



Trabajo Fin de Grado

Estudio del seguimiento de jugadores de baloncesto con una sola cámara en movimiento y con occlusiones mediante técnicas de visión por computador.

Autor

Carlos Dieste Orús

Director

Emiliano Bernués del Río

Escuela de Ingeniería y Arquitectura

2014

ESTUDIO DEL SEGUIMIENTO DE JUGADORES DE BALONCESTO CON UNA SOLA CÁMARA EN MOVIMIENTO Y CON OCLUSIONES MEDIANTE TÉCNICAS DE VISIÓN POR COMPUTADOR

RESUMEN

En este Trabajo Final de grado se estudian y comparan diferentes métodos con los que realizar el seguimiento de los jugadores sobre la pista de juego, a partir de un vídeo procedente de una sola cámara, para determinar la posición de los diez jugadores en cada imagen del vídeo.

Primero se realiza una revisión del estado del arte, en la que se estudian los algoritmos con los que se consiguen los mejores resultados en lo que respecta al seguimiento.

A continuación se comprueban las prestaciones de los métodos analizados mediante su implementación, realizando modificaciones sobre los mismos para mejorar en lo posible los resultados que se presentan en los artículos estudiados.

Agradecimientos

Me gustaría expresar mi reconocimiento a todas las personas que me han ayudado a lo largo de esta parte de mi trayectoria universitaria a evolucionar como alumno y como persona, ya que sin cada una de ellas no sería como soy ahora.

En primer lugar, me gustaría dar las gracias a mi tutor, Emiliano Bernués, que me propuso la realización de este trabajo y me dio la oportunidad de trabajar en un tema tan apasionante como el procesado de vídeo e imagen. También quiero agradecer a los profesores que he tenido durante estos cuatro años.

Agradecer tanto a mis amigos de Huesca como a los que he conocido en estos años en Zaragoza. Con todos he pasado momentos que no olvidaré.

Para terminar, gracias a mis padres por haberme ayudado y animado tanto durante toda mi vida.

Muchas gracias.

Índice de contenidos

RESUMEN -----	1
1 INTRODUCCIÓN -----	7
1.1 Motivación-----	7
1.2 Objetivos -----	7
1.3 Herramientas -----	8
2 ESTUDIO DEL ESTADO DEL ARTE -----	9
2.1 Seguimiento de objetos en movimiento-----	9
2.2 Extracción del fondo -----	10
2.3 Sistemas de seguimiento en partidos de baloncesto -----	12
3. DESARROLLO DEL TRABAJO -----	15
3.1 Introducción-----	15
3.2 Proceso de extracción del fondo o background.-----	15
3.3 Proceso para determinar la posición de los jugadores. -----	21
3.4 Reconocimiento y seguimiento del jugador -----	29
3.5 Solución que aporta mejores prestaciones -----	49
4 EXPLICACIÓN DEL SOFTWARE DESARROLLADO -----	51
5 CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO -----	53
5.1 Conclusiones -----	53
5.2 Líneas futuras de trabajo -----	55
6 REFERENCIAS -----	57
ANEXOS-----	59
Anexo 1 - Explicación del software desarrollado-----	61
Anexo 2 - Guía de usuario-----	65

1 Introducción

1.1 Motivación

El seguimiento de jugadores durante una competición deportiva implica un gran abanico de aplicaciones, tanto en lo que se refiere al análisis televisivo como al estudio de equipos rivales.

En lo referente al baloncesto, a lo largo de los últimos años han surgido algunos sistemas que, mediante el calibrado de hasta 6 cámaras de alta resolución instaladas en diversos puntos de un pabellón deportivo, son capaces de establecer y seguir la posición de cada uno de los jugadores sobre la pista de juego, gracias a que pueden obtener información acerca de la estructura tridimensional de la escena mediante diversas técnicas de visión por computador. El seguimiento de cada jugador puede realizarse estimando, a partir de diversas características presentes en una secuencia de imágenes obtenidas del vídeo proveniente de cada una de las cámaras, la trayectoria que sigue dicho jugador a lo largo de la secuencia del vídeo.

En el caso tratado en este trabajo final de grado, sin embargo, solamente se dispondrá de una única secuencia de vídeo de baja resolución. Estas circunstancias van a provocar que la secuencia de vídeo disponible contenga una única perspectiva, produciéndose occlusiones de algunos de los jugadores en determinados momentos del partido, lo que repercutirá en una mayor dificultad a la hora de distinguir un jugador de otro, especialmente en el caso de jugadores del mismo equipo.

1.2 Objetivos

En este trabajo final de grado se van a investigar y probar diferentes algoritmos del estado del arte que se utilizan para el seguimiento de jugadores, de forma aislada o combinada, para realizar el seguimiento de los jugadores de baloncesto presentes en la pista de juego a lo largo del partido. Se estudiarán algoritmos como el filtro de Kalman, el filtro de partículas o el algoritmo de condensación, los cuales ofrecen muy buenas prestaciones en circunstancias concretas. A continuación, se expondrá el algoritmo empleado para implementar dicho seguimiento, teniendo en cuenta la calidad y resolución del vídeo disponible, y por último, se realizará una mejora de las prestaciones del sistema atendiendo a las posibles occlusiones que puedan ocurrir.

1.3 Herramientas

Para probar las prestaciones de los algoritmos estudiados y las mejoras que se puedan plantear a los sistemas analizados, se va a utilizar la herramienta Microsoft Visual Studio 2010, ya que permite realizar sencillas interfaces con las que probar los algoritmos implementados. Se programará en el lenguaje C#, ya que es un lenguaje de programación intuitivo y sencillo de aprender a utilizar. Además, se van a emplear las librerías de AForge y Emgu CV, específicas para el procesado de imágenes si se utiliza el lenguaje de programación C#. En concreto, Emgu CV es una plataforma que permite al programador utilizar las librerías de procesado de imagen de OpenCV desde lenguajes .NET como C#, Visual Basic o IronPython.

2 Estudio del estado del arte

Uno de los principales retos a la hora de afrontar este Trabajo Final de Grado es encontrar un algoritmo que sea capaz de realizar el seguimiento de los jugadores en situaciones de ataque estático, y que además se comporte de forma robusta en las situaciones, frecuentes en un partido de baloncesto, en que se producen cruces y occlusiones entre los jugadores.

A continuación, se van a presentar algunos de los algoritmos y técnicas mediante las cuales se obtienen mejores prestaciones en lo que se refiere al seguimiento y detección de objetos, y que por extensión podrían utilizarse para el seguimiento de jugadores en un evento deportivo, y que utilizan una sola cámara.

2.1 Seguimiento de objetos en movimiento

Existen numerosos algoritmos que pretenden situar un objeto en una imagen a partir de la posición de éste en las imágenes anteriores. Se basan en técnicas recursivas (filtro de Kalman), métodos estocásticos (filtro de partículas) o detección de contornos (algoritmo de condensación).

Filtro de Kalman

Este algoritmo [1] estima de forma recursiva el estado de un sistema con dinámica lineal a partir de una serie de medidas ruidosas. Uno de sus múltiples usos es el seguimiento de objetos: predecir su posición y reducir el ruido introducido por estimaciones incorrectas.

En procesado de imagen, el filtro de Kalman predice la posición de un objeto en la siguiente imagen a partir de la posición del objeto en la imagen anterior y del error cometido al realizar la estimación en la iteración anterior, suponiendo que dicho ruido es gaussiano de media nula.

Filtro de partículas

Esta técnica utiliza estimación a posteriori a partir de un conjunto de observaciones, llamadas partículas. A dichas partículas se les asigna un peso que representa la probabilidad de que dichas observaciones se ajusten o no a la función de densidad de probabilidad que se ha establecido como modelo de color del objeto. Seguidamente, se obtiene un nuevo centroide a partir del conjunto de partículas y sus pesos, que representa la nueva posición del objeto en la imagen. [2]

Algoritmo de condensación

Este método es un caso particular de filtro de partículas para estimar el contorno de un objeto en movimiento. A lo largo de una secuencia de imágenes, la función de densidad de probabilidad de color del contorno de un objeto varía de forma determinista y estocástica al mismo tiempo. Dicho contorno se puede seguir utilizando un modelo dinámico para el movimiento del objeto, basado en ecuaciones diferenciales, que debe adquirirse mediante una fase previa de entrenamiento. [3]



Figura 1 - Imagen en que se observa cómo se ha detectado el contorno de una mano mediante el algoritmo de condensación

En el seguimiento de los jugadores en un partido de baloncesto, sin embargo, alguno de los sistemas de seguimiento mencionados anteriormente, como el filtro de Kalman, no es válido, ya que asume que el movimiento de los jugadores en la escena va a ser lineal, cosa que no ocurre frecuentemente, ya sea por cambios de dirección o de velocidad de cada jugador. Por tanto, para el seguimiento en un partido de baloncesto se utilizarán varias técnicas combinadas para obtener resultados óptimos.

2.2 Extracción del fondo

Para ayudar al algoritmo de seguimiento que se vaya a utilizar, se emplea en la mayoría de los casos un proceso de segmentación, que consiste en separar la parte de la imagen que forme parte del fondo (background) de la parte de la imagen compuesta por el conjunto de los jugadores. En los artículos estudiados se emplean tanto la estimación de un modelo de fondo como la detección y substracción del color dominante.

Estimación del fondo de una secuencia de imágenes

En este método se obtiene, a partir de una secuencia de imágenes adquiridas, una función de densidad de probabilidad tridimensional, en el que cada dimensión corresponde a una de las componentes de color, para cada uno de los píxeles de la imagen. De esta forma, aunque haya jugadores que estén en una zona de la imagen en cierto momento, al realizar una estimación a lo largo del tiempo, será más común el color de la pista de juego que el del jugador en el histograma de cada píxel.

A pesar de ser el método de estimación del fondo que parece más robusto, esta técnica solamente funcionará correctamente en situaciones en que no se produzcan movimientos de cámara, tanto físicos como ópticos.

Detección y substracción de color dominante

Es el más utilizado, ya que el color del campo de juego no es uniforme en la mayoría de los deportes, y varía según las condiciones climáticas o la iluminación.

El método propuesto en [4] se basa en la uniformidad del color del campo de juego en la mayoría de los deportes, y por tanto el fondo puede caracterizarse por un único color dominante, que varía temporalmente, adaptándose a las condiciones de luminosidad de la imagen. Cada píxel de la imagen se clasifica entre jugadores o fondo según una métrica de distancia euclídea (en el caso del espacio de color RGB) o una métrica de distancia cilíndrica (en el caso del espacio de color HSV) entre el color dominante y el color de cada píxel en la imagen. De esta forma, se obtiene una imagen máscara, que se podrá aplicar a la imagen original para obtener solamente la silueta de los jugadores en la imagen.



Figura 2 - Máscara resultado de aplicar este algoritmo sobre una imagen de un partido de fútbol

Una máscara es una imagen en blanco y negro. Si se aplica una máscara a una imagen, todo píxel que en la máscara sea blanco, en la imagen resultante conserva el color de la imagen original, y todo píxel que en la máscara sea negro, en la imagen resultante pasa a ser de color negro.

2.3 Sistemas de seguimiento en partidos de baloncesto

En cuanto al seguimiento en partidos de baloncesto, se han desarrollado numerosos trabajos de investigación, algunos de los cuales han obtenido resultados muy buenos en cuanto al porcentaje de tiempo que el seguimiento se realiza de forma correcta.

Seguimiento utilizando una sola cámara

El sistema que se expone en [5] realiza un seguimiento automático de múltiples jugadores. Para comenzar, se realiza una detección de jugadores utilizando el detector que utiliza el modelo de partes deformables (DPM, Deformable Part Model), desarrollado en [6] para localizar de forma automática a los jugadores.

A continuación, se realiza una clasificación de los jugadores según los equipos, realizando histogramas de color en el espacio RGB para seguidamente entrenar el clasificador que separa los jugadores en equipos.

Por último, se realiza el seguimiento de los jugadores siguiendo un modelo de regresión lineal para el movimiento de los jugadores, que no tiene en cuenta el movimiento del jugador en la imagen anterior. Dado que se utilizan modelos lineales para la posición y para el movimiento, se actualiza la posición de los jugadores usando filtros de Kalman.



Figura 3 – Pasos que sigue el algoritmo: detección, clasificación por equipos y seguimiento

La razón de que funcione correctamente a pesar de utilizar un modelo lineal es que existe una relación lineal entre un instante de tiempo y el estado (posición y velocidad) del jugador en ese instante.

Con este sistema se consigue un seguimiento correcto en el 81% del tiempo. [5]

Seguimiento utilizando múltiples cámaras

Si se dispone de varias secuencias de vídeo provenientes de diferentes cámaras, se puede combinar la información de todas las secuencias disponibles, como se propone en [7].

Primero se realiza un preprocessado, dividiendo el vídeo en secuencias más pequeñas mediante un método para detectar cambios de plano, basado en diferencia de histogramas [8]. De las secuencias obtenidas, solamente se conservan las que tengan una duración adecuada para considerarse jugadas del partido. A continuación, se crea una máscara binaria detectando el color dominante en la escena y se aplican sobre ésta operaciones morfológicas de erosión y dilatación para eliminar el posible ruido y para definir las siluetas de los jugadores.

Después del preprocessado, se procede a realizar la detección de la pista y la calibración de las cámaras. La calibración consiste en una transformación geométrica de las coordenadas de la imagen en coordenadas de un plano de la pista. Esta operación se denomina homografía. [9]

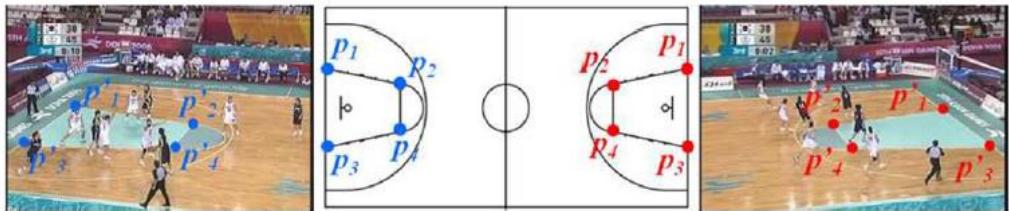


Figura 4 – Imágenes que muestran los puntos de referencia a tener en cuenta para realizar la homografía

Seguidamente, se realiza el seguimiento de los jugadores mediante un algoritmo basado en CamShift [10] [11]. Dicho algoritmo se aplica en varios pasos:

- I. Se agrupan los píxeles de las regiones en que hay jugadores en 2 clústeres mediante el algoritmo de clustering K-means, basándose en la información de color, de forma que cada clúster constituya una distribución de probabilidad de color de uno de los dos equipos.
- II. Se genera el algoritmo de seguimiento (también llamado tracker) para todos los jugadores, y se establece un área de seguimiento (por ejemplo un rectángulo) para cada uno. Se obtiene un modelo (histograma) de color para los píxeles interiores al rectángulo y se aplica un algoritmo mean shift para encontrar el máximo del histograma.
- III. Para cada uno de los trackers, se aplica el algoritmo de seguimiento CamShift. Este algoritmo busca la posición del rectángulo del jugador en la siguiente imagen que maximice la probabilidad de que el jugador se encuentre dentro del rectángulo utilizando la distribución de probabilidad en la dirección del gradiente.

A continuación, se actualiza la distribución de probabilidad de color del jugador.

Cuando un jugador sale fuera de la imagen, los valores de probabilidad dentro del rectángulo de seguimiento disminuyen mucho. Por ello, se aplica un umbral para determinar si el jugador ha salido de la imagen.

Con este algoritmo se consigue una precisión de hasta el 84 %. [7]

3. Desarrollo del trabajo

3.1 Introducción

El objetivo de este trabajo final de grado es automatizar el seguimiento de los jugadores a lo largo de una posesión en un partido de baloncesto. Este proceso se simplifica conforme aumenta el número de secuencias de cámara utilizadas y cuanto mejor es la calidad de la imagen. Sin embargo, en este trabajo final de grado se va a emplear para realizar las pruebas una única fuente de vídeo, con baja resolución. Además, al emplear una sola secuencia de vídeo, se producirán cruces entre jugadores, que complicarán en gran medida el proceso de tracking.

Se debe mencionar que este trabajo final de grado se desarrolla en paralelo con otro, titulado "Extracción de jugadas tipo, en partidos de baloncesto, usando procesado digital de imagen". En éste se propone un análisis automático de ciertas jugadas realizadas por los jugadores en un partido. Este proyecto determina los objetivos que se deben obtener en el proyecto que se describe en esta memoria, ya que deben obtenerse las posiciones de los pies de cada uno de los jugadores sobre la pista a lo largo de la posesión completa. Estas posiciones serán utilizadas en el trabajo final de grado mencionado anteriormente para poder describir las jugadas.

Para alcanzar el objetivo del trabajo final de grado, se va a proceder de la siguiente forma: Dada una secuencia de vídeo de posesión completa de un equipo, primero se realizará la segmentación de los jugadores con un algoritmo de extracción del fondo. A continuación, se situarán los jugadores en la imagen y por último se realizará el seguimiento de cada uno de ellos imagen a imagen.

3.2 Proceso de extracción del fondo o background.

Este procesado, como se ha mencionado en el apartado en que se habla sobre el estado del arte, sirve en primer lugar para facilitar la tarea de situar los jugadores en la imagen y después para ayudar al algoritmo de seguimiento a encontrar a cada uno de los jugadores.

3.2.1 Filtrado de color

Como primera aproximación, se ha realizado un filtrado de color en el espacio HSV, ajustando de forma manual cada componente de color, para eliminar los píxeles de la pista.

El espacio de color HSV (hue, saturation, value) [12][13][14], también llamado HSB (hue, saturation, brightness), es una representación de color alternativa a RGB. Tiene tres componentes:

- Matiz: representa el color. Es un ángulo comprendido entre 0 y 360º.
- Saturación: indica la pureza del color respecto a una referencia de blanco. Normalmente toma valores entre 0% y 100%. Cuando la saturación es 0%, el color es gris o blanco. Cuando la saturación es 100%, el color es un color primario.
- El valor o brillo es la percepción de la cantidad de luz que contiene el color. Si el color refleja mucha luz, se diría que es brillante. Al igual que la saturación, se representa mediante porcentajes. Cuando el brillo es 0%, el color será negro. A medida que se incrementa el valor, el color se vuelve más claro.

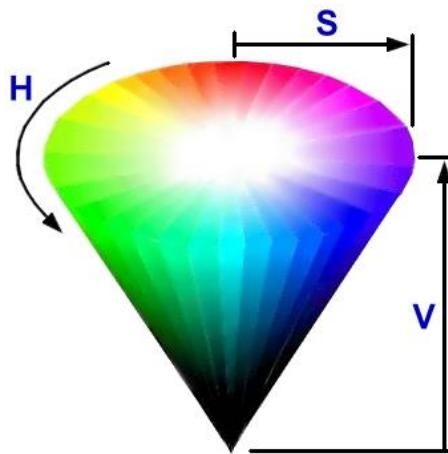


Figura 5 – Representación esquemática del espacio de color HSV

Para realizar el filtrado de color se definen una tripleta de valores de color máximo y mínimo:

$$HSV_{min}: (H_{min}, S_{min}, V_{min}) \quad (1)$$

$$HSV_{máx}: (H_{máx}, S_{máx}, V_{máx}) \quad (2)$$

A continuación, se comprueba para cada píxel si cada una de las tres componentes de color del píxel, están comprendidas entre los valores máximo y mínimo del filtro, es decir, si se cumplen las siguientes condiciones:

$$H_{min} < H_i < H_{máx} \quad (3)$$

$$S_{min} < S_i < S_{máx} \quad (4)$$

$$V_{min} < V_i < V_{máx} \quad (5)$$

Donde (H_i, S_i, V_i) son las componentes de color del pixel i en el espacio de color HSV.

La imagen resultante será una máscara, en la que los píxeles de color blanco serán aquellos que cumplan las tres condiciones en la imagen original, y los píxeles de color negro serán aquellos que no cumplan las condiciones en la imagen original.

Si se cumplen las tres condiciones, se convierte el color del píxel a blanco. En caso contrario, el color del píxel se convierte en negro.



Figura 6 – Ejemplo de filtrado de color, en que se ha intentado eliminar todo aquello que no sea el rojo de las camisetas de los jugadores de equipación roja.

Se han utilizado estos valores límite para el filtrado para el ejemplo anterior:

$$\text{HSV}_{\min} : (131, 133, 46)$$

$$\text{HSV}_{\max} : (255, 255, 184)$$

Una de las ventajas del espacio de color HSV (Hue, Saturation, Value), que provoca que la métrica utilizada sea eficaz para la extracción del fondo, es la similitud de esta representación de color con la percepción del color que realiza el ser humano. Otros espacios de color como RGB (Red, Green, Blue) o CMYK (Cyan, Magenta, Yellow, black) definen las componentes de color a partir de los colores primarios, sin tener en cuenta la percepción humana del color.

De esta forma, se va a aprovechar esta propiedad de la representación de color HSV para ajustar los valores máximo y mínimo del filtro. Para ello, se ha desarrollado una interfaz gráfica con que modificar cada componente de color de los valores límite por separado de las demás componentes.

La interfaz muestra a la izquierda la imagen original y a la derecha la imagen a la que se ha aplicado la máscara resultante en el filtrado de color.

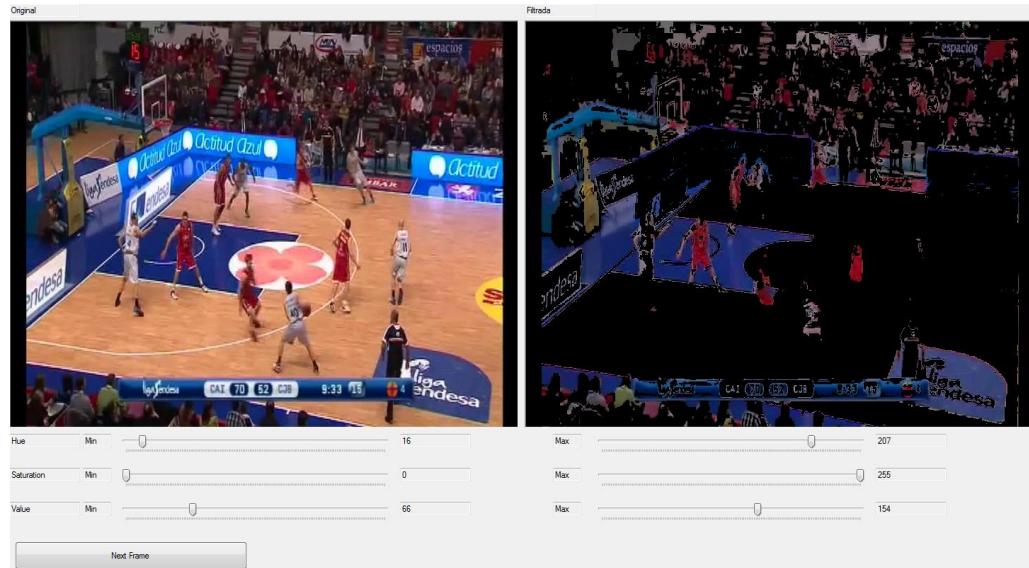


Figura 7 – Imagen que muestra la interfaz en la que se eligen los valores límite del filtrado de color

Este método de extracción de fondo se comporta como se esperaba: es capaz de eliminar la parte de la pista que se deseé, modificando los extremos del filtro.

Sin embargo, este procedimiento no sirve si se quiere automatizar la extracción de fondo, ya que las condiciones de iluminación de la pista pueden variar a lo largo del partido, e incluso a lo largo de una posesión. Por este motivo, se buscó otro método de extracción del fondo, que ajustara de forma dinámica, imagen a imagen los valores del filtro de color.

3.2.2 Detección y extracción del color dominante

El algoritmo utilizado se basa en el método nombrado en [4]. Una vez adquirida la imagen o frame, se obtiene un histograma tridimensional de color en el espacio de color HSV, basándose en las veces que se repite una tripleta de color en los píxeles de la imagen.

Se utilizará un histograma con 32 bins por cada dimensión. Se debe considerar que para cada componente de color de cada píxel se utilizan 256 valores. A partir del histograma de color de la imagen, se busca el valor de color dominante, es decir, el color que más se repite en la imagen.

Con el color dominante, se clasifica cada píxel entre jugadores (foreground) o fondo (background) calculando la distancia entre el color del píxel a clasificar y el color dominante en la imagen.

Se utilizará una métrica cilíndrica:

$$d_V(j, k) = |V_j - V_k| \quad (6)$$

$$d_{cromática}(j, k) = \sqrt{S_j^2 + S_k^2 - 2S_j S_k \cos(H_j - H_k)} \quad (7)$$

$$d_{cilíndrica}(j, k) = \sqrt{(d_V(j, k))^2 + (d_{cromática}(j, k))^2} \quad (8)$$

Donde H, S y V son las componentes de color en el espacio de color HSV.

La razón por la cual se utiliza esta métrica cilíndrica junto con la representación de color HSV es la robustez de esta combinación, corroborada en [4].

Dado que la pista de baloncesto no es de un único color, ya que normalmente el color de la zona es diferente del color del parqué, se debe aplicar este algoritmo recursivamente hasta conseguir extraer lo máximo posible del fondo de la imagen. En las siguientes figuras se aprecia la diferencia entre aplicar una o tres iteraciones del algoritmo descrito.



Figura 8 – Imagen original



Ilustración 9 – Imagen después de una iteración



Figura 10 – Imagen después de 3 iteraciones

Con este algoritmo se consigue extraer la pista en su mayor parte. Sin embargo, la valla de publicidad y el público no son eliminados, debido a la diferencia de color de éstos con respecto al marrón de la pista de juego o al azul de la zona.

3.3 Proceso para determinar la posición de los jugadores.

Para utilizar el algoritmo de seguimiento, se debe definir previamente la superficie de la imagen en la que está situado cada jugador. Esta superficie será un rectángulo que contenga la camiseta y el pantalón de cada uno de los jugadores.

3.3.1 Definición del rectángulo del jugador de forma manual

Para realizar diferentes pruebas, en primer lugar se decidió que el rectángulo de cada jugador fuera determinado realizando dos clics con el ratón en la imagen. Este proceso, aunque preciso, no concordaba con el objetivo de automatizar el seguimiento de los jugadores, así que se investigaron otras formas de determinar los rectángulos que contienen a los jugadores.

3.3.2 Definición del rectángulo utilizando filtros de color

Se decidió que la mejor forma de automatizar el proceso era determinar la zona de la imagen que corresponde a un jugador, para a continuación obtener el rectángulo más pequeño que contenga al jugador.

La forma más evidente de diferenciar la camiseta y el pantalón de los jugadores es realizar un filtrado de color. Para ello, se debe obtener de alguna forma el rango de colores de las equipaciones de los equipos.

I Filtrado de color

Filtrado de color fijo

En principio, el proceso en que se determinan los valores extremos del filtro se realizó utilizando la misma interfaz empleada para realizar el filtrado de color de la pista (página 18), obteniendo los resultados deseados. No obstante, el color de los jugadores de color blanco y rojo no era exactamente el mismo en todas las secuencias de juego. Por ello, los límites para el filtro de color definidos para una secuencia determinada no tenían por qué servir para otras secuencias. Además, una vez que se determinaban los valores extremos del filtro no se podían variar a no ser que se modificara el código.

Filtrado de color basado en entrenamiento

Al ser necesario automatizar el proceso de detección de los jugadores y hacerlo independiente de la secuencia del partido que se esté analizando, se estudió la variación de las componentes de color de los jugadores a lo largo del tiempo.

Variación del color de los jugadores

Seguidamente se muestran la componente de matiz (Hue) de diferentes jugadores con uniformes de color rojo escogidos al azar, cada 125 imágenes.

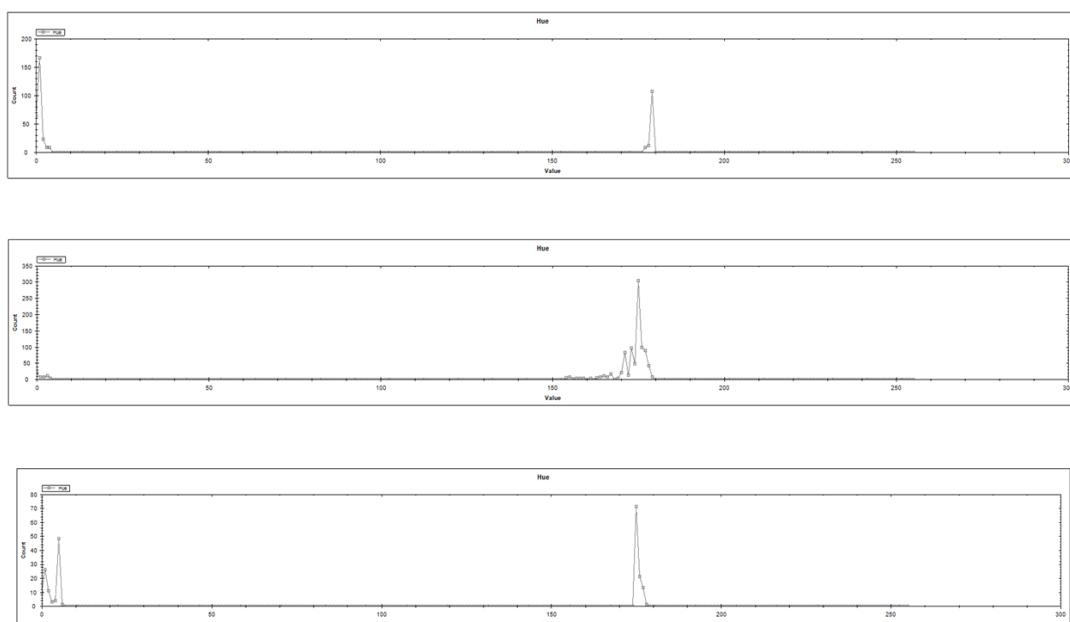


Figura 11 – Se puede observar que existe un máximo en la componente de matiz cerca de 175

A continuación se representan las componentes RGB de esos mismos jugadores.

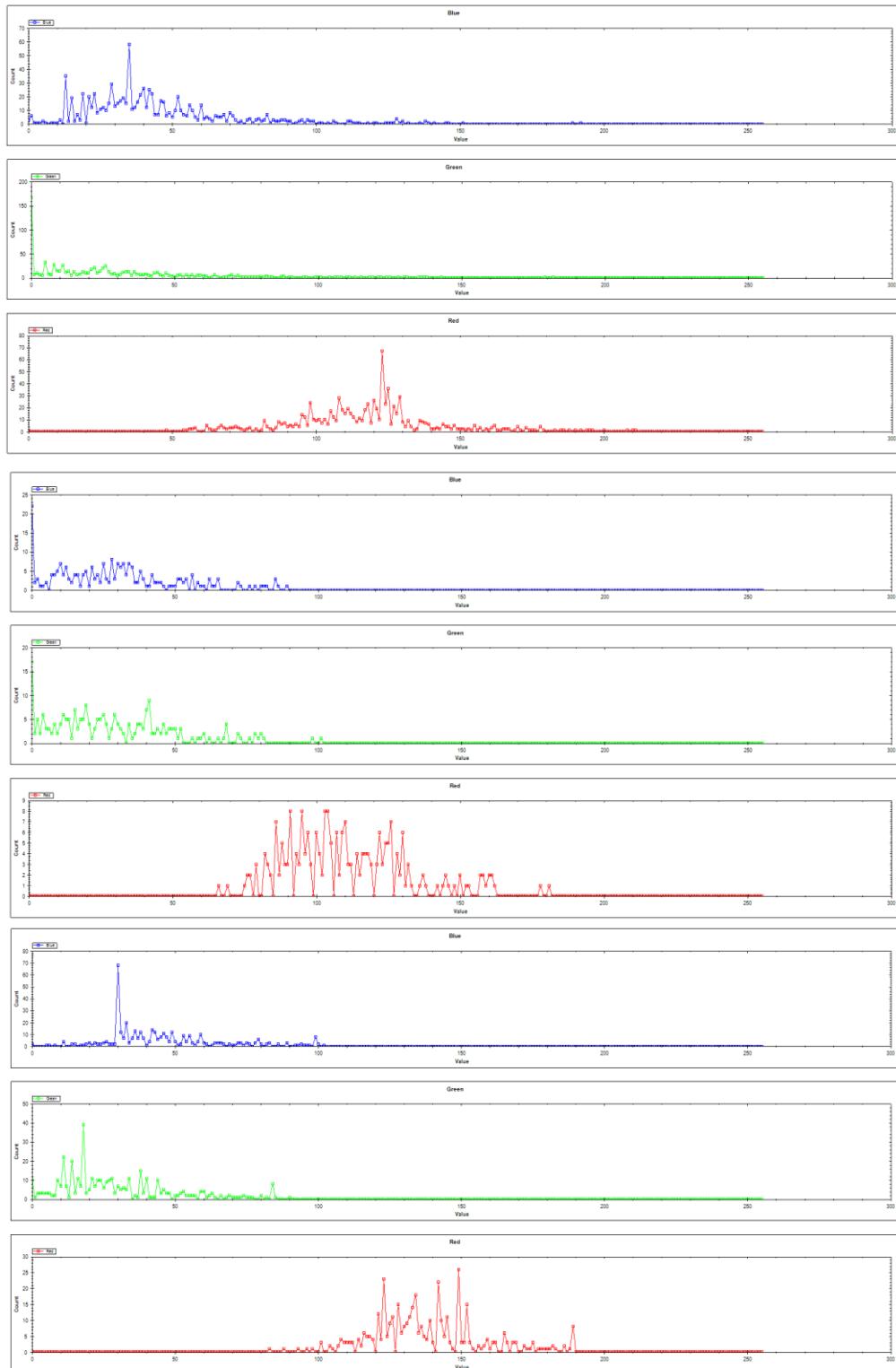


Figura 12 – Se muestran los histogramas de las componentes de azul (gráficas de color azul), verde (gráficas de color verde) y rojo (gráficas de color rojo), en que el eje de valores del color varía entre 0 y 255

De las gráficas obtenidas se puede deducir que existe un valor para cada componente de color en cuyo entorno se encuentra la mayor parte del color de los jugadores que llevan camiseta y pantalón rojos, y que se mantiene a lo largo del tiempo. Para los jugadores que llevan equipación blanca ocurre algo semejante.

Por tanto, se podría realizar un entrenamiento previo para extraer las componentes de rojo, azul, verde y el matiz de los jugadores de cada equipo.

Entrenamiento del sistema y modelado de los jugadores

Para obtener un modelo para los jugadores de color rojo y otro para los jugadores de color blanco, se realiza un histograma para cada componente de color a partir de varios rectángulos que contienen jugadores de cada equipo, por separado. Estos rectángulos se seleccionan de forma manual, realizando dos clics con el ratón.

Filtrado a partir del modelo de los jugadores

De cada histograma modelo se obtendrá el valor máximo para cada componente de color, y se realizará un filtrado de color alrededor de este valor. Se realizaron diferentes pruebas para establecer el rango de colores permitido alrededor del máximo para cada componente de color a la hora de realizar el filtrado. Aplicando este filtrado, se obtiene la imagen siguiente:



Figura 13 – En estas dos imágenes se puede ver que la precisión del filtrado de color tras haber realizado el entrenamiento es aceptable ya que la máscara se ajusta bastante bien a las zonas de la imagen que corresponden con un jugador

II Obtención de los contornos de la imagen filtrada

Una vez obtenida la máscara con los jugadores, se adquieren los contornos a partir de esta máscara. Los contornos son curvas que encierran todos los píxeles que tienen el mismo color o intensidad. Como en este caso se extraen los contornos de una imagen en blanco y negro, los contornos separarán las zonas de píxeles de color blanco de las zonas de píxeles de color negro. El método utilizado para obtener los contornos se basa detectando las diferencias de color entre píxeles adyacentes. Se realiza mediante la función de Emgu CV `findContours()`.



Figura 14 – La imagen inferior muestra los contornos después de aplicar un filtrado de color a la imagen superior

III Obtención de los rectángulos de los jugadores

Una vez obtenidos los contornos, se adquiere para cada contorno el rectángulo de área mínima que lo contiene. Se realiza mediante la función de Emgu CV `minAreaRect()`. [15]

A continuación, se descartan aquellos rectángulos que no tengan un área suficientemente grande, y aquellos cuya relación entre la anchura y la altura del rectángulo no sea adecuada, para descartar la mayor parte de falsos positivos.



Figura 15 – Como se puede apreciar, el filtrado de color detecta, además de los jugadores, otras zonas de la imagen que no contienen un jugador, y pasa por alto jugadores que deberían ser detectados

Para paliar este hecho se ha desarrollado una interfaz sencilla que permite descartar los falsos positivos y añadir jugadores (rectángulos) en el caso de no detección, mediante clics del ratón en la imagen.

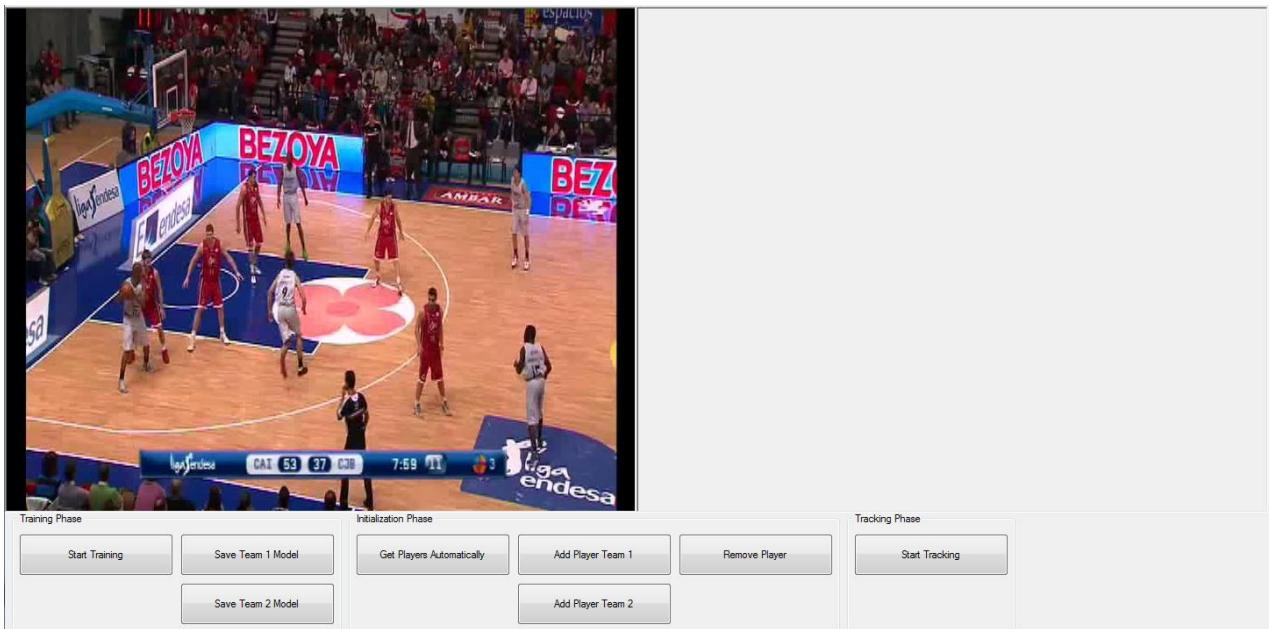


Figura 16 – Interfaz desarrollada para controlar el proceso del sistema de seguimiento

3.4 Reconocimiento y seguimiento del jugador

Una vez obtenidos los rectángulos que contienen la camiseta y el pantalón de cada uno de los jugadores, se procede a realizar el seguimiento de cada rectángulo.

3.4.1 Algoritmo basado en estimación de movimiento

Para comenzar, se utilizó un algoritmo basado en estimación de movimiento, es decir, el método de compresión de vídeo que utiliza el estándar MPEG. Se intentará que el movimiento del rectángulo imite el movimiento del jugador, desplazando el rectángulo del jugador según los llamados *vectores de movimiento*.

I Obtención de los vectores de movimiento

Primero se divide la zona de la imagen interior al rectángulo del jugador en bloques (en este caso se han escogido bloques de 10 x 10 píxeles).

A continuación, para cada uno de los bloques, se busca en la imagen anterior la zona que más se asemeja, asumiendo que el bloque se ha movido. La búsqueda se realiza en los 30 píxeles alrededor del bloque suponiendo que el movimiento no es muy grande al estar procesando una secuencia de vídeo de 25 imágenes por segundo.

El *vector de movimiento* se definirá como la diferencia de posición entre la posición del bloque en la imagen anterior y la posición de la zona más parecida a ese bloque en la imagen siguiente.

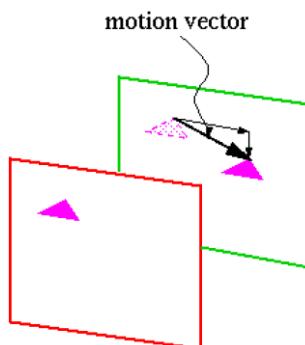


Figura 17 – Descripción esquemática del vector de movimiento

II Desplazamiento del rectángulo

Una vez obtenidos todos los vectores de movimiento, se realiza un histograma bidimensional con todos los desplazamientos, horizontales y verticales, de los vectores de movimiento. A continuación, se obtiene, mediante ese histograma, el vector de movimiento que más se repite, es decir, el que debería definir en qué medida se ha movido el jugador.



Figura 18 – El rectángulo se desplazará en la dirección que corresponda con la dirección de la mayoría de los vectores de movimiento

III Eliminación del movimiento del fondo

En la mayoría de las ocasiones, sin embargo, el rectángulo contiene, además de la equipación del jugador, píxeles del fondo (la pista), es decir, píxeles que no tienen movimiento.

Para evitar la influencia de los bloques que contienen píxeles del fondo que no han podido eliminarse mediante la extracción del fondo, se obtienen los vectores de movimiento del entorno del jugador. Así se consigue descartar el movimiento del fondo (la pista) en el rectángulo del jugador, en caso de que éste predomine sobre el movimiento del jugador, para estimar de manera más acertada hacia dónde se mueve el jugador.



Figura 19 – Se calculan los vectores de movimiento de alrededor del rectángulo del jugador para realizar una estimación del movimiento de la pista de juego (movimientos de cámara, de otros jugadores...)

IV Corrección de la estimación utilizando la imagen de movimiento

A partir de este algoritmo basado en estimación de movimiento se planteó una mejora que tenía en cuenta el movimiento de los jugadores para corregir la posición del rectángulo después de calcularla mediante estimación de movimiento.

La imagen de movimiento

La imagen de movimiento es la imagen diferencia entre una imagen y la anterior.

Para calcular la imagen de movimiento, se realiza la resta píxel a píxel entre la imagen actual y la imagen anterior de las tres componentes del espacio de color RGB:



Figura 20 – Imagen diferencia

A continuación, se realiza una conversión del espacio de color RGB a escala de grises, es decir el valor de cada píxel de la imagen resultante será el valor de la luminancia:

$$Y = R \cdot 0.3 + G \cdot 0.59 + B \cdot 0.11 \quad (9)$$

Donde Y es el valor del píxel en la imagen resultante, y R, G y B son las tres componentes de color de cada píxel de la imagen original en el espacio RGB.

A continuación, se descartan todos aquellos píxeles que no tengan un movimiento suficientemente grande, es decir, que su valor en la imagen diferencia sea pequeño. Se aplica un umbral, de tal forma que todo píxel que tenga un valor que supere el umbral, se transformará a blanco. En caso contrario, se transformará en color negro.



Figura 21 – Imagen una vez aplicado el umbral

Como en la imagen las zonas blancas donde hay movimiento contienen píxeles negros y las zonas donde no hay movimiento contienen píxeles blancos, se deben aplicar técnicas morfológicas como son los filtros de erosión y dilatación, que eliminan el ruido no deseado y establecen zonas negras y blancas bien definidas.

Los filtros de erosión y dilatación

Las operaciones morfológicas tales como la erosión y la dilatación son un tipo de procesado de imagen que ayuda a definir formas y conectar zonas de la imagen, utilizando operaciones lógicas. Se aplican por lo general a imágenes en blanco y negro, comúnmente llamadas imágenes binarias.[16][17]

Tanto la operación de erosión como la de dilatación utilizan un elemento cuadrado de 3x3 píxeles, llamado elemento estructurante, cuyo origen es el píxel central.

1	1	1
1	1	1
1	1	1

Figura 22 – Elemento estructurante

Erosión

La operación de erosión se utiliza para eliminar el ruido presente en una imagen. [19]

Para aplicar erosión a una imagen binaria, se superpone el elemento estructurante a cada píxel blanco de la imagen original, de forma que el origen del elemento estructurante coincide con las coordenadas del píxel de la imagen de entrada. Si para todos y cada uno de los píxeles del elemento estructurante su correspondiente píxel en la imagen original es de color blanco, en la imagen resultante el color del píxel no se modifica. En caso contrario, el color del píxel se transforma en negro. Es decir, todo píxel de color blanco de la imagen original que no esté rodeado de píxeles de color blanco se transformará en la imagen resultante en un píxel de color negro.

A continuación, se expone un ejemplo de la aplicación del filtro de erosión:

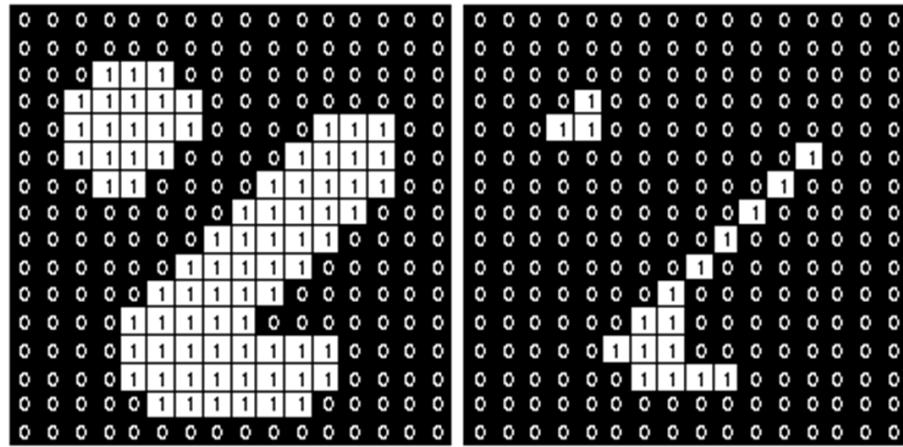


Figura 23 – Ejemplo de aplicación de la operación de erosión

Dilatación

La operación dual de la erosión se denomina dilatación. [20]

Para aplicar la operación de dilatación, se superpone el elemento estructurante en cada uno de los píxeles de color negro de la imagen original, haciendo corresponder el origen del elemento estructurante con el píxel sobre el que se está aplicando la operación, del mismo modo que en la operación de erosión. Si al menos un píxel del elemento estructural corresponde con un píxel de color blanco en la imagen original, el color del píxel será blanco en la imagen resultante. Es decir, todo píxel de color negro en la imagen original que tenga al lado un píxel de color blanco, se convertirá en un píxel de color blanco en la imagen resultante.

A continuación, se expone un ejemplo de la aplicación del filtro de dilatación:

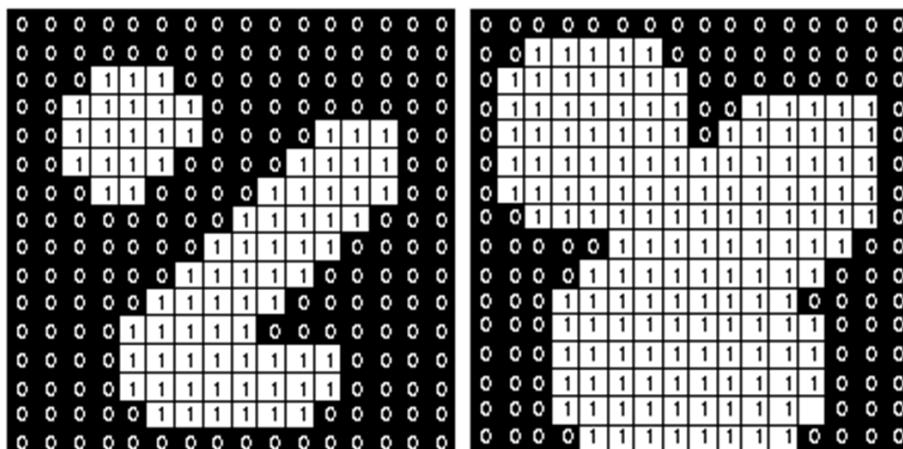


Figura 24 – Ejemplo de aplicación de la operación de dilatación

Aplicando repetidas veces las operaciones de erosión y dilatación, se consigue eliminar el ruido y conectar las zonas de color blanco de la imagen. En la siguiente figura se muestra el resultado de utilizar los filtros de erosión y dilatación sobre la imagen de movimiento.

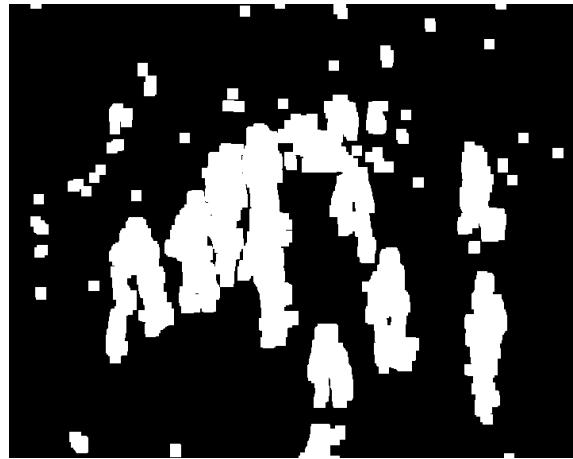


Figura 25 – Ejemplo de aplicación de operaciones morfológicas sobre la imagen binaria

Una vez procesada la imagen de movimiento mediante operaciones morfológicas, se desplaza el rectángulo resultante de aplicar la estimación de movimiento, buscando la posición de éste que contenga la mayor cantidad de píxeles de color blanco, comparado con la cantidad de éstos en el entorno del rectángulo del jugador. La búsqueda se realiza en un pequeño rango de desplazamientos, ya que se asume que la estimación de la posición del rectángulo mediante vectores de movimiento ha sido aceptable.



Figura 26 – Se busca que la relación entre los píxeles de color blanco del interior del rectángulo pequeño en comparación con los píxeles de color blanco del interior del rectángulo grande sea la menor posible

La posición del rectángulo que resulte óptima se utilizará para realizar la estimación de movimiento en la imagen siguiente.

Este algoritmo, sin embargo, no tiene en cuenta el color de la equipación, propiedad que puede ser útil en los casos en que se produzca una oclusión entre dos jugadores de distinto equipo. Además, cuando dos jugadores están próximos entre sí, en la imagen de movimiento se aprecian como uno solo, y el algoritmo planteado no actúa como se esperaba.

Al no haber utilizado hasta el momento un algoritmo que utilizara el color del jugador para realizar el seguimiento y al no haber encontrado un método robusto para realizar el seguimiento del jugador, se decidió utilizar un algoritmo de crecimiento, similar al que se utiliza en [21].

3.4.2 Seguimiento mediante algoritmo de crecimiento

Este algoritmo consiste en, a partir de un píxel, recrear la silueta del jugador según las componentes de color del uniforme de éste, para después obtener el rectángulo del jugador.

I Inicialización

Previo al inicio del algoritmo, se obtiene un histograma de cada una de las tres componentes de color en el espacio RGB, además de la componente de matiz (Hue) del espacio HSV, a partir de los píxeles interiores al rectángulo del jugador en la primera imagen. La razón de utilizar estos histogramas se explica en la página 22. Se utilizan histogramas de 64 bins.

A continuación, se obtienen los valores máximos para los cuatro histogramas.

Para el desarrollo del algoritmo, se va a utilizar un conjunto de posiciones de los píxeles de la imagen que se deben probar. Ese conjunto se inicializa como vacío.

Una vez adquirida la imagen correspondiente, se escoge un píxel de la imagen a partir del cual comenzar el crecimiento. Ese píxel corresponde a la posición central del rectángulo del jugador. Dicho píxel se añade al conjunto de píxeles que se deben probar.

II Algoritmo recursivo. Crecimiento a partir de un píxel

Mientras el conjunto de posiciones a probar no esté vacío (mientras no haya más posiciones que probar), se repiten los siguientes pasos:

- I. Se escoge un píxel del conjunto

- II. Si todavía no se ha probado dicho píxel, se prueba que el píxel cumpla:

$$R_{\max} - Rango < R < R_{\max} + Rango \quad (10)$$

$$G_{\max} - Rango < G < G_{\max} + Rango \quad (11)$$

$$B_{\max} - Rango < B < B_{\max} + Rango \quad (12)$$

$$H_{\max} - Rango < H < H_{\max} + Rango \quad (13)$$

Donde R, G, B y H son las componentes de color de rojo, verde, azul y matiz del píxel, respectivamente; R_{\max} , G_{\max} , B_{\max} y H_{\max} son los valores máximos de los cuatro histogramas de color, y Rango es un valor a optimizar, que se ha tomado como 10.

Si se cumplen las 4 condiciones, en la imagen resultante el píxel que corresponde a la posición que se ha probado se establece con color blanco. Además, se añaden los 8 píxeles de alrededor del píxel que se ha probado al conjunto de píxeles que se deben probar.

Si no se cumple alguna de las condiciones, en la imagen resultante el píxel que corresponde a la posición que se ha probado se establece con color negro.

- III. Se obtienen los contornos de la imagen resultante, y se obtiene el rectángulo de área mínima que los contiene. Si dicho rectángulo tiene las dimensiones adecuadas como para considerarse un jugador, se para el proceso de crecimiento.

El resultado del proceso de crecimiento en cada imagen es un rectángulo que contiene al jugador que se está siguiendo, y que se utilizará para comenzar el crecimiento en la imagen siguiente.



Figura 27- Se muestra la silueta de la camiseta y pantalón del jugador con equipación roja, resultado del algoritmo de crecimiento

Sin embargo, en la pista de juego hay áreas del fondo que no se han conseguido eliminar con un método de extracción de fondo, ya sea porque no tienen un color predominante o porque tienen un color similar al de la equipación de los jugadores. Estas zonas del fondo constituyen un problema a la hora de implementar este algoritmo de crecimiento, ya que el algoritmo podría realizar el crecimiento de forma descontrolada si el jugador se encuentra cerca de esas zonas.

Al no obtener unos resultados adecuados, se decidió utilizar el filtro de partículas propuesto en [22].

3.4.3 Seguimiento mediante filtro de partículas

Mediante el algoritmo del filtro de partículas se estima la posición del jugador en la imagen ponderando las posiciones de las partículas según la medida en que el color de éstas se ajuste al histograma de color del jugador que se está siguiendo.

Cada una de las partículas representa una posición en la imagen, escogida de forma aleatoria o determinista. Sirven para muestrear el color del píxel de la imagen que corresponde con esa posición. Además de la posición, cada partícula tiene una velocidad, que equivale a cuánto se ha desplazado el rectángulo del jugador en la imagen anterior.

I Modelo del jugador

Previamente al comienzo del seguimiento, se realiza, para los píxeles que contiene el rectángulo de cada jugador, un histograma tridimensional de las tres componentes de color en el espacio HSV, que sirva como modelo con el que comparar el color de cada partícula.

II Inicialización de las partículas

También se establecen las posiciones de las partículas, eligiendo píxeles interiores al rectángulo del jugador, siguiendo una distribución gaussiana centrada en el centro del rectángulo, tanto para la componente vertical como horizontal de la posición de la partícula.



Figura 28 – Imagen en que se observa que las partículas se distribuyen de manera aleatoria siguiendo una distribución gaussiana

Tras la realización de varias pruebas, se modificó la varianza de la distribución gaussiana en la componente vertical de la posición, ya que es más probable que el jugador realice un movimiento rápido en horizontal que en vertical.

III Algoritmo iterativo. Estimación de la nueva posición del jugador

Para calcular el centro de masas que representa la nueva posición del jugador, se deben seguir tres pasos.

- I. Se actualiza la posición y la velocidad de cada partícula siguiendo un modelo de movimiento lineal. Además, se suma un ruido aleatorio tanto a la posición como a la velocidad de las partículas para compensar el movimiento impredecible del jugador y el movimiento de la cámara.

$$x_{t+1} = x_t + u_t + w_x \quad (14)$$

$$y_{t+1} = y_t + v_t + w_y \quad (15)$$

$$u_{t+1} = u_t + w_u \quad (16)$$

$$v_{t+1} = v_t + w_v \quad (17)$$

Donde x e y corresponden a la posición de la partícula, u y v son la velocidad en la dirección horizontal y vertical, respectivamente, w_x y w_y son el ruido que se suma a la posición y w_u y w_v son el ruido que se suma a la velocidad de la partícula.

- II. Se obtiene, de la misma forma que se ha hecho con el rectángulo que contiene el jugador, un histograma de color tridimensional a partir de los 24 píxeles del entorno de la posición de la partícula. Se utiliza el mismo número de bins.

A continuación, se calcula la verosimilitud de cada partícula comparando dicho histograma con el histograma de color del rectángulo del jugador. La verosimilitud representa cuánto se ajusta el color de la partícula al modelo de color del jugador.

$$L = \sum_{u=1}^m \sqrt{p_u q_u} \quad (18)$$

Donde L es la verosimilitud de la partícula, m es el número total de bins del histograma de color y p y q son el valor normalizado de cada uno de los bins de los histogramas:

$$\sum_{u=1}^m p_u = \sum_{u=1}^m q_u = 1 \quad (19)$$

- III. Por último se calcula el nuevo centro de masas la posición de cada partículas con la verosimilitud de la partícula correspondiente:

$$(c_x, c_y) = \left(\sum_{i=1}^n L_i x_i, \sum_{i=1}^n L_i y_i \right) \quad (20)$$

Este centro de masas definirá la posición del centro del rectángulo del jugador en la siguiente imagen.

IV Reinicialización de las partículas en cada imagen

Representando la posición de cada una de las partículas en varias imágenes, se observó que una vez que se inicializaba el conjunto de partículas del filtro se dejaba de tener control de la posición de las partículas, dado que en cada frame se añade cierto ruido aleatorio a la posición y la velocidad. Como consecuencia a medida que progresaba el vídeo, la posición de las partículas no tiene por qué permanecer en el interior del rectángulo del jugador. Esto disminuía la eficacia del filtro de partículas, porque las que quedaban fuera del rectángulo del jugador ‘falseaban’ el cálculo de la posición del rectángulo al no tener un histograma de color similar al del rectángulo del jugador.

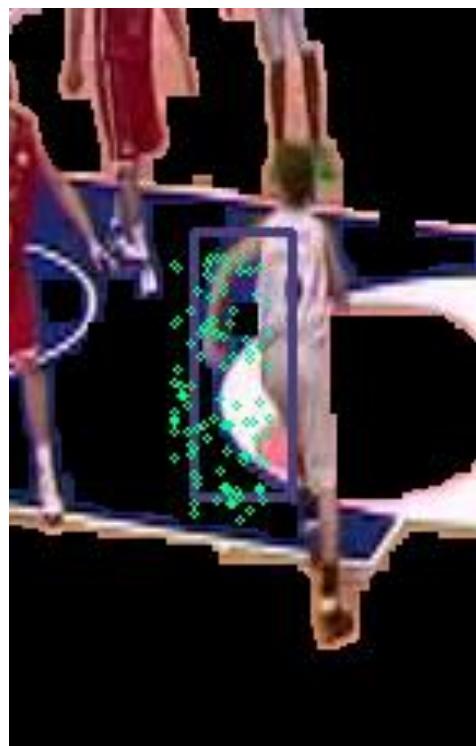


Figura 29 – Se puede apreciar que la posición de algunas partículas, representadas en verde, está fuera del rectángulo del jugador, representado en gris

Para solucionarlo se propuso reiniciar el conjunto de partículas dentro del rectángulo del jugador en cada uno de las imágenes, en vez de actualizar la posición y velocidad de las partículas. Así se conseguía mantener controlada la posición de las partículas en todas las imágenes.

Este algoritmo, de forma conjunta con las mejoras planteadas, funciona correctamente cuando el color del fondo es diferente del color del jugador. En caso contrario, como sucede en el vídeo de prueba, el algoritmo de seguimiento puede no estimar de forma adecuada la posición del jugador.

V Mejora del sistema utilizando la imagen de movimiento

En el fondo (la pista de juego) hay colores rojizos y blancos, que no pueden ser eliminados mediante los métodos de extracción de fondo expuestos. Así, el algoritmo deja de seguir al jugador que atravesese una zona que tenga un color similar al de su camiseta.



Figura 30 – Imagen en que se aprecia que el filtro de partículas deja de seguir al jugador con camiseta y pantalón blancos porque algunas de las partículas se han inicializado en esta imagen en la parte blanca de la pista

Debido a que no se puede eliminar completamente el fondo utilizando las técnicas de extracción de fondo, se decidió utilizar un mecanismo que ponderara más las partículas que están sobre la parte de la imagen que está en movimiento.

Para conseguir diferenciar las zonas de la imagen que tienen movimiento, se calcula la imagen de movimiento, tal como se ha explicado en la página 31.

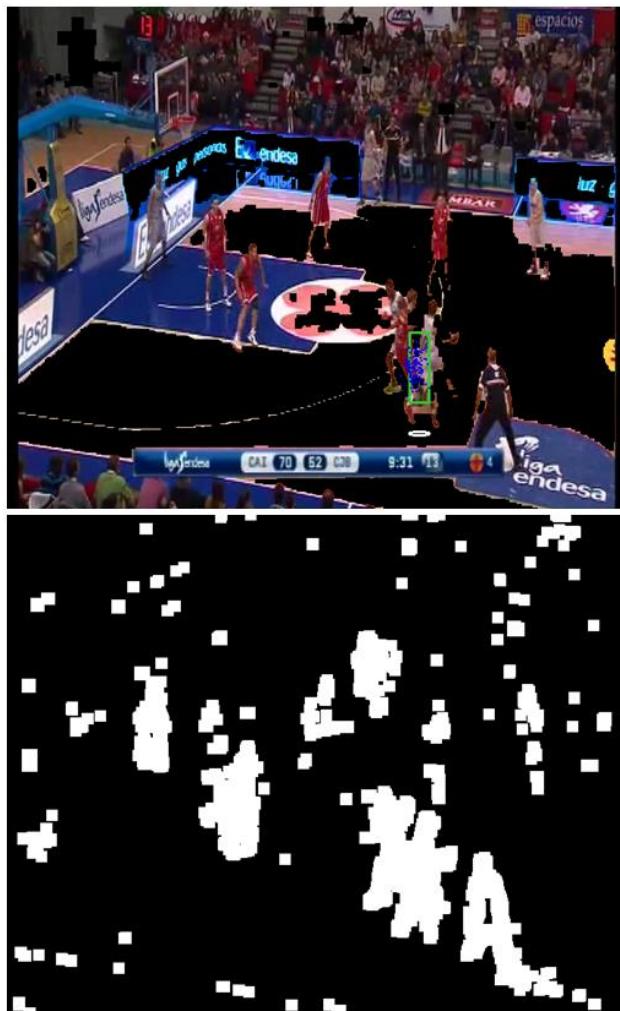


Figura 31 – En la imagen inferior se muestra la imagen de movimiento correspondiente a la imagen superior

Para que el filtro de partículas siga correctamente al jugador, las partículas cuyo píxel en la imagen de movimiento esté en blanco ponderarán más que las partículas cuyo píxel en la imagen de movimiento esté en negro cuando se calcule el nuevo centro de masas.

Se realizaron pruebas y se concluyó que si las partículas en movimiento ponderaban 100 veces más que las partículas que no estuvieran en movimiento se conseguía que el algoritmo de seguimiento funcionara correctamente.

3.4.4 Situar los pies de los jugadores

Como se ha mencionado en la introducción del apartado de desarrollo del trabajo, este trabajo final de grado se desarrolla de forma paralela con otro, que utiliza las posiciones de los jugadores sobre la pista en cada una de las imágenes de una secuencia para describir las jugadas que se van desarrollando. Por ello, es necesario representar las posiciones de los jugadores como puntos de las sucesivas imágenes.

Como el punto de contacto entre los jugadores y la pista son los pies de los jugadores, se decidió utilizar la posición de los pies de los jugadores para representar la posición de cada jugador sobre la pista.

Tras realizar algunas pruebas, se observó que la distancia entre la parte más baja del pantalón del jugador y los pies del jugador era similar para todos los jugadores, independientemente de su altura. Por ello, se decidió que el punto que representaría la posición de los jugadores en la pista sería un punto cuya componente horizontal coincide con la componente horizontal del centro del rectángulo, y cuya componente vertical está 35 píxeles por debajo de la posición inferior del rectángulo del jugador.

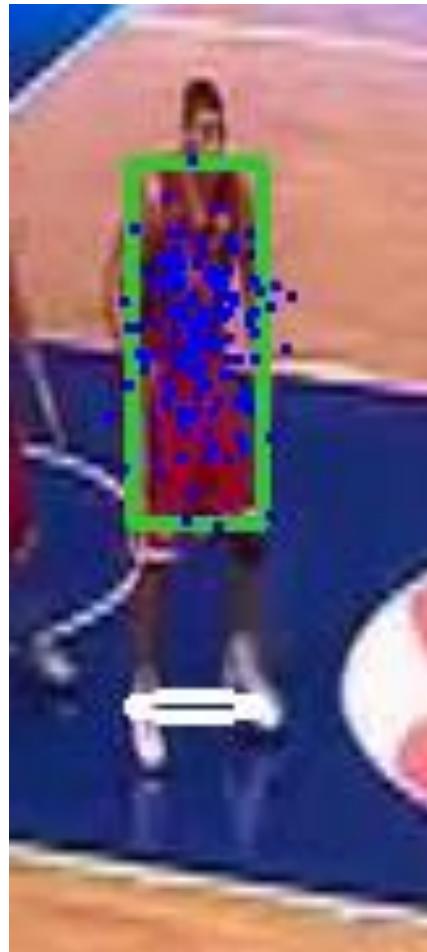


Figura 32 – Imagen en que se representa una elipse cuyo centro corresponde con la posición del jugador en la imagen

3.4.5 Detección de oclusiones

En muchas ocasiones durante el partido se da la circunstancia de que los jugadores realizan cruces y 'bloqueos', para ayudar a los compañeros de su mismo equipo. En esos momentos, dado que solo se dispone del vídeo proveniente de una cámara, se producen oclusiones entre jugadores.

Como consecuencia, el algoritmo de seguimiento puede dejar de seguir al jugador que es 'tapado', ya que durante varias imágenes, ese jugador no aparece en la escena.



Figura 33 – Secuencia de imágenes que muestran un instante anterior a la oclusión, un instante durante la oclusión y un instante posterior a la oclusión

I Detección del inicio de la oclusión

En primer lugar, se debe detectar en qué momento se produce una oclusión. Para ello, se comprueba en cada imagen, una vez calculada la nueva posición del rectángulo para cada jugador, si algunos de los rectángulos tienen área en común. Cuando se haya detectado la oclusión, se guardará la imagen en la que se ha detectado la oclusión.



Figura 34 – Imagen en que se puede ver que los rectángulos

II Resolución de la oclusión

Estimación de movimiento a largo plazo

En la mayoría de los bloqueos o cruces entre jugadores, la oclusión termina después de un segundo como máximo. Por tanto, se espera que transcurra un número suficiente de imágenes para que se haya terminado la oclusión, continuando el seguimiento mediante el procedimiento normal (filtro de partículas) en esas imágenes.

Transcurridas esas imágenes, se realiza la estimación de movimiento entre la imagen actual y la guardada previamente. Del mismo modo que se ha explicado en la página 29, para cada uno de los rectángulos de los jugadores que han provocado la oclusión se obtienen los vectores de entre estas dos imágenes, para conocer hacia dónde se han movido los jugadores.

A continuación, se desplaza el rectángulo de cada uno de los jugadores que han provocado la oclusión, según la dirección del vector de movimiento más repetido.

Este método no funciona correctamente, porque en la mayoría de las situaciones en que se producen oclusiones, como por ejemplo en las jugadas en que se realizan bloqueos, los jugadores giran sobre sí mismos. Este hecho implica que la estimación de movimiento realizada no representa la dirección en la que se ha movido el jugador a lo largo de esas 20 imágenes.



Figura 35 – Se puede apreciar que no hay una dirección predominante en los vectores de movimiento

Estimación de movimiento a corto plazo

Al no conseguir resolver la oclusión utilizando las imágenes anterior y posterior a la oclusión, se planteó realizar la estimación de movimiento cada tres imágenes, es decir, obtener los vectores de movimiento entre una imagen y la imagen correspondiente a tres instantes de tiempo antes. De esta forma, se debería resolver el problema ocasionado por el giro de los jugadores. Sin embargo, al realizar la estimación de movimiento, no se obtenía una dirección predominante en los vectores de movimiento calculados, sino que cada uno tenía una dirección distinta.

Asimismo, se llevaron a cabo varias pruebas realizando la estimación de movimiento entre imágenes consecutivas, sin obtener buenos resultados.

El hecho de no conseguir buenas prestaciones con este método quizá se deba a que existen en la imagen bloques del mismo color, como por ejemplo color rojo, consecutivos. Al realizar la estimación de movimiento, un vector de movimiento correspondiente a un bloque de color rojo de una imagen podría asociarse a cualquiera de los bloques consecutivos de color rojo de la imagen siguiente, falseando la estimación de movimiento e impidiendo que el seguimiento se realice correctamente.



Figura 36 – Se puede apreciar que no hay una dirección predominante en los vectores de movimiento

Filtrado basado en entrenamiento

Como no se obtenían los resultados deseados, se decidió probar el filtrado que se ha utilizado para determinar la posición de los jugadores antes de comenzar el seguimiento. Sin embargo, tras realizar el filtrado se deben aplicar varias operaciones de erosión para eliminar el ruido, lo que hace que se pierda el seguimiento del jugador que es ocluido. Por tanto, este método tampoco es útil para resolver las occlusiones.

III Corrección de la posición en las imágenes intermedias

En caso de que, después de haber desplazado los rectángulos, no exista la oclusión, es decir, los rectángulos de los jugadores que tenían área en común ya no la tengan, se realiza interpolación lineal en las coordenadas vertical y horizontal de la posición de los rectángulos de los jugadores, para estimar en qué posición estarían los jugadores que provocaron la oclusión mientras se estaba resolviendo la oclusión.

3.5 Solución que aporta mejores prestaciones

Después de estudiar e implementar varios sistemas de seguimiento de jugadores, se ha visto que de los sistemas de sustracción del fondo, el que utiliza la detección de color dominante aporta mejores resultados. Además, si este algoritmo de eliminación del fondo se utiliza en combinación con el filtrado basado en entrenamiento que se ha propuesto para detectar la posición de los jugadores en la escena, se eliminan multitud de falsas detecciones.

Por último, el algoritmo de seguimiento que mejor se comporta en cuanto a seguimiento de jugadores es el filtro de partículas expuesto, junto con las mejoras que se han propuesto posteriormente a su explicación y el algoritmo de resolución de occlusiones explicado.

Al combinar estos algoritmos se obtiene un sistema bastante robusto que realiza de forma adecuada la detección y el seguimiento de los jugadores.

4 Explicación del software desarrollado

Como se ha comentado en la introducción de esta memoria, para implementar y comparar los diferentes algoritmos expuestos se ha empleado el lenguaje de programación C# con el entorno de desarrollo Microsoft Visual Studio. Además se han utilizado las librerías de AForge y Emgu CV, específicas para el procesado de imágenes.

Para procesar las imágenes del vídeo que se esté tratando, se ha utilizado el tipo de dato Image de Emgu CV, el cual permite representar imágenes tanto en el espacio de color RGB como en el espacio HSV y en escala de grises.

Con el propósito de simplificar en la medida de lo posible el código programado y facilitar la reutilización del mismo, se han creado varias clases que sirven cada una para una función específica dentro del programa. A continuación, se va a explicar de forma breve en qué consiste la implementación de la solución con la que se han conseguido mejores resultados.

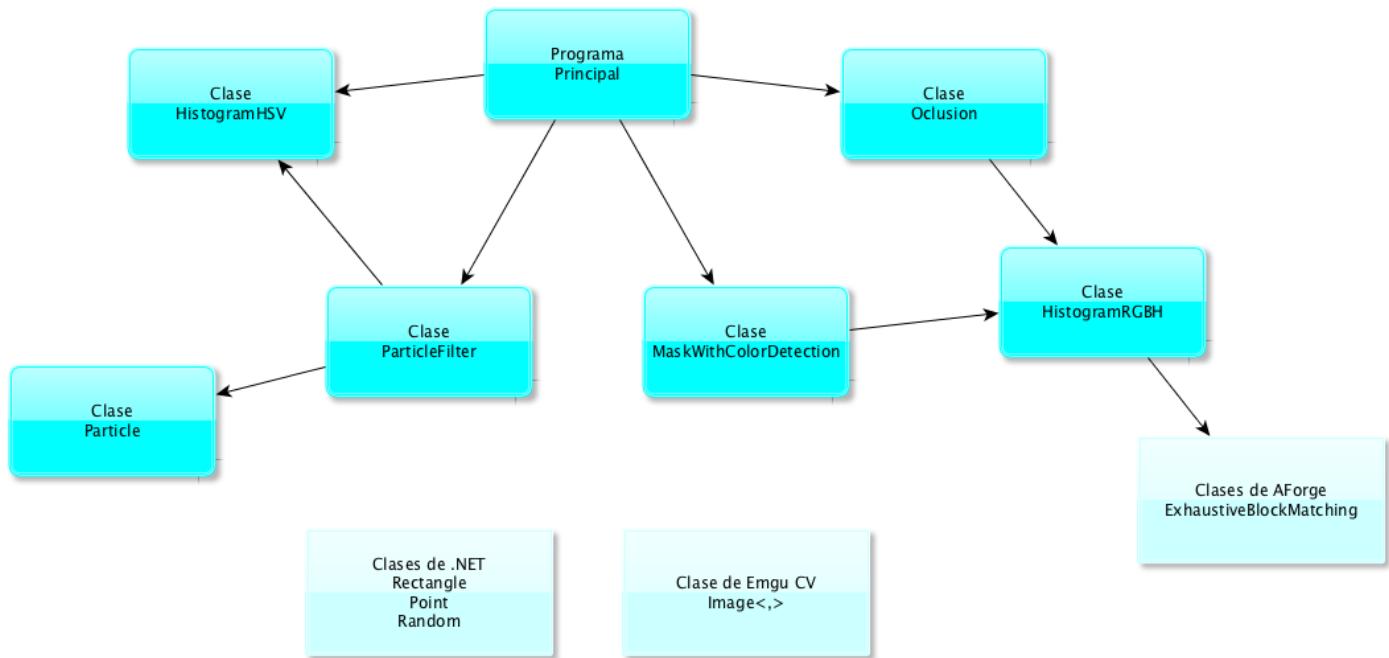


Figura 37 – Diagrama de clases de la solución que aporta mejores prestaciones

Como se puede ver en el diagrama de clases anterior, en la combinación de algoritmos con la que se consiguen mejores resultados se utilizan principalmente seis clases programadas por el autor de este trabajo final de grado. De la misma manera, cada una de estas clases utilizan tanto clases incluidas en las librerías por defecto del entorno .NET, en las librerías de Emgu CV y AForge.

Clase HistogramHSV

Se utiliza para construir un modelo de color de cada jugador con el que comparar cada partícula en el filtro de partículas.

Clase ParticleFilter

Se utiliza para estimar la posición del rectángulo mediante el algoritmo del filtro de partículas. Para ello, utiliza la clase HistogramHSV y la clase Particle.

Clase Particle

Se utiliza para representar la posición y la velocidad de una partícula del filtro de partículas.

Clase MaskWithColorDetection

Utiliza un algoritmo recursivo para detectar y eliminar el color dominante en cada imagen, mediante la clase HistogramRGBH.

Clase HistogramRGBH

Se utiliza para construir un modelo de color para los jugadores de cada equipo para realizar la localización de los jugadores previa al seguimiento. También se utiliza para construir el modelo de color de la imagen para extraer el color dominante.

Clase Occlusion

Se utiliza para detectar y resolver las occlusiones entre jugadores que se puedan producir.

5 Conclusiones y líneas futuras de trabajo

5.1 Conclusiones

En el transcurso de este Trabajo Final de Grado se han estudiado diferentes algoritmos con la finalidad de detectar y seguir a los jugadores en el transcurso de una jugada en un partido de baloncesto. Además, se ha analizado cómo se podría mejorar las prestaciones de dichos sistemas.

Se debe mencionar que el vídeo utilizado para realizar las pruebas complica en gran medida el seguimiento de los jugadores al ser el resultado de una transformación de barrido entrelazado a barrido progresivo. Este vídeo contiene imágenes de baja resolución que impiden determinar con claridad las siluetas de los jugadores. También el hecho de que la pista no esté uniformemente iluminada hace que los jugadores que se trasladan desde la parte de la pista más próxima a la cámara hasta la zona más alejada cambien de color, lo que impide un seguimiento correcto de los jugadores.

En primer lugar se ha realizado un análisis detallado de las diversas técnicas que se utilizan tanto para el seguimiento de objetos como para el seguimiento de jugadores en competiciones deportivas, prestando especial atención a aquellas técnicas que se utilizarán en el análisis de partidos de baloncesto. Este estudio ha servido para familiarizarse con los algoritmos que posteriormente se han probado.

A continuación, se han implementado algunos de los sistemas de seguimiento estudiados en la literatura con el objetivo de verificar sus prestaciones y encontrar el algoritmo que mejor se ajustara a las condiciones con las que se partía, es decir, una secuencia de vídeo para realizar las pruebas obtenida con una sola cámara, con una resolución baja.

Después de haber implementado diversos sistemas, se ha encontrado una combinación de procedimientos con los que se consiguen unas prestaciones aceptables. Entre los algoritmos que se utilizan para eliminar el fondo de la imagen con el objetivo de facilitar el seguimiento, se ha determinado que se consiguen mejores resultados mediante la detección y substracción del color dominante expuesta en [4]. Para determinar la posición inicial de los jugadores que se utilizará en el algoritmo de seguimiento, se ha implementado un filtrado de color basado en un entrenamiento previo para determinar el color de las equipaciones de juego de ambos equipos. Por último, el sistema de seguimiento de los estudiados que resulta más robusto se basa en el filtro de partículas, al que se han añadido algunas mejoras para compensar el hecho de que la pista de juego no fuera de un solo color.

De forma paralela a la implementación de los sistemas, se han realizado diversas pruebas para comprobar que los algoritmos estudiados se comportaban de forma prevista. Esto ha provocado que se probaran otros sistemas de seguimiento si los resultados no eran satisfactorios o que se intentara mejorar aquellos algoritmos con los que se conseguía resultados aceptables.

No obstante, el número de pruebas realizadas no es suficiente como para llevar a cabo un estudio de la precisión de los diferentes sistemas, con lo que no pueden presentarse unos resultados con los que comparar a nivel cuantitativo los diferentes sistemas implementados.

A nivel personal, se ha aprendido a tener iniciativa a la hora de probar diferentes soluciones o alternativas a las que se proponían en los artículos estudiados o a las propuestas del tutor del trabajo.

También se ha tenido la oportunidad de trabajar siguiendo una dinámica de investigación, que aporta perseverancia y paciencia al ver que no se obtienen los resultados que se desean al primer intento.

Asimismo, se ha trabajado teniendo en cuenta una fecha de entrega a largo plazo, lo que ha supuesto la necesidad de organizarse las tareas a realizar tanto a corto como a medio plazo.

Con todo esto, se debe mostrar la satisfacción por el trabajo realizado y por los resultados alcanzados pese a las condiciones adversas de las que se partía, mejorando en algunos casos las prestaciones de los algoritmos del estado del arte estudiados.

5.2 Líneas futuras de trabajo

Tras haber estudiado y probado algunos sistemas de seguimiento de jugadores en un partido de baloncesto, se plantean algunas ideas con las que se podría continuar este trabajo final de grado.

Como se ha mencionado en el apartado de conclusiones, los diferentes sistemas estudiados deberían ser testeados de manera más exhaustiva para obtener resultados cuantitativos que permitan reafirmar los resultados que se exponen de manera cualitativa en esta memoria.

Además, se debería estudiar con mayor detalle la detección y solución de las oclusiones, que se ha abordado en este trabajo final de grado con poca profundidad. Las oclusiones provocan en la mayoría de los casos que el seguimiento se interrumpa y no pueda retomarse.

También se deberían estudiar y probar algoritmos diferentes a los estudiados en este trabajo, con el objetivo de comprobar si podrían funcionar mejor que los que se han probado con las condiciones del vídeo de prueba que se ha utilizado.

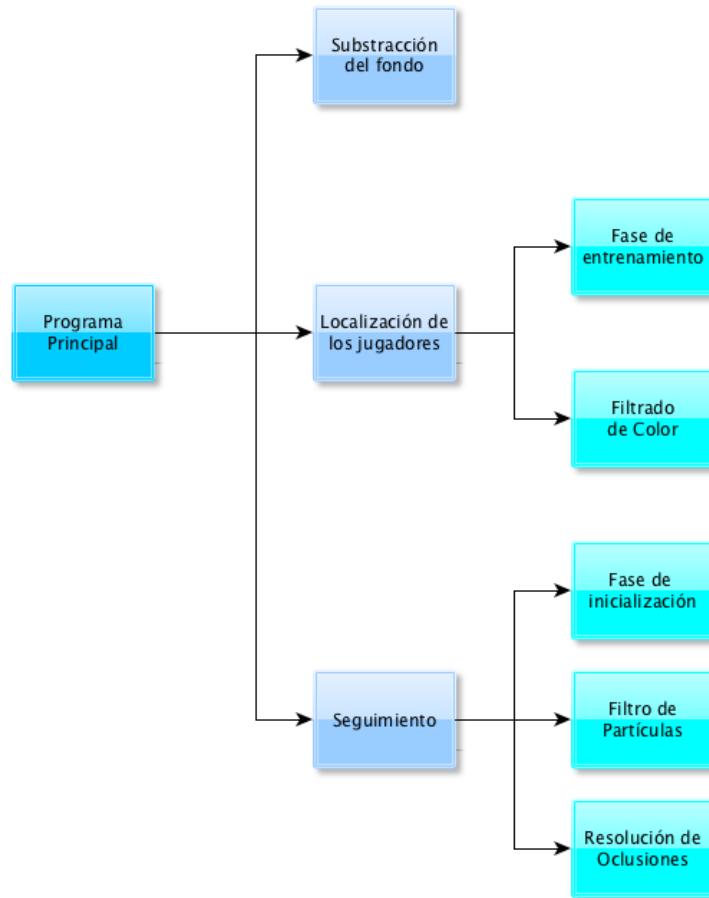
6 Referencias

- [1] Kalman, R.E. "A new aproach to linear filtering and prediction problems." *Transactions of the ASME-Journal of Basic Engineerin*, 82 (Series D):35-45 (1960)
- [2] Wang, Z., Liu, Z., Liu, W. & Kong, Y. "Particle filter algorithm based on adaptative resampling strategy." *International Conference on Electronic & Mechanical Engineering and Information Technology*, vol. 6, pp. 3138 - 3141 (2011)
- [3] Isard, M. & Blake, A. "Condensation-Conditional Density Propagation for visual tracking." *International Journal of Computer Vision* 29 (1), 5-28 (1998)
- [4] Ekin, A. & Tekalp, A. M. "Robust dominant color region detection and color-based applications for sports video." *International Conference on Image Processing (ICIP)*, vol.1, pp. 1 - 21 - 4 (2003)
- [5] Lu, W., Ting, J., Murphy, K. & Little, J. "Identifying players in broadcast sports videos using conditional random fields" *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3249 – 3256, IEEE (2011)
- [6] Felzenszwalb, P., McAllester, D. & Ramanan, D. "A discriminatively Trained, Multiscale, Deformable Part Model in CVPR" *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1 - 8, IEEE (2008)
- [7] Hu, M., Chang, M., Wu, J. & Chi, L. "Robust camera calibration and player tracking in broadcast basketball video" *Transactions on Multimedia*, vol. 13, issue 2, pp. 266-279 (2010)
- [8] Hanjalic, A. "Shot-boundary detection: Unraveled and resolved?" *Transactions on Circuits and systems for video technology*, vol. 12, issue 2, pp. 90 – 105 IEEE (2002)
- [9] Kayumbi, G. & Cavallaro, A. "Robust homography-based trajectory transformation for multi-camera scene analysis" *International Conference on Distributed Smart Cameras (ICDSC)*, pp. 59 – 66, IEEE (2007)
- [10] Bradski, G.R. "Open Source computer Vision Library Reference Manual", Intel (1999)
- [11] Allen, G.R., Xu, R.Y.D. & Jin, J.S. "Object tracking using CamShift algorithm and multiple quantized feature spaces", *School of Information Technologies, University of Sydney* (2006)
- [12] HSV (Hue, Saturation and value, <http://www.tech-faq.com/hsv.html>), fecha consulta 30/7/2014
- [13] HSL, HSB and HSV color: differences and conversion, <http://codeitdown.com/hsl-hsb-hsv-color/>, fecha consulta 30/7/2014
- [14] The HSV Colorspace, <http://ie.technion.ac.il/CC/Gimp/node51.html>, fecha consulta 31/7/2014

- [15] Emgu CV Library Documentation,
<http://www.emgu.com/wiki/files/2.0.0.0/html/748c1975-172a-9b73-a102-cdb695182d68.htm>, fecha consulta 23/8/2014
- [16] Morphological Image Processing,
<https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm>, fecha consulta 2/8/2014
- [17] OpenCV 2.4.9.0 documentation,
http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html,
fecha consulta 2/8/2014)
- [19] Erosion, <http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm>, fecha consulta 2/8/2014
- [20] Dilation, <http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>, fecha consulta 2/8/14
- [21] Verma, O.P., Hanmandlu, M., Susan, S., Kulkami, M. & Jain, P.K. "A Simple Single Seeded Region Growing Algorithm for Color Image Segmentation using Adaptive Thresholding"
International Conference on Communication Systems and Network technologies (CSNT), pp. 500 – 503 (2011)
- [22] Kataoka, H., Hashimoto, K. & Aoki, Y. "Player position estimation by monocular camera for soccer video analysis" *Proceedings of SICE Annual Conference, pp. 1985 – 1990 (2011)*

ANEXOS

Anexo 1 - Explicación del software desarrollado



1 Extracción del fondo de la imagen mediante la detección del color dominante

La clase que se ha desarrollado, llamada `MaskWithColorDetection`, obtiene una máscara que diferencia entre los jugadores y la pista de juego.

A su vez, esta clase utiliza un objeto que representa los histogramas de color del matiz, el rojo, el azul y el verde mediante arrays. La clase que define este objeto se denomina `HistogramRGBH`, y además de obtener los histogramas de las componentes de color de una imagen, permite obtener los valores máximos de dichos histogramas.



2 Localización de los jugadores en la imagen a través de un filtrado de color

Fase de entrenamiento

Previamente al filtrado de color, se realiza un entrenamiento en que el usuario debe seleccionar en diferentes imágenes los rectángulos de los jugadores, para obtener unos histogramas de color que sirvan como modelo para realizar el filtrado.

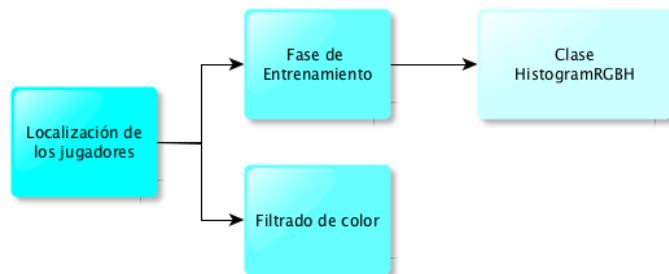
Por ello, en la interfaz gráfica desarrollada se han incluido unos botones que permiten comenzar la fase de entrenamiento y guardar los histogramas que constituyen el modelo de color de ambos equipos por separado. Para que el usuario seleccione los rectángulos de los jugadores, se han implementado unas funciones que se ejecutan cuando el usuario realiza un clic con el ratón sobre la imagen.

Para implementar estos histogramas modelo que se utilizan después en el filtrado de color, se ha utilizado la clase `HistogramRGBH`, mencionada en la página anterior.

Filtrado de color

Con el objetivo de obtener los contornos de los jugadores, se realiza un filtrado alrededor de los valores máximos de los histogramas mediante la función `InRange` de Emgu CV, que se puede aplicar a cualquier objeto `Image`. A continuación, se invoca a la función de OpenCV `FindContours` a través de Emgu CV para obtener los contornos de la máscara que ha resultado del filtrado de color. Por último, se utiliza la función de Emgu CV `MinAreaRect` que se aplica a cada uno de los contornos encontrados.

Del mismo modo que para la fase de entrenamiento, se han incluido algunos botones para corregir las posibles imperfecciones que puedan generarse durante el filtrado de color.



3 Seguimiento de los jugadores

Antes de comenzar el algoritmo del filtro de partículas, se deben obtener el histograma de color del rectángulo del jugador, para que sirva como modelo al realizar el seguimiento.

Inicialización

Para confeccionar un modelo de color con el que comparar las partículas, se ha implementado la clase `HistogramHSV`, que utiliza arrays tridimensionales para representar los histogramas de color tridimensionales en el espacio de color HSV.

Filtro de partículas

Para llevar a cabo el algoritmo del filtro de partículas, se ha desarrollado la clase `ParticleFilter`, que calcula la posición del rectángulo de cada jugador. Por tanto, es necesario crear tantos objetos `ParticleFilter` como jugadores se desee seguir.

En el filtro de partículas, cada partícula estará representada por un objeto `Particle`, que tiene como parámetros dos objetos `Point` que representan la posición y la velocidad de la partícula. Para obtener el histograma de color de cada partícula, que después se compara con el histograma modelo del jugador, se utiliza también la clase `HistogramHSV`.

Mediante esta clase `ParticleFilter` también se obtiene la imagen de movimiento. La diferencia entre dos imágenes se obtiene mediante la función de Emgu CV `AbsDiff` y las operaciones morfológicas de erosión y dilatación se realizan mediante las funciones de Emgu CV `Erode` y `Dilate`.

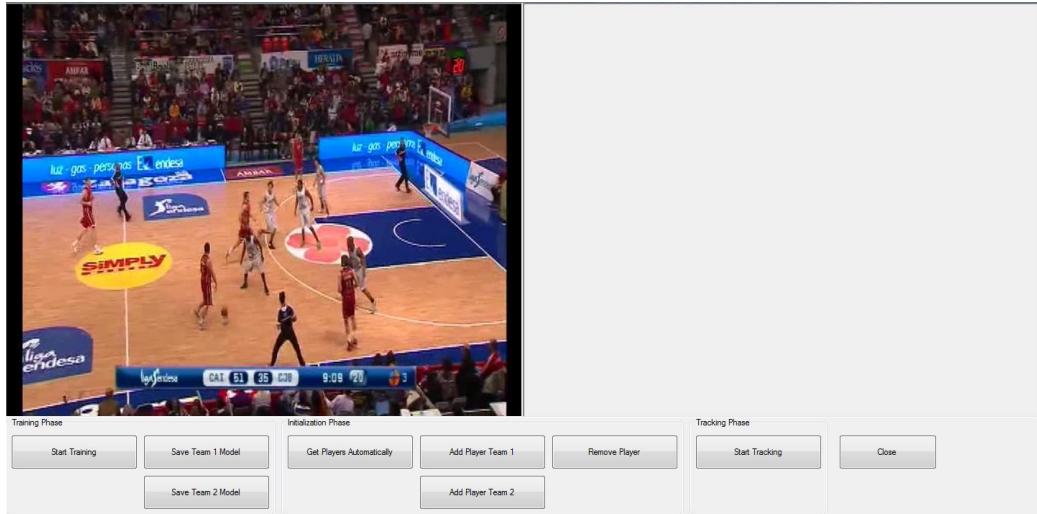
Oclusiones

Para detectar las imágenes en que comienza una oclusión, se verifica en cada una de las imágenes, una vez actualizado el rectángulo de cada jugador, si se produce un solapamiento entre alguno de los rectángulos, mediante la función `IntersectsWith`. En caso de que haya solapamiento, se crea un objeto `Oclusion`, que se ha implementado para resolver las oclusiones.

Dicho objeto `Oclusion` recibe la imagen en que comienza la oclusión y los rectángulos de los jugadores que provocan la oclusión en esa imagen. A continuación, espera a que transcurran 20 imágenes, que implica que ha transcurrido suficiente tiempo como para que se haya resuelto la oclusión. Seguidamente, calcula los vectores de movimiento de los rectángulos entre las imágenes mediante la función de AForge `ExhaustiveBlockMatching`.

Anexo 2 - Guía de usuario

A continuación se va a explicar de forma breve los pasos que se deben seguir para manejar la interfaz y conseguir llevar a cabo la localización y el seguimiento de los jugadores.



1 Entrenamiento

Para crear un modelo de color del jugador, se debe realizar un entrenamiento previo con el objetivo de obtener posteriormente la localización de los jugadores de forma automática.

Primero se debe presionar el botón “Start Training”. A continuación, se deben seleccionar los rectángulos de los jugadores de un equipo mediante clics con el ratón en la imagen. Finalmente, se debe presionar el botón “Save Team 1 Model” o “Save Team 2 Model”, según corresponda.

Si se desea repetir el proceso, se debe presionar el botón “Start Training” de nuevo.

2 Situar a los jugadores en la escena

Este paso puede realizarse de dos formas:

I Determinar la posición de manera automática

Para situar a los jugadores automáticamente en la escena, se debe presionar el botón “Get Players Automatically”. A continuación, se mostrarán en la imagen de la izquierda de la interfaz los rectángulos de los jugadores.

Si la localización de los jugadores no ha sido precisa o existen falsos positivos, se pueden añadir nuevos rectángulos de los jugadores de forma manual, primero presionando el botón “Add Player Team 1” o “Add Player Team 2”, según el jugador sea de un equipo u otro. A continuación, se pueden seleccionar tantos rectángulos como se desee definiendo el rectángulo mediante dos clics con el ratón en la imagen. Si se desea eliminar un rectángulo que constituye un falso positivo, se debe primero presionar el botón “Remove Player”, y seguidamente hacer clic con el ratón en el interior del rectángulo que se desea eliminar.

II Determinar la posición de manera manual

Para situar a los jugadores manualmente en la escena, se debe primero presionar el botón “Add Player Team 1” o “Add Player Team 2”, según el jugador sea de un equipo u otro. A continuación, se pueden seleccionar tantos rectángulos como se desee definiendo el rectángulo mediante dos clics con el ratón en la imagen.

3 Seguimiento

Para comenzar el seguimiento, se debe presionar el botón “Start Tracking”.