UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Mejora de rendimiento deportivo mediante la detección de posturas por visión artificial

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Argente Martí, Vicent

Tutor/a: Ferri Ramírez, César

Cotutor/a: Monserrat Aranda, Carlos

Director/a Experimental: PALENCIA SANLEON, GLORIA

CURSO ACADÉMICO: 2022/2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Mejora de rendimiento deportivo mediante la detección de posturas por visión artificial

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Argente Martí, Vicent

Tutor: Ferri Ramírez, César
Monserrat Aranda, Carlos
Palencia Sanleón, Gloria

Curso 2022-2023

Resumen

La evolución de la informática ha conseguido que muchas tecnologías que anteriormente eran muy caras, hoy en día sean accesibles para cualquier persona. Uno de los ámbitos afectados es el reconocimiento postural, el cual puede ser de mucha utilidad en diversas actividades humanas.

Actualmente esta tecnología es muy utilizada en multitud de deportes profesionales, sin embargo existen muy pocas herramientas que no necesiten material y que se puedan ampliar fácilmente para satisfacer las necesidades del usuario.

Para ello, se ha utilizado la librería OpenPose para obtener la posición de diversos puntos del cuerpo. Con esta información, se ha investigado formas de tratar los datos, tanto para refinarlos como para obtener información útil de ellos. Todo esto se ha implementado mediante C++ en Visual Studio, aunque el proyecto está basado en CMake, por lo que se puede utilizar en cualquier plataforma compatible.

Específicamente, se ha implementado un prototipo de análisis de saque de tenis, quedando disponibles para futuros análisis las diversas herramientas desarrolladas de análisis de los datos.

Palabras clave: reconocimiento postural, OpenPose, C++, CMake, saque de tenis, análisis de los datos

Resum

L'evolució de la informàtica ha aconseguit que moltes tecnologies que anteriorment eren molt cares, hui dia siguen accessibles per a qualsevol persona. Un dels àmbits afectats és el reconeixement postural, el qual pot ser de molta utilitat en diverses activitats humanes.

Actualment aquesta tecnologia és molt utilitzada en multitud d'esports professionals, no obstant això hi ha molt poques eines que no necessiten material i que es puguen ampliar fàcilment per satisfer les necessitats de l'usuari.

Per aconseguir-ho, s'ha utilitzat la llibreria OpenPose per obtenir la posició de diversos punts del cos. Amb aquesta informació, s'ha investigat maneres de tractar les dades, tant per refinar-les com per obtindre informació útil. Tot això s'ha implementat mitjançant C++ a Visual Studio, encara que el projecte està basat en CMake, pel que es pot utilitzar a qualsevol plataforma compatible.

Específicament, s'ha implementat un prototip d'anàlisi de servei de tennis, de manera que queden disponibles per a futures anàlisis les diverses eines desenvolupades d'anàlisi de les dades.

Paraules clau: reconeixement postural, OpenPose, C++, CMake, servei de tennis, anàlisi de les dades

Abstract

The evolution of computing has made many technologies that were previously very expensive, today accessible to anyone. One of the areas affected is pose recognition, which can be very useful in various human activities.

Nowadays this technology is widely used in many professional sports, however there are few tools that do not require material and can be easily expanded to meet the needs of the user.

In order to achieve this, the OpenPose library has been used to obtain the position of various body keypoints. With this information, many methods to process the data have been researched, not only for refining it, but also for extracting useful information from it. All of this has been implemented with C++ in Visual Studio, although the project is based in CMake, allowing it to be used in any compatible platform.

Specifically, a tennis serve analysis prototype has been implemented, making the various data analysis tools developed to be available for future analyses.

Key words: pose recognition, OpenPose, C++, CMake, tennis serve, data analysis

Índice general

Índice general v
Índice de figuras vii
Índice de tablas vii

1 Introducción 1
1.1 Motivación . . . 1
1.2 Objetivos . . . 1
1.3 Estructura de la memoria . . . 2
2 Estado del arte 3
2.1 Tracking óptico . . . 3
2.1.1 YOLO . . . 4
2.1.2 OpenPose . . . 4
2.1.3 MoveNet . . . 5
2.2 Proyectos relacionados . . . 5
2.2.1 Football Players Tracking . . . 5
2.2.2 3D Football Players Pose Estimation . . . 6
2.2.3 Sports counting by pose estimation . . . 6
2.2.4 IncludeHealth . . . 6
3 Conceptos previos 7
3.1 Trayectoria . . . 7
4 Tratamiento de trayectorias 9
4.1 Puntos mal ubicados . . . 9
4.2 Normalización . . . 11
4.2.1 Posición . . . 12
4.2.2 Escala . . . 12
5 Medición de distancias 15
5.1 Distancias entre trayectorias . . . 15
5.1.1 Clasificación de medidas . . . 15
5.1.2 Dynamic Time Warping . . . 16
5.1.3 Funcionamiento de DTW . . . 16
5.2 Distancias entre ángulos . . . 17
5.2.1 Cálculo de ángulos . . . 17
5.2.2 Distancia entre ángulos . . . 18
5.3 Interpretación de distancias . . . 18
6 Detección de movimientos específicos 21
6.1 Posición relativa . . . 21
6.2 Cambios de velocidad . . . 22
6.3 Ángulos . . . 23
7 Aplicación en el tenis 25
7.1 Detección de movimientos . . . 25
7.1.1 Lanzamiento de la pelota hacia arriba . . . 25
7.1.2 Golpeo de la pelota con la raqueta . . . 26

7.2 Aspectos del saque . . . 27
7.3 Distancia entre ángulos . . . 27
7.4 Intervalos válidos . . . 28
7.5 Análisis de una muestra . . . 28
8 Implementación **31**
8.1 Diseño . . . 31
8.1.1 MoveComparator . . . 31
8.1.2 TennisServeComparator . . . 33
8.1.3 MoveAnalyzer . . . 33
8.1.4 TennisServeAnalyzer . . . 33
8.1.5 Skeleton . . . 33
8.1.6 Trajectory . . . 33
8.1.7 Estructuras de apoyo . . . 34
8.2 Pruebas . . . 34
8.3 Ejecución . . . 35
8.4 Uso de recursos . . . 36
9 Conclusiones **39**
9.1 Trabajo futuro . . . 39
Bibliografía **41**

Apéndices

A Configuración del sistema **43**
A.1 Fase de inicialización . . . 43
B Objetivos de desarrollo sostenible **45**

Índice de figuras

4.1 Trayectoria con errores puntuales. 9
4.2 Trayectoria con puntos corregidos. 10
4.3 Trayectorias sin normalizar. 11
4.4 Trayectorias con posición normalizada. 12
4.5 Trayectorias totalmente normalizadas. 13

5.1 Emparejamiento de puntos del DTW. 16
5.2 Triángulo de vértices A, B, C. 17
5.3 Triángulo dividido en dos triángulos rectángulos. 18
5.4 Función de obtención de porcentaje. 20

6.1 Cambios de velocidad. 23

7.1 Movimiento al lanzar la pelota hacia arriba. 26
7.2 Movimiento al golpear la pelota con la raqueta. 26

8.1 Diagrama de clases. 32
8.2 Ejecución de las pruebas unitarias. 35
8.3 Análisis de Open Pose. 35
8.4 Resultado de la ejecución. 36
8.5 Resultado de un vídeo ideal. 36

Índice de tablas

8.1 Especificaciones del equipo de pruebas. 37
8.2 Aceleración total de la aplicación. 38

CAPÍTULO 1
Introducción

A lo largo de la historia el deporte ha estado muy presente en la sociedad humana, y los atletas buscan mejorar lo máximo posible. A medida que ha ido avanzando la ciencia en ámbitos como la fisiología, biomecánica o nutrición, el rendimiento de los deportistas ha ido creciendo rápidamente. Además, con la introducción de la computación en este ámbito, se ha logrado realizar análisis muy precisos sobre diversos aspectos, como la técnica de un movimiento.

Sin embargo, este tipo de análisis normalmente requiere material específico y un equipo de expertos detrás, lo que resulta muy costoso. Es por esto que se han buscado alternativas que no requieran toda esta inversión de forma que cualquier persona pueda mejorar en el deporte que practica.

1.1 Motivación

La posibilidad de poder mejorar en un deporte sin la necesidad de disponer de mucho material y personal costoso es algo que puede ser de mucha utilidad a muchos practicantes de deporte comunes e incluso profesionales, puesto que estos elementos tienen un impacto positivo al momento de progresar.

Con el avance de la tecnología se ha hecho posible obtener los datos posturales sobre el cuerpo humano con materiales que mucha gente tiene, como son una cámara o teléfono móvil y un ordenador. De este forma, el análisis resulta mucho más accesible y, de este modo, mucha más gente tendrá la posibilidad de realizarlo.

1.2 Objetivos

Se quiere conseguir una aplicación, que con la información obtenida de un único vídeo, sin necesidad de utilizar más de una cámara simultánea, sea capaz de analizar un movimiento realizado por el deportista en distintos aspectos, de forma que se obtenga una puntuación para cada uno de ellos.

La finalidad es ofrecer una herramienta que permita a los deportistas conocer sus puntos débiles sin tener que realizar una gran inversión en material específico para este fin, permitiendo que algunas personas puedan disponer de una ayuda para iniciarse en un deporte, o que atletas más experimentados puedan pulir su técnica.

Todo esto se quiere realizar con una aplicación preparada para la ampliación a cualquier deporte de forma que no interfiera con ningún elemento existente, y sin entrenar

ningún tipo de red neuronal, de forma que un desarrollador común pueda agregar nuevo contenido con la información necesaria.

1.3 Estructura de la memoria

De forma general, la memoria trata los siguientes aspectos:

- Estado del arte: Se explica algunas tecnologías gratuitas existentes que pueden servir para el fin del proyecto.
- Tratamiento de los datos: Antes de analizar los datos obtenidos a partir de un vídeo, es necesario realizar un tratamiento de los datos, de forma que el análisis pueda ser mucho más preciso.
- Mediciones a partir de los datos: Este aspecto supone la mayor parte de la memoria. Se explica la forma de obtener y comparar la información que se obtiene a partir de los datos, obteniendo como resultado una puntuación.
- Aplicación en el tenis: Se explica cómo se puede utilizar todos los aspectos anteriores para analizar saques de tenis.
- Implementación: Se muestra una posible implementación del análisis para el saque de tenis, de forma que se realiza las operaciones necesarias para realizarlo de manera automática.
- Conclusiones y trabajo futuro: Se expone una reflexión de qué se ha conseguido y qué se puede hacer para mejorar el proyecto.

CAPÍTULO 2
Estado del arte

Durante mucho tiempo, el análisis de postura corporal ha sido de interés debido al amplio número de ámbitos en el que se puede aplicar. Puede ser útil desde la salud y la seguridad, hasta la mejora en el rendimiento deportivo. Incluso tiene utilidad en el cine, permitiendo incluir elementos que han sido animados de forma precisa y realista mediante reconocimiento postural.

Desde la aparición de la computación, se ha utilizado diferentes métodos para recoger esta información del mundo real y analizarla, ya sea posteriormente o en tiempo real. Algunos de estos requieren el uso de sensores específicos como acelerómetros [1] y así determinar la posición de las partes del cuerpo. Por otra parte, existen tecnologías que nos permiten obtener posturas corporales únicamente mediante el uso de una o más cámaras, de forma que se obtiene la postura a partir del análisis de imágenes.

Nos centraremos en estas últimas debido al poco material que requieren.

2.1 Tracking óptico

El tracking óptico [2] consiste en la obtención de la postura mediante imágenes. Para realizar este tipo de reconocimiento tan solo es necesario capturar con una o más cámaras la imagen o imágenes de la postura siendo realizada por una persona. Con esta información se podrá obtener la postura mediante algoritmos de visión por computador. Algunas ventajas de esta técnica son:

- Coste muy reducido: No se necesita ningún tipo de sensor especial, tan solo una cámara, por lo que no se necesita hacer una gran inversión.
- No invasivo: Al no tener que utilizar sensores, una persona puede realizar una actividad como lo haría normalmente mientras simplemente se le graba.

Por otra parte, también tiene algunas desventajas destacables:

- Baja precisión: Aunque la visión por computador ha mejorado mucho, es normal que pueda cometer errores, que pueden ir desde pequeños temblores, hasta equivocaciones en la identificación de los puntos.
- Alta carga computacional: El análisis de las imágenes no es sencillo, por lo que si queremos analizar las imágenes en tiempo real se necesitará un hardware muy potente, o si queremos analizarlas posteriormente puede tomar mucho tiempo.

Existen algunas herramientas públicas que nos permitirán la obtención de posturas a partir de imágenes.

2.1.1. YOLO

YOLO [3] es un modelo rápido, preciso y fácil de usar, siendo útil para una gran variedad de aplicaciones. En su última versión v8, se ha juntado muchas de las características de las versiones anteriores y se han añadido algunas optimizaciones y mejoras.

Este modelo está conformado por distintos modelos que permiten realizar las siguientes tareas:

- Clasificación: Es capaz de reconocer el elemento que se muestra en una imagen dada.
- Detección: Además de clasificar varios objetos en la misma imagen, es capaz de identificar su posición.
- Segmentación: Es capaz de reconocer varios objetos en una imagen, identificando con alta precisión sus bordes.
- Rastreo: Es capaz de detectar varios objetos en una secuencia de imágenes, rastreando e indicando la trayectoria que siguen.
- Reconocimiento postural: Esta es la característica más llamativa para el objetivo del proyecto, puesto que permite identificar varios puntos clave del cuerpo humano, únicamente con el uso de imágenes.

Este último modelo y la facilidad de instalación, son características favorables para convertir a YOLO en una excelente opción para realizar tracking óptico.

2.1.2. OpenPose

OpenPose [4] es un sistema de reconocimiento postural. Es reconocido por ser capaz de detectar puntos clave de las manos, pies, cara y cuerpo en tiempo real. Al ser una herramienta destinada únicamente a este fin ofrece características interesantes. Algunas de estas son:

- Detección en 2D de múltiples personas: Permite la detección de puntos del cuerpo, de las manos, los pies y la cara para más de una persona simultáneamente. Según la cantidad de puntos que se quiera detectar, será capaz de detectar un mayor o menor número de personas.
- Detección en 3D de una persona: Es capaz de ofrecer coordenadas en 3D de los puntos clave si disponemos de la escena grabada desde dos puntos conocidos. Para ello, la librería facilita el proceso de triangulación y calibración entre las grabaciones para poder ofrecer un buen resultado.

Una virtud muy interesante de OpenPose es que cuando únicamente se necesita detectar 25 puntos clave del cuerpo en 2D, es capaz de estimar la postura para un número indefinido de personas sin reducir su rendimiento. Esta característica puede ser muy útil en contextos en los que se necesite analizar varias personas, como en un partido de fútbol.

Se utilizará esta herramienta a lo largo del proyecto debido a la gran calidad que ofrece, pudiéndose ejecutar completamente de forma local sin ningún tipo de conexión con un servicio externo. Además, ofrece una amplia documentación y ejemplos que permiten reducir el tiempo de aprendizaje para utilizarla.

2.1.3. MoveNet

MoveNet [5] es un modelo de reconocimiento postural desarrollado por Google Research. Es capaz de detectar 17 puntos clave del cuerpo. Su utilización en el deporte es muy interesante debido a que presenta dos características que lo hacen muy adecuado en este contexto:

- Análisis en tiempo real: El modelo está optimizado para ser capaz de procesar las imágenes en tiempo real en un equipo promedio actual. Además ofrece dos variantes llamadas "Lightning" y "Thunder" orientadas a ofrecer menor latencia y más precisión respectivamente.
- Reconocimiento postural para deporte: El modelo ha sido entrenado enfocándose en algunos movimientos deportivos que pueden provocar resultados erróneos cuando se utilizan otros modelos.

Debido a esto, su mayor atractivo se encuentra en la realización de aplicaciones que analicen y ofrezcan información en tiempo real mientras se realiza el ejercicio. Esto puede resultar de gran utilidad en algunos casos como saber al instante si se está realizando un ejercicio de fuerza con un rango de movimiento adecuado, o saber si la posición es adecuada a lo largo de la realización de un ejercicio estático, entre muchos otros.

2.2 Proyectos relacionados

A continuación se exponen brevemente tres proyectos relacionados con la visión por computador aplicada en deportes.

2.2.1. Football Players Tracking

Este proyecto [6] utiliza YoloV5 y ByteTrack para identificar todos los elementos que se encuentran en un partido de fútbol. Concretamente, este proyecto puede realizar las siguientes tareas:

- Clasificar elementos: Es capaz de identificar qué es cada elemento, es decir, sabe donde se encuentra la pelota y cada una de las personas que aparecen en pantalla. Además, es capaz de distinguir si cada persona es un árbitro o un jugador.
- Rastrear elementos: Es capaz de seguir a cada uno de los elementos, asignándoles un identificador único a cada uno de ellos, sin que este cambie a lo largo del análisis del vídeo para cada elemento.
- Posicionamiento y posesión del balón: Es capaz de localizar en todo momento la pelota. Además hace uso de mediciones de distancias para determinar qué jugador está en posesión de la pelota, o predecir qué jugador tiene más probabilidad de conseguir la posesión si nadie la tiene en un momento dado.

Toda esta información puede ser utilizada para obtener mucha información sobre el transcurso del partido y obtener muchas estadísticas de forma automatizada, únicamente con una grabación del partido.

2.2.2. 3D Football Players Pose Estimation

Este proyecto [6] utiliza YoloV7 para diferenciar a un jugador de fútbol de la pelota, y estimar la postura del deportista.

Para ello, se necesita disponer de dos grabaciones del mismo movimiento desde distintos puntos de vista. De esta forma, se puede obtener los puntos clave del cuerpo del deportista en 3D. Esta tarea es muy similar a la que realiza el VAR [7] (Video Assistant Referee) para determinar de forma muy precisa si un jugador se encuentra fuera de juego.

En este caso, el fin del proyecto es comprobar la dificultad de implementar un sistema similar a la detección de un jugador fuera de juego, pero tiene un gran potencial para analizar aspectos sobre la postura del deportista, de forma que se pueda corregir y mejorar si es oportuno.

2.2.3. Sports counting by pose estimation

Este proyecto [8] utiliza la variante "Lightning" de MoveNet para determinar la postura corporal.

La aplicación consiste en contar el número de repeticiones que se realizan a partir de la información que ofrece la postura. Concretamente se ha implementado para contar repeticiones de abdominales o "sit up", aunque la misma idea podría ser utilizada para realizarse con ejercicios similares.

Debido a la elección del modelo, la aplicación se puede ejecutar en tiempo real incluso en dispositivos móviles, lo cual la convierte en una herramienta versátil.

2.2.4. IncludeHealth

IncludeHealth [9] es una empresa que se dedica a la salud musculoesquelética y su objetivo es convertir estos cuidados en algo accesible para todo el mundo.

Su aplicación utiliza MoveNet para determinar la postura corporal. Gracias a esto, la aplicación es capaz de ofrecer distintos ejercicios y rutinas a sus pacientes de forma interactiva desde un dispositivo móvil o un ordenador.

Durante la realización de los ejercicios, ofrece información que permite al usuario conocer la calidad de la técnica que esta realizando en tiempo real. Por ejemplo, al realizar una sentadilla, indicará cuándo se ha realizado el rango de movimiento completo, o en una postura de yoga, puede indicar el ángulo de inclinación del cuerpo y el tiempo que falta para terminar de realizar la postura.

En definitiva, este proyecto, al ser realizado por un equipo de especialistas en los distintos ámbitos que esto envuelve, es un claro ejemplo de la calidad que puede alcanzar una tecnología que es accesible para todos y cómo esta se puede utilizar para mejorar la calidad de vida de las personas.

CAPÍTULO 3
Conceptos previos

En este capítulo se introducirán algunos conceptos para especificar de forma precisa cómo serán considerados, ya que su definición puede variar según el contexto en el que se use. Además, algunos pueden comprender elementos que también conviene explicar.

3.1 Trayectoria

Una trayectoria [10] puede ser definida como una función continua que describe el movimiento que sigue un objeto. Sin embargo, en la realidad las maneras de capturar las posiciones de un objeto funcionan tomando muestras cada cierto tiempo, es decir, funcionan de forma discreta. Esto provocará que obtengamos la trayectoria como una serie de puntos que indican la posición del objeto en cada instante de tiempo. Aunque existen formas de obtener aproximaciones continuas a la trayectoria original a partir de las muestras tomadas, en este caso serán tratadas como series de puntos. A continuación se introducirá algunos elementos que se pueden obtener de una trayectoria T representada como una serie de puntos:

- p: Es un punto, en este caso bidimensional, muestra de la trayectoria. Nos referiremos a su primer componente como $p.x$ y a su segundo componente como $p.y$.
- $d(p_i, p_j)$: Es la distancia entre dos puntos cualesquiera. Se puede utilizar cualquier medida de distancia según las necesidades del problema. En este caso se utilizará la distancia euclídea.
- "Size(T)" o Talla(T): Es el número de muestras o puntos que forman a la trayectoria. La llamaremos n para abreviar, a menos que se haga otra indicación.
- "Length(T)" o Longitud(T): Es la longitud total de la trayectoria, es decir, cuánto mide en el espacio. Se define de la siguiente forma:

$$Length(T) = \sum_{i=1}^{n-1} d(p_i, p_{i+1})$$

- "Centroid(T)" o Centroide(T) [11]: Es el centro de masas C de todos los puntos de la trayectoria, es decir, la posición promedio de todos los puntos. Se define de la siguiente forma:

$$Centroid(T) = \frac{\sum_{i=1}^{n} p_i}{Size(T)}$$

Además, también existen algunas funciones que permiten extraer o descartar ciertos elementos de una trayectoria. Algunas de estas son:

- "Head(T)": Extrae únicamente el primer elemento de la trayectoria. Es decir, para $T = [t_1, t_2, t_3, ..., t_n]$, el resultado será t_1.
- "Rest(T)": Representa a la misma trayectoria dada sin el primer elemento. Es decir, para $T = [t_1, t_2, t_3, ..., t_n]$, el resultado será $[t_2, t_3, ..., t_n]$.

CAPÍTULO 4
Tratamiento de trayectorias

En este capítulo se verá algunos problemas que pueden surgir con las trayectorias obtenidas y posibles soluciones.

4.1 Puntos mal ubicados

Aunque en la mayoría de casos los resultados obtenidos por OpenPose son muy buenos, no siempre es así.

Cabe la posibilidad de que la librería no sea capaz de detectar una zona del cuerpo aunque esta sea visible. En este caso, lo que ocurre es que se situará este punto en las coordenadas (0, 0). Otra posibilidad es que sí que la detecte pero en una posición totalmente errónea. En este caso, lo único que ocurre es que el punto estará situado en un lugar erróneo.

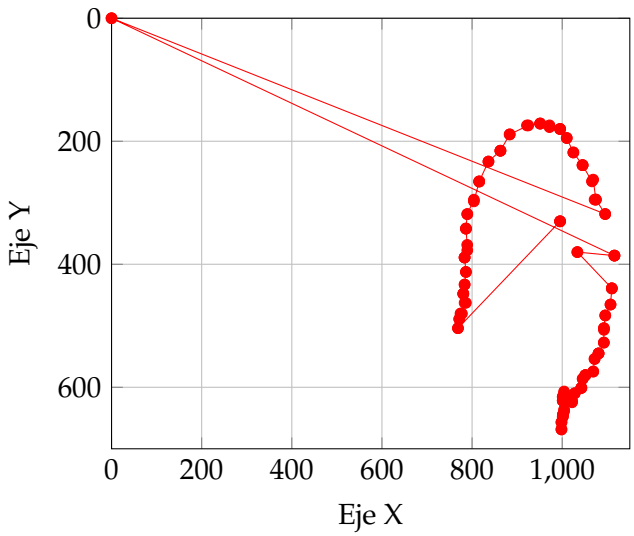

Figura 4.1: Trayectoria con errores puntuales.

Estos errores pueden afectar a la precisión del análisis, pero afortunadamente, existe un valor de confianza que podemos utilizar para detectar posiciones erróneas. Para ello, podemos considerar como incorrectos aquellos puntos que no superen un cierto umbral de confianza. Este umbral deberá ser lo suficientemente alto como para que no tengamos ningún punto en una posición incorrecta, pero deberá estar lo más ajustado posible para

que no considere como malos a puntos que realmente sí que estaban bien. Después de varias pruebas, se ha determinado que un umbral de 70 % de confianza cumple estas dos condiciones.

Una vez tenemos una forma de descartar puntos no válidos, hay varias opciones:

- Descartar el punto por completo.
- Estimar la posible localización real del punto a partir de la información que sí es correcta.

Debido a que los errores ocurren de forma puntual, y en muchos casos disponemos de la información de los fotogramas de alrededor (el anterior y el siguiente) del fotograma incorrecto, se ha optado por realizar una estimación de la posición del punto. La forma más sencilla es realizar una segmentación lineal [12], es decir, que la posición estimada se encontrará en la recta que pasa por el último punto correcto que se ha encontrado, y el siguiente correcto. De esta forma el problema se reduce a dividir una recta en n + 1 segmentos de tamaño similar, siendo n el número de fotogramas incorrectos consecutivos que se encuentran entre dos fotogramas correctos.

Aunque lo más común es que haya a lo sumo uno o dos puntos incorrectos consecutivos entre dos correctos, se deben contemplar los siguientes casos:

- Se ha encontrado un fotograma incorrecto antes que uno correcto, en cuyo caso se ha decidido que la posición estimada de ese punto será la del siguiente punto correcto.
- Se ha encontrado un fotograma incorrecto y ya no existe ninguno correcto posterior a este, en cuyo caso se ha decidido que la posición estimada de ese punto será la posición del último fotograma correcto encontrado.
- No se ha encontrado ningún fotograma correcto. En este caso no existe información con la que poder estimar ninguna posición. Si ocurre este caso, es probable que el punto clave no sea visible en ningún momento del vídeo.

Con esta estimación ya disponemos de una trayectoria mucho más suave y con la que no perderemos mucha precisión.

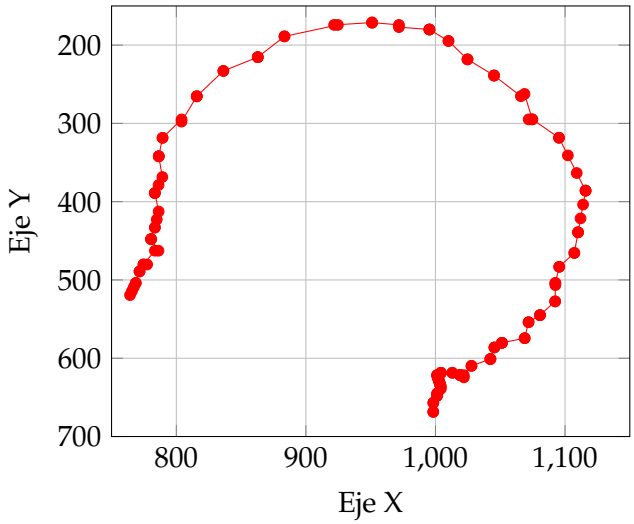

Figura 4.2: Trayectoria con puntos corregidos.

4.2 Normalización

Una vez se han corregido los errores puntuales que puede contener una trayectoria, ya podemos tratar de comparar trayectorias con el fin de determinar qué tanto se parece una trayectoria dada a una trayectoria ideal. Sin embargo, debemos tener en cuenta que solo nos interesa comparar la forma de las trayectorias, y debido a que los vídeos han podido ser tomados desde distancias o posiciones ligeramente diferentes, deberemos asegurar que la posición y la escala de las trayectorias son lo más parecidas posible.

En la Figura 4.3 se muestra de forma gráfica una trayectoria base, que se considera ideal, junto a una muestra que corresponde al mismo movimiento realizado por otro deportista. Como se puede observar, aunque sus formas son relativamente parecidas, su posición y tamaño no lo son:

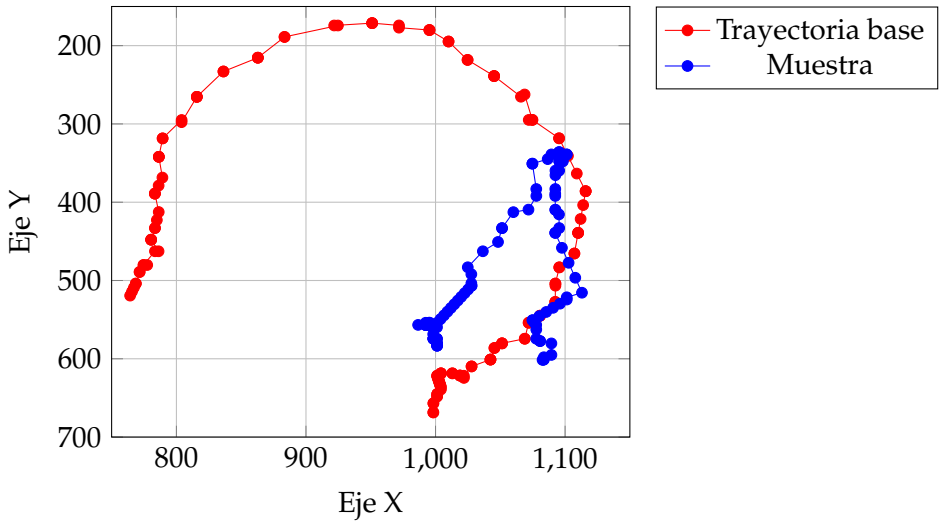

Figura 4.3: Trayectorias sin normalizar.

Nuestro objetivo es conseguir que ambas trayectorias tengan la misma posición y escala. Para ello existen dos opciones:

- Transformar ambas trayectorias para que su posición y escala tengan un valor preestablecido.
- Transformar únicamente una de las dos trayectorias para que su posición y escala tengan el mismo valor que en la otra trayectoria.

Cualquiera de estas dos opciones es válida para realizar una comparación de dos trayectorias, pero debido a que será necesario comparar más de dos trayectorias entre sí, es más conveniente establecer una posición y escala fija a todas ellas.

Para conseguir esto, deberemos encontrar un criterio común a todas las trayectorias para determinar lo que es su posición y su escala. Un procedimiento comúnmente utilizado es la unidad tipificada o "z-score normalization" [13], que consiste en hacer que el promedio de todos los elementos sea cero, y su desviación típica sea uno. A continuación veremos cómo esto se puede aplicar a una trayectoria formada por puntos bidimensionales.

4.2.1. Posición

En cuanto a la posición, debemos modificar la trayectoria T de forma que al obtener el promedio o centroide de todos sus puntos, este sea $(0, 0)$. Para ello, debemos obtener el centroide $C = Centroid(T)$. Una vez calculado el centroide, bastará con restárselo a cada uno de los puntos de T, obteniendo una nueva trayectoria T_{pos} cuyo centroide se encuentra en el origen. Por tanto, la trayectoria T_{pos} se obtiene de la siguiente forma:

$$T_{\text{pos}} = \{(p_i.x - C.x, p_i.y - C.y) | p_i \in T\}$$

En la Figura 4.4 se observa cómo la posición de la muestra se ha modificado para que su centroide coincida con el de la trayectoria base:

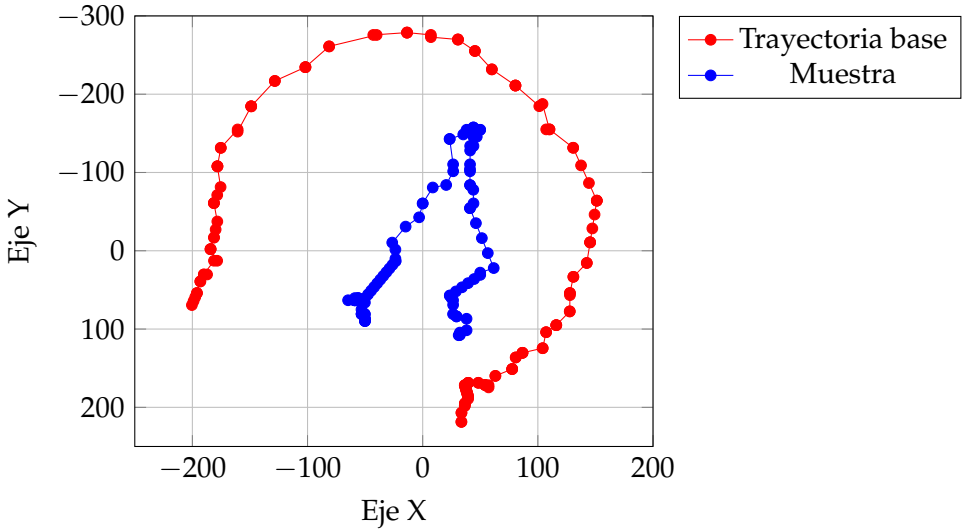

Figura 4.4: Trayectorias con posición normalizada.

Como se puede observar, ambas trayectorias comparten centroide, el cual se encuentra en el origen del plano de coordenadas.

4.2.2. Escala

El siguiente paso consistirá en modificar la escala de la trayectoria T de forma que al obtener su escala, esta sea igual a uno. Para ello, podemos considerar que la escala de una trayectoria es directamente proporcional a la dispersión de los puntos de la trayectoria T respecto a su centroide $C = Centroid(T)$. Es decir, obteniendo la desviación típica, tendremos un valor que podemos considerar escala.

En este caso, obtendremos la dispersión y aplicaremos el reescalado para cada componente de los puntos por separado. De esta forma, estableceremos la relación de aspecto de la trayectoria en 1:1, reduciendo el impacto de las proporciones corporales del deportista sobre la comparación. Por tanto, la escala se puede definir como:

$$\vec{\sigma} = \left(\sqrt{\frac{\sum_{i=1}^{n} (p_i.x - C.x)^2}{n}}, \sqrt{\frac{\sum_{i=1}^{n} (p_i.y - C.y)^2}{n}} \right)$$

Una vez obtenida la escala, en el caso general, para obtener la trayectoria reescalada T_{scl} debemos mover el centroide de la trayectoria al punto de origen, dividir cada componente entre su correspondiente escala, y volver a desplazar el centroide a su posición original.

Es decir, T_{scl} se obtiene de la siguiente forma:

$$T_{\text{scl}} = \left\{ \left(\frac{p_i.x - C.x}{\vec{\sigma}.x} + C.x, \frac{p_i.y - C.y}{\vec{\sigma}.y} + C.y \right) | p_i \in T \right\}$$

En este caso, como nuestro objetivo es hacer que el centroide de la trayectoria permanezca en el origen del sistema de coordenadas, no debemos devolver la trayectoria a su posición original tras aplicarle el reescalado. Por tanto, la función para obtener una trayectoria T' normalizada en cuanto a posición y escala se puede definir como:

$$T' = ZNormalize(T) = \left\{ \left(\frac{p_i.x - C.x}{\vec{\sigma}.x}, \frac{p_i.y - C.y}{\vec{\sigma}.y} \right) | p_i \in T \right\}$$

En la Figura 4.5 se muestra como quedan ambas trayectorias tras realizar los ajustes de posición y escala:

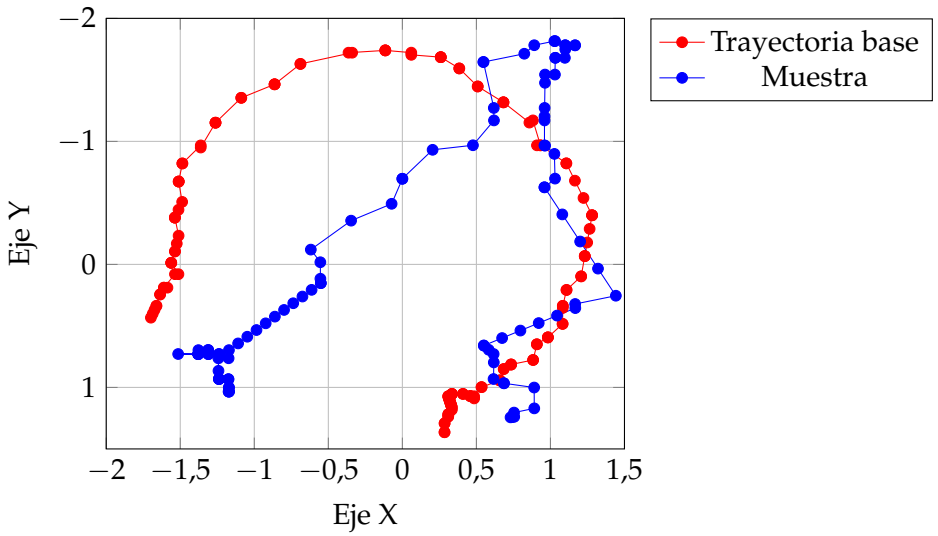

Figura 4.5: Trayectorias totalmente normalizadas.

Como se puede observar en los valores de los ejes, la escala de ambas trayectorias se ha disminuido de forma drástica, y ahora las dos tienen un tamaño similar.

CAPÍTULO 5
Medición de distancias

Una vez las trayectorias tienen la misma posición y escala, podemos compararlas entre ellas para obtener diferentes medidas de distancias. De esta forma se podrá comparar una trayectoria dada con una trayectoria ideal y obtener un valor de similitud entre ellas teniendo en cuenta diferentes aspectos.

Durante este capítulo se explica algunas medidas de distancia que pueden ser de utilidad para este fin.

5.1 Distancias entre trayectorias

Medir una distancia entre trayectorias consiste en obtener una medida de similitud basada en diferentes aspectos de esta. Existen diferentes algoritmos para realizar estas medidas. Cada uno ofrecerá diferentes resultados basados en diferentes aspectos de las trayectorias.

5.1.1. Clasificación de medidas

Los algoritmos de medidas de similitud entre trayectorias [10] se pueden clasificar en dos grandes grupos basándose en los siguientes aspectos:

- Únicamente secuencia o "sequence-only": Este tipo de medidas únicamente tiene en cuenta los aspectos espaciales de la trayectoria. Es decir, se tendrá en cuenta la posición de los puntos que conforman la trayectoria y no el instante de tiempo en que se han tomado. Son útiles en los casos en los que solo interesa comparar la forma de dos trayectorias.

- Espacio temporal: Este tipo de medidas tiene en cuenta tanto los aspectos espaciales como los temporales. Es decir, además de la posición de los puntos que conforman su trayectoria, también se tendrá en cuenta el instante de tiempo en el que se han tomado. Son útiles cuando se requiere tener en cuenta el tiempo, por ejemplo, para comparar el tiempo que le toma a dos atletas recorrer la misma pista.

Debido a ciertos factores, es común que ciertos movimientos no se realicen siempre a la misma velocidad. Por ejemplo, el deportista puede realizar el mismo movimiento a distinta velocidad en cada intento, y esto no hace que uno sea más correcto que otro. También cabe la posibilidad de que el vídeo se haya grabado en cámara lenta. Es por esto que tener en cuenta el aspecto temporal puede dar resultados que no se corresponden con los esperados, por lo que utilizaremos medidas de únicamente secuencia.

De entre todas las medidas de únicamente secuencia, una muy interesante es DTW (Dynamic Time Warping), por sus interesantes características a la hora de comparar trayectorias.

5.1.2. Dynamic Time Warping

DTW [10] es un algoritmo de únicamente secuencia que se caracteriza por encontrar la correspondencia entre puntos de las dos trayectorias que menor distancia acumulada proporcione. De forma intuitiva, esto provocará que cuando se comparen dos trayectorias que tienen forma similar, pero se encuentran desfasadas entre ellas, el algoritmo es capaz de encontrar la correspondencia de cada punto en la otra trayectoria.
En la Figura 5.1 se muestra un ejemplo de forma visual:

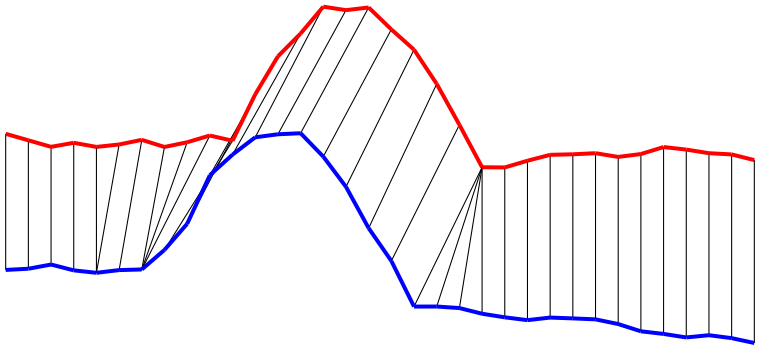

Figura 5.1: Emparejamiento de puntos del DTW.

Como se puede observar, el algoritmo es capaz de encontrar la correspondencia óptima entre los puntos de las dos trayectorias. Esto se hace muy notable en la elevación, que a pesar de que se encuentra desfasada, se ha encontrado una asociación acertada entre ambas trayectorias.

Cabe resaltar que a un punto de una trayectoria le puede corresponder más de un punto de la otra. Debido a esto, el algoritmo tiene algunas propiedades deseables para comparar secuencias de movimientos de un deporte:

- Ofrece buenos resultados tratando trayectorias con longitudes diferentes.
- Es resistente a diferentes tasas de muestreo. Esto es muy interesante, ya que permitirá comparar vídeos que tengan diferente tasa de fotogramas o con diferentes velocidades sin afectar al resultado.

Por otra parte, la principal desventaja de este algoritmo es su sensibilidad al ruido, esto es, los puntos ubicados incorrectamente tendrán un impacto significativo en el resultado, provocando que pueda ser demasiado impreciso. En este caso, las trayectorias ya han sido tratadas con anterioridad, por lo que el ruido se ha reducido de forma muy notable y no producirá resultados incorrectos.

5.1.3. Funcionamiento de DTW

Siendo $n = Size(T_1)$ y $m = Size(T_2)$, el algoritmo generará una matriz de $(n+1)$ x $(m+1)$ de izquierda a derecha y de arriba a abajo. Cada celda (i, j) de la matriz, donde $i, j \in \mathbb{N}, 2 \leq i \leq n+1$ y $2 \leq j \leq m+1$, se calcula como la distancia $d(P_{i-1}, Q_{j-1})$ más el

mínimo entre sus tres celdas vecinas, es decir, las celdas $(i-1, j)$, $(i, j-1)$ y $(i-1, j-1)$. El resultado final se encontrará en la celda $(n+1, m+1)$.

De forma recursiva se puede definir como:

$$dtw(T_1, T_2) = \begin{cases} 0 & \text{si n = 0 y m = 0} \\ \infty & \text{si n = 0 ó m = 0} \\ d(Head(T_1), Head(T_2)) + & \\ \quad min\{ & \\ \quad\quad dtw(T_1, Rest(T_2)), & \\ \quad\quad dtw(Rest(T_1), T_2), & \\ \quad\quad dtw(Rest(T_1), Rest(T_2))\} & \text{en otro caso} \end{cases}$$

Se puede observar que se hace uso de las funciones $d(p_1, p_2)$, "$Head(T)$" y "$Rest(T)$" definidas en el Capítulo 3, y de la función $min(n_1, n_2, n_3)$, que entrega como resultado el menor valor numérico de los dados.

5.2 Distancias entre ángulos

En la realización de movimientos en un deporte puede ser importante el ángulo que se forma en una articulación dada, ya que puede afectar a la calidad global de la técnica.

A lo largo de esta sección, se explica cómo se puede medir el ángulo de una articulación en cada fotograma, de forma que se pueda obtener una serie de ángulos para esta y posteriormente obtener una medida de distancia.

5.2.1. Cálculo de ángulos

Para poder calcular el ángulo de una articulación, es necesario disponer de dos puntos adyacentes a esta. Por ejemplo, se podrá calcular el ángulo que forma la rodilla derecha, ya que se conoce la posición del tobillo derecho y de la parte derecha de la cadera. Por otra parte, no se podrá conocer el ángulo que forma la muñeca, ya que, aunque se conoce la posición del codo, no se conoce la posición de la palma de la mano o de los dedos.

Conociendo los dos puntos adyacentes a la articulación, se puede modelar un triángulo, de forma que se puede calcular el ángulo mediante el uso de trigonometría.

Dado un triángulo de vértices A, B, C como el de la Figura 5.2 se puede obtener los lados a, b, c utilizando la distancia euclídea entre los vértices. Una vez obtenidos los lados, se puede obtener el ángulo de cualquiera de los vértices. En este caso, se mostrará cómo calcular el ángulo α para el vértice B, de forma que este vértice es la articulación deseada.

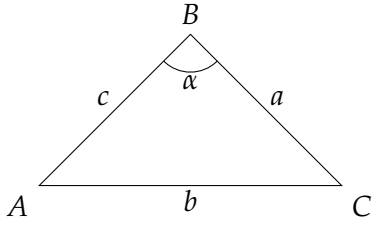

Figura 5.2: Triángulo de vértices A, B, C.

Para ello, primero debemos dividir el triángulo dado en dos triángulos rectángulos, de forma que se facilita la utilización de funciones trigonométricas. Como se va a calcular el ángulo α del vértice B, vamos a dividir el lado b mediante una recta perpendicular a este lado que pase por el vértice B. Con esto tendremos el lado b dividido en dos fragmentos, de los cuales llamaremos a uno x, quedando el restante como $b - x$. Además, así se divide el ángulo α en dos sub ángulos α_1 y α_2 tal que $\alpha = \alpha_1 + \alpha_2$. En la Figura 5.3 se muestra cómo se ha dividido el triángulo.

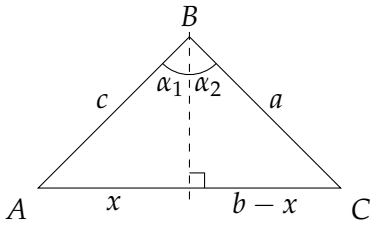

Figura 5.3: Triángulo dividido en dos triángulos rectángulos.

Con esta información, podemos obtener el valor del segmento x de la siguiente forma:

$$x = \frac{c^2 - a^2 + b^2}{2 \cdot b}$$

Una vez calculado x podemos calcular los dos sub ángulos α_1 y α_2 de la siguiente forma:

$$\alpha_1 = sin^{-1}\left(\frac{x}{c}\right)$$
$$\alpha_2 = sin^{-1}\left(\frac{b - x}{a}\right)$$

Finalmente se obtiene el ángulo deseado, como se ha indicado anteriormente, como:

$$\alpha = \alpha_1 + \alpha_2$$

5.2.2. Distancia entre ángulos

Una vez tenemos la capacidad de conocer los ángulos, se puede obtener una secuencia de ángulos que corresponde a los ángulos formados a lo largo del movimiento o en un fragmento de este. El objetivo es encontrar una medida de distancia entre secuencias de ángulos cuyo resultado sea un único número.

Sin embargo, la realidad es que los elementos a tener en cuenta en cada secuencia de ángulos pueden ser diferentes, por lo que no existe una forma global de obtener estas distancias. Por ejemplo, en un movimiento puede ser interesante que el ángulo promedio sea lo más parecido posible, mientras que en otro, el único aspecto importante es que no se alcance un ángulo dado. Por esta razón, se deberá implementar una manera diferente de calcular distancias para cada movimiento específico.

5.3 Interpretación de distancias

Una vez hemos obtenido una distancia, tanto de trayectorias como de ángulos, debemos determinar los valores que se considerarán válidos. Para ello, debemos conseguir

varias trayectorias correspondientes al mismo movimiento, pero de intentos y/o deportistas diferentes. Es importante que en todas ellas el movimiento sea realizado con una técnica correcta, de forma que no existan resultados erróneos causados por estos errores humanos.

El siguiente paso consiste en comparar todas las trayectorias o ángulos deseados entre ellos para obtener las distancias, de forma que obtendremos muchos valores que servirán de guía para obtener un rango válido. Concretamente, para k vídeos, la cantidad de comparaciones a realizar será $\sum_{i=1}^{k}(\sum_{j=i+1}^{k}(1)) = \frac{k^2-k}{2}$, por lo que no será necesario una gran cantidad de trayectorias para obtener muchos resultados.

Una vez tenemos una cantidad razonable de resultados válidos, debemos determinar un límite inferior y uno superior para obtener un intervalo de valores válidos. En un caso ideal, los datos obtenidos seguirán una distribución conocida, por lo que se podría realizar un análisis más exacto. Sin embargo, la realidad es que en cada caso seguirán una distribución impredecible, que incluso puede variar si se decide agregar más trayectorias al análisis. Aunque aumentando enormemente la cantidad de datos se podría determinar una distribución, no se puede asegurar que otro conjunto de trayectorias vaya a seguir la misma. Por esta razón, debemos obtener estos límites de una forma general que funcione para cualquier conjunto de datos.

Para encontrar un intervalo válido de forma general sobre un conjunto X, se puede tomar $[min(X), max(X)]$. El principal riesgo de esta estrategia es que un único dato anómalo excesivamente bajo o alto es capaz de cambiar de forma abrupta el intervalo. Para solucionar esto, podemos escoger un porcentaje de los datos con el que obtener el intervalo, descartando los que se encuentran cerca de los extremos. Esto no tendrá un impacto muy grande en el resultado si no existen datos anómalos, sin embargo sí que puede tener un gran impacto positivo si los hay, siempre y cuando el porcentaje de datos no anómalos sea igual o superior al que escojamos. Para obtener el intervalo con esta corrección debemos seguir los siguientes pasos:

1. Obtener una lista ordenada L de menor a mayor de los elementos del conjunto X.

2. Escoger un porcentaje $P \in [0, 1]$ de los elementos que queremos mantener. Debe ser un valor lo más grande posible pero menor o igual que el porcentaje de datos no anómalos.

3. Calcular los percentiles de los datos que se van a descartar. Para ello calculamos $D = (1 - P)/2$ y se definirán los percentiles inferior y superior como $P_{\text{lower}} = \lfloor D \cdot |X| \rfloor$ y $P_{\text{upper}} = \lfloor (1 - D) \cdot |X| \rfloor$ respectivamente.

4. El intervalo queda definido como $[L[P_{\text{lower}}], L[P_{\text{upper}}]]$.

Una vez obtenido el intervalo de confianza $[lower, upper]$, podemos obtener un porcentaje de validez para un valor dado. Para ello, debemos tener en cuenta los siguientes aspectos:

- Los valores contenidos en $[0, lower]$ tendrán una validez del 100 %. Esto se debe a que cuanto más se parecen dos conjuntos de puntos, menor es la distancia entre ellos. Si se da el caso de que una distancia es más pequeña que la mínima encontrada en esta fase, es porque los conjuntos de puntos se pueden considerar idénticos.

- Para un valor igual a $upper$, se considerará una validez del 50 %. De esta forma también se podrá medir la validez de valores mayores a $upper$.

Teniendo estos puntos en cuenta, debemos encontrar una función continua que devuelva un porcentaje único para cada valor. Hay que tener en cuenta que los valores pueden ser indefinidamente grandes, por lo que será conveniente utilizar una función que tenga una asíntota horizontal en 0^+, de forma que nunca obtengamos un resultado negativo. Para ello, resultará muy útil una función de la forma:

$$f(x) = \frac{1}{a * (x - b) + 1}$$

a la cual se le pasará como x la distancia sobre la que se quiere obtener el porcentaje de validez. Además se han utilizado dos parámetros a y b que sirven para ajustar la escala y desplazamiento horizontales respectivamente, de forma que se cumplan las dos condiciones mencionadas anteriormente. Los valores adecuados para este fin son:

$$a = \frac{1}{upper - lower}$$

$$b = lower \wedge b \neq upper$$

Como se puede observar, para que ambos parámetros funcionen, se debe cumplir que $lower \neq upper$.
Por tanto, la función completa para determinar el porcentaje de validez de una distancia queda de la siguiente forma:

$$validez(x) = \begin{cases} 1 & \text{si } 0 \leq x < lower \\ \frac{1}{a \cdot (x-b)+1} & \text{si } x \geq lower \end{cases}$$

En la Figura 5.4 se muestra un ejemplo de la función de forma visual, donde $lower = 50$ y $upper = 300$:

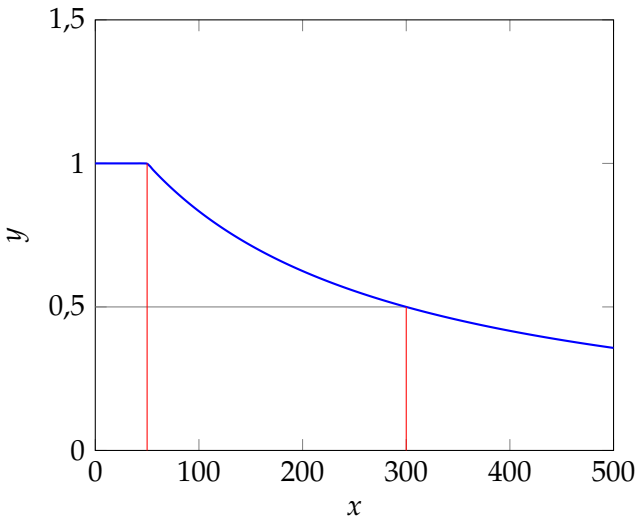

Figura 5.4: Función de obtención de porcentaje.

CAPÍTULO 6
Detección de movimientos específicos

La realización de un movimiento en un deporte, comúnmente se puede dividir en más submovimientos. Por ejemplo, de forma poco detallada, un saque de tenis se puede dividir en el movimiento de lanzar la pelota hacia arriba, y en el movimiento de golpear la pelota con la raqueta.

En este capítulo se va a plantear un método para detectar específicamente el inicio y el fin de cada submovimiento sin la necesidad de entrenar redes neuronales.

Para conseguir esto, podemos basarnos en la velocidad, en la posición relativa y en el ángulo de diferentes partes del cuerpo. Por ejemplo, en un saque de tenis, al momento de golpear la pelota, la mano derecha se estará moviendo más rápido que durante el resto del tiempo y además se debe encontrar por encima de la cabeza. También se realiza un pequeño salto, por lo que fijarse en el ángulo de las rodillas puede ser otro factor que determine la realización de este submovimiento.

6.1 Posición relativa

La posición relativa se refiere a la posición que tiene un punto del cuerpo respecto otro punto del mismo cuerpo.

Tiene una alta utilidad para filtrar partes en las que se tiene total seguridad de que se debe cumplir una posición relativa dada entre dos puntos, pudiendo reducir la probabilidad de que se detecte un submovimiento de forma errónea.

En la mayoría de casos solo será necesario comparar la posición en uno de los dos ejes, por lo que la comparación consistirá únicamente en realizar una resta de los componentes de los puntos:

- $p_A.x - p_B.x$: $\begin{cases} <0\text{: } p_A \text{ se encuentra a la izquierda de } p_B \\ =0\text{: } p_A \text{ y } p_B \text{ tienen el mismo desplazamiento horizontal} \\ >0\text{: } p_A \text{ se encuentra a la derecha de } p_B \end{cases}$
- $p_A.y - p_B.y$: $\begin{cases} <0\text{: } p_A \text{ se encuentra arriba de } p_B \\ =0\text{: } p_A \text{ y } p_B \text{ tienen el mismo desplazamiento vertical} \\ >0\text{: } p_A \text{ se encuentra debajo de } p_B \end{cases}$

  Esto se debe a que el punto de origen se encuentra arriba a la izquierda, por lo que cuanto mayor sea el componente vertical, más baja será la posición del punto.

Además de la comparación de posición entre dos puntos, también puede resultar útil la evolución de la posición de un mismo punto a lo largo del tiempo. Por ejemplo, si el deportista realiza un salto, con total seguridad sus pies se van a elevar.

6.2 Cambios de velocidad

Para detectar que una parte del cuerpo se encuentra en movimiento debido a que está realizando una acción, se asumirá que la velocidad de dicha parte durante la realización de la acción es mayor que cuando no está realizando ninguna. De esta forma, el problema se reduce a detectar en qué momento se mueve más rápido que durante el resto del movimiento. Para ello hay que determinar un umbral que indicará cuándo se mueve más rápido de lo habitual.

El primer paso consistirá en obtener una lista de la velocidad con la que se mueve un punto del cuerpo en el transcurso de cada fotograma al siguiente. Para ello, bastará con calcular la distancia recorrida en cada par de fotogramas, obteniendo así la velocidad en $\frac{u.longitud}{fotograma}$. Es decir, la lista de velocidades se obtiene como:

$$V = \{d(p_{i-1}, p_i) | i \in \mathbb{N}, 2 \leq i \leq Size(T)\}$$

Una vez disponemos de la velocidad que tiene un punto del cuerpo en todo momento, podemos obtener el umbral. Este debe ser lo suficientemente alto como para que no se detecte movimiento cuando el punto está en reposo, pero lo suficientemente bajo como para que detecte movimiento cuando sí lo hay.

Una primera aproximación para el umbral podría ser el promedio de las velocidades, de forma que las que se encuentran por debajo se consideran reposo, y las que se encuentran por arriba se consideran movimiento. El problema es que en casi cualquier caso, un punto se encontrará más tiempo en reposo que en movimiento, por lo que la media se encontrará demasiado cercana a velocidades bajas, haciendo que se esté considerando algunas velocidades bajas como movimiento. Por otra parte, esto hará que exista algo de dispersión, al estar las velocidades que se consideran movimiento ligeramente alejadas de la media. Podemos agregar una medida de dispersión, como la desviación típica, a la media para obtener un umbral algo mayor. De esta forma el umbral U se define como:

$$U = \overline{V} + \sigma_V$$

En la Figura 6.1 se muestra un ejemplo. Este consiste en un histograma en el que se han agrupado las velocidades en intervalos muy pequeños, de forma que sea casi equivalente a contar cuántas veces aparece cada velocidad:

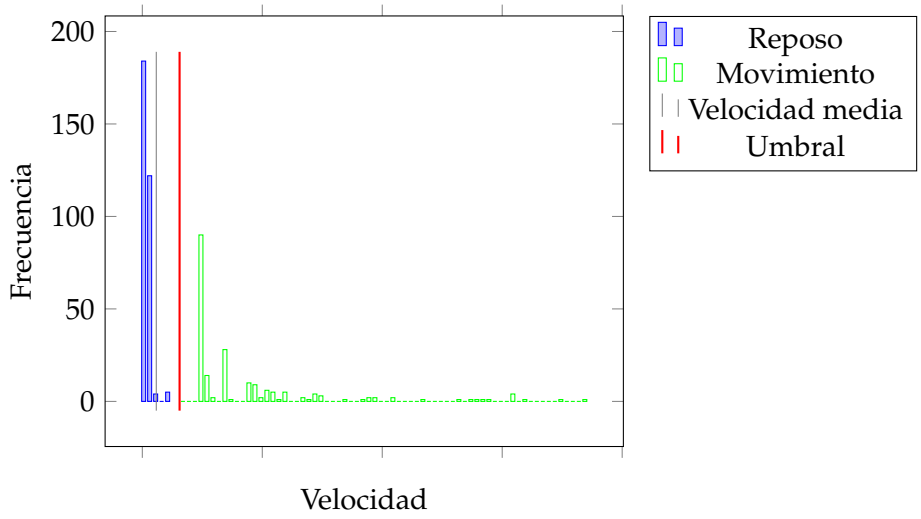

Figura 6.1: Cambios de velocidad.

Como se puede observar, existe una diferencia notable, tanto en cantidad como en valor, entre las velocidades de reposo y las velocidades de movimiento, existiendo un espacio entre ellas, en el que se sitúa el umbral.

6.3 Ángulos

Los ángulos de ciertas articulaciones del cuerpo también pueden ser de utilidad para determinar qué submovimiento se está realizando. Como ya conocemos la forma de obtener ángulos, se podrá utilizar para mejorar la precisión del análisis.

Para conseguirlo se puede enfocar de las siguientes formas:

- El ángulo alcanza o se mantiene con un valor o intervalo dado durante un periodo de tiempo. Esta opción requiere estudiar diversos ejemplares del movimiento para determinar cuál sería el valor más adecuado. Así mismo, existe el riesgo de que el valor o intervalo escogido no sea el más adecuado para todos los ejemplares, por lo que esta opción no es la más recomendable debido al tiempo de estudio y análisis que requiere y el riesgo que tiene de no ofrecer resultados precisos.
- El ángulo sufre una modificación medible a lo largo del tiempo. Esta opción se basa en medir como se transforma el ángulo a lo largo del tiempo para obtener indicios de que se está realizando un movimiento. Haciendo esto se reduce drásticamente el tiempo necesario para preparar el análisis y el riesgo de escoger un valor inadecuado debido a que toda la información se saca del mismo ejemplar. Por esta razón, será una opción interesante en la mayoría de los casos.

En muchos casos la medida de un ángulo servirá principalmente para aumentar la precisión del análisis y no como única fuente de datos. Por ejemplo, para detectar en qué momento se realiza un salto, se puede medir si la posición vertical de algunos puntos, como los pies, se desplaza hacia arriba. Esta puede ser una medición suficiente, pero puede ser mucho más fiable si también se comprueba que la rodilla se ha flexionado y posteriormente se ha extendido de nuevo. Esto se puede medir observando que existe un momento en el que el ángulo de la rodilla se reduce, y poco tiempo después se incrementa.

CAPÍTULO 7
Aplicación en el tenis

Una vez se ha presentado todas las herramientas de análisis, podemos aplicarlo a algún movimiento deportivo para comprobar los resultados que se pueden obtener con ellas. Concretamente se va a analizar el saque de tenis, presentando los pasos a seguir de forma ordenada. Para ello, se asume que ya se dispone de todas las posiciones de los puntos del cuerpo deseados.

Antes de realizar el análisis, cabe destacar que únicamente se tendrá en cuenta a deportistas que sostienen su raqueta con la mano derecha. Esto se debe a que es necesario definir qué movimiento realiza cada parte del cuerpo, y de esta forma el análisis puede englobar a muchos más deportistas que de forma contraria, al ser esta práctica más común.

7.1 Detección de movimientos

El primer paso es obtener los fotogramas de inicio y de fin de cada movimiento del saque. De esta forma, podremos obtener los fragmentos deseados de las trayectorias necesarias.

Para este movimiento nos centraremos en dos momentos clave:

- El deportista lanza la pelota hacia arriba
- El deportista golpea la pelota con la raqueta

A continuación se detalla qué características se pueden tener en cuenta para detectar los fotogramas en los que ocurren.

7.1.1. Lanzamiento de la pelota hacia arriba

Para empezar, para lanzar la pelota hacia arriba, el deportista mueve el brazo izquierdo hacia arriba manteniéndolo estirado. Con esta información, podemos encontrar el momento en el que ocurre esto observando que la velocidad de la mano izquierda se incrementa, a la vez que la mano supera la altura de la cabeza. Como detalle adicional, se puede observar que el ángulo del brazo izquierdo se mantendrá bastante cercano a 180 grados al estar estirado, pero no se tendrá en cuenta, puesto que esto puede ocurrir durante gran parte del saque, y puede llegar a empeorar la calidad del análisis.

Por otro lado, en esta parte del saque el deportista se prepara para realizar un salto, por lo que empieza a flexionar las dos rodillas. A efectos prácticos esto significa que el

ángulo de cada rodilla irá disminuyendo a lo largo de la realización de este movimiento. Se debe tener en cuenta que el momento en el que el deportista flexiona las rodillas puede variar según el intento o la persona, por lo que el resultado no debe depender en gran medida de esta observación, y únicamente debería ser utilizada para reforzar la hipótesis de que lo que se ha encontrado es este movimiento.

En la Figura 7.1 se muestra dos fotogramas clave en los que se puede observar la evolución del movimiento.

(a) Primer fotograma del movimiento.

(b) Último fotograma del movimiento.

Figura 7.1: Movimiento al lanzar la pelota hacia arriba.

7.1.2. Golpeo de la pelota con la raqueta

Para golpear la pelota con la raqueta, lo primero que se puede observar es que la mano derecha, que sostiene la raqueta, se debe mover. Además, en el momento de golpear la pelota, la mano se encontrará por encima de la cabeza. De nuevo, esto se puede detectar determinando los momentos en los que la mano derecha tiene un incremento de velocidad, y alcanza una altura superior a la cabeza.

Adicionalmente al movimiento de la mano derecha, el deportista realiza un pequeño salto, por lo que obligatoriamente los pies se elevarán levemente del suelo. Este es un detalle determinante, ya que a lo largo del saque de tenis no hay ningún otro salto, así que será útil para seleccionar de forma precisa el fotograma que marque el inicio de este movimiento. Además, esto se puede reforzar comprobando que el ángulo de la rodilla crece.

En la Figura 7.2 se muestra dos fotogramas clave en los que se puede observar la evolución del movimiento.

(a) Primer fotograma del movimiento.

(b) Último fotograma del movimiento.

Figura 7.2: Movimiento al golpear la pelota con la raqueta.

Además de las técnicas mencionadas, que ayudan a ubicar un movimiento específico a lo largo del vídeo globalmente, se puede hacer uso de dependencias temporales entre

fragmentos. Por ejemplo, en este caso se puede definir que el fragmento del vídeo en el que se lanza la pelota hacia arriba termina antes de que el fragmento en el que se golpea la pelota con la raqueta empiece. Gracias a esto será posible obtener una mayor confianza en el análisis al tener certeza de que los intervalos de tiempo obtenidos al menos tienen un orden correcto, y no se superponen entre ellos. Además, puede ser útil para apoyarse en fragmentos que son más fáciles de detectar que otros.

7.2 Aspectos del saque

En el caso de un saque de tenis, existen muchos detalles que pueden ser determinantes en la calidad de la técnica. En este caso el análisis se centra en los siguientes:

- La trayectoria que sigue la mano izquierda al lanzar la pelota hacia arriba.
- La trayectoria que sigue la mano derecha al golpear la pelota con la raqueta.
- El ángulo de las rodillas a lo largo del saque.

Para realizar un análisis más detallado se podría incluir más aspectos a tener en cuenta, pero en este caso nos vamos a centrar en estos tres para obtener una primera aproximación de los resultados que se pueden obtener.

7.3 Distancia entre ángulos

Antes de poder seguir, es necesario disponer de una implementación específica de distancia entre ángulos. Anteriormente, en la Subsección 5.2.2, se menciona la necesidad de tener una implementación específica para cada movimiento, o incluso cada punto del cuerpo en el mismo movimiento.

Este es el momento adecuado para definir dichas implementaciones para el saque de tenis, ya que hemos decidido qué aspectos tendremos en cuenta, y los próximos pasos requieren que todas las medidas de distancia ya estén definidas.

En este caso el único ángulo que se va a medir es el de las rodillas, por lo que únicamente será necesario definir una medida de distancia.

En este punto, tenemos dos secuencias de ángulos A_1 y A_2 que van a ser comparadas. Para ello, podemos considerar como características clave a los valores mínimo y máximo que se alcanzan. Además, también se puede hacer uso del valor medio de todos los valores, como indicador de la cantidad de tiempo que el ángulo es un valor grande o un valor pequeño. Hay que tener en cuenta que el valor medio de todos los ángulos de la secuencia es diferente al valor medio entre el mínimo y el máximo, y es por eso que se puede utilizar con este fin.

Para incluir toda esta información en un mismo valor d_{ang}, se puede calcular la diferencia en valor absoluto d_{angmin} entre los valores mínimos y d_{angmax} entre los valores máximos de A_1 y A_2. En cuanto al promedio, se puede normalizar sobre 180 grados, que es el valor máximo que debería alcanzar una rodilla, y de esta forma también se puede calcular la diferencia en valor absoluto d_{angavg}.

Teniendo esto en cuenta, las distancias individuales se pueden obtener como:

$$d_{angmin} = |min(A_1) - min(A_2)|$$

$$d_{angmax} = |max(A_1) - max(A_2)|$$

$$d_{angavg} = \left| \frac{avg(A_1)}{max(A_1)} \cdot 180 - \frac{avg(A_2)}{max(A_2)} \cdot 180 \right|$$

donde *min* encuentra el valor mínimo, *max* el valor máximo, y *avg* el promedio de todos los valores de la secuencia dada.

Por tanto una primera aproximación a la distancia total d_{ang} podría ser:

$$d_{ang} = d_{angmin} + d_{angmax} + d_{angavg}$$

Pero hay que tener en cuenta que es posible que no todas las distancias individuales tengan la misma importancia. Por ejemplo, en este caso podemos considerar que las distancias d_{angmin} y d_{angmax} son igual de importantes, y que la distancia d_{angavg} tiene poca importancia. La solución consiste en aplicar pesos de acuerdo a estas decisiones.

De esta forma, la distancia total entre secuencias de ángulos puede quedar de la siguiente forma:

$$d_{ang} = 0{,}45 \cdot d_{angmin} + 0{,}45 \cdot d_{angmax} + 0{,}10 \cdot d_{angavg}$$

En conclusión, esta forma de calcular la distancia puede ser versátil al contener toda la información importante en una sola cifra. Por otra parte, su única utilidad es determinar cuánto se parecen dos secuencias de ángulos, y no da ningún tipo de información sobre qué factor ha producido ese resultado, aunque como una primera aproximación puede ser suficiente.

7.4 Intervalos válidos

El siguiente paso consiste en obtener el intervalo válido para cada punto del cuerpo durante un tiempo dado.

Para poder obtener los intervalos de validez, necesitamos realizar las comparaciones de los aspectos deseados con un conjunto de vídeos de deportistas realizando el saque con una técnica que se considera ideal. Para obtener el máximo de resultados útiles y sin repetir, se debe comparar todos los vídeos entre ellos, evitando comparar un vídeo con él mismo, y evitando comparar cada par de vídeos más de una vez. Esto debe hacerse así, puesto que todas las medidas de distancia utilizadas son métricas [14], es decir, cumplen que $d(x, y) = 0$ si y solo si $x = y$, y que $d(x, y) = d(y, x)$.

Teniendo esto claro, debemos obtener un intervalo para cada uno de los aspectos mencionados en la Sección 7.2. Para ello, para cada par de vídeos, debemos calcular las distancias correspondientes y guardar el resultado en una lista separada para cada aspecto.

Una vez realizadas todas las comparaciones, debemos ordenar las listas y obtener los valores mínimo y máximo del intervalo como se indica en la Sección 5.3 a partir de cada lista.

Estos intervalos podrán almacenarse, ya que únicamente cambiarán si se cambia algún aspecto a cerca de las distancias o los intervalos de tiempo, o en el caso de que se añada, se retire o se modifique algún ejemplar de saque correcto.

7.5 Análisis de una muestra

En este punto ya se dispone de todas las herramientas y datos necesarios para poder realizar el análisis a una muestra para obtener información sobre la calidad de la técnica.

Para obtener los porcentajes de similitud de una muestra, debemos compararla con algún ejemplar correcto de los que disponemos. Sin embargo, debido a que no es posible saber cuál es el ejemplar más adecuado para cada caso, debemos comparar la muestra con todos los ejemplares.

Específicamente, se comparará la muestra con todos los ejemplares en cada uno de los aspectos deseados, y elegiremos los resultados que menor distancia tengan para cada aspecto por separado. Haciendo esto, se asume que la combinación de diferentes técnicas correctas en cada aspecto, forma una técnica global del movimiento que también es correcta.

Una vez se dispone de todas las distancias mínimas, únicamente resta obtener el porcentaje de similitud usando los intervalos válidos calculados anteriormente. Para ello se utilizará una función como la que se describe en la Sección 5.3.

Este proceso puede llegar a ser muy costoso si los vídeos son demasiado largos, o si hay demasiados vídeos. Es por esto que se debe intentar que los vídeos tengan el contenido mínimo indispensable, y que se escoja una cantidad no muy grande de estos, priorizando que la calidad de la técnica mostrada sea la mejor posible.

CAPÍTULO 8
Implementación

Tras haber planteado todo el funcionamiento del análisis, es momento de implementarlo para comprobar los resultados que es capaz de ofrecer.

Para hacerlo, se ha escogido C++ como lenguaje de la aplicación. De esta forma se podrá aprovechar la alta velocidad de ejecución de los lenguajes compilados, junto con las optimizaciones adicionales que realiza el compilador Microsoft Visual C++ (MSVC) [15].

A lo largo de este capítulo se mostrará cómo se ha implementado el análisis, con el objetivo de que se pueda modificar y ampliar de forma sencilla. También cabe resaltar que se trata de una aplicación de terminal, puesto que lo importante es el resultado que ofrece, y el usuario no va a tener que interactuar continuamente con ella.

8.1 Diseño

Ya que C++ es un lenguaje orientado a objetos, se puede aprovechar para separar las responsabilidades de diferentes cálculos en diferentes entidades. Esto es muy conveniente, ya que la información que se obtiene directamente de OpenPose para cada fotograma del vídeo, consiste en un array de 75 números de coma flotante de la siguiente forma:

$$\{kp0_x, kp0_y, kp0_r, kp1_x, kp1_y, kp1_r, ..., kp74x, kp74_y, kp74_r\}$$

donde kpi_x es el componente horizontal del punto clave i, kpi_y es el componente vertical del punto clave i, y kpi_r es la confianza del punto clave i. De esta manera, si se tuviera toda la lógica de extracción de información del array y la lógica del análisis en el mismo sitio, desembocaría en una cantidad considerable de código poco legible y poco mantenible.

En la Figura 8.1 se muestra el diagrama de clases de UML de la aplicación. Este contiene algunos detalles que dejan ver las funciones principales de cada clase y cómo se ha distribuido la responsabilidad. Se puede observar cómo se ha optado por utilizar la herencia para implementar los análisis de cada movimiento de forma separada, aprovechando la información que las superclases pueden ofrecer de forma general. De esta forma la aplicación es fácilmente ampliable a más movimientos, aunque en este caso únicamente se ha implementado el análisis del saque de tenis.

A continuación se explica la utilidad de cada una de las clases.

8.1.1. MoveComparator

La única finalidad de esta clase es garantizar la existencia de un método abstracto analyze(), que contendrá las llamadas necesarias para realizar el análisis. Es decir, es el

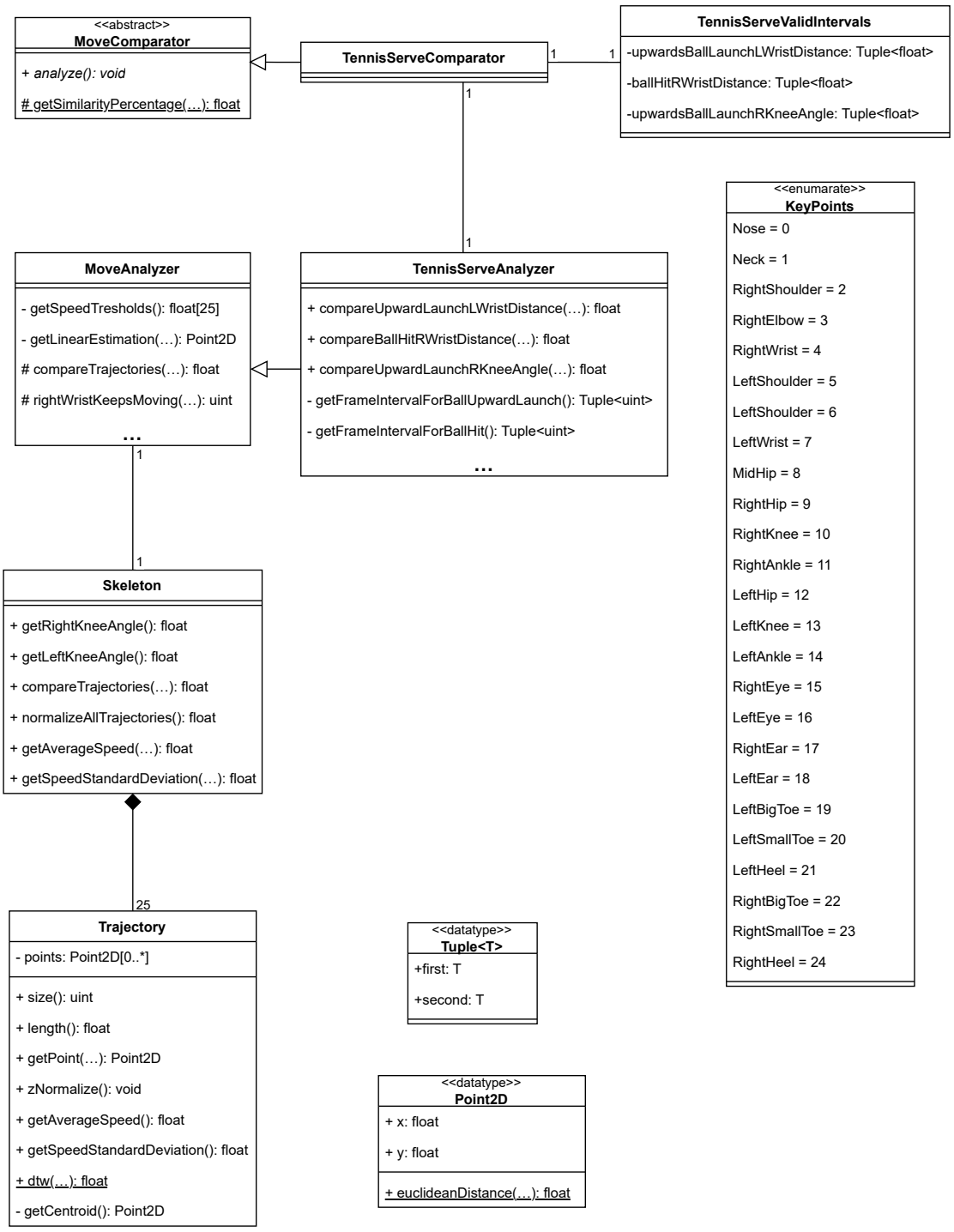

Figura 8.1: Diagrama de clases.

punto de comienzo del análisis. También ofrece la función para obtener el porcentaje de similitud a partir de un valor y un intervalo de validez dados. Al ser una clase abstracta, obligatoriamente se deberá crear subclases específicas que implementen la lógica necesaria.

8.1.2. TennisServeComparator

Esta es la subclase de MoveComparator encargada de iniciar el análisis de un saque de tenis. Su implementación de analyze() hace exactamente lo que se ha descrito en la Sección 7.5, utilizando su instancia de TennisServeAnalyzer.

8.1.3. MoveAnalyzer

Esta clase es la única que es capaz de trabajar con la información que llega directamente de OpenPose. De esta forma puede construir en su inicio las abstracciones necesarias para realizar los cálculos posteriores de forma mucho más ordenada. De base, realiza la corrección del array que recibe de OpenPose mediante interpolación lineal como se explica en la Sección 4.1, y también calcula los umbrales de velocidad para determinar cuándo un punto clave se mueve más rápido de lo normal como se describe en la Sección 6.2. Además, ofrece distintos métodos para conocer si un punto clave del cuerpo se encuentra por encima de otro, si se desplaza hacia alguna dirección, etc. Es decir, es capaz de ofrecer toda la información genérica necesaria que será útil para los análisis de movimientos específicos.

8.1.4. TennisServeAnalyzer

Esta es la subclase de MoveAnalyzer encargada de implementar las herramientas de análisis específicas para el saque de tenis. Concretamente, se encarga de obtener los intervalos temporales de cada submovimiento del saque. Esto lo hace utilizando los métodos que ofrece su superclase. Además, ofrece directamente los métodos que comparan cada aspecto deseado del saque. Existirá una instancia de esta clase para cada vídeo.

8.1.5. Skeleton

La finalidad de la clase Skeleton es actuar como contenedor de trayectorias. Esto es útil para ofrecer información general de las trayectorias a los niveles superiores, o para realizar cálculos sobre la trayectoria correspondiente a un punto del cuerpo dado. Por ejemplo, es la encargada de calcular los ángulos de las rodillas en el saque de tenis.

8.1.6. Trajectory

Por último, la clase Trajectory es un envoltorio para una secuencia de puntos, es decir, una trayectoria. Como se menciona en el Capítulo 3, una trayectoria tiene algunas propiedades y existe mucha información que se puede calcular a partir de ella, como el tamaño, la longitud o el centroide entre otras. Además también ofrece métodos para realizarse la unidad tipificada o “z-score normalization”, o entregar un fragmento de la trayectoria para poder analizar únicamente el fragmento deseado. Por último, también tiene una función para calcular la distancia DTW entre dos trayectorias.

8.1.7. Estructuras de apoyo

Además se ha creado algunas estructuras adicionales que permiten facilitar el almacenamiento y tratamiento de la información más básica. Estas son:

- Point2D: Consiste en una estructura que representa puntos de dos dimensiones y permite encapsular los dos componentes en un mismo lugar, lo que facilitará el almacenamiento y tratamiento de estos. Además, ofrece una función que calcula la distancia euclídea entre dos instancias de Point2D, lo cual será de gran utilidad en varias partes del análisis.

- TennisServeValidIntervals: Es una estructura cuya única finalidad es almacenar los intervalos de valores válidos para cada aspecto del movimiento en memoria, a la vez que facilita los nombres del intervalo. Existen otras formas de almacenar esta información con estructuras genéricas como un diccionario, pero se ha optado por esta solución para evitar errores al escribir los nombres y por el tiempo inmediato de acceso a los datos.

- Tuple<T>: En el diagrama de clases se ha representado como Tuple a cualquier estructura de datos que contiene dos elementos, como `std::pair<T1, T2>` de C++.

8.2 Pruebas

Para probar la corrección de los algoritmos y los resultados obtenidos se ha hecho uso de pruebas unitarias. Estas son automáticas, de forma que se agiliza mucho el hecho de cambiar la implementación de un algoritmo o cálculo, al comprobar que los resultados siguen siendo correctos tras el cambio de forma muy rápida.

Principalmente las pruebas se han centrado en realizar comprobaciones sobre el resultado obtenido de los miembros públicos de las clases. Además, se ha intentado provocar la ruptura de la aplicación con conjuntos de datos con características que puedan comprometer el funcionamiento de las funciones y evitar por completo que la ejecución finalice. Las características principales de estos conjuntos de datos han sido:

- Simular vídeos sin fotogramas. El principal fin de esto ha sido comprobar el comportamiento de ciertas funciones que necesitan series de datos para funcionar, y la protección que ofrecen contra índices fuera de los límites.

- Simular vídeos con más de un fotograma, pero eliminando información necesaria para realizar algunos cálculos. El objetivo de estos casos ha sido comprobar que la ausencia de esa información no implica la parada abrupta de la aplicación, sino que simplemente da un resultado sin significado como un NaN.

El resultado de la ejecución de estas pruebas ha sido satisfactorio, ya que ha servido para detectar varios casos de índice fuera de los límites, y algunas divisiones entre cero.

Como se puede observar en la Figura 8.2, se ha corregido estos problemas que podían afectar a la aplicación, reduciendo los casos que pueden provocar su ruptura.

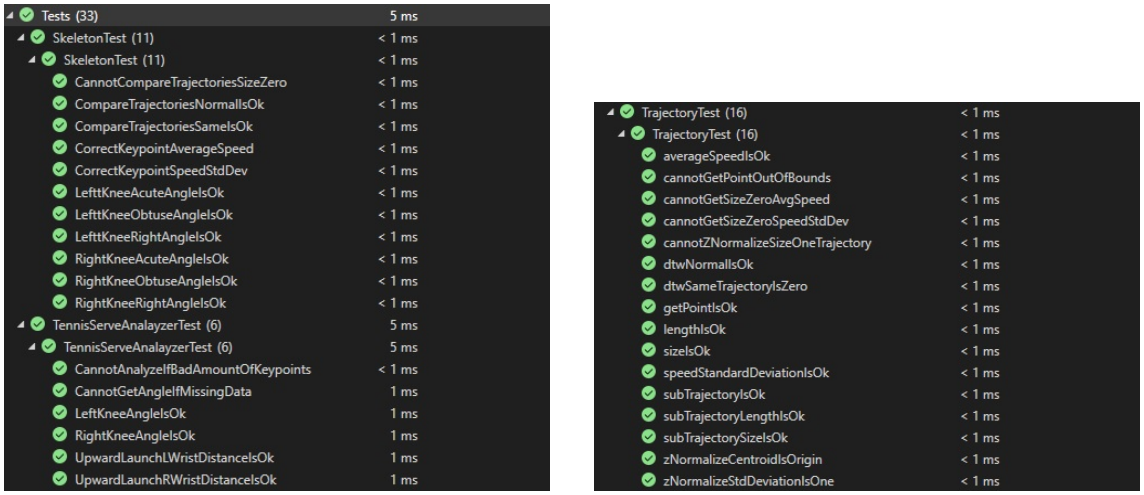

Figura 8.2: Ejecución de las pruebas unitarias.

8.3 Ejecución

Una vez implementada la aplicación, se puede ejecutar con un vídeo real diferente a cualquiera de los utilizados como referencia para observar qué resultados proporciona. Para ello utilizaremos un vídeo adicional que se pasará como entrada a la aplicación.

Cuando ejecutamos la aplicación, lo primero que se muestra es una ventana que muestra el análisis postural que realiza OpenPose en tiempo real. En ella se puede ver de forma visual cómo la librería identifica cada parte del cuerpo. Aunque no se puede interactuar con la aplicación de ninguna forma, es importante que haya supervisión humana, puesto que en algunas ocasiones, puede haber más de una persona, y que la librería decida analizar a alguien que no es el deportista a analizar, en cuyo caso se deberá editar el vídeo para hacer desaparecer a la persona no deseada, sin comprometer la visibilidad de la figura principal.

En la Figura 8.3 se muestra un fotograma de lo que se puede ver durante este proceso.

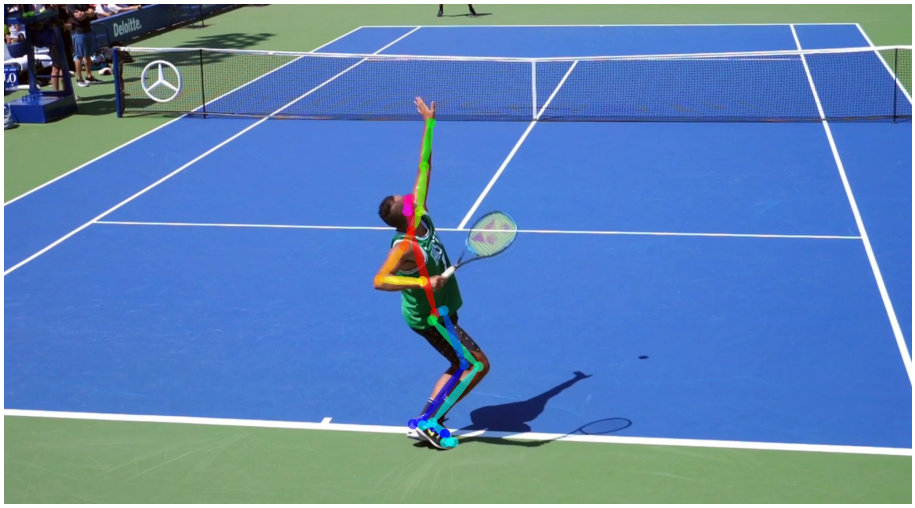

Figura 8.3: Análisis de Open Pose.

Una vez terminado el análisis de la postura, se realiza el análisis implementado para saques de tenis y seguidamente se muestra el resultado obtenido en la terminal. En este

se indicará los porcentajes de similitud en cada uno de los aspectos analizados por separado, y el porcentaje promedio de estos. Este promedio es la media aritmética entre los aspectos, pero se podría ponderar según la importancia de cada uno.

En la Figura 8.4 se puede observa el resultado obtenido a partir de esta muestra.

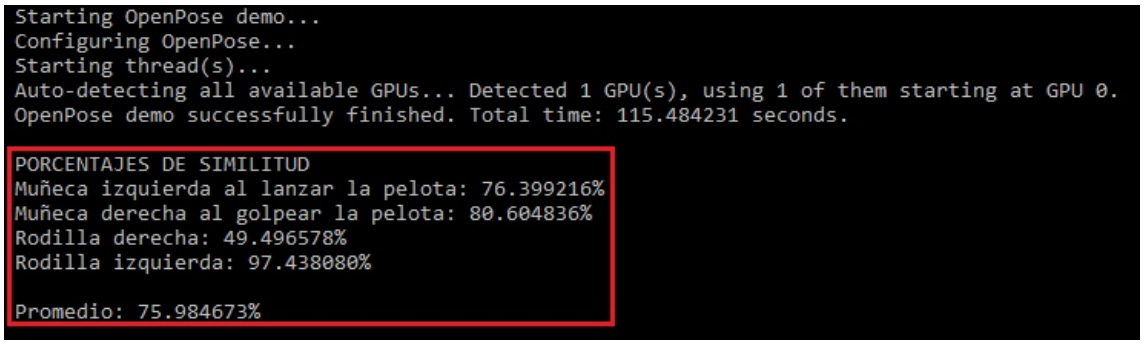

Figura 8.4: Resultado de la ejecución.

De esta forma, se obtiene una puntuación por cada uno de los aspectos analizados y la puntuación global del saque siguiendo los criterios establecidos en el análisis.

Además, cabe resaltar que, debido a la forma de comparar una muestra con los vídeos base, si utilizamos como muestra uno de los vídeos base, el porcentaje de similitud en todos los aspectos será del 100 %, como se puede apreciar en la Figura 8.5. Esto es consistente con la decisión tomada al inicio de que los vídeos base son considerados ideales.

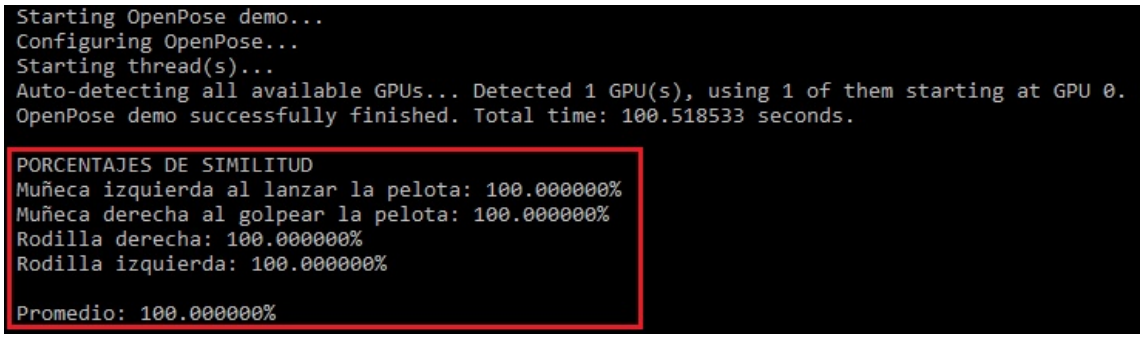

Figura 8.5: Resultado de un vídeo ideal.

8.4 Uso de recursos

Debido al elevado tiempo de procesamiento que toma Open Pose para obtener las coordenadas de los puntos, es conveniente realizar una medición general de recursos del equipo, de forma que se pueda conocer dónde se encuentra el principal cuello de botella y poder reemplazar ese hardware si se requiere menor tiempo de procesamiento.

Para ello, se va a realizar una medición superficial de utilización de recursos con el equipo que se muestra en la Tabla 8.1 y con la aplicación compilada con el modo "Release" del MSVC, mejorando así la velocidad con las optimizaciones que aplica el compilador.

Componente	Modelo
Procesador	AMD Ryzen 7 5800X 8-Core @4.4GHz
Memoria RAM	32 GB DDR4 3600 MT/s
Tarjeta gráfica	NVIDIA GeForce GTX 1060 6 GB GDDR5
Sistema operativo	Windows 10 (64 bits)

Tabla 8.1: Especificaciones del equipo de pruebas.

Las pruebas se han enfocado en la parte en la que Open Pose analiza el vídeo, ya que es la que más recursos consume.

Con el equipo de pruebas, la librería es capaz de procesar una media de 5,1 fotogramas por segundo, necesitando aproximadamente seis veces la duración del vídeo para extraer las coordenadas, siempre y cuando el vídeo tenga una tasa de 30 fotogramas por segundo.

Se ha medido la utilización de recursos en el periodo de tiempo en el que se ejecuta la librería, analizando vídeos con una resolución de 1920x1080 píxeles, con la ayuda del perfilador de Visual Studio [16] y con MSI Afterburner [17] para medir también el uso de memoria gráfica. Los resultados obtenidos son los siguientes:

- Procesador: Ha tenido un uso continuado del 6,2 % del total de su capacidad. Este porcentaje es el resultado de la carga que se divide entre cuatro núcleos del procesador que selecciona el sistema operativo.
- Memoria RAM: Ha tenido un uso continuado de 6,6 GB, siendo también el máximo valor alcanzado.
- Tarjeta gráfica: Ha tenido un uso continuado del 80 % de su capacidad aproximadamente.
- Memoria gráfica: Ha tenido un uso continuado de 5,8 GB. Este valor se encuentra muy cerca del límite de memoria de vídeo del que se dispone.

Tras la recolección de resultados, se puede observar que se hace un uso muy intensivo de la tarjeta gráfica, tanto en procesamiento como en memoria. Es por esto que aparentemente, reemplazando este hardware por otro más potente se puede obtener una mejora de rendimiento. Sin embargo, no se puede concluir si el cuello de botella es provocado por el procesador gráfico o por la memoria de vídeo.

Para medir esto, se ha analizado un vídeo de una resolución de 720x480 píxeles, una resolución muy inferior a la utilizada anteriormente. Los resultados han sido un 80 % de uso para el procesador gráfico y un uso continuado de 5,6 GB para la memoria de vídeo, obteniendo un tiempo de procesamiento similar al anterior.

Con esta información, se puede concluir que el limitante no es la capacidad de la memoria de vídeo, sino la potencia del procesador de vídeo. Sin embargo, no se está utilizando toda su capacidad. Con alta probabilidad, esto ocurre debido al pequeño periodo de tiempo que transcurre desde que la tarjeta gráfica termina de analizar un fotograma hasta que el procesador le entrega el siguiente.

Para determinar la mejora que puede llegar a suponer reemplazar la tarjeta gráfica por una más potente, se ha estimado con el perfilador que aproximadamente un 79,1 % del tiempo de ejecución a la hora de analizar un video recae sobre este componente, lo cual encaja mucho con el porcentaje de uso obtenido anteriormente. Con la ley de Amdahl [18] se puede obtener la mejora de rendimiento total que se obtendrá al reemplazar este

hardware por uno más rápido. A continuación se muestra algunos resultados obtenidos mediante esta ley:

Mejora GPU	Aceleración total (veces más rápido)
x2	x1.65
x3	x2.12
x5	x2.72
x10	x3.47
x15	x3.82
x20	x4.03

Tabla 8.2: Aceleración total de la aplicación.

La máxima aceleración obtenible tiende a una tasa de 4.79 veces más rápido. Por tanto, se puede concluir que para obtener una ganancia de velocidad sustancial al utilizar Open Pose, la mejor opción es reemplazar la tarjeta gráfica, priorizando la velocidad de su procesador, y cuidando que no disponga de menos de 6 GB de memoria VRAM.

CAPÍTULO 9
Conclusiones

Una vez finalizados los procesos de investigación e implementación, se puede obtener unas conclusiones finales acerca de los resultados y objetivos alcanzados.

En líneas generales, se ha logrado solventar los problemas básicos que existen cuando se obtienen coordenadas de puntos mediante tracking óptico utilizando vídeos desde un único punto de vista, sin ninguna referencia adicional. La calidad de la información bruta obtenida de OpenPose, sumada al tratamiento de los datos para evitar estos problemas, consigue que la información para poder realizar análisis sea suficientemente precisa.

Respecto a la detección de movimiento, debido a la decisión inicial de no utilizar redes neuronales, se ha tenido que dedicar un esfuerzo mayor en la búsqueda de elementos identificativos de cada movimiento, y en su programación para la detección automática. Además, es muy probable que la precisión alcanzada con este método sea inferior. Por otra parte, se ha reducido considerablemente el esfuerzo en conseguir grandes conjuntos de datos de entrenamiento.

En relación con los análisis realizados, es decir, análisis de trayectorias y de ángulos de articulaciones, se trata de dos elementos básicos a tener en cuenta en una gran cantidad de deportes y movimientos, aunque puede haber más elementos a tener en cuenta como velocidades y aceleraciones. Sin embargo, en la implementación de comparación de ángulos, solo se obtiene una medida de parecido entre secuencias de ángulos, pero no se conoce la razón que marca las diferencias entre ellas, lo cual podría llegar a ser muy conveniente.

Por último, en cuanto a la implementación, se ha priorizado la posibilidad de ampliar el código de la forma más sencilla posible, con el fin de poder realizar análisis de más movimientos, o ampliar los elementos de análisis sin que exista conflicto con las técnicas ya existentes.

En términos generales, se trata de un proyecto que ofrece las herramientas básicas para realizar un análisis deportivo que se puede pulir, mejorar y ampliar con el tiempo. Su principal fin ha sido hacer una demostración de cómo se puede utilizar el tracking óptico con este fin, y mostrar el grado de precisión alcanzable mediante esta técnica.

9.1 Trabajo futuro

Más allá de la ampliación a más deportes y movimientos, hay varios elementos existentes que se pueden mejorar, y otros que se pueden incorporar para mejorar la calidad del análisis.

Los más destacables son:

- Mejorar la estimación de puntos incorrectos. Como ya se ha visto, actualmente se realiza una interpolación de puntos lineal. Se podría mejorar mediante interpolaciones de curvas más suaves, como la interpolación cúbica.
- Incluir más elementos de análisis. Un elemento adicional a analizar podría ser la velocidad y aceleración de un punto clave dado. Pero como se menciona anteriormente, no se puede comparar velocidades entre vídeos diferentes, ya que estos han podido ser alterados en cuanto a velocidad de reproducción, por lo que se requeriría un estudio que permita obtener una medida de velocidad obtenida con la información del mismo vídeo.
- Incluir más detalle en el análisis de ángulos. Como se ha mencionado en las conclusiones, puede ser de gran ayuda conocer la razón por la que una comparación de ángulos ha tenido ese resultado. Por ejemplo, que la distancia entre secuencias de ángulos sea grande puede significar que era necesario que se alcanzara un ángulo mucho menor, y esta información puede ser muy útil.

En cuanto al análisis específico del tenis, aunque la obtención de intervalos de movimientos ha dado resultados bastante adecuados, se puede llegar a mejorar más incluyendo más parámetros o ajustando los ya existentes. Además, se podría incluir el análisis de más puntos clave si así se deseara.

Bibliografía

[1] Rolland, Jannick and Baillot, Yohan and Goon, Alexei. A survey of tracking technology for virtual environments, emitido en enero de 2001. Consultado en https://www.researchgate.net/publication/242415577_A_survey_of_tracking_technology_for_virtual_environments en marzo de 2023.

[2] William R. Sherman, Alan B. Craig. *Understanding Virtual Reality*. Elsevier Science, 2003. Páginas 81–82.

[3] YOLO. Consultado en https://docs.ultralytics.com/ en agosto de 2023.

[4] OpenPose. Consultado en https://github.com/CMU-Perceptual-Computing-Lab/openpose en agosto de 2023.

[5] Ronny Votel, Na Li. Next-Generation Pose Detection with MoveNet and TensorFlow.js. *Google Research*, mayo, 2021. Consultado en https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html en septiembre 2023.

[6] Football Players Tracking with YOLOv5 + ByteTrack. Consultado en https://github.com/SkalskiP/sport en agosto de 2023.

[7] James Purtill. VAR offside technology at 2022 World Cup, and how player data is changing professional sport. *ABC*, diciembre, 2022. Consultado en https://www.abc.net.au/news/science/2022-12-16/world-cup-2022-qatar-var-offside-technology/101762934?utm_campaign=abc_news_web&utm_content=link&utm_medium=content_shared&utm_source=abc_news_web en agosto 2023.

[8] Sports counting by pose estimation. Consultado en sports_counting_by_pose_estimation en septiembre de 2023.

[9] IncludeHealth. Consultado en https://www.includehealth.com/ en septiembre de 2023.

[10] Su, H., Liu, S., Zheng, B. et al. A survey of trajectory distance measures and performance evaluation, emitido en octubre de 2019. Consultado en https://zheng-kai.com/paper/vldbj_2019.pdf en marzo de 2023.

[11] Weisstein, Eric W. Geometric Centroid. *MathWorld–A Wolfram Web Resource*. Consultado en https://mathworld.wolfram.com/GeometricCentroid.html en marzo de 2023.

[12] Interpolación segmentaria. Consultado en http://interpolacion.wikidot.com/spline-teoria en abril de 2023.

[13] Zach. Z-Score Normalization: Definition & Examples. Consultado en https://www.statology.org/z-score-normalization/ en abril de 2023.

[14] Pedro José Herrero Piñeyro. Topología de Espacios Métricos. Consultado en https://www.um.es/documents/4874468/11035667/cap-1-esp-met.pdf/6c66d56e-c01d-4270-a56c-50ec184a830e en julio de 2023.

[15] Referencia de compilación de C/C++. Consultado en https://learn.microsoft.com/es-es/cpp/build/reference/c-cpp-building-reference?view=msvc-170 en agosto de 2023.

[16] Primer vistazo a las herramientas de generación de perfiles. Consultado en https://learn.microsoft.com/es-es/visualstudio/profiling/profiling-feature-tour?view=vs-2022 en agosto de 2023.

[17] MSI Afterburner. Consultado en https://www.msi.com/Landing/afterburner/graphics-cards en agosto de 2023.

[18] Yuan Shi. Reevaluating Amdahl's Law and Gustafson's Law, emitido en febrero de 2006. Consultado en https://web.archive.org/web/20060205041012/http://joda.cis.temple.edu/~shi/docs/amdahl/amdahl.html en agosto de 2023.

APÉNDICE A
Configuración del sistema

La siguiente guía explica cómo se debe ejecutar la aplicación en Windows.

El resultado de la compilación es un ejecutable en formato .exe que se debe ejecutar de forma adecuada para indicar la ruta del vídeo que se desea analizar. Para hacerlo la aplicación espera un parámetro "video" seguido de la ruta al vídeo. Además, por la configuración de OpenPose, únicamente es posible ejecutar la aplicación desde la raíz del proyecto, de forma que no existan errores de dependencias.

A.1 Fase de inicialización

Teniendo todo lo mencionado anteriormente en cuenta, para ejecutar la aplicación de forma correcta, se debe abrir una terminal, como cmd o PowerShell, en la raíz del proyecto y ejecutar la siguiente línea:

```
.\build\x64\Release\UserCustomCode.exe video "ruta al video"
```

Se debe reemplazar "ruta al video" por la ruta completa al archivo de vídeo deseado.

Debido a que se requiere situarse en la raíz del proyecto para poder ejecutar la aplicación, puede no ser muy práctico ejecutarla. Para ello, se puede elaborar un archivo .bat que ejecute esta tarea de forma automática. Este archivo debe tener un contenido similar al siguiente:

```
@echo off
set "DirectorioOriginal=%cd%"
cd /d "ruta a la raiz del proyecto"
.\build\x64\Release\UserCustomCode.exe cd /d "pause
```

Se debe reemplazar "ruta a la raíz del proyecto" por la ruta completa a la raíz del proyecto.

De esta forma, desde cualquier localización de nuestro sistema operativo, podemos ejecutar la aplicación de forma muy similar a la anterior. Por ejemplo, si nuestro script se llama *ejecutar.bat*, podemos ejecutar la siguiente linea desde una terminal desde donde se encuentre el script:

```
ejecutar.bat video "ruta al video"
```

APÉNDICE B
Objetivos de desarrollo sostenible

Objetivos de desarrollo sostenible	Alto	Medio	Bajo	No procede
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**	X			
ODS 4. **Educación de calidad.**		X		
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**				X
ODS 9. **Industria, innovación e infraestructuras.**				X
ODS 10. **Reducción de las desigualdades.**				X
ODS 11. **Ciudades y comunidades sostenibles.**				X
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**				X
ODS 17. **Alianzas para lograr objetivos.**				X

El proyecto está relacionado con algunos de los objetivos de desarrollo sostenible. A continuación se expone la relación que guarda con estos:

- Salud y bienestar: Al ayudar en la mejora de la calidad de la técnica en los deportes, puede tener una influencia positiva en los deportistas, que pueden aumentar su rendimiento. Además, este tipo de herramientas puede promover a la iniciación en un deporte por parte de mucha gente que no practicaba ninguno, al disponer de una pequeña ayuda que puede facilitar los primeros pasos.
- Educación de calidad: Al tener la posibilidad de realizar un análisis de ciertos aspectos de un deporte, puede ser utilizado en la educación física, sirviendo de ayuda a un profesor que quiera enseñar a sus alumnos la realización correcta de un movimiento. Además, también puede servir como herramienta de apoyo en la evaluación.

En conclusión, el proyecto puede llegar a tener un impacto positivo en aquellas personas que desean mejorar, iniciar o enseñar un deporte, sirviendo de guía para todas ellas.