



Pelota 98%

Seguridad y Seguimiento de Canastas en Baloncesto



TRABAJO FINAL VISIÓN POR ORDENADOR

Lucía Tamarit e Irene Turrado

1. Introducción (alcance del proyecto)

El objetivo de este proyecto es el diseño e implementación de un sistema de Visión por Ordenador en tiempo real basado en una cámara monocular. El sistema integra técnicas clásicas de procesamiento de imagen para resolver dos bloques principales: un sistema de seguridad basado en patrones visuales y un sistema de seguimiento (tracker) que se activa únicamente cuando la secuencia de patrones introducida es correcta.

El sistema de seguridad actúa como un mecanismo de control de acceso visual, validando la presencia y el orden de distintos objetos en la escena. Una vez superada esta etapa, se habilita el sistema propuesto, consistente en el seguimiento de una pelota de baloncesto y la detección automática de canastas en tiempo real.

El proyecto incluye la calibración previa de la cámara, la segmentación por color en el espacio HSV, la detección y análisis de contornos, y el seguimiento temporal de objetos. Todo el sistema ha sido desarrollado en Python utilizando la librería OpenCV, garantizando una ejecución fluida en tiempo real y una arquitectura modular fácilmente ampliable.

2. Metodología

2.1 Calibración de la cámara

La calibración de la cámara se realiza de forma offline, utilizando un patrón de tablero de ajedrez. A partir de múltiples imágenes del patrón, se detectan automáticamente las esquinas internas y se emplea la función `cv2.calibrateCamera()` para estimar la matriz intrínseca de la cámara, los coeficientes de distorsión y el error medio cuadrático de reproyección (RMS).

Los valores obtenidos durante la calibración se incluyen en la figura 1 y se utilizan posteriormente en el módulo de tracking para aplicar la corrección de distorsión a cada frame del vídeo.

```
# Print outputs
print("Intrinsics:\n", intrinsics)
print("Distortion coefficients:\n", dist_coeffs)
print("Root mean squared reprojection error:\n", rms)

Intrinsics:
[[851.09792811  0.          496.3082453 ]
 [ 0.          824.72695001 337.64549556]
 [ 0.           0.           1.          ]]
Distortion coefficients:
[[-0.16626935  0.82529379  0.00386685 -0.06644165 -0.88137497]]
Root mean squared reprojection error:
3.356730076413496
```

Figura 1: resultados de la calibración de la cámara

La Figura 1 muestra una captura de la ejecución del proceso de calibración junto con los parámetros resultantes empleados en el sistema.

2.2 Diagrama de bloques del sistema

El sistema implementado se organiza en una arquitectura modular claramente definida. En primer lugar, la cámara proporciona imágenes en tiempo real que son procesadas por el sistema de seguridad. Este bloque se encarga de analizar los frames de entrada, detectar patrones visuales y validar una secuencia concreta.

Una vez superado el sistema de seguridad, se habilita el acceso al sistema propuesto, que corresponde al módulo de tracking deportivo. De forma independiente, la calibración de la cámara proporciona los parámetros necesarios para la corrección de la distorsión, que se aplican dentro del tracker. La Figura 2 recoge el diagrama de bloques completo del sistema, mostrando los distintos módulos y el flujo de datos entre ellos.

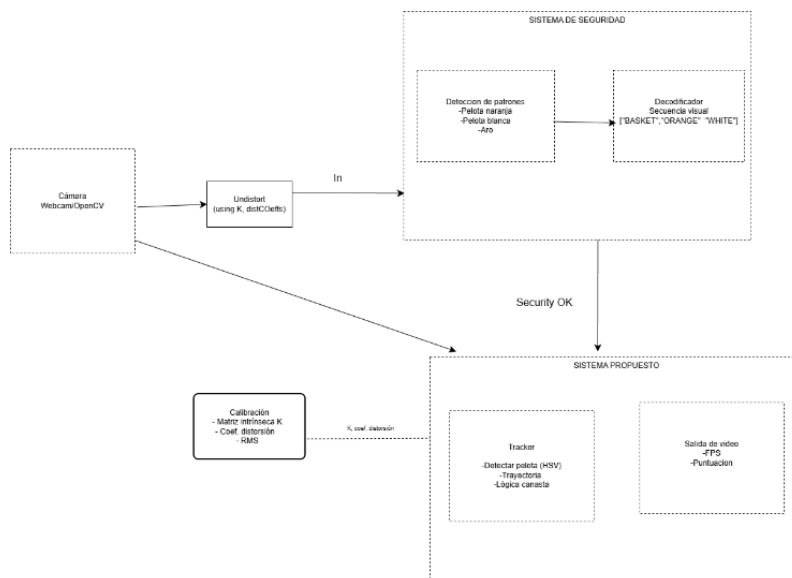


Figura 2: Diagrama de bloques

2.3 Secuencia de transformación de la imagen

Cada frame capturado por la cámara sigue una secuencia de procesamiento bien definida. En primer lugar, la imagen se adquiere en formato BGR y se transforma al espacio de color HSV, ya que este modelo separa de forma más robusta la información de color (tono) de la iluminación. Esta transformación facilita la segmentación de los objetos de interés, permitiendo aislar de manera más precisa los distintos patrones visuales presentes en la escena.

Sobre dichas máscaras se aplican operaciones morfológicas de apertura y cierre con el objetivo de reducir el ruido y mejorar la calidad de las detecciones. Posteriormente, se extraen los contornos relevantes y se calculan propiedades geométricas como el área, la circularidad o las relaciones de aspecto, que permiten validar cada detección.

El resultado de este procesamiento se utiliza tanto en el sistema de seguridad como en el sistema de seguimiento del objeto.

2.4 Sistema de seguridad

El sistema de seguridad es el primer bloque que se ejecuta y tiene como objetivo controlar el acceso al sistema principal. Este módulo analiza continuamente los frames capturados por la cámara y detecta la presencia de distintos patrones visuales, como la canasta, la pelota naranja, la pelota blanca y el marcador

En la parte superior de la pantalla se muestra, en tiempo real, la lista de patrones detectados en cada instante, independientemente de su orden, como se puede ver en la figura 3 .De forma paralela, el sistema implementa un decodificador de secuencia que comprueba si los patrones aparecen siguiendo un orden concreto predefinido. En este proyecto, la secuencia correcta está compuesta por la detección de la canasta, seguida de la pelota naranja, la pelota blanca y, finalmente, el marcador.

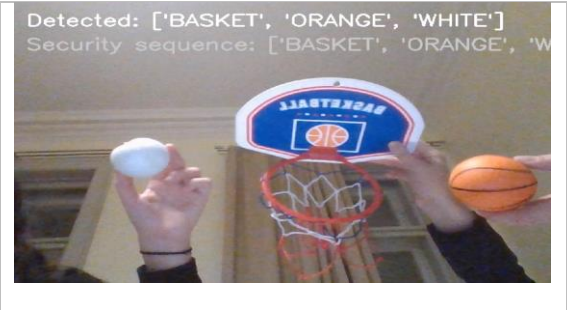
	
Vemos que en <i>detected</i> esta tanta la canasta (basket) como la bola blanca. Sin embargo, en la security squence solo aparece la canasta, ya que, para localizar la bola blanca, antes debe ser detectada la bola naranja.	Ahora hemos añadido la bola naranja, con lo que ambas pelotas pasan a la security sequence.

Figura 3: ejemplo de ejecución de Security.py

El decodificador solo añade un patrón a la secuencia válida si este aparece en el orden correcto. Si un patrón se detecta fuera de lugar o desaparece el primer elemento de la secuencia, la secuencia se invalida y se reinicia. De este modo, el sistema distingue entre lo que simplemente está siendo detectado y lo que forma parte de una secuencia válida. Cuando la secuencia completa se detecta correctamente, la interfaz muestra un mensaje

de confirmación en color verde indicando que la seguridad ha sido superada y el sistema permite el acceso al módulo de tracking, como podemos ver en la figura 4.

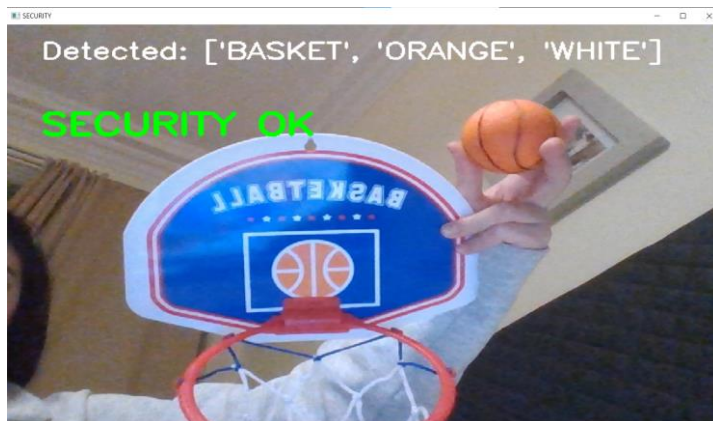


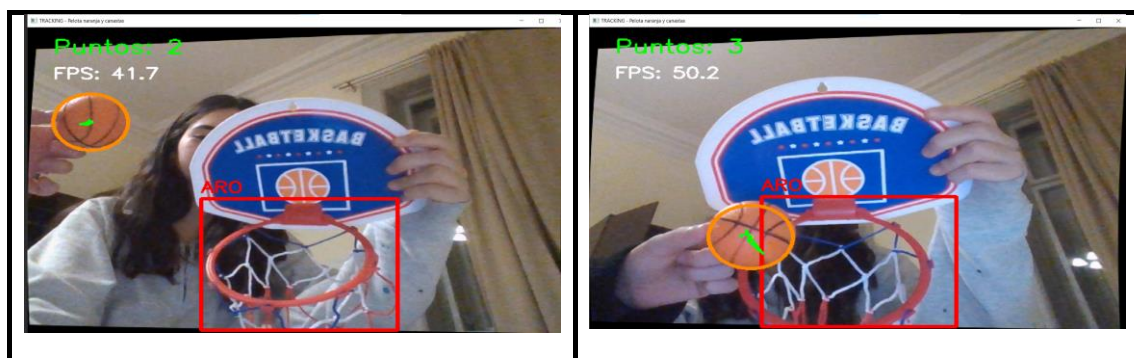
Figura 4: Security OK completado

2.5 Sistema propuesto: Tracker y salida de video

Una vez superado el sistema de seguridad, se ejecuta el sistema propuesto, que consiste en un tracker deportivo en tiempo real. En una primera fase, el sistema detecta la posición del aro de la canasta. Esta detección se representa visualmente mediante una bounding box rectangular dibujada sobre el vídeo.

A continuación, el sistema detecta la pelota de baloncesto mediante segmentación por color en el espacio HSV y calcula su posición y tamaño. La pelota se representa mediante un círculo que marca su centro y radio, y se almacena su trayectoria a lo largo de los frames sucesivos. Esta trayectoria se visualiza mediante una traza que permite observar el movimiento del objeto en el tiempo.

El sistema incorpora además una lógica específica para detectar canastas. Cada vez que la pelota cruza la zona del aro siguiendo una trayectoria descendente válida, el sistema incrementa el contador de puntos. La salida de vídeo muestra en tiempo real la bounding box del aro, el marcador de la pelota, la trayectoria seguida, la puntuación acumulada y la tasa de refresco del sistema. La ejecución finaliza cuando el usuario pulsa la tecla correspondiente, cerrando la aplicación de forma controlada.



Ejemplo tracking. Primero se detecta el aro rojo y se realiza el seguimiento de la pelota	Al meter la pelota sobre la canasta se van contando las canastas con la variable Puntos (vemos como va subiendo)
---	--

Figura 5: ejecución del tracking

```

• nvironment/Lab_Project/src/tracking.py"
Buscando aro rojo en los primeros frames...
Aro detectado: (204, 268, 231, 206)
TRACKING iniciado. Pulsa 'q' para salir.
CANASTA! Puntos: 1
CANASTA! Puntos: 2
CANASTA! Puntos: 3

```

Figura 6: salida tracking conteo de canastas

Además, se muestra en pantalla la tasa de refresco del sistema (FPS), que se va actualizando en cada frame. En la ejecución se observan valores en torno a 40–60 FPS, lo que indica que el sistema es capaz de procesar las imágenes y realizar las tareas de detección, seguimiento y conteo de canastas en tiempo real, garantizando una visualización fluida y una respuesta inmediata del sistema.

3. Resultados

El sistema desarrollado funciona correctamente en tiempo real, manteniendo una tasa de refresco estable y permitiendo una interacción fluida con el usuario. El sistema de seguridad demuestra un comportamiento robusto, diferenciando claramente entre la simple detección de patrones y la validación de la secuencia correcta. Por su parte, el tracker permite seguir de forma precisa la pelota, visualizar su trayectoria y contabilizar correctamente las canastas realizadas.

Las capturas incluidas en esta sección muestran tanto el funcionamiento del sistema de seguridad como la ejecución del tracker deportivo, evidenciando que los objetivos planteados al inicio del proyecto se han cumplido satisfactoriamente.

4. Futuros desarrollos

Como líneas de trabajo futuro se proponen:

- Uso de trackers más avanzados y comparación de rendimiento.
- Seguimiento simultáneo de múltiples objetos.
- Corrección de distorsión adaptativa en tiempo real.
- Inclusión de métricas cuantitativas del error de tracking.
- Integración de distintos modos de juego o secuencias de seguridad configurables.

Uso de IA:

Durante el desarrollo del proyecto se ha utilizado una herramienta de Inteligencia Artificial como apoyo puntual para la resolución de dudas técnicas, la mejora de la estructura del código y la redacción de documentación. En todo momento, las decisiones de diseño, la implementación final y la validación del sistema han sido realizadas por las autoras del proyecto.