Odometría Visual Monocular

Descripción del problema:

Se tiene una secuencia de imágenes capturadas con una sola cámara en movimiento, de la cual se obtuvieron previamente sus parámetros intrínsecos mediante un proceso de calibración. A partir de un par de imágenes obtenidas en el tiempo $I(t)$ e I$(t+1)$ se quiere estimar la matriz de rotación R y el vector de translación T que relacionan el movimiento entre estos dos cuadros, y por lo tanto de nuestro sistema, conocidos como parámetros extrínsecos de la cámara. Al ser un esquema monocular, donde no se puede determinar la profundidad, tampoco podemos determinar con exactitud el vector de traslación T, solo puede ser estimado a un factor de escala desconocido si no se cuenta con información externa a la proporcionada por las imágenes, como puede ser información previamente conocida de las características del entorno u otro tipo de sensor como IMU/GPS [1].

Marco teórico

La relación entre los sistemas de coordinadas del mundo real y los de la cámara están relacionados por un conjunto de parámetros físicos, como la distancia focal de la lente, el tamaño de los pixeles, la posición del centro de la imagen y la posición y orientación de la cámara [2]. Uno de los modelos más que hacen esta relación es el de la cámara *pinhole*, que se describe en la figura 2.

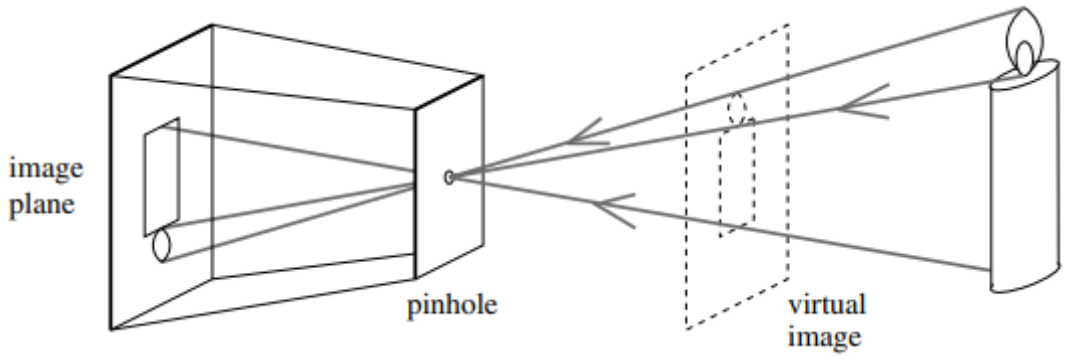

Figura. Modelo de una cámara pinhole. Obtenida de [2]

Con este modelo, el plano de la imagen es formado al proyectar los puntos 3D utilizando una transformación de perspectiva mostrada en la figura 5 y definida en la ecuación (3)

$$(3) \qquad s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r11 & r12 & r13 & t1 \\ r21 & r22 & r23 & t2 \\ r31 & r32 & r33 & t3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

o la ecuación (4)

$$(4) \qquad sm' = K[R|T]M'$$

Donde:

- (X, Y, Z) es la posición de un punto 3D en el sistema de coordenadas del mundo real.
- (u, v) son las coordenadas de la proyección del punto en pixeles
- K es la matriz de la cámara y contiene sus parámetros intrínsecos.
- $[R|T]$ es la matriz de los parámetros extrínsecos. Contiene una matriz de rotación R y un vector de traslación T. Es usada para describir el movimiento de una escena transformando las coordenadas de (X, Y, Z) al de la cámara.
- (c_x, c_y) es la posición del centro de la imagen o punto principal.
- f_x, f_y son las distancias focales expresadas en pixeles. Es la distancia desde el pinhole hasta el plano de la imagen.

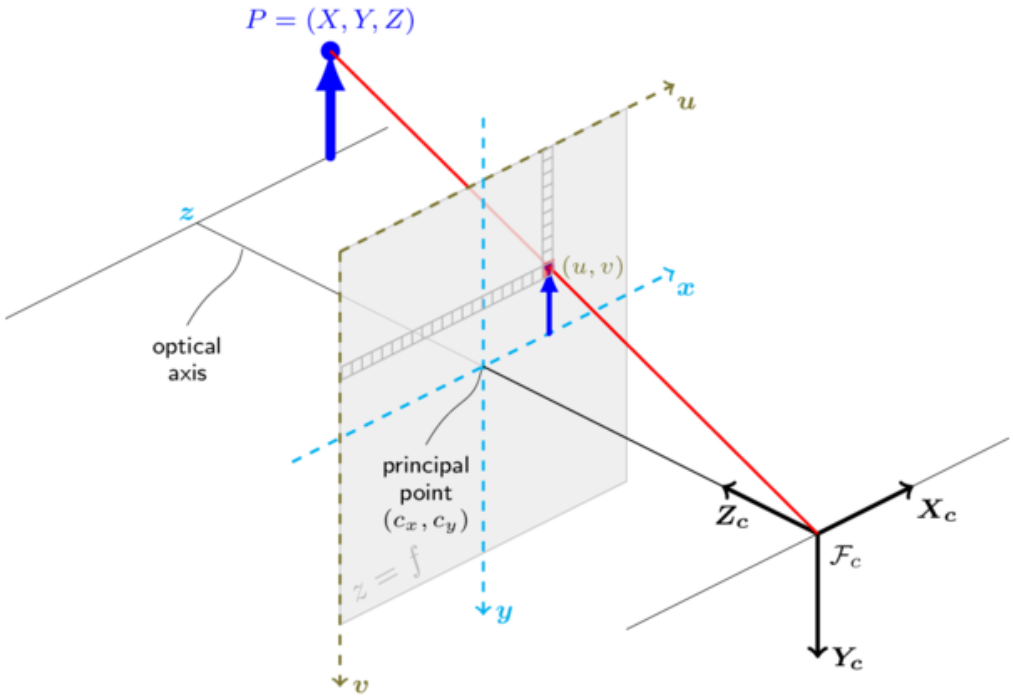

Figura. Proyección de un punto (X, Y, Z) en el plano de la imagen. Obtenida de [3].

Los lentes reales además contienen distorsión radial y en menor medida distorsión tangencial [3], al considerar estos factores el modelo anterior se extiende a las ecuaciones (5) y (6).

$$(5) \qquad u = f_x x'' + c_x$$
$$(6) \qquad u = f_y y'' + c_y$$

Donde:

- $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T$
- $x' = x/z$
- $y' = y/z$
- $x'' = x' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + 2p_1 x' y' + p_2\left(r^2 + 2x'^2\right)$
- $y'' = y' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + p_1\left(r^2 + 2y'^2\right) + 2p_2 x' y'$

- Los coeficientes k_n son los pertenecientes a la distorsión radial
- Los coeficientes p_n son los pertenecientes a la distorsión tangencial

En la figura 9 se muestras los efectos de la distorsión de la lente en una imagen.

Figura. Efectos de la distorsión radial y tangencial. Obtenida de [3]

El proceso para obtener los parámetros intrínsecos de la cámara, incluyendo los coeficientes de distorsión, es conocido como calibración de una cámara. Una manera común de obtenerlos es utilizando un patrón con medidas previamente conocidas y características fáciles de definir, como esquinas definidas y altos contrastes de color. La matriz de la cámara obtenida es específica para el dispositivo utilizado durante el proceso y una vez obtenida puede ser reutilizada en otras imágenes [4].

Flujo óptico es el patrón de movimiento aparente de objetos en la imagen en dos cuadros consecutivos, que puede ser causado por el movimiento del mismo objeto o de la cámara. En las técnicas de flujo óptico se supone que las intensidades de los pixeles de una imagen no cambian entre cuadros consecutivos y que los pixeles adyacentes tienen un movimiento similar [5]. Considerando estas condiciones en un píxel $I(x, y, t)$ que se mueve una distancia (dx, dy) en el siguiente cuadro después de un tiempo dt, podemos decir que

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Después se realiza una aproximación con series de Taylor, se remueven términos comunes y se divide entre dt para obtener la ecuación (8), conocida como la ecuación del flujo óptico.

$$(8) \quad f_x u + f_y v + f_t = 0$$

Donde:

- $fx = \frac{\partial f}{\partial x}$ y $fy = \frac{\partial f}{\partial y}$ son los gradientes de la imagen
- $u = \frac{dx}{dt}$ y $v = \frac{dy}{dt}$ son desconocidos.

No es posible solucionar esta sola ecuación con dos variables desconocidas, por lo que se han inventado varios métodos para resolver este problema y uno de ellos es Lucas-Kanade (LK), el cual toma el conjunto de pixeles que rodean al punto de interés, formando una matriz de 3x3. Suponiendo que todos los pixeles tienen el mismo movimiento, el problema anterior se convierte en resolver 9 ecuaciones con 2 incógnitas

sobredeterminadas. Se puede obtener una solución con el método de ajuste por mínimos cuadrados [6], resultando en la ecuación (9)

$$(9) \quad \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum ifxi^2 & \sum ifxifyi \\ \sum ifxifyi & \sum ifyi^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum ifxifti \\ -\sum ifyifti \end{bmatrix}$$

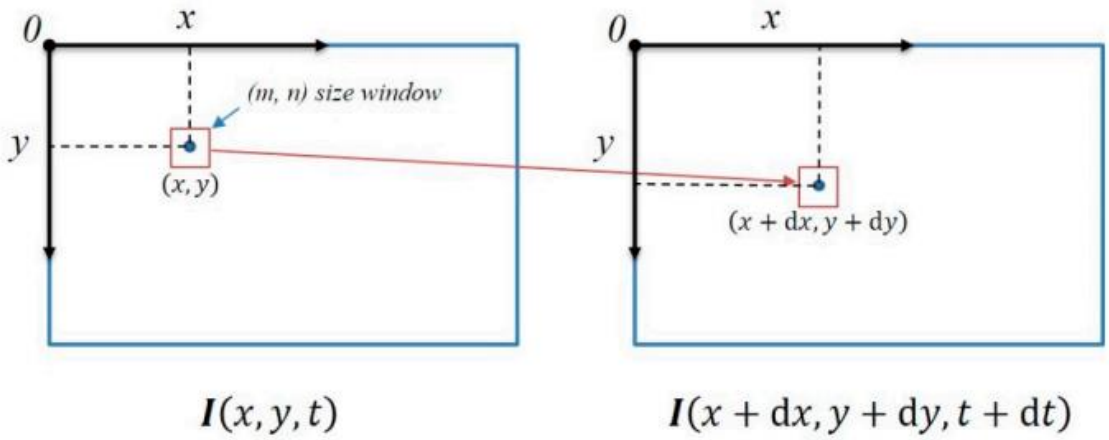

Figura 9. Diagrama del método de Lucas-Kanade. Obtenido de [7].

Para seleccionar puntos que sean fácilmente identificables por LK, conocidos como *features*, se buscan regiones donde existan variaciones notables, normalmente esquinas donde la derivada alrededor del punto cambie en distintas direcciones. Basado en el detector de esquinas de Harris, el detector de esquinas de Shi-Tomasi [8] incluye un criterio de selección distinto, basado en los eigenvalores (λ1 y λ2) de un conjunto de pixeles, que puede ser utilizado como una mejor entrada para la estimación del movimiento de estas *features* en imágenes consecutivas [9]. A la combinación de estos dos métodos se le conoce como el algoritmo de Kanade-Lucas-Tomasi (KLT).

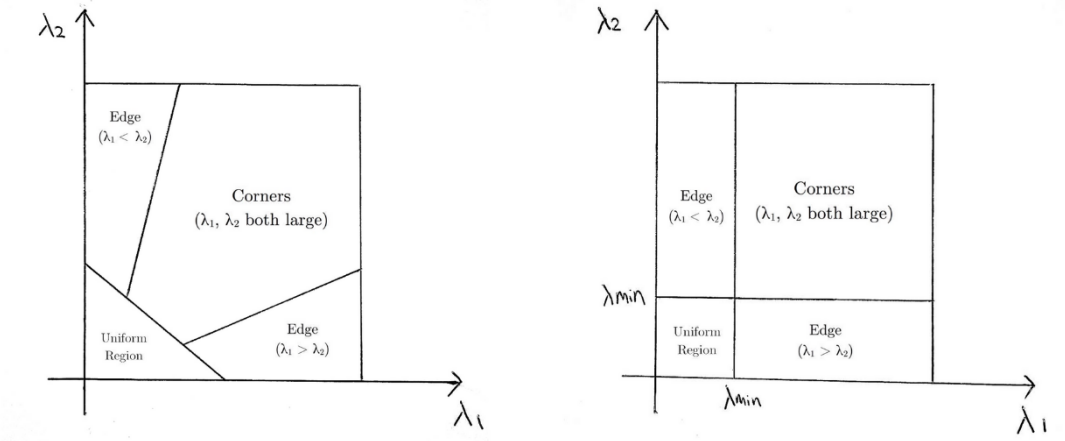

Figura 22. Comparación de criterios de selección de Harris y Shi-Tomasi. Obtenida de [9].

Cuando se captura una imagen utilizando una cámara *pinhole* se pierde información respecto a la profundidad de la imagen. La geometría epipolar describe la correspondencia entre las diferentes proyecciones de un objeto y nos permite estimar la profundidad a un factor de escala desconocido a partir de múltiples imágenes de una misma cámara en tiempo distintos. Para determinar el factor de escala es necesario introducir información

adicional a la proporcionada por la cámara, como pueden ser los datos de una IMU, o información geométrica previa de una escena conocida [10]. La matriz fundamental (F) es la representación algebraica de la geometría epipolar.

Para representar este concepto, en la figura (10) se muestran dos imágenes tomadas desde el centro de cámara C y C' respectivamente. Ambas imágenes proyectan un punto X en las coordenadas del mundo al sistema de coordenadas de la cámara en x y x'. El plano formado por todos los puntos de las posibles profundidades de x y x' es conocido como plano epipolar [11], y las intersecciones de dicho plano con los de la imagen se conocen como líneas epipolares, que restringen las posibles correspondencias de la proyección del punto x en la otra imagen y viceversa.

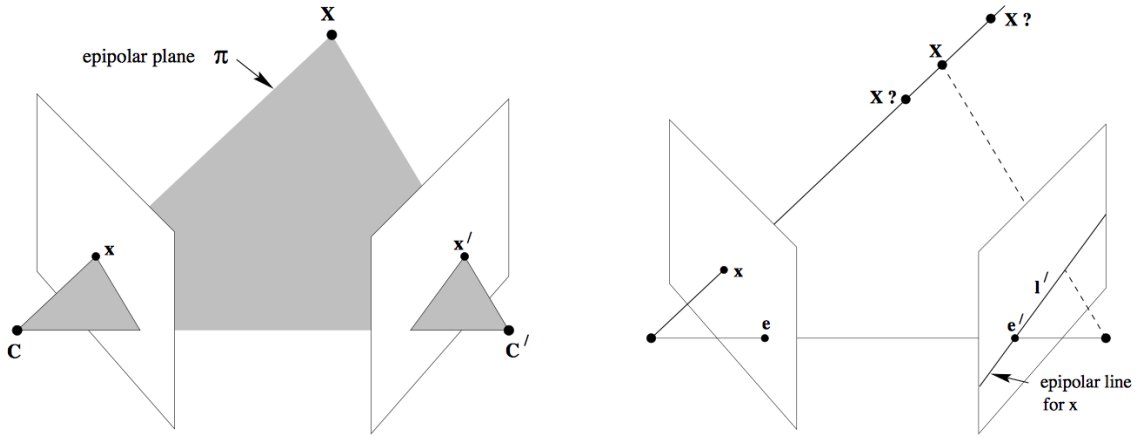

Figura 10. Correspondencia de un punto utilizando geometría epipolar. Obtenido de [11].

La matriz esencial (E) contiene información sobre la traslación y rotación entre los dos planos de la imagen en un espacio físico, y es puramente geométrica. La matriz F contiene esta información además de los parámetros intrínsecos de la cámara.

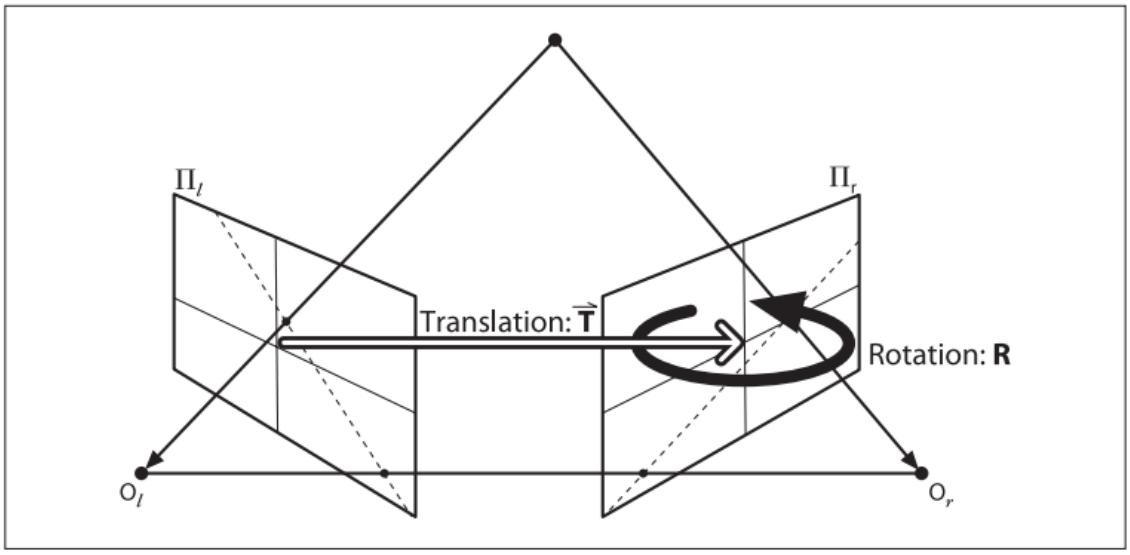

Figura. Descripción de la información obtenida con la matriz esencial. Obtenido de [10]

Existen dos métodos para extraer la estructura tridimensional de un par de imágenes [12]. En el primero es necesario calibrar la cámara o cámaras de cada uno de los puntos de vista respecto al sistema de coordenadas del mundo mediante la obtención de los parámetros extrínsecos de la cámara, calcular la matriz esencial a partir de la geometría epipolar y a partir de esta información extraer la matriz esencial y de ella obtener la estructura tridimensional de la escena.

El segundo método, a partir de un sistema no calibrado respecto a las coordenadas del mundo, la matriz fundamental es calculada a partir de correspondencias en los pares de imágenes, y a partir de ella se obtiene la matriz esencial con la información tridimensional de la escena.

Uno de los métodos más simples y computacionalmente económicos para obtener la matriz fundamental a partir de las correspondencias de puntos en un par de imágenes es con el algoritmo de 8 puntos [13], que como su nombre lo indica necesita al menos 8 correspondencias para resolver F como un sistema lineal de ecuaciones. Si más puntos son proporcionados se utilizan mínimos cuadrados. El problema con este algoritmo es que es extremadamente sensible a valores atípicos, incluso utilizando más de 8 puntos como entrada, por lo que técnicas adicionales para filtrar las correspondencias deben de ser utilizados.

Random sample consensus (RANSAC) [14], es un robusto método para remover valores atípicos, donde el problema es solucionado en múltiples ocasiones utilizando distintos subconjuntos de las muestras seleccionadas al azar, y después selecciona la solución particular que sea más cercana a la media de las soluciones.

Descripción del algoritmo:

A continuación, se describe la metodología utilizada para la estimación del movimiento de un vehículo a partir de una secuencia de imágenes obtenida por una cámara monocular.

Calibración de la cámara:

Este paso solo se realiza una vez para cada cámara nueva a utilizar. Se siguió el procedimiento descrito en [4].

Detección y rastreo de *features*

- Se obtiene la imagen I(t), cada que se obtiene una imagen esta se convierte a escala de grises y se corrige su distorsión mediante los parámetros intrínsecos de la cámara.

Figura 17. Imagen sin distorsión

- Se detecta un número predefinido N de *features* que son susceptibles a ser rastreadas mediante el método de Shi-Tomasi.

Figura 18. Features detectadas en I(t)

- Se obtiene la imagen I(t+1) y se rastrean las *features* obtenidas en I(t) con el método de Lucas-Kanade, conformando junto con la etapa de detección de *features* el método Kanade–Lucas–Tomasi. Se eliminan las correspondencias erróneas, que se encontraron fuera de la imagen o muy cerca de sus bordes.

Figura 19. *Features* de I(t) rastreadas en I(t+1)

- De las *features* restantes en I(t+1), se calcula un vector de dirección promedio respecto a su *feature* correspondiente en I(t), y se eliminan las correspondencias que no vayan en la misma dirección.
- Después de eliminar las correspondencias erróneas, se cuenta con N – n *features* rastreadas correctamente en I(t+1), donde n es el número de *features* que no pudieron ser rastreadas. Ahora I(t+1) toma el lugar de I(t) y se reponen las *features* faltantes con el método de Shi-Tomasi, cuidando que las nuevas se encuentren a mínimo 5 pixeles de las ya existentes, y el conjunto de puntos resultantes se utiliza para ser rastreado en I(t+2), actualizando la cuenta para cada iteración.

Figura 20. Features rastreadas en múltiples imágenes consecutivas.

Geometría epipolar

- El algoritmo de 8 puntos, como su nombre lo indica, requiere al menos 8 correspondencias para poder implementarse, por lo que, si no se tienen disponibles después de filtar los resultados de LK, toda esta sección no puede ser aplicada aún y pasamos a la siguiente imagen.
- Una vez cumplida esta condición, se toman las *features* que tuvieron correspondencia en los actuales I(t) e I(t+1), para obtener su matriz fundamental.
- Con la matriz fundamental se proyectan los puntos correspondientes a las *features* de I(t) en I(t+1).

Figura 20. Líneas epipolares resultantes de proyectar los puntos de I(t) en I(t+1)

- Se descartan todos los puntos que no cumplan con un máximo de distancia previamente especificado respecto a su correspondiente línea epipolar.

Figura 21. Líneas filtradas que cumplieron con el criterio máximo de distancia entre la línea y su punto.

- Si los puntos resultantes son mayores a 5, estos se utilizan en el algoritmo de 5 puntos, el cuál es una variante para estimar la matriz esencial en conjunto con el método iterativo RANSAC y los parámetros intrínsecos de la cámara. De no contar con los puntos suficientes, este paso se ignora y se obtiene la siguiente imagen.

- A partir de la matriz esencial es posible recuperar la matriz de rotación R y el vector de traslación T a un factor de escala desconocido mediante la técnica de *Single Value Descomposition*
- Es posible reconstruir una trayectoria de la posición y orientación del sistema $(Rpos, Tpos)$ a partir de cada R y T estimadas mediante las ecuaciones

$$Rpos = R * Rpos$$

$$Tpos = Tpos + T * Rpos$$

  Sin embargo, el factor de escala que traslada la trayectoria a coordenadas del mundo real sigue siendo desconocido a menos que se obtenga de alguna fuente externa de información.
- Este proceso se repite para cada nueva imagen obtenida.

Referencias

[1] Y. D. H. L. Dingfu Zhou, «Reliable scale estimation and correction for monocular Visual Odometry,» *IEEE Intelligent Vehicles Symposium (IV),* pp. 490-495, 2016.

[2] J. P. David A. Forsyth, Computer Vision, A Modern Approach, Prentice Hall, 2003.

[3] OpenCV, «Camera Calibration and 3D Reconstruction,» 2014. [En línea]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html. [Último acceso: 22 Noviembre 2019].

[4] OpenCV, «OpenCV Documentation,» 24 Noviembre 2019. [En línea]. Available: https://docs.opencv.org/master/d4/d94/tutorial_camera_calibration.html. [Último acceso: 24 Noviembre 2019].

[5] OpenCV, «OpenCV Documentation,» 26 Noviembre 2019. [En línea]. Available: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html. [Último acceso: 26 Noviembre 2019].

[6] S. J. Miller, «Williams College,» 2007. [En línea]. Available: https://web.williams.edu/Mathematics/sjmiller/public_html/BrownClasses/54/handouts/MethodLeastSquares.pdf. [Último acceso: 26 Noviembre 2019].

[7] J.-C. &. K. S.-D. Piao, «Adaptive Monocular Visual–Inertial SLAM for Real-Time Augmented Reality Applications in Mobile Devices,» *Sensors. 17. 2567. 10.3390/s17112567. ,* vol. 11, nº 17, 2017.

[8] U. Sinha, «AI Shack,» 2017. [En línea]. Available: http://aishack.in/tutorials/shitomasi-corner-detector/. [Último acceso: 23 Marzo 2019].

[9] C.-e. Lin, «NanoNets, Machine Learning API,» 23 Abril 2019. [En línea]. Available: https://nanonets.com/blog/optical-flow/. [Último acceso: 26 Noviembre 2019].

[10] A. K. Gary Bradski, Learning OpenCV, Sebastopol: O'Reilly, 2008.

[11] S. Kapoor, «Sanyam Kapoor,» Courant Institute, 8 Agosto 2017. [En línea]. Available: https://www.sanyamkapoor.com/machine-learning/an-introduction-to-epipolar-geometry/. [Último acceso: Noviembre 28 2019].

[12] R. Owens, «School of Informatics official page,» The University of Edinburgh , 29 Noviembre 1997. [En línea]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT10/node3.html. [Último acceso: 28 Noviembre 2019].

[13] M. G. S. S. D. Randeniya, «Fusion of vision inertial data for automatic geo-referencing,» *Knowledge Discovery From Sensor Data,* pp. 106-127, 2008.

[14] M. A. Fischler y R. C. Bolles, «Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,» *Communications of the ACM,* vol. 24, nº 6, pp. 381-395, 1981.

[15] OpenCV, «Camera Calibration and 3D Reconstruction,» 2014. [En línea]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html. [Último acceso: 22 Noviembre 2019].