

# Detección semiautomática de puntos de control en tierra (GCP) usando flujo óptico

Jhacson A. Meza-Arteaga

Facultad de ingeniería, Universidad Tecnológica de Bolívar, Cartagena de Indias, Colombia

[jhacsonmeza121@gmail.com](mailto:jhacsonmeza121@gmail.com)

**Resumen**—En este trabajo se propone una metodología para la detección semiautomática de puntos de control en tierra (GCP), los cuales son necesarios para el proceso de reconstrucción de terrenos a través de la técnica fotogramétrica de Structure from Motion (SfM). Esta detección se propone usando flujo óptico con el algoritmo de Lucas-Kanade-Tomasi (KLT) implementado con la librería de código abierto OpenCV.

## I. INTRODUCCIÓN

En los años recientes la fotogrametría se ha convertido en una herramienta poderosa y ampliamente utilizada para el modelado topográfico tridimensional. Para atender este problema se ha propuesto la técnica estéreo-fotogramétrica de Structure from Motion (SfM) para la generación de reconstrucciones topográficas de alta resolución donde se resuelve la geometría 3D de la escena y la posición y orientación de la cámara automáticamente a través de las características de una serie de imágenes superpuestas en un proceso de optimización conocido como *bundle adjustment*, generando así una nube de puntos de la escena.

El problema de esta técnica es que con simplemente las imágenes y los puntos característicos extraídos de estas, la reconstrucción obtenida no suele tener una escala adecuada a la real, es decir, no suele ser métrica. Esto es debido a que, con solo las fotografías, se puede recuperar las poses de la cámara pero en una escala arbitraria, lo que genera que los puntos 3D calculados se encuentren en un sistema coordenado arbitrario o un sistema de coordenadas relativo en el “espacio imagen” que debe ser convertido a un sistema coordenado real o un sistema coordenado en el “espacio objeto” en el cual la reconstrucción sí es métrica [1].

Es aquí donde los puntos de control en tierra (GCP), que son puntos con coordenadas conocidas en el espacio objeto, juegan un papel fundamental para poder transformar la nube de puntos del espacio imagen al espacio objeto. Estos GCP deben ser fáciles de ver tanto en las imágenes como en la nube de puntos y su posición en el mundo real suele ser medida con ayuda de un GPS u otro tipo de técnica de levantamiento topográfico.

Normalmente, para cumplir con el requisito de que los GCP deben ser fáciles de identificar, se suelen usar elementos físicos con un cierto patrón pintado de alto contraste y que posea un centroide claramente definido, como por ejemplo el de la figura 1. Luego estos son puestos sobre la zona de interés a reconstruir antes del proceso de adquisición de imágenes. Este enfoque nos ayuda a quitar la ambigüedad en la ubicación del



Figura 1. Ejemplo de target que puede ser usado como GCP.

GCP tanto en el espacio objeto como en el espacio imagen permitiéndonos poder medir un error en la reconstrucción.

Teniendo en cuenta entonces que los puntos de control tienen un centroide definido que es fácil de identificar y que además carecen de ambigüedad, en este trabajo se propone una metodología para la detección semiautomática de puntos de control a través de las imágenes adquiridas de un cierto terreno. Para esto haremos uso del concepto de flujo óptico el cual nos permite saber la dirección del movimiento aparente de los diferentes elementos en una escena causado por el movimiento relativo de la cámara [2]. Es en base a esto que sabremos hacia dónde se moverá el centroide del target entre una imagen y otra de forma que haremos un seguimiento o tracking de este punto en todas las imágenes donde aparezca.

## II. ENFOQUE

Para la generación de reconstrucciones topográficas a través de la técnica de SfM en los últimos años se ha propuesto el uso de vehículos aéreos no tripulados (UAV), o drones comerciales, para la adquisición de imágenes. Normalmente se suelen capturar las fotografías del terreno a reconstruir haciendo que el dron siga una ruta de vuelo específica, como la que se puede ver en la figura 2. Donde tenemos que el recorrido del dron está dado por la línea blanca en la dirección indicada por las flechas. A través de esta ruta se capturan una serie de imágenes con un cierto porcentaje de traslape entre ellas, de tal forma que luego del proceso de adquisición, se tenga un conjunto de fotografías con cierta sucesión entre ellas parecidas a varios fotogramas de un video. Esto nos ayuda a suponer de entrada que hay cierto orden en las imágenes capturadas y que el movimiento de la cámara con respecto a la



Figura 2. Ejemplo de ruta de vuelo para adquisición de imágenes hecha con la aplicación móvil Altizure.

escena no es demasiado exagerado, sino, que va a estar dado por ese porcentaje de traslape entre las fotografías ajustado antes del proceso de adquisición. Este aspecto es importante ya que vamos a poder usar la técnica de flujo óptico sin tener la preocupación de que no se pueda hacer el tracking del target entre cualquier par de fotografías consecutivas.

### III. FLUJO ÓPTICO

La idea con el flujo óptico es poder calcular el vector de movimiento de una serie de puntos de un primer frame (o vista) al segundo. Para esto se parte de dos supuestos: los valores de intensidad de los píxeles entre dos vistas no cambian significativamente en intensidad y además los píxeles vecinos a un punto tienen un movimiento similar.

Es decir que, si considerando una imagen  $I(x, y, t)$  en la primera vista, añadiendo el tiempo como una nueva dimensión ya que las dos vistas son capturadas en instantes de tiempo diferentes. En la segunda vista los puntos de la imagen  $I$  se mueven una distancia  $[dx \ dy]$  después de un tiempo  $dt$ . Partiendo del primer supuesto de que los píxeles en las dos imágenes no cambian en intensidad, nosotros podemos decir que:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) . \quad (1)$$

Matemáticamente esta invarianza en intensidad la podemos expresar como:

$$\frac{\partial I(x, y, t)}{\partial t} = 0$$

Entonces aplicando la regla de la cadena para varias variables en la ecuación (1), tenemos:

$$\begin{aligned} \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} &= 0 \\ \frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} &= 0 , \end{aligned} \quad (2)$$

donde  $V_x$  y  $V_y$  son las componentes en  $x$  y  $y$  de la velocidad o el flujo óptico de  $I(x, y, t)$  las cuales son incógnitas.

Esta ecuación 2 suele ser conocida como la ecuación de flujo óptico y para resolver este problema se han propuesto

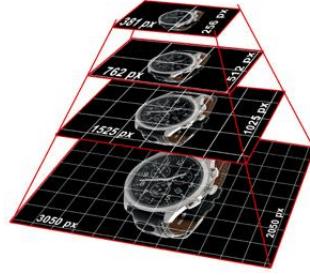


Figura 3. Imagen piramidal con 4 niveles de submuestreo.

varios métodos, ya que como vemos, tenemos dos incógnitas y una sola ecuación. Entre esos tenemos el método de Lucas-Kanade (LK) el cual se basa en el segundo supuesto de que los píxeles vecinos a un punto tienen un movimiento similar, lo que permite tomar una ventana de  $3 \times 3$  alrededor del punto teniendo todos los 9 píxeles que componen esta ventana un movimiento similar, es decir, la componente  $V_x$  y  $V_y$  para todos estos es igual. De esta manera aumentamos el número de ecuaciones a 9 (una para cada píxel de la ventana) y mantenemos el mismo número de 2 incógnitas debido al supuesto. La mejor solución para este problema es obtenido usando mínimos cuadrados. Para ello haremos lo siguiente: tomando  $\nabla I$  como

$$\nabla I = \begin{bmatrix} \partial I / \partial x \\ \partial I / \partial y \end{bmatrix} = \begin{bmatrix} I_x \\ I_y \end{bmatrix} ,$$

y tomando  $V$  como:

$$V = \begin{bmatrix} V_x \\ V_y \end{bmatrix} ,$$

podemos reescribir la ecuación de flujo óptico (2) de la siguiente manera:

$$\nabla I^T V = -\frac{\partial I}{\partial t} . \quad (3)$$

Por tanto la solución por mínimos cuadrados de  $V$  en (3) multiplicando por  $\nabla I$  a ambos lados de la igualdad y despejando  $V$ , está dada por:

$$V = (\nabla I \nabla I^T)^{-1} \nabla I \left( -\frac{\partial I}{\partial t} \right) ,$$

que finalmente nos queda como:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_{x_i}^2 & \sum_i I_{x_i} I_{y_i} \\ \sum_i I_{x_i} I_{y_i} & \sum_i I_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_{x_i} I_{t_i} \\ -\sum_i I_{y_i} I_{t_i} \end{bmatrix} . \quad (4)$$

### IV. TRACKING DE PUNTOS CON ALGORITMO KLT

Para el cálculo del flujo óptico para el tracking de los GCP vamos a usar el algoritmo Kanade-Lucas-Tomasi (KLT). El cual consiste en el detector de puntos característicos de Shi-Tomasi y el método de Lucas-Kanade piramidal [3].

El detector de Shi-Tomasi nos proporciona puntos característicos óptimos para el proceso de tracking. Este es parecido

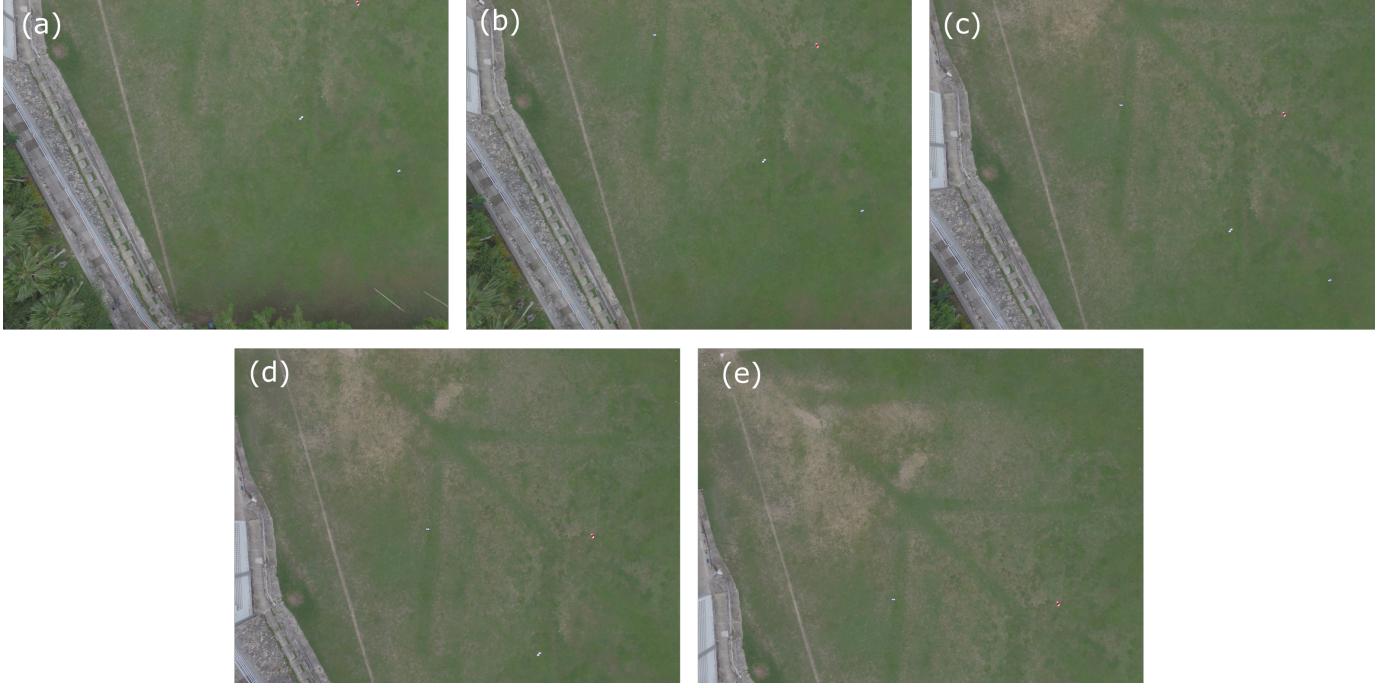


Figura 4. Imágenes de prueba para la detección del GCP seleccionado manualmente en la figura (a) el cual tiene un punto rojo. En las demás figuras podemos ver la estimación del algoritmo de la posición del GCP.

al detector de esquinas de Harris, donde se usa una función de respuesta a esquinas dada por  $R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$ . A diferencia de esto Shi-Tomasi propone que se calcule la respuesta a las esquinas como  $R = \min(\lambda_1, \lambda_2)$ , donde los puntos con valores de  $R$  mayor a cierto umbral son considerados como detecciones [4]. La idea de este detector, tal como se llama el artículo donde Shi-Tomasi propone esta metodología [4], es poder calcular puntos que puedan ser bien rastreados y correspondan a puntos físicos en el espacio objeto.

Una vez tengamos características buenas en un primer frame para el proceso de tracking en un segundo frame, vamos a hacer uso del método de Lucas-Kanade piramidal, del cuál podemos saber el flujo óptico entre un frame y otro a través de la ecuación (4) ya explicada. El problema con usar simplemente esa ecuación es que está diseñada para movimientos muy pequeños entre la cámara y la escena, para movimientos grandes el algoritmo falla. Para superar este problema lo que se recomienda hacer es una pirámide con la imagen, de tal forma que se vaya submuestreando en varios niveles hasta formar una pirámide (como el de la figura 3), para así, convertir los grandes desplazamientos en pequeños a medida que se sube en la pirámide, logrando hacer tracking en el caso de desplazamientos más grandes.

## V. ALGORITMO PROPUESTO

El algoritmo propuesto para el tracking de GCP parte del supuesto de que se escoge manualmente todas las imágenes donde se vea un target determinado. Con lo explicado en la adquisición de imágenes, vamos a tener que estas fotografías van a tener una sucesión entre ellas. Como parte de la inicialización se calcularán  $n$  puntos característicos, en la primera fotografía del conjunto de imágenes del target, con

la función `goodFeaturesToTrack` que ofrece OpenCV el cual usa el detector de características de Shi-Tomasi. Luego se muestra una ventana con dicha fotografía y todos los puntos detectados, donde el usuario deberá escoger de forma manual cuál es el centroide del target (GCP) que se desea seguir a través de las demás imágenes. Una vez hecho esto se guarda automáticamente la posición  $(x, y)$  y el índice del punto de interés escogido.

Después de escoger el GCP se procederá a través de un bucle `for` donde se recorra el resto de cada imagen, a estimar los puntos del primer frame en el segundo a través del métodos de Lucas-Kanade piramidal implementado en OpenCV a través de la función `calcOpticalFlowPyrLK`. Esta estimación se hace conservando el orden de los índices de la imagen del primer frame, de tal forma que si el usuario escogió el punto  $k$  del conjunto de  $n$  puntos detectados en la imagen 1, el mismo índice  $k$  en el resultado de las estimación con el algoritmo Lucas-Kanade de los  $n$  puntos en la imagen 2 también corresponderá al centroide del target siempre y cuando la estimación no haya sido errónea. Con esto, se vuelven a calcular puntos característicos en el segundo frame y se mide la distancia del centroide del target estimado con flujo óptico con el resto de puntos detectados para así refinar la posición de este. Luego de esto se actualiza el índice el target  $k$  y los puntos del segundo frame pasan a ser los del primero y el índice del GCP de la imagen 2 pasa a ser la de la 1 para la siguiente iteración.

El proceso de la metodología propuesta explicado anteriormente es ilustrado en el algoritmo 1.

---

**Algorithm 1** Tracking GCP

---

**Require:** Ruta a la carpeta con las  $N$  imágenes de un target físico específico.

- 1: **Inicialización**
  - 2: Cargar primera imagen de la carpeta como  $im_1$
  - 3: Detectar  $n$  características de Shi-Tomasi  $p_1$  en  $im_1$
  - 4: Mostrar  $im_1$  y  $p_1$  para escoger manualmente el GCP.
  - 5: Guardar posición  $x_1$  del GCP y el indice  $k_1$  del punto en el conjunto de  $n$  características.
  - 6: **for** el resto de  $N - 1$  imágenes de la carpeta **do**
  - 7:     Cargar imagen  $im_2$
  - 8:     Hallar  $p_2LK$ : puntos en  $im_2$  con LK.
  - 9:     Estimar  $n$  puntos Shi-Tomasi  $p_2$ .
  - 10:    Hallar distancia  $d = |p_2 - p_2LK(k_1)|$
  - 11:     $k_2 = \min(d)$
  - 12:     $x_i = p_2(k_2)$
  - 13:    Guardar  $im_2$  con un punto en  $x_i$
  - 14:    Actualizar:  $im_1 = im_2$
  - 15:    Actualizar:  $p_1 = p_2$
  - 16:    Actualizar:  $k_1 = k_2$ .
  - 17: **end for**
- 

## VI. PRUEBAS Y RESULTADOS

Para probar el algoritmo propuesto se han capturado un total de 5 imágenes con varios puntos de control sobre la escena, habiendo un 80 % de traslape entre ellas. Manualmente se ha escogido uno de los GCP, el de la figura 4(a), y este punto se ha estimado en las demás fotografías automáticamente con el algoritmo diseñado.

## VII. CONCLUSIONES

Es posible usar la técnica de tracking de puntos con flujo óptico para estimar de forma automática la posición de un determinado punto físico en una serie de sucesión de imágenes. Vemos que también se ha obtenido una detección exitosa respecto al traslape de las imágenes que ha sido de 80 %. Sería bueno probar para porcentajes mayores para saber el alcance y la eficiencia del algoritmo.

## REFERENCIAS

- [1] M. Westoby, J. Brasington, N. Glasser, M. Hambrey, and J. Reynolds, “structure-from-motion photogrammetry: A low-cost, effective tool for geoscience applications,” *Geomorphology*, vol. 179, pp. 300–314, 2012.
- [2] A. Burton and J. Radford, *Thinking in perspective: critical essays in the study of thought processes*. Methuen, 1978.
- [3] J.-Y. Bouguet, “Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm,” *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.
- [4] J. Shi *et al.*, “Good features to track,” in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 593–600, IEEE, 1994.