

Prueba y Resultados: Auto-calibración de cámara usando superficies planares

Liz Lima *Universidad Católica San Pablo*
Arequipa, Peru
liz.lima@ucsp.edu.pe

Josué Ttito *Universidad Católica San Pablo*
Arequipa, Peru
josue.ttito@ucsp.edu.pe

18 de febrero de 2019

Resumen

Realizar una calibración de cámara requiere usualmente de un patrón que pueda ser fácilmente reconocido y procesado. Sin embargo, no siempre se encuentran patrones en cualquier momento, mientras que sí sucede con cualquier superficie planar. En el presente documento, se presentan los resultados obtenidos a partir de un método de auto-calibración de cámara utilizando una superficie planar con una textura desconocida. La precisión obtenida es equivalente a una realizada con la calibración usando el patrón del tablero de ajedrez.

Calibración de cámara, patrón de ajedrez, superficie planar, visión computacional.

1. Introducción

La calibración de cámaras y la obtención de los parámetros intrínsecos y extrínsecos es un problema fundamental en el área de visión computacional para tareas tales como reconstrucción 3D, navegación de robots, realidad aumentada, etc. Ankur [1] presenta dos categorías para los métodos actuales de calibración. El primero que es realizada sobre escenarios en los cuales se conoce la geometría 3D, y el segundo es que es realizado de manera automática utilizando un patrón fácil de reconocer (círculos, tablero de ajedrez, anillos, etc).

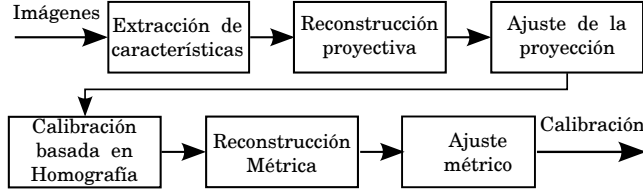


Figura 1: Proceso de auto-calibración usando superficies planares.

El artículo continúa de la siguiente manera: En la sección 2 se resume el proceso en el artículo. En la sección 3 se describe las pruebas realizadas y los resultados obtenidos, y por último en la sección 4 se presentan las conclusiones.

2. Sistema de calibración planar

2.1. Modelo de cámara

El modelo de cámara usado es el conocido como cámara estenopeica (*pin-hole camera* en inglés). Luego se necesita una función $\mathcal{P}(x)$ que transforma un punto en el espacio 3D al espacio 2D. La función de proyección está compuesto por tres funciones: la transformación extrínseca \mathcal{T} , la transformación intrínseca \mathcal{K} y la función de distorsión \mathcal{D} , $\mathcal{P}(x) = \mathcal{D} \cdot \mathcal{K} \cdot \mathcal{T}(x)$

2.2. Reconstrucción Proyectiva

El primero paso del algoritmo es la obtención de la reconstrucción proyectiva de la escena. La reconstrucción incluye la posición de los puntos observados que pueden ser representados como un vector en dos dimensiones.

Los puntos observados son obtenidos usando el algoritmo de *Oriented FAST and BRIEF* [3], el cual es un descriptor y es más conocido como ORB. El descriptor es una fusión de los métodos *FAST* y *BRIEF*.

La reconstrucción proyectiva se realiza usando homografía, para ellos se toma el fotograma 0 como la imagen de referencia. Las posiciones de los demás fotogramas son obtenidos calculando la homografía entre el fotograma de referencia y el fotograma seleccionado.

Los resultado obtenidos de ésta re-proyección no son óptimos por dos motivos: Primero, la homografía obtenida no considera los demás frames. Segundo, la distorsión no ha sido obtenida aún. Así que para mitigar éstos

errores el autor utiliza una función no lineal de minimización sobre todos los parámetros obtenidos.

2.3. Calibración basada en homografía

La entrada para ésta etapa sera todo el conjunto de Homografías obtenidas H_1 en el paso anterior. La calibración recupera los parámetros intrínsecos f_x, f_y , y p_0 .

Se describe la escena planar en el fotograma de referencia que se considera con un vector de norma n_0 . Junto a ello se encuentran dos vectores a_0 y a_1 que describen todo el espacio 3D.

Los parámetros intrínsecos luego son obtenidos de manera iterativa, considerando un $K = (f, f, 1)$ inicial, debido a que el fotograma inicial se considera fronto-paralelo al plano.

2.4. Reconstrucción métrica

Una vez que los parámetros intrínsecos han sido recuperados, se requiere una actualización de la reconstrucción proyectiva. Como, en el paso anterior se obtiene los vectores a , b , que abarca todo el espacio, se inicializa $x = [x, y, 0]$ considerando nuevamente que el fotograma inicial es fronto-paralelo respecto al plano.

3. Pruebas y Resultados

Las pruebas fueron realizadas con dos cámaras: una cámara web HD, y una cámara de celular en 4K.

Se han grabado dos videos con cada cámara y se han obtenido los parámetros intrínsecos y extrínsecos.



Figura 2: Secuencia de los fotogramas del vídeo 1 capturado con la cámara web.

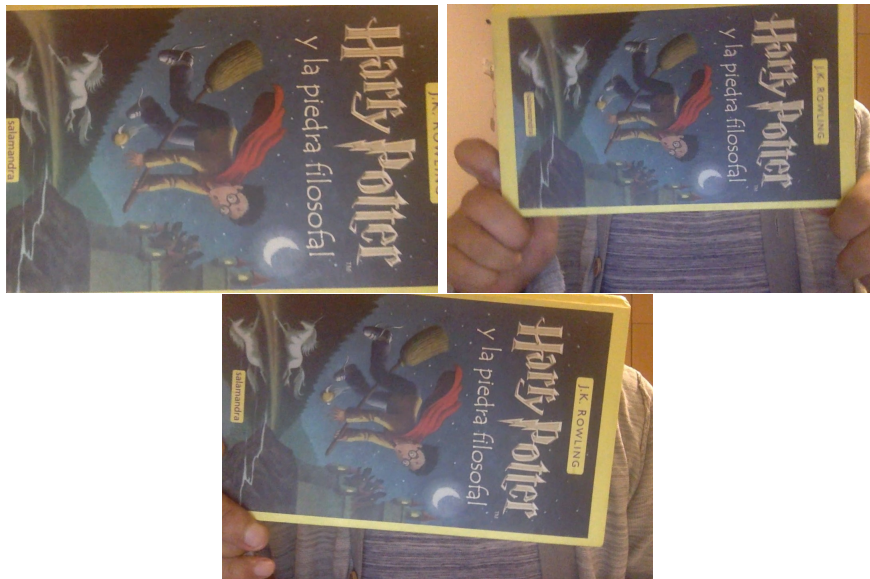


Figura 3: Secuencia de los fotogramas del vídeo 2 capturado con la cámara web.

Los resultados obtenidos son mostrados en la tabla 1. Como se puede ver los puntos focales obtenidos son muy similares cuando se comparan las mismas cámaras.

		FP	PP
Vídeo 1	Webcam	539.08; 539.08	270; 180
	Mobile	483.72; 470.49	308.71; 170.63
Vídeo 2	Webcam	508.49; 508.49	270; 180
	Mobile	478.40; 478.40	320; 180

Cuadro 1: Tabla de resultados de los vídeos 1 y 2, obtenidas con ambas cámaras.

Los puntos característicos usados en cada vídeo son mostrados en las figuras 4 y 5.

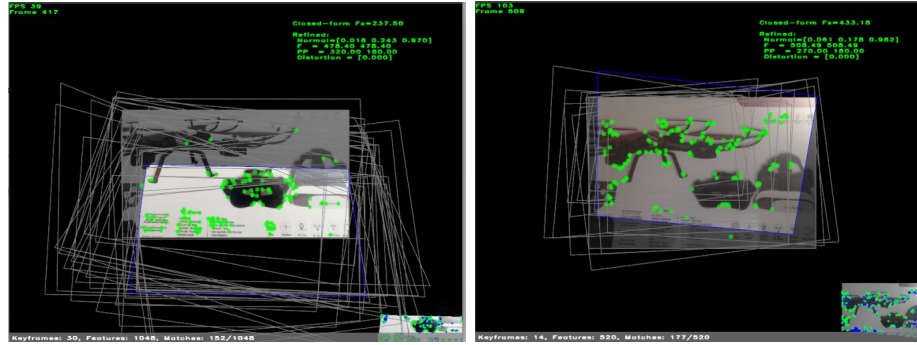


Figura 4: Puntos característicos usados para realizar la calibración de cámara en el vídeo 1.

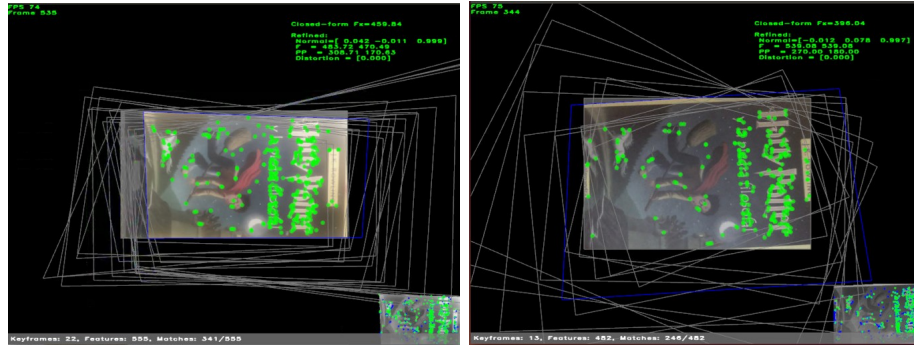


Figura 5: Puntos característicos usados para realizar la calibración de cámara en el vídeo 2.

4. Problemas y conclusiones

Como se puede ver en las pruebas realizadas, la calibración de cámara, puede ser obtenida con superficies planares. Sin embargo, existen ciertos problemas que a continuación detallamos:

1. A pesar de que existen muchas superficies planares, no todos son fáciles de reconocer por el algoritmo.
2. Para obtener una buena calibración, el número de puntos característicos debe ser suficientemente grande. Por ejemplo, una superficie que no tenga demasiado detalle no permitirá una buena calibración.
3. Las condiciones de luz y resolución de la cámara influyen en la obtención de los puntos característicos. Lo cual percute en la calibración.
4. Colocar el fotograma 0 como el fotograma de referencia, obliga que la calibración empiece con la superficie cubriendo toda la cámara.

Por último para poder compilar el código fuente, se han realizado cambios que han sido subidos al siguiente repositorio: <https://github.com/joshjo/planecalib>. Los cambios fueron aceptados en el repositorio oficial del autor.

Referencias

- [1] Datta, Ankur Kim, Jun-Sik Kanade, Takeo. (2009). Accurate Camera Calibration using Iterative Refinement of Control Points. 1201 - 1208. 10.1109/ICCVW.2009.5457474.

- [2] Herrera, D., Kannala, C. J., Heikkila, J. (2016, March). Forget the checkerboard: Practical self-calibration using a planar scene. In 2016 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1-9). IEEE.
- [3] Rublee, E., Rabaud, V., Konolige, K., Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF.