

Proyecto final Computer Vision

1. Introducción

El presente proyecto aborda el diseño e implementación de un sistema integral de visión por ordenador en tiempo real, concebido para operar mediante una única entrada de vídeo.

La solución desarrollada unifica los conocimientos adquiridos durante el curso en dos módulos funcionales diferenciados: un **bloque de seguridad**, que gestiona el control de acceso mediante el reconocimiento y decodificación secuencial de letras, y un **Tracker** orientado al arbitraje automático y análisis de ping-pong.

Integrando técnicas de calibración de cámara, rectificación geométrica (homografías), segmentación por color y algoritmos de estimación de estado (Filtro de Kalman), el sistema es capaz de interpretar la escena, validar credenciales y monitorizar la física de la pelota con robustez y fluidez.

2. Calibración de cámara

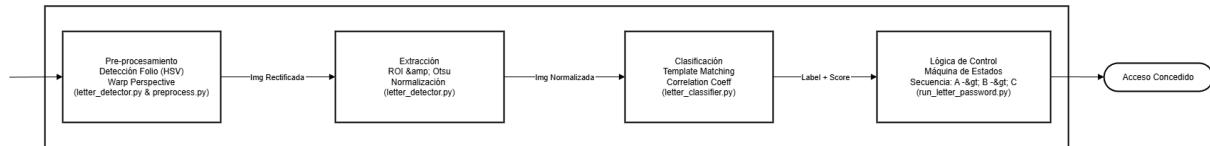
Previo a la ejecución del sistema, se realizó una calibración offline para caracterizar la óptica del móvil y corregir las deformaciones de lente (radiales y tangenciales). Se utilizó un patrón de ajedrez estándar de 7x9 esquinas internas con casillas de 30 mm. Mediante la captura de múltiples vistas y el uso del algoritmo de Zhang, se obtuvieron la matriz de parámetros intrínsecos y los coeficientes de distorsión.

La calidad del proceso se validó obteniendo un error medio de reproyección (RMS) de **0.92** píxeles, un valor adecuado para garantizar la precisión en las mediciones posteriores.

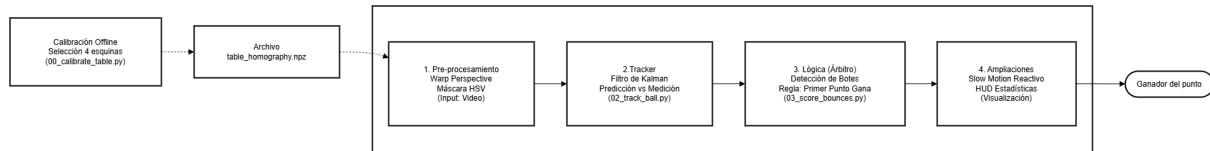
```
K (intrínsecos):  
[[1.10054092e+03 0.00000000e+00 5.27092410e+02]  
[0.00000000e+00 6.18211160e+02 4.16648125e+02]  
[0.00000000e+00 0.00000000e+00 1.00000000e+00]]  
dist:  
[ 4.19454254e-01 -3.29208988e+00 1.30588068e-02 -6.84045734e-03  
1.09305339e+01]
```

3. Diagramas de bloques

1. Sistema de seguridad



2. Ping- pong tracker



4. Metodología: Sistema de Seguridad

El objetivo de este apartado es reconocer una secuencia de caracteres predefinida (A,B,C) mostrada mediante tarjetas físicas ante la cámara. El proceso se divide en cuatro etapas secuenciales: detección del soporte, extracción del glifo, clasificación y lógica temporal.

4.1. Detección y Rectificación del Soporte

El primer paso consiste en aislar la tarjeta del entorno (folio blanco). Se realiza una conversión al espacio de color **HSV**, donde se aplica un umbralizado para segmentar tonos blancos (baja saturación, alto valor).

Para obtener una geometría consistente, se procesa la máscara binaria resultante:

1. **Filtrado Morfológico:** Se aplican operaciones de cierre y apertura para eliminar ruido y consolidar la región del papel.
2. **Extracción de Vértices:** Se aproximan los contornos mediante el algoritmo de Douglas-Peucker. Si se detecta un cuadrilátero válido, se ordenan sus vértices.
3. **Homografía (Warping):** Se aplica una transformación de perspectiva, proyectando el contenido de la tarjeta a una vista frontal plana de dimensiones fijas, eliminando deformaciones perspectivas y facilitando la lectura posterior.

4.2. Segmentación y Normalización del Glifo

Una vez rectificada la imagen, el sistema extrae la letra contenida en ella. Se define una Región de Interés (ROI) central para ignorar los bordes del papel y los dedos del usuario.

- **Binarización:** Se convierte la ROI a escala de grises y se aplica un umbralizado invertido.
- **Limpieza de Componentes:** Se eliminan artefactos conectados a los bordes de la imagen, garantizando que solo queden elementos centrales.

- **Selección de Candidato:** Mediante análisis de componentes conectados se seleccionan los "blobs" que cumplen criterios geométricos específicos de área y relación de aspecto, descartando ruido o sombras.
- **Normalización:** El candidato seleccionado se recorta y se redimensiona a una plantilla cuadrada de, centrando el glifo y manteniendo su relación de aspecto para evitar deformaciones.

4.3. Clasificación por Correlación Cruzada (Template Matching)

La identificación del carácter se realiza mediante Template Matching. El glifo normalizado se compara contra un banco de imágenes de referencia (plantillas "A", "B", "C") cargadas previamente en memoria.

Se utiliza el método de Coeficiente de Correlación Normalizado, que devuelve un valor de similitud entre -1 y 1. Este método fue seleccionado por su robustez ante cambios lineales de iluminación. Si la puntuación máxima supera un umbral de confianza (0,45) se considera que la letra ha sido reconocida.

4.4. Lógica de Estado y Decodificación Temporal

Para garantizar la robustez del desbloqueo y evitar falsos positivos por parpadeos de detección, se implementó una **Máquina de Estados Finitos** que gestiona la secuencia de la contraseña.

- **Estabilidad (Debouncing):** El sistema exige que un carácter sea detectado de forma continua durante un número mínimo de frames antes de considerarlo válido.
- **Estado de Reposo (Release):** Entre cada letra de la contraseña, el sistema obliga al usuario a retirar la tarjeta (estado "sin detección") durante un periodo de enfriamiento. Esto previene que una misma tarjeta sea contada dos veces por error.

Solo cuando la secuencia detectada coincide exactamente con el vector objetivo **["A", "B", "C"]**, el sistema valida el acceso y transiciona al módulo principal del proyecto (Tracker).

4.5 Ampliación

Como valor añadido a la robustez del módulo de seguridad, se implementaron técnicas avanzadas de pre-procesamiento y lógica temporal:

- Se dotó al sistema de **invariancia a la perspectiva** mediante el cálculo dinámico de la homografía de la tarjeta en cada fotograma, permitiendo una clasificación correcta incluso cuando el soporte presenta inclinaciones severas respecto a la cámara.
- Se integró un mecanismo de **anti-rebote (debouncing)** en la máquina de estados, el cual impone un tiempo de "enfriamiento" y exige la retirada física de la tarjeta (*release state*) entre lecturas, eliminando así falsos positivos por duplicidad en la secuencia de entrada.

5. Metodología: Sistema Propuesto (Ping-Pong Tracker)

Esta sección describe la implementación del bloque de aplicación libre, consistente en un sistema de arbitraje y análisis de ping-pong. El sistema es capaz de rastrear la pelota en tiempo real, detectar botes sobre la mesa y determinar el ganador del punto.

Dado el proyecto que hemos planteado nos ha resultado más sencillo y práctico usar un vídeo, en vez de la web-cam, para obtener un mejor ángulo para la detección.

5.1. Pre-procesamiento y Homografía

Para garantizar que el análisis espacial sea consistente independientemente de la perspectiva de la cámara, se implementó una transformación de perspectiva. El usuario selecciona las cuatro esquinas de la mesa, y se calcula la matriz de homografía.

Esta matriz permite proyectar la imagen de la mesa a una vista cenital, simplificando la lógica física, ya que la velocidad y las distancias se vuelven lineales.

5.2. Detección y Segmentación (HSV)

La detección de la pelota se realiza mediante segmentación por color en el espacio **HSV**. Se definieron umbrales y se aplicó una máscara binaria que elimina el ruido.

Dado que la perspectiva original altera el tamaño aparente de la bola, se implementó un filtrado de contornos asimétrico:

- **Zona Superior (Top):** Se permiten áreas y radios menores, ya que la bola está más lejos de la cámara.
- **Zona Inferior (Bot):** Se exigen áreas mayores y mayor circularidad.

Esto reduce falsos positivos en la zona alejada de la red y mejora la robustez del sistema.

5.3. Tracking con Filtro de Kalman

Para solucionar problemas de desenfoque por movimiento, se integró un Filtro de Kalman lineal.

- **Estado:** El filtro estima un vector de estado de 4 variables
- **Predicción:** En cada iteración, el filtro predice la nueva posición basándose en la velocidad constante.
- **Corrección:** Si el detector visual encuentra la bola (candidato válido), se corrige la estimación. Si no se detecta, el sistema confía en la predicción del Kalman, permitiendo mantener la estela de la bola incluso si esta desaparece momentáneamente por la velocidad.

5.4. Lógica de Juego y Detección de Botes

El sistema almacena un historial de las últimas posiciones y evalúa vectores de movimiento consecutivos. Se considera que ha ocurrido un **bote** si se cumplen simultáneamente tres condiciones físicas:

1. **Cambio brusco de ángulo:** El ángulo entre vectores supera un umbral (cambio de dirección vertical)
2. **Pico de aceleración:** La magnitud del cambio de velocidad supera un umbral
3. **Velocidad mínima:** Para filtrar ruido estático.

Si se detectan dos botes consecutivos en el mismo lado de la mesa (TOP o BOT), el sistema asigna el punto al oponente, detiene el juego y anuncia al ganador en pantalla.

5.5. Ampliaciones y Visualización

Como valor añadido, el sistema incluye:

- **Slow Motion Reactivo:** Al detectar un bote, el sistema reduce automáticamente el *delay* de refresco durante 30 frames, creando un efecto de cámara lenta para facilitar la visualización del impacto.
- **HUD en Tiempo Real:** Se superponen estadísticas de velocidad, aceleración y FPS, así como el estado del juego.

6-. Resultados y futuros desarrollos

El sistema desarrollado ha demostrado un funcionamiento robusto en tiempo real. El módulo de seguridad logra una decodificación fiable de la secuencia de acceso (A, B, C), gracias a la normalización de perspectiva que mitiga los errores por inclinación del soporte.

Por su parte, el *tracker* deportivo mantiene un seguimiento estable de la pelota, donde el Filtro de Kalman compensa eficazmente las pérdidas momentáneas de detección durante movimientos rápidos. La implementación de la homografía permite calcular métricas físicas (velocidad y aceleración) consistentes en toda la mesa, facilitando que el algoritmo de arbitraje identifique correctamente los botes y determine al ganador. La tasa de refresco (FPS) se mantiene estable, validando la eficiencia del código para aplicaciones en vivo.

Como líneas de trabajo futuro, se podría sustituir la segmentación cromática (HSV) por un modelo de *Deep Learning* para aumentar la tolerancia frente a cambios drásticos de iluminación, que actualmente requieren un ajuste manual de umbrales.

Asimismo, se podría expandir la lógica de juego para gestionar partidas completas (11 puntos) y añadir un módulo de re-identificación que permita distinguir entre la pelota de juego y otros objetos de color similar en el entorno.