

Automatic Scene Reconstruction Algorithm for Planialtimetric Applications

Marcos Gomez-Redondo, Hector Fretes, José Rodríguez-Piñeiro, Jorge Rodas, and Raul Gregor

Resumen—The development of digital photogrammetry solutions applied to aerial images has gained interest in recent years due to its diverse areas of application, besides the decreasing production costs of high-resolution cameras and unmanned aerial vehicles. The implementation of an automatic scene reconstruction algorithm by using a pair of aerial images as input is proposed in this paper, as a first approach to the full reconstruction algorithm with multiple images. The current work can be used for educational purposes as it contains the general method of reconstruction, excepting for the densification process, which is tested in this paper. The reconstruction algorithm is tested with pairs of aerial pictures taken from a controlled scene, in order to analyze its viability and then with aerial pictures taken with a drone.

Keywords—Digital Photogrammetry, 3D Reconstruction, Epipolar geometry, Point Cloud, Essential matrix, Aerial Photogrammetry

I. INTRODUCCIÓN

EN las últimas décadas, debido a su disponibilidad comercial y precios relativamente bajos, los vehículos aéreos no tripulados o drones han ganado relevancia en diversas aplicaciones tales como la agricultura de precisión, vigilancia, transporte de medicinas a lugares remotos o de difícil acceso, entre otras [1]-[3]. Una de las principales aplicaciones es la toma de fotografías aéreas, dado que los drones permiten un barrido rápido, preciso y económico de una superficie en comparación con métodos terrestres, satelitales y aéreos tripulados [4]-[6]. La aplicación de técnicas de fotogrametría digital permite expandir drásticamente la utilidad de las fotografías aéreas en distintos campos [7]. Resulta de particular interés la obtención de modelos tridimensionales debido a la gran cantidad de información que puede obtenerse de estos modelos y su diversidad de aplicaciones, como cálculo de curvas de nivel o medición de áreas y volúmenes, entre otros.

La creciente capacidad de procesamiento de los computadores, la constante reducción de sus costos de producción y las ventajas propias del procesamiento digital han despertado interés por lograr la reconstrucción automática de escenarios. Este proceso de reconstrucción debe contemplar tres etapas. La primera etapa es la detección, identificación y pareo de puntos de interés utilizando descriptores como BRISK [8]. La segunda es el cómputo de la geometría, como por ejemplo

M. Gomez-Redondo, H. Fretes, J. Rodas and R. Gregor are with the Laboratory of Power and Control Systems, Facultad de Ingeniería, Universidad Nacional de Asunción, Luque 2060, Paraguay (e-mail: gomez-redondomarcos@gmail.com; hectorfretesac@gmail.com; jrodas@ing.una.py; rgregor@ing.una.py).

J. Rodríguez-Piñeiro is with the College of Electronics and Information Engineering, Tongji University, China. (e-mail: j.rpineiro@tongji.edu.cn).

el algoritmo de los cinco puntos de [9], para obtener la denominada matriz esencial, que permitirá hallar la posición relativa entre cámaras. Por último, la triangulación de los puntos mediante métodos como el de triangulación lineal [10], utilizado en el presente trabajo.

En este trabajo se implementa un algoritmo de reconstrucción de escenarios tridimensionales a partir de un par de imágenes aéreas. Para ello se supone que se posee una cámara calibrada, es decir, se conocen todos los parámetros internos de la cámara con la cual se obtuvieron las imágenes. El algoritmo propuesto se encarga de realizar el pareo de puntos homólogos, determinar la geometría del escenario y, finalmente, de realizar el cómputo de la nube de puntos correspondientes, para un par de imágenes cualesquiera. Se propone además un algoritmo de densificación de la nube obtenida que requiere, además de los requisitos previos, que las imágenes sean capturadas a una misma altura respecto a un nivel de referencia previamente determinado y que exista una rotación despreciable de la cámara entre ambas tomas.

II. HERRAMIENTAS MATEMÁTICAS

En lo sucesivo, se empleará la geometría proyectiva. En la presente sección se introduce la base matemática necesaria en el ámbito de este artículo.

II-A. Coordenadas Homogéneas

La representación homogénea de la recta $ax + by + c = 0$ de \mathbb{R}^2 es $[a, b, c]^T \in \mathbb{P}^2$, y es equivalente a $[ka, kb, kc]^T$, con $k \in \mathbb{R}$, ya que:

$$kax + kby + kc = k(ax + by + c) = 0. \quad (1)$$

La representación del punto $[x, y] \in \mathbb{R}^2$ es, de este modo, $[x, y, 1]^T \in \mathbb{P}^2$, y es equivalente a $[kx, ky, k]^T$ por la misma razón. De esta forma, la relación de incidencia entre un punto y una recta puede escribirse mediante el producto escalar de estos vectores, como:

$$\mathbf{r}^T \mathbf{x} = \mathbf{x}^T \mathbf{r} = 0, \quad (2)$$

donde \mathbf{r} es la recta $[a, b, c]^T$ y \mathbf{x} el punto $[x, y, 1]^T$, es decir, el punto $[x, y] \in \mathbb{R}^2$ en notación homogénea.

La intersección \mathbf{x} de dos rectas \mathbf{r}_1 y \mathbf{r}_2 puede obtenerse entonces como el producto vectorial:

$$\mathbf{x} = \mathbf{r}_1 \times \mathbf{r}_2, \quad (3)$$

ya que $\mathbf{r}_1^T(\mathbf{r}_1 \times \mathbf{r}_2) = \mathbf{r}_2^T(\mathbf{r}_1 \times \mathbf{r}_2) = 0$ por ser $\mathbf{r}_1 \times \mathbf{r}_2$ un vector perpendicular a \mathbf{r}_1 y a \mathbf{r}_2 . Por la dualidad (2) se

obtiene la recta \mathbf{r} que pasa por dos puntos \mathbf{x}_1 y \mathbf{x}_2 como $\mathbf{r} = \mathbf{x}_1 \times \mathbf{x}_2$. Estas relaciones entre puntos y rectas son las operaciones básicas utilizadas en el trabajo. A continuación se presentan las transformaciones proyectivas.

II-B. Transformaciones Proyectivas

Un mapeo $h : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ es una proyectividad si y sólo si existe una matriz no singular H de dimensiones 3×3 tal que para cualquier punto en \mathbb{P}^2 representado por un vector \mathbf{x} se cumple que $h(\mathbf{x}) = H\mathbf{x}$.

De esta forma todas las rectas se transforman según $h(\mathbf{r}) = H^{-T}\mathbf{r}$, así la relación de incidencia de puntos se mantiene, de acuerdo a

$$\mathbf{r}^T \mathbf{x} = \mathbf{r}^T H^{-1} H \mathbf{x} = 0, \quad (4)$$

y se mantiene consecuentemente la colinealidad.

Según la forma de la matriz H la transformación proyectiva puede clasificarse en cuatro jerarquías [10]: la primera es para isometrías, donde se preserva la norma euclidiana; la segunda es para similitudes, si la isometría posee una escala isotrópica. La tercera es la transformación afín, que puede verse como un escalado no isotrópico. Finalmente, la cuarta es la transformación proyectiva genérica, que es el caso más general de transformación, en donde se cambia la perspectiva y los puntos del infinito pueden mapearse a puntos finitos.

Es posible determinar H dado un conjunto de cuatro o más correspondencias de puntos de 2D a 2D, $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. La transformación dada por la ecuación $\mathbf{x}'_i = H\mathbf{x}_i$ puede ser expresada en términos de un producto vectorial. Si la j -ésima fila de la matriz H se denota como \mathbf{h}^{jT} , se puede escribir:

$$H\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1T} \mathbf{x}_i \\ \mathbf{h}^{2T} \mathbf{x}_i \\ \mathbf{h}^{3T} \mathbf{x}_i \end{pmatrix}. \quad (5)$$

Escribiendo $\mathbf{x}_i = (x'_i, y'_i, w'_i)^T$, el producto vectorial puede expresarse como

$$\mathbf{x}'_i \times H\mathbf{x}_i = \begin{pmatrix} y'_i \mathbf{h}^{3T} \mathbf{x}_i - w'_i \mathbf{h}^{2T} \mathbf{x}_i \\ w'_i \mathbf{h}^{1T} \mathbf{x}_i - x'_i \mathbf{h}^{3T} \mathbf{x}_i \\ x'_i \mathbf{h}^{2T} \mathbf{x}_i - y'_i \mathbf{h}^{1T} \mathbf{x}_i \end{pmatrix}. \quad (6)$$

Así, considerando $\mathbf{h}^{jT} \mathbf{x}_i = \mathbf{x}_i^T \mathbf{h}^j$ para $j = 1, 2, 3$, se pueden obtener las componentes de H como

$$\begin{bmatrix} \mathbf{0}^T & -w'_i \mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ w'_i \mathbf{x}_i^T & \mathbf{0}^T & -x'_i \mathbf{x}_i^T \\ -y'_i \mathbf{x}_i^T & x'_i \mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}. \quad (7)$$

Estas ecuaciones tienen la forma $A_i \mathbf{h} = \mathbf{0}$, donde A_i es una matriz de 3×9 , y \mathbf{h} es un vector columna de 9 elementos

$$\mathbf{h} = \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix}, \quad (8)$$

formado por las componentes de la matriz H ,

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}, \quad (9)$$

siendo h_i como el i -ésimo elemento de \mathbf{h} .

La ecuación $A_i \mathbf{h} = \mathbf{0}$ es una ecuación lineal en la incógnita \mathbf{h} . A pesar de que hay tres ecuaciones en (7), solamente dos de ellas son linealmente independientes. Así, cada correspondencia de puntos resulta en dos ecuaciones en las componentes de H . Luego, el conjunto de ecuaciones se convierte en

$$\begin{bmatrix} \mathbf{0}^T & -w'_i \mathbf{x}_i^T & y'_i \mathbf{x}_i^T \\ w'_i \mathbf{x}_i^T & \mathbf{0}^T & -x'_i \mathbf{x}_i^T \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}, \quad (10)$$

que puede expresarse como

$$A_i \mathbf{h} = \mathbf{0}, \quad (11)$$

siendo A_i es ahora la matriz presente en (10).

Conocidos 4 puntos se obtienen 8 ecuaciones que permiten obtener la matriz H , ya que se tiene un grado de libertad para H por estar expresada en coordenadas homogéneas. Si se tienen más de 4 ecuaciones, el conjunto es sobredeterminado y, en presencia de ruido en la medición, se puede encontrar una solución aproximada, i.e., un vector \mathbf{h} que minimiza una función de costo aplicable. La solución propuesta por [10] es el vector singular unitario correspondiente al valor singular más pequeño de A , que puede obtenerse mediante la descomposición en valores singulares (o SVD por sus siglas en inglés).

II-C. Caracterización de las Cámaras

El modelo de cámara utilizado en el presente trabajo es de proyección central, que es una especialización de la cámara proyectiva general. El mismo es llamado pinhole camera según [10], que en español podría traducirse como cámara estenopeica. En el caso más general, se puede modelar la cámara mediante la siguiente matriz:

$$P = K[R|t]. \quad (12)$$

Los parámetros contenidos en K se llaman *parámetros internos* o *intrínsecos* de la cámara. La matriz K es llamada *matriz intrínseca* o *matriz de calibración* y tiene la forma:

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

donde f es la distancia focal y p_x y p_y representan las coordenadas en píxeles del punto principal, respectivamente. Los parámetros R y \mathbf{t} , que se relacionan con la orientación y posición de la cámara respecto al sistema de coordenadas fijo se llaman *externos* o *extrínsecos*.

Se puede entender la geometría epipolar como la geometría proyectiva intrínseca entre dos vistas [10]. Es independiente de la estructura del escenario y sólo depende de los parámetros internos de las cámaras y su posición relativa. La matriz fundamental F encapsula la geometría intrínseca. Es una matriz de 3×3 de rango 2. Si un punto \mathbf{X} del espacio es proyectado como \mathbf{x} en la primera vista y \mathbf{x}' en la segunda, entonces los puntos de la imagen satisfacen la ecuación.

$$\mathbf{x}'^T F \mathbf{x} = 0. \quad (14)$$

Se considera una matriz de cámara descompuesta como en (12), y sea $\mathbf{x} = P\mathbf{X}$ un punto en la imagen. Si la matriz de calibración K es conocida, podemos aplicar su inversa al punto \mathbf{x} para obtener $\hat{\mathbf{x}} = K^{-1}\mathbf{x}$. Así $\hat{\mathbf{x}} = [R|\mathbf{t}]\mathbf{X}$, donde $\hat{\mathbf{x}}$ es el punto de imagen expresado en coordenadas normalizadas. Se puede entender como una proyección del punto \mathbf{X} con respecto a la cámara $[R|\mathbf{t}]$ teniendo la matriz de identidad I como matriz de calibración. La matriz de cámara $K^{-1}P = [R|\mathbf{t}]$ es llamada matriz de cámara normalizada, donde el efecto de la conocida matriz de calibración se ha eliminado.

Así, considerando un par de matrices de cámara normalizadas $P = [I|\mathbf{0}]$ y $P' = [R|\mathbf{t}]$, se puede obtener la matriz fundamental correspondiente al par de cámaras normalizadas, denominada matriz esencial, como

$$E = [\mathbf{t}]_{\times} R = R[R^T \mathbf{t}]_{\times}, \quad (15)$$

donde se verifica

$$\hat{\mathbf{x}}'^T E \hat{\mathbf{x}} = 0 \quad (16)$$

en términos de las coordenadas de imagen normalizadas para los puntos correspondientes $\mathbf{x} \leftrightarrow \mathbf{x}'$.

Sustituyendo $\hat{\mathbf{x}}$ y $\hat{\mathbf{x}}'$, se tiene $\mathbf{x}'^T K'^{-T} E K^{-1} \mathbf{x} = 0$. Comparando esta relación con (14), se tiene:

$$E = K'^T F K. \quad (17)$$

Tanto la matriz fundamental como la matriz esencial pueden ser estimadas a partir de un conjunto de correspondencias de puntos $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, y a partir de estas matrices es posible obtener las matrices de cámara para así realizar el cómputo de la estructura.

En cuanto al método de triangulación, se utilizó el método de triangulación lineal. Considerando dos imágenes, se tienen dos mediciones $\mathbf{x} = P\mathbf{X}$ y $\mathbf{x}' = P'\mathbf{X}$. Estas expresiones pueden combinarse como $A\mathbf{X} = \mathbf{0}$, la cual es una ecuación lineal en \mathbf{X} .

Para resolver la ecuación, primero se elimina el factor de escala por medio de un producto vectorial que proporciona tres ecuaciones por cada punto de la imagen, los cuales son linealmente independientes. Por ejemplo, para la primera imagen, $\mathbf{x} \times (P\mathbf{X}) = \mathbf{0}$, teniéndose

$$\begin{aligned} x(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{1T}\mathbf{X}) &= 0, \\ y(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{2T}\mathbf{X}) &= 0, \\ x(\mathbf{p}^{2T}\mathbf{X}) - y(\mathbf{p}^{1T}\mathbf{X}) &= 0, \end{aligned} \quad (18)$$

donde \mathbf{p}^{iT} son las filas de P . Estas ecuaciones son lineales en las componentes de \mathbf{X} . Una ecuación de la forma $A\mathbf{X} = \mathbf{0}$ puede ser descompuesta, con

$$A = \begin{bmatrix} x\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'\mathbf{p}'^{3T} - \mathbf{p}'^{1T} \\ y'\mathbf{p}'^{3T} - \mathbf{p}'^{2T} \end{bmatrix}, \quad (19)$$

donde se incluyen dos ecuaciones por imagen, obteniéndose un total de cuatro ecuaciones en cuatro incógnitas homogéneas. Éste es un conjunto redundante de ecuaciones, ya que la solución es determinada para un determinado factor de escala, por la utilización de coordenadas homogéneas. Una manera de resolver el conjunto de ecuaciones $A\mathbf{X} = \mathbf{0}$ es el método

homogéneo (DLT), que encuentra como solución el vector singular unitario correspondiente al valor singular más pequeño de A .

III. RECONSTRUCCIÓN DEL ESCENARIO

En esta sección se describe el programa implementado, que permite llevar a cabo la reconstrucción del escenario. El lenguaje utilizado para la implementación es C++, en conjunto con la librería OpenCV [11]. Las entradas del programa son la matriz de cámara, y dos imágenes del escenario a reconstruir. Estas imágenes deben poseer un formato compatible con la librería OpenCV, como JPEG o PNG, entre otros. Por su parte, la salida consiste en una nube de puntos, con formato .PLY, de acuerdo a lo propuesto en [12]. La nube contiene los puntos reconstruidos. El origen de coordenadas de esta nube tiene el centro de cámara correspondiente a la primera imagen.

III-A. Detección y Descripción de Puntos Clave

La detección de puntos clave se realizó utilizando el algoritmo BRISK [8]. Para ello se utilizó un umbral igual a 30, que define la diferencia mínima necesaria entre los valores del punto central y algún punto de su vecindario según el criterio de BRISK para evaluar el punto clave. Se emplea una escala unitaria para el patrón utilizado en el muestreo del vecindario de los puntos de interés y se limita la búsqueda a tres octavas.

Tras la detección se realizó la descripción de los puntos clave, también utilizando el descriptor BRISK, posibilitando el pareo de estos puntos.

III-B. Pareo de puntos claves

En esta etapa se realiza una comparación de los descriptores de la primera imagen con los descriptores de la segunda, en busca de puntos homólogos. Esta comparación se hace por fuerza bruta y se califican estas comparaciones según la distancia de Hamming (cantidad de bits que deben cambiarse en una cadena para que sea idéntica a otra).

Para cada punto clave de la primera imagen \mathbf{x} se buscan los dos puntos clave de la segunda imagen \mathbf{x}'_1 , \mathbf{x}'_2 con la menor distancia Hamming, ya que ambos candidatos son necesarios para la utilización del criterio de Lowe [13] para la eliminación de pares incorrectos. El criterio puede formularse de la manera siguiente:

Si \mathbf{x} es un punto de la primera imagen y \mathbf{x}'_1 y \mathbf{x}'_2 los dos mejores candidatos correspondientes a la segunda imagen, se dice que \mathbf{x}'_1 es un par correcto si:

$$d_1 < kd_2 \quad \text{con: } 0 < k < 1, \quad d_1 < d_2$$

donde d_1 y d_2 son las distancias de Hamming entre \mathbf{x} y \mathbf{x}'_1 y entre \mathbf{x} y \mathbf{x}'_2 , respectivamente, y k es la razón de Lowe.

La razón de Lowe se determina empíricamente en la práctica. Valores de k comunes pueden encontrarse entre 0,7 y 0,8, por lo que el valor de la razón de Lowe elegido en este trabajo es de 0,8. Tras aplicar este criterio se eliminan la mayoría de los pares incorrectos debidos a patrones repetitivos o regiones de estructuras similares en las imágenes.

III-C. Cálculo de la matriz Esencial

Para el cálculo de la matriz esencial se utiliza la función de OpenCV `cv::findEssentialMat()` contenida en el módulo “Camera Calibration and 3D Reconstruction” [11].

Esta función recibe como parámetros de entrada dos vectores de puntos de tipo `vector<Point2f>`, correspondientes a los puntos de cada imagen; y una matriz de tipo `Mat`, matriz intrínseca de la cámara.

La función hace uso del algoritmo de los cinco puntos [9] para la estimación de la matriz esencial y se aplica el algoritmo *Consenso de Muestreo Aleatorio (RANSAC)* [14] para obtener la matriz esencial que satisface a la mayor cantidad de puntos. El algoritmo consiste en seleccionar repetidamente un conjunto aleatorio mínimo de datos para calcular el modelo y aceptar el modelo que tenga mayor apoyo. En este contexto, se define el modelo con mayor apoyo como aquél que incluye a la mayor cantidad de puntos dato dentro de un rango determinado por umbral (inliers), descartando los que no se adaptan al modelo (outliers). La salida, por su parte, es una matriz del tipo *Mat* y consiste en la matriz esencial, que contiene implícitamente la rotación y la translación relativa entre cámaras (bajo cierta escala, ya que solo se obtiene un vector unitario en la dirección de translación como se indicará en la siguiente sección).

III-D. Obtención de las Matrices de Cámara

Para obtener las matrices de cámara se utiliza la función de OpenCV `cv::recoverPose()` contenida también en el módulo “Camera Calibration and 3D Reconstruction” [11].

Los parámetros de entrada son una matriz *E* del tipo `Mat`, es decir, la matriz esencial; dos vectores de puntos de tipo `vector<Point2f>` correspondientes a las coordenadas de puntos de imagen de ambas vistas; la matriz de cámara, *K*, también del tipo `Mat`; y una máscara en la que se incluyen los puntos que cumplen con el cheirality check, es decir, que se encuentran enfrente de la cámara, tal como explica se explica en [15]. Las salidas de interés son la matriz de rotación, de tipo `Mat`; y el vector de translación del tipo `Mat`.

Esta función calcula la rotación y translación relativa entre ambas cámaras a partir de la matriz esencial y utiliza los puntos para elegir un par entre las posibles soluciones de *R* y *t*. Cabe destacar que el vector *t* se devuelve como un vector unitario (sólo su dirección) debido al grado de libertad de la notación proyectiva, por lo que la reconstrucción se obtiene salvo por un factor de escala, pero basta con escalar este vector a la escala real para obtener la reconstrucción Euclídea.

Una vez obtenida la rototranslación relativa entre cámaras es posible ubicar el centro de la primera cámara en el origen del sistema de coordenadas con el eje *Z* en la dirección del eje principal de la cámara y dejando a la imagen en el semiespacio positivo respecto de *Z*, y con los ejes *X* e *Y* respectivamente paralelos a los ejes *x* e *y* de la imagen. De este modo, las matrices de las cámaras vienen dadas por:

$$P_1 = K[I|0], \quad (20)$$

$$P_2 = K[R|t]. \quad (21)$$

Con estas matrices de cámara es posible realizar la reconstrucción de la nube de puntos por medio empleando

triangulación. Además, es posible realizar la rototranslación, ya sea de las cámaras o de los puntos reconstruidos por medio de triangulación, a cualquier otro sistema de referencia en el mundo tridimensional.

III-E. Triangulación de Puntos

Una vez obtenidas las matrices de cámara es posible realizar la triangulación lineal [10]. Para ello se cuenta con la función `cv::triangulatePoints()` de OpenCV.

La salida es una matriz del tipo `Mat` de puntos 3D en coordenadas homogéneas en \mathbb{P}^3 , i.e., cada punto tiene cuatro componentes y con ello la matriz tiene cuatro filas, conteniendo a un punto cada columna.

Cada uno de estos puntos puede transformarse a sus coordenadas Euclidianas, normalizando cada punto por el valor de su última coordenada, así esta última se iguala a la unidad y las primeras tres son respectivamente *X*, *Y* y *Z* correspondientes a \mathbb{R}^3 .

Estos puntos pueden registrarse, con su correspondiente color, en un archivo PLY [12], obteniéndose una nube de puntos.

III-F. Densificación de la Nube de Puntos

Los puntos de la nube descrita en la sección precedente no son identificables para el ojo humano por la ausencia de detalles, por lo que se realizó una densificación con la finalidad de facilitar el reconocimiento de puntos en la nube y las posteriores mediciones. Pese a tratarse de una etapa previa a la triangulación, se incluye como último apartado ya que este proceso es opcional. Consiste en añadir pares de puntos al conjunto de puntos que satisfacen la geometría epipolar, consiguiendo así mayor detalle en la nube debido a la mayor densidad de puntos.

Para realizar la densificación se usó un algoritmo de propagación basado en el propuesto por Lhuillier y Quan [16], utilizando como entrada sólo aquellos puntos que satisfacen la geometría epipolar. Además, son añadidos al conjunto solución *map* y a la semilla *seed* únicamente los puntos que verifican la restricción epipolar $\mathbf{x}_i'^T F \mathbf{x}_i < 0,01$.

Esta etapa permite rellenar los vecindarios de los puntos clave, es decir, utiliza los puntos iniciales como semillas a partir de las cuales se expande para así incrementar los detalles alrededor de puntos aislados y así facilitar el reconocimiento de puntos y objetos en la imagen. Se incluyó esta etapa debido a la dificultad para determinar con precisión los segmentos en la nube de puntos al momento de realizar mediciones.

IV. RESULTADOS OBTENIDOS

Para las pruebas experimentales se realizaron dos conjuntos de prueba. Primeramente, fotografías a baja altura de un escenario controlado en el cual se introdujeron detalles con textura en los vecindarios de los puntos que definirían los segmentos a ser medidos. Para este caso la cámara utilizada fue la Canon PowerShot S100. Para el segundo conjunto de prueba se tomaron fotografías de un escenario que presenta características similares al escenario controlado utilizando el

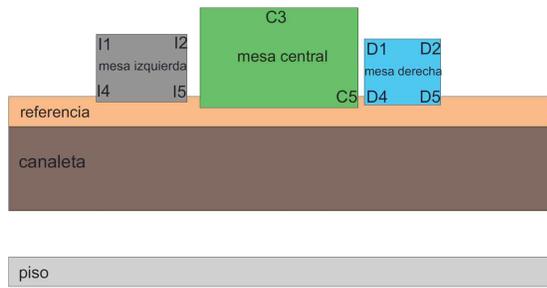


Figura 1. Cada plano evaluado se muestra en un color, además, se muestran los puntos extremos de los segmentos medidos.



Figura 2. Conjunto de imágenes de entrada para el escenario controlado.

drone Autel X-Star Premium, a alturas de 15 m y 20 m. Ambas cámaras fueron previamente calibradas utilizando la rutina `cv::calibrateCamera()` contenida en la librería OpenCV. Para un correcto funcionamiento del algoritmo de relleno las capturas de imágenes se realizaron de modo que los objetos de las mismas mantengan en lo posible un tamaño similar, ubicando la cámara a la misma altura. En el caso del escenario controlado además se dispuso la cámara con su eje principal perpendicular a los distintos planos presentes en el escenario, trasladándola una distancia de 1 m entre cada toma, en dirección horizontal (a lo largo del eje X), evitando la posibles rotaciones con ayuda de un nivel de burbuja. De esta forma se garantiza que la información de la altura se encuentra contenida exclusivamente en el eje Z . La característica evaluada en el presente trabajo es la conservación de la norma euclidiana de segmentos, para lo cual se realizaron mediciones de los escenarios a reconstruir con cinta métrica y, posteriormente, se registraron las medidas de los mismos segmentos en cada reconstrucción obtenida.

IV-A. Fotografías a Baja Altura

Para el caso de fotografías a baja altura, las mediciones se clasificaron en dos grupos: alturas, esto es, distancias de los planos al plano de referencia (denominadas distancias verticales) y longitudes de segmentos determinados por puntos identificados sobre las tres mesas presentes en el escenario (denominadas distancias horizontales). Estos planos y puntos, con sus respectivos identificadores, se presentan esquemáticamente en la Figura 1. El conjunto de imágenes utilizado se presenta en la Figura 2. Las mediciones fueron realizadas utilizando el software CloudCompare [17].

En el Cuadro I se recogen las mediciones de las distancias verticales, esto es, las alturas de los distintos planos del escenario respecto al nivel de referencia, medidas sobre el eje Z , tal como fue descrito anteriormente, para una nube obtenida a partir de dos imágenes del conjunto. El Cuadro II contiene las mediciones de las distancias horizontales, esto es,

Cuadro I
DISTANCIAS VERTICALES PLANOS PARA UNA NUBE DEL ESCENARIO CONTROLADO. LAS DISTANCIAS SE INDICAN EN CM.

Plano	Valor medido [cm]	Valor real [cm]	Error [cm]	Error porcentual %
Mesa cen	78.2	75	3.2	4.3
Mesa izq	70.1	66.1	4.0	6.1
Mesa der	50.6	50.1	0.5	1.0
Piso	21.5	20.1	1.4	7.0
Canaleta	-28.2	-26.7	-1.5	5.6

Cuadro II
DISTANCIAS HORIZONTALES PARA UNA NUBE DEL ESCENARIO CONTROLADO. LAS DISTANCIAS SE INDICAN EN CM.

Segmento	Valor medido (nube)	Valor real (escenario)	Error	Error porcentual (%)
I1I2	46.9	45.5	1.4	3.1
I4I5	47.9	46.5	1.4	3.0
D4D5	50.5	50.3	0.2	0.4
I1I4	36.0	34.9	1.1	3.2
I2I5	36.5	35.7	0.8	2.2
D1D4	40.2	39.7	0.5	1.3
C3C5	62.3	59.8	2.5	4.2

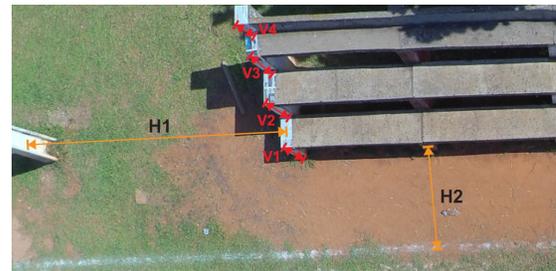


Figura 3. Segmentos medidos en las fotografías aéreas.



Figura 4. Par de imágenes a 20m de altura.

distancias entre los puntos elegidos sobre los distintos planos del escenario para la misma nube.

IV-B. Fotografías aéreas

Luego de obtener los resultados anteriores se tomaron fotografías con el drone Autel X-Star Premium sobre un terreno real, a 15 y 20 m de altura, en donde se realizaron las mediciones de los segmentos que se muestran en la Figura 3. Las imágenes de entrada para una altura de 20 m se presentan en la Figura 4 y los resultados de las mediciones se recogen en los Cuadros III (15 m) y IV (20 m).

Cuadro III
RESULTADOS PARA UNA ALTURA DE 15 m. LAS DISTANCIAS SE INDICAN EN cm.

Plano	Valor medido [cm]	Valor real [cm]	Error [cm]	Error porcentual (%)
H1	487.56	487	0.56	0.11
H2	184.86	180	4.86	2.7
V1	65.16	69	-3.84	5.57
V2	57.59	56	1.59	2.84
V3	53.64	56	-2.36	4.21
V4	29.41	30	-0.59	1.97

Cuadro IV
RESULTADOS PARA UNA ALTURA DE 20 m. LAS DISTANCIAS SE INDICAN EN cm.

Plano	Valor medido [cm]	Valor real [cm]	Error [cm]	Error porcentual (%)
H1	482.59	487	-4.41	0.90
H2	184.2	180	4.2	2.33
V1	62.31	69	-6.69	9.69
V2	57.19	56	1.19	2.12
V3	53.3	56	-2.7	4.82
V4	30.2	30	0.2	0.66

IV-C. Análisis de los resultados

Se logró una reconstrucción con exactitud aceptable, obteniendo errores por debajo de 5 cm en las alturas de zonas planas con gran detalle, para alturas de hasta 20 m, cubriendo áreas de hasta aproximadamente $37 \times 22 \text{ m}^2$.

El tiempo de cómputo está directamente relacionado con el orden de complejidad del algoritmo de densificación, $O(n \log(n))$, siendo n la cantidad de puntos que se requieren como salida.

La densificación presenta problemas en el caso de propagación en las direcciones de las rectas epipolares, debido a que la restricción epipolar es insuficiente para determinar pares erróneos en casos de alta correlación. Sin embargo, permite incrementar el detalle y favorece la identificación de puntos a simple vista.

La resolución de la cámara limita la altura a la que es posible capturar las imágenes y el nivel de detalle en las reconstrucciones. Por tanto, se recomiendan cámaras de alta resolución y sensores de buena calidad.

V. CONCLUSIONES

En este artículo se ha descrito la implementación de un método de reconstrucción automático de escenarios empleando un par de imágenes aéreas capturadas con un vehículo aéreo no tripulado (Unmanned Aerial Vehicle, UAV). Se logró satisfactoriamente la reconstrucción con un par de vistas y las distintas etapas que conllevan el proceso. Sin embargo, en cuanto al algoritmo de densificación integrado, se concluye que el método genera múltiples restricciones y, aunque aumenta el detalle en la reconstrucción, genera gran cantidad de falsos positivos que, si bien pueden ser eliminados manualmente, atentan contra el proceso totalmente automático, por lo que

se opta por posponer el estudio de la densificación para un momento posterior a la integración de múltiples vistas en el proceso de reconstrucción, el cual vendría a ser el trabajo futuro en esta línea de investigación.

ACKNOWLEDGMENT

La publicación se ha realizado en el marco del proyecto de investigación PINV15-0136 denominado "Vehículos aéreos no tripulados (UAV) en aplicaciones de fotogrametría para el análisis planialtimétrico y cálculo de volumen" cofinanciado por el Consejo Nacional de Ciencia y Tecnología - CONACYT (Paraguay) con recursos del FEEL.

REFERENCIAS

- [1] L. A. Haidari, S. T. Brown, M. Ferguson, E. Bancroft, M. Spiker, A. Wilcox, R. Ambikapathi, V. Sampath, D. L. Connor, and B. Y. Lee, "The economic and operational value of using drones to transport vaccines," *Vaccine*, vol. 34, no. 34, 2016.
- [2] A. Tariq, S. M. O. A. Gillani, H. K. Qureshi, and I. Haneef, "Heritage preservation using aerial imagery from light weight low cost unmanned aerial vehicle (uav)," in *2017 International Conference on Communication Technologies (ComTech)*, April 2017.
- [3] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision Agriculture*, vol. 13, no. 6, Dec 2012.
- [4] A. Tariq, S. M. Osama, and A. Gillani, "Development of a low cost and light weight uav for photogrammetry and precision land mapping using aerial imagery," in *2016 International Conference on Frontiers of Information Technology (FIT)*, Dec 2016.
- [5] A. Segales, R. Gregor, J. Rodas, D. Gregor, and S. Toledo, "Implementation of a low cost uav for photogrammetry measurement applications," in *Proc. International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 926–932.
- [6] E. Paiva, M. Llano, J. Rodas, R. Gregor, J. Rodríguez-Pineiro, and M. Gomez-Redondo, "Design and implementation of a VTOL flight transition mechanism and development of a mathematical model for a tilt rotor UAV," in *Proc. IEEE International Conference on Automation /Congress of Chilean Association of Automatic Control (IEEE ICA-ACCA)*, 2018, pp. 1–6.
- [7] Y. Yang, Z. Lin, and F. Liu, "Stable imaging and accuracy issues of low-altitude unmanned aerial vehicle photogrammetry systems," *Remote Sensing*, vol. 8, no. 4, 2016.
- [8] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, Nov 2011.
- [9] D. Nistér, "An efficient solution to the five-point relative pose problem," in *CVPR (2)*. IEEE Computer Society, 2003, pp. 195–202.
- [10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [11] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [12] G. Turk. (2016, Marzo) Ply - polygon file format. [Online]. Available: <http://paulbourke.net/dataformats/ply/>
- [13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, 2004.
- [14] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
- [15] R. I. Hartley, "Cheirality invariants," in *Proc. DARPA Image Understanding Workshop*, 1993.
- [16] M. Lhuillier and L. Quan, "Robust Dense Matching Using Local and Global Geometric Constraints," in *15th International Conference on Pattern Recognition (ICPR '00)*, vol. 1, Sep. 2000.
- [17] (2018) Cloudcompare. [Online]. Available: <http://www.cloudcompare.org/>