How to Tracking a Rings Pattern

Daniel Palomino
dpalominop@gmail.com

***Abstract*— In several machine vision applications, a fundamental step is to precisely determine the relation between the image of the object and its physical dimension by performing a calibration process. The aim is to devise an enhanced mechanism for camera calibration in order to improve the already existing methods in OpenCV. A good calibration is important when we need to reconstruct a world model or interact with the world as in case of robot, hand-eye coordination. In the paper that i have based is proposed an enhanced camera calibration procedure using a special grid pattern of rings. The overall objective is to minimize the re-projection for good camera calibration.**

I. INTRODUCTION

A la hora de diseñar un sistema de Visión por Computador siempre se tienen en cuenta una serie de parámetros que serán decisivos para que la apariencia de los objetos en la imagen sea la mejor de cara a los posteriores algoritmos de análisis. Así se elegirá una óptica con una distancia focal, f, que permita que se observe con el suficiente tamaño en la imagen el elemento a buscar o que tenga una lente con la suficiente calidad como para que los objetos no se vean deformados. Respecto a la colocación de la cámara, ésta se realizará de forma que se perciba de la mejor manera posible el espacio por el que pueden aparecer los objetos. Para muchas aplicaciones estas medidas son suficientes.

Sin embargo, para aquellos algoritmos de Visión por Computador que necesiten extraer información 3D de una imagen o una secuencia de ellas o establecer la correspondencia entre dos o más cámaras, la calibración de los parámetros intrínsecos y extrínsecos del sistema de visión es una etapa fundamental.

II. ALGORITMO

El métrodo propuesto se describe en los siguientes pasos:

1. Detección de los Círculos en cada imagen.
2. Usando los círculos detectados, encontrar el centro de cada anillo concéntrico.
3. Separación de los correctos de los incorrectos.
4. Ordenamiento de los centros de todos los puntos en un órden sistemático que es universalmente seguido por todas las imágenes.

II-A. Detección de los Círculos

La detección de los círculos se realiza con el siguiente pipeline:

1. Convertir a escala de grises.
2. Suavizado usando filtros gaussianos.
3. Binarización de la imagen.
4. Búsqueda de contornos.
5. Cálculo del mínimo area rectangular que envuelve a los contornos.
6. Ajuste de elipses a los rectángulos calculados.

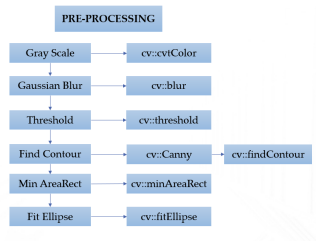

Fig. 1. Pipeline de la detección de los Círculos

II-B. Localización de los Centros

El algoritmo de localización de los centros es descrito en la siguiente figura:

```
Point2f last(-10, -10);
std::vector<node> centers;
int n = 0;
for (int i = 0; i < contours.size(); i++)
{
    Scalar color = Scalar(rng.uniform(0, 255), rng.uniform(0, 255), rng.uniform(0, 255));
    Point2f np(minEllipse[i].center.x, minEllipse[i].center.y);

    if (norm(last - np) < 5) {
        centers[n - 1].count++;
        float w_np = pow(minEllipse[i].size.area(), 3);
        centers[n - 1].center = (np*w_np + centers[n - 1].center*centers[n - 1].w) / (w_np + centers[n - 1].w);
    }
    else {
        n++;
        centers.push_back(node(np, 1, pow(minEllipse[i].size.area(), 3)));
    }
    last = np;
}
```

Fig. 2. Algoritmo de Localización de los Centros

II-C. Segregación de los Centros

El algoritmo de segregación de los centros es descrito en la siguiente figura:

```
std::vector<Point2f> PointBuffer;

for (auto center : centers) {
    if (center.count > 3)
        PointBuffer.push_back(center.center);
}
```

Fig. 3. Algoritmo de Segregación de los Centros

III. RESULTADOS

Para la presentación de los resultados, haré uso de 2 escenarios, uno con padrón rotado y otro con el padrón original sin rotar.

III-A. *Padrón Rotado*

El padrón a utilizar es el mostrado en la siguiente figura:

Fig. 4. Frame usado como imagen fuente

Luego de terminado la detección de los círculos, en la etapa de preprocesamiento, se obtiene el siguiente resultado:

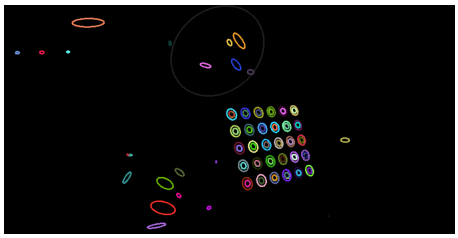

Fig. 5. Resultado luego del pre-procesamiento

Seguidamente, se realiza la ubicación de los centros y su posterior segregación:

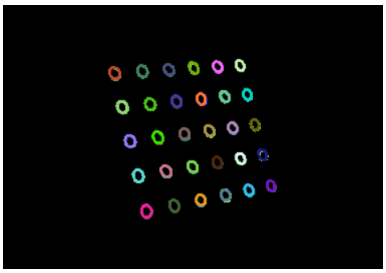

Fig. 6. Resultado luego de la segregación de los centros

Finalmente, luego de la ubicación de los centro correctos, se procede a ubicar la grilla padrón dentro de la imagen original:

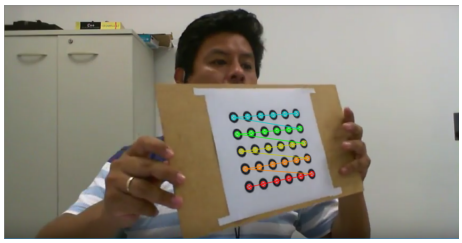

Fig. 7. Detección del padrón dentro de la imagen original

III-B. *Padrón Original sin Rotar*

Al igual que en el caso anterior, primero se define el frame a utilizar, el cual es mostrado en la siguiente figura:

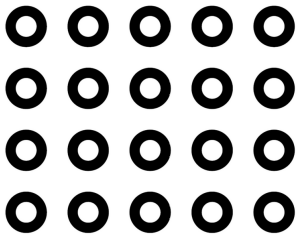

Fig. 8. Frame que contiene el padrón original sin rotar

Luego, se realiza el preprocesamiento para la detección de los círculos, seguido de la ubicación de sus centro y su posterior segregación:

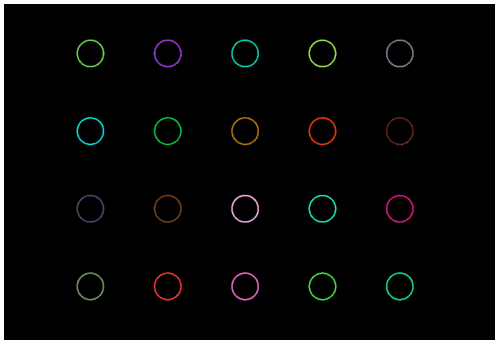

Fig. 9. Resultado luego de la segregación de los círculos correctos

Finalmente, luego de la ubicación de los centro correctos, se procede a ubicar la grilla padrón dentro de la imagen original:

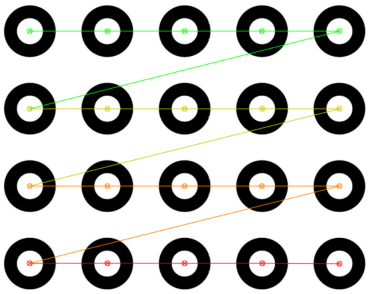

Fig. 10. Detección del padrón dentro de la imagen original

REFERENCES

[1] Asthana, Shubham. (2014). Enhanced Camera Calibration for Machine Vision using OpenCV. International Journal of Artificial Intelligence. Volume 3.

[2] https://docs.opencv.org/