

Calibración de una cámara

Jose H. Jaita A.

San Pablo Catholic University

Abstract. Este proyecto consiste en la calibración de una cámara, el padron utilizado consta de círculos concéntricos, para lo cual se hallaran los centros y se hará un tracking.

1 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.

2 Detección de la plantilla

La primera parte del algoritmo consiste en detectar un patrón en este caso un círculo concéntrico, originalmente se usó la FFT para hallar el patrón entero, debido a la frecuencia originada por los círculos, si bien la FFT es más rápida que la DFT, su complejidad computacional aún es alta, por lo que la otra propuesta es binarizar la imagen usando el método de Otsu, luego hallar los contornos, para esta parte del algoritmo solo necesitamos encontrar un círculo concéntrico, ya que este será nuestra plantilla para la siguiente parte del algoritmo.

3 Template matching dinámico

Como sabemos entre frames consecutivos hay mucha relación, entonces lo que se hace es que a partir de una plantilla encontramos todas las coincidencias en la imagen de entrada, luego tenemos que actualizar la plantilla para el siguiente frame, primero escogemos el círculo con el que el matching fue máximo luego extraemos el círculo seleccionado, le hacemos un padding más que todo para no perder detalles de bordes, a esta región de interés que contiene al que será la nueva plantilla le aplicamos una binarización con el método de Otsu, luego calculamos los bordes y con esto ya tenemos nuestra nueva plantilla, ahora si en la detección de bordes encontramos otro borde que no es círculo podríamos tener problemas ya que nuestra nueva plantilla sería cualquier cosa menos el

circulo concentrico, para esto almacenamos el area de la plantilla anterior y lo comparamos con el area de la nueva plantilla, si guardan relacion entonces actualizamos la plantilla.

4 Propagacion

El template matching dinamico funciona muy bien ante cambios de iluminacion, pero no tan bien ante cambios de forma, debido a que el padron se mueve en tres dimensiones, con algunas posiciones los circulos pierden su propiedad, convirtiendose en elipses, entonces el template matching no es suficiente ya que detecta algunos circulos pero no todos, para esto se propone el metodo de propagacion el cual consiste en usar otra plantilla para expandir el area detectada, en la imagen de abajo se muestra como una plantilla solo detecta algunos circulos mientras que usando otra se completa el matching.

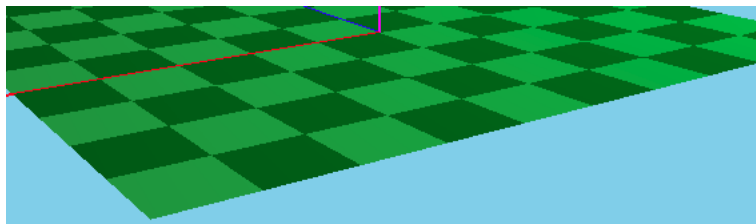


Fig. 1. Metodo de la propagación

La idea es simple, de generacion en generacion se va perdiendo similitud pero generaciones intermedias son parecidas a las nuevas que las primeras generaciones.

5 Detección de centro del los circulos

Una ves que tenemos la region de interes de cada circulo, binarizamos la imagen, luego calculamos los contornos, y finalmente hallamos el centro de masa tanto del circulo mayor como del menor, ambos difieren poco cuando la imagen de entrada no esta rotada, en cambio cuando es una elipse, ambos centros difieren por unos pocos pixeles, haciendo pruebas este centro de masa calculado usando las funciones de opencv lo hace bastante bien.

6 Orden correcto de los circulos

Para tener siempre el orden correcto de los circulos hacemos un transformacion de espacio, rotando la imagen, para esto tenemos que ir almacenado el angulo

de rotacion del ultimo frame. En el nuevo espacio , tenemos todos los circulos correctamente ordenados, en este espacio se no es mas facil saber cual es el orden correcto.

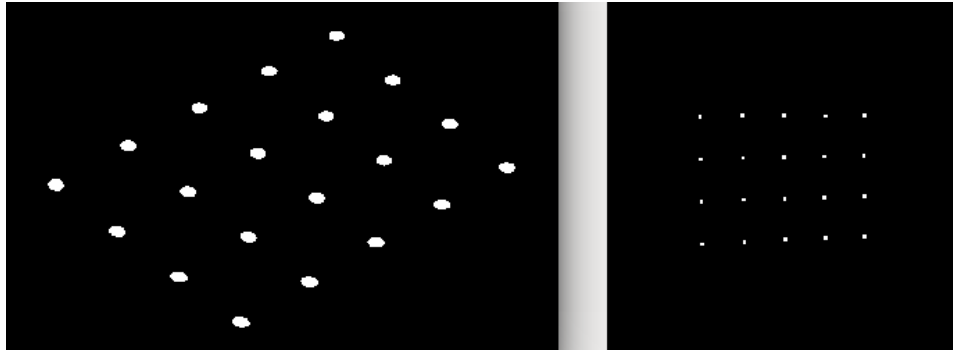


Fig. 2. Al lado izquierdo vemos la imagen original, y en lado derecho vemos la imagen en un nuevo espacio

7 Experimentos

Hasta el momento se probó el algoritmo con casi todos los videos propuestos en clase. Hay videos que tienen un bajo fps y los movimientos son bruscos, por lo que la variación entre frame y frame dificulta el trabajo del algoritmo. Hay otros videos en los cuales la iluminación es bastante cambiante, en este tipo de videos el algoritmo propuesto funciona bien.

8 Conclusion

El algoritmo funciona relativamente bien, dependiendo de los videos, hasta el momento se reconoce el patrón, su centro y se hace un tracking. El problema fundamental es la gran dependencia entre frames consecutivos, por ejemplo si la última plantilla almacenada no es buena esto afectará al reconocimiento de los demás frames. La transformación de espacio para saber la secuencia correcta de los anillos funciona bastante bien, en los experimentos se tuvo problemas cuando el ángulo de giro era mayor a 90 grados, esto es debido a que se usó el arcotangente, este varía completamente cuando se sobrepasa los 90 grados, esto se soluciona encontrando una fórmula que describa el fenómeno o simplemente corregir mediante código. Algo que también ayuda mucho es la reducción de la región de interés, esto se calcula en cada frame dependiendo de los círculos encontrados se agrega un pequeño padding, luego este será nuestro nuevo ROI para el siguiente frame.

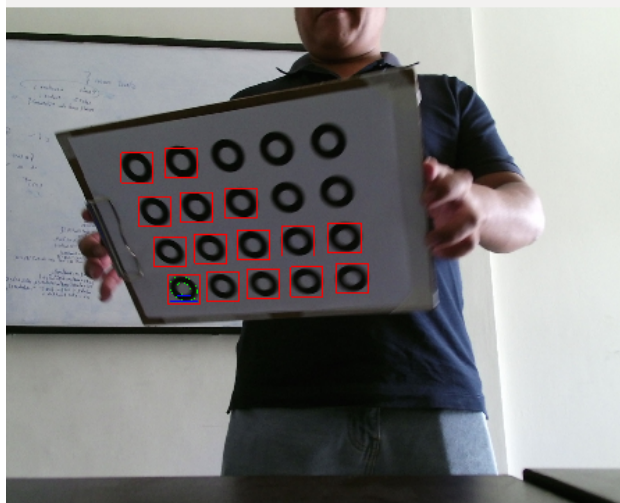


Fig. 3. Deteccion de padron sin propagacion

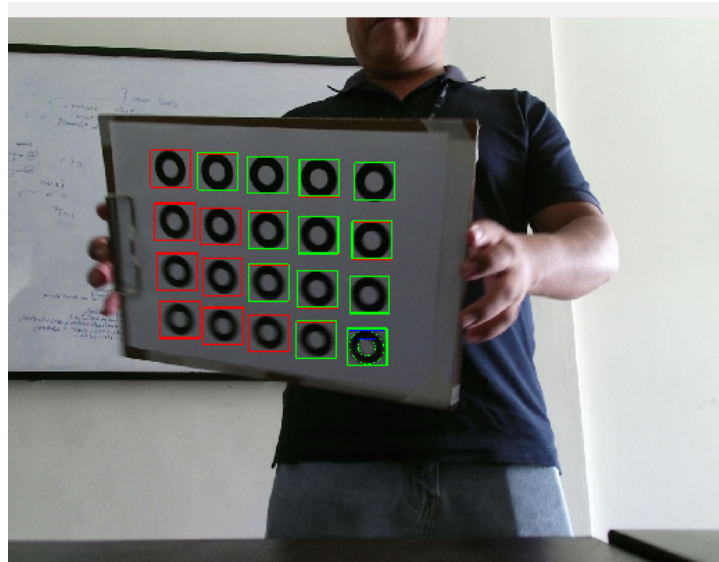


Fig. 4. Deteccion de padron usando propagacion

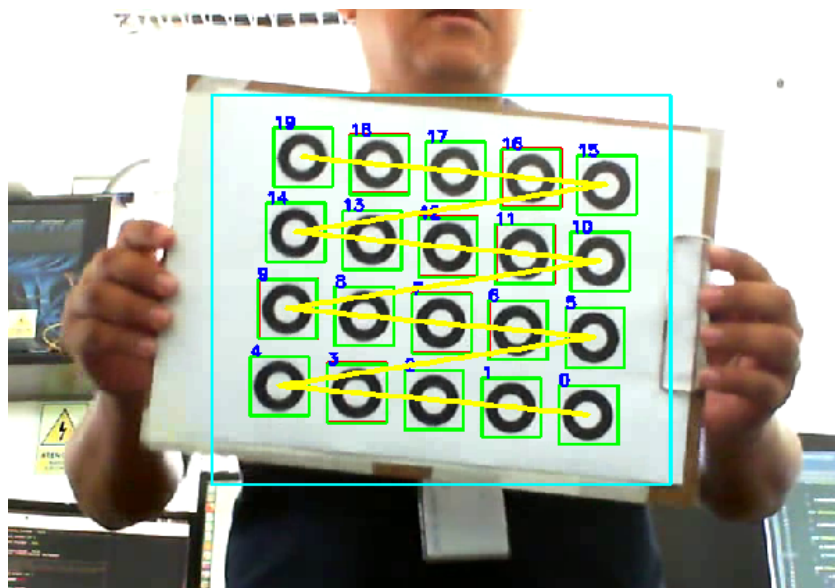


Fig. 5. Deteccion de padron + tracking