



Ingeniería Matemática e Inteligencia Artificial

Proyecto final

Visión por Ordenador I

Group 4

Alejandro Martínez de Guinea García - 202113492

Javier Prieto Domínguez - 202103783

Indice

Introducción y alcance del proyecto	3
Metodología.....	4
Resultados y Futuros Desarrollos	6

Introducción y alcance del proyecto

El objetivo de este proyecto es analizar la información que obtenemos mediante una cámara conectada a una Raspberry Pi. Ese análisis se realiza con técnicas aprendidas en la asignatura de Visión por Ordenador I.

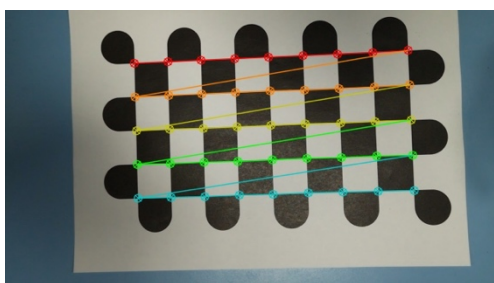
Nos centramos principalmente en algunas de ellas: la calibración de una cámara, la detección de *edges*, y la segmentación por color. Todas ellas se unen para formar una aplicación de golf. Consiste en, después de calibrar la cámara e introducir una "contraseña" basada en imágenes de patrones, hacer un tracker de una pelota de golf y detectar si se mete en un agujero o no.

Metodología

El proyecto se ejecuta de la siguiente manera. Hemos creado un main.py que integra todo lo mencionado previamente en orden:

1. Calibración:

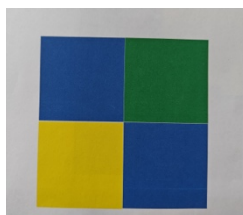
En este apartado calibramos la cámara con el patrón que aparece en la imagen de la izquierda desde distintas posiciones. Al calibrar la cámara con funciones de *opencv-python*, imprimimos la matriz de parámetros intrínsecos y el error de reproyección. Una vez calibrada la cámara, pasamos a la detección de patrones para la introducción de la contraseña.



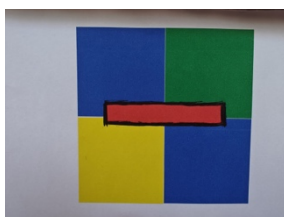
```
pi@imat-rpi-015:~/Desktop/PracticaFinal/cv_lab_project_2023-main/Proyecto_fin
oyecto_final_vision/main.py
Calibrating camera...
Corners standard intrinsics:
[[665.69110935  0.          336.77036801]
 [  0.         664.34949146 175.27852091]
 [  0.          0.          1.         ]]
Corners standard dist_coefs:
[[-0.03683731  0.05453481  0.00499428  0.00302772  0.29314468]]
root mean square reprojection error:
0.9690940996263868
```

2. Detector de patrones:

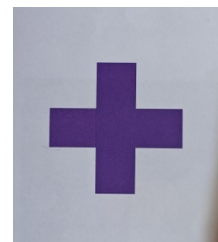
Inicialmente, nos definimos unos patrones y el orden en el que habrá que introducirlos al sistema. Sabiendo eso, ya podíamos segmentar las imágenes por color. Por cada patrón vemos si coincide con las especificaciones de cada patrón. Por ejemplo, para el Patrón , comprobamos que no haya una región de color morado ni rojo, que haya regiones de colores azul verde y amarillo, que esas regiones estén en sus respectivos cuadrantes. Al presionar la tecla "q" cuando el patrón está en el frame de la cámara, comprueba si el patrón es correcto y si tiene la orientación correcta. Si coinciden ambas, pasa al siguiente patrón. Así hasta que se termine de comprobar los patrones. Si en algún momento del proceso de comprobación hay un error al introducir el patrón, ya sea por ser el patrón erróneo o por estar en un orden equivocado, se volverá a empezar el proceso entero.



Patrón 1



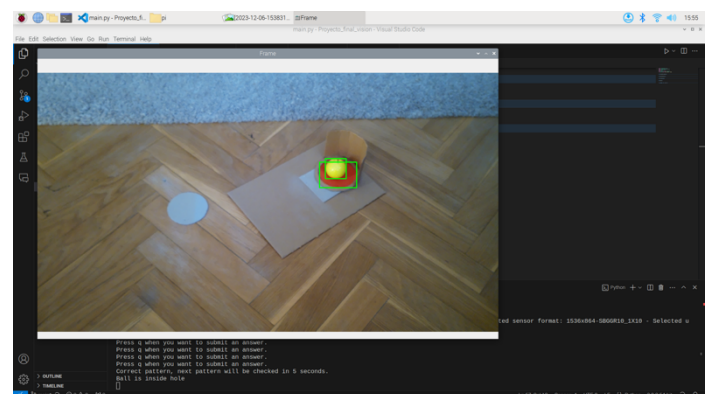
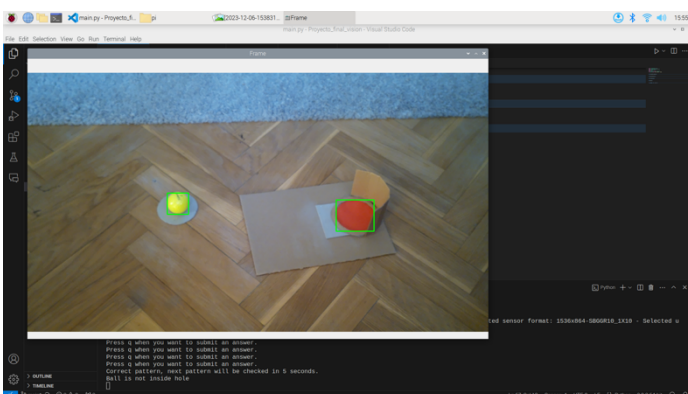
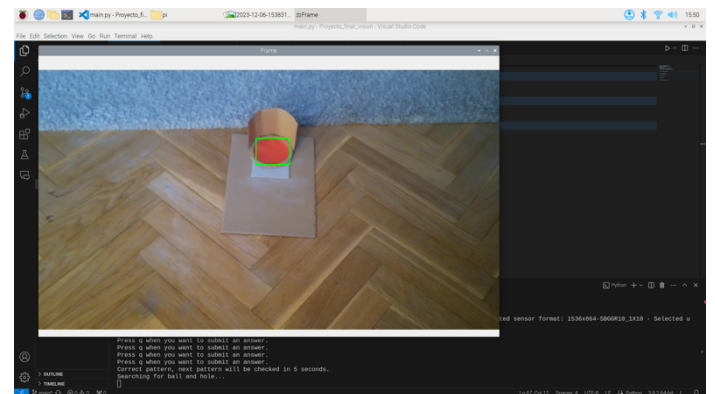
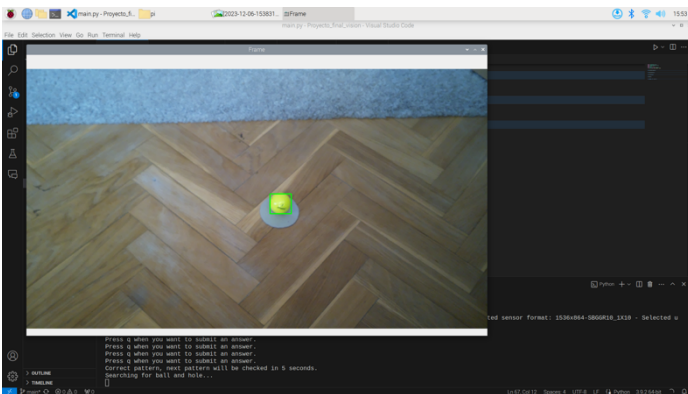
Patrón 2



Patrón 3

3. Tracker

El proceso que seguimos es el mismo para la pelota que para el agujero. Para esta parte volvemos a hacer una segmentación de color a la imagen de la cámara. Al *mask* resultante de dicha segmentación le aplicamos un *blur* y un *threshold* para evitar que pequeños fallos en la segmentación afecten. Sobre esa segmentación aplicamos una función que encuentra los *contours* de la imagen con una función de *opencv*, que genera *edges* cerrados sobre una imagen y luego hacemos su aproximación poligonal. Al ser tanto el hoyo como la pelota círculos, dicha aproximación tendrá muchas esquinas, por lo que hacemos una comprobación de que el output tenga más de 6 esquinas. Además, para evitar la detección de otros objetos, elegimos dos colores raros para la pelota y para el hoyo. Habiendolos detectado ya ambos, solo faltaría detectar si la pelota ha entrado o no. Hacemos una especie de *IOU_Score*, que detecta la proporción de área común que tienen 2 bounding boxes y el área entre las 2. Nosotros solo consideramos el área de la pelota y su intersección con el área del agujero. Si más del 50% del área de la pelota interseca con el área del agujero, consideramos que está dentro.



Resultados y Futuros Desarrollos

Los resultados que hemos obtenido han sido buenos y era lo que esperábamos del proyecto. Tuvimos también problemas con la detección de patrones pero los conseguimos arreglar.

El alcance del proyecto en este caso era hacer un tracker. En un futuro se podrían hacer cosas más avanzadas. En vez de comprobar si la pelota ha entrado al agujero, podríamos hacer un modelo que predijera la trayectoria de la pelota para que entre. Se necesitarían materiales más avanzados, como sensores de la altura del terreno y algún ordenador con más capacidad de procesamiento que la Raspberry Pi. Además, se podría tratar de implementar algún otro método de detección de objetos que nos permitiese hacer un track de cualquier pelota independientemente del color. Un ejemplo de esto podría ser la herramienta Yolo de OpenCV, que es un algoritmo que se puede entrenar para detectar cualquier objeto.

