas zonas presentan diferentes características del terreno que pueden afectar al proceso de detección de puntos y hay que tenerlo en consideración. Se decidió determinar 3 tipos de zona: bosque, ciudad y mixto. La zona de bosque se caracteriza por no contener edificios en los que un detector de esquinas se pudiese apoyar, el de ciudad resulta todo lo contrario y el mixto se caracteriza por tener zona de bosque y edificios en él. De estos vídeos se procedió a la extracción de fotogramas con los que iniciar las pruebas.



Fig.6 – Tipos de zona. Bosque, ciudad y mixto.

3.2 Algoritmos de detección de puntos característicos

La siguiente fase del proceso consiste en la obtención de puntos característicos. Existen diversos algoritmos que se encargan de ello y como la creación de uno propio no se contempla se procedió a seleccionar algunos de ellos. En la documentación inicial se encontraron menciones a algoritmos como SIFT ("Scale Invariant Feature Transform") [6] y SURF ("Speeded Up Robust Features") [7] por lo cual se decidió incluirlos desde un principio.

Matlab implementa algunos de estos de forma nativa en el Computer Vision System Toolbox el cual se ha obtenido. Este toolbox no dispone de una implementación del algoritmo SIFT por lo que se ha obtenido una de un tercero a través del mathworks [8]. El algoritmo SURF es in-

cluido en este toolbox junto con otros como MSER [9] y FAST [10] que han sido escogidos para su uso en un principio por presentar la implementación de forma nativa.

Tanto la implementación SIFT encontrada como SURF, MSER y FAST trabajan con imágenes en escala de grises por tanto para su uso hay que proceder de la siguiente forma: extracción de imágenes del vídeo, copia de la imagen en escala de grises, llamada a los algoritmos proporcionándole la copia de la imagen y guardado de los puntos encontrados en una lista.

3.3 Extracción de características y emparejamiento

La siguiente fase usa los puntos obtenidos a través de los diferentes algoritmos. Consiste en la extracción de características de los puntos obtenidos de dos métodos distintos. El primero y más sencillo usado con los algoritmos SIFT y FAST en los que únicamente al aplicar el algoritmo detector de puntos se obtiene una matriz de dos columnas (posición x e y) y tantas filas como puntos detectados. El método asociado para extraer características consiste en extraer los valores de intensidad de los píxeles colindantes al punto en forma de bloque. El segundo, más complicado, retorna valores porcentuales de los píxeles en forma de esfera teniendo en cuenta la orientación para así mejorar la descripción en cualquier posible posición.

Después de la descripción continúa el emparejamiento. En este punto se podría haber realizado un simple método que calcule una matriz de distancia euclidiana entre las características de los puntos y asociarlos conforme a la menor distancia. Matlab implementa este cálculo para asociar puntos y otros métodos. En concreto se ha decidido usar uno de los que implementa que consiste en, aparte de calcular la distancia entre los puntos, elimina posibles puntos que son ambiguos cuando su distancia entre el más cercano y los siguientes no supera un ratio.

3.4 Obtención de la transformación

Seguidamente al emparejamiento de los puntos característicos proseguiría el cálculo de la transformación con esos puntos en caso de que todos los emparejamientos fuesen correctos. Con esto hago referencia a la existencia de errores en la asociación de estos puntos que pueden afectar a la calidad de la transformación. Para eliminar estos "outliers" se podría haber creado un método propio el cual elimine las asociaciones de puntos más diferentes hasta el cumplimiento de un cierto criterio. Ya existen algoritmos que realizan este proceso por lo cual se decidió usar el método RANSAC ya existente [11], [12]. Matlab proporciona una implementación parametrizable de este con el proceso de cálculo de transformación la cual se decidió dejar por defecto al obtener buenos resultados con esa combinación.

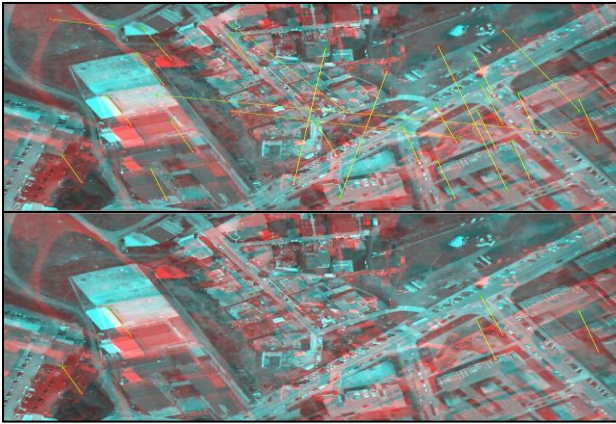


Fig.7 - Emparejamiento y uso del RANSAC.

A continuación esta la selección del tipo de transformación a aplicar. Teniendo dos opciones posibles, afin y proyectiva, se realizó pruebas con las dos. La transformación afin consiste en una matriz $M_{2 \times 3}$ que contiene: traslación, escalado, rotación e inclinación. Fue considerada en un principio pero obtuvo resultados no aceptables y fue descartada inmediatamente. La transformación proyectiva consiste en una matriz $M_{3 \times 3}$ con el siguiente funcionamiento: $salida[x', y', w'] = M_{3 \times 3} \cdot p[x, y, 1]$ donde p es el vector que contiene la posición 2D en la imagen mas un 1, y $resultado[x'', y''] = [x'/w', y'/w']$ es la posición 2D en el plano de salida. A diferencia de una transformación afin, la proyectiva no mantiene el paralelismo de las líneas de la imagen resultando ser más adecuada. Los resultados obtenidos aplicando este tipo de transformación, obtenida con los puntos de una imagen a otra, y uniéndolos formando el inicio de un mapa si bien no son perfectos sí son muy aceptables.



Fig.8 - Transformacion proyectiva.

3.5 Análisis de los algoritmos

A continuación se procedió a evaluar los diferentes algoritmos de detección de características. La evaluación consiste en contabilizar el tiempo y cerciorarse de obtener un mínimo de puntos después del RANSAC para diferentes pares de imágenes. La decisión de probarlo en la zona mixta es que su comportamiento supone un reflejo de las zonas de bosque y ciudad a la par. Los detectores probados han sido: SIFT [13], FAST [14], SURF [15] y MSER [16]. Junto a sus tiempos de ejecución se procedió a obtener una visión de la disposición de todos los puntos en las

imágenes.

Hay que mencionar que se descarto el uso del detector SIFT en un principio y por ello no aparecen resultados de su ejecución. La causa es la implementación del SIFT. Ésta era muy costosa en tiempo de ejecución así como el número de puntos detectados era demasiado grande y sin un método eficaz para reducir estos. Otro punto importante es que se ha definido que el máximo de puntos a obtener al usar FAST sean 1000 para reducir la carga ya que encuentra varios miles en cada imagen y se ha aumentado el número de descriptores que usa a 284 a diferencia de MSER y SURF que usan 64 únicamente.

Con los resultados se determinó que el algoritmo FAST ha de ser descartado debido a su ineficiencia en la detección de puntos en las zonas de bosque y el tiempo necesario para su finalización. Los otros dos algoritmos, SURF y MSER, han demostrado ser capaces de encontrar suficientes puntos en ambos tipos de zonas. Los dos son determinados como aptos para ser usados como el detector final pero en este trabajo se decide continuar a partir de este momento con el detector MSER. La razón de escoger MSER se basa en que aún detectando menos puntos en la imagen su relación de puntos en zona bosque/zona ciudad está más equilibrada.

Imag	Pts	Descr.	Match	RANSAC	Tiempo
Im 1-3	1000	284	208	120	5.24s
Im 3-5	1000	284	324	204	5.29s
Im 5-7	1000	284	416	299	5.31s
Im 7-9	1000	284	517	345	5.32s
Im 9-11	1000	284	486	374	5.34s
Im 11-13	1000	284	453	348	5.33s
Im 13-15	1000	284	463	290	5.30s
Im 15-17	1000	284	305	232	5.31s
Im 17-19	1000	284	494	318	5.31s
Im 19-21	1000	284	335	203	5.24s

Tabla 1 - FAST: Lectura, detección, extracción, emparejamiento, RANSAC, transformación (proyectiva)

Imag	Pts	Descr.	Match	RANSAC	Tiempo
Im 1-3	951	64	125	85	1.05s
Im 3-5	1141	64	259	182	1.25s
Im 5-7	1499	64	385	291	1.54s
Im 7-9	1365	64	507	329	1.69s
Im 9-11	1403	64	449	377	1.65s
Im 11-13	1288	64	400	324	1.58s
Im 13-15	1268	64	420	290	1.51s
Im 15-17	1093	64	308	294	1.32s
Im 17-19	1044	64	376	273	1.08s
Im 19-21	916	64	274	200	0.99s

Tabla 2 - MSER: Lectura, detección, extracción, emparejamiento, RANSAC, transformación (proyectiva)

Imag	Pts	Descr.	Match	RANSAC	Tiempo
Im 1-3	1588	64	363	258	1.16s
Im 3-5	2120	64	801	553	1.48s
Im 5-7	2714	64	1136	924	2.01s

Im 7-9	2593	64	1468	1045	2.29s
Im 9-11	2578	64	1364	1207	2.23s
Im 11-13	2112	64	1155	961	1.96s
Im 13-15	2146	64	1161	832	1.73s
Im 15-17	1628	64	659	571	1.43s
Im 17-19	1585	64	926	693	1.20s
Im 19-21	1450	64	586	414	1.02s

Tabla 3 – SURF: Lectura, detección, extracción, emparejamiento, RANSAC, transformación (proyectiva)

3.6 Lectura, base y ciclo

Hasta este momento el trabajo se ha realizado manualmente cargando los fotogramas a las que se le aplicaba todos los pasos necesarios para la obtención de la transformación entre ellos. Este proceso se desea realizar únicamente especificando un vídeo. En un principio se buscó la forma de guardar los fotogramas deseados en una lista con la que trabajar posteriormente. Debido a su consumo elevado de memoria se descartó. El método seleccionado consiste en la carga del vídeo y el posterior acceso a los fotogramas. Con Matlab se requiere únicamente de dos funciones para actualizar la imagen con la que trabajar: “video = VideoReader('ruta')” para cargar el video y “imagen = read(video,indice)” para extraer el fotograma. Estas funciones, mediante el uso de librerías, tienen equivalentes en otros lenguajes de programación. La variable “índice” indica la posición a extraer. Esta variable se va incrementando en función otra variable “lapso” dentro de un bucle.

Otro punto a tener presente es lo que llamamos base. Ésta hace referencia a la imagen que contiene la ortofoto y sobre la que se ponen los fotogramas transformados para formar el mapa completo. Los métodos de Matlab están pensados para superponer los fotogramas transformados en una base de tamaño fijo. Este método tiene el inconveniente de requerir posicionar el primer fotograma sobre el que trabajas en el centro de la base y definir esta base con un gran tamaño. Es necesario un gran tamaño porque a priori no conoces la ruta del avión y la posición de los fotogramas podrían corresponder a una zona fuera de la base. Para solucionar este inconveniente se ha creado un método que permite definir la base como el primer fotograma seleccionado. Una vez obtenida la transformación del fotograma el método recibe: la base, el fotograma, la transformación y adelantando mejoras un desplazamiento y una matriz de ponderación. Sin tener en cuenta los dos últimos parámetros el método realiza los siguientes pasos:

- Calcular las posiciones de los extremos del fotograma en el mapa aplicando la transformación.
- Analizar que las posiciones correspondan al interior de la base.
- En caso negativo calcular la ampliación necesaria a la base por los cuatro lados y la posición correspondiente a la base en el nuevo mapa.

- Poner la base en el nuevo mapa y seguidamente el fotograma con la transformación aplicada.

Con estos dos últimos puntos los bloques del programa ya pueden ser definidos. Si bien, con las futuras mejoras, varía la forma de realizar algunos puntos la estructura se mantiene. El primer bloque del programa consiste en la carga del vídeo, preparación de las variables y de la base. El segundo bloque consiste en el bucle para la creación del mapa que consiste en los siguientes pasos:

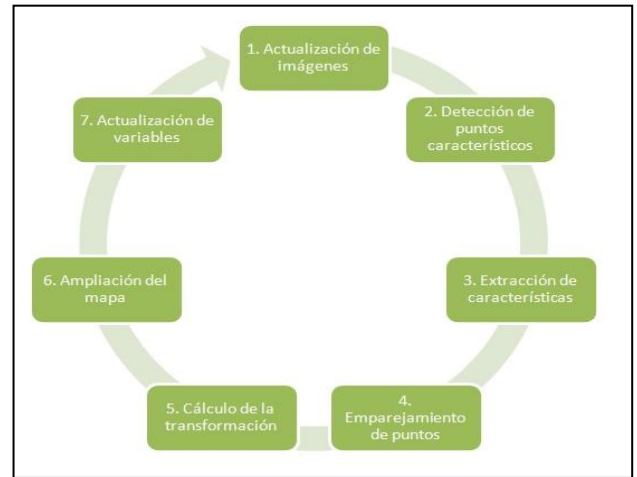


Grafico 1 – Ciclo del programa.

3.7 Modo de trabajo

En esta fase se determina el modo en que la transformación se calcula. Con los métodos mencionados anteriormente podemos obtener: un gran número de puntos, la descripción de estos, su emparejamiento y la transformación necesaria para ponerlos en correspondencia. Esto, hasta este momento, tiene la inconveniencia de tener que hacerse fotograma a fotograma. Pero es necesario tener la transformación respecto a un plano común. En nuestro caso decidimos que sea el primer fotograma.

Escojido el plano de referencia hemos de determinar como obtener todas las transformaciones respecto a este plano. La primera opción que se nos presenta es, como hasta el momento, obtener la transformación entre fotogramas que llamaremos $T_{i,j}$. $T_{i,j}$, si recordamos bien, es una matriz de 3×3 que podemos obtener para cada par de fotogramas. En caso de desear la matriz que describa la transformación correspondiente al fotograma “j” con respecto al “i” puede ser obtenida mediante la multiplicación de las matrices intermedias. Ejemplo: para obtener la transformación del tercer fotograma respecto al primero es: $T_{1-3} = T_{1-2} \times T_{2-3}$. Este proceso requiere el almacenaje de la $T_{i,j}$, donde “j” corresponde al último fotograma e “i” el primero. No obstante este método tiene un fallo. Las transformaciones calculadas entre fotogramas tienen un pequeño error, este error entre fotogramas es prácticamente imperceptible si unes únicamente dos o tres fotogramas. Asimismo este error

crece y se hace intolerable a los 10 fotogramas o más.



Fig.10 - Resultado transformación acumulativa.

El método final por el que se ha optado consiste en crear pequeñas ventanas que centren la última zona insertada. La $T_{i,j}$ obtenida está en relación a esta ventana por lo cual se evita la acumulación de error por sucesivas multiplicaciones. En el momento de unir los fotogramas con el mapa si hasta el momento se calculaba como " $\text{posición mapa} = \text{posición fotograma} \times T_{i,j}$ " en adelante se calculará como " $\text{posición mapa} = \text{posición fotograma} \times T_{i,j} + \text{desplazamiento ventana}$ ".

3.8 Métodos de unión

Con este paso se ha buscado solucionar los pequeños errores que se encuentran en diversas zonas correspondientes al borde de un fotograma en el mapa. Este error consiste en discontinuidades a lo largo de una carretera o en las paredes de un edificio. La causa de esta discontinuidad en el momento en que un fotograma es unido con el mapa general es debido a la transformación. Esta transformación es calculada, en su mayoría, con puntos centrales a la imagen. La transformación, aún debiendo ser igual en toda la superficie del fotograma, resulta no corresponder bien con los bordes mientras que las zonas centrales sí. Para intentar que el programa tenga en cuenta este hecho se ha buscado las siguientes estrategias en el pegado:

- Común: Todas las estrategias tienen en común la necesidad de crear una matriz que represente todos los píxeles del fotograma y del mapa. Los valores de cada celda de las diferentes matrices varían según la estrategia elegida. Cada celda corresponde a un píxel de la correspondiente imagen. En el momento de unir el mapa con el fotograma transformado, en caso de ocupar la misma posición, la elección de si se superponen los píxeles es relativa a los valores de las celdas que le corresponden.
- Estrategia 1: Consiste en definir los valores de las celdas en función de la distancia euclidiana al centro del fotograma. La condición que determina si una zona es añadida o no al mapa, en caso de que corresponda con una zona ya asignada, es el

valor mínimo. Esta estrategia suaviza el cambio causado por la inclusión de un nuevo fotograma en el mapa y los bordes tengan menor prioridad frente al centro del enfoque. Otro efecto positivo conseguido es, en caso de que en un determinado momento, un nuevo fotograma sea añadido con una transformación incorrecta, los resultados futuros se vean afectados mínimamente por esto. Aún obteniendo esta inesperada mejora el fotograma con la mala transformación continúa apareciendo, en este caso su centro.

- Estrategia 2: Muy similar a la anterior únicamente cambiando el valor de las celdas de las matrices. En esta estrategia el valor que se le asigna a cada celda es una distancia respecto a un punto al que se le da el nombre de centro G. Este centro G es obtenido como: suma de los puntos válidos con los que se obtiene la transformación dividido entre la cantidad de ellos. Esta estrategia requiere recalcular los valores de las celdas del mapa correspondientes a la ventana del mapa usada y al fotograma a unir. Los resultados obtenidos han mejorado respecto, al anterior, en el efecto visual que provoca un fotograma con una mala transformación.
- Estrategia 3: Esta estrategia define los valores de cada celda con un valor de densidad relativa. Este valor utiliza los puntos usados para el cálculo de la transformación. La distancia entre la posición actual y cada uno de los puntos divide un valor y el resultado de la suma de estas divisiones es el valor de cada celda. Al igual que con la anterior estrategia los valores de las celdas del mapa correspondientes a la ventana del mapa usada tienen que actualizarse. La elección de la zona que permanecerá después de la unión se determina en función del valor más grande en lugar del más pequeño como anteriormente se realizaba. Los resultados han mostrado más errores visuales que los dos anteriores al debido al hecho de que puedan aparecer múltiples zonas del nuevo fotograma a lo largo del mapa. Estas zonas no todas tienen un cambio correcto y pueden aparecer zonas repetidas en el mapa.

La estrategia 1 y 2 han mostrado ser efectivas mientras que la 3 provee peores resultados que con la superposición del nuevo fotograma en el mapa. Se determina que es aconsejable la inclusión de zonas continuas y grandes del nuevo fotograma en el mapa para reducir el error visual. En función del cálculo que se esté dispuesto a realizar se puede decantar por superponer la nueva imagen o la 1 y 2 (mayor cálculo).

3.9 Soporte con mapa

Esta última fase es necesaria a causa de los resultados obtenidos. Los resultados obtenidos han sido creados todos en función de un primer fotograma dándole a este la presunción de correcta visualización. Esto es erróneo.

El primer fotograma también está tomado con perspectiva y por tanto no dispone de una vista de pájaro y sin deformaciones de la zona. A causa de esto, el mapa que se construye mantiene la perspectiva del primer fotograma lo cual provoca una gran deformación. Esta deformación provoca que el mapa no pueda ser usado para ser visto en una escala real. Las opciones encontradas para subsanar este hecho requieren del apoyo de un mapa a escala como puede ser un callejero y la entrada de puntos a mano.

La primera opción consiste en usando el callejero y el mapa creado hasta el momento. Se obtienen puntos en común del mapa y el callejero para obtener una transformación proyectiva y se aplica. El proceso inicial provoca que conforme te alejas del inicio pierdas información a causa de reducir la zona en que la zona ha de establecerse o que no puede ser obtenida a causa de que la zona asignada para un fotograma tiene más superficie que el mismo fotograma. También, esto visto a través de la estimación del error en el apartado de resultados de este documento, se aprecia que el error de crearse el mapa sin consideras la perspectiva y luego sea corregida acumula bastante error.

Dado que con la opción anterior no se obtienen unos resultados muy aceptables se decidió buscar nuevas vías de mejora. La vía hallada consiste en aplicar sobre el primer fotograma el proceso anterior. Esta acción ocasiona que el primer fotograma sea dispuesto con la correcta perspectiva permitiendo trabajar a partir de él sin estar creando el mapa en perspectiva. Este proceso requiere de dos posibles vías. La primera presenta la necesidad de preparar el mapa en el que se encuentra el primer fotograma con la perspectiva correcta en el y cargar este mapa específicamente dentro del proceso. La segunda vía es disponer del proceso de transformación dentro del proceso principal. Asimismo preparar el mapa puede necesitar de una supervisión por diversos motivos que es conveniente separar del proceso de construcción.

La supervisión y adaptación del mapa es causada por diversos requisitos que deben ser cumplidos. El primer requisito consiste en que la escala del callejero usado tiene una diferencia de tamaño considerable puede provocar dos problemas. El primero si la escala es muy grande que el fotograma se reduzca demasiado en el mapa para poder ser comparado con los siguientes fotogramas. El segundo es, si esta escala es muy pequeña, el redimensionamiento del fotograma en el mapa deja píxeles por toda la zona ocupada sin un valor de correspondencia creando puntos y líneas negras que confundirán al detector de puntos. Otro requisito que debe cumplirse es que el mapa resultante tenga una resolución mínima a causa de la necesidad de preparar ventanas con la misma resolución del vídeo.

4 RESULTADOS

A continuación se puede ver el resultado obtenido en

en la zona de ciudad con todas las mejoras aplicadas. Estos mapas tienen pequeños fallos, en especial en las zonas que corresponden a los bordes de los fotogramas. Estos errores consisten en líneas oscuras que delimitan la superficie máxima de los fotogramas. La causa de este pequeño defecto se debe a que la cámara ensombrece los píxeles correspondientes al contorno del enfoque. Esto se ha decidido ignorar por no suponer un gran problema y tener como solución la no utilización de estos píxeles. Por lo que respecta al error visual provocado por un mal ajuste entre los cambios de fotogramas se puede determinar que es prácticamente nulo.



Fig.11 – Resultado optimo.

El último error que puede ser tenido en consideración es el error que represente la relación entre las distancias reales y las que son mostradas en el mapa. Para evaluar el error en nuestro mapa primero es necesario tener un mapa con el que comparar. Este mapa, el cual se le llamara MapaSatelite de ahora en adelante, ha sido obtenido a partir de la unión de capturas de pantallas de un SIG fiable. Las capturas, en nuestro caso, han sido tomadas del portal Web del IGN-CNIG SIGNA [17].

4.1 Método de estimación de error

Con el fin de evaluar los diferentes resultados obtenidos se barajaron diversas métricas con las que determinar el error cometido. Se terminó decantándose por un método que simplizase la deriba que han sufrido los diversos puntos de la imagen. La deriba consiste en el desplazamiento de las zonas respecto a su entorno. Un ejemplo son dos edificios a una distancia X en la realidad los cuales en el imagen aérea construida tienen una distancia Y diferente a la real.

Si bien la deriba sufrida en los puntos, que representan las zonas, suele calcularse en una medida de distancia como metros no es posible en nuestro caso. No obstante podemos usar los píxeles entre dos puntos como medida de error obteniendo un resultado igual. Para estimar este error se ha creado el siguiente método que consiste en:

1. Mostrar el MapaSatelite y uno de los resultados obtenidos.
2. Clicar manualmente los mismos puntos en ambas imágenes.
3. Estimar una transformación afín en función de los puntos introducidos.

4. Calcular nuevos puntos del resultado aplicando la transformación.
5. Calcular la distancia entre los nuevos puntos y los del MapaSatelite.
6. Dar la media de esas distancias.

Como punto aclaratorio en método de estimación de error, aplicamos una transformación. Esta transformación aunque en un principio puede considerarse que afecta a la estimación real del error no es así. Esta transformación consiste en rotación, traslación y escalado y nos pone en correspondencia los dos mapas independientemente de la escala y posición de estos concentrándose únicamente en las distancias que se encuentren entre los puntos.

4.2 Estimación del error

Con el fin de estimar el error se han definido 5 categorías de comparación de imágenes. La categoría 1 se compara el MapaSatelite con el mismo para estimar el error por la introducción de puntos a mano. La categoría 2 compara el resultado proporcionado por AirPhotoSE con el MapaSatelite. La categoría 3 compara el resultado obtenido mediante la aplicación de un rectificado manual de perspectiva al primer frame con el MapaSatelite. La categoría 4 compara el resultado obtenido mediante la aplicación de un rectificado manual de la perspectiva al mapa creado por defecto con el MapaSatelite. La categoría 5 compara un resultado por defecto con el MapaSatelite.

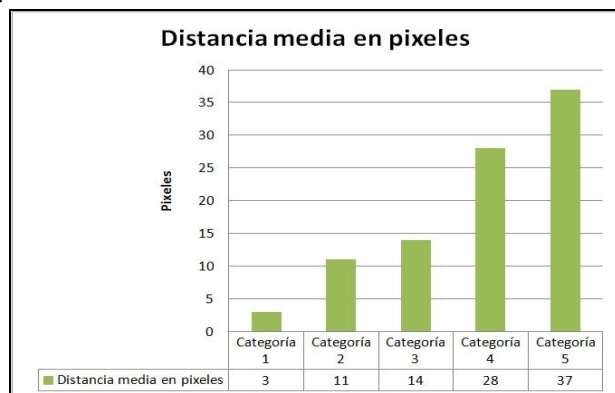


Grafico 2 – Error en contraste con SIGNA.

Extraídos los resultados se puede determinar que a través de nuestro método utilizado con soporte en el primer fotograma obtiene unos resultados muy similares al software comercial. Esto determina que nuestro método obtiene unos resultados decentes con un apoyo manual inicial mínimo. No obstante el método creado tiene la limitación actual de un número reducido de frames capaces de ser procesados.

5 CONCLUSIONES

Llegados al final de la realización de este trabajo de fin de grado se ha obtenido conocimiento sobre las técnicas necesarias para la creación de ortofotos. Como objetivo principal era determinar el proceso para obtención de

mapas. Este proceso consiste en:

1. Carga del vídeo y preparación de variables (entre el que se incluye soporte con un mapa a escala fiable)
2. Obtención del fotograma correspondiente y preparación de una ventana del mapa.
3. Extracción de puntos característicos del fotograma y la ventana.
4. Extracción de descriptores y emparejamiento de puntos.
5. Obtención de la transformación proyectiva.
6. Unión del fotograma con el mapa.
7. Actualización de variables y repetición del bucle (punto 2).

A partir de este conocimiento obtenido por la experiencia se pueden extraer diversas conclusiones.

La primera conclusión que obtenemos es en relación con los algoritmos detectores de puntos, determina que los resultados obtenidos con estos son suficientemente correctos para la obtención de buenos resultados. Esto implica el hecho de crear nuevos algoritmos que intenten mejorarlos resultaría un sobreesfuerzo que posiblemente no se vea compensado. Asimismo entre los ya existentes tanto MSER como SURF proporcionan los resultados necesarios y son dos opciones válidas. La siguiente conclusión hace referencia a los descriptores de los puntos y determina que describir los puntos con estrategia de valor porcentual de los pixeles en forma esférica es la indicada para el correcto emparejamiento.

En relación al método de unión de imágenes se concluye que realizarla en función a una distancia con un punto enmascara posibles errores en las transformaciones, y siendo este punto el central del fotograma o el "centro de masas" opciones igualmente válidas. No obstante aunque todos estos métodos mejoran los resultados la acción que mejora en más medida y aporta mejores resultados es el la transformación del primer fotograma con que se trabaja usando un mapa a escala fiable.

Otra conclusión extraída hace referencia al uso de Matlab ha demostrado mucho más cómodo que C++ al implementar con el entorno funciones requeridas para el trabajo mientras que en el caso del lenguaje C++ se requiere librerías externas como OpenCV. Estas librerías están respaldadas por una comunidad pero, a diferencia de la de Matlab, el soporte que ofrecen es de baja calidad dificultando en gran medida su uso y comprensión. Para finalizar se determina que hasta el momento se ha trabajado con pequeños volúmenes de estructuras y que en caso de ser necesario un mayor tamaño sí es aconsejable el cambio al uso de lenguajes como C++ o Java con el soporte de la librería OpenCV o otras.

6 TRABAJO FUTURO

Como líneas de trabajo futuro podemos citar el trabajar con un lenguaje más potente como C++ o Java los cuales permitirían la obtención de programas prototipo para un uso serio que permitiría el trabajo con zonas mayores. Respecto a la mejora de los resultados cabe destacar que no son perfectos y por ello podrían tomarse diferentes vías para su mejora. La primera vía consiste en una mejora del trabajo con soporte de un mapa base cada cierto tiempo con tal de eliminar el error que se acumula con el paso de los fotogramas. Otra vía consistiría en disponer de niveles de resolución del mapa con tal de evitar los problemas de pixeles oscuros que confunden los detectores. Otra tercera vía consistirá en el flujo sin necesidad de un soporte de un mapa siendo esta mejora la culminación del proceso.

AGRADECIMIENTOS

Este proyecto de fin de grado a resultado ser más arduo de lo en un principio se tenía pensado que sería un trabajo de fin de grado. El proyecto también ha sufrido de dilataciones en el trabajo necesario para la realización de diversas partes provocando un aumento en el trabajo requerido. Por aligerar el peso quiero mostrar mi agradecimiento a:

Felipe Lumbreras como tutor del proyecto por sus consejos, en especial relacionados con los informes y su mejora antes de las entregas definitivas.

A los foros y comunidades que proporcionan la posibilidad de buscar dudas y a los usuarios de estos que dan respuestas.

Y también a los académicos que han puesto sus conocimientos relacionados con mi proyecto en la red de manera gratuita.

BIBLIOGRAFÍA

- [1] MosaicMill. Ensomosaic, <http://mosaicmill.com/>, Octubre 2013.
- [2] CATUAV. Catuav, <http://www.catuav.com/es/>, Enero 2014.
- [3] Hugin. Hugin - panorama photo stitcher, <http://hugin.sourceforge.net/>, Octubre 2013.
- [4] PTgui. Create high quality panoramic images, <http://www.ptgui.com/>, Octubre 2013.
- [5] BASP. Airphoto special edition - a program for the geometric rectification of aerial images orthophotos from multiple images, <http://www.uni-koeln.de/~al001/airphotose.html>, Enero 2014.
- [6] Pablo Flores, Juan Braun. Algoritmo SIFT: fundamento teórico. <http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/FundamentoSIFT.pdf>. Agosto 2011.
- [7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf:

- Speeded up robust features. In *In ECCV*, pages 404–417, 2006.
- [8] Cheggaju Naveen. Sift (scale invariant feature transform) algorithm, <http://www.mathworks.com/matlabcentral/fileexchange/43723-sift-scale-invariant-feature-transform-algorithm>, Octubre 2013.
- [9] Maximally Stable Extremal, J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from. In *In British Machine Vision Conference*, pages 384–393, 2002.
- [10] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *In European Conference on Computer Vision*, pages 430–443, 2006.
- [11] Juan Braun, Pablo Flores. Algoritmo ransac: fundamento teórico. <http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2011/keypoints/FundamentoRANSAC.pdf>. Agosto 2009.
- [12] P.V.S.S Prakash P.Saravanan, Narayanan .C.K. and Prabhakara Rao.G.V. Techniques for video mosaicing. 2007.
- [13] Wikipedia. Scale-invariant feature transform, Octubre 2013.
- [14] Wikipedia. Features from accelerated segment test, Noviembre 2013.
- [15] Wikipedia. SURF, Octubre 2013.
- [16] Wikipedia. Maximally stable extremal regions, Noviembre 2013.
- [17] IGN-CNIG. Sina, sistema de información geográfica nacional de España, <http://signa.ign.es/signa/>, Octubre 2013.