

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



ANÁLISIS DE ACTIVIDAD EN VÍDEOS DEPORTIVOS MULTICÁMARA

Ángel Mora Sánchez
Tutor: Rafael Martín Nieto
Ponente: Jose María Martínez Sánchez

-PROYECTO FIN DE CARRERA-

Ingeniería de Telecomunicación

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Marzo 2016

ANÁLISIS DE ACTIVIDAD EN VÍDEOS DEPORTIVOS MULTICÁMARA

Ángel Mora Sánchez

Tutor: Rafael Martín Nieto

Ponente: Jose María Martínez Sánchez



Video Processing and Understanding Lab

Departamento de Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Marzo 2016

Trabajo parcialmente financiado por el gobierno español bajo el proyecto
TEC2014-53176-R (HAVideo)



Resumen.

Los sistemas de análisis de contenido en vídeos deportivos están en continuo auge desde un punto de vista tanto comercial como investigador. A este respecto, se desarrolló en el grupo de procesamiento de vídeo (VPU-Lab) de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid un prototipo de gestión de contenidos de vídeos deportivos con anterioridad a la realización de dicho proyecto. Este prototipo realizaba la detección y seguimiento de jugadores en vídeos deportivos, y sacaba algunos estadísticos de ellos.

Dicho prototipo, con buenos resultados cuantitativos y cualitativos, presentaba una serie de deficiencias que motivaron la realización de este proyecto fin de carrera. Estas deficiencias estaban principalmente relacionadas con aspectos de usabilidad, interacción con el mismo, visualización de resultados y ajuste del funcionamiento.

En este proyecto, se ha trabajado en dar solución a estos problemas en tres tareas fundamentales. En primer lugar, se trabajó en compactar el prototipo, anteriormente dividido en módulos que había que enlazar manualmente y en distintos lenguajes de programación. Al concluir el proyecto se dispone de un prototipo unificado en C++, con completa funcionalidad y portabilidad.

En segundo lugar, se trató de mejorar tanto la usabilidad como la visualización e interacción con el mismo. Para ello se desarrollaron dos aplicaciones adaptadas a dos deportes concretos, fútbol y tenis, que permiten a un usuario no-experto, manejar el prototipo y visualizar los resultados del mismo, todo ello gracias a una Interfaz Gráfica de Usuario (GUI, *Graphical User Interface*).

Por último, dado que, a la hora de comercializar estos prototipos, su funcionamiento en cliente suele ser supervisado, se ha dotado de herramientas al prototipo y a la interfaz para que sea posible una corrección del funcionamiento del algoritmo de manera *online*. Así un futuro supervisor puede interactuar con la aplicación y garantizar unos resultados muy superiores a los logrados por cualquier algoritmo automático.

Palabras clave.

Detección y seguimiento de personas, fusión de trayectorias, interfaz gráfica de usuario, vídeos deportivos.

Abstract.

Sport video-content analysis systems are on the rise both from the commercial viewpoint and the researching viewpoint. In this scope, the video processing group (VPU-Lab) of Escuela Politécnica Superior, Universidad Autónoma de Madrid developed a prototype for sport video-content analysis previously to the beginning of this master thesis. This prototype performs the detection and tracking of players in sport videos, and provides statistical information about their behavior.

This prototype achieved good results in quantitative and qualitative terms, presenting some deficiencies which motivate this mater thesis. These deficiencies were mainly related to aspects as usability, system interaction, results visualization and fine-tuning.

This project has been focused in providing a solution for those problems by working on three main tasks. Firstly, the work focused in compacting the prototype. In origin it was divided in modules which have to be manually linked and which are programmed in different languages. As a result of the project there is an unified prototype fully programmed in C++, full working and portable.

Secondly, efforts were aimed to improve the usability, interactions and results visualization of the prototype. Two applications were developed and adapted to guarantee specific sports support, football and tennis. They allow a non-expert user to fully control the prototype and visually obtain its results, via a Graphical User Interface (GIU).

Finally, keeping in mind that this products use to work under supervision in commercial applications, the prototype and the interface have been equipped with tools to allow the online interaction with its results. This improvement allows a supervisor to control the application and correct its results when necessary, obtaining more reliable results than any other automatic system.

Keywords

People detection and tracking, trajectories fusion, graphical user interface, sport videos.

Agradecimientos.

En primer lugar, quiero dar mi más sincero agradecimiento a mi tutor Rafa por haberme dado la oportunidad de realizar este proyecto. Por su atención y constante apoyo, siempre dispuesto a atenderme con buen humor para resolver dudas y comentar soluciones que hiciesen el trabajo más gratificante. Además quisiera hacer especial hincapié en la paciencia que ha tenido en este último año que ha sido bastante difícil para mí.

También, a cada uno de los componentes del VPU Lab que de una forma u otra me han ayudado a realizar este trabajo. La verdad que en este laboratorio se respira muy buen ambiente y todo el mundo se encuentra abierto a ayudar en lo que se necesite.

Me gustaría dedicar unas líneas de estos agradecimientos a mi compañero Raúl, que desde la etapa de bachillerato me ha acompañado. Hemos compartido muchos momentos, muchos trayectos en metro y autobús, muchas memorias, prácticas y tardes de sufrimiento y otras muchas de diversión.

A cada uno de mis compañeros de clase y amigos, Guille, Juanma, Mario, Pencho, David y Jorge por cada uno de los momentos que hemos compartido en estos años, tardes de trabajo sin trabajar y otras muchas apoyándonos en la lucha con las prácticas. En este apartado quisiera hacer un inciso para destacar la figura de Pencho por su ayuda y grandes consejos hasta el último momento.

Pero sin lugar a dudas a quien más tengo que agradecer su apoyo constante, su confianza en mí en todo momento y su ayuda incondicional, es a mis padres, a mi hermana Vanesa y a mi cuñado, a mi sobrina Laura que me hace los momentos más especiales, y a mi abuelo y abuela que aunque esta última ya no esté seguro que me sigue guiando por el buen camino.

A todos muchas gracias.

Ángel Mora Sánchez

Marzo 2016.

Índice general

Resumen	v
Abstract	vii
Agradecimientos	ix
1. Introducción.	1
1.1. Motivación.	1
1.2. Objetivos.	2
1.3. Estructura de la memoria.	2
2. Estado del arte.	5
2.1. Introducción al análisis de vídeos deportivos	5
2.2. Técnicas de análisis de vídeos deportivos	8
2.2.1. Extracción de fondo	8
2.2.2. Análisis de frente y detección de blobs	10
2.2.3. Identificación de equipo	11
2.2.4. Seguimiento del jugador	11
2.2.5. Seguimiento de la pelota	14
2.2.6. Asociación y fusión de la trayectoria	15
2.3. <i>Datasets</i>	18
2.3.1. <i>Dataset</i> 3DLife ACM Multimedia Grand Challenge 2010	19
2.3.2. <i>Dataset</i> de fútbol ISSIA	20
2.4. Productos comerciales	21
2.4.1. AmiscoPros	21
2.4.2. TRACAB Player Tracking System	22
2.4.3. Vis Track	23
2.4.4. Stadionus	24
2.4.5. Kizanaro Sport Technology	25
2.4.6. Nacsport	25
2.5. Conclusiones	26
3. Sistema base.	29
3.1. Descripción general del sistema base	29

4. Aplicación deportes individuales.	31
4.1. Prototipo original - deportes individuales	32
4.2. Etapa de fusión	33
4.3. Ajustes al prototipo inicial	34
4.3.1. Modificaciones paramétricas	34
4.3.2. Modificaciones de implementación	35
4.4. Interfaz gráfica de usuario	36
4.4.1. Ventana principal - <i>Application</i>	37
4.4.2. Ventana de corrección de errores - <i>Correction</i>	40
4.5. Conclusiones	42
5. Aplicación deportes de equipo	45
5.1. Prototipo original - deportes de equipo	46
5.2. Etapa de fusión	47
5.2.1. Fusión base	48
5.2.2. Fusión de blobs	49
5.2.3. Valores específicos de fusión	49
5.3. Ajustes al prototipo inicial	50
5.3.1. Modificaciones paramétricas	50
5.3.2. Modificaciones de implementación	51
5.4. Interfaz gráfica de usuario	52
5.4.1. Ventana principal - <i>Application</i>	53
5.4.2. Ventana de visualización de trayectorias - <i>Trajectories</i>	56
5.4.3. Ventana de visualización de <i>frames</i> - <i>View</i>	59
5.4.4. Ventana de corrección de errores - <i>Correction</i>	61
5.5. Resultados y conclusiones	61
6. Conclusiones y trabajo futuro.	65
6.1. Conclusiones.	65
6.2. Trabajo futuro.	67
Bibliografía	68
A. Guía de uso de aplicaciones	73
A.1. Manual de GUI de deportes individuales	73
A.2. Manual de GUI de deportes equipo	80
B. Sistema base	89
B.1. Introducción	89
B.2. Módulo de segmentación de frente	89
B.2.1. Sustracción de fondo (Background Subtraction)	90
B.2.2. Eliminación de sombra (Shadow removal)	91
B.2.3. Eliminación de ruido (Noise removal)	92
B.2.4. Extracción del blob (Blob extraction)	92
C. Presupuesto	93

Índice de figuras

2.1.	Diagrama de bloques del sistema global principal. Fuente [1]	8
2.2.	Posiciones y ejemplos de los diferentes vídeos del <i>dataset</i> 3DLife	19
2.3.	Posiciones de las seis cámaras del <i>dataset</i> de fútbol ISSIA	20
2.4.	Ejemplos de los diferentes vídeos del <i>dataset</i> de fútbol ISSIA	21
3.1.	Diagrama de bloques del sistema base. Fuente [1]	29
4.1.	Diagrama de bloques del prototipo para deportes. Fuente [1]	32
4.2.	Vista de la ventana principal.	37
4.3.	Vista de un momento determinado de la ejecución	38
4.4.	Representación de la posición del jugador en función del número de cámaras que le graban en ese momento. De izquierda a derecha y de arriba abajo, resultados con una, dos y tres cámaras.	39
4.5.	Vista de la ventana de corrección de errores.	40
4.6.	Generación del nuevo <i>bounding box</i>	41
5.1.	Diagrama de bloques del prototipo para deportes. Fuente [1]	46
5.2.	Ejemplo de LOA	48
5.3.	Vista de la ventana principal.	54
5.4.	Vista de un momento determinado de la ejecución	55
5.5.	Vista de la ventana de visualización de trayectorias.	56
5.6.	Representación de trayectorias	57
5.7.	<i>Zoom</i> aplicado sobre las trayectorias	58
5.8.	Ejemplo de LOA	59
5.9.	Vista de la ventana de visualización de <i>frames</i>	60
5.10.	Elección de <i>frame</i>	60
5.11.	Vista de la ventana de corrección de errores.	61
5.12.	Vista de la ventana con datos de ground truth	63
A.1.	Aspecto visual de la ventana principal antes de interactuar con ella.	74
A.2.	Resultado de seleccionar ambos ficheros de inicialización.	75
A.3.	Habilitado el botón Play	76
A.4.	Aspecto visual de la ventana de corrección antes de interactuar con ella.	77
A.5.	Habilitados los botones para desplazarse por los frames.	78
A.6.	Proceso de corrección de la posición de un jugador	79

A.7. Error al seleccionar el archivo de configuración.	80
A.8. Aspecto visual de la ventana principal antes de interactuar con ella. . .	81
A.9. Resultado de seleccionar ambos ficheros de inicialización.	82
A.10. Habilitado el botón Play	83
A.11. Aspecto visual de la ventana de visualización de trayectorias antes de interactuar con ella.	84
A.12. Muestra la trayectoria del identificador elegido	85
A.13. Muestra la lista de asociaciones de blobs	86
A.14. Aspecto visual de la ventana de visualización de frames antes de interactuar con ella.	87
A.15. Desplazamiento por los frames.	87
A.16. Aspecto visual de la ventana de corrección de la fusión	87
B.1. Diagrama de bloques del módulo de segmentación de frente (Foreground Segmentation Module). Fuente [1]	90
B.2. Máscara de frente en diferentes etapas: máscara de frente inicial (a), después de la eliminación de la sombra (b) y después de la eliminación del ruido (c)	91

Capítulo 1

Introducción.

1.1. Motivación.

A nivel de entretenimiento los vídeos deportivos forman una parte muy importante del número de emisiones totales tanto en la televisión pública como en la privada. Esta alta demanda por parte del gran público provoca que, aunque se haya avanzado bastante en el análisis del contenido de estos vídeos, se continúe trabajando en el enriquecimiento del material ofrecido. Este enriquecimiento se ve muy reflejado, por ejemplo, en el aporte de estadísticas sobre los deportistas. Un caso muy conocido es el análisis actividades de los jugadores en los partidos de fútbol, donde al espectador se le aporta información sobre la distancia recorrida por cada jugador, la velocidad media, zona de influencia, etc.

A nivel profesional, los avances tecnológicos en el ámbito audiovisual e informático hacen que las actuales cámaras y software de que disponen los entrenadores sean utilizados como un recurso para ayudarse en su labor de analizar la técnica de sus atletas y mejorar la táctica y el rendimiento de sus equipos. En este contexto, las herramientas de análisis de contenido en vídeos deportivos son también de gran utilidad.

Como respuesta a dichas necesidades, se desarrolló el proyecto [1], en el que se desarrolló un prototipo completo que llevaba a cabo la localización y el seguimiento de jugadores a partir de vídeos deportivos multicámara, obteniendo los estadísticos más representativos de cada uno de ellos. Este prototipo, aunque dotado de completa funcionalidad presentaba ciertas carencias que compondrán la motivación de este trabajo.

En primer lugar, el prototipo fue desarrollado en módulos implementados en distintos lenguajes. Cualquier aplicación posterior de este prototipo, tanto para su uso fuera de los ámbitos de investigación como para su mejora desde un punto de vista

funcional, requerirá de una unificación de lenguajes que facilite estas tareas posteriores.

En segundo lugar, el prototipo en el estado actual es un sistema en modo desarrollador. Por lo tanto, su manejo se ve supeditado a la presencia de un desarrollador.

Por último, el funcionamiento del algoritmo, pese a ser muy satisfactorio como se ha podido comprobar en sus posteriores publicaciones en revistas [2], congresos [3] y libros [4] internacionales, también sufre de errores. La corrección de estos errores de manera *online*, con el algoritmo en el estado actual no es posible.

1.2. Objetivos.

En base a las motivaciones expuestas, el objetivo de este proyecto será darles solución mediante una serie de objetivos parciales como serán:

- Llevar a cabo la unificación de lenguajes de los distintos módulos del proyecto, eligiendo como plataforma común aquella que garantice mejor funcionalidad de cara a una aplicación real del prototipo.
- Diseñar una aplicación con interfaces de usuario para permitir a un usuario no experto el manejo de la herramienta y la obtención de los datos que ésta facilita.
- Introducir modificaciones en el algoritmo para que sea posible la corrección de errores *online* a través de las interfaces gráficas diseñadas anteriormente. Con estos tres objetivos parciales se dispondrá de un prototipo con las características de: completa funcionalidad, manejabilidad por cualquier usuario no experto (ver guía de usuario Anexo A), capacidad de trabajar como una herramienta semi-supervisada para mejorar sus resultados, y con posibilidad de incluir mejoras y/o pos procesados directamente.

Destacar, por último, que las aplicaciones se diseñarán orientadas a las modalidades deportivas sobre las que se van a ejecutar, con el objetivo de poder hacerlas lo más específicas y adaptadas posible. En base a la funcionalidad original del prototipo, se trabajará con una modalidad de deporte individual, el tenis, y una de deporte colectivo, el fútbol.

1.3. Estructura de la memoria.

La memoria del proyecto consta de los siguientes capítulos:

- Capítulo 1: Introducción y motivación del proyecto.

- Capítulo 2: Estudio detallado del estado del arte de análisis de vídeos deportivos y de las aplicaciones comerciales disponibles.
- Capítulo 3: Descripción del sistema base.
- Capítulo 4: Trabajo desarrollado para los deportes individuales: descripción del sistema, aplicación diseñada, ajustes, pruebas y conclusiones.
- Capítulo 5: Trabajo desarrollado para los deportes de equipo: descripción del sistema, aplicación diseñada, ajustes, pruebas y conclusiones.
- Capítulo 6: Conclusiones y trabajo futuro
- Referencias y anexos.

Capítulo 2

Estado del arte.

En este capítulo se realiza un estudio del estado del arte del análisis de vídeos deportivos. Para ello, en primer lugar, se presenta el análisis de este tipo de vídeos y su aplicación e importancia en la actualidad (sección 2.1). Seguidamente, se han analizado las distintas técnicas presentes en el estado del arte y sus características (sección 2.2). Posteriormente, se han resumido los *datasets* públicos que permiten la evaluación y pruebas del sistema (sección 2.3), así como diversos productos comerciales con sus características y utilidades (sección 2.4). Por último, se han extraído una serie de conclusiones de cada una de las secciones que contiene el capítulo (sección 2.5). Nótese que algunas secciones de este capítulo utilizan como base el trabajo realizado en [1].

2.1. Introducción al análisis de vídeos deportivos

El procesamiento de vídeos deportivos ofrece una gran cantidad de posibilidades para el análisis debido a que las reglas del juego se encuentran perfectamente definidas. Los vídeos deportivos forman una parte muy importante del número de emisiones totales tanto de la televisión pública como de la privada. La alta demanda comercial ha hecho que la comunidad científica haya centrado sus esfuerzos en esta área de manera reseñable en los últimos tiempos, dando lugar a continuas mejoras y propuestas en casi todos los ámbitos deportivos.

Con el objetivo de realizar una primera clasificación, se van a estudiar los tipos de vídeos utilizados para estas tareas. Para realizar el análisis se dispone principalmente de dos tipos de vídeos deportivos:

- *Broadcast* editados [5, 6, 7, 8]: son editados para ser difundidos por televisión. En este tipo de vídeos hay escenas del juego, repeticiones y cambios del punto

de vista del observador. Es el tipo de vídeos más común.

- Vídeos multicámara: normalmente sin editar y que se pueden dividir en dos clases:
 - ✧ Móviles [9]: tienen operadores de cámara que siguen a los jugadores modificando su orientación. Frecuentemente se utilizan para generar los vídeos *broadcast*, mezclando y editando sus partes.
 - ✧ Fijos [10, 11, 12]: cámaras que permanecen inmóviles desde el inicio al final de la grabación. No es necesario que intervengan operadores de cámara para controlar la orientación de las mismas.

Una clasificación desde el punto de vista de los objetivos finales se presenta en [13]. En este trabajo se da una visión general de la investigación sobre vídeos deportivos, describiendo tanto las técnicas algorítmicas básicas como las aplicaciones. En base a esta información, se puede dividir la investigación en este ámbito en los dos siguientes objetivos principales:

- Sistemas de indexación y recuperación (basados en consultas semánticas de alto nivel).
- Sistemas de realidad aumentada (para presentar información adicional y proporcionar una nueva experiencia de visualización a los usuarios).

También es posible realizar una clasificación en función de los eventos que se quieren detectar y de los atributos a analizar para llevar a cabo la detección. En primer lugar, en la detección de eventos a nivel genérico, las características pueden ser extraídas de tres canales: vídeo, audio y texto. La definición de lo que es interesante difiere para cada tarea, e incluso dentro de una misma tarea, como los vídeos deportivos, de cada usuario. Mientras que los aficionados al deporte están más interesados en eventos como goles o puntos espectaculares en el tenis, los entrenadores suelen estar más interesados en los errores de los jugadores para ayudarles a mejorar sus capacidades.

Más concretamente, el análisis de los eventos en vídeo puede llevarse a cabo mediante la combinación de varios atributos del mismo, incluyendo su estructura, eventos y otras propiedades del contenido. Los diferentes enfoques y métodos de diseño de los algoritmos de detección de eventos son (clasificación extraída de [13]):

- Características de bajo nivel vs. Características relacionadas con el objeto: Las características de bajo nivel son las adquiridas directamente del vídeo de entrada, y las características relacionadas con el objeto son atributos de los objetos

tales como la posición de la pelota o la forma del jugador, adquiridas por algoritmos más complejos.

- Monocanal vs. Múltiples canales: algunos eventos pueden ser detectados mediante características de un sólo canal (ej. vídeo) y para otros eventos se necesitan las características de múltiples canales, por lo general, vídeo y audio.
- Juego específico vs. Genérico: la mayoría de los detectores de eventos desarrollados son para un juego específico porque los diferentes juegos tienen diferentes eventos, pero el objetivo ideal es un algoritmo que pueda detectar los eventos de múltiples juegos.
- Modelo de características vs. Patrón de contexto: algunos algoritmos modelan directamente las relaciones entre las características y los eventos, pero algunos eventos no tienen relaciones intuitivas con las características, requiriendo información de contexto para la detección y el uso de patrones para capturar la información de contexto.
- Basado en aprendizaje vs. No basado en aprendizaje (llamado "normal" en [13]): los primeros algoritmos capturan las relaciones entre las características y los eventos mediante análisis estadístico y optimización. En general, se elige el enfoque de aprendizaje de la máquina cuando las relaciones entre las características y los eventos son difíciles de definir intuitivamente.
- Vídeo *broadcast* vs. Vídeo no *broadcast*: la mayoría de los algoritmos son diseñados para vídeos *broadcast* de los que se puede utilizar la información tras la edición como fuente adicional de información.

Además de todas estas opciones a valorar, los algoritmos pueden incluir en sus tareas o como información útil para la tarea final, aspectos también relevantes como pueden ser: el análisis de la estructura (por ejemplo de un partido), detección de objetos y segmentación (por ejemplo, la posición de los jugadores, su forma y los movimientos de los atletas), calibración de la cámara (para determinar la correlación entre las coordenadas en la imagen obtenida y las posiciones en el mundo real) o la reconstrucción 3D: varias cámaras pueden ser utilizadas para obtener vistas desde diferentes posiciones y calcular una reconstrucción completa del campo de juego.

Por último, cabe realizar una clasificación en base a la información o aplicación para la que se usará principalmente el procesamiento de vídeos deportivos:

- Resúmenes [14]: creación de una versión reducida conteniendo la información esencial. Para ello, es necesario encontrar los puntos de mayor interés.

- Táctica y análisis de rendimiento [15, 16]: permitir evaluar el rendimiento de los jugadores a través de sus movimientos en el terreno de juego.
- Reconstrucción 3D de jugadas [17]: su objetivo es proporcionar al usuario un punto de vista más real o visualizar la jugada desde la perspectiva del árbitro.
- Vídeos para pequeños dispositivos móviles [18]: transmisión de eventos deportivos en PDA, *smartphones*, etc.
- Asistencia arbitral: para ayudar al árbitro en los casos de decisiones difíciles y parcialmente sustituir el trabajo del árbitro.

2.2. Técnicas de análisis de vídeos deportivos

El sistema completo para la detección y el seguimiento de los jugadores sobre el campo de juego puede ser dividido en varios bloques, como se muestra en la Figura 2.1. Los distintos bloques serán descritos en las siguientes subsecciones. Destacar que la mayoría de estos módulos son tareas comunes en gran parte los sistemas de análisis de vídeo deportivo.

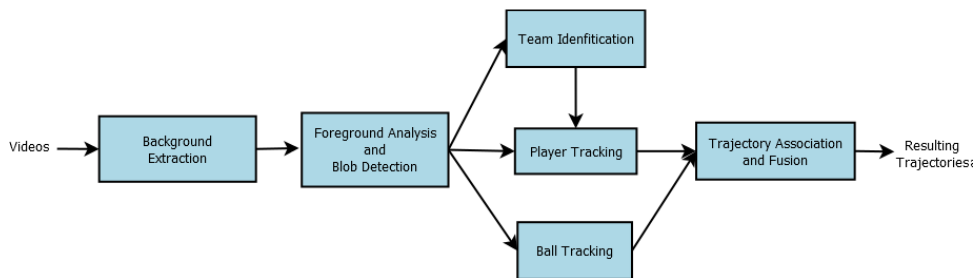


Figura 2.1: Diagrama de bloques del sistema global principal. Fuente [1]

2.2.1. Extracción de fondo

La extracción de fondo es una tarea del procesamiento de vídeo que busca la segmentación de objetos de frente, que serán típicamente aquellos objetos en movimiento. Habitualmente se basa en un modelo de fondo. Este modelo se genera a partir de una imagen vacía del campo de juego (si está disponible) o de un fragmento del vídeo con jugadores no estáticos. Algunos de los principales problemas en la extracción de fondo son los cambios en el entorno, tales como cambios de iluminación, sombras u objetos del fondo con movimiento, que se conocen como fondo dinámico.

En [8] se presenta una aproximación de modelo de fondo basada en la extracción de información de cada píxel a lo largo de un tiempo y modelar de forma estadísti-

ca, mediante aproximaciones gaussianas, los valores que va tomando. Estos métodos actualizan modificando la distribución gaussiana según llegue nueva información (se supone que el fondo será la información dominante). Los resultados demuestran que estos algoritmos trabajan bien en escenarios relativamente simples.

Con una aproximación más orientada a aplicación, [5] propone entrenar histogramas para detectar los píxeles del campo de juego, utilizando un conjunto de vídeos de fútbol de entrenamiento. Los píxeles en los vídeos de entrenamiento se etiquetan como pertenecientes al campo de juego o no pertenecientes al mismo, ya sea manualmente o utilizando un método semi-supervisado. Basado en su vector de color RGB, cada píxel del campo de juego marcado se coloca en el lugar apropiado del histograma del campo de juego. Un proceso similar se lleva a cabo para los píxeles etiquetados como no pertenecientes al campo de juego.

En [11], el fondo se modela con una mezcla de gaussianas y antes del partido aprende sin la necesidad de una escena vacía. El detector de fondo se utiliza primero para extraer una máscara del campo, y de esta forma ahorrar tiempo de procesamiento y evitar falsas alarmas. Esta máscara de campo se calcula utilizando el histograma de tonalidad y la proyección de la geometría del campo conocida previamente.

Un modelo de fondo se genera en [9] con un procedimiento de actualización de fondo utilizado para adaptar continuamente el modelo a las variaciones en las condiciones lumínicas.

En [12], se extrae la parte de fondo perteneciente al campo de juego, en primer lugar mediante la segmentación de las regiones que no pertenecen a lo que es estrictamente el terreno de juego, como por ejemplo los anuncios, y luego se extraen los jugadores sobre la base del fondo obtenido. El campo tiene un color uniforme y ocupa gran área en la imagen. Así, se calculan los histogramas de cada color, R / G / B. Usando los valores de pico, la imagen se digitaliza, teniendo en cuenta un valor umbral para cada color. Después de obtener la imagen binaria, se aplica filtrado morfológico y el mayor componente conexo se extrae, que representa el campo, pero tiene muchos huecos causados por los jugadores. La máscara de campo se obtiene llenando el interior de estos huecos.

El campo está representado en [7] por un valor medio de color que se obtiene a través de estadísticas previas sobre un amplio conjunto de datos. La distancia de color entre un píxel y el valor medio del campo se utiliza para determinar si pertenece al campo o no. En el sistema, solo se calculan el tono y los componentes de saturación en el espacio de color HSV para excluir el efecto de la iluminación.

2.2.2. Análisis de frente y detección de blobs

La detección de blobs se basa en la extracción de componentes conexas de frente que, a priori, corresponderán a los elementos de interés, en este caso, los jugadores. El objetivo es ajustar un blob tanto como sea posible al contorno del jugador, tratando de garantizar la separación de los jugadores y teniendo en cuenta las posibles superposiciones de éstos. La determinación del número de blobs es una etapa de gran relevancia para tomar decisiones correctas durante las etapas posteriores de seguimiento (*tracking*).

En [8], los nodos se agrupan teniendo en cuenta los bordes comunes. El área de los blobs es el parámetro utilizado para definir el número de componentes de cada nodo. El grafo se construye a partir del conjunto de blobs obtenidos durante la etapa de segmentación de tal manera que los nodos representan los blobs, y los bordes representan la distancia entre ellos. Estos blobs son considerados como regiones de las imágenes y se aplica una secuencia de operaciones morfológicas para el pos procesamiento y su división. Un blob en el *frame* actual se puede dividir en dos o más, en función del número de componentes del nodo correspondiente en el grafo y en su configuración en el *frame* anterior.

En [5], el análisis de componentes conexas (*connected component analysis*, CCA) estudia la máscara binaria y agrupa sus píxeles en componentes basados en la conectividad de píxeles. Idealmente, los píxeles que no pertenecen al terreno de juego dentro del área de campo extraída deben ser los píxeles de frente que se pueden agrupar en diferentes blobs de frente por CCA. Las zonas ruidosas tienen que ser eliminadas más tarde mediante análisis de forma y modelos de apariencia de objetos.

La máscara de frente se filtra morfológicamente en [10] (cierre tras apertura) para deshacerse del ruido. Seguidamente se lleva a cabo un análisis de componentes conexas que permite la creación de blobs individuales y *bounding boxes* (cajas delimitadoras que contienen la máscara de cada blob).

En [9] se introduce una mejora en el análisis de conectividad para la obtención de regiones conectadas y eliminar pequeñas regiones debidas al ruido. Este procedimiento escanea la imagen total y agrupa píxeles vecinos en regiones. El análisis de conectividad elimina sombras utilizando consideraciones geométricas sobre la forma de cada región. Las extensiones de áreas en las direcciones ortogonales con respecto a la posición vertical esperada de jugadores se eliminan durante la construcción de las regiones conexas.

Al final de este paso, cada región conectada, que se tiene que considerar en la etapa de seguimiento (*tracking*) posterior, es encuadrada en un *bounding box*.

2.2.3. Identificación de equipo

La información de los uniformes de los equipos se utiliza para identificar el equipo del blob y diferenciar entre los jugadores, los porteros o los árbitros. En general, un jugador puede ser modelado como un grupo de muchas regiones, sin embargo, al analizar cada región se pueden observar algunos colores predominantes.

En [8] se propone dividir el modelo del jugador en dos o más regiones, de modo que cada región represente una parte del uniforme del equipo, por ejemplo, la camiseta, los pantalones o los calcetines. Para cada región, se define un filtrado basado en la distribución de intensidad vertical de los blobs. Los valores límite son definidos a partir del valor mínimo, máximo y medio de la distribución vertical para cada región.

Los histogramas de color se utilizan para clasificar los objetos en clases en [10]. Las diferentes clases consisten en el equipo 1, el equipo 2, portero del equipo 1, portero del equipo 2, y árbitro. Se toma la parte central de los blobs detectados y se calculan los histogramas de color. Los histogramas se comparan, después de la normalización, con los histogramas de color de las plantillas utilizando la distancia Bhattacharyya.

Podrían usarse técnicas de *clustering* automático, sin embargo, esto supondría aumentar la complejidad de aprendizaje y producir resultados más inciertos, por lo que no existen muchas aproximaciones en este sentido.

La selección de las plantillas para las métricas de histogramas puede ser muy sencilla, por ejemplo, al usuario se le puede pedir clasificar las *bounding boxes* del primer *frame*, o de mayor complejidad, implementado usando un método de histograma-intersección como en [11]. El resultado para cada jugador es un vector de cinco elementos, lo que indica la probabilidad de que el jugador observado esté usando una de las cinco categorías de uniforme (las categorías son las mismas que en [10]).

Por último, en [12] se analiza utilizando un histograma de retro proyección de color para aislar jugadores en cada equipo. Pero como diferentes equipos pueden tener histogramas parecidos, se utiliza la distribución vertical de colores. Después de calcular la distribución vertical de RGB, se compara con el modelo de distribución de cada equipo. La distribución modelo se obtiene en la primera imagen. La medida de similitud se calcula por mediante una convolución, dentro de un pequeño rango porque la distribución del histograma vertical de jugadores del mismo equipo puede variar ligeramente.

2.2.4. Seguimiento del jugador

Una vez detectada la posición de cada jugador, el siguiente objetivo es su seguimiento de tal modo que se pueda extraer información del mismo a lo largo del

tiempo.

En [8], el seguimiento de cada jugador se lleva a cabo mediante la búsqueda de una ruta óptima en el gráfico definido en la detección de blob. Con el fin de realizar el seguimiento de un jugador, se define en cada *frame* y para cada jugador un blob, y se busca en un grafo su nodo correspondiente. En cada paso, el camino mínimo en el grafo se considera utilizando la información de la distancia entre los blobs. Durante una oclusión de dos jugadores, se utiliza la información sobre la dirección de la trayectoria de cada uno a fin de decidir los caminos correctos. El seguimiento de los jugadores en caso de contacto u oclusiones por otros jugadores es uno de los retos más complejos del sistema.

El *bounding box* y las coordenadas del centroide de cada jugador se utilizan en [11] como estado y variables en un filtro de Kalman. Para el proceso de seguimiento de múltiples vistas, se aplica un procedimiento en tres pasos. El primer paso es asociar las medidas a las trayectorias establecidas y actualizar estas trayectorias. El segundo paso es inicializar trayectorias para las medidas que no están asociadas a ninguna trayectoria existente. Por último, se fija la población para cada categoría de jugadores (diez jugadores de campo y un portero por equipo, tres árbitros) y se utiliza esa información para reconocer a los miembros de cada categoría y mejorar los resultados del seguimiento.

En la actualización de trayectoria, a cada jugador se le modela como una trayectoria y tiene su estimación de posición actualizada, si es posible, mediante una medición global fusionada para las cámaras disponibles. En inicialización de seguimiento, las mediciones no asociadas después de cotejarlas con trayectorias existentes se comparan entre sí para encontrar nuevas trayectorias potenciales. Por último, en la selección de trayectorias hay un procedimiento de seguimiento asistido por el reconocimiento de los 25 jugadores más probables (11 de cada equipo y 3 árbitros). Debido a las falsas alarmas y errores de seguimiento, normalmente existen más de 25 trayectorias para los jugadores.

El cálculo de la posición probable de un jugador se realiza en base a la confianza de la categoría estimada, el número de cámaras de soporte, conocimiento del dominio en las posiciones (para los porteros y jueces de línea), *frames* en los que ha sido seguido o perdido, así como la restricción de población fijada. Dado el elevado número de restricciones que se pueden imponer, un método sencillo de búsqueda su óptima, con bajo coste computacional garantiza resultados razonables.

En [9] se utiliza la técnica de Máxima Probabilidad a Posteriori (MAP) para conseguir el mejor ajuste de estados y de observaciones en el seguimiento. El vector de estado incluye información sobre la ubicación, velocidad, aceleración y la dimensión

del *bounding box* individual. La configuración proveniente del MAP es la seleccionada como mejor ajuste entre la observación y la predicción del estado. En este punto es necesario un paso más para la validación de la predicción. Mediante la comparación de las uniones entre las observaciones con las predicciones, se obtienen dos situaciones: hay algunas observaciones que no coinciden con ninguna predicción y hay algunas predicciones que no se corresponden con ninguna observación.

En el primer caso pueden producirse situaciones diferentes:

1. La observación podría ser un nuevo blob de entrada si está en el borde de la imagen, luego un nuevo blob es generado con un estado entrante.
2. La observación podría ser un blob reanudado si está cerca de una predicción con un estado desaparecido.
3. La observación podría ser generada por el ruido, a continuación, se crea una nueva entidad y se observa a lo largo de una ventana temporal hasta que se toma una decisión sobre su persistencia.

En el segundo caso, si la predicción no tiene una correspondencia entre las observaciones y no está en el borde de la imagen (que no está en una situación saliente), significa que el paso de segmentación de frente no estaba detectando el blob y el vector de estado se mantiene como blob desaparecido. Se requiere un análisis adicional cuando la fusión se produce entre dos o más blobs. Predecir que dos o más blobs se fusionarán es posible, pero para mantener su vector de estado separado hay que dividirlos lógicamente en las observaciones correspondientes. Este procedimiento de división podría ser difícil especialmente cuando dos o más jugadores están muy cerca entre sí y la oclusión es casi total.

El emparejamiento de plantillas y el filtro de Kalman se utilizan en [12]. Las plantillas de jugadores se extraen de la máscara del jugador utilizando extracción de componentes conectadas. En primer lugar, los nuevos jugadores que no se superponen significativamente con el *bounding box* de un jugador ya seguido se detectan como tal. Luego, los jugadores nuevos se insertan a la lista de seguimiento. La ubicación de los jugadores en el siguiente *frame* se predice por el filtro de Kalman y se realiza el emparejamiento de plantillas en esa ubicación. Por último, la plantilla del jugador se actualiza. El principal problema del seguimiento de jugadores es la oclusión y en [12] sólo se considera la oclusión entre los diferentes equipos.

Por último, en [19], necesita partir de un *frame* donde no haya jugadores que se ocluyen entre sí. Desde este punto, los jugadores son seguidos automáticamente teniendo en cuenta sus coordenadas 3D. Cuando una fusión de dos (o más) jugadores

es detectada, la separación se realiza por medio de la correlación entre las plantillas de los mismos.

2.2.5. Seguimiento de la pelota

El método para el seguimiento de la pelota es similar al método utilizado para el seguimiento de jugador, pero es más difícil que el seguimiento de jugadores ya que la detección automática de la pelota es más difícil, debido a sus reducidas dimensiones en la imagen. Además, si un jugador tiene la pelota, el seguimiento es difícil porque la pelota es frecuentemente ocluida por el jugador.

A continuación, se describen las diferencias y rasgos característicos de las propuestas a este respecto realizados por distintos sistemas del estado del arte.

En [5], hay un rango predefinido para el área del blob de la pelota según la configuración de la cámara. Mientras tanto, la proporción de píxeles blancos detectados en el blob indica la posibilidad de ser una pelota. La redondez y la excentricidad de un blob candidato deben estar cerca de 1.0, así se diferencia de los segmentos desconectados de las líneas del campo de juego. Otra opción es, usando una transformada de Hough sobre la máscara de frente [10] para detectar los píxeles candidatos que se corresponden con los círculos.

La información temporal se utiliza entre los *frames* mediante la creación de una ventana de búsqueda alrededor de la última posición encontrada de la pelota. Sólo en esta ventana, y sólo en los blobs de frente correspondientes se busca la ubicación nueva de la pelota. Esto permite la prevención de que se busque por la totalidad del *frame*, evitando coste de procesado y muchos posibles falsos positivos. Para evitar la propagación de errores debido a fallos en la detección de la pelota, cada 10 *frames* se realiza una búsqueda menos restringida del mejor candidato a pelota.

Cuando el jugador lleva la pelota pegada al cuerpo la detección cae. Sin embargo, esta información puede ser utilizada. La ventana de búsqueda se amplía para incluir el *bounding box* del jugador. Como tal, el balón se puede detectar cuando el jugador lo pasa. Por último, los círculos detectados en cada *frame* son evaluados por comparación del color medio en el círculo con el color medio de una plantilla de la pelota. Esto permite producir una mejor combinación para cada *frame* dentro de cada vista de cámara.

En [12], la posición y el *bounding box* del balón se inicializan manualmente en el momento de partida. Si un jugador está corriendo cerca de la pelota, él se marca como "tiene la pelota". Después de que el seguimiento de la pelota se ha detenido, la pelota se busca en torno al jugador que tiene el balón. Si se encuentra, su seguimiento continúa otra vez de forma independiente hasta que se asocie a otro jugador.

En [6], el diagrama de flujo del sistema se compone de dos módulos: detección de pelota y seguimiento de la misma.

El procedimiento de detección del balón, se basa en el color, la forma y el tamaño. La idea de la detección de pelota es utilizar un grafo para contener múltiples hipótesis de su localización. Mediante tres pasos básicos para este procedimiento: detección de candidatos, construcción del grafo y extracción de la ruta óptima al candidato de pelota. Se aplica el algoritmo de Viterbi para extraer el camino óptimo.

Una vez que se detecta la pelota, el procedimiento de seguimiento de la pelota se basa en el filtrado de Kalman y se inicia con el emparejamiento de plantillas. El filtro de Kalman predice la ubicación de la pelota en el próximo *frame* y filtra el resultado de seguimiento en el actual. Para obtener la observación se usa la comparación de plantillas. El filtro de Kalman y la plantilla se inicializan usando los resultados de la detección.

Se propone un método de dos niveles de precisión en [7] para la detección y seguimiento de pelota basado en la condensación. Tomando la similitud del histograma ponderado de la región de objeto como modelo de observación, el algoritmo de condensación se utiliza en el seguimiento de la pelota. El marco de trabajo del algoritmo consiste en tres grandes módulos de procesamiento:

1. Detección del balón de fútbol: se compone de tres partes principales. La extracción del campo de juego, análisis de la forma (una estrategia de búsqueda de poca precisión a mucha precisión se utiliza para identificar una única pelota y el contorno de cada objeto es trazado y utilizado para el análisis) y el análisis del color de las regiones y las probabilidades de fusión (la región de similitud final se fusiona por el color y la forma similar, que debe ser normalizado antes de la fusión).
2. Seguimiento del objeto. El algoritmo de condensación (basado en un muestreo factorizado pero extendido a aplicar iterativamente para imágenes sucesivas en una secuencia) se utiliza para el seguimiento de la pelota en este método. Trata la secuencia de imágenes en movimiento como un sistema dinámico.
3. Optimización de la región del objeto, actualización y medida de confianza.

2.2.6. Asociación y fusión de la trayectoria

La reconstrucción de las trayectorias globales a través de los múltiples puntos de vista requiere la fusión de las vistas simultáneas del mismo objeto, así como la fusión de los fragmentos de trayectoria capturados por los sensores individuales. Los datos

de trayectoria de los sensores individuales pueden contener errores e imprecisiones causadas por el ruido, las re entradas de los objetos y las oclusiones.

Hay dos enfoques principales para la generación de las trayectorias fusionadas: fusionar y luego seguir [20], y seguir y después fusionar [9, 11, 20, 21, 22].

Un filtro de partículas para seguir antes de detectar destinado a multicámara (TBD) y que utiliza *mean-shift* se presenta en [23].

La información de múltiples cámaras se fusiona primero para obtener un volumen de detección utilizando una homografía multicapa. Para realizar un seguimiento de varios objetos en el volumen de detección, a diferencia de los algoritmos de seguimiento tradicionales que incorporan la medición a nivel de probabilidad, TBD utiliza la intensidad de la señal en la representación del estado. Además, como diferentes objetivos pueden tener diferentes intensidades de señal en el volumen de detección, se contabiliza esta variación en la estrategia de actualización de peso. Por último, a diferencia de los métodos tradicionales que utilizan K-means o Mezcla de Gaussianas (MoG) en las detecciones, el enfoque propuesto no requiere inicialización manual de los objetivos o el conocimiento previo del número de grupos ya que utiliza media de desplazamiento (*mean-shift*) de las partículas, después de la etapa de actualización.

La matriz de asociación también se puede decidir de acuerdo a la distancia de Mahalanobis [11] entre las coordenadas de medición y la predicción del seguimiento. Para resolver el problema de correspondencia, el mejor conjunto de trayectorias se calcula hallando el camino de máximo peso. Este paso se puede realizar utilizando el algoritmo de Hopcroft-Karp [9].

En [20], después de obtener las trayectorias transformadas, el siguiente paso es calcular sus pares similares para la asociación y la fusión con el fin de tener una sola trayectoria correspondiente a un objeto a través de todo el campo. La siguiente suposición se hace para asociación y fusión:

1. Dos trayectorias que están cerca en el espacio y el tiempo.
2. Tener una forma similar se considera que son generados a partir del mismo objeto observado por dos cámaras. Dos cámaras montadas en diferentes posiciones y con diferentes orientaciones puede forzar una transformación afín en la trayectoria del objeto.

Los parámetros utilizados para la asociación en [21] son: forma, longitud, velocidad media, cambios de dirección (que define las características direccionales estadísticas de una trayectoria), trayectoria media y el análisis de componentes PCA. Con todos esos parámetros, se genera el vector y la correlación cruzada se utiliza como medida de proximidad. Para la fusión de la trayectoria, se utiliza un método de ponderación

adaptativa, donde se calculan los pesos en función del número de observaciones para cada trayectoria.

Los algoritmos de análisis de multicámara que generan trayectorias globales pueden ser separados en tres clases principales: basada en la apariencia, basada en geometría y aproximaciones híbridas. En base a [20], las ideas principales de cada clase con algunos ejemplos para cada uno se describen a continuación.

2.2.6.1. Aproximaciones basadas en apariencia

Los enfoques basados en la apariencia utilizan el color para relacionar los objetos a través de las cámaras. La información de color y el filtro de asociación de datos de probabilidad conjunta se utilizan en [24] para un seguimiento aproximado de jugadores de fútbol. Una aproximación del seguimiento de objetos en base al color con la implementación de un filtro de partículas se utiliza en [25] en un entorno multicámara. Las imágenes difundidas de TV y un método de seguimiento basado en el emparejamiento de plantillas y en un histograma de retro proyección se utilizan en [12] para resolver el problema de la oclusión. Los métodos basados en la apariencia generalmente sufren de variaciones de iluminación que socavan la eficacia de color como una señal. Además, la información de color por sí sola no basta para eliminar la ambigüedad de los elementos de un grupo, como entre los miembros de un mismo equipo en un escenario deportivo.

2.2.6.2. Aproximaciones basadas en la geometría

Los enfoques basados en la geometría establecen correspondencias entre los objetos que aparecen simultáneamente en diferentes puntos de vista. Estas aproximaciones generalmente explotan la geometría epipolar, la homografía y la calibración de la cámara. Un método que rectifica trayectorias y modela los caminos de la gente se propone en [26]. Utilizando una aproximación no lineal, las cámaras están calibradas durante un entrenamiento no supervisado, y esta información se usa en dicho algoritmo para rectificar las trayectorias. Los modelos globales de trayectoria son construidos a partir de estas trayectorias y una medida de similitud que se utiliza para relacionarlas con las trayectorias de entrada de dichos modelos.

En [27], se utilizan ocho cámaras que cubren todo el área, y se obtienen los datos de seguimiento mediante el uso de homografía y una imagen de tierra virtual. Se propone una extensión de los modelos de distribución de puntos en [28] para analizar el movimiento del objeto en sus dimensiones temporales, espaciales y espacio-temporales. Los movimientos se expresan en términos de modos y asociados a un comportamiento

particular. Los métodos basados en restricciones geométricas puras dependen en gran medida de la precisión durante el proceso de correspondencia. Por ejemplo, la geometría epipolar sufre de ambigüedad generada por la correspondencia punto a línea [29].

2.2.6.3. Aproximaciones híbridas

Los métodos híbridos utilizan múltiples características para integrar la información en la red de la cámara. Un enfoque estadístico para asociar trayectorias a través de múltiples puntos de vista de las cámaras se presenta en . Se supone la disponibilidad de un plano de tierra y una duración mínima durante la cual al menos un objeto es observado por dos cámaras. Tomando [22] como entrada trayectorias con información temporal de cada punto de vista, el algoritmo calcula las transformaciones, la asociación de objetos a través de puntos de vista y las trayectorias canónicas. Estas se consideran las mejores estimaciones en el sentido de máxima verosimilitud.

En [30], se presenta un algoritmo de seguimiento multicámara que utiliza una representación de grafos para encontrar la posición de los jugadores de fútbol en el terreno de juego. Se propone la integración de múltiples características (por ejemplo textura, color, región, movimiento) y de múltiples puntos de vista [31].

Un algoritmo que combina el filtrado de las partículas y la propagación en un sistema unificado se presenta en [32]. Algoritmos de seguimiento basados en filtros de partículas locales interactúan entre sí para compensar las pobres observaciones individuales. Este algoritmo no funciona cuando ocurren oclusiones, pero lo justifica su irrelevancia en la corta duración de estas situaciones.

2.3. Datasets

La selección de *datasets* tiene como objetivo escoger el conjunto de datos que permita probar el funcionamiento del prototipo diseñado. Esta selección ya se llevó a cabo en [1], y por lo tanto solo se van a describir brevemente a continuación aquellos que se han usado para probar los prototipos desarrollados y las modificaciones para corrección *online* de errores de manera semi-supervisada. Destacar antes de entrar en detalle en cada uno, que se seleccionaron buscando que contuvieran idealmente no sólo las secuencias de vídeo multicámara, sino también datos del *ground-truth* o los datos de calibración, que permitieran llevar a cabo evaluaciones cuantitativas. Todos los *datasets* usados se han grabado con cámaras estáticas. Por lo tanto, el fondo de la vista es la misma para cada secuencia de cada cámara, lo que permite aplicar una homografía constante, lo cual fue de gran relevancia a la hora de realizar el diseño del

prototipo original. Por último, tal y como se comentó en los objetivos del proyecto, se pretende diseñar un sistema en dos modalidades deportivas, tenis y fútbol, por lo que cada *dataset* contendrá secuencias sobre uno u otro deporte.

2.3.1. *Dataset* 3DLife ACM Multimedia Grand Challenge 2010

Este *dataset*¹, tiene un aspecto como el mostrado en la siguiente Figura 2.2, y presenta las siguientes características:

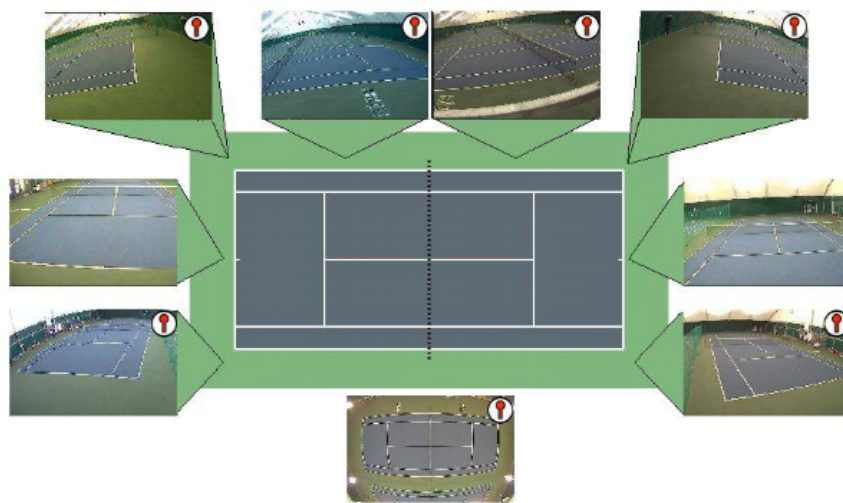


Figura 2.2: Posiciones y ejemplos de los diferentes vídeos del *dataset* 3DLife

- Deporte: Tenis.
- Sistema de cámaras: 9 cámaras de circuito cerrado de televisión colocadas en diferentes puntos alrededor de toda una pista de tenis (ver Figura 2.2). Disponible el audio de las cámaras 7 y 9 (cada cámara con el símbolo de micrófono en la parte superior derecha de la imagen).
- Características de los vídeos: Los vídeos son archivos ASF y codificados usando un *codec* MPEG-4. Siete de los vídeos (tomados de las cámaras con el símbolo de micrófono en la parte superior derecha de la imagen) se registran con una resolución de 640x480 píxeles de las cámaras de red Axis 212PTZ. Las otras dos cámaras tienen una resolución de 704x576 píxeles y se capturaron con las cámaras de red Axis 215PTZ.

¹http://www.cdvp.dcu.ie/tennisireland/TennisVideos/acm_mm_3dlife_grand_challenge/

- Datos relevantes para la evaluación: Aunque la hora de inicio de cada vídeo se sincroniza a través de software en el inicio de cada secuencia, los vídeos no están referenciadas juntos.

2.3.2. *Dataset* de fútbol ISSIA

El *dataset* público ISSIA² mostrado en la Figura 2.3 posee las siguientes características:

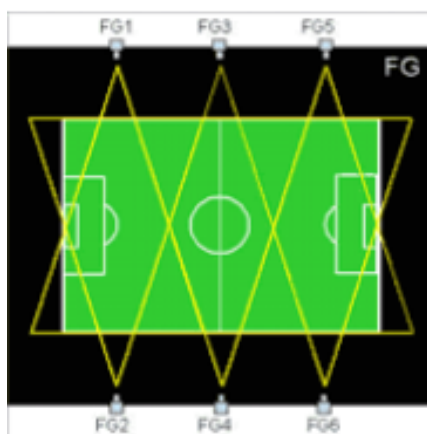


Figura 2.3: Posiciones de las seis cámaras del *dataset* de fútbol ISSIA

- Deporte: Fútbol.
- Sistema de cámaras: Seis vistas sincronizadas adquiridas por seis cámaras de alta definición, tres por cada lateral del campo de juego. Todas las cámaras son DALSA 25-2M30. Las posiciones concretas de las seis cámaras en los dos laterales del campo se muestran en la Figura 2.3, y un ejemplo de las vistas reales se observan en la Figura 2.4.
- Características de los vídeos: Vídeos capturados a 25 imágenes por segundo (6 archivos AVI).
- Datos relevantes para la evaluación: Anotación manual de la posición de los objetos de dos minutos del partido. Estos metadatos proporcionan las posiciones de los jugadores, árbitros, y la pelota en cada *frame* de cada cámara (6 archivos XML). Los jugadores tienen las mismas etiquetas mientras se mueven en los seis puntos de vista que se corresponden con los números en sus uniformes. Las etiquetas de jugador del primer equipo empiezan desde 1 mientras las etiquetas

²<http://www.issia.cnr.it/htdocs%20nuovo/progetti/bari/soccerdataset.html>

del segundo equipo parten de 201. Los primeros 300 *frames* de cada secuencia no se han etiquetado con el fin de proporcionar una fase inicial para inicializar los algoritmos de extracción de fondo.



Figura 2.4: Ejemplos de los diferentes vídeos del *dataset* de fútbol ISSIA

2.4. Productos comerciales

En este apartado se realiza la descripción de algunos productos comerciales, cuyas características se han extraído de la documentación pública y páginas web de los mismos, ya que, al no ser gratuitos, no se ha tenido la posibilidad de probarlos y analizarlos.

2.4.1. AmiscoPros

El Sistema Amisco³ es una solución para la creación y gestión de los datos de análisis deportivo. La funcionalidad clave es medir y almacenar datos tácticos y físicos para el deporte. El sistema se basa en una tecnología de seguimiento patentada, y utiliza un conjunto de sensores instalados en el estadio para producir una animación en 2D con datos exclusivamente tácticos y físicos. Este sistema proporciona una medición objetiva de las posiciones y los movimientos de los jugadores en el terreno de juego.

Los datos obtenidos se destinan a software interactivo AMISCOPRO para su consulta. Se implantan ocho sensores alrededor del campo. Capturan y miden obje-

³<http://www.prozonesports.com/subsector/foot ball/>

tivamente la posición y los movimientos de los jugadores, los árbitros y la pelota 25 veces por segundo durante todo el partido.

Dentro de la plataforma existen herramientas con diversas funcionalidades en base al usuario objetivo al que van dirigidas:

- PROZONE3/AMISCO PRO es una plataforma de análisis de rendimiento de fútbol. Ofrece datos técnicos, tácticos y físicos únicos a través múltiples cámaras en el estadio, lo cual permite el análisis posterior al partido completo y el seguimiento de jugadores. Utiliza una instalación multicámara en el estadio para garantizar el 100 % de cobertura y un seguimiento preciso jugador. Es un sistema validado de forma independiente. Utilizado por clubes de fútbol como Ajax, Arsenal, Chelsea FC, Bayern de Munich, Bayer Leverkusen, etc., facilita el análisis objetivo con un gran nivel de detalle. Totalmente interactivo y personalizable, presenta una gran cantidad de datos de rendimiento a través de una interfaz intuitiva y atractiva.
- Prozone también ofrece servicios integrales de análisis a los árbitros de élite para ayudar con el seguimiento y la evaluación de su labor. PROZONE3 Referee ofrece imágenes de vídeo detallado e información de la aptitud para ayudar a los árbitros y sus asesores en la evaluación objetiva de su trabajo para asegurar el más alto nivel de eficiencia. PROZONE3 Referee entrega el material de archivo y los datos necesarios para apoyar el análisis objetivo de las decisiones clave y así colaborar con las organizaciones que buscan mejorar la precisión y consistencia del arbitraje.

2.4.2. TRACAB Player Tracking System

El Sistema TRACAB⁴ de ChyronHego utiliza una avanzada tecnología patentada de procesamiento de imágenes para identificar la posición y la velocidad de todos los objetos que se mueven dentro del campo de juego y lo hace en tiempo real.

Los datos resultantes de las coordenadas X, Y, Z de alta precisión se suministran 25 veces por segundo para cada uno y todos los objetos visibles, ya sean jugadores, árbitros o incluso la pelota. Abarca casi una década de desarrollo continuo y desplegado a través de los ambientes deportivos en vivo más exigentes y muchos miles de eventos deportivos, el sistema TRACAB ha madurado hasta convertirse en el estándar que define el seguimiento de los deportes en vivo.

El sistema consta de dos unidades multicámara, que contiene ocho pequeñas cámaras cada una. Mediante el uso de más de una cámara el sistema se hace menos

⁴<http://chyronhego.com/sports-data/player-tracking>

sensible a un fallo de cámara, incluso si una cámara falla la unidad seguirá siendo operativa. También proporciona las unidades de cámara con la función de la resolución dinámica, lo que significa que las cámaras individuales pueden estar en ángulo de manera que la unidad de cámara proporciona una resolución más alta en las partes del terreno de juego, donde hay más oclusiones, como las áreas de penalti en el fútbol.

Los datos creados por el sistema TRACAB ofrecen enormes posibilidades de análisis, ya sea para las transmisiones deportivas, otros medios de comunicación, las federaciones deportivas, clubes o aficionados. Las métricas que evalúan el rendimiento son distancia recorrida, velocidad, resistencia, formaciones de equipo y muchas más. También puede alimentar a una amplia gama de plataformas de visualización, ya sean en 3D, interactivo o móvil, para ayudar a explicar mejor cómo se desarrolla el juego.

Las transmisiones deportivas pueden utilizar los datos para ayudar a contar mejor lo que ocurre en el terreno de juego para los aficionados y para que los analistas expertos pueden utilizar los datos para explicar empíricamente la dinámica del juego.

2.4.3. Vis Track

El sistema VIS.TRACK⁵ permite evaluar los datos de rendimiento de todos los jugadores en tiempo real y archivarlos. Con dos cámaras, para analizar todo lo que sucede en un partido es posible, porque no sólo los datos de jugador, sino también los datos de la pelota se capturan. Con el software VIS.TRACK, esta información se puede representar en paralelo en animaciones 3-D y gráficos. Una instalación permanente no es necesaria para el sistema VIS.TRACK. El sistema se puede configurar en sólo 30 minutos, por lo que puede ser utilizado no sólo en el estadio, sino también en el campo de entrenamiento.

Algunas de las funciones que la aplicación ofrece son: seguimiento de todos los jugadores y la pelota en tiempo real, las trayectorias de los jugadores, la velocidad en tiempo real y la distancia, visión general del rendimiento. La ventana táctica proporciona información sobre cómo ejecutar el rendimiento y el comportamiento táctico del equipo. Por ejemplo: los jugadores pueden estar vinculadas entre sí o con los rivales que muestran la distancia entre ellos. Permite elegir el punto de vista de todas las animaciones, lo que proporciona una idea de cómo son los hechos sobre el terreno de juego.

Esta ventana puede ser equipada con varios diagramas elegidos independientemente. Además de la velocidad en vivo de un jugador, la distancia recorrida y los sectores de velocidad, los diagramas también muestran valores de equipo y comparaciones de jugadores.

⁵<http://www.cairos.com/unternehmen/vistrack.php>

Goles, disparos a gol, tiros de esquina, duelos y otras acciones están vinculadas al código de tiempo de vídeo y se pueden mostrar con un clic del ratón. Esto hace posible la creación de un vídeo con las jugadas destacadas. La ventana de estadística proporciona una buena visión general de los datos estadísticos del partido y ofrece la posibilidad de comparar jugadores.

2.4.4. Stadionus

Stadionus⁶ es un programa analítico dirigido esencialmente a entrenadores, jugadores, agrupaciones deportivas, periodistas y sitios web de fútbol. Proporciona datos estadísticos detallados para su análisis, además de dar la posibilidad de crear modelos realistas en 3D de los partidos. Sin embargo, también es una solución interesante para quienes simplemente disfrutan de ver partidos de fútbol, porque muchas de las funciones del programa son gratuitas.

Para su correcto funcionamiento debe cargarse el vídeo de origen y calibrar las cámaras para que el modelo del partido sea preciso y realista. Gracias a la herramienta de calibración de Stadionus, es muy sencillo. Cuando se introduce la información sobre los jugadores y el partido, el programa calcula el resto. Como resultado, se consigue un modelo animado en 3D, además de las estadísticas completas del juego.

Las estadísticas consisten en un informe detallado del partido que se cargó, gráficos de la velocidad y la aceleración de los jugadores, tablas comparativas de la precisión de los pases, gráficos dinámicos, etc. En general, se puede comparar distintos aspectos del rendimiento de los equipos y de los jugadores (goles marcados, pases, faltas, eventos, táctica, etc.). Aunque se proporcionan grandes cantidades de datos, se presentan de forma clara y cómoda. Además, es posible crear informes para imprimir, con gran cantidad de detalles.

Cabe destacar que los modelos 3D de partidos de fútbol ofrecen grandes oportunidades y ventajas. Es posible cambiar la posición de las cámaras virtuales para ver el juego desde ángulos que en la realidad son imposibles, como el del balón, el del jugador, etc. Además, puede que sea útil para los entrenadores de fútbol de niños y jóvenes. Valorarán que el análisis del juego se realice gracias a los efectos aplicados al modelo en 3D como, entre otros, la trayectoria del balón o la línea de pase.

Gracias a sus muchas funciones, se puede crear vídeos con los ángulos de cámara más favorables, primeros planos, *zoom* y otro tipo de efectos. También es posible crear archivos de vídeo para verlos con gafas estereoscópicas.

⁶<http://www.stadionus.com>

2.4.5. Kizanaro Sport Technology

Este sistema⁷ brinda información objetiva que permite al cuerpo técnico analizar y evaluar el desempeño de un equipo en base a resúmenes de vídeo. Los análisis objetivos de Kizanaro se realizan a partir de los vídeos de los partidos jugados. Una vez terminado el partido y recibido el material audiovisual, se realiza el proceso de análisis objetivo de un equipo mediante tecnologías desarrolladas por la empresa.

Entre 24 y 48 horas después, el técnico recibe en su computadora un informe impreso con información objetiva del partido sobre el equipo analizado, y un archivo para visualizar el análisis en base a vídeo en el software K-Studio Professional. Este software clasifica y desglosa todas las acciones del partido, segundo por segundo, y permite visualizarlas en el vídeo a través de un click.

Las características del software son las siguientes: visualización de análisis de los partidos enviados por Kizanaro, clasificación de las acciones del partido en 150 categorías personalizadas, hasta 2500 eventos respaldados por el vídeo, visualización en pantalla completa y personalización de las consultas por acción de juego, sector de campo, intervalo de tiempo y jugador.

En cuanto a los clientes que se sirven de esta tecnología para identificar fortalezas y debilidades del equipo rival y del propio destacan selecciones nacionales de fútbol como la de Colombia, Argentina, Arabia Saudí, etc.

2.4.6. Nacsport

Desde 2008, Nacsport⁸ desarrolla y comercializa software de vídeo análisis del rendimiento deportivo para todos los niveles profesionales, desde organizaciones de élite a técnicos de base, con más de 4.000 licencias de sus programas vendidas en todo el mundo.

A partir del rendimiento en competición, descubre el comportamiento individual o colectivo del deportista sin tener que recurrir a herramientas de edición de vídeo. Proporcionan datos objetivos del rendimiento, tanto cuantitativos como cualitativos, ordenados según las necesidades de análisis para extraer los que más interesen.

Permite comparar diferentes acciones del juego, dibujar sobre ellas y un múltiples de herramientas para hacer el trabajo de análisis. Los programas han sido concebidos para adecuarse a todos los niveles profesionales, desde la élite hasta la base.

Entre los clientes de esta tecnología destacan clubes de fútbol de primer nivel de España y el Reino Unido, selecciones de baloncesto y de fútbol sala o equipos de

⁷<http://kizanaro.com/web/>

⁸<http://www.nacsport.com/es/>

hockey.

2.5. Conclusiones

Durante este capítulo se ha realizado un estudio del estado del arte del análisis de vídeos deportivos. Para ello, en primer lugar, se ha presentado el análisis de este tipo de vídeos y su aplicación e importancia en la actualidad (sección 2.1). Seguidamente, se han analizado, las distintas etapas comunes en estas técnicas en los trabajos del estado del arte, así como las características de cada una de ellas (sección 2.2). Posteriormente, se han resumido los *datasets* públicos sobre los que se va a trabajar (sección 2.3), así como diversos productos comerciales con sus características y utilidades (sección 2.4).

En la sección 2.1 se han presentado diversas categorizaciones realizadas en el estado del arte del análisis del vídeo deportivo. Se ha expuesto una clasificación en función de los principales modos de vídeos de eventos deportivos existentes. También se han presentado los diferentes enfoques y métodos de diseño de los algoritmos de detección de eventos, así como los diversos atributos del vídeo, incluyendo su estructura, conocimientos a priori y otras propiedades del contenido que se usan con tales fines. Por último, se han expuesto exhaustivamente algunas de las aplicaciones en las que es de gran utilidad el procesamiento automático de contenidos en vídeos deportivos.

En la sección 2.2 se ha sintetizado la información más relevante sobre las diversas técnicas y métodos (comunes en el estado del arte) para obtener la información deseada en el procesamiento del vídeo deportivo, dándole una orientación más remarcada hacia vídeos de fútbol. Poniendo en conjunto el análisis realizado, las etapas comunes del estado del arte y sus objetivos son:

- La extracción de fondo que permite realizar un posterior análisis del frente para determinar los blobs que identifican a los jugadores en el campo.
- La clasificación de esos blobs en jugadores, pelota u otros elementos de etiquetados.
- El seguimiento de cada jugador, y tras un proceso de fusión, la obtención de las trayectorias de los integrantes del evento.

En la sección 2.3 se han descrito brevemente los *datasets* públicos utilizados en este proyecto. No tanto con el objetivo de su orientación a una evaluación como se hizo en [1] al diseñar el prototipo, sino para que se conozcan los datos sobre los que se van a presentar las aplicaciones y mejoras realizadas.

Por último, en la sección 2.4 se han estudiado diversas aplicaciones y productos comerciales presentes en el mercado, analizando sus características y las posibilidades que ofrecen a los usuarios potenciales. En todas ellas se observan dos aspectos fundamentales:

- La aplicación presenta los datos con la mejor calidad posible, no permitiendo realizar grandes ajustes al entorno en el que se vaya a utilizar (a nivel de software).
- Una gran automatización y herramientas muy poco supervisables.

La herramienta aquí presentada permitirá solucionar ambos aspectos, permitiendo un ajuste al entorno de aplicación (se posee control total del algoritmo), y una corrección de errores *online* en caso de que la aplicación esté funcionando de manera incorrecta o presente errores puntuales.

Capítulo 3

Sistema base.

En este capítulo se describe el sistema base, punto de partida para el desarrollo del prototipo original sobre el que se desarrollarán las aplicaciones del proyecto. Todos los contenidos de las secciones aquí descritas se utilizarán en para las aplicaciones de deportes individuales, tenis, y deportes de equipo, fútbol.

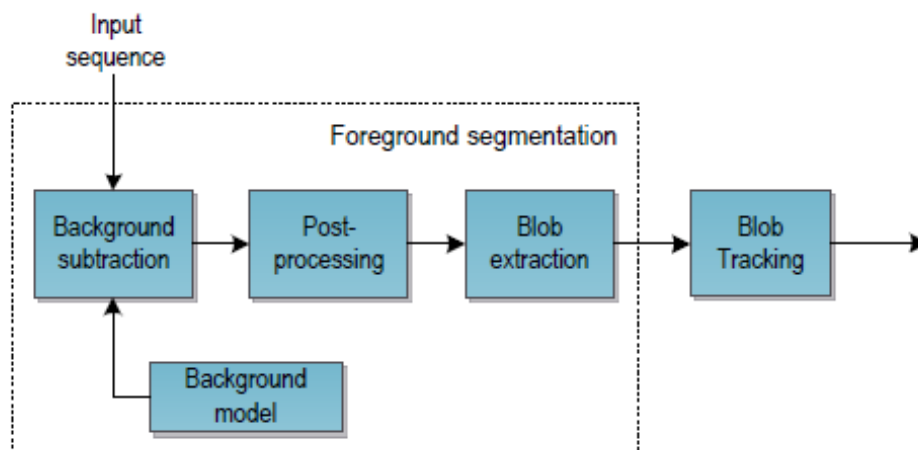


Figura 3.1: Diagrama de bloques del sistema base. Fuente [1]

3.1. Descripción general del sistema base

El sistema base parte de un sistema de análisis de vídeo para detección de objetos robados o abandonados [33]. Este sistema estaba diseñado para funcionar como parte de un sistema de vídeo vigilancia capaz de activar alarmas para eventos detectados en tiempo real. Este requisito impone límites de tiempo a la complejidad de los algoritmos utilizados en cada uno de los módulos de análisis. El diagrama de bloques del sistema

original se muestra en la Figura 3.1.

Después de la adquisición del *frame*, en el módulo de segmentación de frente (*Foreground Segmentation Module*) se genera una máscara de frente para cada *frame* entrante. Esta máscara de frente consiste en una imagen binaria que identifica los píxeles que no pertenecen al fondo, bien sean objetos en movimiento o estacionarios. Las técnicas de pos-procesamiento se aplican en este punto sobre esta máscara con el objetivo de eliminar ruidos y sombras. Después de ello, el módulo de extracción de blob (*Blob Extraction Module*) determina las componentes conexas de la máscara de frente y las extraer. En la etapa siguiente, el módulo de seguimiento de blob (*Blob Tracking Module*) intenta asociar un identificador único para cada blob extraído a través de la secuencia de *frames*.

El sistema base se describe con mayor detalle en el Anexo B. Este anexo se ha extraído de [33] y está basado en el sistema descrito en [34].

Capítulo 4

Aplicación deportes individuales.

En este capítulo se va a describir la primera de las aplicaciones realizada. Se trata de la relacionada con el análisis de contenido en vídeos deportivos para deportes individuales, más concretamente para el tenis.

La aplicación ha sido desarrollada sobre la base del prototipo completo presentado en [1]. Dicho prototipo partía de un sistema base expuesto en la sección 3, y tiene un esquema completo de funcionamiento descrito a continuación (sección 4.1).

Puesto que el proceso de fusión de las salidas de los algoritmos de seguimiento de cada cámara individual al espacio común difiere notablemente entre los tipos de eventos deportivos a analizar, se va a presentar un resumen de la fusión llevada a cabo en los sistemas para deportes individuales (sección 4.2).

Posteriormente, fue necesario llevar a cabo unos ajustes del prototipo original para garantizar su funcionamiento en el entorno de aplicación (sección 4.3). Algunos ajustes fueron de carácter paramétrico, mientras que otros fueron orientados a la implementación y homogeneización de los lenguajes de programación de los distintos módulos, tal y como se buscaba en uno de los objetivos del proyecto.

Una vez presentados los ajustes realizados, se va a pasar a presentar la interfaz de aplicación desarrollada (sección 4.4). Con ella se da satisfacción en este tipo de vídeos a los otros dos objetivos fundamentales del proyecto: por una parte, la de proveer de una herramienta para que interaccionen con el algoritmo usuarios no expertos y puedan extraer la información relevante suministrada por el prototipo; y por otra, la posibilidad de llevar a cabo la corrección de los errores que vaya dando el prototipo de forma *online*.

Por último se cerrará este capítulo con unas breves conclusiones del sistema realizado (sección 4.5).

4.1. Prototipo original - deportes individuales

A continuación, se va a esbozar el prototipo original del que se partía al comienzo de este proyecto. En la Figura 4.1 se presenta un esquema general del mismo.

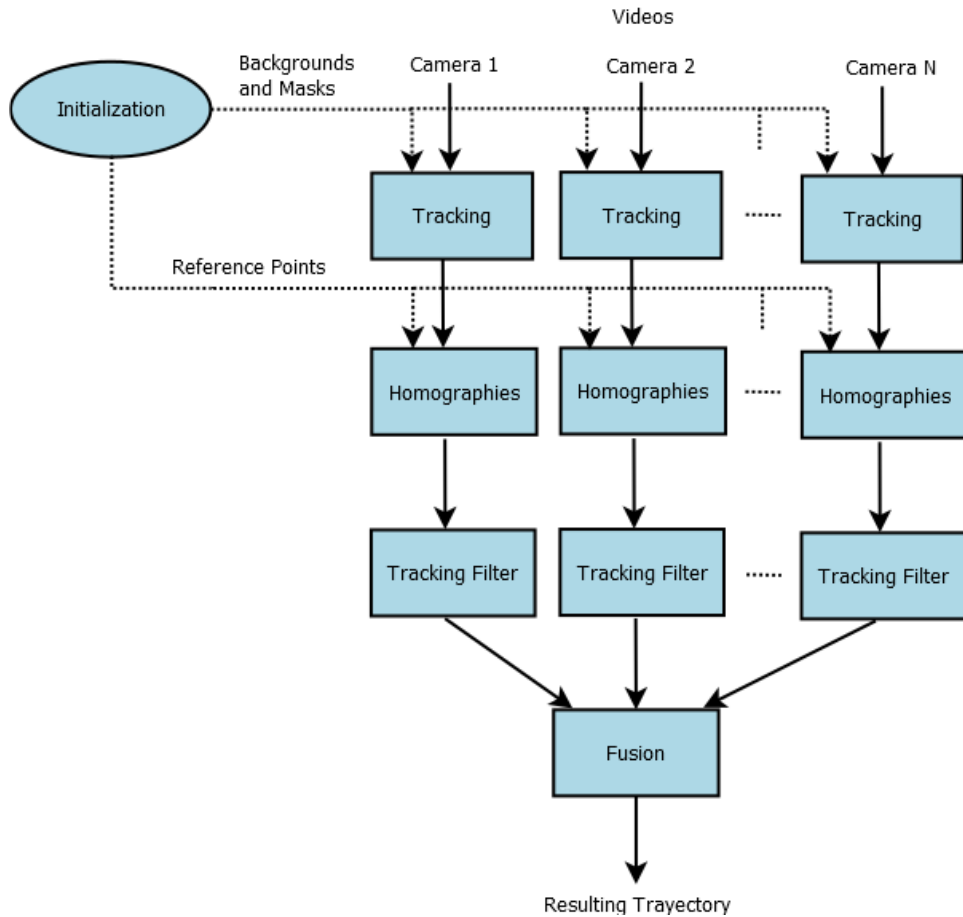


Figura 4.1: Diagrama de bloques del prototipo para deportes. Fuente [1]

Puesto que se trata de un sistema para deportes individuales, es decir, a priori se puede gestionar cada jugador de manera independiente, el diagrama de bloques presentado es individual para cada jugador. Es asumido, por lo tanto, que no se producirán interacciones de otros objetivos con el objetivo principal. De hecho, los vídeos utilizados del *dataset* son los que cubren sólo las áreas de cada jugador, no incluyéndose las vistas globales que sí mostrarían a ambos jugadores simultáneamente. De igual modo, para poder evaluar a cada jugador, serán necesarios sistemas paralelos. Esta condición es necesaria para las características del bloque de la fusión. Destacar que, al tratarse de tenis, solo será necesario gestionar dos usuarios en paralelo. El proceso bloque a bloque, y de acuerdo a la guía del diagrama, se resume como sigue:

- Bloque de inicialización: se ejecuta en primer lugar. Se genera la representación del terreno de juego, los fondos de escena, las máscaras para excluir zonas de no interés, y los puntos de referencia para la realización de las homografías. El método de generación de cada uno de estos elementos se detalla en el apartado correspondiente del sistema base incluido en el Anexo B.
- Bloque de seguimiento: recibe los vídeos y las máscaras de las diferentes cámaras que graban a cada jugador y genera el seguimiento desde su propia perspectiva.
- Bloque de homografías: recibe el seguimiento del jugador de cada cámara y los puntos de referencia, y genera la vista superior de seguimiento de cada cámara en el plano común.
- Bloque de filtrado del seguimiento: recibe y filtra los resultados de la etapa de seguimiento en base a la información trasladada a la vista superior de cada cámara. Algunos ejemplos de los criterios de filtrado son: posibles regiones de no interés en el campo, número mínimo de *frames* para el blob, blobs distantes no pertenecen al jugador, etc.

Por último, quedaría el Bloque de fusión, que crea la trayectoria general a lo largo de todo el campo. El método de fusión se describe en la siguiente sección de manera más detallada dada su relevancia.

4.2. Etapa de fusión

Para deportes individuales, el proceso de fusión es relativamente simple. La ventaja principal es que cada blob en el *frame* t pertenece al mismo jugador.

$$x_t = \frac{1}{N} \sum_{i=1}^N x_{i,t} \quad y_t = \frac{1}{N} \sum_{i=1}^N y_{i,t}$$

Donde (x_t, y_t) son las coordenadas resultantes de la fusión de todas las cámaras que graban a un jugador en el *frame* t ; $(x_{(i,t)}, y_{(i,t)})$ son las coordenadas en el *frame* t de la cámara i , e $i = 1 \dots N$, siendo N el número de cámaras que graban un blob correspondiente al jugador en el *frame* t .

Además, se puede añadir un peso específico a cada coordenada del blob en función de la posición, altura y la precisión de cada cámara.

4.3. Ajustes al prototipo inicial

4.3.1. Modificaciones paramétricas

Para el procesado se han utilizado los vídeos del *dataset* 3DLife ACM Multimedia Grand Challenge 2010 (véase la sección 2.3.1). En el prototipo sobre el que se basa el sistema implementado, sólo se utilizaban los vídeos de las cámaras 1, 3 y 4, que graban a un determinado jugador (jugador azul) durante un juego concreto (juego 1).

Se observa que los otros vídeos tienen problemas durante el seguimiento debido a diversas causas que se pueden resumir en: - La red que separa los campos se mueve por el viento produciendo falsas detecciones.

- La cámara se enfoca automáticamente y se producen desenfoques, dificultando notablemente la tarea del seguimiento.
- Las cámaras son ligeramente movidas por el viento, invalidando en gran medida el modelo de fondo generado inicialmente.

Para solucionar esos problemas, que fueron detectados tras un exhaustivo estudio del estado del prototipo original, se observa que se necesita un ajuste de parámetros del algoritmo al escenario de aplicación.

El principal problema que se trata de paliar es la influencia que tiene el ruido ocasionado en los vídeos, por los problemas anteriormente mencionados. Para ello, se modifica el valor que los siguientes parámetros, ya presentes en el prototipo original:

- **VARNOISE**: Umbral que controla la sensibilidad del detector ante la variación de ruido. Cuanto más alto sea este valor del umbral, más baja es su sensibilidad y, al contrario, cuanto más bajo sea este umbral, más alta es la sensibilidad. Por esto, se modifica el valor de partida que era 8.0 a un valor de 15.0 para reducir su sensibilidad.
- **TH_MOTION**: Es un parámetro para el cálculo de la función Gamma. Se reduce el valor de este umbral desde el valor de partida $9.99 \text{ e-}5$ a un valor de $11 \text{ e-}11$ para de esta forma, reducir su sensibilidad a ligeros movimientos de la cámara.

Para la estimación de estos valores se llevaron a cabo una serie de barridos paramétricos siguiendo la estrategia *rough-to-fine*. La evaluación en todo momento fue cualitativa, sin embargo, no es objetivo principal de este proyecto mejorar el rendimiento cuantitativo del prototipo. Pese a ello, si se pretendía que ajustando el algoritmo al escenario se solucionasen los problemas con las secuencias no incluidas en la evaluación original del prototipo. Destacar que con estas modificaciones ya se

podieron utilizar también las cámaras de este mismo *dataset* que graban al segundo jugador (jugador rojo) y cuyas vistas corresponden con las cámaras 5, 6 y 8. Se logra también que se puedan usar las secuencias correspondientes al juego 2 del partido.

Otros parámetros o métricas definidas empíricamente se encuentran en el bloque que filtra el seguimiento. Los criterios establecidos para el filtrado son:

- Mínimo de 10 *frames* por blob para considerarlo.
- Región no analizada en el campo de aproximadamente 70 cm de distancia con la red. Esta distancia se aplica para filtrar las detecciones "fantasma" causadas por las deformaciones de la red durante el juego. Este ajuste es debido a que la red no está tensada por la parte de abajo. En caso de redes de competición reglamentarias, este ajuste no es necesario.

En cuanto al bloque de fusión, la trayectoria fusionada empieza en el *frame* 25, comprobado mediante evaluación cualitativa, que ese tiempo era suficiente para permitir la inicialización del sistema de seguimiento al completo.

4.3.2. Modificaciones de implementación

Se ha modificado la carga de la información de inicialización, así como de los datos de análisis. Si bien anteriormente esto se realizaba de manera estática en el código del prototipo, en el sistema actual se cargan unos ficheros de configuración de donde se obtienen la información sobre los directorios donde se encuentran los vídeos que se deben procesar, así como los datos para generar los archivos y vídeos de salida tras el proceso.

Cada fila del archivo de configuración contiene los datos de inicialización de cada una de las cámaras del *dataset*. De izquierda a derecha y separados por un espacio en blanco contienen la siguiente información:

- Directorio del vídeo a procesar.
- Vídeo de salida tras el procesado.
- Fichero de texto de salida de posiciones del jugador por *frame*.
- Máscara de procesamiento.
- *Frames* por segundo.

El módulo de seguimiento utilizado es el del sistema base que se describe en el Anexo B, con las modificaciones descritas en las siguientes secciones de dicho anexo. Destaca

que el tiempo de ejecución del prototipo es de 21,6 FPS para los vídeos de tenis, lo que se considera en el estado del arte, tiempo real. Además, la ejecución se ha realizado en un PC muy limitado (procesador Core2, 3GB de RAM, 2.93GHz), por lo que si se ejecuta la aplicación en un ordenador más potente, los 25 FPS se superarán de forma sencilla.

Para que el módulo diseñado para calcular la homografía funcione correctamente, se recibe un fichero de configuración con los puntos de referencia que se calculan y se introducen manualmente en el sistema. El procedimiento utilizado es:

- Se obtiene el fondo.
- Se representan y se seleccionan los puntos de referencia con el cursor de datos
- Se anotan para aplicarse más adelante en las homografías.

El fichero de configuración con los puntos de referencia que debe recibir este bloque, y cuya forma de recibirlo será descrita en la siguiente sección, debe tener una estructura fija:

- A nivel de fila: cada fila del archivo de texto contiene las coordenadas x e y de los puntos de referencia separadas por un espacio en blanco.
- A nivel general: las primeras cuatro filas son las coordenadas de referencia en píxeles correspondientes a la cámara 1 y las siguientes cuatro filas las coordenadas en metros que corresponden a la misma cámara. Esta estructura se repite para las coordenadas de referencia de las cámaras 2, 4, 5, 6 y 8.

Por último, otra de las modificaciones realizadas al prototipo original, y también una de las más relevantes, ha sido la homogeneización del lenguaje. El sistema de seguimiento se encuentra en lenguaje C++ y para ello se han convertido los módulos de fusión y de homografía que se encontraban disponibles en *scripts* de Matlab a lenguaje C++.

4.4. Interfaz gráfica de usuario

Una vez se han hecho las modificaciones necesarias para intentar solucionar las carencias observadas en el prototipo original para su uso en análisis de vídeos de deportes individuales, se ha llevado a cabo una interfaz de usuario que permite satisfacer dos de los objetivos de este proyecto, es decir, la creación y desarrollo de una interfaz gráfica de usuario que permita su uso a un usuario no experto, y la posibilidad de

realizar correcciones de los posibles errores de detección y/o seguimiento del jugador de manera *online*.

Esta interfaz consta de dos ventanas, una principal llamada *Aplicacion* en la que se llevan a cabo los procesamientos de los vídeos y se muestra la localización del jugador en cada *frame*, y otra derivada de la principal y que se llama *Correction*, que tiene como objetivo permitir al usuario corregir los resultados del algoritmo en caso de que estos se observen como erróneos. En las siguientes secciones se describirán en mayor profundidad ambas ventanas. El Anexo A.1 presenta un manual de usuario de la interfaz gráfica para deportes individuales.

4.4.1. Ventana principal - *Aplicacion*

La ventana principal de la interfaz gráfica de usuario para deportes individuales tiene el aspecto que se puede apreciar en la Figura 4.2.

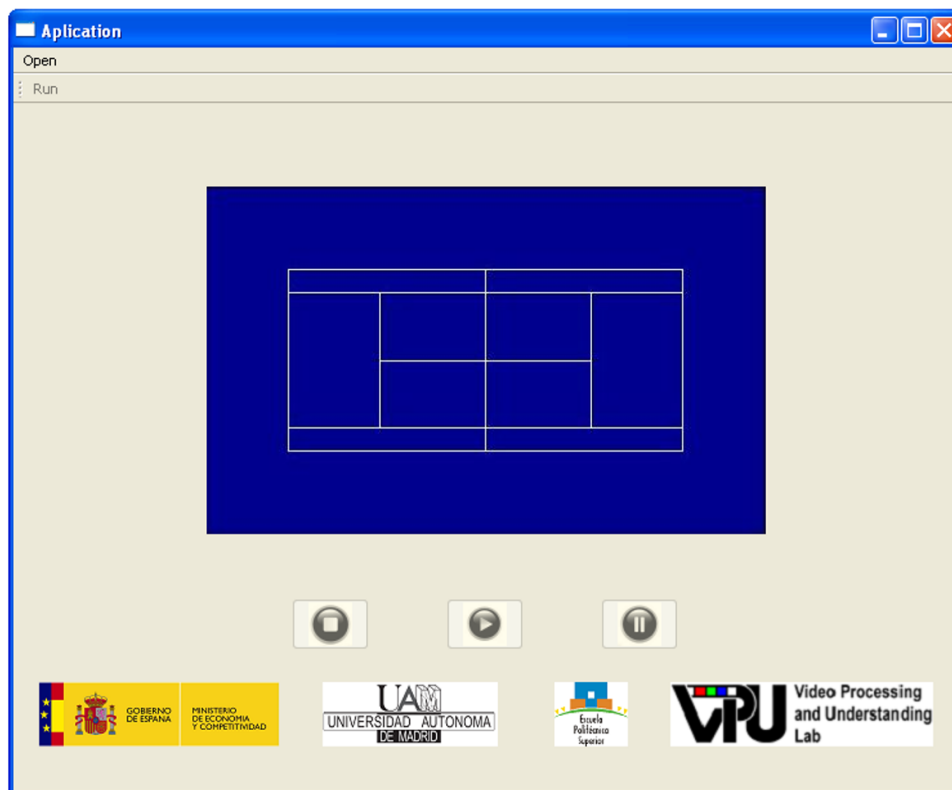


Figura 4.2: Vista de la ventana principal.

En la imagen anterior se aprecia como sólo se encuentra habilitada la opción Open (extremo superior izquierdo), la cual contiene un desplegable con dos opciones: *Configuration file* y *Homography file*.

- *Configuration file*: permite seleccionar el fichero de texto que contiene la información inicial necesaria para la correcta ejecución del sistema.
- *Homography file*: permite seleccionar el archivo de texto con los puntos de referencia para la posterior realización por el bloque de homografías de la misma. Las estructuras de ambos ficheros se encuentran descritas en el Anexo B. Si la selección se ha hecho de la forma correcta, se habilita la opción *Run* (justo debajo de la anterior), que es necesario pulsar para habilitar el botón *Play* e iniciar la ejecución.

Todas estas opciones disponibles y botones se pueden ver descritos con mayor profundidad en el Manual de Usuario que se ha creado en el Anexo A.1.

Una vez pulsado el botón *Play* se inicia la ejecución de los vídeos de las seis cámaras. Un ejemplo del procesado de los vídeos se puede ver en la Figura 4.3.

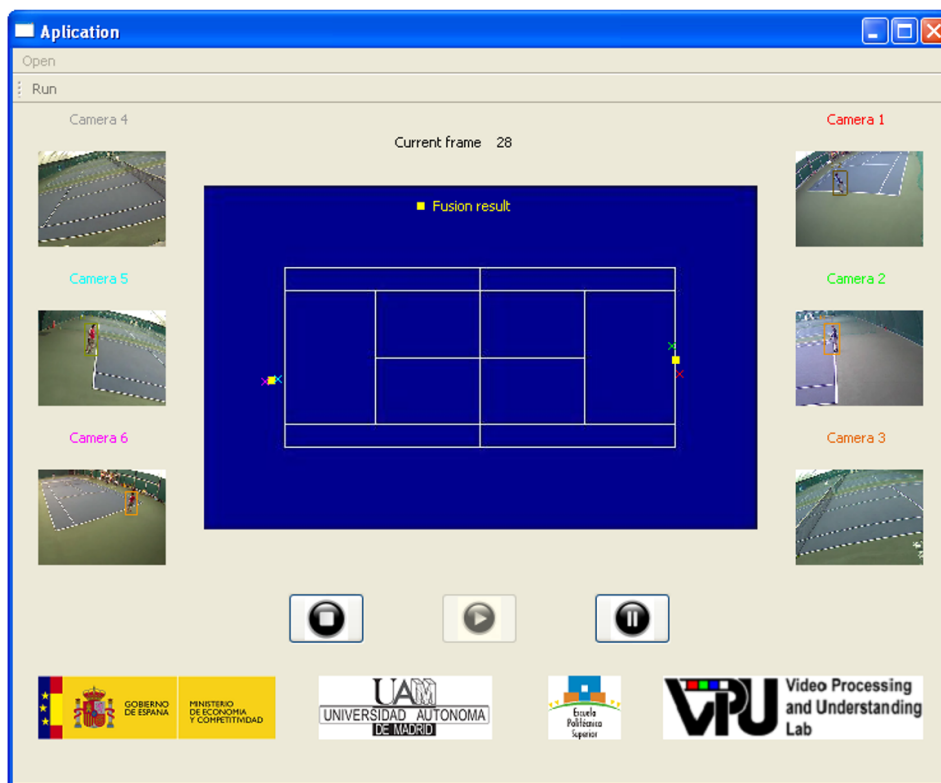


Figura 4.3: Vista de un momento determinado de la ejecución

En la imagen anterior se aprecia en el centro - parte superior el número de *frame* que se encuentra en ejecución en ese momento (*Current frame*) y, tanto a la izquierda como a la derecha de la representación del terreno de juego, se encuentran las visuali-

zaciones de las cámaras. Las de la derecha corresponden con las cámaras que graban al jugador número 1 y las de la izquierda con las que graban al jugador número 2. La relación del número de cámara que aparece en la interfaz con la del *dataset* utilizado es la siguiente:

- Cámara 1 de la interfaz es la cámara 1 del *dataset*.
- Cámara 2 de la interfaz es la cámara 3 del *dataset*.
- Cámara 3 de la interfaz es la cámara 4 del *dataset*.
- Cámara 4 de la interfaz es la cámara 5 del *dataset*.
- Cámara 5 de la interfaz es la cámara 6 del *dataset*.
- Cámara 6 de la interfaz es la cámara 8 del *dataset*.

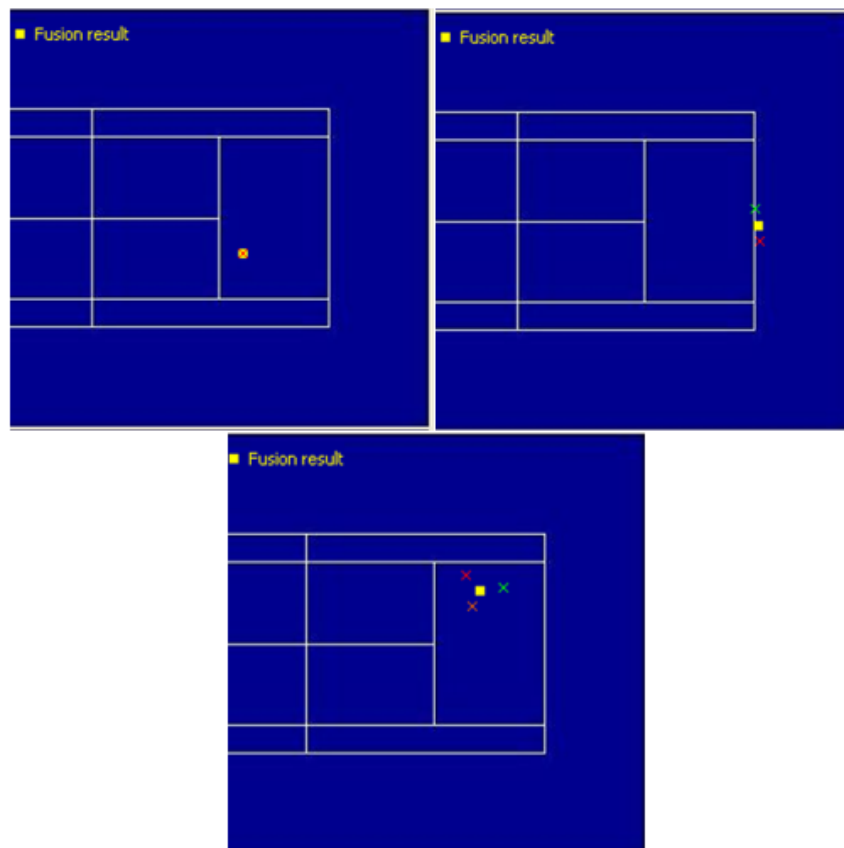


Figura 4.4: Representación de la posición del jugador en función del número de cámaras que le graban en ese momento. De izquierda a derecha y de arriba abajo, resultados con una, dos y tres cámaras.

En el centro de esta ventana principal se encuentra la representación del terreno de juego con la leyenda *Fusion result* en color amarillo, lo que indica que los puntos de color amarillo en la representación corresponden al punto de coordenadas obtenido en el proceso de fusión de las tres cámaras que graban a cada uno de los jugadores. Las coordenadas propias de cada cámara se representan con un punto de un color igual al que tiene la leyenda de cada una de las cámaras individuales. Algunos ejemplos de esto aparecen en la Figura 4.4.

Para que aparezca la ventana de corrección de errores (*Correction*) se precisa que el usuario haga clic sobre la imagen de la cámara en la que detecte un error o posible error en la detección o seguimiento del jugador.

4.4.2. Ventana de corrección de errores - *Correction*

La ventana de corrección de errores de la interfaz gráfica de usuario para deportes individuales tiene el aspecto que se muestra en la Figura 4.5.

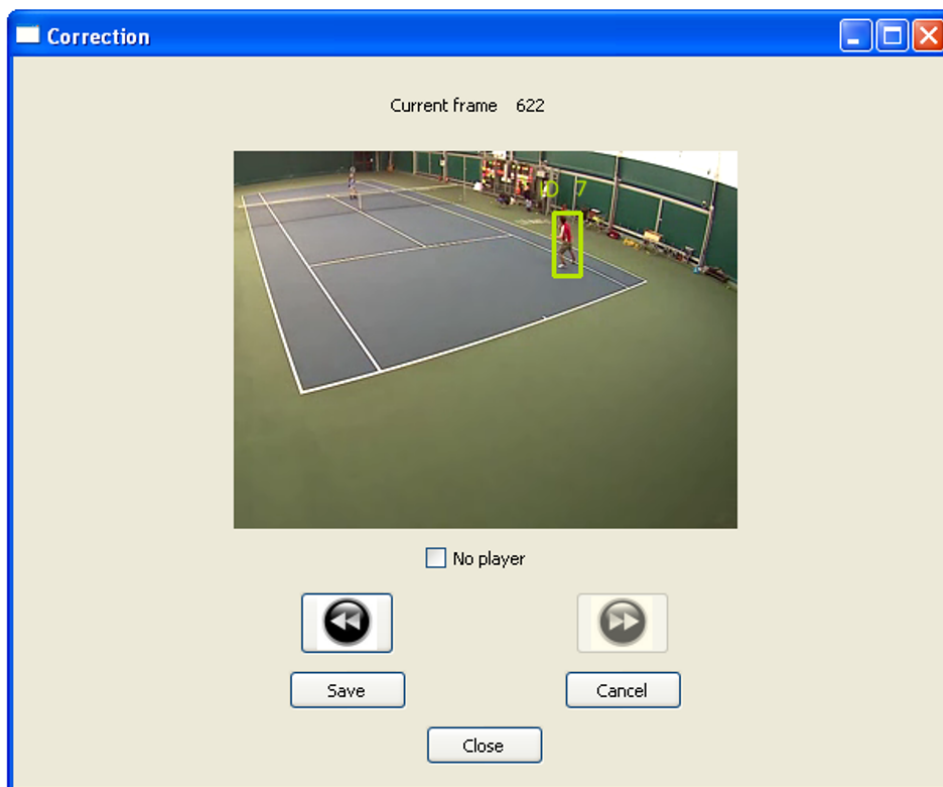


Figura 4.5: Vista de la ventana de corrección de errores.

En la figura anterior se muestra un *frame* sobre el que se pretende llevar a cabo la

corrección del error observado, así como el número de *frame* (*Current frame*). Se presentan dos botones que permiten navegar entre *frames*. Se asume que el usuario tarda un tiempo en reaccionar ante la observación del error, por lo que se le proporciona un *batch* de 50 *frames* para que navegue hasta el origen del error.

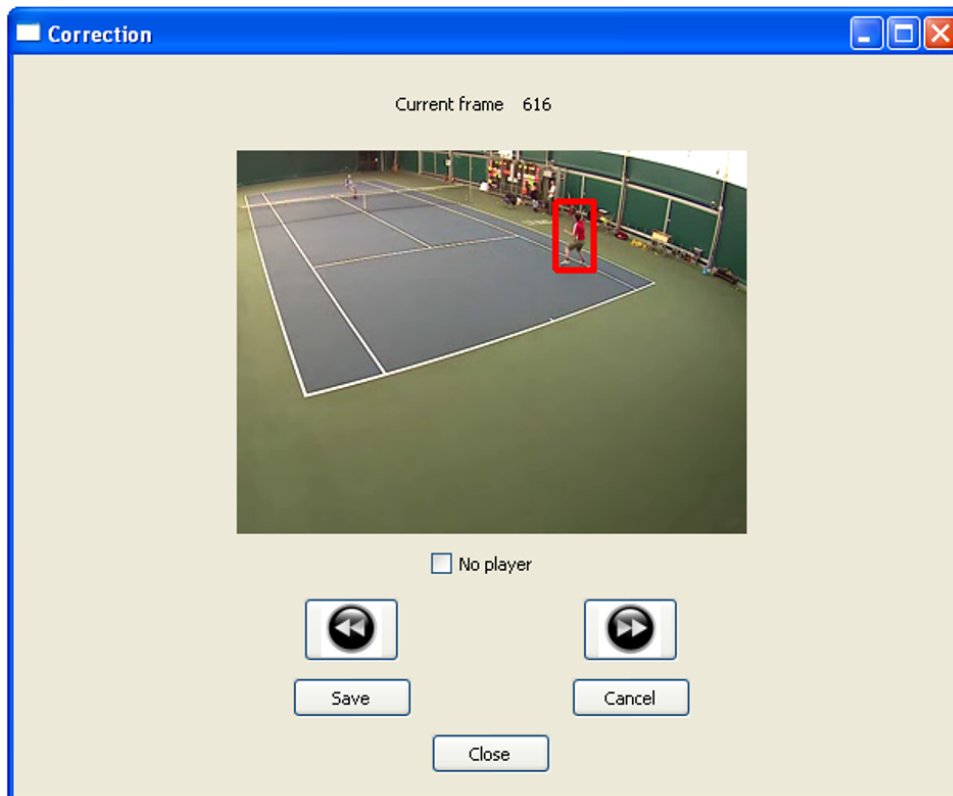


Figura 4.6: Generación del nuevo *bounding box*

Para rectificar el error observado que no ha sido posible filtrar mediante la adaptación de parámetros descritos en la sección anterior, se dan dos posibilidades:

- Errores de falsos positivos, esto es, que algún objeto o persona externa al evento deportivo, p.ej. el público, se detecte como un jugador. El usuario puede interactuar con la interfaz y eliminar esa detección. Para ello sirven los botones *No player* y *Save*. El primero elimina la detección y el segundo confirma y almacena dicha información.
- Errores de falsos negativos o fallos de precisión, lo que se define para esta aplicación como jugadores que no han sido detectados o no han sido seguidos de manera correcta. En esta situación, se permite al usuario encuadrar al jugador perdido dentro de un *bounding box* de manera manual. Para ello, la interfaz per-

mite dibujar sobre la imagen, mediante la técnica de *click&drag*, un rectángulo que encuadre al objetivo, y después salvarlo para continuar con la ejecución normal. La corrección de un error de seguimiento sería similar pero eliminando el *bounding box* erróneo antes de incluir el correcto. Un ejemplo de esta funcionalidad se ha incluido en la Figura 4.6.

Una vez que se guardan los cambios, el sistema procede a realizar una nueva fusión para ese *frame* utilizando los nuevos datos. Los datos de esta nueva fusión se sobrescriben los anteriores, y será a partir de ellos sobre los que se extraigan los resultados que arroja el prototipo en forma de estadísticos como distancia recorrida, representación de trayectorias, etc.

4.5. Conclusiones

Tras la inclusión de todas las modificaciones y las interfaces gráficas, el funcionamiento del sistema cumple con todos los objetivos propuesto y satisface las motivaciones del proyecto. Es necesario reseñar, que la tarea de análisis, deportes individuales, presentaba pocos retos en cuanto a complejidad, y por lo tanto es difícil apreciar en gran medida la utilidad de los cambios realizados y las nuevas funcionalidades incluidas.

Sin embargo, algunas de los aspectos en los que se han visto reflejadas las mejoras que han sido:

- Errores por mal alineamiento de las cámaras que daban lugar a pequeños errores en la estimación de la posición del jugador en el plano común.
- Distorsión en las proyecciones de la posición del jugador al plano común porque el objetivo se aproxime físicamente a una cámara en exceso.
- Puntos ciegos de algunas cámaras suponen cambios bruscos de posición, pues se deja de usar información de una de ellas y se re-estima la posición con la información restante.

Todos estos errores han sido resueltos, mediante la interfaz gráfica, el ajuste de parámetros y las interacciones manuales, permitiendo un funcionamiento en tiempo real, se introduce un retraso por el tiempo que le lleve al usuario realizar las correcciones manuales.

Con estos datos, se puede concluir, que no se han incidido en las capacidades originales del algoritmo, de ejecución en tiempo real, y se han mejorado aquellos aspectos que motivaron el trabajo realizado. No obstante, estas mejoras quedarán

más patentes en la siguiente sección, cuando se presenten sobre una tarea de mayor complejidad, como lo es la de deportes colectivos, en concreto el fútbol.

Destacar también, que la interfaz cumple con los requisitos de poder ser utilizada por un usuario no experto, y se le ha dotado de funcionalidad que, pese a no haber sido incluida en esta descripción, si se encuentra en el manual de usuario. Esta funcionalidad es básicamente a nivel de interacción: selección de cámaras, pausas de ejecución, salidas a representar, y otras funcionalidades que hagan más sencilla la tarea del usuario final de la aplicación.

Capítulo 5

Aplicación deportes de equipo

En este capítulo se va a describir la segunda de las aplicaciones realizada. Se trata de la relacionada con el análisis de contenido en vídeos deportivos para deportes colectivos, más concretamente para el fútbol.

La aplicación ha sido desarrollada de nuevo sobre la base del prototipo completo presentado en [1]. Dicho prototipo partía de un sistema base expuesto en la sección 3, y tiene un esquema completo de funcionamiento descrito a continuación (sección 5.1).

Este segundo sistema se centra en los deportes de equipo, entendidos como aquellos deportes en los que dos o más jugadores se pueden mover por las mismas áreas del campo e incluso los árbitros pueden entrar en esas áreas y moverse ente los jugadores. En este caso, la fusión es más compleja, porque todos los blobs de seguimiento no pertenecen al mismo jugador como en el caso de deportes individuales. Durante el seguimiento, se pueden producir oclusiones, causando problemas en el seguimiento de los jugadores como la pérdida o la generación de blobs fusionados de más de un jugador. Se ha incluido por lo tanto una sección en este capítulo detallando ésta etapa (sección 5.2).

Posteriormente, fue necesario llevar a cabo unos ajustes del prototipo original para garantizar su funcionamiento en el entorno de aplicación (sección 5.3) al igual que se hizo con los deportes individuales. Se realizó igualmente la homogeneización de los lenguajes de programación de los distintos módulos, tal y como se buscaba en uno de los objetivos del proyecto.

Una vez presentados los ajustes realizados, se va a pasar a presentar la interfaz de aplicación desarrollada (sección 5.4). Con ella se da satisfacción en este tipo de vídeos a los otros dos objetivos fundamentales del proyecto: por una parte, la de proveer de una herramienta para que interaccionen con el algoritmo usuarios no expertos y

puedan extraer la información relevante suministrada por el prototipo; y por otra, la posibilidad de llevar a cabo la corrección de los errores que vaya dando el prototipo de forma *online*.

Por último se cerrará este capítulo con unas breves conclusiones del sistema realizado (sección 5.5).

5.1. Prototipo original - deportes de equipo

A continuación, se presenta el prototipo original del que se partía al comienzo de este proyecto para la tarea de deportes de equipo. En la Figura 5.1 se presenta un esquema general del mismo.

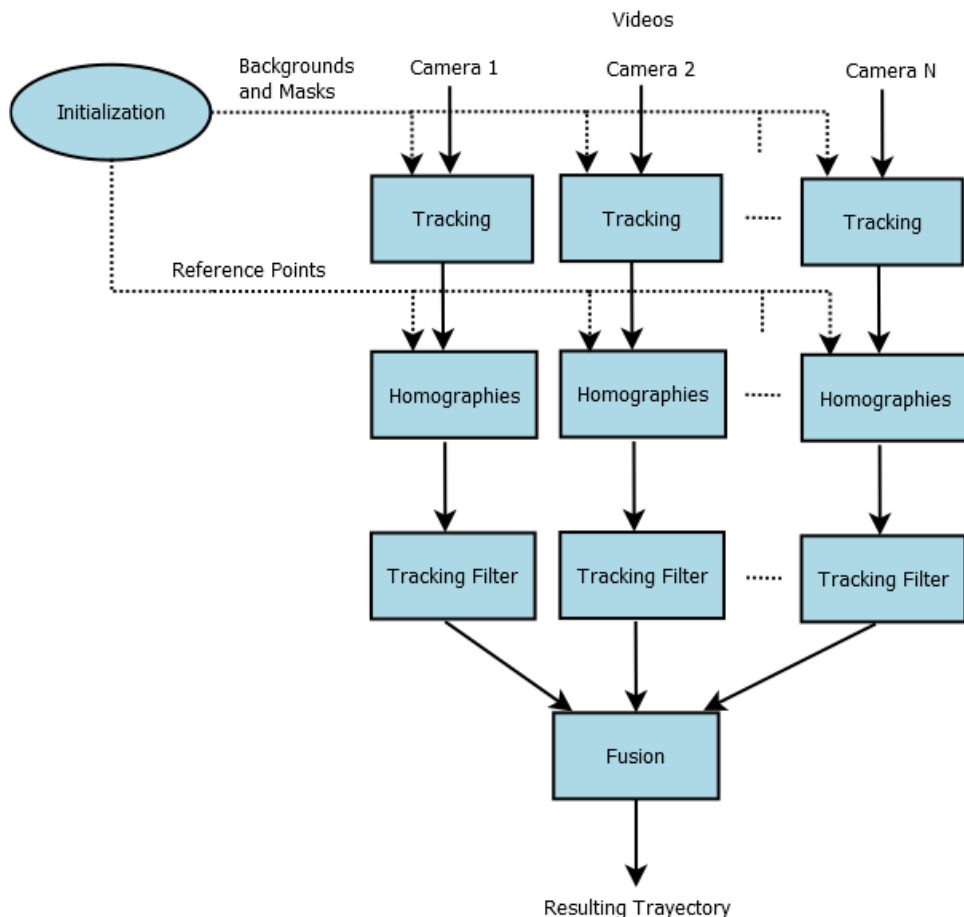


Figura 5.1: Diagrama de bloques del prototipo para deportes. Fuente [1]

El diagrama de bloques del sistema es similar al diagrama de bloques de los deportes individuales, pero con algunas modificaciones que se detallan a continuación:

- Bloque de inicialización: se ejecuta en primer lugar. Se genera la representación del terreno de juego, los fondos, las máscaras, los puntos de referencia para la realización de las homografías y los umbrales para la fusión. El método de generación de cada uno de estos elementos se detalla en el apartado correspondiente y los umbrales para la fusión se definen en la sección siguiente.
- Bloques de seguimiento: reciben los videos, los fondos y las máscaras de las diferentes cámaras que están grabando a los jugadores y generan los datos de seguimiento de las diferentes perspectivas.
- Bloques de homografías: reciben el seguimiento de los jugadores de cada cámara y los puntos de referencia, y se genera la vista superior de seguimiento de cada cámara.
- Bloque de filtrado del seguimiento: recibe y filtra el seguimiento para la vista superior de cada cámara. Algunos ejemplos de los criterios de filtrado son: posibles regiones en el campo, número mínimo de *frames* para el blob, la anchura o la altura máxima del blob. El bloque genera la vista superior de seguimiento filtrada de cada cámara.
- Bloque de fusión: recibe los umbrales, que se definen en la inicialización, un el procesado de cada cámara y genera las trayectorias generales a lo largo de todo el campo. El método de fusión se describe en la siguiente sección.

El método de fusión, al igual que se hizo en la aplicación para deportes individuales, se describe en la siguiente sección de manera más detallada dada su relevancia.

5.2. Etapa de fusión

El prototipo utilizado realiza un proceso de fusión para los deportes de equipo que es más complejo que para los deportes individuales, debido a que todos los blobs no pertenecen al mismo jugador. El proceso realizado se describe a continuación.

En primer lugar, una vez obtenidos los blobs de frente, la información que se requiere para cada uno es la siguiente:

- Número de *frames* en los que aparece.
- *Frame* inicial.
- *Frame* final.
- Coordenadas (x e y) por cada *frame*.

En primer lugar se va a definir el proceso de fusión base, que como se verá depende de un valor de *score*. Posteriormente, en tres subsecciones se van a presentar distintas estrategias para obtener dicho *score*.

5.2.1. Fusión base

La fusión siempre se va a llevar a cabo entre dos blobs de cámaras enfrentadas en este caso (a partir de ahora grupos).

Para decidir que blobs se fusionarán o no, en primer lugar, es necesario definir un umbral. Todas las puntuaciones (*scores*) bajo el umbral indican que ambos blobs pertenecen al mismo jugador. Las puntuaciones, como se ha mencionado anteriormente, se calcula a partir de dos blobs, uno de cada grupo. Ambos grupos tienen un plano común sobre el que se proyectarán los resultados de la fusión de los blobs, en caso de terminar realizándose, o los blobs individualmente, en caso de no realizarse la fusión.

Cuando se obtiene una puntuación (*score*) para cada par de blobs a partir de dos grupos de seguimiento diferentes, el siguiente paso es obtener una lista de las asociaciones de blobs (LOA) que contiene una fila para cada blob de uno de los grupos. Cada fila indica qué blobs en el otro grupo de seguimiento deben ser fusionados con el blob que corresponde a esa fila. No es necesario calcular una lista similar para el otro grupo, ya que es redundante, aunque a veces es útil para simplificar aún más el procesamiento (por ejemplo, en la etapa en la que se calculan los blobs resultantes del proceso de fusión). La Figura 5.2 muestra un ejemplo de esta lista de asociaciones.

	1	2
1	5	15
2	20	0
3	0	0
4	21	0
5	4	19
6	0	0

Figura 5.2: Ejemplo de LOA

La información que se puede extraer de la figura es la siguiente:

- El blob 1 del primer grupo de seguimiento se fusiona con los blobs 5 y 15 del segundo grupo (si hubiera más blobs fusionados, aparecerían más columnas).
- El blob 2 del primer grupo se fusiona con el blob número 20 del segundo grupo de seguimiento.

- El blob 4 del primer grupo de seguimiento se fusiona con el blob número 21 del segundo.
- El blob 5 del primer seguimiento se fusiona con los blobs 4 y 19 del segundo grupo de seguimiento.
- Los blobs 3 y 6 no se fusionan con ningún blob del segundo grupo.

Por último, se procesa la lista de asociaciones para obtener el grupo de los blobs en cada cámara que debe ser fusionado, y se trasladan los resultados al plano común siguiendo un esquema de fusión por pares de blobs asociados similar al que se hacía en el caso de deportes individuales.

En la subsección siguiente se describe la fusión que se han desarrollado en el sistema original para combinar blobs de las distintas cámaras.

5.2.2. Fusión de blobs

Esta fusión extraída del prototipo consiste en dados dos blobs, la puntuación (*score*) se define como la distancia cuadrática media por *frame*, para aquellos *frames* en los que están presentes ambos blobs.

$$Score(blobI, blobJ) = \begin{cases} \sum_{f=n}^m ((x_{f,i} - x_{f,j})^2 + (y_{f,i} - y_{f,j})^2), & \text{ambos blobs en frame} \\ \infty, & \text{en caso contrario} \end{cases}$$

Donde $x_{(f,i)}$ e $y_{(f,i)}$, son las coordenadas del blob I en el *frame* f , $x_{(f,j)}$ e $y_{(f,j)}$ son las coordenadas del blob J en el *frame* f , y $n...m$ son los *frames* donde el blob I y el blob J son seguidos de forma simultánea.

5.2.3. Valores específicos de fusión

Una vez explicadas la fusión principal y las dos mejores que permiten obtener unos resultados más correctos, se debe explicar el proceso que se lleva a cabo.

En primer lugar, se calculan las puntuaciones (*scores*) entre los blobs de las cámaras enfrentadas que son las cámaras superpuestas que cubren un área del campo. Hay tres pares de cámaras que se enfrentan: las cámaras 1 y 2, la cámaras 3 y 4, y las cámaras 5 y 6. Para cada una de estas parejas se elabora una lista de asociación de blobs independiente.

La fusión resultante de cada par de cámaras enfrentadas se denomina región. Hay tres regiones, una por cada par de cámaras que se enfrentan. El resultado de la combinación de las tres regiones, cubriendo toda el área del campo, por la fusión de las dos zonas de superposición de las regiones (región de las cámaras 1-2 y región de las cámaras 3-4, y la región de las cámaras 3-4 con la región de las cámaras 5-6) nos proporciona la trayectoria global de cada jugador por todo el terreno de juego.

Los resultados y *testeos* de esta fusión en el sistema se encuentran presentes en [1]. En ellos también aparece una comparación con los resultados obtenidos a partir del *ground truth* que proporciona el creador del *dataset* utilizado.

5.3. Ajustes al prototipo inicial

5.3.1. Modificaciones paramétricas

Para el procesado se han utilizado los vídeos del ISSIA Soccer Dataset (sección 2.3.2). Tras el análisis de las secuencias y de los resultados obtenidos por el prototipo original se han detectado errores que se pretenderá corregir mediante el ajuste paramétrico (esta sección), mediante modificaciones de implementación (sección 5.3.2), o a través de las interfaces gráficas y la corrección *online* (sección 5.4). Los más relevantes han sido:

- Presencia de ruido.
- Autoenfoco de la cámara que produce emborronados.
- Cámaras con efecto de *jitter* por el viento.

El principal problema que se trata de paliar, al igual que en el caso de deportes individuales, es la influencia que tiene el ruido ocasionado en los vídeos, por los problemas anteriormente mencionados. Para ello, se modifica el valor que los siguientes parámetros, ya presentes en el prototipo original:

- **VARNOISE**: Umbral que controla la sensibilidad del detector ante la variación de ruido. Cuanto más alto sea este valor del umbral, más baja es su sensibilidad y al contrario, cuanto más bajo sea este umbral más alta es la sensibilidad. Por esto, se modifica el valor de partida que era 8.0 a un valor de 15.0 para reducir su sensibilidad.
- **TH_MOTION**: Es un parámetro para el cálculo de la función Gamma. Se reduce el valor de este umbral desde el valor de partida 9.99×10^{-5} a un valor de 11×10^{-11} para de esta forma, reducir su sensibilidad a ligeros movimientos de la cámara.

Para la estimación de estos valores se llevaron a cabo de nuevo una serie de barridos paramétricos siguiendo la estrategia de *rough-to-fine*. La evaluación en todo momento fue cualitativa, sin embargo, no es objetivo principal de este proyecto mejorar el rendimiento cuantitativo del prototipo, pese a haberse logrado. Con estas modificaciones ya se comprueba que se obtienen unos mejores resultados en cuanto a detección y seguimiento de los jugadores sobre el terreno de juego.

Se ha definido empíricamente que en el bloque que filtra el seguimiento, se deben obtener resultados para el mismo blob en un mínimo de 10 *frames* consecutivos.

En cuanto al bloque de fusión, la trayectoria fusionada empieza en el *frame* 25, comprobado mediante evaluación cualitativa, que ese tiempo era suficiente para permitir la inicialización del sistema de seguimiento al completo. El valor de *score* para la fusión se ha fijado en 7.5, calculado a partir de una secuencia de entrenamiento con el sistema de evaluación del que dispone el prototipo, valor utilizado para comparar los *scores* hallados para cada pareja de blobs.

5.3.2. Modificaciones de implementación

Se ha modificado la carga de la información de inicialización, así como de los datos de análisis. Esta modificación es similar a la introducida en la aplicación de deportes individuales. Si bien anteriormente esto se realizaba de manera estática en el código del prototipo, en el sistema actual se cargan unos ficheros de configuración de donde se obtienen la información sobre los directorios donde se encuentran los vídeos que se deben procesar, así como los datos para generar los archivos y vídeos de salida tras el proceso.

Cada fila del archivo de configuración contiene los datos de inicialización de cada una de las cámaras del *dataset*. De izquierda a derecha y separados por un espacio en blanco contienen la siguiente información:

- Directorio del vídeo a procesar.
- Vídeo de salida tras el procesado.
- Fichero de texto de salida de posiciones del jugador por *frame*.
- Máscara de procesamiento.
- *Frames* por segundo.

El módulo de seguimiento utilizado es el del sistema base que se describe en el Anexo B, con las modificaciones descritas en las siguientes secciones de dicho anexo. Destaca que el tiempo de ejecución del prototipo es de 8.4 FPS para los vídeos de fútbol,

mucho más bajo que en los vídeos de tenis debido a la mayor resolución de estos vídeos y al mayor número de jugadores que hay que detectar y seguir.

Para que el módulo diseñado para calcular la homografía funcione correctamente, se recibe un fichero de configuración con los puntos de referencia que se calculan y se introducen manualmente en el sistema. El procedimiento utilizado es:

- Se obtiene el fondo.
- Se representan y se seleccionan los puntos de referencia con el cursor de datos.
- Se anotan para aplicarse más adelante en las homografías.

El fichero de configuración con los puntos de referencia que debe recibir este bloque, y cuya forma de recibirlo será descrita en la siguiente sección, debe tener una estructura fija:

- A nivel de fila: cada fila del archivo de texto contiene las coordenadas x e y de los puntos de referencia separadas por un espacio en blanco.
- A nivel general: las primeras cuatro filas son las coordenadas de referencia en píxeles correspondientes a la cámara 1 y las siguientes cuatro filas las coordenadas en metros que corresponden a la misma cámara. Esta estructura se repite para las coordenadas de referencia de las cámaras 2, 4, 5, 6 y 8.

Por último, otra de las modificaciones realizadas al prototipo original, y también una de las más relevantes, ha sido la homogeneización del lenguaje. El sistema de seguimiento se encuentra en lenguaje C++ y para ello se han convertido los módulos de fusión y de homografía que se encontraban disponibles en *scripts* de Matlab a lenguaje C++.

5.4. Interfaz gráfica de usuario

Una vez se han hecho las modificaciones necesarias para intentar solucionar las carencias observadas en el prototipo original para su uso en análisis de vídeos de deportes individuales, se ha llevado a cabo una interfaz de usuario que permite satisfacer dos de los objetivos de este proyecto, es decir, la creación y desarrollo de una interfaz gráfica de usuario que permita su uso a un usuario no experto, y la posibilidad de realizar correcciones de los posibles errores de detección y/o seguimiento de los jugadores de manera *online*. El Anexo A.2 presenta un manual de usuario de la interfaz gráfica para deportes de equipo.

Esta interfaz consta de cuatro ventanas:

- Ventana principal llamada *Aplication* en la que se llevan a cabo los procesamientos de los vídeos y se muestran las localizaciones de los jugadores en cada *frame*.
- Ventana derivada de la principal y que se llama *Trajectories*, que tiene como objetivo permitir al usuario ver las trayectorias que describe un determinado jugador durante un número determinado de *frames* y poder modificar si se ha producido un error al fusionar y no se ha hecho con el identificador de blob adecuado.
- A partir de esta segunda ventana se puede llegar a la tercera llamada *View*, que tiene como objetivo permitir al usuario visualizar los últimos *frames* de vídeo que estén implicados en las cámaras de la lista de asociación de blobs que se esté tratando en ese momento.
- Por último se dispone de la ventana de nombre *Correction*, la cual permite una vez hayan transcurrido al menos dos bloques del número de *frames* que se elija, observar y modificar si se cree conveniente la fusión de las trayectorias globales de esos dos bloques.

En las siguientes secciones se describirán en mayor profundidad.

5.4.1. Ventana principal - *Aplication*

La ventana principal de la interfaz gráfica de usuario para deportes de equipo tiene el aspecto que se puede apreciar en la Figura 5.3.

En la imagen anterior se aprecia como sólo se encuentra habilitada la opción Open (extremo superior izquierdo), la cual contiene un desplegable con dos opciones: *Configuration file* y *Homography file*.

- *Configuration file*: permite seleccionar el fichero de texto que contiene la información inicial necesaria para la correcta ejecución del sistema.
- *Homography file*: permite seleccionar el archivo de texto con los puntos de referencia para la posterior realización por el bloque de homografías de la misma. Las estructuras de ambos ficheros se encuentran descritas en el Anexo B. Si la selección se ha hecho de la forma correcta, se habilita la opción Run (justo debajo de la anterior), que es necesario pulsar para habilitar el botón Play e iniciar la ejecución.



Figura 5.3: Vista de la ventana principal.

Todas estas opciones disponibles y botones se pueden ver descritos con mayor profundidad en el Manual de Usuario que se ha creado en el Anexo A.2.

Una vez pulsado el botón *Play* se inicia la ejecución de los vídeos de las seis cámaras. Un ejemplo del procesado de los vídeos se puede ver en la Figura 5.4.



Figura 5.4: Vista de un momento determinado de la ejecución

En la imagen anterior se aprecia en el centro en la parte superior el número de *frame* que se encuentra en ejecución en ese momento (*Current frame*) y, tanto arriba como abajo de la representación del terreno de juego, se encuentran las visualizaciones de las cámaras. Las de la derecha corresponden con las cámaras enfrentadas que graban la portería de la derecha (cámaras 1 y 2), las del centro corresponden con las cámaras enfrentadas que graban el centro del campo (cámaras 3 y 4) y las de la izquierda corresponden con las cámaras enfrentadas que graban la portería de la izquierda (cámaras 5 y 6).

En el centro de esta ventana principal se encuentra la representación del terreno

de juego y las coordenadas propias de cada cámara se representan con un punto de un color igual al que tiene la leyenda de cada una de las cámaras individuales.

Para que aparezca la ventana de visualización de trayectorias (*Trajectories*) se precisa que pasen un número de *frames* que previamente ha sido determinado. En este caso se ha establecido que sean 500, es decir, 20 segundos de partido.

5.4.2. Ventana de visualización de trayectorias - *Trajectories*

La ventana de visualización de trayectorias de la interfaz gráfica de usuario para deportes de equipo tiene el aspecto que se puede apreciar en la Figura 5.5.



Figura 5.5: Vista de la ventana de visualización de trayectorias.

En la imagen anterior se aprecia como en la parte superior de la ventana hay una representación del terreno de juego sobre el que se dibujarán las trayectorias

de los identificadores de cada cámara que determine el usuario como se explicará a continuación.



Figura 5.6: Representación de trayectorias

Si el usuario selecciona alguno de los apartados de los que dispone (*camera 1 and camera 2*, *camera 3 and camera 4* o *camera 5 and camera 6*), tendrá acceso a opciones que le permitirán ver las trayectorias de los blobs en las cámaras enfrentadas o tener acceso a la fusión de ambas.

También dispone de la posibilidad de escoger un identificador en concreto y así ver su representación en color amarillo, además de mostrar las coordenadas en metros en las que empieza la trayectoria de ese jugador. Seleccionando *tracks* fusión se puede observar la trayectoria global en todo el campo. De cara a la presentación de resultados, esta ventana tiene gran utilidad, y estas gráficas, de este modo o por ejemplo en

forma de mapas de calor son muy recurridas en tareas de análisis de vídeo deportivo. Véase un ejemplo en la Figura 5.6.

Para facilitar al usuario que pueda ver de forma ideal las trayectorias más pequeñas y que no se pueden ver del todo bien al tamaño inicial, se ofrece la opción de realizar un rectángulo *zoom*. Para ello, se debe realizar un recuadro con el ratón sobre la zona del campo que se quiere aumentar y se realizará dicho *zoom*. Si se presiona el botón *Zoom out* se vuelve al tamaño original. Un ejemplo de este *zoom* se ve en la Figura 5.7.

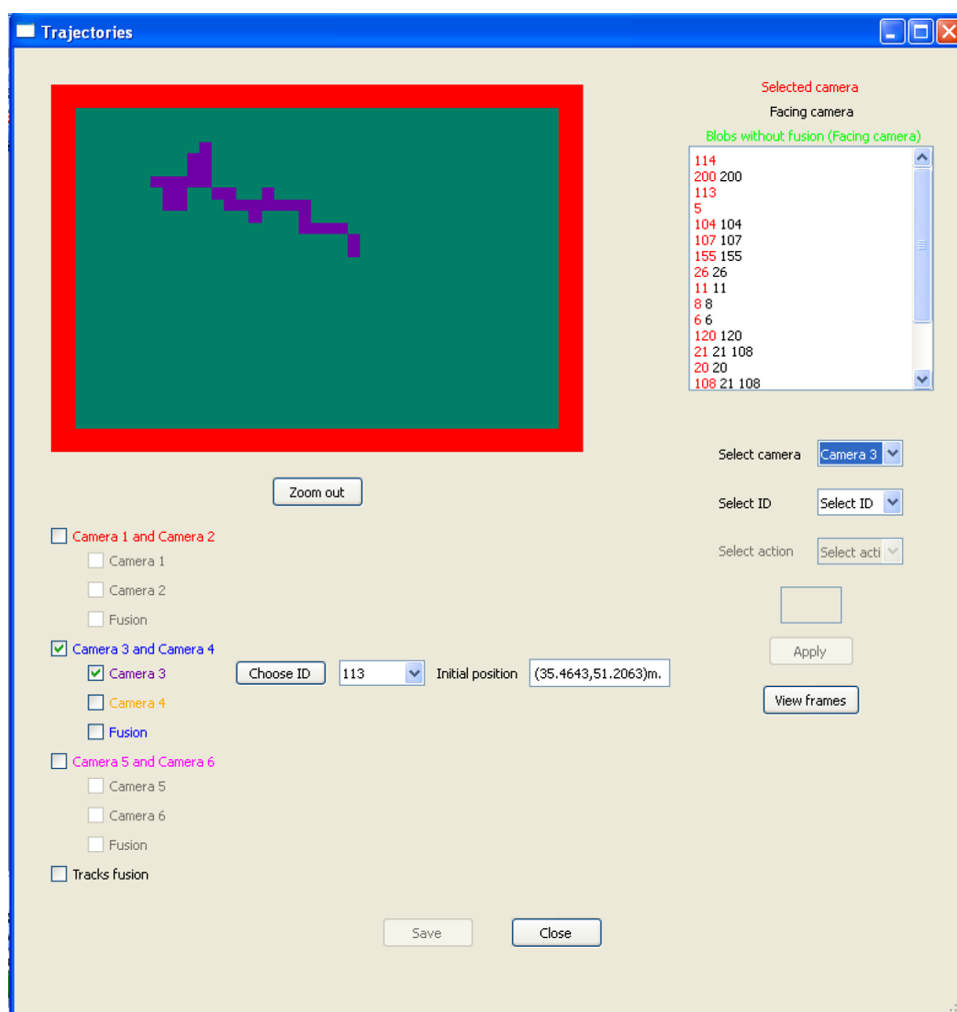


Figura 5.7: *Zoom* aplicado sobre las trayectorias

En la parte derecha de esta pantalla se puede elegir una cámara y ver la lista de asociaciones de blobs de esa cámara con respecto a su cámara enfrentada. Además, se permite elegir un identificador y eliminar o añadir otro blob de la cámara enfrentada

que el usuario considere que se tiene que fusionar con el primero. Para confirmar esta acción hay que apretar el botón *Apply* y posteriormente guardar los cambios en *Save*. Un ejemplo de esta lista de asociaciones se observa en la Figura 5.8.

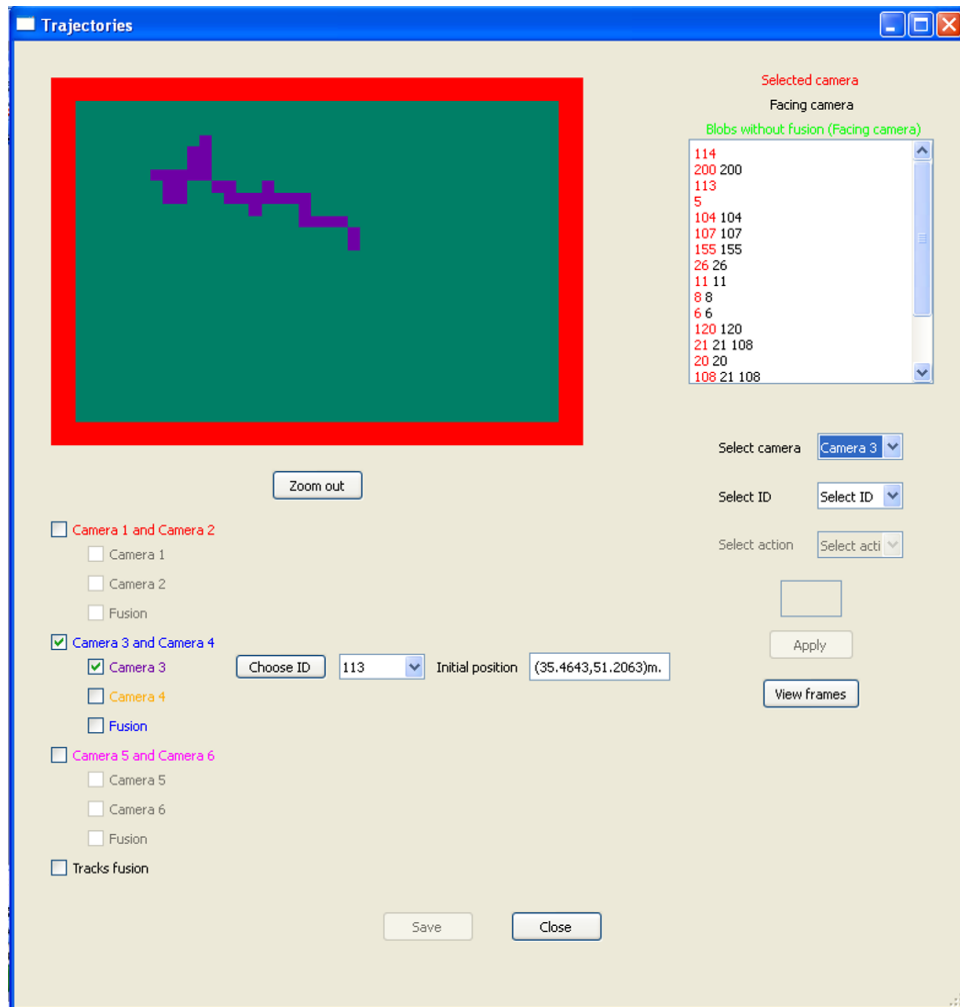


Figura 5.8: Ejemplo de LOA

Por último, si se quiere acceder a una visión de los últimos *frames* de la cámara elegida y su par, se pulsa el botón *View frames* y la aplicación nos lleva a la ventana denominada *View*.

5.4.3. Ventana de visualización de *frames* - *View*

La ventana de visualización de *frames* de la interfaz gráfica de usuario para deportes de equipo tiene el aspecto que se puede apreciar en la Figura 5.9.

En la imagen anterior se aprecia como en la parte superior de la ventana hay

una representación de los últimos *frames* de la cámara elegida y su respectiva cámara enfrentada. Debajo de ello, está la opción *Current frame* para determinar el *frame* actual que se está visualizando, así como la posibilidad de indicar manualmente el número de *frame* al que se quiere ir siempre y cuando esté entre los observables. Véase en la Figura 5.10.



Figura 5.9: Vista de la ventana de visualización de *frames*



Figura 5.10: Elección de *frame*

5.4.4. Ventana de corrección de errores - *Correction*

La ventana de corrección de errores de la interfaz gráfica de usuario para deportes de equipo tiene el aspecto que se puede apreciar en la Figura 5.11.

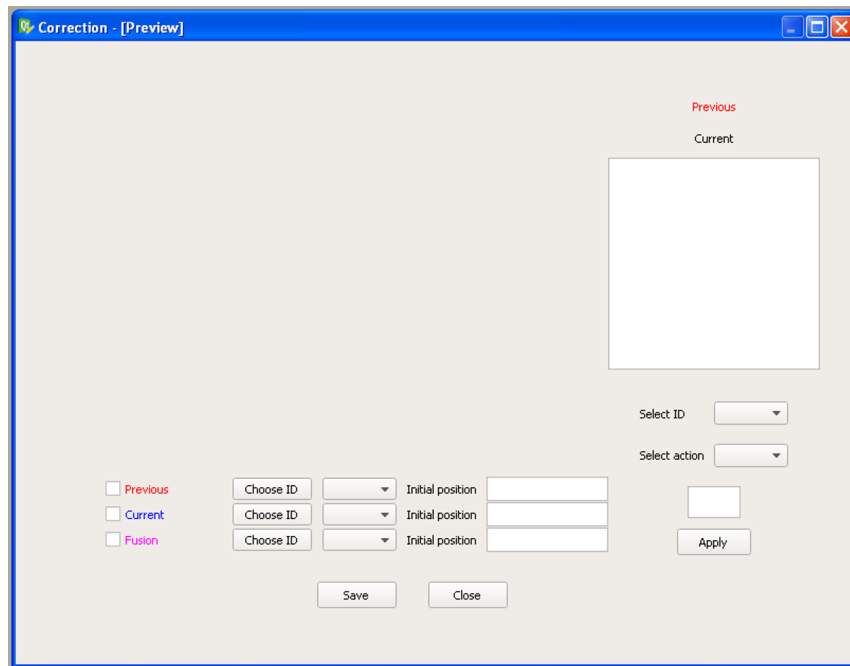


Figura 5.11: Vista de la ventana de corrección de errores.

Esta pantalla se ofrece al usuario cuando han transcurrido al menos dos bloques de los *frames* seleccionados y permite de una forma similar a la ventana de visualización de trayectorias, ver el recorrido de cada jugador en cada uno de los bloques y modificar la fusión de los identificadores del bloque anterior con el posterior.

5.5. Resultados y conclusiones

Tras la inclusión de todas las modificaciones y las interfaces gráficas, el funcionamiento del sistema cumple con todos los objetivos propuesto y satisface las motivaciones del proyecto.

Los resultados del prototipo inicial eran bastante positivos pese a la alta complejidad de la tarea. Sin embargo, los errores por una parte de adaptación al entorno, y por otra de la propia tarea de seguimiento eran dos factores a tener en cuenta como puntos de mejora.

Mediante esta herramienta, a través de los ajustes que se pueden realizar con la interfaz diseñada se pueden corregir gran cantidad de estos, pese a que no se han

realizado evaluaciones cuantitativas, pues están muy supeditadas a la capacidad del usuario de corregir los errores que van surgiendo.

Es necesario comentar que la inclusión de la aplicación ha supuesto un pequeño empeoramiento en el coste computacional de la aplicación. Anteriormente se había determinado que el tiempo medio de ejecución de los vídeos con el seguimiento de los jugadores realizado era de 8.4 FPS, y aunque una vez realizada la interfaz no se ha podido medir este tiempo porque hay periodos de interacción del usuario que alteran este valor, se ha podido estimar que presenta un tiempo de ejecución ligeramente superior al original. Esto es debido a que el bloque de fusión representa un gran coste computacional, y se producen grandes usos de memoria además de los realizados en la detección y seguimiento de los jugadores, al almacenar gran cantidad de *frames* para su posterior uso.

Con estos datos, se puede concluir que, pese a poder realizarse mayores mejoras, principalmente en el aspecto del coste computacional, se ha incidido positivamente en las capacidades originales del algoritmo, alcanzado además aquellos aspectos que motivaron el trabajo realizado. Destacar también, que la interfaz cumple con los requisitos de poder ser utilizada por un usuario no experto, y se le ha dotado de funcionalidad que, pese a no haber sido incluida en esta descripción, si se encuentra en el manual de usuario. Esta funcionalidad es básicamente a nivel de interacción: selección de cámaras, pausas de ejecución, salidas a representar, y otras funcionalidades que hagan más sencilla la tarea del usuario final de la aplicación.

Por último, dada la complejidad de la tarea, se llevó a cabo el diseño de una aplicación de verificación del funcionamiento de la interfaz original. Se ha modificado el sistema de tal forma que se elimina el bloque de seguimiento y se parta de los datos proporcionados por el *ground truth*, disponiendo de únicamente 25 identificadores distintos.

- Árbitro: 200.
- Jueces de línea: 201, 202.
- Portero del equipo blanco: 1.
- Portero del equipo azul: 105
- Jugadores del equipo blanco: 5, 6, 8, 9, 10, 11, 14, 20, 21, 26
- Jugadores del equipo azul: 104, 107, 108, 113, 114, 120, 125, 126, 128, 155

Los resultados obtenidos han permitido verificar que los objetivos a priori marcados para esta aplicación han sido satisfechos, y su funcionalidad es la deseada. Se muestra

un ejemplo de esta segunda aplicación en la Figura 5.12, donde se observa en el cuadro superior derecho como las fusiones son ideales.



Figura 5.12: Vista de la ventana con datos de ground truth

Capítulo 6

Conclusiones y trabajo futuro.

6.1. Conclusiones.

El objetivo principal de este proyecto era llevar a cabo una serie de mejoras sobre un prototipo inicial, capaz de realizar análisis de vídeo sobre distintos eventos deportivos.

Más concretamente, este proyecto surgió como respuesta a las necesidades del prototipo original desde el punto de vista de:

- Portabilidad, usabilidad y futuras mejoras del prototipo original.
- Usabilidad desde el punto de vista de un usuario no experto.
- Visibilidad de los resultados aportados por el análisis del vídeo realizado.
- Supervisión y corrección *online* de los resultados del prototipo.

Para la validación del grado de solución aportado por este proyecto a estas mencionadas necesidades, se escogieron dos tipos de tareas, deportes, que cubrían el espectro en el que era capaz de funcionar el prototipo. La primera de ellas, el tenis, se trataba de un deporte de los clasificados como individuales. Esta tarea era relativamente sencilla. La segunda de ellas, el fútbol, se trataba de una tarea de gran complejidad, tanto a nivel de resultados de análisis, tarea del prototipo original, como de gestión de la aplicación, tarea de este proyecto. Se trata de un deporte etiquetado como colectivo, y para el que las aplicaciones y el margen de mejora de las mismas es muy grande.

Encontrado en este marco, las conclusiones que se pueden sacar del proyecto son las siguientes.

En primer lugar, se ha logrado satisfacer la necesidad de tener un sistema portable y sobre el que trabajar de cara a mejoras sobre el prototipo. Para ello se realizó la unión

de todos los módulos originales, obtenidos en lenguajes de programación distintos, en un solo lenguaje, C++. Se re-implementaron aquellos módulos que fueron necesarios y se adaptaron otros para realizar un prototipo único, compacto, funcional y portable.

En segundo lugar, mediante el diseño de unas interfaces gráficas con un diseño orientado al usuario no-experto, y acompañadas de un sencillo manual de usuario, se ha logrado que esta herramienta se utilizable por usuarios sin conocimientos a priori.

Como tercer aspecto a destacar, la misma interfaz gráfica ha dotado de una mayor visibilidad y facilidad de interpretación de los resultados arrojados por el prototipo. Esta funcionalidad resultad de utilidad tanto desde un punto de vista comercial, pues la hace más atractiva y completa, como desde el punto de vista del investigador, pues a la hora de mejorar el prototipo tendrá *feedback* inmediato de cómo está influyendo su trabajo.

Por último, la inclusión en la aplicación de métodos para la gestión de estados intermedios del prototipo, y de herramientas en la interfaz para interactuar con ella, ha hecho posible que un supervisor del prototipo pueda al mismo tiempo estar visualizando la salida y corrigiendo *online* el funcionamiento de la misma, por lo que los resultados que se obtienen son mucho más fiables.

Se puede concluir, por lo tanto, que a la finalización de este proyecto se han satisfecho los objetivos planteados al comienzo de este trabajo, pese al existente margen de mejora.

Como aspectos un poco más apartados de las motivaciones formales de este proyecto, destacar algunas conclusiones extraídas tras el trabajo con el prototipo y la aplicación.

En el aspecto del funcionamiento de algoritmos de análisis de vídeo deportivo, se ha detectado que el sub-sistema de captura es uno de los elementos más relevantes del sistema completo. Problemas en él, son difícilmente solucionables en etapas posteriores.

En el aspecto de la gestión de los sub-sistemas, los eventos deportivos colectivos entrañan una gran complejidad desde el punto de vista de la gestión de recursos. La gestión de múltiples cámaras, múltiples objetivos y toda su información, requiere una gran minuciosidad en la implementación para no saturar recursos de memoria o computacionales.

Por último, en el aspecto de las interfaces gráficas, las pruebas con usuarios no expertos, realizadas aunque no incluidas en el proyecto por haber sido realizadas sin ser completamente rigurosas, aportan gran cantidad de información a la hora de decidir cómo presentar la información, o para el control de errores.

6.2. Trabajo futuro.

Tras la realización de este proyecto, son numerosas las opciones de trabajo futuro en las que se considera que se podría avanzar con el fin de mejorar los resultados obtenidos, tanto a nivel del propio prototipo, como de las aplicaciones realizadas. A continuación, se resumen algunas ideas que se consideran de mayor utilidad o necesidad.

En primer lugar, a nivel del prototipo se podría mejorar el extractor de fondo del sistema ya que el utilizado es relativamente simple y los resultados que obtiene, con numerosos errores, han de ser corregidos a posteriori. Los trabajos podrían centrarse en la fabricación de un extractor más complejo o uno con un menor coste computacional, por ejemplo, la selección de píxeles invariantes (o dentro de los márgenes) para un cierto número de *frames*, aprovechando el conocimiento previo del entorno de aplicación.

Siguiendo con trabajo sobre el prototipo, se considera que el aspecto donde más se puede seguir estudiando y mejorando, también por ser el más complejo, es el sistema de seguimiento. El sistema utilizado fue adaptado de uno diseñado para videovigilancia, con lo que se cree, que la inclusión de uno más *ad-hoc* para vídeos deportivos podría dar mejores resultados.

Sobre las aplicaciones creadas, en las interfaces gráficas de usuario se podría trabajar en la búsqueda de que el procesamiento sea más rápido, para ello se podría organizar el sistema de una manera más modular y buscar la forma de rebajar el coste computacional en operaciones que retardan el proceso.

También se podría incluir conocimientos de más alto nivel, para que la interpretación de los resultados fuera más completa. Por ejemplo, reglas para analizar el comportamiento de los jugadores, zonas de influencia, etc. que podrían ser de utilidad para su comercialización deportiva.

En cuanto al resto del sistema, se podrían obtener *datasets* con mejores condiciones de captura, como mejores posicionamientos de las cámaras, mayor resolución y estabilidad, o sensores más diversos.

Bibliografía

- [1] R. Martín, “Detección y seguimiento en vídeo deportivos multicámara,” *Proyecto Fin de Carrera, Escuela Politécnica Superior, Universidad Autónoma de Madrid*, Septiembre 2012.
- [2] R. Martín and J. M. Martínez, “A semi-supervised system for players detection and tracking in multi-camera soccer videos,” *Multimedia Tools and Applications*, vol. 73, no. 3, pp. 1617–1642, 2014.
- [3] R. Martín Nieto and J. M. Martínez Sánchez, “An automatic system for sports analytics in multi-camera tennis videos,” in *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pp. 438–442, IEEE, 2013.
- [4] R. Martín and J. Martínez, *Human Behaviour Understanding in Networked (distributed) Sensing - Theory and applications of networks of sensors*, vol. Chapter VI - Action and gestures: Automatic players detection and tracking in multi-camera tennis videos. Springer, 2014.
- [5] Y. Huang, J. Llach, and S. Bhagavathy, “Players and ball detection in soccer videos based on color segmentation and shape analysis,” in *Multimedia Content Analysis and Mining*, pp. 416–425, Springer, 2007.
- [6] D. Liang, Y. Liu, Q. Huang, and W. Gao, “A scheme for ball detection and tracking in broadcast soccer video,” in *Advances in Multimedia Information Processing-PCM 2005*, pp. 864–875, Springer, 2005.
- [7] X.-F. Tong, H.-Q. Lu, and Q.-S. Liu, “An effective and fast soccer ball detection and tracking method,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4, pp. 795–798, IEEE, 2004.
- [8] P. Figueroa, N. Leite, R. M. Barros, I. Cohen, and G. Medioni, “Tracking soccer players using the graph representation,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4, pp. 787–790, IEEE, 2004.
- [9] G. Kayumbi, P. L. Mazzeo, P. Spagnolo, M. Taj, and A. Cavallaro, “Distributed visual sensing for virtual top-view trajectory generation in football videos,” in *Proceedings of the 2008 international conference on Content-based image and video retrieval*, pp. 535–542, ACM, 2008.

- [10] C. Poppe, S. D. Bruyne, S. Verstockt, and R. de Walle, "Multi-camera analysis of soccer sequences," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 26–31, IEEE, 2010.
- [11] M. Xu, J. Orwell, and G. Jones, "Tracking football players with multiple cameras," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 5, pp. 2909–2912, IEEE, 2004.
- [12] Y. Seo, S. Choi, H. Kim, and K.-S. Hong, "Where are the ball and players? soccer game analysis with color-based tracking and image mosaick," in *Image Analysis and Processing*, pp. 196–203, Springer, 1997.
- [13] D. Farin *et al.*, "Current and emerging topics in sports video processing," in *2005 IEEE International Conference on Multimedia and Expo*, pp. 526–529, IEEE, 2005.
- [14] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," in *Proceedings of the eighth ACM international conference on Multimedia*, pp. 105–115, ACM, 2000.
- [15] T. Taki, J.-i. Hasegawa, and T. Fukumura, "Development of motion analysis system for quantitative evaluation of teamwork in soccer games," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 3, pp. 815–818, IEEE, 1996.
- [16] X. Yu, H. W. Leong, J.-H. Lim, Q. Tian, and Z. Jiang, "Team possession analysis for broadcast soccer video based on ball trajectory," in *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, vol. 3, pp. 1811–1815, IEEE, 2003.
- [17] N. Inamoto and H. Saito, "Free viewpoint video synthesis and presentation from multiple sporting videos," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 90, no. 2, pp. 40–49, 2007.
- [18] K. Wan, X. Yan, and C. Xu, "Automatic mobile sports highlights," in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pp. 4–pp, IEEE, 2005.
- [19] T. Bebie and H. Bieri, "Soccerman-reconstructing soccer games from video sequences," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1, pp. 898–902, IEEE, 1998.
- [20] G. Kayumbi, N. Anjum, and A. Cavallaro, "Global trajectory reconstruction from distributed visual sensors," in *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, pp. 1–8, IEEE, 2008.
- [21] N. Anjum and A. Cavallaro, "Trajectory association and fusion across partially overlapping cameras," in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pp. 201–206, IEEE, 2009.

- [22] Y. A. Sheikh and M. Shah, "Trajectory association across multiple airborne cameras," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 361–367, 2008.
- [23] M. Taj and A. Cavallaro, "Multi-camera track-before-detect," in *Distributed Smart Cameras, 2009. ICDCS 2009. Third ACM/IEEE International Conference on*, pp. 1–6, IEEE, 2009.
- [24] J. Kang, I. Cohen, G. Medioni, *et al.*, "Tracking people in crowded scenes across multiple cameras," in *Asian conference on computer vision*, vol. 7, p. 15, Citeseer, 2004.
- [25] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. Van Gool, "Color-based object tracking in multi-camera environments," in *Pattern Recognition*, pp. 591–599, Springer, 2003.
- [26] I. N. Junejo and H. Foroosh, "Trajectory rectification and path modeling for video surveillance," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–7, IEEE, 2007.
- [27] S. Iwase and H. Saito, "Parallel tracking of all soccer players by integrating detected positions in multiple view images," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4, pp. 751–754, IEEE, 2004.
- [28] Y. L. de Meneses, P. Roudit, F. Luisier, and J. Jacot, "Trajectory analysis for sport and video surveillance," in *Progress In Computer Vision And Image Analysis*, pp. 369–379, 2010.
- [29] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [30] P. J. Figueroa, N. J. Leite, and R. M. Barros, "Tracking soccer players aiming their kinematical motion analysis," *Computer Vision and Image Understanding*, vol. 101, no. 2, pp. 122–135, 2006.
- [31] T. Misu, S. Gohshi, Y. Izumi, Y. Fujita, and M. Naemura, "Robust tracking of athletes using multiple features of multiple views," 2004.
- [32] W. Du, J.-B. Hayet, J. Piater, and J. Verly, "Collaborative multi-camera tracking of athletes in team sports," in *Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE)*, vol. 5439, pp. 2–13, 2006.
- [33] L. A. C. Campos, "Contribuciones a la detección de objetos robados y abandonados en secuencias de vídeo-seguridad," *Proyecto Fin de Carrera, Escuela Politécnica Superior, Universidad Autónoma de Madrid*, Julio 2011.
- [34] J. C. SanMiguel and J. M. Martínez, "A semantic-based probabilistic approach for real-time video event recognition," *Computer Vision and Image Understanding*, vol. 116, no. 9, pp. 937–952, 2012.

Apéndice A

Guia de uso de aplicaciones

Este apéndice contiene las guías de usuario de las aplicaciones desarrolladas para deportes individuales, tenis, y colectivos, fútbol.

A.1. Manual de GUI de deportes individuales

En esta sección se va a redactar una guía de usuario sobre la Interfaz Gráfica de Usuario desarrollada para deportes individuales. Con este manual se pretende que cualquier persona que disponga de la posibilidad de acceder a esta aplicación pueda basándose sólo en esto disponer de todos los datos necesarios para ejecutarla e interactuar con ella como desee.

La interfaz consta de dos ventanas, una principal y otra de corrección de errores que depende de la primera. Entonces se va a proceder a explicar la funcionalidad de cada uno de los botones y opciones que tiene esta ventana principal que recibe el nombre de *Application*.

El aspecto visual de *Application* se muestra en la Figura A.1.

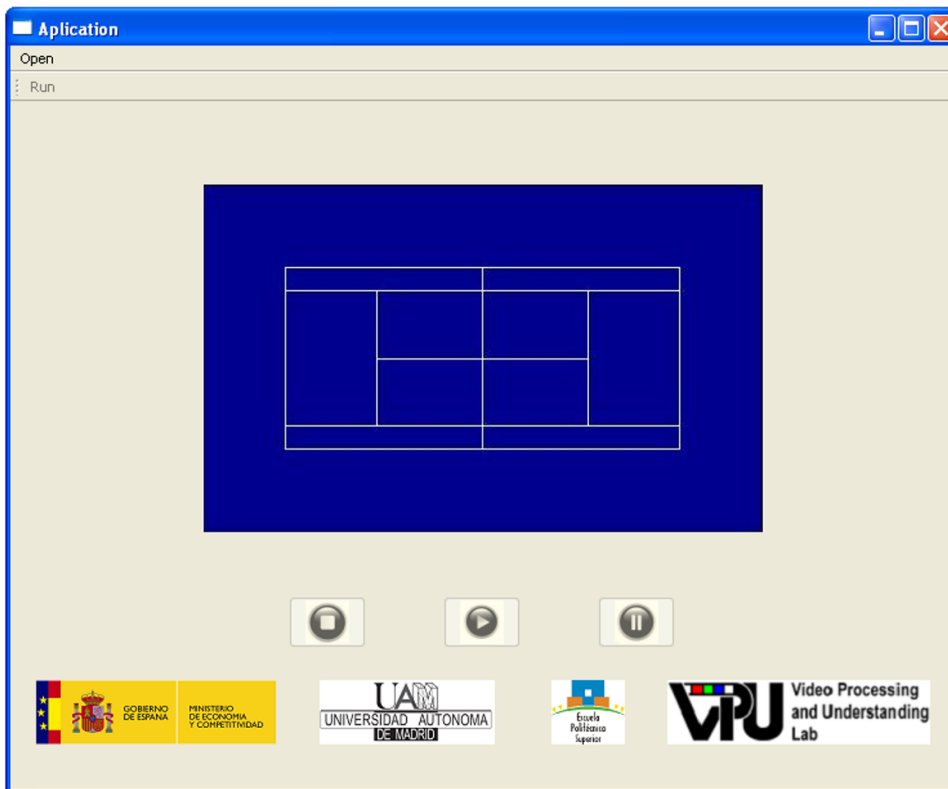


Figura A.1: Aspecto visual de la ventana principal antes de interactuar con ella.

Como se aprecia en la imagen anterior se encuentran todas las opciones inhabilitadas exceptuando la pestaña Open que tiene como función realizar la inicialización del sistema y, para ello, necesita cargar ficheros de texto que dispongan de los parámetros y de la información necesaria para llevar a cabo esa inicialización.

Pulsando la pestaña Open se abre un desplegable que ofrece al usuario dos opciones:

- Configuration file: Esta opción permite navegar por los diferentes directorios para cargar el fichero de configuración que permita inicializar el sistema. Una vez seleccionado se añadirá un tick a la izquierda del nombre de esta opción como se verá a continuación.
- Homography file: Esta opción permite navegar por los diferentes directorios para cargar el fichero que contiene los puntos de referencia para calcular las homografías. Una vez seleccionado se añadirá un tick a la izquierda del nombre de esta opción como se verá a continuación.

En la Figura A.2 se observa el resultado derivado de realizar el proceso anterior.

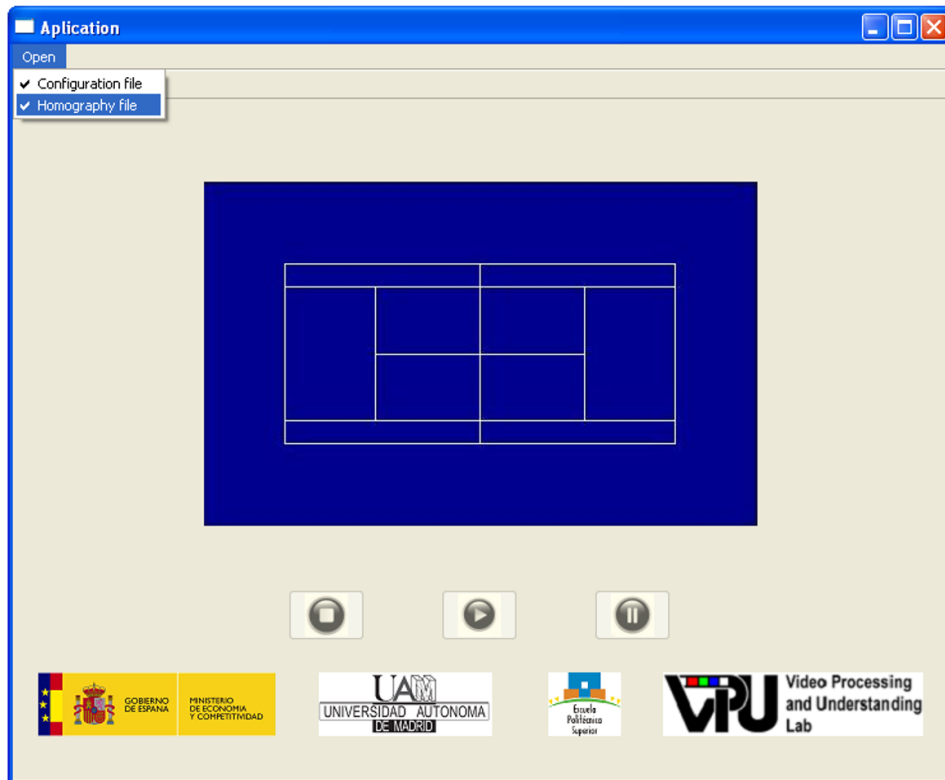


Figura A.2: Resultado de seleccionar ambos ficheros de inicialización.

Una vez seleccionados ambos ficheros de datos se habilita la opción Run, la cual debe ser pulsada para habilitar los botones que permiten al usuario reproducir los vídeos e interactuar con ellos. Véase la Figura A.3.

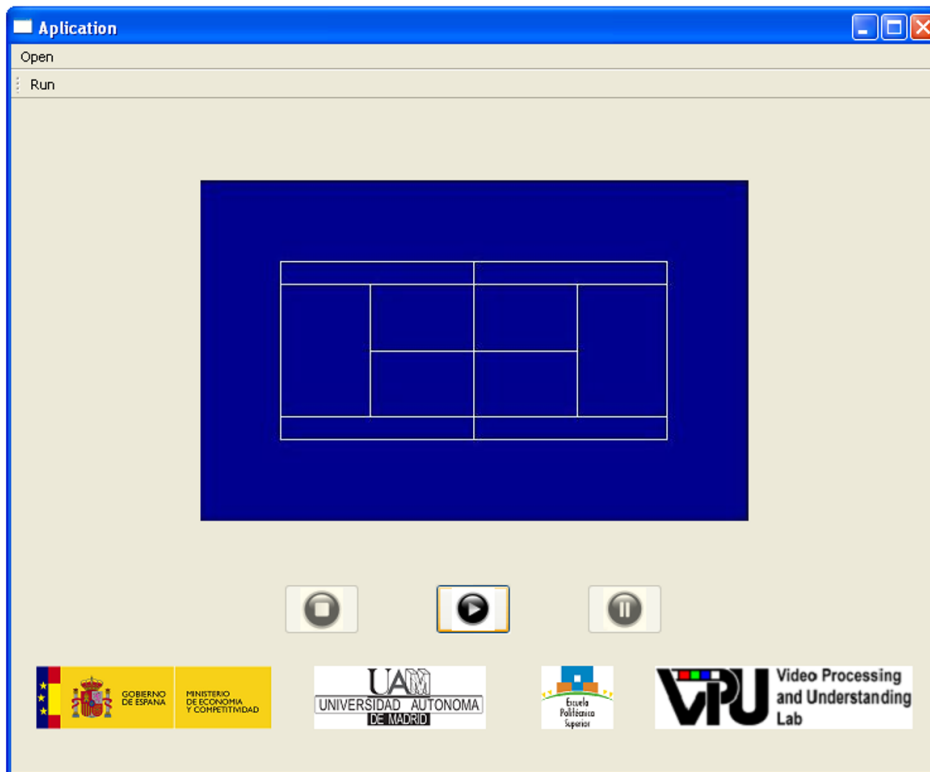


Figura A.3: Habilitado el botón Play

Ahora ya está todo preparado para comenzar el procesado y ejecución de los vídeos de las seis cámaras. Para ello sólo se debe pulsar el botón Play. Esto además de comenzar la ejecución y permitir visualizar al usuario las seis cámaras y los resultados de sus posiciones en el terreno de juego (representado en el centro), permite habilitar los botones de Stop (a la izquierda de Play) y de Pause (a la derecha de Play).

Si se presiona el botón de Stop se para la ejecución y todo vuelve al estado inicial, eso sí, los ficheros seleccionados se mantienen y está habilitada únicamente la opción Run.

Si se presiona el botón Pause se detiene momentáneamente la ejecución y se activa el botón Play. Los vídeos observados en pantalla muestran el último frame procesado antes de parar la ejecución.

En el extremo superior derecho se encuentran las opciones que permiten maximizar o minimizar la pantalla y la que permite cerrar la ventana en cuyo caso se saldría de esta aplicación y se daría por finalizada.

Pulsando con el botón izquierdo del ratón sobre una de las cámaras en ejecución aparece la ventana de corrección de errores, llamada *Correction*.

El aspecto visual de *Correction* se muestra en la Figura A.4.

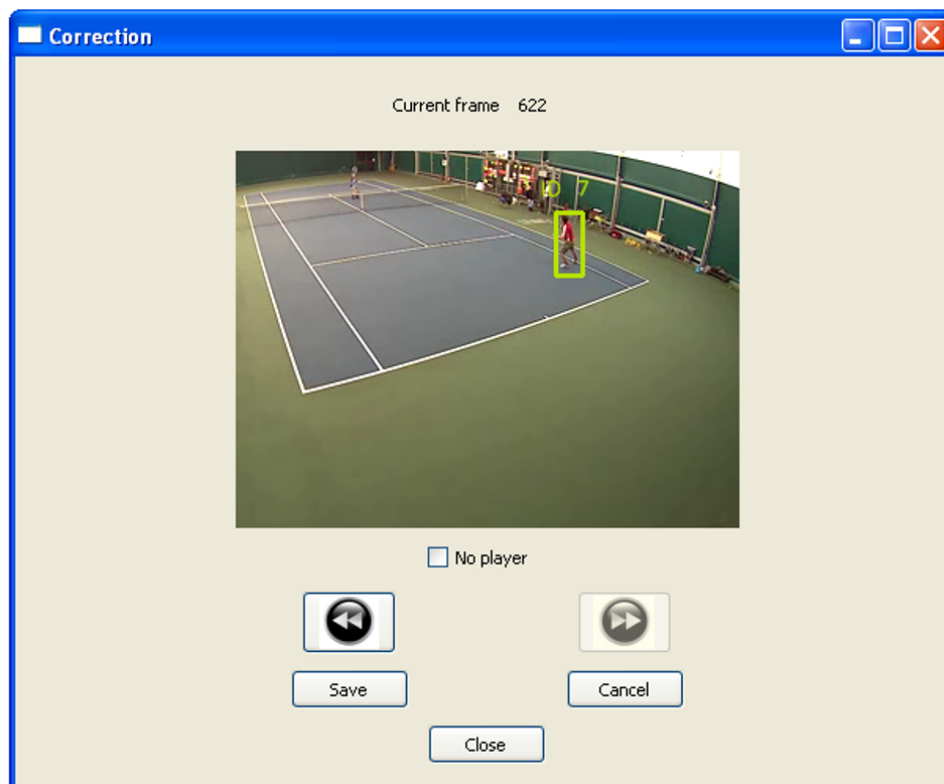


Figura A.4: Aspecto visual de la ventana de corrección antes de interactuar con ella.

En la imagen anterior se aprecia como el usuario se encuentra la ventana de corrección tras pulsar en una cámara de la ventana principal. En este momento se para la ejecución del sistema y sólo se puede interactuar con esta ventana para corregir errores.

En la parte superior aparece Current frame que informa del frame exacto que se ha seleccionado e inmediatamente debajo aparece la imagen fija que estaba grabando la cámara elegida en ese momento determinado, junto con un botón para ir a un frame anterior (hasta 49 anteriores) o para ir a uno posterior siempre y cuando sea posible. Véase en la Figura A.5.

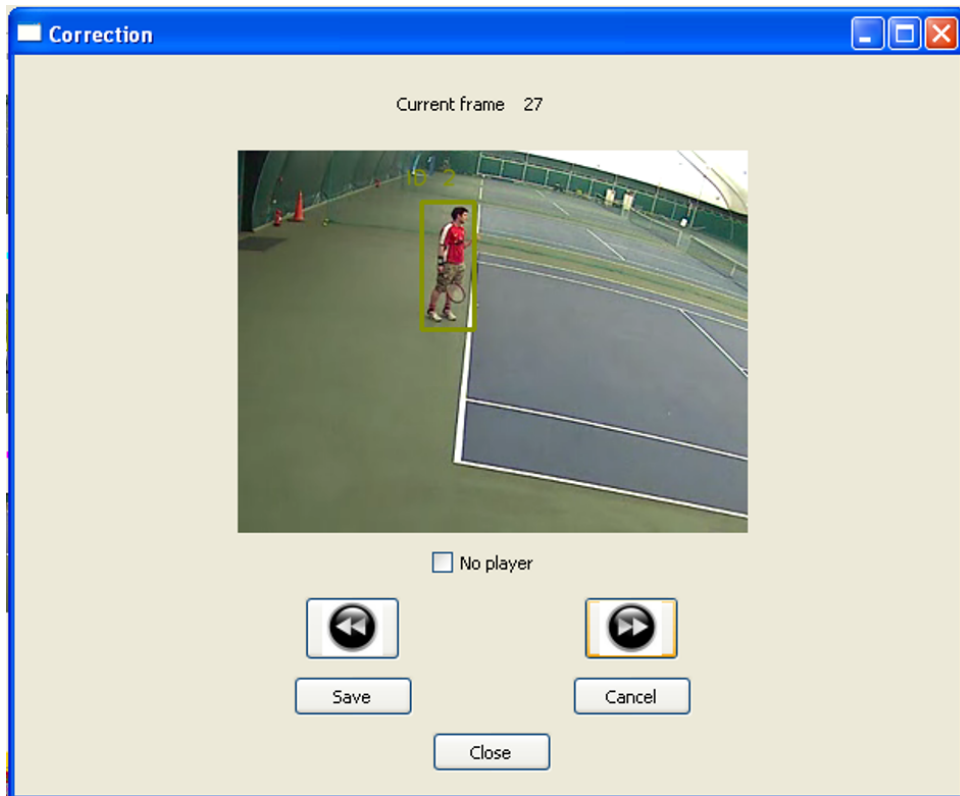


Figura A.5: Habilitados los botones para desplazarse por los frames.

Las dos maneras que se ofrecen para solventar un fallo en la detección o el seguimiento del jugador son:

- Modificar la posición del jugador creando un nuevo bounding box, para ello se pincha con el botón izquierdo del ratón sobre el frame que se quiera corregir y ese será uno de los vértices de la nueva localización del jugador y se arrastra hasta formar el rectángulo (Figura A.6).

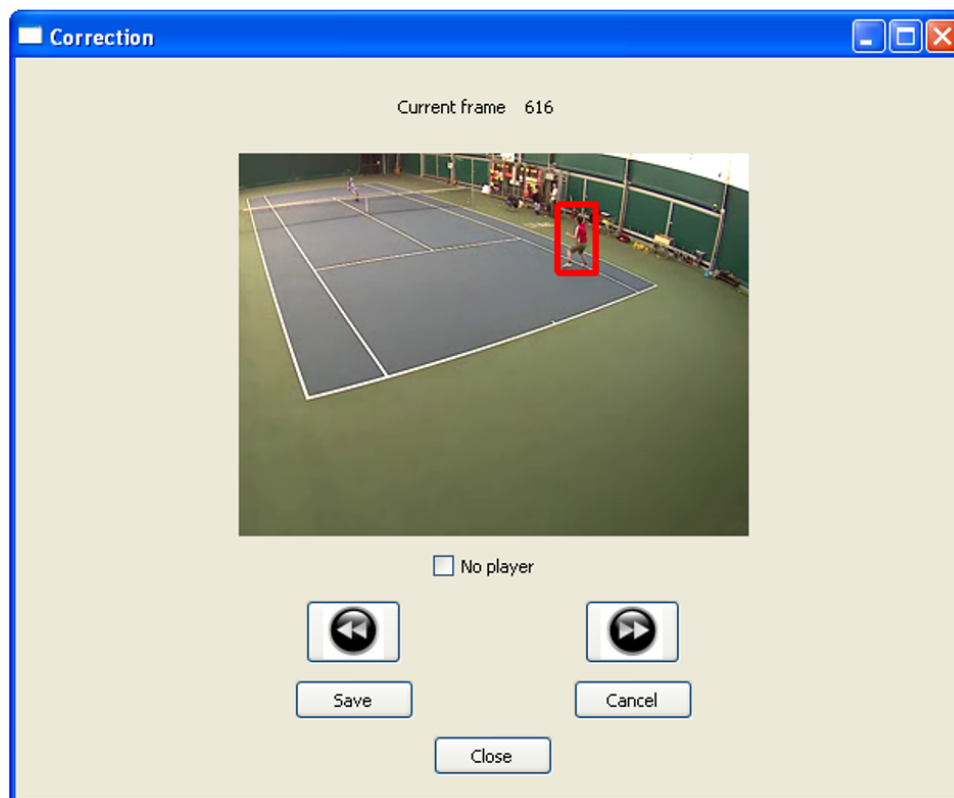


Figura A.6: Proceso de corrección de la posición de un jugador

- Habilitar la opción No player que sirve para indicar que en ese frame en el que se ha podido detectar un jugador no hay ninguno.

Para que las dos opciones descritas anteriormente se lleven a cabo es indispensable pulsar el botón Save para que se guarden los cambios. En caso contrario, si no se quieren confirmar los cambios se debe pulsar el botón Cancel, que anulará todas las modificaciones realizadas.

Por último, como en el caso de la ventana principal de la aplicación en el extremo superior derecho se encuentran las opciones que permiten maximizar o minimizar la pantalla y la que permite cerrar la ventana en cuyo caso se saldría de esta ventana de corrección y se iría a la principal. Esta misma función cumple el botón Close.

NOTA: Es importante destacar que se ha diseñado la aplicación de tal forma que si se realiza una acción de forma errónea, la interfaz avisa al usuario del problema ocasionado con un mensaje de error. Véase en la Figura A.7.

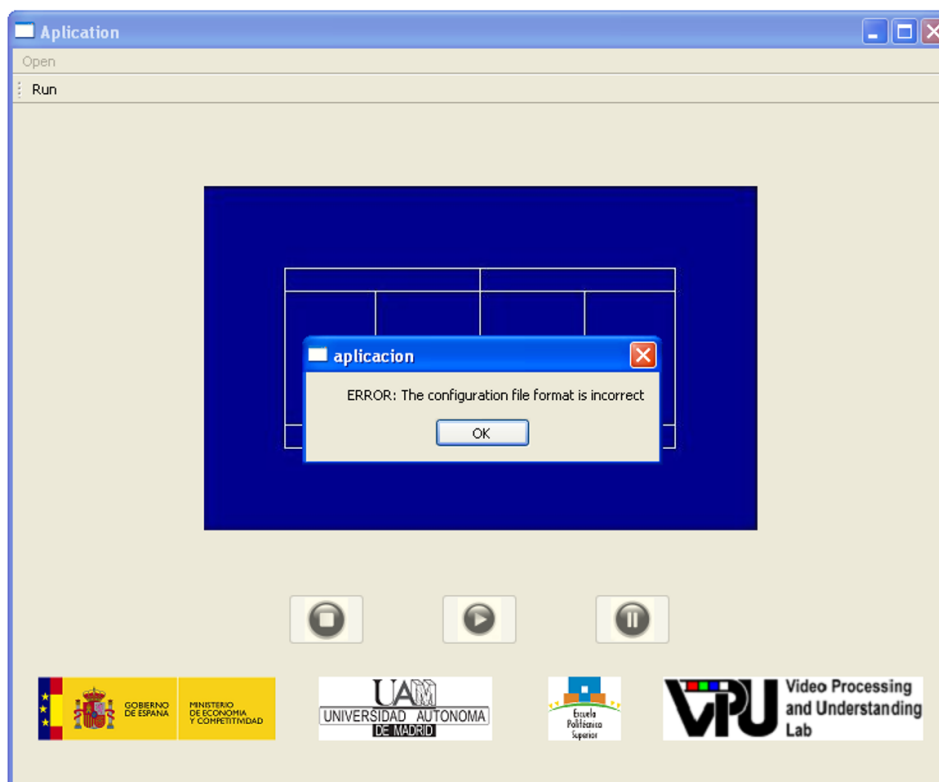


Figura A.7: Error al seleccionar el archivo de configuración.

A.2. Manual de GUI de deportes equipo

En esta sección se va a redactar una guía de usuario sobre la Interfaz Gráfica de Usuario desarrollada para deportes de equipo. Con este manual se pretende que cualquier persona que disponga de la posibilidad de acceder a esta aplicación pueda basándose sólo en esto disponer de todos los datos necesarios para ejecutarla e interactuar con ella como desee.

La interfaz consta de cuatro ventanas, una principal, otra de visualización de trayectorias, una tercera de visualización de frames y una cuarta de corrección de errores en la fusión. Entonces se va a proceder a explicar la funcionalidad de cada uno de los botones y opciones que tiene esta ventana principal que recibe el nombre de *Aplication*.

El aspecto visual de *Aplication* se muestra en la Figura A.8.

Como se aprecia en la imagen anterior se encuentran todas las opciones inhabilitadas exceptuando la pestaña *Open* que tiene como función realizar la inicialización del sistema y, para ello, necesita cargar ficheros de texto que dispongan de los parámetros



Figura A.8: Aspecto visual de la ventana principal antes de interactuar con ella.

y de la información necesaria para llevar a cabo esa inicialización.

Pulsando la pestaña *Open* se abre un desplegable que ofrece al usuario dos opciones:

- Configuration file: Esta opción permite navegar por los diferentes directorios para cargar el fichero de configuración que permita inicializar el sistema. Una vez seleccionado se añadirá un tick a la izquierda del nombre de esta opción como se verá a continuación.
- Homography file: Esta opción permite navegar por los diferentes directorios para cargar el fichero que contiene los puntos de referencia para calcular las homografías. Una vez seleccionado se añadirá un tick a la izquierda del nombre de esta opción como se verá a continuación.

En la Figura A.9 se observa el resultado derivado de realizar el proceso anterior.

Una vez seleccionados ambos ficheros de datos se habilita la opción Run, la cual debe ser pulsada para habilitar los botones que permiten al usuario reproducir los vídeos e interactuar con ellos. Véase la Figura A.10.



Figura A.9: Resultado de seleccionar ambos ficheros de inicialización.

Ahora ya está todo preparado para comenzar el procesado y ejecución de los vídeos de las seis cámaras. Para ello sólo se debe pulsar el botón Play. Esto además de comenzar la ejecución y permitir visualizar al usuario las seis cámaras y los resultados de sus posiciones en el terreno de juego (representado en el centro), permite habilitar los botones de Stop (a la izquierda de Play) y de Pause (a la derecha de Play).

Si se presiona el botón de Stop se para la ejecución y todo vuelve al estado inicial, eso sí, los ficheros seleccionados se mantienen y está habilitada únicamente la opción Run como se ve en la Figura A.9.

Si se presiona el botón Pause se detiene momentáneamente la ejecución y se activa el botón Play. Los vídeos observados en pantalla muestran el último frame procesado antes de parar la ejecución.

En el extremo superior derecho se encuentran las opciones que permiten maximizar o minimizar la pantalla y la que permite cerrar la ventana en cuyo caso se saldría de esta aplicación y se daría por finalizada.

Cuando se reproduce el número de frames previamente asignado aparece la venta de visualización de trayectorias, llamada *Trajectories*.

El aspecto visual de Trajectories se muestra en la Figura A.11.



Figura A.10: Habilitado el botón Play

En la imagen anterior se aprecia como el usuario se encuentra la ventana de visualización de trayectorias una vez que han transcurrido en este caso 500 frames. En este momento se para la ejecución del sistema y sólo se puede interactuar con esta ventana para ver y modificar las trayectorias obtenidas de la fusión.

En la parte superior izquierda aparece el terreno de juego en el cual se van a representar las trayectorias que el usuario desee. Debajo de esto se encuentra el botón Zoom out que permite deshabilitar la opción de zoom, la cual se habilita haciendo un recuadro con el ratón sobre la zona del terreno de juego que se quiera observar con más profundidad.

Más abajo se presentan cuatro opciones que se pueden habilitar: Camera 1 and Camera 2, Camera 3 and Camera 4, Camera 5 and Camera 6 y Tracks fusión. Al habilitar una de las opciones anteriores se abren nuevas posibilidades que son las cámaras independientes y la fusión de ambas cámaras enfrentadas. En cada una de ellas se puede elegir un único identificador (Choose ID) para observar su trayectoria en color amarillo y las coordenadas de inicio de su trayectoria. Véase en la Figura A.12.

En la parte derecha de la ventana se ofrece la posibilidad de elegir una de las



Figura A.11: Aspecto visual de la ventana de visualización de trayectorias antes de interactuar con ella.

cámaras en el desplegable denominado Select camera y una vez seleccionada se puede ver justo arriba la lista de asociaciones de blobs correspondiente a esa cámara y a su correspondiente enfrentada.

Cuando la lista se ha desplegado, se permite seleccionar un identificador y realizar sobre él la opción de añadir o eliminar un blob que tiene asignado para realizar la fusión. Para ello, los desplegables son Select ID y Select action. En el caso de que la opción elegida sea añadir, se debe escribir el número del nuevo blob en el recuadro inmediatamente inferior. Para confirmar los cambios se aprieta el botón Apply y luego Save para guardar los cambios. Un ejemplo de este uso se muestra en la Figura A.13.

En el extremo superior derecho se encuentran las opciones que permiten maximizar o minimizar la pantalla y la que permite cerrar la ventana en cuyo caso se saldría de esta ventana hacia la principal. Para esto último también sirve el botón Close.

Cuando se quiere ver con más detenimiento los últimos frames de las cámaras implicadas en la lista de asociación se pulsa sobre el botón View frames y aparece la ventana de visualización de frames, llamada View.

El aspecto visual de View se muestra en la Figura A.14.

En la imagen anterior se aprecia como el usuario se encuentra la ventana de visualización de frames. Se observan los últimos frames procesados de la cámara seleccionada



Figura A.12: Muestra la trayectoria del identificador elegido

y de su opuesta, así como la opción Current frame que indica el frame que se está visualizando.

Además existe la posibilidad de desplazarse por los frames disponibles ya sea yendo a uno indicado directamente (Go to) o con los botones que permiten ir al anterior o al posterior. Véase en la Figura A.15.

En el extremo superior derecho se encuentran las opciones que permiten maximizar o minimizar la pantalla y la que permite cerrar la ventana en cuyo caso se saldría de esta ventana hacia la principal. Para esto último también sirve el botón Close.

Cuando se reproduce el número de frames previamente asignado en al menos dos ocasiones aparece la venta de corrección de la fusión, llamada *Correction*.

El aspecto visual de Correction se muestra en la Figura A.16.

En la imagen anterior se aprecia como el usuario se encuentra la ventana de corrección. En la parte superior izquierda aparece el terreno de juego en el cual se van a representar las trayectorias que el usuario desee. Debajo de esto se encuentra la posibilidad de elegir ente tres opciones Previous, Current y Fusion, con las trayectorias del grupo de frames anterior, del actual o la fusión de ambas.

En cada una de ellas se puede elegir un único identificador (Choose ID) para observar su trayectoria en color amarillo y las coordenadas de inicio de su trayectoria.

En la parte derecha de la ventana se ofrece la posibilidad de elegir una de las



Figura A.13: Muestra la lista de asociaciones de blobs

cámaras en el desplegable denominado Select camera y una vez seleccionada se puede ver justo arriba la lista de asociaciones de blobs correspondiente a esa cámara y a su correspondiente enfrentada.

Cuando la lista se ha desplegado, se permite seleccionar un identificador y realizar sobre él la opción de añadir o eliminar otro id que tiene asignado para realizar la fusión. Para ello, los desplegables son Select ID y Select action. En el caso de que la opción elegida sea añadir, se debe escribir el número del nuevo identificador en el recuadro inmediatamente inferior. Para confirmar los cambios se aprieta el botón Apply y luego Save para guardar los cambios.

En el extremo superior derecho se encuentran las opciones que permiten maximizar o minimizar la pantalla y la que permite cerrar la ventana en cuyo caso se saldría de esta ventana hacia la principal. Para esto último también sirve el botón Close.

NOTA: Es importante destacar que se ha diseñado la aplicación de tal forma que si se realiza una acción de forma errónea, la interfaz avisa al usuario del problema ocasionado con un mensaje de error.



Figura A.14: Aspecto visual de la ventana de visualización de frames antes de interactuar con ella.



Figura A.15: Desplazamiento por los frames.

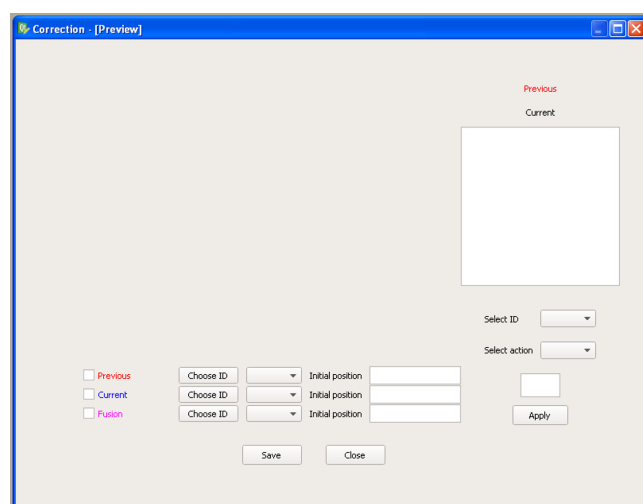


Figura A.16: Aspecto visual de la ventana de corrección de la fusión

Apéndice B

Sistema base

Este anexo ha sido extraído de [30], el cual está basado en el sistema descrito en [31].

B.1. Introducción

El trabajo presentado en este documento parte de un sistema de análisis de vídeo para detección de objetos robados y abandonados proporcionado por el Video Processing and Understanding Lab (VPU Lab). Este sistema está diseñado para trabajar como parte de un marco de vídeo-vigilancia capaz de activar alarmas para eventos detectados en tiempo real. Este requisito impone límites a los tiempos de procesamiento de los algoritmos utilizados en cada uno de los módulos de análisis. Después de la etapa inicial de adquisición de frames, en el módulo de segmentación de frente (Foreground Segmentation Module) se genera una máscara de frente para cada frame entrante. Esta máscara de frente consiste en una imagen binaria que identifica los píxeles que pertenecen al movimiento o blobs fijos. Tras esto, algunas técnicas de post-procesamiento se aplican a esta máscara con el objetivo de eliminar ruido y sombras. Después de eso, el módulo de extracción de blob (Blob Extraction Module) determina los componentes conexos de la máscara de frente. En la etapa siguiente, el módulo de seguimiento de blobs (Blob Tracking Module) intenta asociar un identificador único para cada blob extraído a lo largo de los distintos frames.

B.2. Módulo de segmentación de frente

El propósito del módulo de segmentación de frente (Foreground Segmentation Module) es la generación de máscaras binarias que indican si los píxeles de la imagen

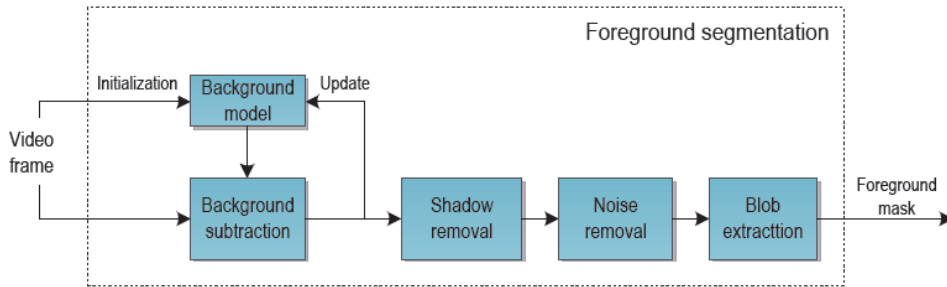


Figura B.1: Diagrama de bloques del módulo de segmentación de frente (Foreground Segmentation Module). Fuente [1]

pertenecen al fondo o al frente.

Basado en la técnica de segmentación de fondo (Background Subtraction, BGS), se parte de un modelo de fondo inicial que posteriormente se compara con los frames entrantes. La máscara inicial obtenida se somete después a operaciones de eliminación de ruido y de sombra con el fin de obtener la máscara final de primer plano para el frame actual y se realiza un análisis de componentes conexos para la extracción de blobs. La Figura B.1 representa el diagrama de bloques del módulo de segmentación de frente (Foreground Segmentation Module).

B.2.1. Sustracción de fondo (Background Subtraction)

El modelo de fondo se inicializa con el valor promedio de una secuencia corta de entrenamiento de frames que no contienen objetos en primer plano.

Este modelo se actualiza de forma adaptativa para considerar cambios lentos en las condiciones de iluminación global utilizando un método de promedio móvil. Entonces, la distancia al modelo de fondo es calculada para cada frame entrante. Consiste en la diferencia al cuadrado entre las dos imágenes (fondo y actual), calculada alrededor de una ventana cuadrada para cada píxel. Finalmente, la segmentación de frente se calcula introduciendo un umbral para la distancia obtenida según la siguiente ecuación:

$$F(I[x, y]) \Leftrightarrow \sum_{i=-W}^W \sum_{j=-W}^W (|I[x + i, y + j] - B[x + i, y + j]|)^2 > \beta$$

Donde W es una ventana cuadrada centrada en cada píxel, I es el frame actual, B es el modelo de fondo y β es el umbral para la segmentación de frente.

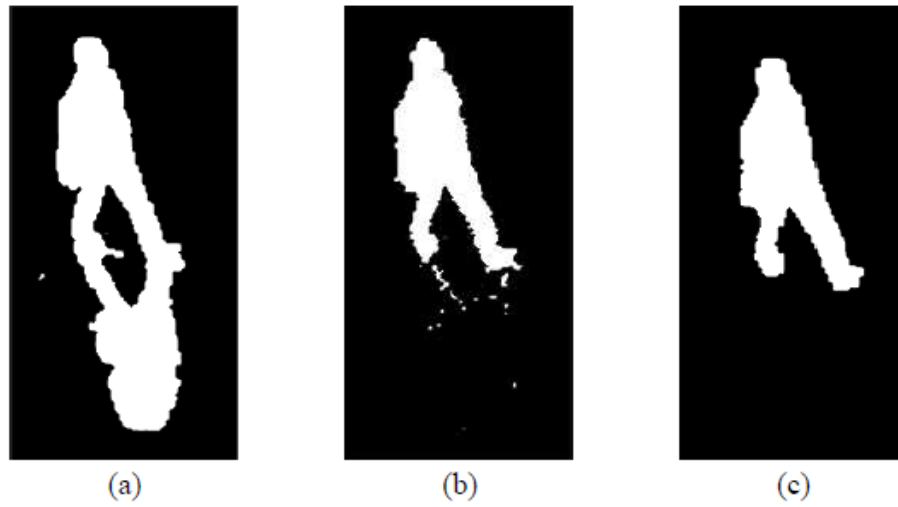


Figura B.2: Máscara de frente en diferentes etapas: máscara de frente inicial (a), después de la eliminación de la sombra (b) y después de la eliminación del ruido (c)

B.2.2. Eliminación de sombra (Shadow removal)

Las sombras proyectadas por los objetos y las personas se suelen clasificar erróneamente como parte del frente debido a su diferencia significativa con el modelo de fondo. Por lo tanto, las etapas de alto nivel de análisis, que toman como válida la información de las máscaras de frente (por ejemplo, el contorno del blob), se verán también afectadas cuando sombras adyacentes se consideran erróneamente como parte del objeto y por lo tanto, disminuye su rendimiento.

Una técnica de eliminación de sombra se aplica a la máscara de frente para la eliminación de los píxeles que pertenecen a las sombras producidas por entidades fijas (por ejemplo, objetos y personas).

Para este propósito, se utiliza el espacio de color HSV, ya que nos permite separar explícitamente entre cromaticidad y la intensidad. Este enfoque se aprovecha del hecho de que para sombras, el cambio de cromaticidad (tonalidad y saturación) entre la imagen actual y de fondo no es significativo. En la Figura B.2 se aprecian las máscaras obtenidas tras las distintas etapas.

La intensidad de la relación entre ambas imágenes también se calcula, para detectar los cambios de intensidad que son probablemente debido a la presencia de sombras. Para clasificar un píxel como parte de una sombra, se utiliza una siguiente función de decisión. Un ejemplo de eliminación de sombras se muestra en la Figura 3 1 b).

B.2.3. Eliminación de ruido (Noise removal)

Además, se realizan operaciones morfológicas sobre la máscara de frente resultante para la eliminación de ruidos. En particular, se aplica una combinación de la erosión y dilatación llamada. Su objetivo es eliminar los objetos pequeños (en nuestro caso blobs provocados por el ruido), al tiempo que conserva la forma y tamaño de todos los otros blobs en la máscara de frente.

La reconstrucción morfológica implica una imagen X (la máscara de frente para ser procesada), un marcador Y , y un elemento estructurante B . En el enfoque seleccionado, el marcador Y se calcula primero realizando la operación de erosión en X , y se obtiene entonces la máscara final, mediante la realización de dilatación iterativamente.

El tamaño y la forma del elemento estructurante determinará los ruidos que serán retirados de la máscara. En la Figura B.2 c) podemos ver un ejemplo de la operación descrita que se aplica a una máscara de frente ruidoso con un elemento estructurante cuadrado de 3 por 3.

B.2.4. Extracción del blob (Blob extraction)

Después de aplicar la sustracción de fondo y de obtener la máscara de frente, el módulo de extracción de blob etiqueta cada grupo aislado de píxeles de la máscara usando Análisis de Componentes Conectados (CCA). Después del proceso de etiquetado, algunas regiones muy pequeñas se pueden detectar. Esto puede ser debido a ruido que no fue eliminado correctamente por el módulo de eliminación de ruido, o debido a pequeños residuos como resultado de la segmentación errónea. Con el fin de evitar que los módulos de nivel superior arrastren estos errores y analicen regiones incorrectas, se descartan las que estén por debajo de un área determinada (en píxeles).

Apéndice C

Presupuesto

1. Ejecución Material

- Compra de ordenador personal (Software incluido)2.000€
- Alquiler de impresora láser durante 6 meses260 €
- Material de oficina 150 €
- Total de ejecución material 2.400 €

2. Gastos generales

- 16 % sobre Ejecución Material..... 352 €

3. Beneficio Industrial

- 6 % sobre Ejecución Material..... 132 €

4. Honorarios Proyecto

- 1800 horas a 15 € / hora27.000 €

5. Material fungible

- Gastos de impresión.....280 €
- Encuadernación 200 €

6. Subtotal del presupuesto

- Subtotal Presupuesto.....32.774 €

7. I.V.A. aplicable

- 21 % Subtotal Presupuesto6.882,5 €

8. Total presupuesto

- Total Presupuesto39.656,5 €

Madrid, marzo 2016

El Ingeniero Jefe de Proyecto

Fdo.: Ángel Mora Sánchez

Ingeniero de Telecomunicación

Apéndice D

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de *análisis de actividad en vídeos deportivos multicámara*. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho entorno. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado

en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean

oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.