

Calibración de una cámara usando un padrón en forma de anillo

Alejandra Callo¹ and Jose Jaita²

Abstract—En este trabajo se propone un método para la calibración de cámara usando un padrón de anillos, posteriormente se compara los resultados usando otros padrones como el chessboard y el de symmetric circles. Una parte del pipeline es la detección de anillos, para lo cual se propone un enfoque simple, robusto y rápido para dicha tarea.

I. INTRODUCCIÓN

La calibración geométrica de la cámara, también conocida como resección de cámara, estima los parámetros de una lente y un sensor de imagen de una imagen o cámara de video. Puede usar estos parámetros para corregir la distorsión de la lente, medir el tamaño de un objeto en unidades mundiales o determinar la ubicación de la cámara en la escena. Estas tareas se usan en aplicaciones tales como visión artificial para detectar y medir objetos. También se usan en robótica, para sistemas de navegación y reconstrucción de escenas tridimensionales.

El trabajo basicamente consta de detectar los anillos, luego con los puntos de control (centro de los anillos) se pasa a la parte de la calibración usando OpenCV [1] para dicha tarea.

El enfoque que se propone para la detección es simple, debido a que no se usan heurísticas complejas, esto conlleva a que sea muy rápido y pueda correr en tiempo real.

II. DETECCIÓN DEL PADRÓN

II-A. Detección de contornos

Primero, para la detección de los anillos, la imagen de entrada se convierte a gray, luego se le aplica una binarización usando el método de Otsu[2], para esto usamos la función de OpenCV "threshold", con la particularidad que le decíamos que nos calcule un threshold adecuado usando el método de Otsu, se pudo haber elegido manualmente un threshold, pero debido a la variación de iluminación, esta etapa no hubiese sido robusta. Luego tenemos una imagen binaria a la cual le aplicamos la función "findContours", entonces tenemos todos los contornos de los anillos pero también de otros objetos.

II-B. Filtros

Esta etapa consiste en remover todo el ruido de las imágenes, es decir remover todos los contornos que no formen parte de los anillos, el primer filtro que se usa es en cuanto a la relación de ancho y altura, debido a que es un círculo, la relación de tales parámetros debería estar cerca a 1, en

la práctica se comprobó que dicha relación fue hasta un límite de 0.4, esto se debió a la perspectiva en que se tomaba la imagen. El segundo filtro era el área que formaba el contorno detectado, para el video de Kinetic2 se tuvo áreas que fueron desde 1000 hasta 9000 píxeles, el tercer filtro fue escoger solamente los contornos que sean padres, como se sabe el anillo tiene dos contornos, el círculo exterior y el interior, entonces al usar la función "findContour" se obtenía la jerarquía de los contornos encontrados, entonces al tener dicha información se filtraba todo contorno que no sea padre. Como se menciona los parámetros de relación y área de los contornos eran variables, entonces dichos parámetros se actualizaban con cada imagen de entrada.

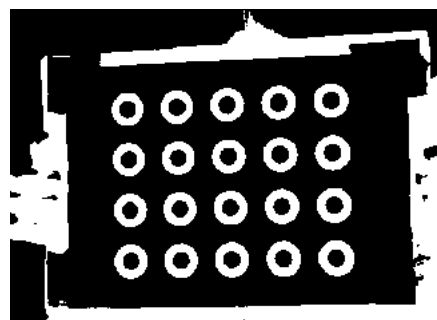


Fig. 1. Imagen calibrada, usando lifecam

III. REGIÓN DE INTERÉS (ROI)

La eficacia del método Otsu para hallar un threshold adecuado, depende mucho a que región de la imagen se aplique, lo ideal es que solamente se use la zona en donde se encuentran los anillos.

III-A. ROI

Una vez detectado todos los anillos se define un roi, para esto simplemente hallamos la ubicación del anillo que se encuentra más cerca al origen y el más lejano, de este modo se define el roi, pero debido a que dicho roi se aplicará en el siguiente frame, y los anillos podrían cambiar de posición, entonces se le aplicaba un pequeño padding de 10% del tamaño del roi, con tal ajuste era suficiente ya que entre frame y frame no hay mucha variación ahora, si subimos a porcentaje, se corre el riesgo de que el método de Otsu use píxeles irrelevantes haciendo que no de un buen threshold. En la Fig 2. se muestra el roi de color celeste.

IV. TRACKING

IV-A. Transformación

Para la calibración es necesario hacer el tracking de los anillos detectados, cuando hallamos los contornos en la

*Este trabajo es patrocinado por CONTYTEC

¹A. Callo, estudiante de la maestría en ciencias de la computación, Universidad Católica San Pablo, alejandra.callo@ucsp.edu.pe

²P. J. Jaita, estudiante de la maestría en ciencias de la computación, Universidad Católica San Pablo, jose.jaita@ucsp.edu.pe

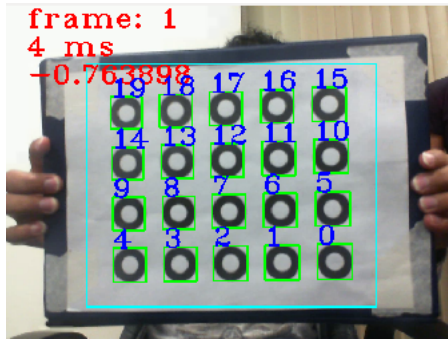


Fig. 2. Imagen calibrada, usando lifecam

etapa anterior, si la imagen se encontraba sin ningun tipo de rotación, nosotros ya teniamos los anillos ordenados, la funcion "findconoturs" devuelve los contorno con respecto al eje y. Los problemas comenzaban cuando se rataba la imagen entonces lo que se hizo fue una tranformacion de espacio, simplemente rotabamos la imagen y en esta nueva imagen encontrabamos los contornos, de esta forma teniamos el orden correcto de los contorno en la imagen original. Para que la rotación funcionase se debía conocer el angulo correcto, entonces nosotros actualizabamos constantemente dicho angulo, el cual era hallado usando la funcion de arcotangente entre la posicones del anillo 0 y el 1, obviamente el siguiente frame puede ser que cambie el angulo entonces no se tendra una ratación perfecta, ya que en cada frame se usa el angulo hallado con los anillos del frame anterior , entonces lo que se hace es llamar dos veces a la funcion de tranformacion, primera con el angulo pasado, luego hallabamos el nuevo angulo, de nuevo llamamos a la tranformación , entonces se tenia una tranformacion perfecta.

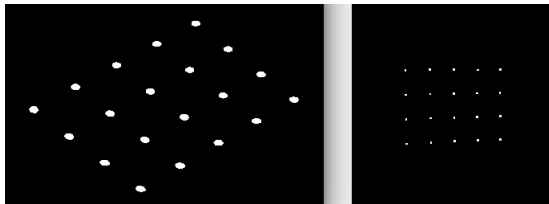


Fig. 3. Parte izquierda, la imagen original, lado derecho imagen rotada

En la Fig 3. vemos el resultados de transformar una imagen hacia otro espacio, en donde tenemos perfectamente ordenados lo anillos.

IV-B. Puntos de control

La calibración depende exclusivamente de cuan preciso calculemos los centros de los anillos (puntos de control), una vez detectado el anillo volviamos a la imagen original para no perder informacion ya que , todos los metodos aplicados anteriormet se aplicaba al roi entero , el cual mete ruido , perdiendo detalles de los anillos ,por lo tanto para detectar el centro del anillo se trabaja solo en la region del contorno. Se aplica el metodo de Otsu para binarizar luego se encuentra los contornos, en este caso obteniamos

solamente 2 contorno el del anillo exterior y el del interior, luego hallamos el centro de masa del objeto formado por cada contorno, en nuestros experimetos los centros de los dos círculo diferian por poco, menos de un pixel en promedio, entonces simplemente hallabamos el promedio de los dos, y este vendria hacer nuestro punto de control para nuestra calibración.

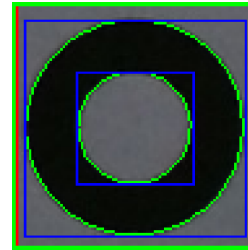


Fig. 4. Anillo segmentado

[1206.02, 656.321]	[1205.97, 655.897]
2	2
[1084.8, 621.017]	[1084.88, 620.831]
2	2
[967.232, 587.129]	[967.041, 586.782]
2	2
[852.644, 553.738]	[852.473, 553.547]
2	2
[1232.72, 541.848]	[1232.76, 541.443]
2	2
[741.03, 520.803]	[741.055, 520.456]

Fig. 5. Diferencia de centros de masa, izquierda exterior, derecha interior

En la Fig 4. se muestra un anillo segmentado con los bordes dectados, ahora se necesita el punto de control, en la Fig 5. se muestra la diferencia entre los centros de masa calculados para el círculo exterior y el interior. Por ejemplo [1206.02, 656.321] es la ubicacion del círculo exterior y [1205.97, 655.897] del círculo interior para el anillo de la fig 1. Los resultados anteriores fue par un frame relativamente bueno, en cambio cuando se calculo el centro de masa para otros frames en donde habia cierto blur debido al movimiento, la diferencia llego hace de hasta 1.2pixeles.

V. CALIBRACIÓN

Una vez hallado los 20 anillos pasamos a la calibración usando la función de OpenCV "calibratecamera"[1], a dicha funcion debiamos pasarle los puntos de control de la imagen (en nuestro caso los centros de los anillos) y las coordenadas de los objetos, en este caso seria la ubicacion de los puntos fisicos, para esto se uso como referencia a la imagen mas no a la cámara, entonces para nuestros puntos fisicos nuestro eje de coordenadas es el tablero del padrón , y ya que cada punto de control esta distribuido uniformemente no era necesario medir distancia.

V-A. Parámetros intrínsecos

La preposion anterior era valida debido a que solo estamos hallando los parametros intrinsecos, basicamente en la calibracion se halla la matriz de la cámara, la cual contiene

las distancia focales f_x y f_y además de los centros opticos c_x y c_y expresados en pixeles. Pero la funcion de calibracion no solo calculaba eso, sino que además nos dio los coeficiente de distorsión radial.

VI. EXPERIMENTOS

Para nuestros experimentos usamos dos camaras (Playstation PS3 y lifecam), para la calibración de dichas camaras se uso 3 padrones:

- Chessboard: los puntos de control eran hallados usando la funciones de Opencv "findChessboardCorners".
- Symmetric circle: usamos la funcion de OpenCV "findCirclesGrid".
- Rings: para los anillos usamos nuestro metodo.

TABLE I
RMS

Camera	Chessboard	Circle	Ring
LifeCam	0.254622	0.233688	0.227625
PS3	0.345601	0.261889	0.185876

En la tabla 1 se muestra los resultados llevados a cabo, usando tres patrones , dos cámaras y 20 imagenes por cada video, como se puede ver los mejores resultados fueron obtenidos usando el padrón de anillos.

Como se sabe un parámetro para saber cuan bien se calibro es el RMS, pero no solo es ese, sino que además de calcular dicho parámetro. se necesita ver como actua la calibracion en imagenes, en nuestros experimentos se probó muchas calibraciones, y cuando se aplicaba la corrección usando su respectiva matriz de la camara y los factores de distorsión a veces se obtenia una imagen concava y a veces convexa. En la figura se muestra el resultado de la calibracion usando el PS3, como se observa antes de la calibracion las lineas tendian a curvarse pero con la rectificacion dichas lineas se vuelven mas rectas.

VII. RESULTADOS



Fig. 6. Imagen sin calibración, tomada con PS3

VIII. CONCLUSIONES

El método propuesto para la detección de los anillos es robusto y rapido, corre en tiempo real, es mas puede correr a 60fps, usando las imagenes de la camara lifescan le tomaba entre 5 a 10ms hacer el procesamiento. El que los anillos



Fig. 7. Imagen calibrada, tomada con PS3

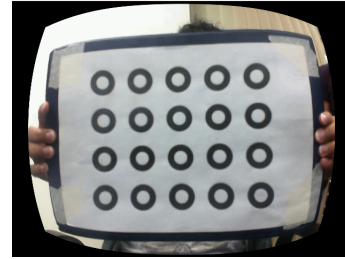


Fig. 8. Imagen calibrada, usando lifecam

tenga dos círculos ayuda mucho a la detección, facilmente se elimina falsos negativos. En cuanto a la calibración se escogio 20 imagenes por video, fue con el que nos dio los mejores resultados, la tabla 1 son los mejores resultados obtenidos despues de haber corrido varios experimentos con diferentes numero de imagenes. Tanto el padrón circular como el de anillos muestra una mejora con respecto al clasico chessboard, con esto se concluye que el padron en forma de anillo es el mejor para la calibración. El motivo que nosotros pensamos es porque los círculos son invariantes a la rotación, aunque con cierta perspectiva el círculo tiende a convertirse en elipse, pero aun asi se conserva el mismo punto de control, ya que lo que se halla es el centro, en cambio en el chessboard cuando hay rotacion mas un cambio de perspectiva se obtienen puntos de control diferentes. Se concluye que el parametro principal para que la calibracion sea buena, son los puntos de control calculados, estos deben ser lo mas preciso posible, tambien ayuda el numero de imagenes y cuantos puntos de control se utilizan por cada imagen, por ejemplo si hay un error en un punto y si los demas estan bien , entonces el error se reduce.

IX. TRABAJOS FUTUROS

Para el siguiente trabajo se mejorara el calculo de los puntos de control usando el método de refinamiento propuesto en A. Datta[3].

REFERENCES

- [1] OpenCV, camera calibration: http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html
- [2] PoenCV, Otsu: <https://docs.opencv.org/3.0-beta/doc/tutorials/imgproc/threshold/threshold.html>

- [3] A. Datta, J. S. Kim and T. Kanade, "Accurate camera calibration using iterative refinement of control points," *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, 2009*, pp. 1201-1208.