

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Desarrollo e implementación de un sistema de
visión artificial para el análisis automático de
golpeos y dinámicas de movimiento en emisiones
de tenis**

Autor: Víctor Lizana Sánchez

Tutor: Pablo Carballeira

junio 2024

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© 29 de abril de 2024 por UNIVERSIDAD AUTÓNOMA DE MADRID
Francisco Tomás y Valiente, nº 1
Madrid, 28049
Spain

Víctor Lizana Sánchez

Desarrollo e implementación de un sistema de visión artificial para el análisis automático de golpes y dinámicas de movimiento en emisiones de tenis

Víctor Lizana Sánchez

RESUMEN

La visión artificial es un campo en auge que se trata de integrar en cada vez más ámbitos, entre ellos el deporte. Para el tenis existen diferentes soluciones comerciales enfocadas, sobre todo, a la recopilación de estadísticas de partidos profesionales para mostrarlas a los seguidores de este deporte.

El sistema desarrollado en este proyecto se centra en procesar un vídeo de una retransmisión de un partido analizando cómo se mueven los jugadores a lo largo de cada punto y cómo direccionan la pelota con sus golpes. Estudiando estas dinámicas, se clasificarán los golpes según sean de ataque o de defensa y se representará sobre el vídeo de salida, además de otros apoyos visuales como dibujar las trayectorias de la pelota. De este modo, viendo el vídeo resultante, se facilita el estudio y comprensión de las consecuencias que tiene para un jugador de tenis las diferentes formas de posicionarse en pista y de golpear la pelota.

Para el desarrollo, se han investigado y comparado proyectos de código abierto existentes para los que ya se han desarrollado diferentes técnicas de visión artificial para partidos de tenis, como clasificación de imágenes o detección de objetos. De estos proyectos se han extraído y adaptado técnicas útiles en nuestro sistema. Sobre ellos desarrollaremos nuevos módulos que reduzcan errores y pérdidas, aumentando la precisión de la información obtenida para finalmente realizar nuestros cálculos sobre los golpes y movimientos de los jugadores a lo largo de los puntos jugados en el vídeo.

Finalmente, analizaremos los resultados que hemos ido obteniendo a lo largo de nuestro sistema para diferentes vídeos, comparándolos con información etiquetada manualmente sobre ellos, de forma que podamos validar los diferentes procesos desarrollados en este proyecto.

PALABRAS CLAVE

Visión artificial, tenis, redes neuronales convolucionales, Faster R-CNN, YOLO, TrackNet, homografía, transformaciones de Hough

ÍNDICE

1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	2
2 Estado del arte	3
2.1 Tecnologías de Visión Artificial existentes	3
2.1.1 Clasificación de imágenes según enfoque a pista	3
2.1.2 Transformaciones de Hough para detección de lineas de pista	4
2.1.3 Homografías para localizar coordenadas en el plano de la pista	5
2.1.4 Detección de personas en imágenes	5
2.1.5 Detección de pelota con TrackNet	6
3 Descripción del sistema	7
3.1 Esquema general	7
3.2 Parte I: detecciones y procesado de bajo nivel	9
3.2.1 Clasificación de frame según pista completa	9
3.2.2 Detección de puntos de pista	9
3.2.3 Detección de jugadores	13
3.2.4 Detección de pelota	16
3.3 Parte II: post-procesado de bajo nivel	17
3.3.1 Suavizado de puntos de pista	18
3.3.2 Eliminación de jugadores erróneos	19
3.3.3 Interpolado de jugadores	19
3.3.4 Eliminación de pelota errónea	20
3.3.5 Interpolación de pelota	21
3.4 Parte III: procesado de alto nivel	21
3.4.1 Detección de golpes	22
3.4.2 Corrección de golpes	23
3.4.3 Cálculo de trayectoria de la pelota	24
3.4.4 Clasificación de golpes	25
3.4.5 Cálculo de movimiento de jugadores	28
4 Experimentos, resultados y validaciones	29
4.1 Vídeos utilizados para la validación	29

4.1.1	Vídeo de frames aleatorios	29
4.1.2	Vídeo corto	31
4.1.3	Vídeo largo	31
4.2	Métricas	32
4.3	Resultados	34
4.3.1	Validación de información de bajo nivel sin post-procesado	34
4.3.2	Validación de post-procesado de información de bajo nivel	36
4.3.3	Validación de información de alto nivel	39
5	Conclusiones y trabajos futuros	41
5.1	Conclusiones	41
5.2	Trabajos futuros	42
Bibliografía		44
Apéndices		45
A	Errores reconocidos en los resultados de los experimentos	47
A.1	Mala clasificación de pista completa en vídeos de Wimbledon	47
A.2	Carteles en la red que cortan las líneas de la pista	48
A.3	Momentos de golpeo acortados por mala detección de la pelota	49
A.4	Falsos momentos de golpeo por mala detección de la pelota	51
B	Etiquetado de vídeos con MatLab	53

LISTAS

Lista de ecuaciones

3.1	Cálculo de umbral máximo para distancia entre punto de pista actual y su anterior para considerarlo correcto	11
3.2	Coordenadas definitorias de la pista sobre el plano (en metros)	12
3.3	Descripción matemática de una bounding box	14
3.4	Ecuación para suavizar un punto (coordenada) de la pista	18
3.5	Ecuación para calcular el umbral máximo de distancia entre los pies de un jugador respecto de su última medición para ser considerada como buena	19
3.6	Ecuación para interpolar una coordenada en un frame en posición j, dadas una coordenada anterior en un frame en posición i y una coordenada posterior en un frame en posición k	19
3.7	Ecuaciones para calcular los umbrales máximo y mínimo de distancia entre dos detecciones de la pelota para ser considerada como buena	21
3.8	Cálculo de desventaja por desplazamiento para su uso en la fórmula de desventaja total del jugador contrario 3.11	27
3.9	Cálculo de desventaja horizontal para su uso en la fórmula de desventaja total 3.11 ..	28
3.10	Cálculo de ventaja vertical para su uso en la fórmula de desventaja total 3.11	28
3.11	Fórmula para calcular la desventaja total con la que se deja al siguiente golpeador ...	28

Lista de figuras

2.1	Comparación de frames enfocando la pista completa y otros enfoques	4
2.2	Arquitectura de una red TrackNet	6
3.1	Diagrama de flujo del sistema	8
3.2	Diagrama de flujo de la Parte I del sistema	9
3.3	Ejemplo de procesado de una imagen para obtener los 8 puntos de la pista	10
3.4	Plano cenital con las medidas estándares de una pista de tenis	13
3.5	Frame con personas detectadas sobre la imagen original con la máscara aplicada....	15
3.6	Frame con jugadores detectados sobre la imagen original con la máscara aplicada ...	15
3.7	Detección de personas y jugadores sin utilizar una máscara	16
3.8	Frame con la pelota detectada	17
3.9	Diagrama de flujo de la Parte II del sistema	18

3.10	Aproximación de coordenadas de pelota	21
3.11	Diagrama de flujo de la Parte III del sistema	22
3.12	Frames clasificados según si son de golpeo	23
3.13	Comparación de frames antes, durante y después de impacto	24
3.14	Frame con trayectoria de pelota dibujada	25
3.15	Frame del primer momento de golpeo después de un momento de no juego, clasificado como saque	25
3.16	Frames de los dos últimos momentos de golpeo antes de un momento de no juego, el primero siendo golpeo final y el segundo un passing	26
4.1	Ejemplo formación de vídeo con sets de frames aleatorios	30
4.2	Ejemplo de frames que forman el vídeo corto	31
4.3	Ejemplo de frames que forman el vídeo largo	32
4.4	Secuencia de un movimiento de golpeo de revés	32
4.5	Ejemplo de puntos de pista con IoU en el umbral de 0.85	33
A.1	Comprobación de enfoques a pistas de hierba	48
A.2	Frame con puntos de pista mal encontrados por haber carteles en la red	49
A.3	Ejemplo de mala detección de golpes provocada por mala detección de la pelota	50
A.4	Ejemplo de mala detección de saque provocada por mala detección de la pelota	52
B.1	Ejemplo del etiquetado con MatLab del vídeo de frames aleatorios	54

Lista de tablas

4.1	Resultados de la validación de la información de bajo nivel, procesando el vídeo de frames aleatorios	35
4.2	Comparación validación de puntos de pista con vídeo de frames aleatorios, sin y con correcciones	36
4.3	Comparación de modelos y umbrales en el reconocimiento de jugadores sobre el vídeo de frames aleatorios	36
4.4	Resultados de la validación de la información de bajo nivel con post-procesado, procesando el vídeo corto	37
4.5	Comparación validación de puntos de pista en vídeo corto	38
4.6	Comparación validación de jugadores en vídeo corto	38
4.7	Comparación validación de pelota en vídeo corto	38
4.8	Comparación de tiempo sin usar y usando el postprocesado en el vídeo corto	39
4.9	Resultados de validación de vídeo largo	39
4.10	Comparación validación de tipos de golpeo en vídeo largo	40

4.11 Comparación validación de tipos de golpeo en vídeo largo	40
---	----

INTRODUCCIÓN

En esta sección hablaremos sobre las motivaciones que nos han impulsado a realizar este proyecto y sobre los objetivos que fijaremos para nuestro sistema.

1.1. Motivación

Un buen jugador de tenis se basa en 4 pilares: la técnica, el físico, la concentración y la estrategia. Una de las formas más eficaces para mejorar esta última, a parte de jugar, es ver y analizar partidos de profesionales. El problema es que el tenis es un deporte muy rápido y complejo, y en la mayoría de situaciones puede ser difícil apreciar ciertos aspectos del juego que son los que acaban dictando que gane un jugador u otro. Por ello, una herramienta que nos permita analizar y visualizar las dinámicas de movimiento de los jugadores y direccionamientos de la pelota a lo largo del punto sería útil para estudiar las consecuencia de diferentes estrategias de posicionamiento en pista o de diferentes tipos de golpeo.

Actualmente existen diferentes tipos de tecnologías enfocadas al análisis deportivo del tenis. En retransmisiones de partidos profesionales se usan sistemas de obtención de estadísticas, TennisViz es el sistema utilizado por la ATP, que recogen datos y estadísticas como la puntuación, la cantidad de derechas y reveses, cantidad de golpes ganadores, errores no forzados, saques directos, etc [1]. Toda esta información se facilita a los locutores en tiempo real para mejorar la calidad de la información que transmiten y se cuelgan en sitios oficiales para que cualquier interesado las pueda consultar (la página oficial de la ATP para esto es: estadísticas ATP Tour). Para los clubes de tenis existen tecnologías parecidas que, a partir de varias cámaras instaladas en la pista, recopilan esta misma información para poder revisarla después de un partido.

El problema con estas tecnologías es que no aportan información acerca de las dinámicas antes mencionadas. Por ello la motivación principal de este proyecto es crear un sistema que represente visualmente sobre un mismo partido información a cerca de los golpes y de las consecuencias de posicionarse en pista o de direccionar la pelota para mover al jugador contrario.

1.2. Objetivos

Los objetivos principales de proyecto son:

- Diseñar e implementar un sistema capaz de procesar un vídeo de un partido de tenis:
 - Clasificando los frames en las que se está jugando un punto.
 - Detectando en los frames necesarios la pista, los jugadores y la pelota con suficiente precisión como para realizar cálculos posteriores.
 - Con la información extraída de los frames, reconocer secuencias de frames en los que se estén produciendo golpes y clasificarlos como: saque, ataque, defensa o final.
 - Representar visualmente las trayectorias de la pelota y los movimientos de los jugadores sobre la pista.
- Evaluar el sistema desarrollado analizando la diferente información extraída o calculada a lo largo del los diferentes procesados, con las métricas adecuadas.

ESTADO DEL ARTE

El objetivo principal del sistema es detectar los golpes de un punto y clasificarlos, así como representar la trayectoria de la pelota durante el golpeo. Para llegar a esta información haremos uso de diferentes técnicas del ámbito de la visión artificial para realizar detecciones sobre el vídeo de entrada y con ello realizar los cálculos necesarios que nos lleven a esta información.

En la Sección 1.1 hemos mencionado sistemas comerciales que obtienen también información sobre ciertos aspectos del juego, pero no son de código abierto, por lo que no podemos investigarlos y reutilizar partes de ellos. En cambio si existen proyectos abiertos en los que se utiliza visión artificial para detectar información en vídeos de partidos de tenis.

En esta sección hablaremos sobre tecnologías y sistemas existentes que compararemos e integraremos en nuestro proyecto para ayudarnos a extraer la información necesaria para hacer nuestros cálculos y procesamientos que nos lleven a la información objetivo. También mencionaremos los problemas que presentan los proyectos antes mencionados, que nos han llevado a la decisión de implementar nuestro propio sistema de detecciones.

2.1. Tecnologías de Visión Artificial existentes

Ahora expondremos diferentes tecnologías y técnicas de visión artificial que tendremos en cuenta o utilizaremos en nuestro proyecto.

2.1.1. Clasificación de imágenes según enfoque a pista

A lo largo de un partido de tenis es normal que la cámara vaya realizando diferentes enfoques o que se inserten repeticiones, anuncios, enfoques al público, etc. Queremos poder distinguir los enfoques en los que se está jugando un punto (figura 2.1(a)), y por tanto sobre los que querremos realizar ciertas detecciones o cálculos, del resto (figura 2.1(b)). La forma en la que se enfoca un punto en juego está estandarizada para las retransmisiones, se trata de un encuadre en el que aparece la pista completa, grabado desde detrás de uno de los jugadores a cierta distancia y altura. Para distinguir entre ambos

casos necesitaremos un clasificador binario de imágenes.



(a) Frame enfocando a pista completa



(b) Frame no enfocando a pista completa

Figura 2.1: Comparación de frames enfocando la pista completa y otros enfoques

En el proyecto [2] hay ya implementada una arquitectura de red convolucional (CNN) que incorporaremos a nuestro proyecto para realizar esta clasificación. Esta arquitectura consta de dos capas convolucionales 2D, con sus capas max pooling correspondientes, una capa aplanadora, una capa densa de activación con ReLU, una capa de DropOut y una capa final densa de activación sigmoidal.

El proyecto no consta de un modelo entrenado, pero si de varios métodos que nos permiten clasificar manualmente frames de un vídeo y entrenar el modelo.

Para entrenar el modelo se han usado 473 imágenes, de las cuales un 47% están etiquetadas como enfoques a pista completa. Las imágenes se han obtenido de vídeos con diferentes tipos de pista, en los que se usan diferentes ángulos de enfoque y con diferentes calidades de imagen (aun que se transforman a 720p para el procesado). Utilizamos Binary Cross Entropy como función de pérdida y Adam como optimizador.

En la tabla 4.1 se pueden ver los resultados de la validación de este clasificador.

2.1.2. Transformaciones de Hough para detección de líneas de pista

Para reconocer las líneas de la pista utilizaremos las transformada probabilística de Hough para líneas. Estas transformaciones son un conjunto de técnicas y ecuaciones que permiten detectar formas geométricas simples en una imagen. Esta detección consiste en coger la definición matemática de una forma geométrica, líneas en nuestro caso, y calcular la cantidad de veces que un píxel pertenece a una de estas formas geométricas (a lo que llamamos votos). Guardando estos valores en una matriz de acumulación podremos buscar picos de votos, que corresponderán a los píxeles que forman la forma geométrica. [3]

En el proyecto [4] está implementada esta técnica, utilizando un umbralizado de la imagen para mantener solo los píxeles blancos (color de las líneas de la pista), y a estos se les aplica la transformada de Hough para encontrar las líneas. Integraremos estas técnicas en un módulo propio para la detección

de los puntos definitorios de la pista, como se explica en la Sección 3.2.2.

Al poner a prueba esta detección del proyecto [4] se han observado los siguientes problemas:

- Problemas con imágenes de menor calidad: en ciertas condiciones, las líneas de la pista pueden ser tan finas que no se detecten.
- Problemas con ángulos de enfoque más variados: si las líneas de la pista quedan con mayor ángulo respecto del eje vertical no se reconocen.
- Problemas con enfoques en movimiento: al moverse el enfoque de la cámara, los ángulos de las líneas cambian y pueden no reconocerse.
- Baja consistencia: al utilizar secuencias largas de frames, en muchos no es posible realizar la detección por diferentes causas, resultando en numerosas pérdidas.

Para resolver estos se ha tenido que trabajar en reajustar los parámetros del umbralizado de los píxeles, de la transformada de Hough. Además, se han tenido que crear nuevos métodos agrupación y filtrado de líneas y métodos que detecten y corrijan resultados erróneos y pérdidas. Estos procesos están explicados en detalle en la Sección 3.2.2.

2.1.3. Homografías para localizar coordenadas en el plano de la pista

En geometría, las homografías se utilizan para relacionar geométricamente dos planos y cualquier punto sobre ellos. Esto puede utilizarse en visión artificial para relacionar coordenadas o píxeles entre dos imágenes o entre una imagen y un plano imaginario. En nuestro caso las utilizaremos para trasladar coordenadas de la imagen al plano cenital de la pista y viceversa. De esta forma podremos localizar a los jugadores sobre el plano de la pista o trasladar puntos del plano de la pista a la imagen y realizar cálculos como medidas de distancias reales.

2.1.4. Detección de personas en imágenes

Mientras se esté jugando un punto, una de las informaciones que queremos extraer de los frames es la posición en cada imagen de los jugadores. Para ello se han explorado diferentes modelos y arquitecturas de reconocimiento de objetos en imágenes, personas en nuestro caso.

Faster R-CNN Las redes R-CNN son una de las principales arquitecturas utilizadas en la detección de objetos en imágenes. Estas son redes neuronales convolucionales basadas en regiones, capaces de dividir la imagen en regiones de interés y, después, clasificar su contenido. Mejorando la forma en la que se generan y seleccionan las regiones de interés, aparecen Fast R-CNN y luego Faster R-CNN. Las ventajas del uso de este modelo son la precisión, la flexibilidad en el tamaño y forma de los objetos a detectar y un buen manejo de objetos superpuestos. Un proyecto que utiliza este tipo de modelo es [4].

YOLO Los modelos YOLO (You Only Look Once) se diferencian de otros por enfocarse en el rendimiento, sacrificando precisión. YOLO trabaja dividiendo con una cuadrícula una imagen en celdas, luego sobre estas celdas se predice la probabilidad de que contenga alguna clase de objeto y se filtran para obtener las bounding boxes con mayor confianza. Aun que obtengan resultados en general menos precisos, el alto rendimiento y velocidad hacen que sea una opción muy popular, en especial para la identificación de objetos en tiempo real. [5]

Comparación Faster R-CNN vs YOLO Para la elección de arquitectura de detección de personas se han comparado un modelo Faster R-CNN RestNet-50 FPN [6] y YOLOv5 de Ultralytics [7]. En la tabla 4.3 se muestran los diferentes resultados de nuestro sistema en función del modelo y umbral de confianza utilizado. Se puede ver que con YOLO llegamos a obtener mejores resultados con un tiempo de procesamiento considerablemente más bajo, por ello es el que utilizaremos en nuestro proyecto.

2.1.5. Detección de pelota con TrackNet

Entre los objetos que queremos detectar en cada frame se encuentra la pelota. La detección de objetos pequeños y rápidos en un vídeo es un reto para los sistemas de detección convencionales, como los mencionados en la Sección 2.1.4. Para resolver esto existe una arquitectura de entrenamiento profundo diferente desarrollada con este objetivo, TrackNet [8], con la arquitectura mostrada en la figura 2.2 . En el proyecto de GitHub [9] hay una implementación de esta arquitectura, basada en el artículo [8], que adaptaremos para ser usada en nuestro sistema.

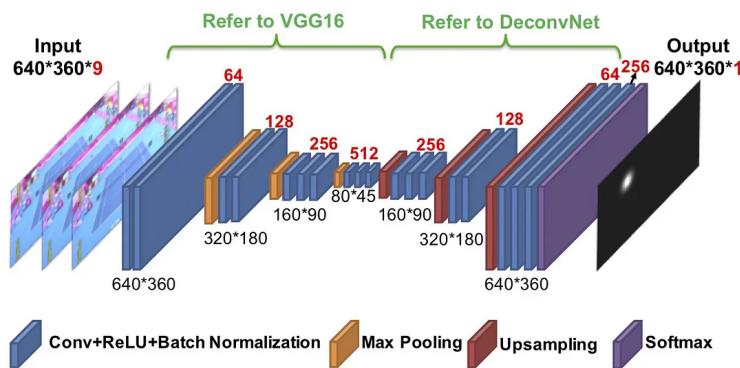


Figura 2.2: Arquitectura de una red TrackNet (obtenido de [10])

DESCRIPCIÓN DEL SISTEMA

En esta sección se describirá en detalle el funcionamiento del sistema, los módulos que lo forman y el procesamiento que se realiza en cada uno de ellos en las diferentes etapas.

3.1. Esquema general

El objetivo de este proyecto es detectar los golpeos y clasificarlos según si son saque, de ataque, de defensa o de finalización. Además, para ayudarnos a visualizar las dinámicas entre el direccionamiento de los golpeos y el posicionamiento en pista, queremos dibujar la trayectoria que va a tener la pelota en cada golpeo.

Esta información la calcularemos a partir de las posiciones de los jugadores y de la pelota sobre la pista o sobre las imágenes que forman el vídeo de entrada. Esta información, a su vez, la obtendremos realizando detecciones sobre los frames.

Para realizar este procesamiento se ha dividido el sistema en tres partes:

- 1.– En una primera parte analizaremos todas las imágenes que forman el vídeo clasificando qué frames deben ser procesados y, sobre ellos, realizando las detecciones de la pista, jugadores y pelota. A esta información la calificaremos como de **bajo nivel**.
- 2.– En la segunda parte realizaremos un **post-procesado** de la información de bajo nivel, que constará de detecciones y correcciones de errores, así como de varias interpolaciones que nos permitirán liberar a la parte anterior de carga computacional.
- 3.– En la tercera parte realizaremos los cálculos necesarios para llegar a la información objetivo acerca de los golpeos. A la información obtenida en esta parte la llamaremos de **alto nivel**.

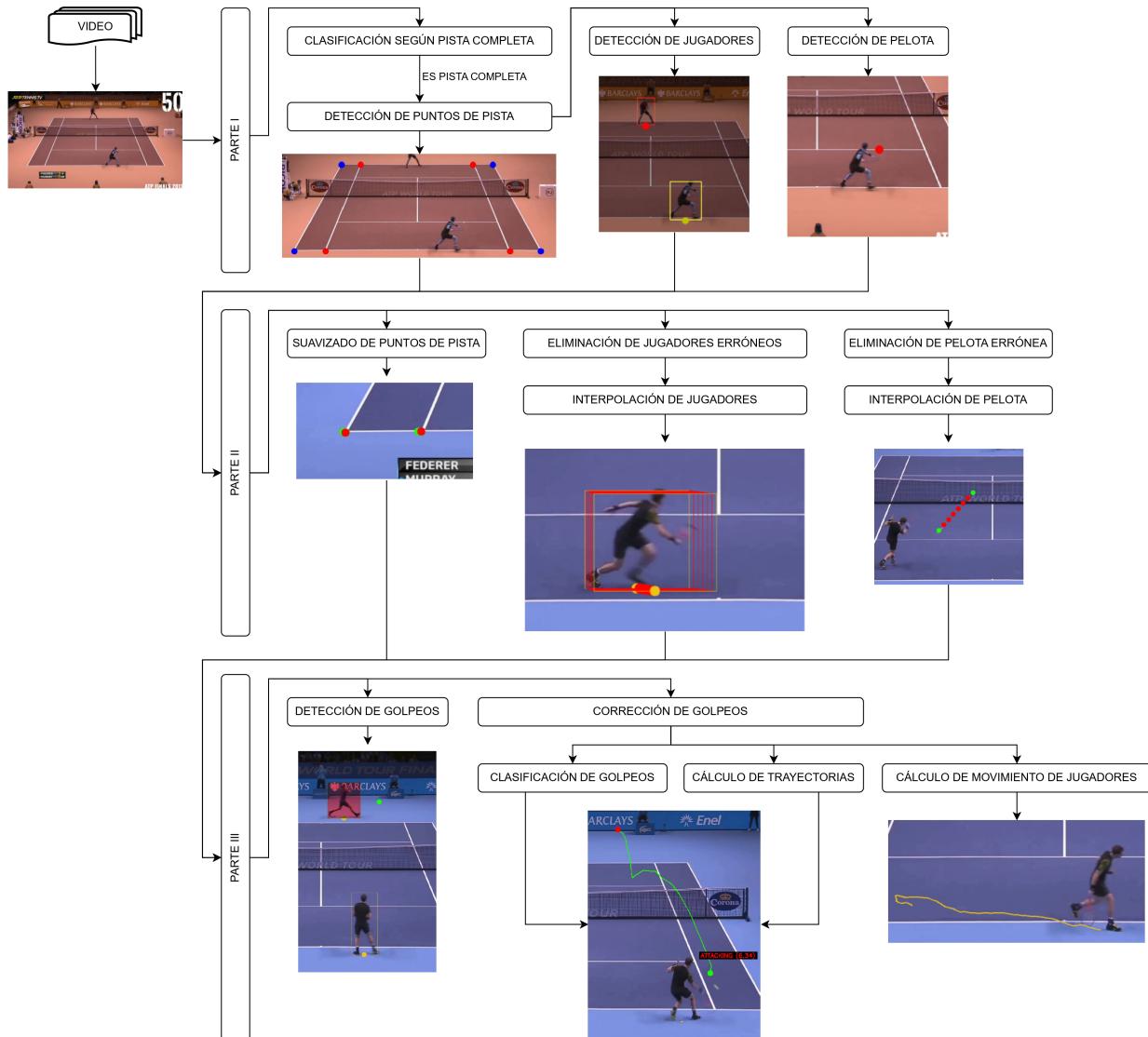


Figura 3.1: Diagrama de flujo del sistema, dividido por partes del procesado, en los que se muestra qué módulo intervienen en cada parte y un ejemplo visual de la información resultante.

Conforme vamos procesando frames en las diferentes partes del sistema, iremos guardando junto a ellos la información que vayamos obteniendo de cada uno, de forma que sea accesible desde otros módulos y durante el procesamiento de otros frames. Esto será necesario para realizar diferentes cálculos, detecciones y comprobaciones.

En las siguientes subsecciones hablaremos detalladamente sobre estas partes y módulos. Por simplificación, al hablar de píxeles y distancias en imágenes estaremos hablando sobre imágenes en resolución 720p.

3.2. Parte I: detecciones y procesado de bajo nivel

La entrada a nuestro sistema es un vídeo, en esta Parte I recorremos todos los frames que lo forman y realizaremos las clasificaciones y detecciones (y en algunos casos ciertas correcciones) con las que obtendremos la información de bajo nivel. Para esto cada frame atravesará de forma secuencial los diferentes módulos que intervienen en esta parte.

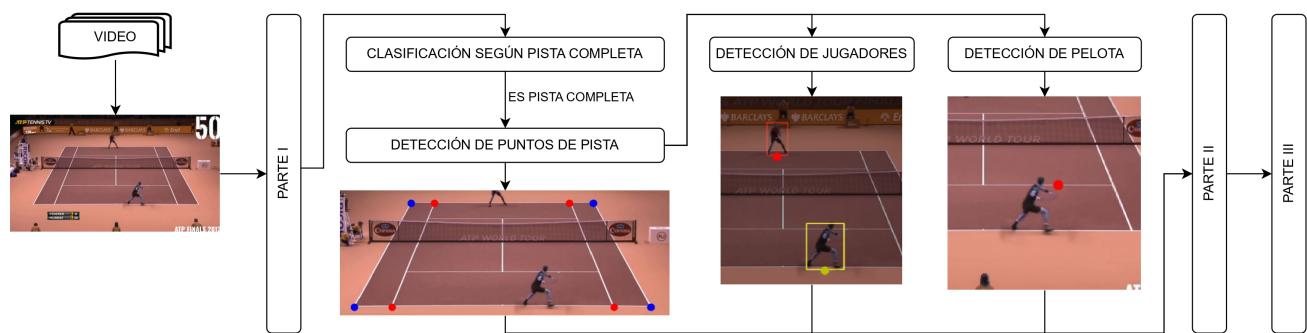


Figura 3.2: Diagrama de flujo de la Parte I del sistema

3.2.1. Clasificación de frame según pista completa

El primer módulo por el que pasa cada frame es el de clasificación según se esté enfocando la pista completa (como la figura 2.1(a)) o no (como la figura 2.1(b)). Esto es así ya que, durante la retransmisión, solo se enfoca la pista completa cuando se está jugando un punto, y no queremos procesar otros momentos del vídeo. Por ello si en este módulo el frame es clasificado como “no enfoque a pista completa” se marcará y no será procesado por ningún otro módulo del sistema. En caso contrario, se marcará como “enfoque a pista completa” y se continuará con su procesado.

Al haber atribuido “1” a enfoques a pista y “0” a lo contrario al entrenar nuestro modelo, si obtenemos de él un resultado mayor que 0.7 calificaremos el frame como pista completa, si es menor que 0.3 como de lo contrario. Si el resultado se encuentra entre estas dos cotas, calificaremos el frame actual como del mismo tipo que el frame anterior. Podemos hacer esto ya que, estudiando el comportamiento del modelo, se ha visto que estos casos ocurren puntualmente y de esta forma una secuencia de frames de la misma clasificación no queda cortada por uno de estos frames ambiguos.

3.2.2. Detección de puntos de pista

En este módulo se obtendrán las coordenadas que definen la pista de tenis en la imagen, estos son los 4 puntos que encuadran la pista de juego individual y los 4 para los dobles. En la figura 3.3 se ve un ejemplo del procesado aplicado en la imagen de un frame para llegar a los 8 puntos de la pista,

visibles en la figura 3.3(e).

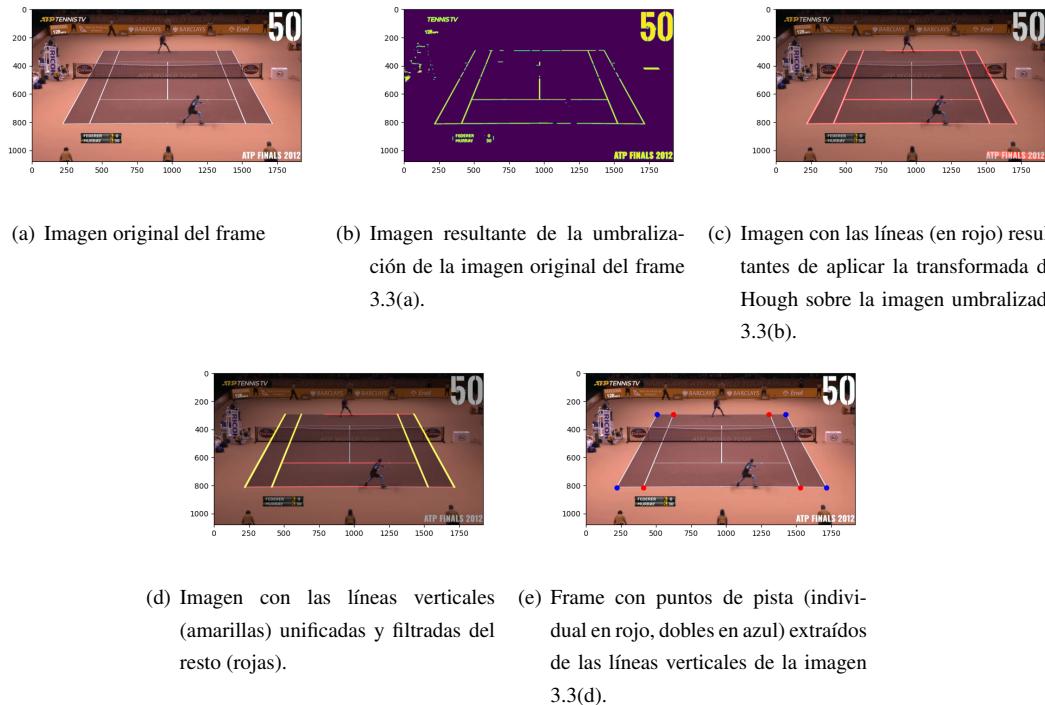


Figura 3.3: Ejemplo de procesado de una imagen para obtener los 8 puntos de la pista.

Para obtener estos 8 puntos aplicaremos las siguientes técnicas, partiendo de la imagen original:

Detección de líneas Detectaremos las líneas de la pista en la imagen original siguiendo los siguientes pasos:

1.– **Umbralizado:** Convertiremos los píxeles de la imagen original a una escala de grises y aplicaremos un umbralizado, de forma que todo píxel con valor mayor a 200 quede en blanco (255) y el resto negro (0). De esta forma los píxeles que forman las líneas blancas de la pista quedan por encima del umbral (junto con otras partes de la imagen), como se ve en la figura 3.3(b).

2.– **Transformada de Hough:** Sobre la imagen umbralizada aplicaremos la transformada probabilística de Hough para detectar las líneas presentes en ella. Como resultado se obtienen una lista de líneas (como las dibujadas en la figura 3.3(c)), definidas por sus coordenadas inicial y final.

3.– **Filtrado por pendiente:** Sobre las líneas aplicaremos un filtrado para quedarnos con las que tengan una pendiente respecto del eje horizontal mayor que 1,3. En la figura 3.3(d) se observa esta separación.

4.– **Agrupación y unificación:** La transformada de Hough puede habernos dado diferentes líneas superpuestas para una misma “línea real”, por lo que realizaremos una agrupación y luego una unificación:

4.1.– **Agrupación:** Comprobaremos para cada par de líneas si existe un punto de corte entre ellas y las agruparemos de tal forma que una línea pertenezca a un grupo si esta corta al menos a una de las integrantes.

4.2.– **Unificación:** Para cada grupo de líneas aplicaremos una unificación de forma que obtengamos una única resultante, que tendrá como coordenada de origen la

más baja (en la imagen) de las integrantes y como coordenada final la más alta.

5.– **Filtrado por longitud:** Filtraremos las líneas resultantes para quedarnos solo con las que tengan una longitud mayor que un 0.35 de la altura de la imagen (70 píxeles para una image de 720p).

Comprobación de número de líneas En este punto se contemplan tres posibilidades en función del número de líneas que hayamos obtenido:

- **0 líneas:** se considerará que no es posible realizar una corrección y daremos la detección como perdida. En consecuencia, no se continuará con el procesado de este frame durante el resto de módulos del sistema.
- **4 líneas:** consideraremos que las líneas obtenidas se corresponden con las líneas de la pista, aun que pueden haber quedado movidas, cortadas o distorsionadas. Por ello, si este es el caso, pasaremos a comprobar los puntos que definen nuestras líneas en la siguiente fase: **Comparación de puntos de pista**.
- **Número de líneas diferente de 4 y 0:** existe la posibilidad de haber detectado más o menos líneas que las deseadas. En este caso pasaremos directamente a la fase de **Comparación de líneas**.

Comparación de puntos de pista Habiendo obtenido 4 líneas, tomaremos sus puntos definitorios (punto inicial y punto final de cada una) como los puntos de la pista. En esta fase debemos comprobar que estos puntos sean correctos, ya que existen diferentes situaciones que pueden provocar que no lo sean. Estudiando el comportamiento de esta detección se han distinguido los siguientes:

- Los jugadores, al moverse, pueden aparecer en las imágenes por delante de las líneas de la pista, provocando que alguna línea quede acortada. Esto suele suceder más con el jugador inferior.
- El marcador puede quedar cerca o encima de alguna línea de la pista y hacer que la línea detectada quede más larga.

Estas razones son las situaciones más comunes que provocan un error momentáneo en la detección. Para corregirlos en esta fase, se compararán los puntos definitorios actuales con los puntos obtenidos en el frame anterior. Para realizar estas comparaciones y otros cálculos mantendremos siempre la misma ordenación de los puntos a la hora de guardarlos como información relacionada con cada frame: antihoraria comenzando por el punto inferior izquierdo.

La comparación consiste en calcular las distancias euclídeas entre cada par de puntos “actual-anterior” y comprobar si alguna supera un umbral:

para $POINTS = \text{"conjunto de puntos actuales"}$

siendo $d_A = \text{"distancia euclídea de punto A con su anterior"}$

$$\text{umbral}_A = 3 \cdot \frac{\sum_{i \text{ in } POINTS \setminus \{A\}} d_i}{7} \quad (3.1)$$

Es decir, para unos puntos A, B, ..., H, el umbral para A es igual a 3 por la media de las

distancias euclídeas de los puntos B, ..., H respecto de sus anteriores. De esta forma si todos los puntos se mueve (por que el enfoque de la cámara se mueva) no detectaremos puntos erróneos, pero si algún punto se mueve drásticamente (por alguna de las razones explicadas anteriormente) sí lo detectaremos. Si algún punto supera su umbral, lo daremos por erróneo y lo marcaremos como tal asignándole un valor nulo para tenerlo en cuenta en futuras fases.

Cabe decir que la corrección realizada en esta fase solo es posible si el frame anterior consta de 8 puntos de pista detectados correctamente. En caso contrario, consideramos como mejor opción dar los puntos actuales por buenos antes que dar la detección por perdida, ya que los podremos seguir utilizando en procesamientos posteriores (aun que obtengamos resultados menos precisos).

La siguiente fase será **Cálculo de transformadores homográficos**.

Comparación de líneas En el caso de haber obtenido un número diferente a 4 de líneas en la fase **Comprobación de número de líneas**, ahora averiguaremos cuáles faltan o sobran. Esto lo haremos comparándolas con las líneas del frame anterior. Al igual que en la fase **Comprobación de puntos**, si el frame anterior no consta de los 8 puntos detectados correctamente no podemos realizar esta corrección. En este caso sí tendremos que dar la detección del frame por perdida y no procesaremos este frame en ningún módulo más del sistema.

La comprobación de esta fase consiste en intentar emparejar cada línea anterior con una de las actuales. Para ello, para cada línea anterior buscaremos una línea actual que tenga alguno de sus puntos definitorios a una distancia menor o igual que un 2 % de la altura de la imagen. Si no la encontramos, le asignaremos una línea “nula”. De esta forma acabaremos con una línea actual para cada línea anterior, que al estar ordenadas de izquierda a derecha, las actuales también lo estarán. Tras esto, si no tenemos al menos 2 líneas no nulas daremos por perdida esta detección y no procesaremos este frame en ningún módulo más del sistema.

Ahora, de las 4 líneas obtendremos los 8 puntos de la pista, asignando puntos nulos a los correspondientes a líneas nulas. Es decir, si por ejemplo la segunda línea es nula, los puntos resultantes serán $p_1, null, p_3, p_4, p_5, p_6, null, p_8$.

La siguiente fase será **Cálculo de transformadores homográficos**.

Cálculo de transformadores homográficos Las medidas de una pista de tenis están estandarizadas, por lo que podemos relacionar puntos de la pista en la imagen con puntos de la pista sobre un plano. Según las medidas oficiales de la Federación Internacional de Tenis [11], obtendremos las 8 coordenadas que los puntos obtenidos en fases previas tendrían en un plano cenital alto nivelde la pista (en metros):

$$[(4, 30, 97), (5, 37, 30, 97), (13, 6, 30, 97), (14, 97, 30, 97), (14, 97, 8), (13, 6, 8), (5, 37, 8), (4, 8)] \quad (3.2)$$

Ahora tenemos 8 pares de coordenadas “punto en la imagen - punto sobre el plano”, de los que descartaremos los pares con puntos de la imagen nulos. Para poder continuar con el procesado necesitamos obtener al menos 4 pares no nulos. Con estos pares restantes podemos hacer uso de la librería cv2 para obtener dos matrices transformadoras [12] que nos permitan trasladar puntos del plano de la pista a la imagen (siempre que se encuentren sobre la superficie, que no tengan altura) y viceversa.

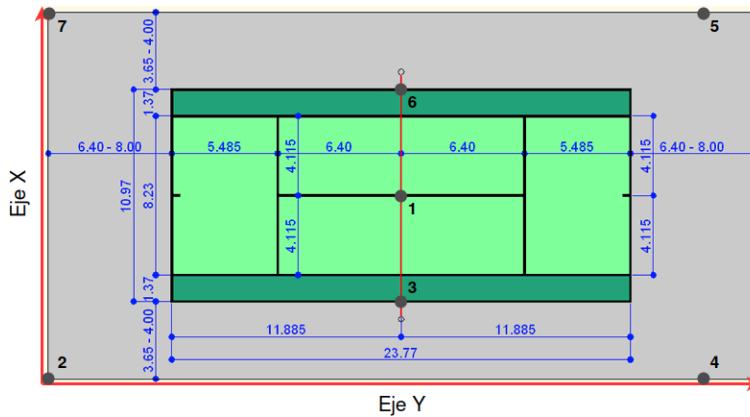


Figura 3.4: Plano cenital con las medidas estándares de una pista de tenis. Adaptado de [13]

Reconstrucción de puntos perdidos En las fases de **Comparación de puntos de pista** y **Comparación de líneas** puede que hayamos declarado algunos puntos como nulos. Ahora que tenemos los transformadores homográficos, podemos reconstruir estos puntos conociendo sus coordenadas sobre el plano (figura 3.2), obteniendo los 8 puntos de la pista.

Resultados de este módulo De este módulo guardaremos en la información referente al frame actual las 8 coordenadas de los puntos de la pista y las 2 matrices transformadoras homográficas. Si en algún momento requerimos conocer las líneas de un frame, basta con reconstruirlas a partir de los 8 puntos sabiendo que se han guardado de forma ordenada.

3.2.3. Detección de jugadores

En este módulo realizaremos la detección de los jugadores sobre la imagen original con un modelo pre-entrenado de detección de objetos con arquitectura YOLOv5 [7].

En este y en otros módulos del sistema queremos poder situar a las personas sobre la superficie de la pista en la imagen, para ello utilizaremos lo que hemos calificado como los **pies de los jugadores** (o de personas). Esto es el punto medio del lado inferior del bounding box de la persona, y asumiremos

que siempre está a la altura del suelo. De este modo podemos comparar localizaciones, calcular distancias y trasladar la posición de los jugadores al plano cenital con las transformaciones homográficas obtenidas en la Sección 3.2.2.

Ahora describiremos los pasos que se han seguido para realizar la detección de los jugadores:

Enmascaramiento Queremos aplicar una máscara sobre la imagen original para tapar ciertas partes que nos puedan dar problemas en detecciones o cálculos posteriores. Para crear una máscara específica para cada imagen, trasladaremos ciertas coordenadas del plano cenital a la imagen con las homografías:

- Puntos de corte entre la red y los laterales exteriores de los pasillos: (4, 19'885), (14'97, 19'885). Correspondientes a los puntos 3 y 6 de la imagen 3.4.
- Puntos del fondo superior de la pista: (0, 0), (18,97, 0). Correspondientes a los puntos 2 y 7 de la imagen 3.4.
- Puntos “casi” del fondo inferior de la pista: (0, 36), (18,97, 36). Correspondientes a los puntos 4 y 5 de la imagen 3.4. La razón para escoger estas coordenadas y no el fondo “real” es que al trasladarlas a la imagen original, quedan fuera de esta. En cambio, al escoger estas coordenadas, aproximadamente 4 metros por fuera de la línea de fondo de la pista, al trasladarlas a la imagen original conseguiremos tapar al público que aparece en la parte inferior.

Trasladaremos estas coordenadas con la homografía a la imagen y haremos los siguientes ajustes a los resultantes:

- A los puntos 2 y 7, dividiremos sus componentes Y entre 2, añadiendo un margen que evite que la máscara corte o dificulte la detección del jugador superior.
- A los puntos 4 y 5, sustituiremos sus componentes X por 0 y el ancho de la imagen respectivamente, de forma que llevemos estos puntos a los extremos horizontales de la imagen. De nuevo, para evitar tapar áreas en las sí queremos realizar la detección.

Con estas coordenadas crearemos dos máscaras, definidas por los circuitos de puntos:

- $4 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow (\text{ancho maximo}, 0) \rightarrow (0, 0) \rightarrow 4$
- $(0, \text{altura maxima}) \rightarrow (\text{ancho maximo}, \text{altura mxima}) \rightarrow 5 \rightarrow 4 \rightarrow (0, \text{altura maxima})$

La máscara resultante quedaría como se ve en la imagen 3.5.

Pre-procesado de la imagen Para poder utilizar nuestra imagen enmascarada con el modelo hace falta realizar un pre-procesado. Este consiste en convertir la imagen del espacio de color BGR al espacio RGB.

Detección de personas Sobre la imagen pre-procesada aplicaremos nuestro modelo de detección de personas. Este nos dará como resultado una lista de bounding boxes, dibujados en la imagen 3.5. Estos se definen por dos coordenadas opuestas, la inferior izquierda y la superior derecha:

$$\text{bounding box} = [x_{izquierda}, y_{inferior}, x_{derecho}, y_{superior}] \quad (3.3)$$



Figura 3.5: Frame con personas detectadas sobre la imagen original con la máscara aplicada, en rojo las personas en la mitad superior de la pista, en amarillo las personas en la mitad inferior

Clasificación de personas por altura en la imagen Ahora dividiremos los resultados según se encuentren en la mitad superior o inferior de la pista. Para ello trasladaremos el punto medio del plano cenital de la pista, con coordenadas (9,485, 19,885) (punto 1 de la imagen 3.4), a la imagen con las homografías. Diremos que todo bounding box con pies por encima de esta coordenada trasladada se encuentra por encima de la red, y los que tengan los pies por debajo que se encuentran por debajo de la red. En la imagen 3.5 se puede observar esta clasificación.

Distinción de jugadores del resto de personas Para distinguir a los jugadores entre todas las personas detectadas, diremos que el bounding box de cada mitad con pies más cerca de la red (coordenada trasladada anteriormente) es el jugador.

Podemos realizar esto ya que:

- Los jueces de línea de fondo, los recoge pelotas de fondo y el público de fondo siempre quedan por detrás de cada jugador en su parte del campo.
- El juez de silla, los jueces de línea laterales, los recoge pelotas de la red y el público lateral ha sido tapado por la máscara



Figura 3.6: Frame con jugadores detectados sobre la imagen original con la máscara aplicada, en rojo el jugador superior, en amarillo el inferior, con los pies de cada uno marcados con puntos

En la figura 3.7 se puede ver cómo habría sido la detección de personas y jugadores sin hacer uso de la máscara, en la que las personas de cada mitad más cerca de la red son,

en este caso, los recoge pelotas de la red.



(a) Personas detectadas en la imagen original sin máscara



(b) Recoge pelotas detectados como jugadores en la imagen original sin máscara

Figura 3.7: Detección de personas y jugadores sin utilizar una máscara, reconociendo a los recoge pelotas de la red como jugadores

Guardaremos en la información del frame actual los bounding boxes y coordenadas de los pies de ambos jugadores.

Acción ante pérdidas Durante esta detección, aun aplicando la máscara, normalmente encontramos más personas a parte de los jugadores en cada mitad de la pista. Por ello, si no se ha detectado al jugador real habrá otra persona en su mitad a la que calificaremos como tal, por lo que tendremos un sistema que raramente tendrá pérdidas. En el excepcional caso no daremos el frame por perdido, sino que lo marcaremos como “pendiente de interpolar jugadores” y continuaremos con su procesamiento por el resto de módulos del sistema. En el módulo 3.3.3 interpolaremos la posición de los jugadores de este frame.

Preparación para interpolaciones Este módulo es computacionalmente costoso, por ello, al recorrer el vídeo en esta parte del sistema solo procesaremos con él 1 frame de cada 6, es decir, procesaremos 1 frame y después dejaremos 5 sin procesar. Estos los marcaremos como “pendiente de interpolar jugadores” y en el módulo 3.3.3 interpolaremos la posición de los jugadores de estos frames. En el caso de que en el frame que toca ser procesado se produzca una pérdida, el siguiente sí se procesará y repetiremos esto hasta lograr procesar un frame correctamente. Luego dejaremos 5 sin procesar y repetiremos el ciclo.

3.2.4. Detección de pelota

En este módulo utilizaremos un modelo TrackNet adaptado del proyecto [9], como se menciona en la Sección 2.1.5, para encontrar la pelota en el frame.

Pre-procesado de la entrada del modelo La entrada a este modelo no es 1 imagen, sino una concatenación de 3. Las imágenes elegidas son la del frame actual, la del primer frame anterior y la del tercer frame anterior, en ese orden. Antes de la concatenación se deben redimensionar las imágenes a una resolución de 640x360. El método para hacer esto también

se ha extraído del proyecto [9].

Resultado del modelo El resultado del modelo será una coordenada sobre la imagen re-dimensionada del frame actual, que tendremos que re-dimensionar de vuelta al tamaño de imagen original. Guardaremos esta coordenada en la información referente al frame actual.



Figura 3.8: Frame con la pelota detectada (punto rojo)

Si el modelo no ha sido capaz de realizar la detección, devolverá una valor nulo. En tal caso marcaremos el frame actual como “pendiente de interpolar pelota” y continuaremos con su procesamiento por el resto de módulos del sistema. En el módulo 3.3.5 interpolaremos la coordenada de la pelota para este frame.

Preparación para interpolaciones Este módulo representa la mayor parte del coste computacional del sistema, por ello utilizaremos la misma técnica que con el módulo 3.2.3, en este caso dejando 4 frames sin procesar y calificándolos como “pendiente de interpolar pelota”. La comparación en tiempo de procesado por frame usando esta técnica o no en la detección de la pelota se puede ver en la tabla 4.7.

3.3. Parte II: post-procesado de bajo nivel

Los objetivos de este módulo son:

- 1.– Realizar ajustes a los puntos de la pista para mejorar la precisión y reducir el ruido de las detecciones.
- 2.– Comprobar la información acerca de los jugadores y la pelota para encontrar detecciones erróneas
- 3.– Realizar las interpolaciones de las posiciones de los jugadores y de la pelota en los frames no procesados, con pérdidas o con detecciones erróneas.

Para ello, los módulos descritos en esta sección realizaran una o varias vueltas al vídeo, frame a frame, esta vez no procesando las imágenes sino a la información de bajo nivel asociada a cada uno de ellos. Tanto para el reajuste de los puntos como para las comprobaciones e interpolaciones necesitaremos acceder a información de los frames de alrededor (anteriores y posteriores) del que se esté procesando. Ahora se describirán estos módulos y los post-procesados que realizan.

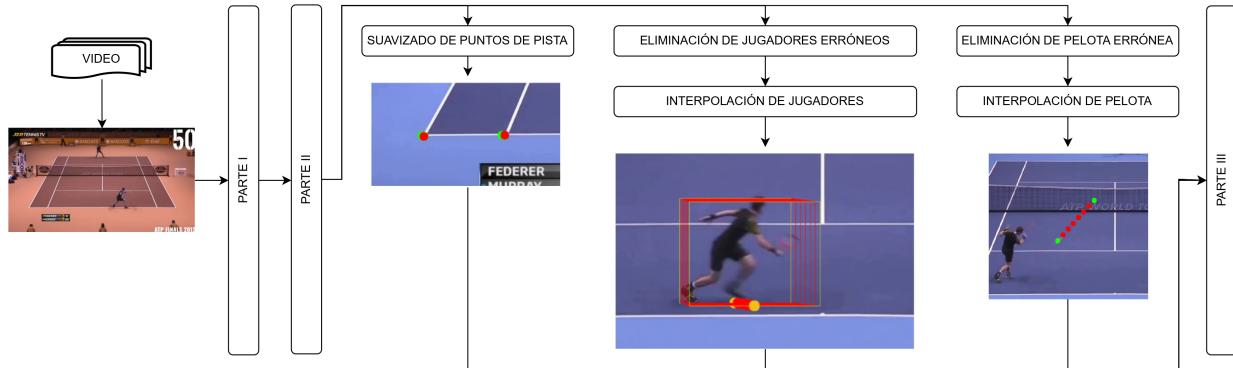


Figura 3.9: Diagrama de flujo de la Parte II del sistema

3.3.1. Suavizado de puntos de pista

Respecto a los puntos de la pista, ya se han hecho las comprobaciones y correcciones de errores en el módulo 3.2.2. Tras representar visualmente estos puntos en el vídeo, se ha observado que contienen mucho ruido, es decir, los puntos parecen vibrar continuamente a lo largo de la reproducción. Esto lo solucionaremos con el suavizado desarrollado en este módulo.

El suavizado de un punto consiste en calcular la media de las coordenadas de ese punto en un rango de 9 frames consecutivos: los 4 anteriores, el actual, y los 4 posteriores. Asignaremos esta nueva coordenada al punto.

siendo $court_point_j$ un punto en el frame j
y $court_point_{j-1}$ el mismo punto en el frame anterior
y $court_point_{j+1}$ el mismo punto en el frame siguiente

$$court_point_j = (x_j, y_j) = \left(\frac{\sum_{i=j-4}^{j+4} x_i}{9}, \frac{\sum_{i=j-4}^{j+4} y_i}{9} \right) \quad (3.4)$$

Para aplicar esto, recorreremos el vídeo frame a frame aplicando la fórmula 3.4 a cada uno de los 8 puntos de pista cuando sea posible (cuando se han detectado los puntos en los 9 frames consecutivos).

Podemos usar esta técnica ya que:

- Si la pista está quieta en el vídeo, hacer esta media reducirá el ruido y aumentará la precisión de las coordenadas.
- Si la pista se está moviendo en el vídeo, se mueve de una forma suficientemente lenta como para que la dispersión provocada por el ruido sea mayor que la del desplazamiento de la pista. Además, al escoger un rango tan corto de frames, el desplazamiento real del punto es aproximadamente constante y lineal. Esto nos resultará en unas coordenadas más precisas que las detectadas inicialmente.

3.3.2. Eliminación de jugadores erróneos

En este módulo, como parte del post-procesado, se comprobarán qué detecciones de jugadores son erróneas. Para ello se recorrerá el vídeo del mismo modo que en el módulo 3.3.1, frame a frame pero solo analizando la información de bajo nivel asociada a cada uno y no las imágenes. En cada uno de ellos consideraremos la detección de alguno de los jugadores como mala si la coordenada de sus pies se encuentra a una distancia mayor a un umbral respecto de la coordenada en el último frame disponible.

Como en el módulo 3.2.3 se han dejado frames sin procesar o pueden haber ocurrido pérdidas, es posible que el frame inmediatamente anterior al actual no contenga una detección de los jugadores. Por ello se utilizará el último frame con esta información y la distancia umbral se calculará como:

siendo $pies_i$ los pies de un jugador en el $frame_i$
y el $frame_j$ el último frame con detecciones de jugadores disponibles

$$umbral_i = 60 * (i - j) \quad (3.5)$$

Es decir, consideraremos que un jugador no ha podido moverse más de 60 píxeles entre cada frame, y cualquier detección que la supere se considerará como errónea. En este caso eliminaremos la detección de ese jugador y marcaremos el frame como “pendiente de interpolar jugadores”.

El sistema continuará su procesado con el módulo 3.3.3

3.3.3. Interpolado de jugadores

En los módulos 3.2.3 y 3.3.2 hemos etiquetado frames como “pendiente de interpolar jugadores”. Ahora recorreremos el vídeo y para cada frame así etiquetado:

- 1.– Obtendremos el último frame que sí tenga una detección, con posición i (sin tener en cuenta posibles frames ya interpolados).
- 2.– Obtendremos el siguiente frame que sí tenga una detección, con posición k .
- 3.– En los frames i y k , para cada jugador, calcularemos las 2 coordenadas que definen su bounding box (según la fórmula 3.3), obteniendo 2 puntos iniciales: p_{i_1}, p_{i_2} ; y 2 finales: p_{k_1}, p_{k_2}
- 4.– En el frame actual j , para cada jugador y para cada una de estas 2 coordenadas, interpolaremos el punto actual p_j según la fórmula 3.6. Con las 2 coordenadas resultantes definiremos el bounding box para el jugador y calcularemos sus pies.
- 5.– Guardaremos el bounding box y pies de los jugadores junto con la información respectiva al frame.

$$p_j = (x_j, y_j) \quad x_j = x_i + (j - i) \cdot \frac{x_k - x_i}{k - i} \quad y_j = y_i + (j - i) \cdot \frac{y_k - y_i}{k - i} \quad (3.6)$$

3.3.4. Eliminación de pelota errónea

Este módulo realizará el primer post-procesado sobre las detecciones de la pelota: detección y eliminación de errores.

Para detectar si una coordenada de la pelota es errónea nos basaremos en que la pelota se mueve por la imagen en un rango de velocidades (píxeles por frame) que podemos acotar tanto por encima como por debajo. Podemos afirmar por razones obvias que existe una cota superior, y podemos explicar que existe una inferior ya que durante el punto en juego la pelota siempre va a estar en movimiento.¹

Estudiando visualmente los resultados de del módulo 3.2.4 vemos que las detecciones erróneas se dan cuando el modelo nos devuelve una coordenada cualquiera de la imagen, dando como resultado un desplazamiento respecto de la anterior medición muy superior al que tendría una detección correcta. Ocurre también que, tras una detección incorrecta, en algunos frames inmediatamente posteriores el modelo da como resultado esta misma coordenada, resultando en un desplazamiento aproximado a 0. Dado este comportamiento, recorreremos el vídeo comprobando en cada frame el desplazamiento de la pelota respecto del último frame con esta detección también realizada y respecto del siguiente. En función de estas 2 distancias decidiremos si la detección es errónea si se cumplen ciertas condiciones, en tal caso la eliminaremos y marcaremos el frame como “pendiente de interpolar pelota”. Las condiciones para dar una detección por mala, entre las que se tienen que cumplir una, son:

- Si la distancia respecto de la anterior detección es mayor que el **umbral máximo** y respecto de la posterior es también mayor que el **umbral máximo**.
- Si el frame anterior se ha eliminado por ser erróneo y la distancia respecto del frame posterior es mayor que 3 veces el **umbral máximo** (al no tener una distancia anterior, requeriremos de un mayor desplazamiento posterior para asegurar que la detección es errónea).
- Si la distancia respecto del anterior es mayor que el **umbral máximo** y la distancia respecto del posterior está por debajo del **umbral mínimo**.
- Si el frame anterior se ha eliminado por ser erróneo y la distancia respecto del frame posterior es menor que el **umbral mínimo**.
- Si la distancia respecto del anterior es menor que el **umbral mínimo** y respecto del posterior es mayor que el **umbral máximo**.

Estos umbrales se calculan en función del número de frames que haya entre frames a comparar, según las ecuaciones 3.7:

siendo i la posición del frame actual

¹ Se podría cuestionar esto último argumentando que, al estar midiendo la velocidad en función de la distancia que se desplaza entre frames, en una secuencia de frames que esté captando un golpeo en el que la pelota impacta con la raqueta e invierte su dirección, podríamos medir un desplazamiento de la pelota cercano a 0. Esto no ocurre en nuestro sistema ya que, al dejar frames sin procesar en el módulo 3.2.4, no medimos distancias entre dos frames inmediatamente consecutivos, sino que están espaciados temporalmente, resultando en un desplazamiento de la pelota asegurado.

y j la posición del frame (anterior o posterior) respecto del que se quiere medir la distancia

$$\text{umbral_max}_i = 20 * |(i - j)| \quad \text{umbral_min}_i = 3 * |(i - j)| \quad (3.7)$$

El sistema continuará su procesado con el módulo 3.3.5.

3.3.5. Interpolación de pelota

En los módulos 3.2.4 y 3.3.4 hemos dejado frames etiquetados como “pendiente de interpolar pelota”. Ahora se recorrerá el vídeo y, para cada uno de estos frames, aplicaremos una técnica similar a la realizada en la interpolación de jugadores del módulo 3.3.3:

- 1.– Obtendremos el último frame que sí tenga una detección de la pelota, con posición i (sin tener en cuenta posibles frames ya interpolados)
- 2.– Obtendremos el siguiente frame que sí tenga una detección de la pelota, con posición k
- 3.– Dadas las coordenadas de la pelota en los frames i y k , interpolaremos la posición actual de la pelota p_j según la fórmula 3.6.
- 4.– Guardaremos esta nueva coordenada de la pelota junto con la información respectiva al frame actual.

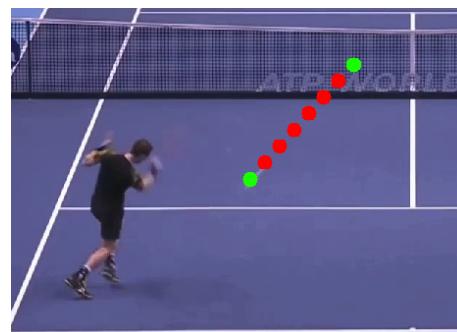


Figura 3.10: Ejemplo de resultados de interpolación (puntos rojos) a partir de unas coordenadas anterior y posterior (puntos verdes)

En la figura 3.10 podemos ver 6 coordenadas interpoladas, dibujadas en rojo, a partir de dos detecciones (anterior y posterior), dibujadas en verde.

3.4. Parte III: procesado de alto nivel

En este punto del sistema ya se ha extraído toda la información de bajo nivel de cada frame. En esta parte se realizarán con ella diferentes procesados que hemos calificado como de “alto nivel”. Este procesado consiste en:

- Clasificar qué rangos de frames constituyen un golpeo
- Estudiar el desplazamiento sobre la pista que realiza cada jugador entre y durante los golpeos
- Clasificar los golpeos según sean saques, de ataque, de defensa o finales.
- Calcular en cada frame de golpeo la trayectoria que va a seguir la pelota
- Calcular en cada frame el movimiento de los pies de cada jugador entre cada golpeo propio.

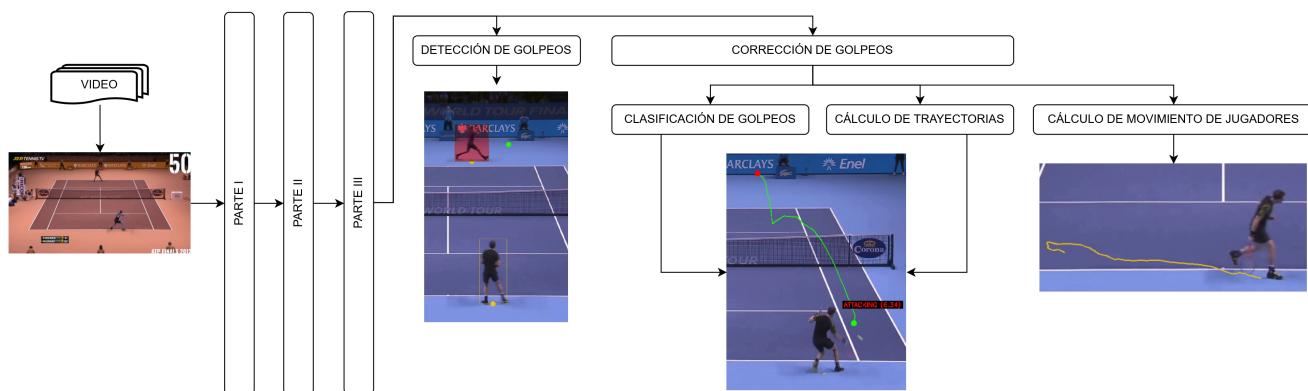


Figura 3.11: Diagrama de flujo de la Parte III del sistema

3.4.1. Detección de golpeos

La forma ideal de averiguar si en algún instante se está produciendo un golpeo sería conociendo las posiciones de los jugadores y de la pelota en las tres dimensiones que tiene realmente la pista. Esta información la podemos obtener de los jugadores, ya que como asumimos que el lado inferior de su bounding box está siempre en contacto con el suelo, su lado superior coincidiría con la altura del jugador respecto del suelo. En cambio, de la pelota solo podemos averiguar su posición en la imagen. Esto nos limita a la hora de averiguar si en un frame se está realizando un golpeo, ya que si tuviésemos las coordenadas tridimensionales de la pelota podríamos saber si esta se encuentra lo suficientemente cerca de alguno de los jugadores como para estar siendo golpeada por alguno de ellos. En cambio, nosotros solo podemos utilizar las posiciones relativas de los jugadores y de la pelota respecto de la imagen.

Otro aspecto a tener en cuenta antes de definir qué es un frame de golpeo son las necesidades futuras de nuestro sistema. Para cálculos y condiciones posteriores queremos englobar en esta definición, no solo el instante en el que el jugador está realmente realizando el golpeo, sino cuando debería estar haciéndolo. De este modo, estaremos teniendo en cuenta también los instantes en los que un jugador no ha llegado a golpear la pelota y se está produciendo un “passing”, ya sea por que no ha sido capaz de llegar (perdiendo el punto) o por que la pelota ha ido fuera (ganando el punto).

Teniendo todo esto en cuenta, definiremos como **frame de golpeo** de un jugador uno en el que la pelota se encuentre dentro una región de la imagen en función del bounding box de ese jugador. Esta

región será la delimitada:

- Lateralmente, por los propios límites de la imagen.
- Verticalmente, por dos límites a la misma altura que los lados superior e inferior del bounding box del jugador, desplazados hacia fuera una distancia igual al 30 % de la altura del propio bounding box ($|altura\ superior - altura\ inferior|$).

En este módulo recorreremos los frames del vídeo clasificando cada uno de ellos si es de golpeo y de qué jugador según esta definición. En la figura 3.12 se pueden ver los tres casos reales: golpeo, no golpeo y passing; y como se han clasificado según esta definición.



Figura 3.12: Ejemplos de clasificación de frames según si son golpeos. Con la detección de la pelota dibujada con un punto verde, los bounding boxes de los jugadores y sus pies en amarillo y el bounding box del jugador detectado como golpeador (si lo hay) relleno con una máscara roja.

3.4.2. Corrección de golpeos

En el módulo 3.4.1 hemos definido qué es un frame de golpeo. Ahora definiremos los siguientes momentos:

- **Momento de golpeo:** secuencia de frames clasificados (por dirección o corregidos) como de golpeo del mismo jugador.
- **Momento de no golpeo:** secuencia de frames no clasificados como de golpeo.
- **Momento de juego:** secuencia de momentos de golpeo o de no golpeo.
- **Momento de no juego:** secuencia de frames clasificados en el módulo 3.2.1 como no enfoques a pista completa.

A lo largo de un punto en juego, se deberían intercalar momentos de no golpeo y de golpeo, siendo cada uno de estos últimos sobre un jugador diferente. Existen varias razones por las que necesitamos realizar correcciones en las secuencias de golpeo y no golpeo para que se siempre se siga esta patrón:

- **Errores o pérdidas en las detecciones** de jugadores o de la pelota en uno o varios frames, que dividan lo que debería ser una única secuencia de frames de golpeo. Esto resultará en: momento de no golpeo → momento de golpeo de jugador X → frame o secuencia de frames mal detectados → momento de golpeo de jugador X → momento de no golpeo.
- **Detecciones de falsos momentos de golpeo** a causa de globos. Si el jugador inferior golpea la pelota de forma bombeada, esta puede coger la suficiente altura en la pista como para entrar en la región de la imagen de

detección de golpeo del jugador superior, haciendo que se detecte como momento de golpeo. Despu s bajar  y volver  a entrar en esta regi n, siendo esta vez el golpeo real. Esto resultar  en una secuencia del tipo: momento de no golpeo → falso momento de golpeo de jugador X → momento de no golpeo → momento de golpeo de jugador X → momento de no golpeo.

En base a esto, recorreremos el v deo comprobando las secuencias de momentos de golpeo, y cuando se reconozca una secuencia del tipo “momento de golpeo de jugador X → momento de no golpeo → momento de golpeo de jugador X”, corregiremos los frames que forman el “momento intermedio de no golpeo” a “frames de golpeo de jugador X”, convirtiendo las 3 secuencias en un solo momento de golpeo. Aun que de esta forma incluyamos las falsas detecciones dentro de los momentos de golpeo y esto haga que perdamos precisi n en futuros procesos, seguir  siendo posible realizar con esta informaci n los c lculos que nos lleven a la informaci n objetivo.

3.4.3. C lculo de trayectoria de la pelota

Durante este m dulo recorreremos el v deo analizando “tr os de momentos” consecutivos con la forma: “**momento de golpeo actual → momento de no golpeo intermedio → momento de golpeo siguiente**”. Habiendo hecho bien las anteriores correcciones, ambos momentos de golpeo ser n de jugadores contrarios. Para esta secuencia de momentos trataremos de clasificar el momento de golpeo actual y de calcular la trayectoria de la pelota consecuencia de este golpeo.

Para realizar los c lculos que nos lleven a esta informaci n, definiremos como **frame de impacto** de la pelota, el frame (dentro del momento de golpeo) con la posici n de la pelota m s alta si est  golpeando el jugador superior, y con la posici n de la pelota m s baja si est  golpeando el jugador inferior. Haremos esto bas ndonos en que la \'unica raz n por la que la pelota cambie de direcci n vertical y, en consecuencia, alcance un m ximo o un m nimo, es por que est  siendo golpeada. Esta lo podemos visualizar en la figura 3.13



(a) Frame anterior al de impacto

(b) Frame de impacto

(c) Frame posterior al de impacto

Figura 3.13: Comparaci n de frames antes, durante y despu s de impacto; siendo el punto verde la posici n de la pelota y la l nea la trayectoria que va a seguir la pelota (en rojo si es frame de impacto y en verde si no)

Para cada frame del momento de golpeo actual queremos calcular la **trayectoria** que va a tener la pelota, que consistir  en una lista de las coordenadas de la pelota desde el frame en el que se est  calculando hasta el frame de impacto del momento de golpeo siguiente.

Representación de trayectoria para vídeo de salida Esta trayectoria es una de las informaciones objetivo del sistema que queremos representar visualmente en el vídeo de salida. Para ello dibujaremos en cada frame con una trayectoria una línea verde hecha a partir de las coordenadas que la forman, también representaremos la posición de la pelota en el frame como un punto verde y el final de la trayectoria con un punto rojo. Esto se aprecia en la figura 3.14.

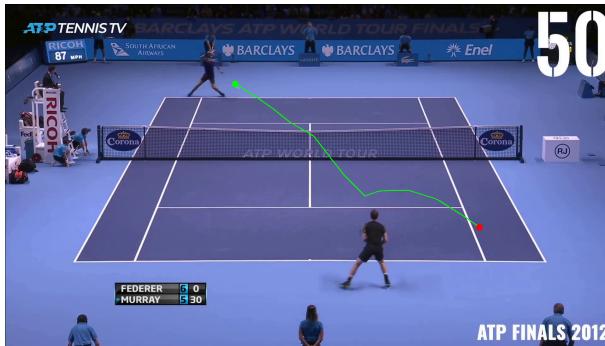


Figura 3.14: Frame con trayectoria de pelota dibujada por una linea verde, siendo el punto verde la posición actual de la pelota y el punto rojo la posición que tendrá en el frame de impacto siguiente

3.4.4. Clasificación de golpeos

En este módulo recorreremos los momentos del vídeo, y en función del orden en el que aparezcan y de diferente información contenida en sus frames calificaremos los momentos de golpeo en 4 tipos: saque, ataque, defensa, final. Para la clasificación de los dos primeros utilizaremos cálculos e información diferente que para los dos últimos.

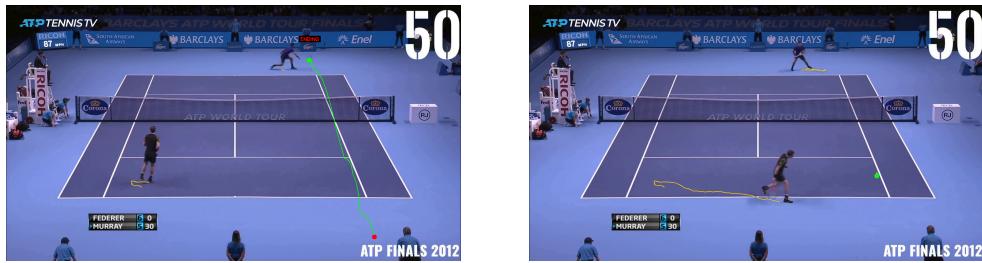
Clasificación de saque y final El primer momento de golpeo después de un momento de no juego (aun teniendo momentos de no golpeo entremedias) lo calificaremos como de **saque** (figura 3.15), ya que es la forma de comenzar cualquier punto.



Figura 3.15: Frame del primer momento de golpeo después de un momento de de no juego, clasificado como saque. Con la detección de la pelota dibujada con un punto verde, su trayectoria como una linea verde y el destino de la pelota como un punto rojo; el movimiento de los jugadores desde su respectivo último golpeo como una línea amarilla.

El penúltimo momento de golpeo antes de un momento de no juego lo calificaremos como

final, ya que el ultimo momento corresponderá a un passing o a un golpeo a la red, como vemos en la figura 3.16.



(a) Frame de penúltimo momento de golpeo, clasificado como final (b) Frame de último momento de golpeo, atribuido a un passing

Figura 3.16: Frames de los dos últimos momentos de golpeo antes de un momento de no juego, el primero siendo golpeo final y el segundo un passing. Con la detección de la pelota dibujada con un punto verde; el movimiento de los jugadores desde su respectivo último golpeo como una línea amarilla.

Podemos utilizar esta lógica ya que durante la retransmisión de un partido de tenis, entre punto y punto mientras se vuelven a preparar los jugadores, se suele mostrar alguna repetición, un primer plano de algún jugador, un enfoque al público, alguna estadística, etc.

Clasificación golpeos de ataque y de defensa En cualquier otro momento de golpeo, para distinguirlos entre ataque o defensa, nos basaremos en la siguiente premisa: **Si el jugador A se encuentra en una situación cómoda para golpear (por que el jugador B haya cometido un error, bajado el ritmo, etc) y decide iniciar un ataque con su golpeo, el ataque consistirá en poner en una situación incómoda al jugador rival B o, en ciertas ocasiones, en realizar un golpe ganador.** Si consigue obligar al jugador B a jugar de una forma más incómoda, el jugador A estará aumentando las probabilidades de que el rival cometa un error o que le de la posibilidad de realizar un golpe ganador.

En el tenis existen diferentes formas de incomodar al rival con un golpeo, las principales son:

- Mover al rival: cuanta más distancia se le oblique al rival a recorrer para llegar a golpear, con mayores dificultades devolverá la pelota y más desgaste físico sufrirá.
- Desplazar al rival lateralmente: si se obliga al contrincante a golpear en una zona muy alejada lateralmente del centro de la pista, este tendrá menos ángulo para direccionar su golpeo y estará dejando más zona de la pista desprotegida, que se puede aprovechar para continuar con eñ ataque o para realizar un golpe ganador.
- Desplazar al rival al fondo de la pista: cuanto más lejos de la red se esté obligando al rival a golpear, más distancia tendrá que recorrer su pelota para entrar en la pista, perdiendo precisión y fuerza y permitiendo al jugador a adelantarse hacia la red para continuar con su ataque. En cambio si se permite al rival adelantarse, más ángulos tendrá a su disposición para direccionar la pelota y en menor tiempo y con mayor velocidad le llegará al jugador, obligándole a entrar en modo defensa o, incluso, a perder el punto.
- Obligar al rival a jugar con sus peores golpes: si al rival se le da peor hacer ciertos golpes (dar

de derecha o de revés, devolver golpes con efectos liftados o cortados, que lleven mucha o poca altura, etc) y se le obliga a jugar con ellos, aumentarán las probabilidades de que cometa errores o de que juegue a un ritmo más bajo.

- Aumentar el ritmo de juego: cuanto más rápido o fuerte el jugador esté golpeando, aun que más esté arriesgando y más precisión pierda, menos tiempo le estará dejando al rival para reaccionar al golpeo y con mayor dificultades realizará él el suyo.

En base a estas tácticas de juego y a la información que hemos sido capaces de extraer del vídeo, se han creado diferentes métricas que miden cómo de incómodo un jugador está obligando al rival a jugar. Si este nivel de incomodez (combinación de las métricas) supera un umbral podremos afirmar que el jugador se encuentra realizando un ataque, en el caso contrario diremos que se encuentra en modo defensivo.

Durante este módulo recorreremos los momentos del vídeo analizando “tríos” consecutivos con la forma: “**momento de golpeo (actual)** → **momento de no golpeo (intermedio)** → **momento de golpeo (siguiente)**”. Analizando estos momentos y los frames que lo forman, podremos calcular las siguientes métricas, tomando como **jugador actual** el jugador que está realizando el golpeo en el momento de golpeo actual y como **jugador contrario** el que está realizando el golpeo en el momento de golpeo siguiente:

- **Desventaja por desplazamiento:** queremos medir la distancia que se está obligando al jugador contrario a moverse para realizar su golpeo. El punto inicial para calcular esta distancia será la coordenada media de los pies del jugador contrario, trasladadas al plano cenital de la pista (calculada con los transformadores homográficos obtenidos en el módulo 3.2.2), durante los frames del golpeo actual ². El punto final será la coordenada de los pies del mismo jugador contrario en el frame de impacto del golpeo siguiente, también trasladada al plano cenital. Matemáticamente calculado con la fórmula 3.8.

$$\forall i \in \{\text{frames de momento de golpeo actual}\}$$

$$\forall j = \text{frame de impacto de momento de golpeo siguiente}$$

p_k = coordenada de los pies del jugador contrario en el frame_k trasladada al plano cenital de la pista

$$\text{dist}(p_a, p_b) = \text{distancia euclídea entre la coordenada } p_a \text{ y } p_b$$

$$\text{desventaja por desplazamiento} = \text{dist}\left(\frac{\sum_i p_i}{\text{len}(\{\text{frames de golpeo actual}\})}, p_j\right) \quad (3.8)$$

- **Desventaja horizontal:** queremos medir la distancia horizontal respecto del centro de la pista a la que se le ha obligado al jugador contrario a alcanzar para realizar su golpeo. Esto lo calcularemos con la fórmula 3.9.

posizquierda = coordenada de los pies del jugador contrario más a la izquierda, durante el golpeo siguiente, trasladada al plano cenital

posderecha = coordenada de los pies del jugador contrario más a la derecha, durante el golpeo siguiente, trasladada al plano cenital

centro_vertical = línea vertical central de la pista en el plano cenital

²Se utiliza la media de las coordenadas en todos los frames del momento de golpeo actual y no la del frame de impacto actual ya que se considera que el jugador contrario puede tanto intuir la dirección del golpeo del jugador actual iniciando su movimiento antes del impacto, como no saber en qué dirección moverse hasta ver la pelota desplazarse después del impacto, comenzando su desplazamiento más tarde.

$$\text{desventaja horizontal} = \max(\text{dist}(\text{pos}_{izquierda}, \text{centro_vertical}), \text{dist}(\text{pos}_{derecha}, \text{centro_vertical})) - 1 \quad (3.9)$$

A esta distancia le restamos 1 ya que si se le deja al jugador contrario demasiado cerca del centro, se le estará dando cierta ventaja. De esta forma, si queda a menos de un metro de distancia del centro, la métrica se volverá negativa.

- **Ventaja vertical:** queremos calcular la distancia entre la posición más adelantada del jugador contrario durante el siguiente golpeo respecto de la línea de fondo de su lado de la pista, solo si en algún momento se adelanta a ella, en caso contrario esta métrica tendrá valor 0.

$\text{pos}_{adelantada}$ = coordenada de los pies del jugador contrario más cercana a la red, durante el golpeo siguiente, trasladada al plano cenital

linea_fondo = línea de fondo de la pista del lado del jugador contrario, sobre el plano cenital

$$\text{ventaja vertical} = \text{dist}(\text{linea_fondo}, \text{pos}_{adelantada}) \quad (3.10)$$

Para calcular la desventaja total con la que el jugador actual, con su golpeo, deja al jugador contrario utilizamos la fórmula 3.11, que combina las métricas anteriores.

$$\text{desventaja total} = \text{desventaja por desplazamiento} + 1,3 \cdot \text{desventaja horizontal} - 2 \cdot \text{ventaja vertical} \quad (3.11)$$

Si esta **desventaja total** supera el umbral de 4 clasificaremos el golpeo actual y todos los frames que lo forman como **ataque**, en caso contrario como **defensa**.

Representación de clasificación para vídeo de salida La clasificación del los golpeos es una de las informaciones objetivo del sistema, por ello debe representarse en el vídeo de salida del sistema. Esto se hará escribiendo el tipo de golpeo junto al jugador golpeando para cada frame clasificado como tal. Se puede apreciar esto en las figuras 3.15 y 3.16(a).

3.4.5. Cálculo de movimiento de jugadores

Queremos poder representar en el vídeo final el movimiento que realiza un jugador sobre la pista entre golpeos propios, de esta forma podremos visualizar cuánto y cómo se ha visto obligado a moverse desde un golpeo propio hasta su siguiente. Para ello calcularemos para cada jugador y en cada frame dentro de un “momento de golpeo” o de un “momento de no golpeo” la trayectoria de sus pies. Esta trayectoria consiste en la lista de puntos de los pies del jugador desde su último frame de impacto hasta el frame para el que se estén realizando los cálculos. Para cada frame en el que se calcule esta trayectoria, se dibujará como una línea amarilla hecha a partir de las coordenadas que la forman, como se ve en la secuencia 3.16.

EXPERIMENTOS, RESULTADOS Y VALIDACIONES

En esta sección explicaremos los diferentes experimentos que hemos realizado para validar y medir el rendimiento del sistema y de las diferentes partes y módulos.

Esta validación se ha dividido en tres experimentos:

- 1.– **Validación de información de bajo nivel en frames aleatorios**
- 2.– **Validación de información de bajo nivel en un vídeo corto**
- 3.– **Validación de información de alto nivel en un vídeo largo**

En cada una de ellas se procesará un vídeo con unas características diferentes y en los que se ha etiquetado información diferente. El propósito de esta división es poder analizar los resultados y rendimiento de diferentes partes del sistema y en diferentes situaciones.

4.1. Vídeos utilizados para la validación

Ahora se explicarán las características de los vídeos utilizados en cada experimento y la información etiquetada en cada uno de ellos.

4.1.1. Vídeo de frames aleatorios

El objetivo del experimento en el que se usará este vídeo como entrada es poner a prueba la Parte I (Sección 3.2) de nuestro sistema en la mayor variedad posible de situaciones: diferentes superficies de pista, ángulos de enfoque (aun que todos dentro de los enfoques estandarizados), características de la pista, luces y sombras, calidad de retransmisión, etc. Para ello extraeremos conjuntos de frames de diferentes vídeos y los concatenaremos para crear uno solo. Esto se hará de la siguiente forma:

Elección de vídeos Se han elegido los siguientes vídeos para extraer de ellos frames:

- Recopilación de mejores puntos de la década de 2010 [14]: Este vídeo contiene una recopilación de 50 puntos de partidos jugados entre el 2010 y 2020, en diferentes superficies, ángulos y condiciones lumínicas.

- Nadal contra Djokovic en Roland-Garros 2020 [15]: Partido jugado sobre superficie de tierra batida.
- Serena Williams contra Venus Williams en Australia 2017 [16]: Partido jugado sobre superficie de pista rápida.
- Alcaraz contra Djokovic en Madrid 2022 [17]: Partido jugado sobre tierra batida.
- Alcaraz contra Djokovic en Wimbledon 2023 [18]: Partido jugado sobre hierba.

Estos 4 últimos vídeos no son los partidos completos, sino compilaciones de los mejores momentos de cada uno de ellos. Aun que algunos vídeos tienen resolución mayor, al extraer las imágenes se han re-dimensionado para tener todas una resolución de 720p.

Extracción de frames de vídeos De cada vídeo se extraerán conjuntos o sets de 7 frames consecutivos, espaciados por un número de frames que se ignorarán. Este número de frames ignorados entre cada set extraído será diferente para cada vídeo, de forma que obtengamos la cantidad de sets que queramos de cada uno de ellos. En total se han extraído 72 sets de frames, de los cuáles un 48 % corresponden al primero de los vídeos antes mencionados y unos 7 %, 6 %, 11 % y 13 % a los otros respectivamente.

La razón de extraer sets de frames consecutivos y no frames separados, es que estos conjuntos sean procesados en esta prueba de la manera más parecida posible a como se harían en un procesado normal: como secuencias congruentes y pudiendo acceder a frames anteriores para realizar detecciones y correcciones necesarias en la Sección 3.2.

Concatenación de frames Para evitar que dos sets consecutivos, extraídos de situaciones separadas, se procesen como una única secuencia dando pie a posibles errores, se insertará un frame completamente negro entre cada set. De esta forma el sistema lo reconocerá como un “no enfoque a pista completa” en el módulo 3.2.1 y aprovecharemos la lógica del propio sistema para evitar que durante el procesado de un frame se usen frames de un set diferente en, por ejemplo, la detección de la pelota en el módulo 3.2.4 o correcciones de los puntos de la pista en el módulo 3.2.2.

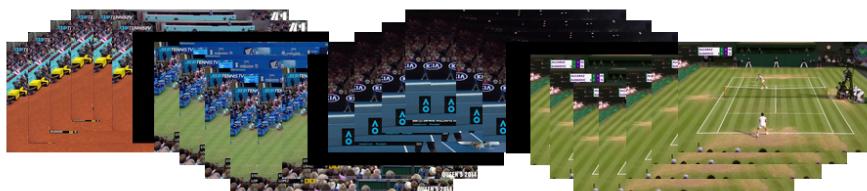


Figura 4.1: Ejemplo de formación de vídeo con sets de frames aleatorios, en los que se intercala cada conjunto de 7 frames por uno completamente negro

Etiquetado de información de bajo nivel Como este vídeo solo será procesado por la Parte I, solo se ha etiquetado en los frames la información de bajo nivel: clasificación de enfoque a pista completa, 4 puntos definitorios de la pista de dobles¹, bounding box de ambos

¹Con el fin de reducir la cantidad de información a etiquetar manualmente, se ha decidido etiquetar solo estos 4 puntos y no los 8 de la pista.

jugadores y coordenada de la pelota. Esto lo haremos con ayuda de matlab, añadiendo estas etiquetas a cada frame del vídeo como se explica en la Sección B del Apéndice.

De cada set extraído solo se etiquetarán, y por lo tanto validarán, los 5 últimos. La razón por la que se incluyen los 2 primeros es para que los módulos que requieren información de frames anteriores (de nuevo los módulos 3.2.2 y 3.2.4) funcionen correctamente para todos los frames que queremos validar de cada set.

Características finales del vídeo El vídeo final tiene una resolución de 720p, con una duración de 10 segundos y contendrá 360 frames etiquetados.

4.1.2. Vídeo corto

El objetivo del experimento en el que se usará este vídeo como entrada es poner a prueba las Partes I y II (Secciones 3.2 y 3.3) en conjunto, es decir, la información de bajo nivel tras el post-procesado. Para esto se ha extraído una sección del vídeo “Recopilación de mejores puntos de la década de 2010 [14]” de 7 segundos de duración, en el que se han etiquetado los 401 frames que lo forman con la información de bajo nivel: 8 puntos de la pista, bounding boxes de ambos jugadores, coordenada de la pelota. El etiquetado se ha realizado con MatLab como se explica en la Sección B del Apéndice.

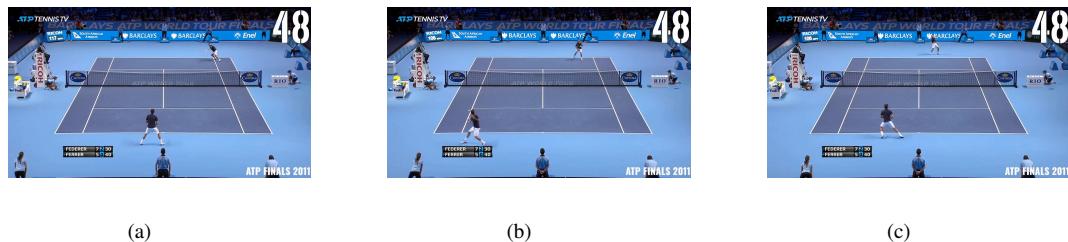


Figura 4.2: Ejemplo de frames que forman el vídeo corto

4.1.3. Vídeo largo

Con el experimento en el que usaremos este vídeo queremos validar la información final del sistema: detección y de clasificación golpes. Para ello se han extraído los primeros 5 minutos y 57 segundos del vídeo “Serena Williams contra Venus Williams en Australia 2017 [16]”. Este nuevo vídeo contiene 68 puntos intercalados por descansos, repeticiones, primeros planos, etc; de forma que los resultados obtenidos de la validación de la información del vídeo sean representable del funcionamiento del sistema para una retransmisión completa. En él se ha etiquetado para cada frame: si es un “frame de golpe real” (y por tanto pertenece a un “momento de golpe real”) y su clasificación. Consideramos un **momento de golpe real** el rango de frames desde que el jugador haya terminado

de posicionarse y comience a hacer el movimiento de golpeo hasta que lo termina. Este movimiento lo podemos observar en la figura 4.4.



Figura 4.3: Ejemplo de frames que forman el vídeo largo



Figura 4.4: Secuencia de un movimiento de golpeo de revés. Obtenido de [19].

4.2. Métricas

A lo largo del sistema, cada módulo se ha encargado de extraer o corregir un tipo de información diferente. Ahora explicaremos las métricas que se han utilizado para validar cada una de ellas.

Enfoques a pista completa utilizaremos **Accuracy**, **Precision** y **Recall**.

Puntos de la pista Para evaluar la precisión con la que hemos detectado los 8 puntos definitorios de la pista en cada frame:

- 1.– Dividiremos los 8 puntos resultantes del procesamiento (visibles en la figura 3.3(e)) en 2 grupos: 4 puntos de la pista individual y 4 puntos de la pista de dobles.
- 2.– Formaremos un polígono rectangular con cada grupo: polígono individual y polígono de dobles.
- 3.– Repetiremos los dos pasos anteriores para los 8 puntos etiquetados en el mismo frame (visibles en la figura B.1), obteniendo 2 pares de polígonos rectangulares: polígono resultado - polígono etiquetado.
- 4.– Para cada par de polígonos “resultado - etiquetado” calcularemos la **Intersection over Union** entre ellos [20].
- 5.– Consideraremos la métrica IoU del frame como la media entre las dos IoU obtenidas ².

Para esta métrica definiremos un **umbral de 0.85**, por encima del cuál daremos una detección por correcta (True Positive), en caso contrario la daremos por incorrecta (False Negative). Con esto podremos calcular el **Recall** para los frames de un vídeo.

²En el caso de haber etiquetado solo uno de los dos grupos de puntos (de pista individual o de dobles), solo tendremos en cuenta la IoU obtenida



Figura 4.5: Ejemplo de puntos de pista que forman un polígono (rojo) que produce un IoU en el umbral de 0.85 al calcularse respecto de los puntos de pista etiquetados que forman otro polígono (verde)

Jugadores Para evaluar la detección de los jugadores (superior e inferior) utilizaremos la **Intersection over Union** entre las bounding boxes obtenidas para cada uno de ellos y las etiquetadas. Utilizaremos el valor medio de ambas IoU como métrica para evaluar la detección de jugadores en cada frame. Como umbral para calificar una detección como correcta utilizaremos el valor estándar de 0.5 [21].

Otra métrica que usaremos será la **Distancia Euclídea** en la imagen entre los pies de los bounding boxes detectados y de los etiquetados. Para considerar una detección correcta, deberá estar por debajo de un umbral de 70 píxeles (4 % de la diagonal de la imagen). La razón por la que utilizaremos también esta métrica es que son los pies los que trasladaremos al plano cenital con las homografías para localizar a los jugadores sobre la pista; y aun que estos se calculen con los bounding boxes, puede haber imprecisiones en ellos que no se traduzcan en imprecisiones en los pies.

De ambas métricas, al distinguir entre detecciones correctas e incorrectas (entre las que incluiremos las pérdidas de detecciones), calcularemos el **Recall** para los frames de un vídeo. También calcularemos el valor medio de ellas a lo largo del vídeo.

Pelota Para evaluar la detección de la pelota en cada frame utilizaremos la **Distancia Euclídea** entre la coordenada obtenida y la etiquetada. Para considerar una detección correcta, deberá estar por debajo de un umbral de 70 píxeles (4 % de la diagonal de la imagen). Para al evaluación general del vídeo se calcularán el **Recall** de las detecciones correctas e incorrectas (incluidas pérdidas) de los frames y la media de las distancias.

Momentos de golpeo Queremos comprobar si cada golpeo real en el partido ha sido detectado por nuestro sistema, es decir, que cada “momento de golpeo real” etiquetado esté contenido en un “momento de golpeo” resultado del procesado, sin que estos últimos contengan más de un momento real. Para comprobar esto, dado que un momento es un rango de frames a lo largo del vídeo y por lo tanto podemos calcular la intersección entre dos de

entre el polígono construido con él y el polígono resultado del procesamiento homónimo.

ellos, calcularemos para cada momento real etiquetado si existe un momento resultado del proceso con el que tenga una intersección mayor que 0, en tal caso daremos el golpeo por detectado y el momento resultado no lo volveremos a tener en cuenta para las comprobaciones de otros momentos reales. En caso de no encontrar ningún momento resultado con intersección mayor que 0, daremos el golpeo por no detectado.

A lo largo del vídeo encontraremos detecciones correctas (True Positives), golpeos no detectados (False Negatives) y falsas detecciones (False Positives) y calcularemos los **Accuracy, Precision y Recall**.

Clasificación de golpeos En el cálculo de la validación anterior, en el caso de detecciones correctas, se ha emparejado cada momento de golpeo real con uno resultante del proceso (con el que tiene intersección mayor que 0). Compararemos la clasificación de ambos llenando una **Matriz de confusión** para cada tipo de golpeo, de la que calcularemos los **Accuracy, Precision y Recall**. Para tener una medida general del clasificador, calcularemos el **Accuracy** general y los **micro Precision y micro Recall**. Utilizaremos el cálculo micro y no el macro ya que el micro tiene en cuenta el volumen de cada clase [22].

Tiempo medio de procesado Uno de los propósitos de la Parte II del sistema es reducir el tiempo de procesado. Para medir la ganancia en eficiencia computacional y tiempo de procesado calcularemos el **Tiempo medio de procesado por frame** en segundos.

4.3. Resultados

En esta sección explicaremos los resultados obtenidos de los diferentes experimentos realizados para validar los procesos del sistema.

4.3.1. Validación de información de bajo nivel sin post-procesado

En esta prueba queremos validar los procesos de obtención de información de bajo nivel explicados en la Sección 3.2, utilizando el vídeo 4.1.1 como entrada al sistema. Para ello solo han estado activos durante el procesado los módulos pertenecientes a la Parte I.

Dadas las métricas explicadas en la Sección 4.2, se han obtenido los resultados de la tabla 4.1. Analizándola, vemos en las columnas de IoU, Distancia y Accuracy que, cuando se realiza una detección, estas en general son buenas. En cambio, al ver la columna de Pérdidas de los puntos de la pista y de la pelota vemos que tenemos una gran cantidad de pérdidas en estas detecciones, lo que provoca un Recall para ambas informaciones bajo. Esto resalta la importancia del post-procesado de la Parte II, que tiene como uno de sus objetivos el reducir estas pérdidas. Ahora discutiremos más en detalle cada información y sus validaciones.

	Pérdidas	IoU	Distancia	Accuracy	Precision	Recall
Enfoques a pista completa				0.8259	0.7143	0.8434
Puntos de pista	36 %	0.9316				0.5422
Jugadores (boxes)	0 %	0.7513				0.9619
Jugadores (pies)	0 %		15.6997			0.9952
Pelota	34 %		18.8758			0.6436

Tabla 4.1: Tabla con los resultados de la validación de la información de bajo nivel obtenidos de la Parte I al procesar el vídeo de frames aleatorios (explicado en la Sección 4.1.1)

Enfoque a pista completa Revisando los resultados, atendiendo a los frames clasificados incorrectamente, observamos que todos los frames extraídos de un vídeo concreto nunca se clasifican como enfoque a pista. Estos son los frames obtenidos del partido “Alcaraz vs Djokovic en Wimbledon 2023” [18]. Si procesamos el vídeo original entero, veremos que no hay ningún enfoque a pista reconocido correctamente, sino que todos los frames del vídeo se clasifican como no enfoques a pista completa. Se han realizado varios experimentos con el fin de ahondar en este error, estas pruebas están descritas en la Sección A.1 del Apéndice. En ellas, aun que parece haber un error con los partidos jugados en Wimbleldon, no creemos que podamos sacar una respuesta concluyente.

Al tratarse de un caso excepcional, en este experimento y en los resultados de la tabla 4.1 no se han tenido en cuenta las validaciones de los frames extraídos del vídeo original “Alcaraz vs Djokovic en Wimbledon 2023” [18].

Puntos de la pista Uno de los propósitos de este experimento es conocer la capacidad del sistema de extraer la información de bajo nivel sin las correcciones e interpolaciones de la Parte II. En cambio, respecto a los puntos de la pista, por decisiones de implementación, sus correcciones se llevan a cabo en la Parte I (Sección 3.2.2), por lo que los resultados de la tabla 4.1 son sobre puntos de la pista ya con estas correcciones. Para ver la mejora que suponen estas correcciones, se ha repetido este experimento modificando el sistema para que no las realice y validado sus resultados en la tabla 4.2. En ella parece que estas mejoras no han sido tan significativas, pero la causa de esto es que el vídeo usado como entrada solo contiene 7 frames consecutivos antes de cambiar de punto, impidiendo que el sistema se beneficie de estas mejoras que requieren de información de frames anteriores para funcionar correctamente. Este beneficio se verá en el experimento explicado en la Sección 4.3.2.

Observando los resultados de este experimento, se ha encontrado un caso de error que provoca una mala detección (no pérdidas) de los puntos por culpa de una característica de la pista. Esto lo explicamos en la Sección A.2 del Apéndice.

	IoU	Recall
Sin correcciones	0.9272	0.5301
Con correcciones	0.9316	0.5422

Tabla 4.2: Comparación validación de puntos de pista con vídeo de frames aleatorios, sin y con correcciones.

Jugadores Como se ha discutido en la Sección 2.1.4, existen diferentes modelos de detección de objetos que podemos utilizar para nuestra detección de personas y jugadores en el módulo 3.2.3. Se ha repetido este experimento utilizando los modelos Faster R-CNN RestNet-50 FPN [6] y YOLOv5 de Ultralytics [7] y variando el umbral de confianza de ambos, obteniendo los resultados de la tabla 4.3. En ella se puede observar que con YOLO no solo necesitamos un 1 % del tiempo de procesamiento que requiere Faster, sino que con un umbral de confianza bajo de 0.20 llegamos a conseguir los mejores resultados. La razón por la que un umbral tan bajo nos funciona mejor es que al moverse los jugadores pos la pista estos puede aparecer borrosos en algunos frames, provocando que al utilizar un umbral de confianza alto los jugadores no lo superen.

Tipo de detección	Pérdidas	IoU	Recall	Tiempo por frame (s)
Faster u=80	1 %	0.7212	0.9047	3.78
Faster u=50	0 %	0.7265	0.9190	3.78
Faster u=20	0 %	0.7159	0.8952	3.78
YOLO u=80	42 %	0.7088	0.5333	0.05
YOLO u=50	0 %	0.7320	0.9381	0.05
YOLO u=20	0 %	0.7513	0.9619	0.05

Tabla 4.3: Comparación de modelos y umbrales en el reconocimiento de jugadores sobre el vídeo de frames aleatorios

Pelota Analizando la tabla 4.1 vemos la gran cantidad de pérdidas, esto son detecciones en las que el modelo ha devuelto un valor nulo. En cambio, cuando sí obtenemos resultados de la detección la distancia media respecto de las coordenadas etiquetadas son buenas, obteniendo una sola medición por encima del umbral. Esto pone en evidencia la necesidad de realizar un post-procesado como el explicado en la Sección 3.3.5 que reduzca estas pérdidas, ya que la posición de la pelota es clave para la detección y clasificación de golpes.

4.3.2. Validación de post-procesado de información de bajo nivel

En este experimento queremos validar los procesados de las partes I y II en conjunto. Esto son las clasificaciones y detecciones de la Sección 3.2 y las correcciones, interpolaciones y suavizados

de la Sección 3.3. Para ello solo han estado activos en el sistema los módulos de estas dos partes y utilizaremos el “vídeo corto” descrito en la Sección 4.1.2.

Dadas las métricas explicadas en la Sección 4.2, se han obtenido los resultados de la tabla 4.4. En ella podemos ver que estos resultados son significativamente mejores que los obtenidos en el experimento 4.3.1, habiendo reducido las pérdidas de las detecciones de los puntos de la pista y de la pelota y mejorando en general casi todas las métricas.

	Pérdidas	IoU	Distancia	Accuracy	Precision	Recall
Enfoques a pista completa				1	1	1
Puntos de pista	0 %	0.9798				1
Jugadores (boxes)	0 %	0.8365				1
Jugadores (pies)	0 %		8.0012			1
Pelota	0 %*		25.9936			0.8816

Tabla 4.4: Tabla con los resultados de la validación de la información de bajo nivel obtenidos de la Parte II al procesar el vídeo corto (explicado en la Sección 4.1.2)

Ahora describiremos para cada información qué cambios han sufrido con el post-procesado y cómo ha afectado al rendimiento del sistema.

Enfoque a pista completa Como se explica en la Sección 4.1.2, el vídeo utilizado en este experimento es un extracto de un punto en juego, por lo que todos sus frames son enfoques a pista y así son clasificados por el sistema.

Puntos de la pista El propósito de este experimento es conocer como mejora o cambia la información de bajo nivel con el post-procesado de la Parte II. Respecto a los puntos de la pista, este post-procesado consiste en un suavizado de las coordenadas, mientras que las correcciones se realizan en la Parte I, en la Sección 3.3.1. En el experimento anterior de la Sección 4.1.1 explicamos por qué las características de su vídeo no nos permiten beneficiarnos de estas correcciones, mientras que en este experimento sí deberían ser notables. Para observar esto, hemos validado en este experimento no solo la información final del sistema (con correcciones y post-procesado), sino también los puntos de pista antes de las correcciones de la Sección 3.2.2 (sin correcciones ni post-procesado) y antes del suavizado de la Sección 3.3.1 (con correcciones sin post-procesado). Los resultados de estas tres validaciones las podemos ver en la tabla 4.5.

En este caso sí podemos observar la mejora que suponen las correcciones de esta información de la Parte I. Respecto al suavizado, como se explica en la Sección 3.3.1, el suavizado de los puntos no supone un cambio tan significativo como mejorar notablemente la IoU, ya que el tamaño de los polígonos formados para calcular la métrica son demasiado grandes en comparación con los pocos píxeles de diferencia entre un punto antes y después del suavizado. En cambio, en la reproducción del vídeo resultante sí se aprecia visualmente

	Pérdidas	IoU	Recall
Sin correcciones y sin post-procesado	11 %	0.9709	0.869
Con correcciones y sin post-procesado	0 %	0.9775	1
Con correcciones y con post-procesado	0 %	0.9789	1

Tabla 4.5: Comparación validación de puntos de pista en vídeo corto con diferentes opciones de procesado

este post-procesado.

Jugadores Queremos comparar la información de los jugadores con y sin post-procesado.

Los resultados con post-procesado son los obtenidos en este experimento. Para obtener los resultados sin el post-procesado, se ha repetido este experimento desactivando los módulos 3.3.2 y 3.3.3 y no dejando frames sin procesar en el módulo 3.2.3. La validación de ambos resultados se puede ver en la tabla 4.6.

	Pérdidas	IoU	Distancia	Recall
Sin post-procesado	0 %	0.8584	6.9124	1
Con post-procesado	0 %	0.8365	8.0012	1

Tabla 4.6: Comparación validación de jugadores en vídeo corto, sin y con postprocesado.

Podemos ver que las detecciones sin el post-procesado son suficientemente buenas, al igual que se ve en la tabla 4.1. Con el post-procesado hemos conseguido unos resultados casi igual de buenos habiendo procesado un sexto de los frames del vídeo.

Otra ventaja observada en el vídeo resultante de este experimento es una mayor fluidez en el movimiento de los bounding boxes y pies de los jugadores a lo largo de los frames, mientras que sin el post-procesado este movimiento tiene mayor ruido.

Pelota Del mismo modo que con los jugadores, para comparar la información que obtiene el sistema con y sin post-procesado, se ha repetido este experimento desactivando los módulos 3.3.4 y 3.3.5 y no dejando sin procesar ningún frame en el módulo 3.2.4. La comparación de estos resultados sin post-procesado con los del experimento se observa en la tabla 4.7

	Pérdidas	Distancia	Recall
Sin postprocesado	44 %	19.3336	0.5315
Con postprocesado	0 %	25.9936	0.8816

Tabla 4.7: Comparación validación de pelota en vídeo corto, sin y con postprocesado.

Podemos ver en estos resultados el gran beneficio del post-procesado de la información de la pelota, habiendo reducido completamente las pérdidas sin reducir apenas la distancia media. Esta mejora es esencial para el calculo de la información de alto nivel.

Tiempo Uno de los beneficios de realizar el post-procesado de los jugadores y de la pelota

en la Parte II es reducir el coste computacional de sistema. En este experimento se ha medido el tiempo medio que requiere un frame para ser procesado y hemos repetido el experimento desactivando los módulos de la Parte II y no dejando frames sin procesar en los módulos 3.2.3 y 3.2.4 y medido el mismo dato. Ambos resultados se ven en la tabla 4.8.

	Tiempo
Sin post-procesado	2.42 segundos
Con post-procesado	0.99 segundos

Tabla 4.8: Comparación de tiempo sin usar y usando el postprocesado en el vídeo corto.

Aplicar el psot-procesado que forma la sección 3.3 provoca una reducción del 60 % del tiempo de procesado.

4.3.3. Validación de información de alto nivel

El objetivo de este experimento es analizar el rendimiento del sistema completo en un vídeo lo suficientemente largo como para que los resultados sean representativos del sistema para cualquier vídeo. Por ello en este experimento se usa el vídeo descrito en la Sección 4.1.3.

El resultado de la validación de la información sobre los golpeos, según las métricas explicadas en la Sección 4.2, la podemos ver en la tabla 4.9. Como la información validada es la información objetivo del sistema, estas métricas son las que describen el funcionamiento y rendimiento global del sistema. Siendo ambas tan buenas, consideramos que se han cumplido los objetivos declarados en la Sección 1.2.

	Accuracy	Precision	Recall
Detección de golpeos	0.9167	0.9429	0.9706
Clasificación de golpeos	0.8082	0.8939	0.8939

Tabla 4.9: Tabla con los resultados de la validación del vídeo largo, por módulos y métricas.

Ahora hablaremos sobre cada tipo de información.

Detección de golpeos Como se ha explicado en el la sección 4.1.3, el vídeo usado tiene 68 golpeos, de los cuales detectamos correctamente 66 y tenemos 4 falsas detecciones. Al estudiar los resultados se ha descubierto una situación causada por una mala detección de la pelota que provoca que 2 detecciones de “momentos de golpeo” acaben prematuramente, evitando que en este experimento engloben sus “momentos reales de golpeo”, resultando en las 2 detecciones no realizadas y en 2 de las falsas detecciones. Esta situación la describimos en la Sección A.3 del Apéndice, pero es un caso poco habitual.

A parte de esto, se han conseguido unos buenos resultados a la hora detectar los golpeos

de un partido.

Clasificación de golpes Los datos de la tabla 4.9 sobre la clasificación nos muestran que nuestro clasificador tienen un buen desempeño, habiendo clasificado correctamente 59 de los 66 golpes detectados. Si desgranamos estos datos según el tipo de golpeo etiquetado obtenemos los resultados expuestos en la tabla 4.10, en la que vemos unos resultados generalmente buenos.

Tipo	Volumen	Accuracy	Precision	Recall
Saque	24 %	0.9375	1	0.9375
Ataque	22 %	0.6667	0.8	0.8
Defensa	32 %	0.7917	0.8636	0.9048
Final	21 %	0.8667	0.9286	0.8667

Tabla 4.10: Comparación validación de tipos de golpeo en vídeo largo

A parte de esta comparativa, nos parece interesante agrupar los resultados de saque/final y ataque/defensa, ya que sus cálculos se basan en afirmaciones y cálculos diferentes, como se explica en la Sección 3.4.4.

Tipo	Volumen	Accuracy	Precision	Recall
Saque / Final	45 %	0.9032	0.9655	0.9333
Ataque / Defensa	54 %	0.7380	0.8377	0.8611

Tabla 4.11: Comparación validación de tipos de golpeo en vídeo largo

En la tabla 4.11 podemos ver que la clasificación de los saques y finales son mejores que la de los ataques y defensas. Podemos explicar esto ya que la lógica detrás de la clasificación de los saques y golpes finales es correcta y aplicable a cada punto: el primer golpeo siempre es un saque; el penúltimo golpeo es el finalizador (que provoca el final); y el último golpeo detectado se atribuye a un passing, la pelota yendo fuera o un golpeo dirigido a la red. Si esta clasificación falla será por culpa de una mala detección de otra información, como con los problemas descritos en las Secciones A.3 y A.4 del Apéndice. En cambio la clasificación de ataques y defensas, aparte de verse afectados por estos mismos problemas, también pueden fallar por culpa de estar limitados en cuanto a la cantidad de información que podemos extraer del vídeo para realizar los cálculos en la Sección 3.4.4 y a que estos mismos cálculos no sean capaces de cubrir todas las situaciones posibles.

Aun así, dados los resultados de este experimento consideramos que hemos cumplido con el objetivo de detección y clasificación de golpesos.

CONCLUSIONES Y TRABAJOS FUTUROS

En esta sección discutiremos las conclusiones generales acerca del sistema, sus procesos internos y los resultados obtenidos de los experimentos.

5.1. Conclusiones

El objetivo principal de nuestro sistema es realizar las detecciones y cálculos necesarios para poder obtener información relevante de los golpeos durante un punto en un partido de tenis.

El primer paso ha sido investigar tecnologías existentes en el ámbito de la visión artificial que nos fuesen útiles para nuestras detecciones. Además, también buscar proyectos ya existentes que realizasen trabajos parecidos a los aquí propuestos. Desarrollando, incorporando y ajustando estas tecnologías hemos logrado extraer información necesaria, pero no con la precisión y consistencia suficientes para inferir la información de carácter deportivo que deseábamos, como vemos en la tabla 4.1.

Para conseguir unos resultados suficientemente buenos, así como para hacer del sistema uno más eficiente y rápido, se han tenido que crear módulos extra dedicados a correcciones e interpolaciones. Con estos hemos logrado los resultados de la tabla 4.4, que ya son lo suficientemente buenos como para extraer a partir de ellos la información acerca de los golpeos que buscamos.

Conociendo con seguridad la posición de los jugadores, tanto en la imagen como sobre el plano de la pista, y la de la pelota, se han creado diferentes métricas que nos dirán como de aventajado está cada jugador en cada golpeo. Finalmente, con esto hemos conseguido calificar los golpeos y diferenciar qué jugador está atacando y cual defendiendo, así como mostrar en el vídeo información que nos ayude a visualizar esto.

Como vemos en los resultados de la tabla 4.9, hemos obtener la información objetivo con buenos resultados.

5.2. Trabajos futuros

En esta sección propondremos posibles trabajos futuros para mejorar o continuar con la idea presentada en este proyecto.

Detección y análisis de la pelota

Como vemos en la tabla comparativa 4.7, sin aplicar ninguna corrección o postprocesado obtenemos un gran número de pérdidas. A parte, incluso después del procesado, obtenemos detecciones erróneas que nos hacen perder precisión en la detección de la pelota y en la clasificación de los golpes, como explicamos en la Sección A.

Para mejorar la detección existirían varias opciones, como mejorar o ajustar el modelo TrackNet usado con un conjunto de datos más amplio. Otra forma podría ser limitar un área de búsqueda en cada frame en función de la posición anterior de la pelota.

El propósito de medir con mayor precisión esta información sería, aparte de mejorar los cálculos existentes, poder realizar un análisis más elaborado del movimiento de la pelota en la pista. Atendiendo a sus cambios bruscos de dirección se podría saber más información sobre en qué momento esta está siendo golpeada o está rebotando en el suelo. Esto no solo nos serviría para ajustar mejor los momentos de golpeo y la posición de los jugadores durante ellos, sino para extraer nueva información como la posición sobre la pista de la pelota, una estimación de su velocidad o saber si un golpe ha entrado o no. Con ello podríamos crear nuevas métricas para nuestro sistema de clasificación, explicado en la sección 3.4.4.

Partidos de dobles

El sistema actual está preparado para procesar vídeos en los que se juega un partido de tenis individual. Un futuro proyecto podría consistir en actualizar este sistema para funcionar también para partidos de dobles.

Los cambios planteados para lograr esto serían:

- Implementar algún algoritmo o comprobación en el módulo 3.2.3 para detectar automáticamente si hay 1 o 2 jugadores por cada lado de la pista.
- Ahora que hay 2 jugadores en cada lado, modificar en el módulo 3.4.1 la forma de detectar si un frame es de golpeo, ya que la forma actual de comparar la altura en la imagen de la pelota con la de los jugadores no nos sirve tal cuál está pensada.
- Modificar el cálculo de las métricas del módulo 3.4.4 o crear nuevas para posibilitar la clasificación de los golpes con 4 jugadores en pista.

BIBLIOGRAFÍA

- [1] “Tennis data innovations and tennisviz unveil new fan insights. partnership announcement,” 2022. Disponible en: enlace.
- [2] U. de GitHub r3curs10n, “tennis-cv,” 2018. Disponible en: enlace.
- [3] OpenCV, *Hough Line Transform*. Disponible en: enlace.
- [4] U. de GitHub sethah, “deeptennis,” 2019. Disponible en: enlace.
- [5] R. Gandhi, “R-cnn, fast r-cnn, faster r-cnn, yolo — object detection algorithms,” 2018. Disponible en: enlace.
- [6] PyTorch, *Docs >Models and pre-trained weights >Faster R-CNN >fasterrcnn_resnet50_fpn*. Disponible en: enlace.
- [7] Sergiu Waxmann, Burhan, Glenn Jocher, *Comprehensive Guide to Ultralytics YOLOv5*. Ultralytics, 2023. Disponible en: enlace.
- [8] C.-H. C. Yu-Chuan Huang, I-No Liao, “Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications,” *IEEE Computer Society*, p. 8, 2019. enlace.
- [9] U. de GitHub yastrebksv, “Tracknet,” 2022. Disponible en: enlace.
- [10] S. Sung, “Track a high speed moving object with tracknet,” 2021. Disponible en: enlace.
- [11] “Itf rules of tennis.” Disponible en: enlace.
- [12] OpenCV, *Feature Matching + Homography to find Objects*. Disponible en: enlace.
- [13] “Dimensiones de una pista de tenis.” Disponible en: enlace.
- [14] “Top 50 atp shots and rallies of 2010s decade!,” 2020. Disponible en: enlace.
- [15] “Rafael nadal vs novak djokovic - resumen de la final | roland-garros 2020,” 2020. Disponible en: enlace.
- [16] “Serena williams v venus williams extended highlights | australian open 2017 final,” 2022. Disponible en: enlace.
- [17] “Carlos alcaraz vs novak djokovic first-ever match! | madrid 2022 extended highlights,” 2023. Disponible en: enlace.
- [18] “Carlos alcaraz vs novak djokovic: Extended highlights | wimbledon 2023 final,” 2020. Disponible en: enlace.
- [19] “Guias de tenis >la tecnica del tenis.” Disponible en: enlace.
- [20] P. Hallaj, “What is intersection over union (iou) in object detection?,” *Medium*, 2023. enlace.
- [21] “Evaluación de los resultados del experimento de aprendizaje automático automatizado,” 2023. enlace.
- [22] Ploomber, *Micro and Macro Averaging*. Disponible en: enlace.
- [23] “Jan-lennard struff vs frances tiafoe for the title | stuttgart 2023 final highlights,” 2023. Disponible en: enlace.

- [24] “Feliciano lopez plays final atp match in mallorca! | mallorca 2023 highlights,” 2023. Disponible en: enlace.
- [25] “Roger federer vs rafael nadal wimbledon 2019 semi-final highlights,” 2019. Disponible en: enlace.
- [26] “Crazy atp grass-court tennis points!,” 2022. Disponible en: enlace.

APÉNDICES

ERRORES RECONOCIDOS EN LOS RESULTADOS DE LOS EXPERIMENTOS

En esta sección describiremos varios casos de error descubiertos durante los experimentos, explicaremos qué pruebas se han hecho al respecto, qué conclusiones sacamos al respecto y cómo afectarán al sistema.

A.1. Mala clasificación de pista completa en vídeos de Wimbledon

En el experimento descrito en la Sección 4.1.1 se ha detectado una situación en la que nuestro clasificador de frames según si se está enfocando la pista completa, explicado en la Sección 3.2.1: en las secuencias de frames del vídeo de frames aleatorios (explicado en la Sección 4.1.1) en las que aparecen fragmentos del vídeo “Alcaraz vs Djokovic en Winbledon 2023” [18] nuestro clasificador no funciona.

Si procesamos el vídeo original entero, veremos que no hay ningún enfoque a pista reconocido correctamente, sino que todos los frames del vídeo se clasifican como no enfoques a pista completa. Viendo el vídeo de frames aleatorios usado en este experimento, vemos que hay varios sets de frames en los que se juega sobre hierba y que sí se clasifican corretamente, ninguno de ellos de Wimbledon.

Al ser Wimbledon, y la pista ser de hierba, se podría pensar que el clasificador tiene un problema reconociendo este tipo de pista. Para comprobarlo, procesaremos los vídeos “Struff vs Tiafoe en Stuttgart 2023” [23], “Feliciano López vs Hanfman en Mallorca 2023” [24] y “Federer vs Nadal en Wimbledon 2019” [25], tres partidos jugados en hierba y aparentemente sin diferencias visuales importantes, como se ve en la figura A.1. Con los dos primeros vídeos el clasificador funciona correctamente; en cambio con el tercero, que también es en Wimbledon, ocurre el mismo problema en el que todos los frames se clasifican como no enfoques a pista completa.



(a) Pista detectada en Mallorca



(b) Pista detectada en Stuttgart



(c) Pista no detectada en Wimbledon

Figura A.1: Comprobación de enfoques a pistas de hierba

Para continuar con las comprobaciones sobre este problema hemos procesado un último vídeo, un recopilatorio exclusivamente de puntos jugados en hierba [26], entre los que no se encuentra ningún punto jugado en Wimbledon. Sobre este vídeo no se ha observado ningún error en el clasificador.

Tras estas pruebas podemos afirmar que existe un problema con la clasificación de frames según se enfoca la pista completa o no si el partido o punto se está jugando en Wimbledon, pero al no haber reconocido la causa de esto no sacaremos más conclusiones sobre este tema.

Al tratarse de un caso excepcional, en este experimento y en los resultados de la tabla 4.1 no se han tenido en cuenta las validaciones de los frames extraídos del vídeo original “Alcaraz vs Djokovic en Winbledon 2023” [18].

A.2. Carteles en la red que cortan las líneas de la pista

En el experimento descrito en la Sección 4.1.1 se ha detectado una situación que provoca una mala detección de los puntos de la pista en los procesos explicados en la Sección 3.2.2.

En algunos partidos individuales se cuelgan carteles a los lados de la red, normalmente con anuncios de algún patrocinador. Cuando estos carteles quedan por encima de las líneas de pista de dobles y no tienen un color claro, estos pueden provocar que estas dos líneas queden cortadas y la detección final resulte en una como la visible en la figura A.2.

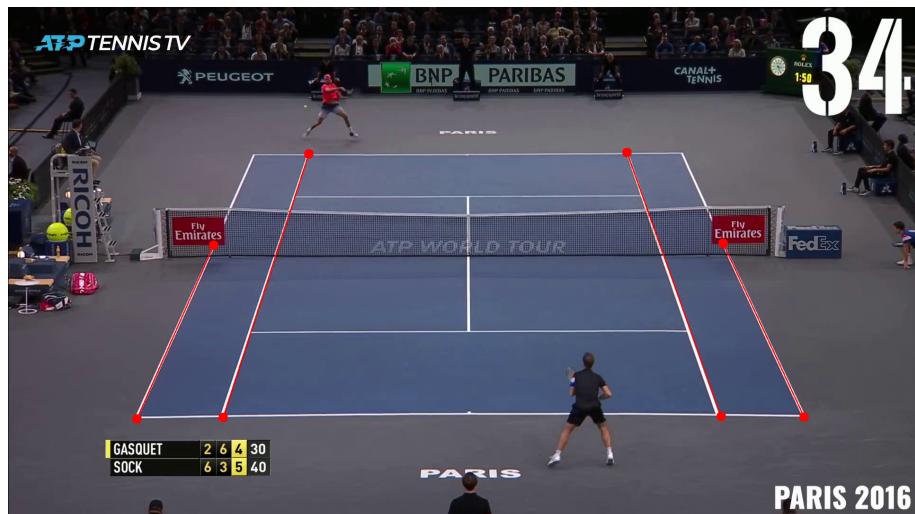


Figura A.2: Frame con puntos de pista (puntos en rojo) mal encontrados por haber carteles en la red

A.3. Momentos de golpeo acortados por mala detección de la pelota

Revisando los resultados del experimento descrito en la Sección 4.3.3 y el vídeo resultante, se ha encontrado que 2 de las falsas detecciones producidas quedan muy cerca temporalmente de las 2 detecciones no realizadas. Hemos visto que una mala detección de la pelota provoca que estos “momentos de golpeo” calificados como falsas detecciones acaben prematuramente, lo suficiente como para no englobar temporalmente sus “momentos reales de golpeo”, provocando que estos se califiquen como golpes no detectados.

Una de estas situaciones es la representada en la figura A.3. En esta secuencia vemos como al interpolar la coordenada de la pelota entre los frames b) y f), provocamos que el “momento de golpeo” termine prematuramente provocando que no englobe los frames del “momento real de golpeo” etiquetado, resultando en la validación en un golpeo no detectado (False Negative) y en una falsa detección (False Positive).



(a) Frame antes del golpeo, con pelota detectada correctamente



(b) Frame detectado como golpeo (impacto), con pelota detectada correctamente dentro de la región de golpeo del jugador (explicada en la Sección 3.4.1)



(c) Frame detectado como golpeo, con la pelota alejándose del jugador a causa de la interpolación entre los frames b) y f)



(d) Frame no detectado como golpeo, con la pelota alejándose del jugador a causa de la interpolación entre los frames b) y f)



(e) Frame correspondiente al impacto real, no detectado como golpeo, con la pelota alejándose del jugador a causa de la interpolación entre los frames b) y f)



(f) Frame posterior al golpeo real y al golpeo detectado, con la pelota detectada correctamente

Figura A.3: Ejemplo de mala detección de golpeos provocada por mala detección de la pelota. Detección de la pelota representada por punto verde, trayectoria de la pelota representada con una línea verde o roja (si el frame es de impacto), clasificación del golpeo (si se ha detectado) en letras rojas junto al jugador, tipo de golpeo con el que se ha etiquetado un frame en letras rojas a la izquierda de la imagen, posición del frame en el vídeo en la parte superior izquierda de la imagen.

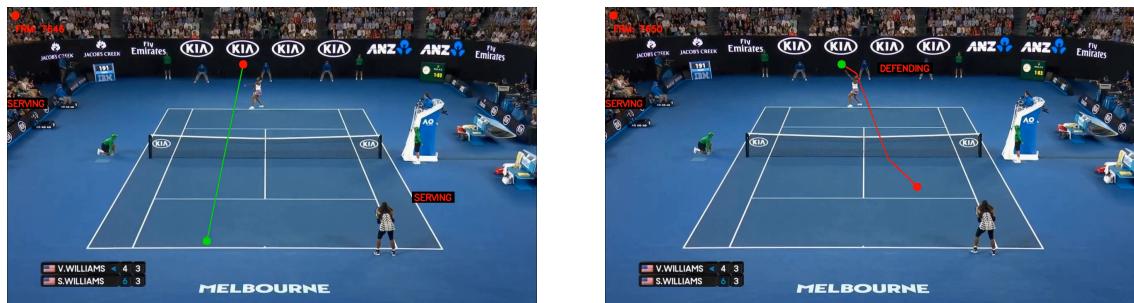
La secuencia mostrada en la figura A.3 ocurre cuando coincide que:

- Sejamos sin procesar varios frames (4 como se explica en el módulo 3.2.4, del 3416 al 3420 en la figura).
- En los siguientes se producen pérdidas en la detección de la pelota (en los frames del 3421 al 3425 en la figura), alargando el rango de frames a interpolar.
- Al interpolar las coordenadas de la pelota, esta no entra en la región de detección de golpeo, haciendo que no se detecte, o entra parcialmente, pudiendo provocar que el “momento de golpeo” (frames del 3416 al 3418 en la figura) no englobe el “momento de golpeo real” etiquetado (frames del 3419 al 3426 en la figura).

El uso de la técnica de la interpolación es parcialmente causante de esta situación, aun que es una situación poco común y, en general se han conseguido unos buenos resultados a la hora de detectar los golpeos.

A.4. Falsos momentos de golpeo por mala detección de la pelota

Al estudiar los resultados del experimento descrito en la Sección 4.3.3 y el vídeo resultante se ha encontrado la causa de 2 de las falsas detecciones de “momentos de golpeo”, que a su vez causan 2 clasificaciones incorrectas de saques. El error se ha producido por una detección incorrecta de la pelota como el de la figura A.4(a), en el que la coordenada errónea de la pelota está dentro de la región de detección de golpeo del jugador que está preparado para restar, provocando que se clasifique ese frame como saque del jugador inferior. Después, cuando se detecta correctamente la pelota como en la figura A.4(b), se detecta clasifica correctamente el frame como “momento de golpeo” del jugador superior, pero al haber clasificado el falso momento de golpeo anterior como saque, este se clasificará como de ataque o defensa.



(a) Frame con pelota mal detectada (punto verde), provocando que se atribuya el golpeo del frame al jugador inferior. Como es el primer momento de golpeo, se clasifica como saque.

(b) Frame con pelota detectada correctamente, atribuyendo el golpeo al jugador superior. Como el momento de golpeo en el que se encuentra la imagen a) se ha calificado como de saque, el momento de golpeo de este frame se calculará según la métricas para ataque o defensa.

Figura A.4: Ejemplo de mala detección de saque provocada por mala detección de la pelota. Detección de la pelota representada por punto verde, trayectoria de la pelota representada con una línea verde o roja (si el frame es de impacto), clasificación del golpeo (si se ha detectado) en letras rojas junto al jugador, tipo de golpeo con el que se ha etiquetado un frame en letras rojas a la izquierda de la imagen, posición del frame en el vídeo en la parte superior izquierda de la imagen.

ETIQUETADO DE VÍDEOS CON MATLAB

Para los experimentos explicados en la Sección 4 se han etiquetado los 3 vídeos explicados en esta misma Sección. Para el etiquetado se ha hecho uso de la herramienta MatLab, con la que se han etiquetado todos los frames de cada vídeo de la siguiente forma:

Frames con enfoque a pista completa No se ha etiquetado específicamente qué frames están enfocando la pista completa, sino que si etiquetamos los puntos de ella, en el post-procesado de los datos exportados de MatLab se marcará el frame como tal.

Puntos de la pista Se han etiquetado 2 polígonos rectangulares, uno con sus esquinas en los 4 puntos que definen la pista individual y el otro en los 4 puntos que definen la de dobles. Al exportar los datos, extraeremos los 8 puntos y los ordenaremos de la misma forma que se guardan en la Sección 3.2.2.

Jugadores Se ha etiquetado un rectángulo alrededor de cada jugador. Al exportar los datos, los procesaremos para obtener unos bounding boxes definidos igual que en la Sección 3.2.3.

Pelota Se ha etiquetado con un punto la posición de la pelota en la imagen.

Golpeos Si queremos etiquetar un frame como de golpeo de un jugador y de un tipo (saque, ataque, defensa, fina), lo marcaremos con punto en cualquier parte de la imagen. Al exportar y post-procesar el etiquetado, así reconoceremos los frames de golpeos.

El etiquetado de un frame enfocando a pista se vería como en la figura B.1.

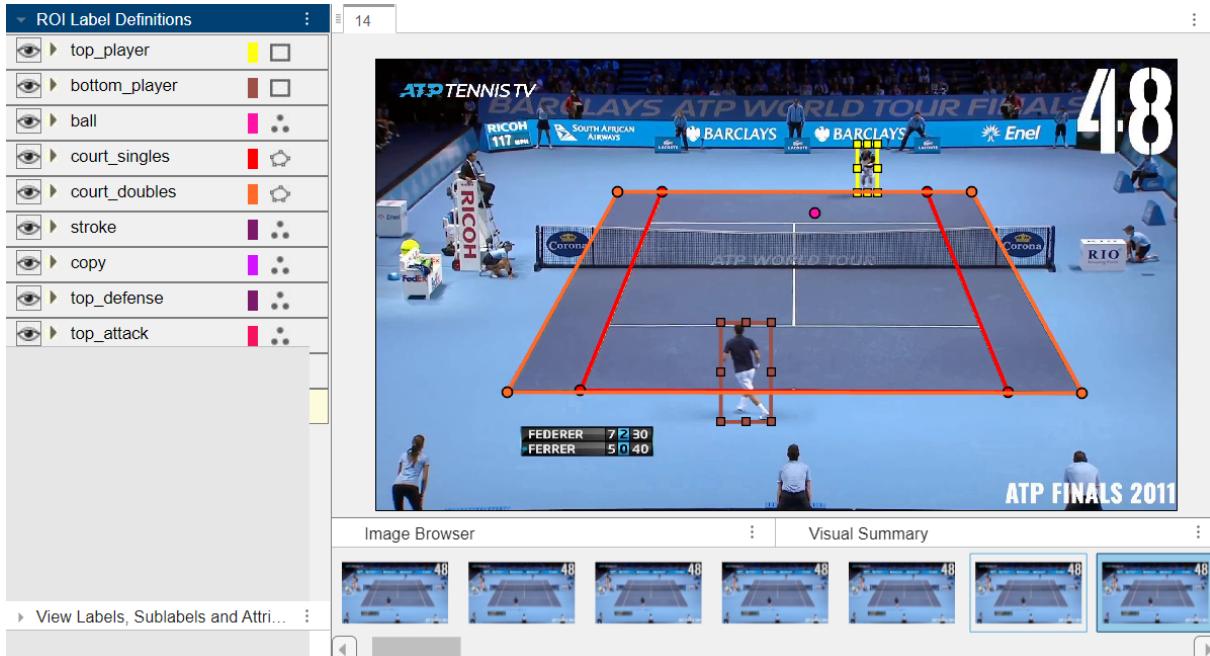


Figura B.1: Ejemplo del etiquetado con MatLab del vídeo de frames aleatorios, en el que se han marcado los 4 puntos definitores de la pista de dobles con un polígono rectangular naranja y los 4 de la pista individual con uno rojo, los bounding boxes de ambos jugadores con rectángulos amarillo y marrón y la pelota con un punto rosa.