

Calibración de Cámara

Joao Pinheiro , Leticia Pinto

Abstract—Esta segunda entrega se basa en crear un algoritmo que permita calcular y extraer el punto central de un padrón definido por un grid de círculos concéntricos, a su vez rastrear el padrón a medida que se mueve por la imagen capturada por la cámara en tiempo real.

1 INTRODUCCIÓN

La hora de trabajar con sistemas de visión 3D, uno de los primeros temas que se tiene que abordar es la calibración de las cámaras. Es por esto que nuestro trabajo va enfocado a la calibración con un padrón conocido de círculos que nos permitirá hallar los parámetros intrínsecos y extrínsecos de la cámara. En esta segunda entrega se busca aplicar el algoritmo a la imagen obtenida por la cámara en tiempo real.

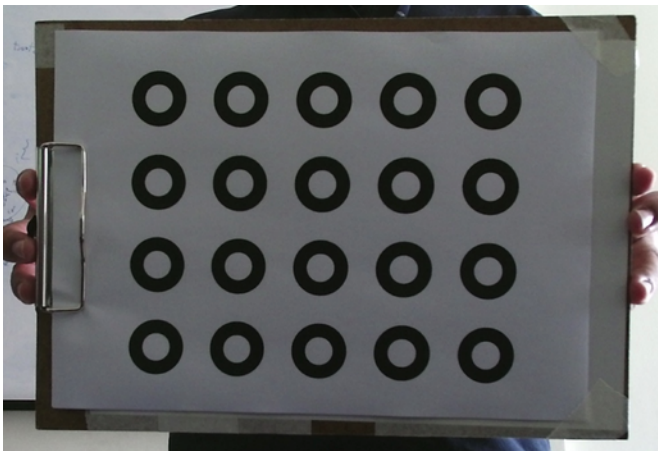


Fig. 1. Padrón de calibración

2 OBJETIVOS

- Crear algoritmo que permita calcular y extraer el punto central de un padrón definido por un grid de círculos concéntricos en tiempo real.

3 ALGORITMO

Para el rastreo del padrón, el primer paso es binarizar la imagen, para así detectar los contornos, en esta etapa se utilizó la técnica de convertir el frame de RGB al espacio de colores HSV, con la intención de filtrar los colores oscuros usando umbrales entre (0,0,0) y (180, 255, 80) en HSV. Una vez binarizada la imagen el siguiente paso fue buscar los contornos, para posteriormente aplicar el algoritmo de ellipse fitting para rastrear los ellipses en la imagen, lo que nos generó un conjunto de ellipses en el espacio de la imagen, finalmente identificar el conjunto de ellipses que pertenecen al padrón, se aplicó una heurística basada en una característica muy notoria en el padrón, son elipses que contienen otra ellipse dentro de ella, esta heurística se basa en los centros geométricos de las ellipses, la cual consiste en solo tomar las ellipses que comparten proximidad de centros con un solo vecino, siendo ésta nuestra primera condición y obteniendo buenos resultados, además para sesgar el espacio de búsqueda se uso otra heurística basada en la búsqueda de vecinos similares, es decir se descartan las ellipses que no cumplan con el criterio de ser cercanos a una ellipse que no

- Maestría en Ciencia de la Computación
Universidad Católica San Pablo
Arequipa, Perú .

cumplan con la primera condición. Esto permite que el ruido disminuya para los casos de elipses generadas fuera del padrón.

PIPELINE PARA LA DETECCIÓN Y TRACKING

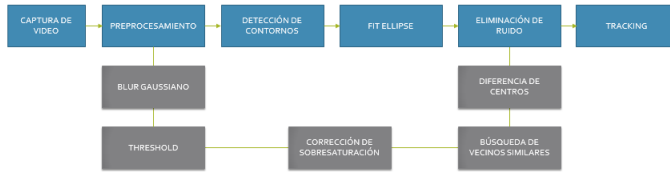


Fig. 2. Pipeline

4 IMPLEMENTACIÓN

El desarrollo de este trabajo se llevó a cabo en el sistema operativo Ubuntu 16.04, el código fue implementado en opencv con c++ y se usó como dataset la captura de imágenes de la cámara en tiempo real.

5 RESULTADOS



Fig. 3. Búsqueda de patrón

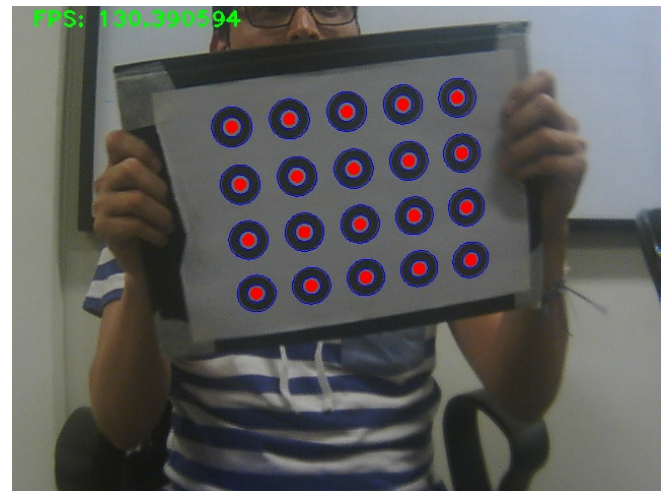


Fig. 4. Reconocimiento del patrón

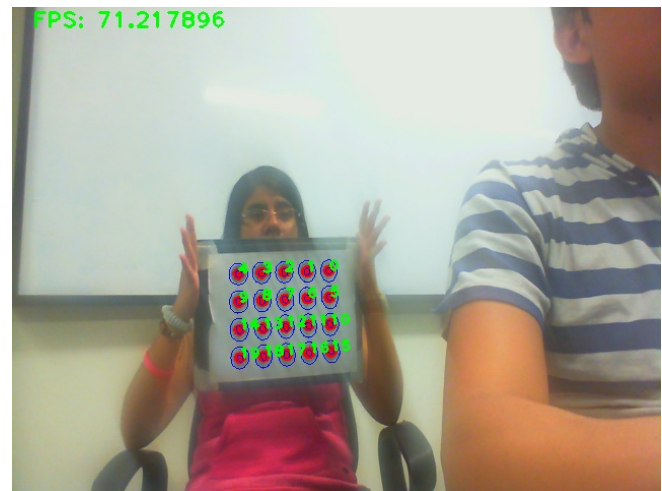


Fig. 5. Detección del patrón

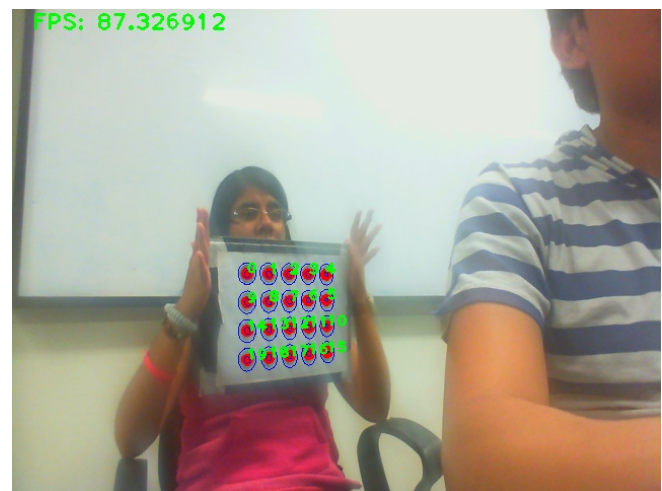


Fig. 6. Detección del patrón en el tracking

6 CONCLUSIONES

Un cálculo de threshold adaptativo provee mejores resultados, debido a que distribuye los niveles de saturación y brillo, logrando una mejor definición de las características. Al ser este de un gran costo y por la necesidad de usarlo en tiempo real, se optó por separar cuando existía un gran brillo en el frame para aplicar un adaptativo, y en los casos en los que no se perdían las elipses seguir con el threshold normal.

Podemos concluir que para la búsqueda del punto central de las elipses encontradas en la imagen permiten limpiar el ruido y centrarnos en los candidatos para que mediante el proceso de heurísticas adecuadas filtrar las elipses y lograr nuestro objetivo de ubicar los centros de los círculos dentro del patrón en tiempo real lo que implica no exagerar en el cálculo de pesados procesos que disminuyen la eficiencia y por tanto el tiempo del algoritmo de calibración.