

Calibración de Cámara Basado en un Patrón de Anillos

Leticia Laura Ochoa, Mg¹, Maribel Molina Barriga, Mg²

¹Universidad Nacional de San Agustín, Perú, letymarisol.laura@gmail.com, maribelmolina22@hotmail.com

Resumen— *La calibración de cámara es muy utilizado en los campos de visión computacional, realidad aumentada, robótica y reconstrucción en 3D por lo que su estudio se hace necesario. Existen diferentes trabajos de calibración de cámaras basados en patrones de planillas planas de cuadrículas, círculos y anillos. En este trabajo se realiza la investigación de los pasos necesarios previos para la calibración de cámara basado en el patrón de anillo, el cual requiere un pre procesamiento de los datos de las imágenes obtenidas a partir de la extracción de frames de un vídeo. Se utilizó binarización adaptativa para mejorar la detección de contornos de las imágenes.*

I. INTRODUCTION

La calibración de cámara es un paso necesario en visión computacional 3D para extraer información métrica de imágenes 2D [1]. La calibración de cámara sirve para modelar computacionalmente la cámara.

La calibración de una cámara representa un paso importante para procesos de reconstrucción 3D o sensorización mediante una cámara. Con la calibración de la cámara se consiguen estimar los parámetros intrínsecos y extrínsecos de la misma los cuales son necesarios para realizar la reconstrucción 3D del entorno y situar la cámara en el mismo [2].

La calibración de cámara es un problema importante en la cual inclusive pequeñas mejoras son beneficiosas para tareas como la reconstrucción 3D, navegación de robots, etc. [3]. Zhang [1] propone una técnica de calibración basada en la observación de una plantilla plana desde varias posiciones.

El proceso de calibración consiste en estimar tanto la posición y orientación de la cámara en la escena, así como sus características intrínsecas. La robustez de la calibración depende de la calidad de las medidas, el modelo que se calcula, el método de calibración y las condiciones bajo las que se realiza [4].

La calibración de cámara se utiliza ampliamente en los campos de visión computacional, robótica y reconstrucción en 3D. La calibración de la cámara es el primer paso para extraer datos 3D de una imagen 2D. Es el proceso en el que se extraen los parámetros ópticos de la cámara (intrínsecos) y la posición de la cámara (extrínseca), ya sea absoluta o con respecto al objeto de interés. La calibración juega un papel crucial en visión computacional y reconstrucción en 3D, ya que la precisión de la reconstrucción y la determinación de

coordenadas tridimensionales dependen en gran medida de la precisión de la calibración de la cámara [5].

En este trabajo se investiga la implementación de algoritmo de Ankur [6] para realizar la calibración de cámara basado en el patrón de anillos.

Los parámetros de calibración internos y externos (modelo de píxeles para círculos/anillos) se obtienen a partir de un patrón de anillos concéntricos con imágenes, cuyo contorno aparente también se explota para obtener una mejor precisión de calibración.

II. PROCEDIMIENTO PARA EL MÉTODO PROPUESTO

A. Preparación de los patrones

Para la preparación de los patrones se utiliza un patrón plano de pares de círculos concéntricos.

En el presente trabajo se extrajo las imágenes a partir de un vídeo con fotografías del padrón de anillos con diferentes posiciones. Al menos 7 imágenes son necesarias, mientras más imágenes de diferentes orientaciones se consideren la calibración es mejor. Esta es la única parte manual que se realiza en los procesos de calibración. Después de esto, los siguientes pasos serán completamente automatizados.

B. Transformación de coordenadas

El modelo de cámara nos permite calcular las coordenadas de un objeto en la imagen en función de las coordenadas en el mundo real. Ejemplo simplificado de una proyección perspectiva (sólo coordenada X)

C. Parámetros internos y externos de la cámara

Estos parámetros son hallados calculando la matriz de proyección. Los valores intrínsecos requieren que la imagen no este distorsionada, y esto lo logramos mediante los parámetros de distorsión. Al corregir la distorsión se obtiene una imagen con perspectiva real, y como resultado de ello, es posible determinar los 5 valores intrínsecos, correspondientes a coordenadas x e y del centro de la proyección, la distancia focal, el aspecto ratio y el ángulo entre los ejes ópticos. Por otra parte los parámetros extrínsecos representan la rotación y traslación en el espacio, asimismo describen la posición y dirección de la cámara en el mundo. La rotación está representada por una matriz de 3×3 , mientras que la traslación por un vector 3×1 . La importancia de la matriz extrínseca de valores describe como el mundo es transformado en relación a la cámara

III. PRE PROCESADO DE LAS IMÁGENES

En la Fig. 1 se muestra los pasos seguidos para el pre procesamiento de las imágenes, los cuales son necesarios para la posterior detección de contornos circulares o elípticos y luego poder hallar el centro de cada anillo.

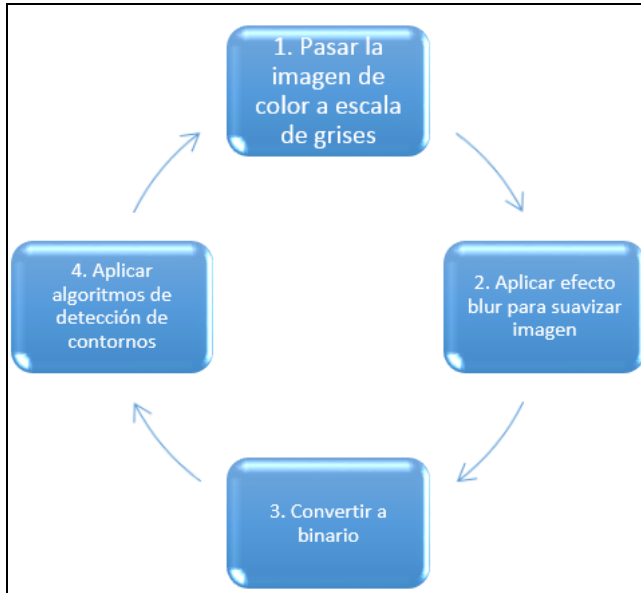


Fig. 1 Pre procesamiento de las imágenes

III. INDICACIONES ÚTILES

A. La detección de los bordes

En el área de procesamiento de imágenes, la detección de los bordes de una imagen es de suma importancia y utilidad, pues facilita muchas tareas, entre ellas, el reconocimiento de objetos, la segmentación de regiones, entre otras. El algoritmo de Canny consiste en tres grandes pasos:

- Obtención del gradiente: En este paso se calcula la magnitud y orientación del vector gradiente en cada píxel.
- Supresión no máxima: En este paso se logra el adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un píxel de ancho.
- Histéresis de umbral: En este paso se aplica una función de histéresis basada en dos umbrales; con este proceso se pretende reducir la posibilidad de aparición de contornos falsos.

IV. PRUEBAS Y RESULTADOS

Se probó la apertura de la cámara para capturar las imágenes utilizando Visual Studio y OpenCV, el código se muestra en la Fig. 2.

```
#include<opencv\cv.h>
#include<opencv\highgui.h>
using namespace cv;
int main()
{
    Mat image; //Crear una matriz para almacenar la imagen
    VideoCapture cap; //Inicializar la captura
    cap.open(0);

    namedWindow("Windows",1); //crear la ventana para mostrarla imagen
    while(1)
    {
        //copiar la secuencia de la cámara web a la imagen
        cap>>image;
        //imprimir imagen en pantalla
        imshow("window",image);
        //retrasar 33 ms
        waitKey(33);
    }
    return 0;
}
```

Fig. 2 Código de apertura de cámara

Se utilizará el patrón plano de pares de círculos concéntricos que se muestra en la Fig. 3. La posición del círculo centro y el radio de cada círculo es conocido. Al menos se necesitan dos pares, se debe tomar imágenes de este patrón plano, al menos 7 imágenes son necesarias.

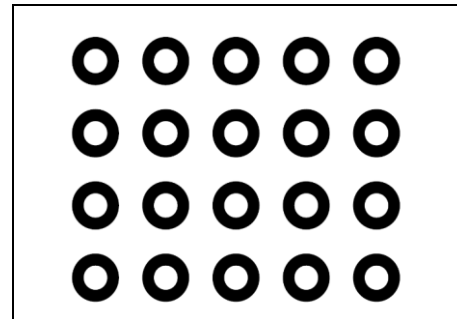


Fig. 3 Patrón de anillo

En la Fig. 4 se muestra el código utilizado para la captura de frames a partir del video llamado "PadronAnillos_02.avi", con la cual se obtuvieron 1964 frames.

```
int main()
{
    std::vector<int> compression_params;
    compression_params.push_back(CV_IMWRITE_JPEG_QUALITY);
    compression_params.push_back(100);
    Mat imagen;
    VideoCapture video("PadronAnillos_02.avi");
    int count = video.get(CV_CAP_PROP_FRAME_COUNT);
    bool guardarFrame;
    std::string filePath = "E:\\FramesVideo\\Frame";
    std::stringstream saveFrame;
    for (int frameNum = 0; frameNum < count; frameNum++)
    {
        video >> imagen;
        namedWindow("Video", 1);
        imshow("Video", imagen);
        saveFrame << filePath << frameNum << ".jpg";
        guardarFrame = imwrite(saveFrame.str().c_str(), imagen, compression_params);
        saveFrame = std::stringstream();
    }
    cvWaitKey();
    return 0;
}
```

Fig. 4 Código para captura de frames del video

En la Fig. 5 se muestra los resultados obtenidos de la conversión a escala de grises a partir del frame del video.

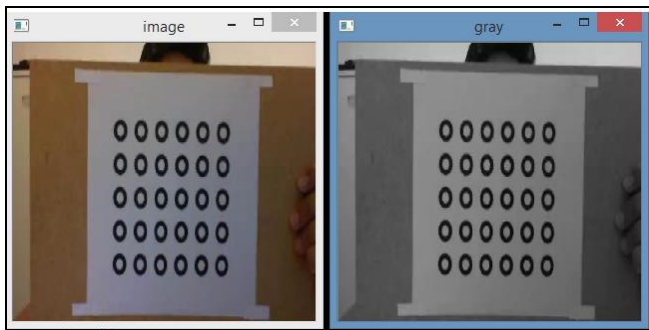


Fig. 5 Conversión de la imagen a escala de grises

En la Fig. 6 se muestra a la izquierda la aplicación de filtro gaussiano para suavizar la imagen y a la derecha la conversión a binario de la imagen utilizando la función `cvThreshold` de OpenCV. Se puede observar que el resultado de la binarización no es buena, esto debido a que en las imágenes pueden tener variaciones de iluminación.

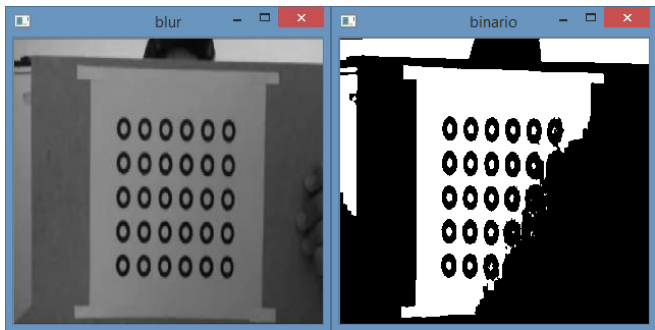


Fig. 6 Efecto blur y conversión a binario

Debido al problema anterior de la binarización, se probó la función `cvAdaptiveThreshold` de OpenCV, cuyo resultado se muestra en la Fig. 7, donde se puede notar una mejora en la conversión a binario.

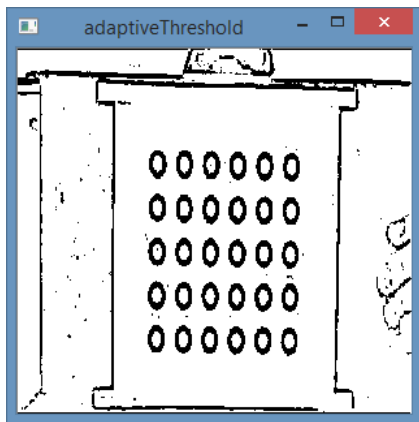


Fig. 7 Conversión a binario utilizando la función `cvAdaptiveThreshold`

En la Fig. 8 se observa que la detección de contornos a partir de la última imagen binarizada presenta algunos ruidos.

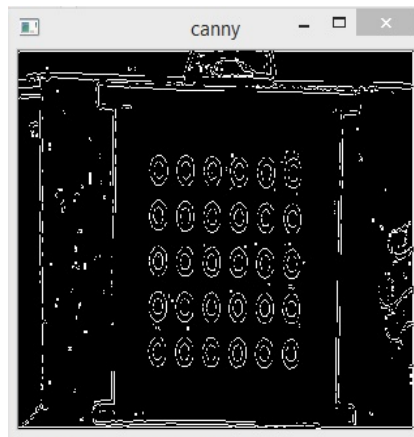


Fig. 8 Detección de contornos con ruido

En la Fig. 9 se muestra el resultado de la aplicación de la técnica de Bradley [7] para la binarización de la imagen, con la cual se mejoró la detección de contornos (Fig. 10) para la calibración de cámara utilizando el padrón de anillos a implementar.

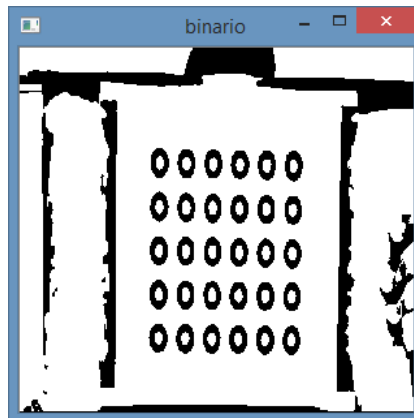


Fig. 9 Conversión a binario utilizando la técnica de Bradley

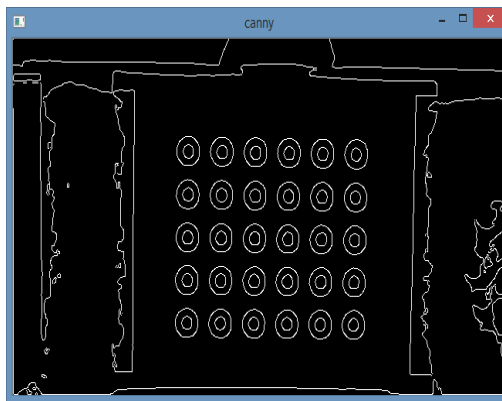


Fig. 10 Detección de contornos corregido

IV. CONCLUSIONES

Se realizó la captura de frames a partir de uno de los videos de anillo proporcionado, conversión a escalas de grises, efectos de filtro gaussiano y binarización de imágenes. También se investigó los pasos necesarios para la calibración basado en el patrón de anillos y formas de detectar los contornos de los círculos y elipses para luego hallar los puntos centrales de los mismos.

REFERENCES

- [1] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [2] C. Ricolfe, A. J. Sánchez and R. Simarro, "Técnicas de calibrado de cámaras", *Journal de Mathematiques Pures et Appliques*, Leon, 2003.
- [3] O. Faugeras and G. Toscani. The calibration problem for stereo. In *CVPR*, 1986.
- [4] C. Ricolfe and A. J. Sánchez, "Procedimiento completo para el calibrado de cámaras utilizando una plantilla plana", *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 5, no. 1, pp. 93-101, 2008.
- [5] C. D. Prakash and L. J. Karam, "Camera calibration using adaptive segmentation and ellipse fitting for localizing control points", In *Image Processing (ICIP), 19th IEEE International Conference on. IEEE*, pp. 341-344, 2012.
- [6] A. Datta, J. S. Kim, and T. Kanade, "Accurate camera calibration using iterative refinement of control points," in *Computer Vision Workshops (ICCV Workshops), IEEE 12th International Conference on. IEEE*, pp. 1201–1208, 2009.
- [7] D. Bradley and G. Roth, "Adaptive Thresholding Using the Integral Image", *Journal of graphics tools*, vol. 12, no 2, pp. 13-21, 2007.