COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI ICADE CIHS

Proyecto Final: Tracker de golf

Luis Gervás Nieto
Carlos González Vives
Grupo B

Computer Vision
3º Grado en Ingeniería Matemática e Inteligencia Artificial

Índice

Introducción

Nuestro proyecto de la asignatura de Computer Vision se basa en realizar de la forma más aproximada posible un tracker de golf. Para ello, contamos con los conocimientos adquiridos en la asignatura a lo largo del año y una Raspberry Pi junto a una cámara a conectar en uno de los pines de esta.

Nuestro tracker se basa en la detección de bolas de golf, extracción de información (número mínimo de bolas de golf para que se inicie el tracker), y el seguimiento de una bola de golf específica. Adicionalmente, mostramos por pantalla a tiempo real los resultados del tracker con una línea que muestra la trayectoria de la bola.

Para este proyecto, trabajamos primero con la calibración de la cámara, que nos daría el rms (*calibration.py*). Para que pudiéramos hacer la calibración correctamente, tomamos distintas fotos de un tablero de ajedrez con la cámara (*capture.py*). Luego, trabajamos en la correcta detección de bolas de golf (*golf_ball_detection.py*). A continuación, nos pusimos con el sistema de seguridad de la cámara (*golf_ball_lock.py).* Finalmente, hicimos el tracker (*golf_ball_tracker.py*).

Por ello, nuestro proyecto consta de los 5 anteriores scripts, seguido de una carpeta llamada *data* que contiene las distintas fotografías del tablero de ajedrez, y una carpeta llamada *videos* que contiene el video de salida que muestra como nuestro tracker funciona correctamente.

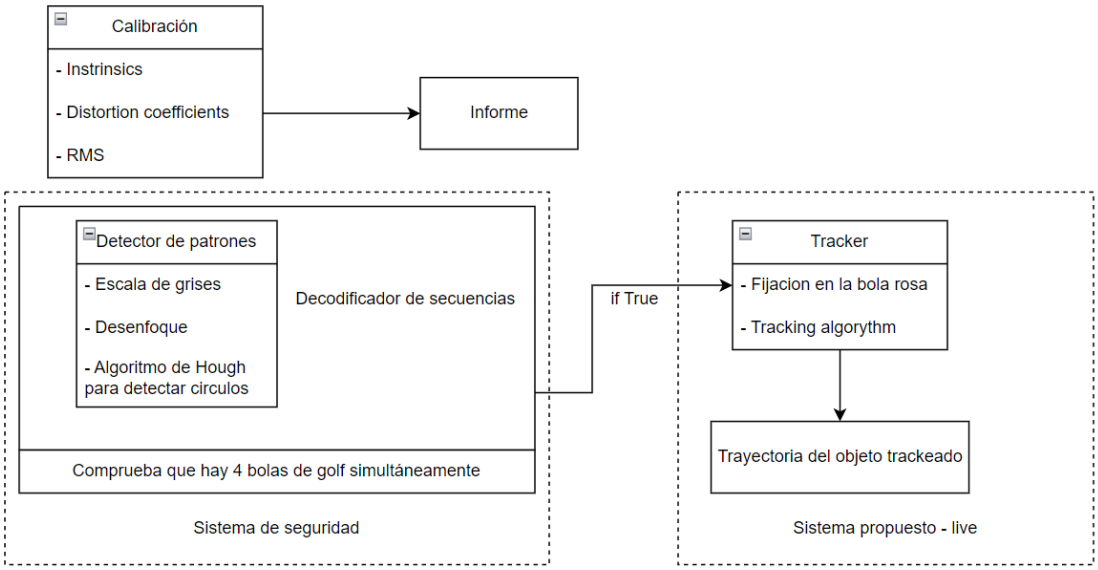
Metodología

Para la calibración de la cámara, decidimos hacer un código que nos sacara una foto y la guardara cada 2 segundos unas 19 veces desde cada lado (derecha e izquierda). Una vez tomadas y almacenadas las fotos, procedemos a la calibración, tal y como hemos hecho este año en clase. Pasamos las imágenes a gris y vemos si la función de OpenCV *findChessboardCorners()* logra encontrar las esquinas correctamente o no. Si lo hace, guardamos la imágenes con las esquinas del tablero y sin ellas mediante *cornerSubPix()*, y con *calibrateCamera()* sacamos los parámetros deseados.

```
PS C:\Users\luisg\OneDrive\Desktop\3º\Cuatri1\Vision por ordenador\CV_FinalProject> & C:/Users/luisg/AppData/Local/Microsoft/WindowsApp
c:\Users\luisg\OneDrive\Desktop\3º\Cuatri1\Vision por ordenador\CV_FinalProject\src\calibration.py:18: DeprecationWarning: Starting wit
disappear) use `import imageio.v2 as imageio` or call `imageio.v2.imread` directly.
  return [imageio.imread(filename) for filename in filenames]
Intrinsics:
 [[1.99740982e+03 0.00000000e+00 6.93057027e+02]
 [0.00000000e+00 5.14074859e+03 5.66331951e+02]
 [0.00000000e+00 0.00000000e+00 1.00000000e+00]]
Distortion coefficients:
 [[ 0.37788854 -0.50057829  0.05137478  0.02590299  0.15798447]]
Root mean squared reprojection error:
 2.3481876728348046
```

Como hemos podido observar, el rms es de 2.348, un valor un tanto grande, pero de esperar teniendo en cuenta que nuestra cámara no es muy buena. Sin embargo, este error de reproyección medio indica que probablemente tengamos problemas a la hora de trabajar con la cámara.

El diagrama de bloques del sistema es el siguiente:



Para la detección de patrones, hemos optado por detectar bolas de golf únicamente, ya que nuestro sistema propuesto se iba a basar en trackear una de ellas. Por ello, primero pasamos los frames en vivo a escala de grises, luego desenfocamos la imagen y realizamos el *HoughCircles()* para identificar objetos circulares, parametrizando esta función para coger solo los círculos pequeños que se asemejen a bolas de golf.

A la hora de hacer la parte de la contraseña, hemos optado por un caso algo mas cotidiano. Como nos gustaría que este tracker fuera un sistema plenamente funcional para el día a día (si quisiéramos sacar el software al mercado, por ejemplo), decidimos hacer que el sistema reaccionara si y solo si hay 4 bolas de golf captadas por la cámara. ¿Por qué 4? Como mencionamos antes, el rms de la cámara es alto y da problemas. Para evitar que el tracker se inicie si cree que esta viendo una bola de golf (cuando en verdad es una forma circular en el espacio, no una bola de golf), hemos decidido que este solo se inicie si determina que se encuentra en un ambiente golfístico, es decir, tiene que detectar muchas bolas de golf.

Como el tracker se inicia cuando se divisan 4 bolas de golf, ¿Cómo sabe la Raspberry Pi la bola que ha de trackear? Pues bien, hemos decidido que únicamente trackee las bolas de color rosa. Esto nos permitirá que el tracking se realice de forma exitosa, ya que la mayoría de las bolas de golf son blancas. Para hacer esto posible, devolveremos en la detección de una bola su color, para poder identificar la bola rosa entre unos rangos de color. Una vez identificadas 4 bolas de golf, entre ellas la rosa, se inicia el tracker sobre ella. Además de traquear su posición, hemos ido guardando la posición de esta en cada tracking en un historial llamado *trajectory*, de tal forma que pudiéramos también pintar la trayectoria durante el tracking.

Hemos empleado para la grabación 10 fps, ya que consideramos que era lo que más se asemejaba a la captura de frames de la cámara en tiempo real. Como hemos trabajado con todo a tiempo real (menos la calibración, claro, que se hacia offline), hemos decidido que un video de 10 segundos bastaría para mostrar nuestro trabajo. Por ello, mientras mostramos la imagen a tiempo real en el tracker, se graba un vídeo que captura 100 frames (10 fps la calidad de la grabación) y se cierra si la letra “q” no es presionada antes.

Resultados

Los resultados fueron gratamente satisfactorios, ya que conseguimos ajustar los parámetros correctamente para identificar las bolas de golf, algo que nos dio problemas en un principio. La contraseña para que se inicie el tracker puede no ser una contraseña muy compleja, pero sentimos que se adapta muy bien al problema que tenemos con la calidad de grabación de la cámara (ve círculos donde a lo mejor no los hay) y sobre todo a la vida cotidiana, ya que esto únicamente se ejecutaría en un entorno de golf.

El tracker a veces pierde la bola debido a que no es capaz de procesar un movimiento tan rápido. Para lidiar con este problema, decidimos pausar el tracker si se pierde de vista la bola rosa, y si se vuelve a localizar, reactivar el tracker. Para pintar la trayectoria en este tipo de casos, decidimos que lo mejor es unir el último punto registrado en *trajectory* donde estuvo la bola, y el nuevo punto donde se inicia el tracker. Creemos que es una gran forma de solucionar el problema de la calidad de la cámara.

El software es muy visual, reproduciéndose en vivo y mandando mensajes por la terminal en caso de algún problema / acontecimiento (como cuando se activa el tracker si la contraseña es correcta).

Futuros Desarrollos

Como futuros desarrollos, estaría bien adaptar este proyecto para golpes de golf mas largos. Para ello, necesitaríamos una mejor cámara y además aplicar redes neuronales para predecir el movimiento de la bola en caso de perderla de imagen debido a su diminuto tamaño cuando observamos la bola de golf a 200 metros, por ejemplo.