



**Universidad**  
Zaragoza

# Trabajo Fin de Máster

Seguimiento de Múltiples Objetos Basado en el  
Algoritmo de Viterbi

Autor/es

Ramón A. Suárez Fernández

Director/es

Carlos Orrite Uruñuela

Máster Oficial en Ingeniería Electrónica  
Escuela de Ingeniería y Arquitectura  
2011-2012



# Seguimiento de Múltiples Objetos Basado en el Algoritmo de Viterbi

---

## *Resumen*

---

El seguimiento de objetos en secuencias de imágenes es actualmente un tema de investigación importante debido a que tiene un amplio rango de aplicaciones tales como video vigilancia, análisis deportivo, etc. Un ejemplo común es el análisis de jugadores en un partido de fútbol. Mediante el procesamiento de las imágenes se puede establecer la trayectoria de cada jugador durante el partido y así proveer información importante sobre su actividad. El problema del seguimiento de objetos tiene dos grandes pasos principales, el primero es detectar y localizar los objetos dentro de los fotogramas del video y el segundo es la parte de seguimiento, esto implica implementar un método que obtenga las trayectorias de los objetos detectados resolviendo las oclusiones que pueden establecer entre ellos. En este trabajo se propone un método para el seguimiento de múltiples objetos. Se parte de un trabajo previo donde se detectó a los jugadores en la imagen y se estableció la localización de todos ellos en el terreno de juego, afrontando el segundo problema explicado, es decir, la asignación de una etiqueta inequívoca para cada jugador a lo largo de todo el partido. Para llevar a cabo esta tarea previamente se ha procedido a un etiquetado manual de todos los jugadores para posteriormente verificar la fiabilidad del método propuesto. El método planteado sigue un análisis de probabilidades de presencia de cada jugador en una posición determinada del campo y un método robusto de asignación temporal de todas las posiciones de los jugadores mediante el algoritmo de Viterbi.

## *Palabras Clave*

---

Seguimiento de Objetos, Algoritmo de Viterbi, Análisis Deportivo, Análisis de Probabilidades.



*Quiero dar las gracias en primer lugar a mi tutor Carlos Orrite Uruñuela por haberme ofrecido la oportunidad de realizar este Trabajo Fin de Master y agradecerle toda la ayuda y el apoyo que he recibido durante todos estos meses. Finalmente, me gustaría dedicarle este trabajo a mi familia y amistades, los cuales siempre han estado conmigo brindándome apoyo y consejos en los momentos buenos y malos.*

*A vosotros, Gracias.*



# Índice

---

Introducción .....	1
1.1 Motivación .....	1
1.2 Objetivos y Enfoque.....	2
1.3 Organización de la Memoria .....	3
Estado del Arte .....	4
Seguimiento de Múltiples Objetos.....	10
3.1 Problemática .....	10
3.2 Modelo de Red.....	11
3.3 Metodología.....	12
3.3.1 Procesos Discretos de Markov.....	12
3.3.2 Elementos de un HMM .....	12
3.3.3 Problemas Básicos de los HMM .....	14
3.3.4 Solución al Problema 2 .....	14
3.3.5 Algoritmo de Viterbi.....	16
Aplicación al Análisis Deportivo.....	18
4.1 Secuencias (Datos) de Vídeo .....	18
4.2 Etiquetado Manual de los Jugadores .....	20
4.3 Transformación con la Matriz de Homografía.....	22
4.4 Elaboración de Mapas Probabilísticos.....	23
4.5 Detalles de Implementación.....	24
4.5.1 Número de Estados del Modelo N.....	25
4.5.2 Probabilidad de Transición $a_{i,j}$ .....	25
4.5.3 Probabilidad de Observación $b_j(k)$ .....	26
4.5.4 Etiquetado de la Posición Mediante Viterbi .....	27
Análisis de Resultados .....	28

5.1 Cálculo del Umbral $\lambda$ .....	28
5.2 Simulación del Sistema Inicial.....	31
5.3 Método de Corrección de Etiquetas.....	33
5.4 Simulación del Sistema Final .....	36
Conclusiones .....	40
6.1 Propuestas de Desarrollo Futuro .....	42
Anexo I: Cálculo de la matriz de Homografía.....	43
Bibliografía .....	46



## Tabla de Figuras

---

Figura 1 - Histograma de la fecha de publicación de los trabajos citados a lo largo del trabajo. ....	4
Figura 2. Modelo de la red para seguimiento de un jugador. ....	11
Figura 3. Esquema del campo de visión del terreno de juego. ....	19
Figura 4. Vistas de las cámaras 1-3 del terreno de juego. ....	19
Figura 5. Asignación de etiquetas a los jugadores en el terreno. Disposición 4-4-220	
Figura 6. Ejemplo del programa de etiquetado manual de jugadores. ....	21
Figura 7 - Ejemplo gráfico del cambio de coordenadas del plano de la imagen al plano del campo .....	22
Figura 8 - Histogramas de las posiciones de los jugadores 1 (arriba) y 9 (abajo)....	23
Figura 9 - Ejemplo de las trayectorias obtenidas por el algoritmo de Viterbi .....	24
Figura 10 - Demostración Grafica del Cálculo de Distancia Euclidea .....	26
Figura 11 - Curva de Error al cambiar el Umbral ( $\lambda$ ) .....	30
Figura 12 - Resultado de Etiquetado Inicial para el fotograma 64 .....	32
Figura 13 - Resultado de Etiquetado Inicial para el fotograma 383 .....	32
Figura 14 - Resultado del Etiquetado Inicial para el fotograma 477 .....	33
Figura 15 - Resultado de Etiquetado Final para el fotograma 64.....	38
Figura 16 - Resultado de Etiquetado Final para el fotograma 383.....	38
Figura 17 - Resultado de Etiquetado Final para el fotograma 477.....	39
Figura 18 - Ejemplo gráfico de la transformación de coordenadas de la imagen al plano .....	43

## Listado de Tablas

---

Tabla 1 - Matriz de Confusión para $\lambda = 0,1$ .....	28
Tabla 2 - Matriz de Confusión para $\lambda=0,3$ .....	29
Tabla 3 - Matriz de Confusión para $\lambda=0,5$ .....	29
Tabla 4 - Porcentaje de Error Dependiendo del Umbral $\lambda$ .....	30
Tabla 5 - Resultados detallados del Etiquetado inicial .....	31
Tabla 6 - Tabla de Valores de $\delta_t(j)$ para el fotograma 383 .....	34
Tabla 7 - $\delta_t(j)$ del Jugador A en los fotogramas 382 - 384 .....	35
Tabla 8 - $\delta_t(j)$ del Jugador B en los fotogramas 382 - 384 .....	35
Tabla 9 - Matriz de confusión de la simulación del sistema final.....	36
Tabla 10 - Resultados detallados del Etiquetado Final .....	37

# Introducción

---

## 1.1 Motivación

---

Este Trabajo Fin de Máster se concentra en problemas de visión por computador, un área del tema de Ambientes Inteligentes que pertenece al Máster Oficial en Ingeniería Electrónica de la Universidad de Zaragoza. El área de la visión por computador es un sub-campo de la inteligencia artificial. El propósito de esta es programar una máquina que sea capaz de “entender” acciones, escenas o características dentro de una imagen. Uno de los objetivos más comunes de la visión por computador es el seguimiento de objetos en secuencias de imágenes.

Siendo uno de los pasos más complicados en el análisis de las imágenes, últimamente ha habido considerable actividad de investigación en el seguimiento de objetos a partir de secuencias de imágenes en vídeo. Este interés está motivado por sus numerosas aplicaciones, tales como, video-vigilancia, video conferencia, interfaces hombre-máquina, y análisis deportivo. En este trabajo se enfrenta el problema de seguimiento de múltiples objetos en secuencias de imágenes en vídeo, donde, en algunos casos, los objetos se ocluyen ya sea parcial o totalmente. El método utilizado se aplicará a un entorno deportivo, en concreto se trabajará con las trayectorias de los jugadores en un partido de fútbol. La motivación de este trabajo nace del conocimiento del Grupo de Visión por Computador y Redes Neuronales de la Universidad de Zaragoza en análisis de imágenes y sistemas de video-vigilancia y parte de un trabajo previo donde se detectó a los jugadores en la imagen y se estableció la posición de todos ellos en el terreno de juego.

## 1.2 Objetivos y Enfoque

---

El principal objetivo de este trabajo es implementar un método que a través de algoritmos de análisis de imagen y probabilidades, sea capaz de seguir personas en una secuencia de video, asignándole una etiqueta inequívoca a cada persona durante toda la secuencia. Concretamente, se aplicará este método al seguimiento de jugadores de fútbol en el terreno de juego para así obtener sus trayectorias en los distintos fotogramas y resolver las oclusiones que pueden establecerse. No obstante, se puede dividir esta tarea global en varios objetivos intermedios:

- Encontrar o desarrollar un algoritmo de seguimiento de múltiples objetos.
- Minimizar la tasa de error en el seguimiento, teniendo en cuenta situaciones como cruces y oclusiones.

El seguimiento es un campo en el que muchos autores han trabajado desde hace muchos años, y existen multitud de métodos desarrollados, muchos de ellos aplicados a problemas como seguimiento de vehículos, personas, aviones, misiles u otros objetos de trayectoria desconocida. Sin embargo, muchos de los problemas que plantean están todavía sin resolver completamente. Para alcanzar los objetivos que se plantean en este proyecto, las tareas que se llevarán a cabo son las siguientes:

**Estudio del estado del arte:** En primer lugar se hará un resumen del estado del arte en los métodos de seguimiento de objetos utilizados por diferentes autores para comentar las ventajas e inconvenientes de cada estrategia.

**Implementación de algoritmos:** Una vez conocido el estado del arte se implementará el algoritmo seleccionado para resolver la problemática del trabajo.

**Análisis de resultados y conclusiones:** Por último, se analizarán los resultados obtenidos los cual nos permitirá extraer las conclusiones finales de este trabajo.

### ***1.3 Organización de la Memoria***

---

La memoria está compuesta por los siguientes capítulos:

**Capítulo 1:** Introducción, motivación y objetivos del proyecto.

**Capítulo 2:** Estudio del estado del arte en los métodos de seguimiento de objetos.

**Capítulo 3:** Seguimiento de múltiples objetos y metodología utilizada en el trabajo.

**Capítulo 4:** Explicación de la aplicación del algoritmo al entorno deportivo.

**Capítulo 5:** Análisis de los resultados obtenidos de la simulación del algoritmo.

**Capítulo 6:** Conclusiones y propuestas de desarrollo futuro.

**Anexo I**

En este capítulo se presenta un estudio de los métodos novedosos publicados por diferentes autores para resolver la problemática del seguimiento de objetos. Este estudio nos permite conocer las distintas maneras posibles de resolver el problema estipulado anteriormente. La Figura 1 demuestra un histograma con las fechas de las publicaciones utilizadas como referencias en este trabajo. Como se puede ver, la mayoría de las publicaciones utilizadas son de trabajos de investigación recientes, sin embargo, hay algunos trabajos de hace muchos años que han sido relevantes en la elaboración de este Trabajo Fin de Master.

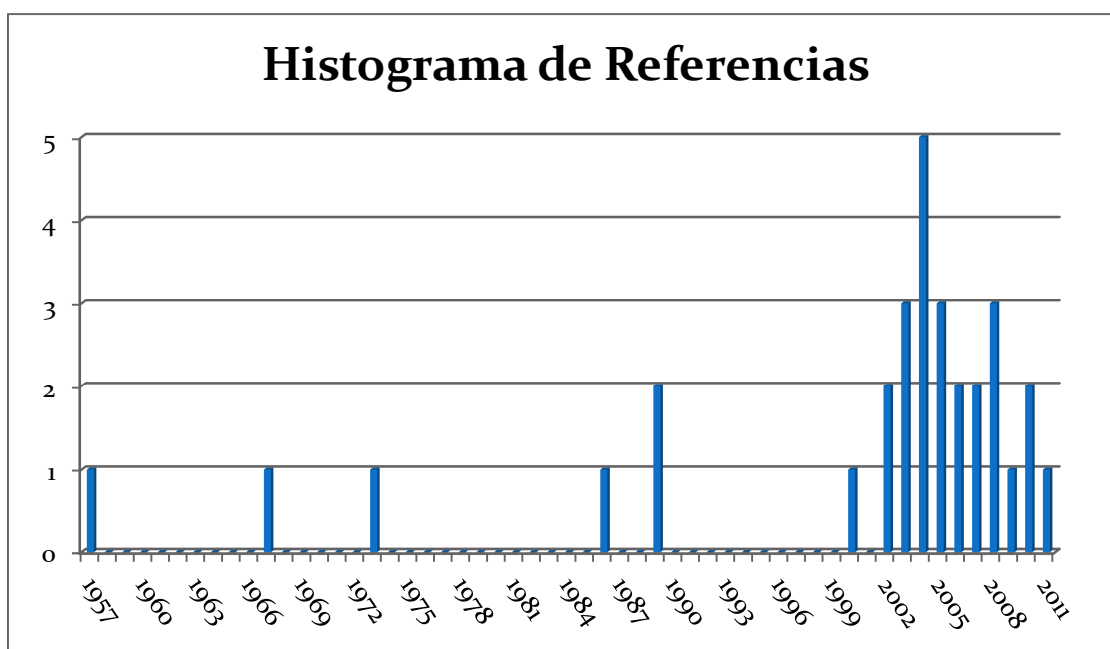


Figura 1 - Histograma de la fecha de publicación de los trabajos citados a lo largo del trabajo.

Una gran variedad de los métodos se basan en la actualización recursiva de trayectorias con los objetos detectados más recientes, por ejemplo, El Filtro de Kalman [1], [2], [3], [4], [5]. Este algoritmo se divide en dos pasos:

1. Predicción de la posición de los objetos que están en movimiento.
2. Comparación de la predicción con las medidas obtenidas en la detección (si las hay) y obtención de la estimación de estado para cada posición del objeto. En caso de no existir medidas que puedan provenir de un objeto se tomará como estimación a la posición de su predicción.

El Filtro de Kalman original presenta el inconveniente de que solo es capaz de estimar de forma correcta las posiciones en sistemas lineales y gaussianos. Debido a esto, se han desarrollado modificaciones de dicho filtro. Las modificaciones más conocidas son:

- “*Extended Kalman Filter*” (EKF):

En dicho algoritmo se aproxima la distribución de estado por una variable aleatoria gaussiana, la cual se propaga a través un sistema no lineal de primer orden. Los errores que introduce pueden hacer que el filtro diverja en caso de que el sistema no sea lo suficientemente lineal [6].

- “*Unscented Kalman Filter*” (UKF):

En dicho algoritmo también se aproxima la distribución de estado por una variable aleatoria gaussiana, solo que en este caso se representa por un conjunto de puntos determinados (los cuáles serán calculados previamente), llamados puntos sigma. Estos puntos disponen de la media y la covarianza de la distribución gaussiana; y cuando se propagan a través de un sistema no lineal, capturan la media y la covarianza hasta el tercer orden [7].

Si se desea realizar el seguimiento de varios objetos a la vez, la complejidad de estos algoritmos residirá en la asociación que se debe de realizar entre las medidas obtenidas para dichos objetos y su posición estimada.

El algoritmo de seguimiento tipo Montecarlo, también conocido como Filtro de Partículas [8], [9], [10], [11], [12], se basa en un muestreo de las distribuciones. Tiene la ventaja de poder realizar el seguimiento con modelos no lineales y/o no gaussianos, representando la densidad de probabilidad mediante un conjunto de puntos, el cuál evoluciona según lo hace la imagen de probabilidad. En lugar de usar medidas discretas como datos de entrada, utiliza imágenes de probabilidad.

La complejidad de estos algoritmos reside en cómo obtener el peso de cada una de las partículas y en cómo realizar la evolución de las muestras usadas en un instante de tiempo para dar lugar a las muestras que van a ser usadas en el instante posterior. Uno de los métodos más utilizados es el “*Sample Importance Resampling*” (SIR); el cual da lugar al algoritmo de Condensación y consiste en tres pasos:

1. Generación de partículas a partir de una función de importancia.
2. Cálculo de pesos.
3. Remuestreo eliminando las partículas con pesos bajos.

Este algoritmo dispone de un inconveniente como es el tiempo de cómputo, ya que es mayor que con otros algoritmos de seguimiento. Por lo tanto será conveniente trabajar con arquitecturas en paralelo para poder trabajar de la forma más cercana al tiempo real.

Por otra parte, si se pretende realizar un seguimiento de múltiples objetos mediante este tipo de algoritmos, existirán bastantes complejidades, sobre todo si las imágenes de probabilidad son ruidosas, de manera que puedan hacer que la posición de los objetos abandone su trayectoria correcta. No obstante, han sido creados diversos sistemas basados en el Filtro de Partículas, los cuáles son capaces de realizar esta tarea de una manera lo más eficiente posible. Esta técnica se ha utilizado para seguir a varios jugadores de hockey en [13].

Además de los dos grupos de algoritmos mencionados anteriormente, los cuáles pueden ser usados para realizar un seguimiento de múltiples objetos, caben destacar los algoritmos de seguimiento "*Multiple Hypothesis Tracking*". En este tipo de algoritmos, en caso de duda a la hora de asociar las medidas a los objetos a seguir, se pospone la decisión, creando de tal manera un árbol de hipótesis en el cuál se van calculando las probabilidades a posteriori. Cuando dichas probabilidades son lo suficientemente altas se toma la decisión. Cada rama del árbol se trata como si fuera una posición distinta. De tal manera se deberían poder alcanzar buenos resultados en un entorno con medidas falsas.

El mayor inconveniente de este algoritmo es que el número de hipótesis con los que se trabaja suele crecer rápidamente, de manera que es necesario usar técnicas de podado de hipótesis ("*prunning*") que vayan eliminando las menos probables. Aun así, es bastante complejo realizar un buen podado de hipótesis, de manera que el problema pueda ser tratable computacionalmente y que a la vez no se descarten hipótesis válidas. A su vez, resulta compleja la asociación entre medidas e hipótesis cuando el número de hipótesis es elevado. Algunos algoritmos multi-hipótesis utilizan el Algoritmo de Viterbi para obtener las asociaciones de medidas. En dicho algoritmo, a partir de los diagramas de "trellis", se puede calcular el camino de mayor probabilidad. Otros algoritmos multi-hipótesis se basan en un modelo probabilístico, como el algoritmo PMHT [14].



Existen diversos algoritmos de seguimiento los cuáles no están basados en una detección previa, sino que realizan el seguimiento de objetos a partir de estadísticas obtenidas directamente de la imagen o a partir de los valores de los píxeles de la misma. El algoritmo “*mean-shift*” se basa en el reconocimiento de patrones en la imagen. En dicho algoritmo se pretende localizar los mínimos de diferencia de histograma entre un patrón de entrada y la imagen. Las funciones de similitud de histograma más usadas son el coeficiente de Bhattacharya o la divergencia de Kullback-Leibler. También existen otros algoritmos de seguimiento los cuáles tienen como entrada la segmentación de la imagen, ya que la segmentación no llega a ser considerada como detección en sí. En ocasiones se diseña a la vez la segmentación y el seguimiento, bien sea la segmentación basada en color y en textura o solo en color.

En un intento de aumentar su fiabilidad, algunos métodos tienen enfoques híbridos. Las detecciones en las imágenes se conectan en trayectorias cortas, que luego son unidas entre sí. Por ejemplo, [15] se basa en el filtro de Kalman para obtener trayectorias básicas, y luego trata de unir o separarlas usando el “Hungarian Algorithm”. Wu y Nevatia en [16] define una medida de la afinidad basada en la posición, tamaño y color luego utiliza el “Hungarian Algorithm” para asociar las hipótesis y las respuestas de detección de objetos en fotogramas vecinos. [17] explora la versión jerárquica del mismo concepto. Por el contrario en [18], Los autores asumen que una gráfica de trayectorias ya se ha producido y se centran en la vinculación de las etiquetas en el gráfico. Ellos formulan el seguimiento de varios objetos como un problema de Inferencia de Red Bayesiana y aplican este método para el seguimiento de varios jugadores de fútbol. Esta clase de métodos es un buen compromiso: La arquitectura de dos etapas les permite escalar de manera eficiente, mientras que al mismo tiempo, tienen en cuenta una ventana de observación más amplia. Sin embargo, mientras que exhiben buenos resultados en algunas situaciones, los métodos no garantizan la convergencia a un óptimo global. Por consiguiente, son propensas a errores tales como cambios de etiquetas.

Para mejorar la robustez, la investigación se ha centrado recientemente en la vinculación de las detecciones en una ventana de tiempo más grande utilizando varios esquemas de optimización. Programación Dinámica [19] se puede utilizar para unir las múltiples detecciones a través del tiempo, y por lo tanto, resolver el problema de seguimiento múltiple. Además, se puede ampliar para permitir la optimización de varias trayectorias simultáneamente [20].

La Programación Lineal es un método de optimización que se ha aplicado para encontrar óptimos globales y resolver el problema del seguimiento de varias personas [21]. Empezando desde la salida de detectores de objetos, este último enfoque construye un gráfico de la red en la que cada nodo es una observación y está completamente conectado a observaciones futuras y pasadas. Oclusiones entre los objetos se modelan mediante la especificación de los conflictos espaciales entre los nodos. Nodos adicionales se crean específicamente para manejar los objetos ocluidos. Finalmente, los pesos de los arcos se eligen de acuerdo a las apariencias de objetos y un modelo de este. Un modelo gráfico similar, con nodos que representan las detecciones, se construye por [22] para seguimiento de múltiples personas. El óptimo global se busca utilizando un algoritmo de flujo de costo mínimo, que se aprovecha de la estructura específica de la gráfica para alcanzar el óptimo más rápido.

# Seguimiento de Múltiples Objetos

---

# 3

En este capítulo se describe el método utilizado para resolver el problema de seguimiento de múltiples objetos en fotogramas de video. Para esto, se comienza por representar todas las posibles posiciones de un objeto dentro de la imagen como nodos. Luego de varios fotogramas, estos nodos interconectados forman un gráfico o “*trellis*”, donde un camino de nodos conectados representa una posible trayectoria de un objeto a través del tiempo en un video.

## 3.1 Problemática

---

Para poder hacer el seguimiento de los objetos, es necesario localizar sus posiciones a través de una secuencia de fotogramas de vídeo. Como se mencionó anteriormente, este trabajo parte de un trabajo previo donde se detectó a los jugadores en las imágenes y se estableció la localización de todos ellos en el terreno de juego. No obstante, ya teniendo esto, se ha procedido a hacer un etiquetado manual de todos los jugadores en el terreno de juego utilizando el programa explicado en la sección 4.2. El etiquetado manual se utilizará como las observaciones para poder verificar el desempeño del método implementado posteriormente. Partiendo de la detección de los objetos en las imágenes nos centraremos en un método que nos permita asignar una etiqueta inequívoca a cada jugador a lo largo de todo el partido.

### 3.2 Modelo de Red

A continuación se estudia el seguimiento de múltiples objetos basado en un modelo de la red en el cual todos los sub-modelos interactúan entre ellos.

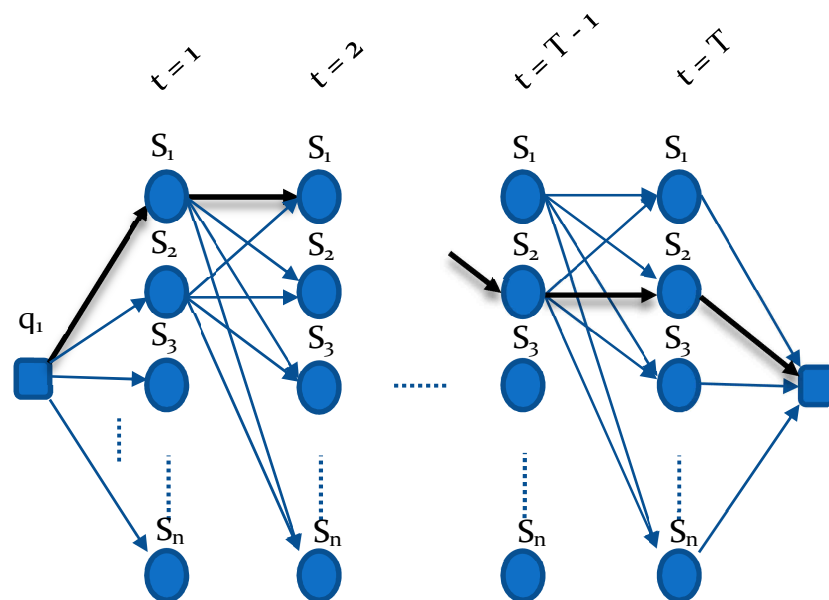


Figura 2. Modelo de la red para seguimiento de un jugador.

La Figura 2 representa la red en la cual se basa el algoritmo. Las posibles posiciones de un jugador son representados como nodos circulares. En cualquier fotograma, las posibles posiciones (es decir, las observaciones) para cada objeto pueden ser diferentes. Los nodos de inicio y final, mostrados como cuadrados en la Figura 2, también son incluidos en cada sub-modelo para representar el inicio y el final de la trayectoria del jugador, sin embargo, los nodos del final no corresponden a ningún estado de los jugadores. Las flechas en negro entre los nodos representan las posibles transiciones de estado. Un conjunto de nodos conectados entre los nodos de inicio y final representan la trayectoria espacial de un objeto. Una vez un nodo es seleccionado como la posición de un objeto, todos los otros objetos deben estar en una posición diferente dentro del fotograma.

### 3.3 Metodología

---

La teoría del Modelo Oculto de Markov (HMM) esta descrita en detalle en los artículos [23 y 24].

#### 3.3.1 Procesos Discretos de Markov

---

Se considera un sistema que puede ser descrito en cualquier instante de tiempo, como que está en uno de  $N$  estados distintos  $S_1, S_2, \dots, S_n$ . En instantes discretos de tiempo, el sistema sufre un cambio de estado conforme con un conjunto de probabilidades asociadas a este estado. Se conocen los instantes de tiempo asociados con los cambios de estado como  $t = 1, 2, \dots$ , y llamamos al estado en el tiempo  $t$  como  $q_t$ . Una descripción probabilística completa del sistema mencionado, generalmente, requiere la especificación tanto del estado actual (en el tiempo  $t$ ) como de los estados anteriores.

#### 3.3.2 Elementos de un HMM

---

Un HMM se caracteriza por lo siguiente:

- 1)  $N$ , el número de estados en el modelo. A pesar de que los estados están ocultos, para muchas aplicaciones prácticas hay algún significado físico adherido a los estados o a conjuntos de estados del modelo. Generalmente los estados están interconectados entre ellos de tal manera que se puede alcanzar un estado desde cualquier otro. Se denominan los estados individuales como  $S = \{S_1, S_2, \dots, S_N\}$ .
- 2)  $M$ , el número de símbolos de observaciones distintas por estado, i.e., el tamaño discreto del alfabeto. Los símbolos de las

observaciones corresponden a la salida física del sistema siendo modelado. Se denominan los símbolos individuales como  $V = \{V_1, V_2, \dots, V_M\}$ .

- 3) La distribución de las probabilidades de transición de estado  $A = \{a_{i,j}\}$  donde,

$$a_{i,j} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N. \quad (1)$$

Para el caso especial donde cualquier estado puede llegar a cualquier otro en un paso, tenemos que  $a_{i,j} > 0$  para todo  $i, j$ . Para otros tipos de HMM, podría ser que  $a_{i,j} = 0$  para uno o más de un par  $i, j$ .

- 4) La distribución de las probabilidades de los símbolos de observaciones en el estado  $j$ ,  $B = \{b_j(k)\}$ , donde

$$b_j(k) = P[V_k \text{ at } t | q_t = S_j], \quad \begin{array}{l} 1 \leq j \leq N \\ 1 \leq k \leq M. \end{array} \quad (2)$$

- 5) La distribución del estado inicial  $\pi = \{\pi_i\}$  donde

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N. \quad (3)$$

Dados los apropiados valores de  $N, M, A, B$  y  $\pi$ , el HMM puede ser utilizado para proveer una secuencia de observaciones,

$$O = O_1, O_2, \dots, O_t. \quad (4)$$

Donde cada observación  $O_t$  es uno de los símbolos de  $V$ , y  $T$  es el número de observaciones en la secuencia.

### 3.3.3 Problemas Básicos de los HMM

---

Dado el modelo del HMM anterior, hay tres problemas básicos de interés que deben ser resueltos para que el modelo pueda ser útil en aplicaciones reales. Los problemas son los siguientes:

Problema 1: Dada la secuencia de observaciones  $O = O_1, O_2, \dots, O_t$ , y un modelo  $\lambda = \{A, B, \pi\}$ , ¿Cómo calculamos la probabilidad de la secuencia de las observaciones  $P(O|\lambda)$ , dado el modelo?

Problema 2: Dada la secuencia de observaciones  $O = O_1, O_2, \dots, O_t$ , y el modelo  $\lambda = \{A, B, \pi\}$ , ¿Cómo escogemos la correspondiente secuencia de estados  $Q = q_1, q_2, \dots, q_t$  óptima?

Problema 3: ¿Cómo ajustamos los parámetros del modelo  $\lambda = \{A, B, \pi\}$  para maximizar  $P(O|\lambda)$ ?

Tal como se explica, el Problema 2 es el caso donde intentamos descubrir la parte oculta del modelo, mejor dicho, encontrar la secuencia de estados correcta. Debido a que la problemática de este trabajo se asimila más al Problema 2 de los HMM, solo se presentara a continuación la solución matemática a este.

### 3.3.4 Solución al Problema 2

---

A diferencia del Problema 1 para el cual una solución exacta puede ser encontrada, hay varias maneras de resolver el Problema 2, es decir, encontrar la secuencia de estados óptima asociada con la secuencia de observaciones dada. Por ejemplo, una posible solución es escoger como criterio de optimalidad los estados  $q_t$  que son individualmente más probables. Este criterio de optimización maximiza el número esperado de estados correctos. Para implementar esta solución al Problema 2, se define la variable,

$$\gamma_t(i) = P(q_t = S_i | O, \lambda), \quad (5)$$

i.e., la probabilidad de estar en el estado  $S_i$  en el tiempo  $t$  dada la secuencia de observaciones  $O$ , y el modelo  $\lambda$ . La ecuación (5) se puede expresar solo en términos de las variables  $\alpha_t(i)$  y  $\beta_t(i)$ ,

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)}. \quad (6)$$

Donde,

$$\alpha_t(i) = P(O_1 O_2 \cdots O_t, q_t = S_i | \lambda), \quad (7)$$

es la probabilidad de la secuencia de observaciones parcial,  $O_1 O_2 \cdots O_t$  (hasta el tiempo  $t$ ) y el estado  $S_i$  en el tiempo  $t$  dado el modelo  $\lambda$ . Y

$$\beta_t(i) = P(O_{t+1} O_{t+2} \cdots O_T | q_t = S_i, \lambda), \quad (8)$$

es la probabilidad de la secuencia de observaciones parcial desde  $t + 1$  hasta el final dado el estado  $S_i$  en el tiempo  $t$  y el modelo  $\lambda$ . El factor de normalización  $P(O|\lambda) = \sum_{i=1}^N \alpha_t(i) \beta_t(i)$  hace a  $\gamma_t(i)$  una medida de probabilidad para que

$$\sum_{i=1}^N \gamma_t(i) = 1. \quad (9)$$

Utilizando  $\gamma_t(i)$  se puede resolver para el estado individual más probable  $q_t$  en el tiempo  $t$  de la siguiente manera

$$q_t = \operatorname{argmax}_{1 \leq i \leq N} [\gamma_t(i)], \quad 1 \leq t \leq T. \quad (10)$$



Aunque (10) maximiza el número de estados correctos, podría haber problemas con la secuencia de estados resultante. Un ejemplo es cuando el HMM tiene valores de transición de estado con probabilidad de cero ( $a_{i,j} = 0$  para algún  $i, j$ ), la secuencia de estado optima puede ser que no sea una secuencia de estado valida. Esto es debido a que la solución de (10) simplemente resuelve por el estado más probable en cada instante sin tomar en cuenta la probabilidad de la secuencias de estados.

Una posible solución al problema anterior es modificar el criterio de optimización. Por ejemplo, se podría resolver por la secuencia de estados que maximizara el número esperado de pares correctos de estados ( $q_t, q_{t+1}$ ) o tríos de estados ( $q_t, q_{t+1}, q_{t+2}$ ), etc. Aunque este criterio puede ser razonable para algunas aplicaciones, el criterio más utilizado es encontrar la mejor secuencia de estados (trayectoria), o sea, maximizar  $P(Q|O, \lambda)$  lo cual es equivalente a maximizar  $P(Q, O|\lambda)$ . Una estrategia formal para encontrar la mejor secuencia de estados existe basada en Programación Dinámica y se llama el Algoritmo de Viterbi [25], [26].

### 3.3.5 Algoritmo de Viterbi

---

Para encontrar la mejor y única secuencia de estados  $Q = \{q_1, q_2, \dots, q_T\}$ , para la secuencia de observaciones dada  $O = O_1, O_2, \dots, O_t$ , necesitamos definir

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \dots q_t = i, O_1, O_2, \dots, O_t | \lambda] \quad (11)$$

i.e.,  $\delta_t(i)$  es la mayor probabilidad a través de una trayectoria, en el tiempo  $t$ , que toma en cuenta las primeras  $t$  observaciones y acaba en el estado  $S_i$ . Por lo que tenemos,

$$\delta_t(i) = [\max_i \delta_t(i) a_{i,j}] \cdot b_j(O_{t+1}). \quad (12)$$

Para realmente obtener la secuencia de estado hay que conocer el índice que maximizo a (12) para cada  $t$  y  $j$ . Esto se hace con la variable  $\psi_t(j)$ . El procedimiento completo para encontrar la mejor secuencia de estados se plantea a continuación:

#### 1) Inicialización

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (13)$$

$$\psi_i(i) = 0. \quad (14)$$

#### 2) Recursión

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{i,j}] b_j(O_t), \quad 2 \leq t \leq T \quad (15)$$

$$1 \leq j \leq N$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{i,j}], \quad 2 \leq t \leq T \quad (16)$$

$$1 \leq j \leq N$$

#### 3) Terminación

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (17)$$

$$q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)] \quad (18)$$

#### 4) "Backtracking"

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T - 1, T - 2, \dots, 1. \quad (19)$$

Los sistemas de seguimiento mediante visión por computador envuelven un rango muy amplio de situaciones para las cuales podrían ser utilizadas. Una de las posibles aplicaciones de estos sistemas es en el análisis deportivo. Los deportes profesionales forman una parte íntegra de nuestra sociedad debido a la fama y al dinero que generan. En estos entornos, donde cualquier información adicional que tengas es aprovechada, las herramientas de análisis deportivo son una parte muy importante del desempeño de los equipos. Debido a que este es un método no intrusivo, tiene ventaja frente a otros métodos en los cuales es necesario el uso de dispositivos electrónicos en los propios jugadores. En este trabajo el método propuesto se utilizará para el seguimiento de jugadores en un partido de fútbol, sin embargo, este puede ser extendido a otros escenarios y utilizado en otros entornos.

## ***4.1 Secuencias (Datos) de Vídeo***

---

Puesto que un campo de fútbol es un escenario bastante grande, Se han utilizado las grabaciones de 3 cámaras para obtener la información de un partido de una manera aceptable. En este caso están colocadas de forma que cada una enfocaba a una zona distinta del campo: izquierda, centro y derecha, a una altura de 25 m del suelo. En la Figura 3 se pueden ver los campos de visión de las cámaras.

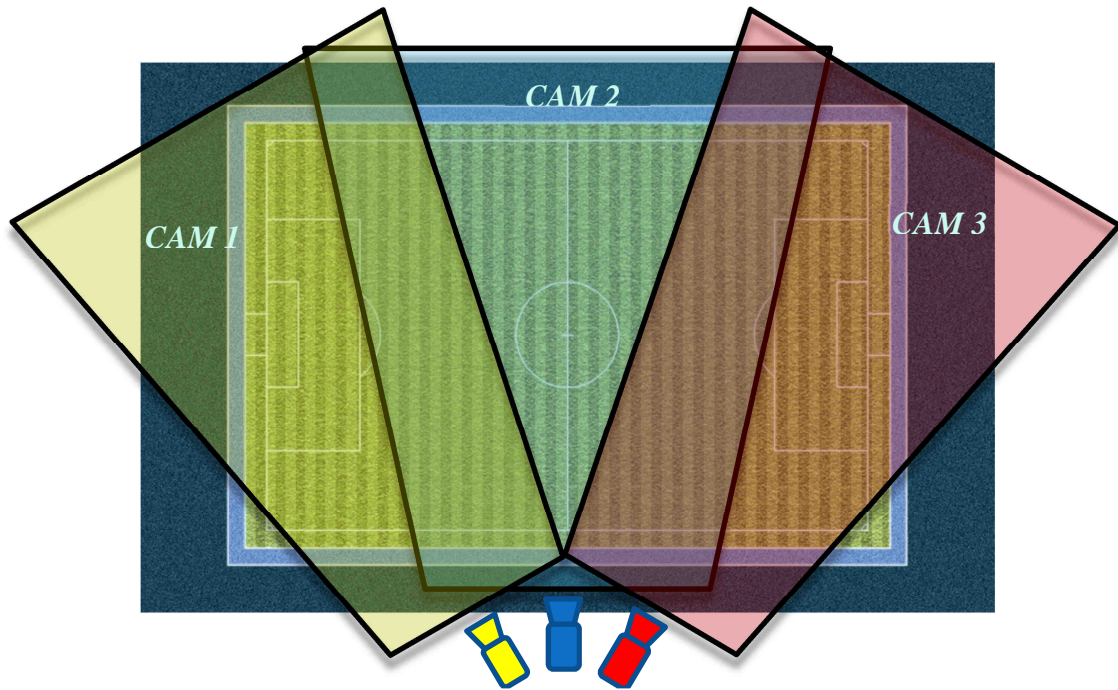


Figura 3. Esquema del campo de visión del terreno de juego.

No obstante, el tener más puntos de vista siempre ayudará a resolver los problemas que puedan aparecer de una manera más eficaz ya que se tendrá más información con la cual resolver el problema. Independientemente del número de cámaras usado, una vez obtengamos las medidas en cada cámara, dichas medidas serán pasadas a un plano común como veremos posteriormente. El método propuesto se aplicará a una secuencia de 1200 fotogramas, que equivalen a unos 15 minutos de partido y tienen una dimensión de 696 x 520 píxeles. La Figura 4 muestra las vistas del campo de fútbol de las cámaras instaladas.



Figura 4. Vistas de las cámaras 1-3 del terreno de juego.

## 4.2 Etiquetado Manual de los Jugadores

---

El primer paso necesario para realizar el seguimiento de los jugadores es detectar y localizar los objetos a seguir dentro de los fotogramas del video. Para llevar a cabo este trabajo, se ha procedido a un etiquetado manual de todos los jugadores. Para esto se les asigna a los jugadores un número (etiqueta) que los identifica dependiendo de su posición inicial en el campo. La asignación de la etiqueta de identificación se presenta en la Figura 5. Luego de que cada jugador tenga su etiqueta, se procede a manualmente asignarle una posición en la imagen para cada fotograma del video.

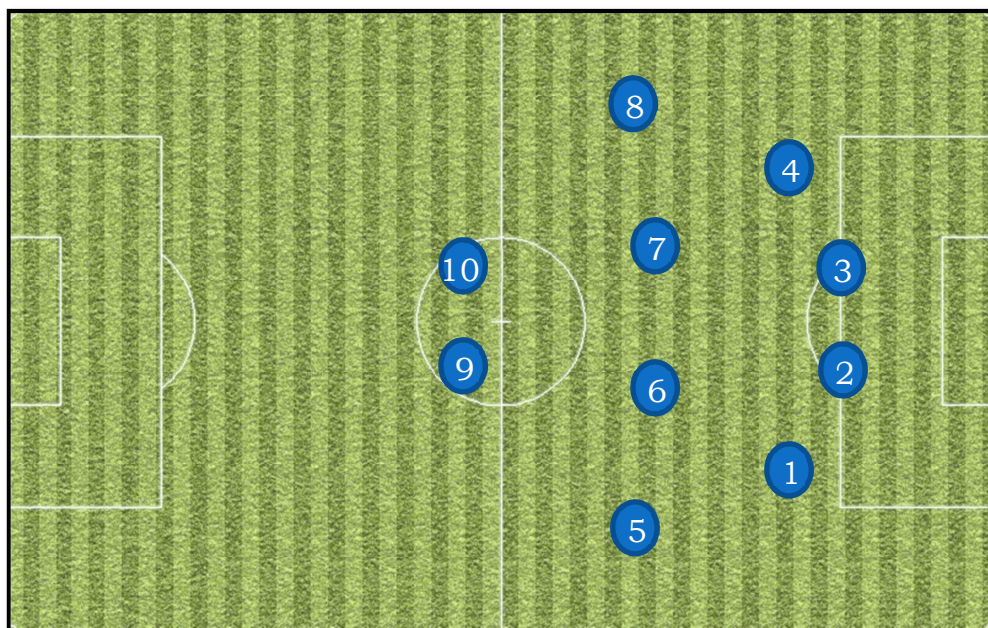


Figura 5. Asignación de etiquetas a los jugadores en el terreno. Disposición 4-4-2

El programa creado permite obtener la posición de los jugadores de cada equipo en la imagen. Se comienza por especificar la etiqueta del jugador que se pretende localizar (1 - 10), luego se procede a pulsar en la posición del jugador seleccionado con el cursor en cada fotograma. El etiquetado de los jugadores te permite almacenar información importante

que se utilizará más adelante en el trabajo. Entre los datos que almacena el programa se encuentra el número de la cámara en la que se localiza el jugador, siendo las opciones la cámara 1, 2 ó 3. Además, almacena el fotograma en el cual se está etiquetando (0 - 2000), el número del jugador en el campo (1 - 10), el equipo al que pertenece (1 ó 2) y las coordenadas en píxeles del jugador en la imagen. Un ejemplo del programa ejecutado se puede ver en la Figura 6 donde se está etiquetando a uno de los jugadores en el campo. El resultado de este ejemplo de etiquetado sería que al jugador número 5 del equipo número 1 se le asigna una coordenada  $(x,y)$ , en píxeles, dentro del fotograma  $n$  de la cámara 2.

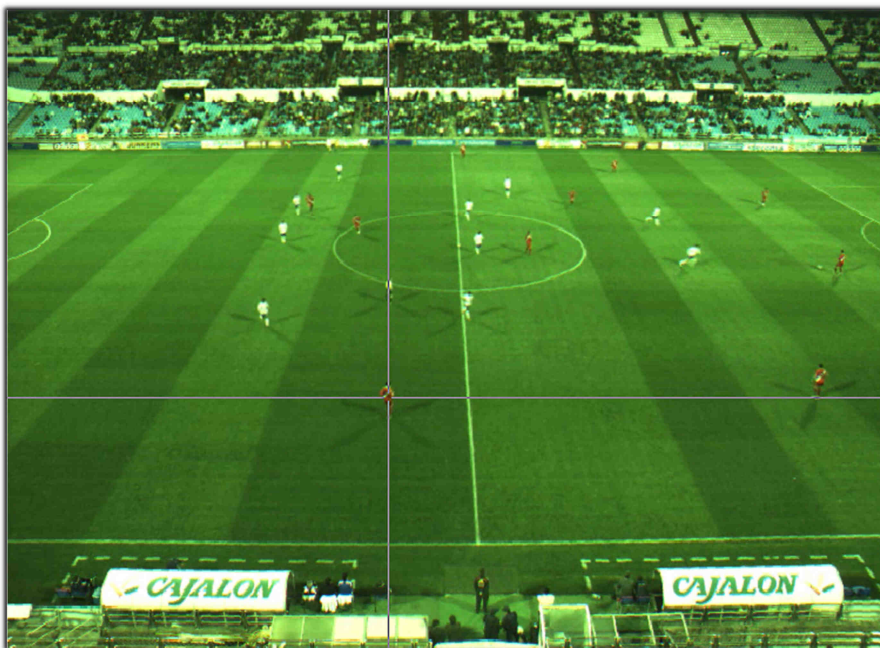


Figura 6. Ejemplo del programa de etiquetado manual de jugadores

### 4.3 Transformación con la Matriz de Homografía

Debido a que la localización de los jugadores es hecha en el plano de la imagen, se necesita una transformación de estas coordenadas a valores reales en el campo para tener una misma referencia para todas las posiciones. Esta transformación se hace utilizando la matriz de homografía.

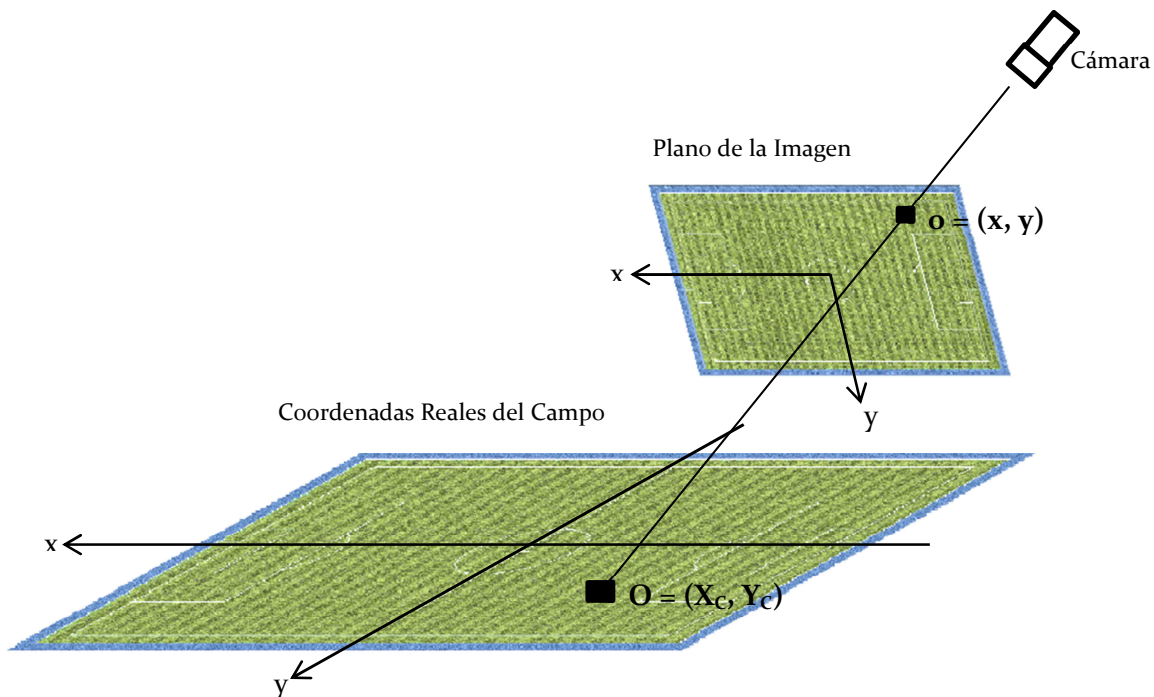


Figura 7 - Ejemplo gráfico del cambio de coordenadas del plano de la imagen al plano del campo

Se puede establecer una aproximación de las posiciones en el plano de la imagen y las posiciones en el campo, tomando un mínimo de cuatro puntos. Con esta transformación, se puede localizar la posición de los jugadores en el terreno de juego, asumiendo que están en contacto con el suelo [27]. Para poder llevar a cabo la transformación se tuvieron que utilizar las dimensiones del campo real, en este caso las dimensiones utilizadas del campo real fue de 100m x 60m. No obstante, dado que son tres cámaras diferentes se debe calcular una matriz de homografía para cada cámara. El método de encontrar la matriz de homografía se presenta en el Anexo I.

## 4.4 Elaboración de Mapas Probabilísticos

Luego de haber transformado las posiciones de los jugadores en el campo de la imagen a sus respectivas posiciones en el terreno de juego, se procedió crear un mapa probabilístico de los jugadores en el terreno de juego. Este mapa probabilístico se creó contando el número de ocasiones en que un jugador  $q = \{1, 2, \dots, 10\}$  se encuentra en la posición  $(x, y)$ , donde  $x \in [0, 100], y \in [0, 60]$  en cada uno de los fotogramas. Con estos valores se elabora un histograma de las posiciones que ocupa el jugador  $q$ . En la Figura 8 se demuestran dos ejemplos de los histogramas creados para los jugadores 1 y 9.

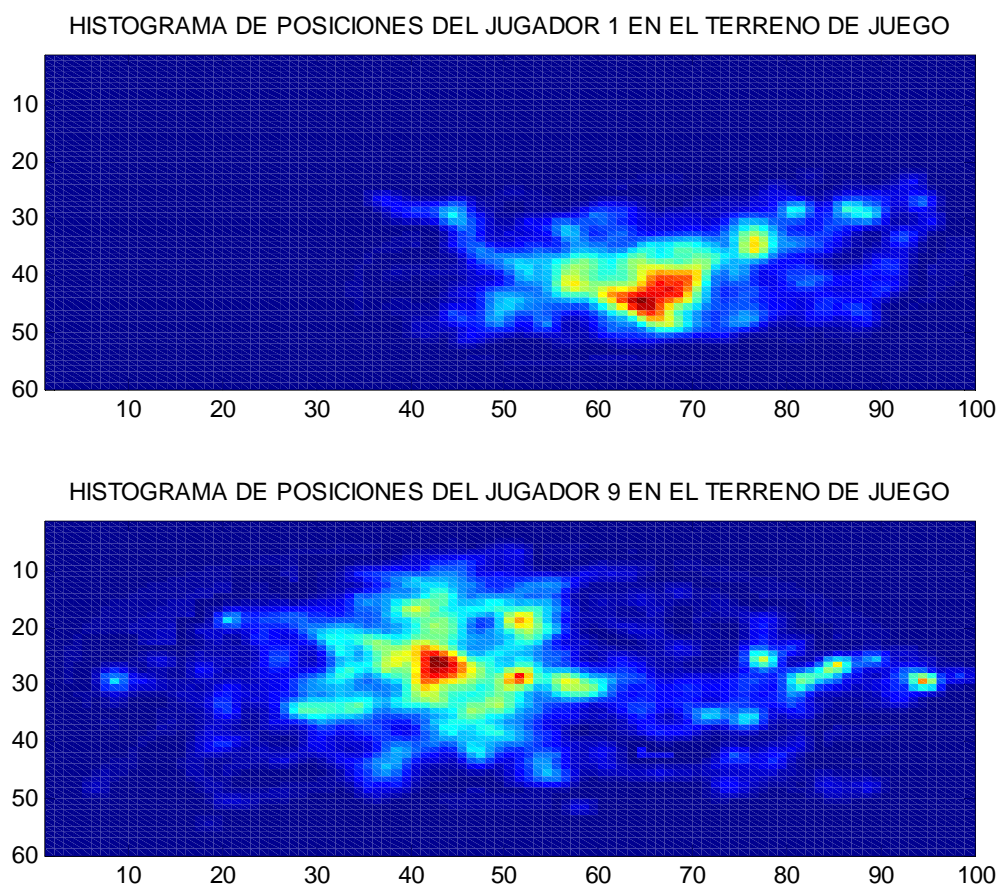


Figura 8 - Histogramas de las posiciones de los jugadores 1 (arriba) y 9 (abajo)

La obtención de un mapa probabilístico es un paso útil ya que esto nos proporciona las probabilidades de que un jugador esté en un punto del campo en cualquier instante de tiempo y se utilizara en la implementación del método planteado.



## 4.5 Detalles de Implementación

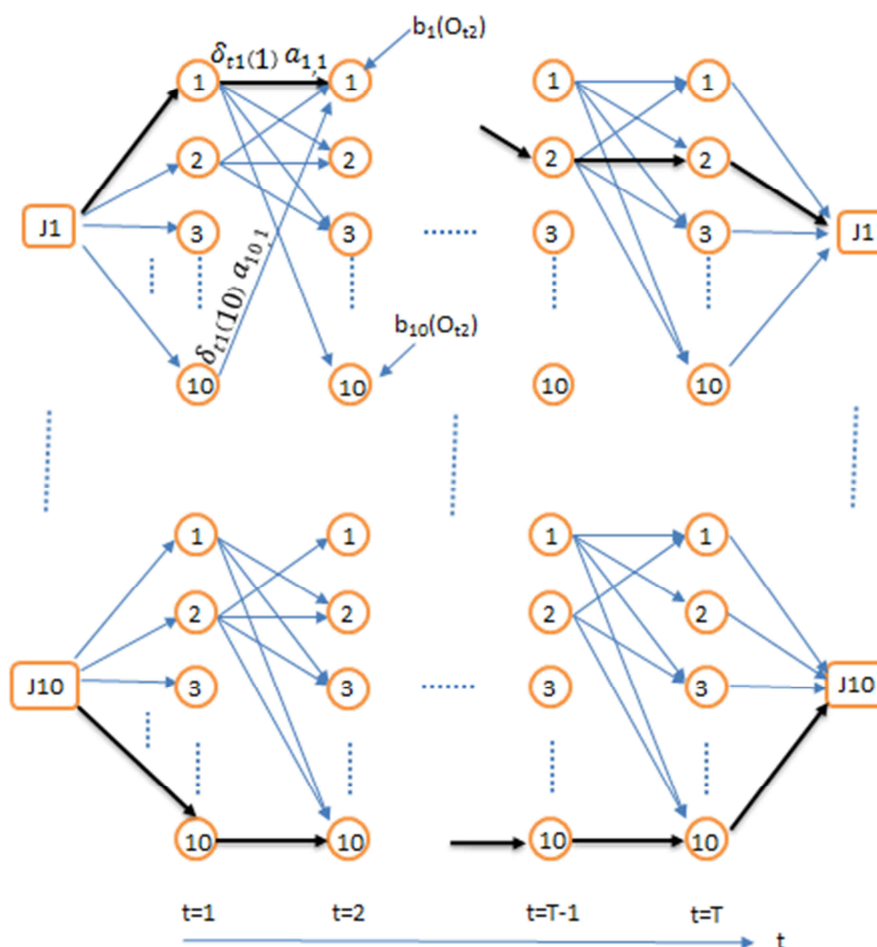


Figura 9 - Ejemplo de las trayectorias obtenidas por el algoritmo de Viterbi

La Figura 9 muestra un ejemplo gráfico del resultado que se espera del algoritmo implementado. El algoritmo de Viterbi obtiene la mejor trayectoria de un solo jugador a través de una secuencia, debido a esto, se ejecutará un algoritmo de Viterbi para cada jugador y así obtener su recorrido individual. El algoritmo se implementará en Matlab ya que éste cuenta con herramientas útiles para el procesamiento de imágenes que pueden manejar los costos computacionales. A continuación se explican detalles de la asignación de variables del algoritmo.

#### 4.5.1 Número de Estados del Modelo $N$

---

Como se ha explicado anteriormente, hay 10 etiquetas posibles para los jugadores de cada equipo. Estas se representan como círculos en el ejemplo de la Figura 9 y dan referencia a su posición física dentro del campo como lo ilustra la Figura 5. Debido a esto, el número de estados posibles en cualquier fotograma del video es de 10. Donde,

$$N = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

#### 4.5.2 Probabilidad de Transición $a_{i,j}$

---

La probabilidad de transición se define como la probabilidad de estar en la posición  $j$  del terreno de juego en un instante de tiempo, dado que te hayas encontrado en la posición  $i$  en el tiempo anterior. Un ejemplo del significado de la probabilidad de transición se puede observar en la Figura 9, donde  $a_{1,1}$  representa la probabilidad de pasar de la posición 1 en el tiempo  $t = 1$ , a la posición 1 en el tiempo  $t = 2$ . De la misma manera,  $a_{10,1}$  denota la probabilidad de pasar de la posición 10 a la posición 1. Para obtener esta probabilidad se comienza por calcular la distancia Euclídea entre la posición de un objeto en el fotograma actual y la posición de todos los objetos detectados en el fotograma siguiente. La representación gráfica del cálculo de esta distancia se encuentra en la Figura 10.

En el ejemplo de la Figura 10, se calcula la distancia de la posición actual de un objeto a las posiciones de todos los objetos detectados en el siguiente fotograma. De esta manera, mientras menor sea la distancia recorrida de un fotograma al siguiente, mayor será la probabilidad de que sea esta la transición del estado.

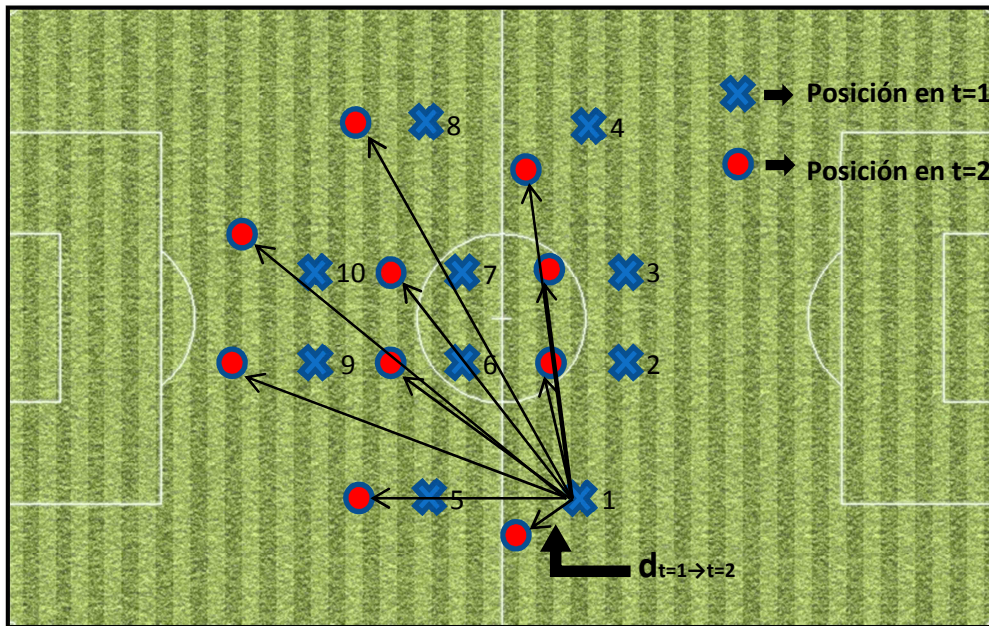


Figura 10 - Demostración Gráfica del Cálculo de Distancia Euclídea

Seguido de obtener las distancias se procede a calcular la probabilidad utilizando la siguiente ecuación:

$$a_{i,j} = P[q_{t+1} = S_j | q_t = S_i] = e^{-\lambda |d_{t_1 \rightarrow t_2}|} \quad (20)$$

Donde  $\lambda$  es un umbral que nos permite afinar el cálculo de la probabilidad de transición, y  $d_{t_1 \rightarrow t_2}$  es la distancia calculada.

#### 4.5.3 Probabilidad de Observación $b_j(k)$

Para definir la probabilidad de observación  $b_j(k)$ , primero se debe detallar el significado de las observaciones. Se tomarán como observaciones las posiciones de los jugadores en el terreno de juego para cada fotograma. Con esta definición de observación se puede resumir la probabilidad de observación como la probabilidad de que un objeto detectado corresponda a un estado u otro dada su posición la imagen. En la Figura 9, la probabilidad de transición en el tiempo  $t = 2$  se representa como  $[b_1(O_{t_2}), \dots, b_{10}(O_{t_2})]$ , donde,  $O_{t_2}$  es la posición en el terreno de juego del jugador en el tiempo  $t = 2$  y  $b_1(O_{t_2})$  es la probabilidad de que el jugador esté en esta posición. Para esta probabilidad se utilizaran los resultados de los mapas probabilísticos y los

histogramas mencionados anteriormente ya que precisamente estos mapas proporcionan la probabilidad de un jugador dada la posición en el terreno de juego.

#### 4.5.4 Etiquetado de la Posición Mediante Viterbi

---

En la Figura 9 se presenta un simple ejemplo de cómo funciona el etiquetado mediante el algoritmo de Viterbi. En dicha figura se ve representado el resultado de la ecuación (19):

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T - 1, T - 2, \dots, 1$$

Esta ecuación obtiene las etiquetas de los jugadores,  $q_t^*$ , desde  $t = 1$  hasta  $t = T$ . En el ejemplo de la Figura 19 se pueden observar los cambios de etiqueta de los jugadores  $J1$  y  $J10$  desde  $t = 1$  hasta  $t = T$  y se representa la trayectoria como la conexión de las flechas negras. De acuerdo con este ejemplo el resultado del etiquetado del jugador  $J1$  es la secuencia,

$$q(J1) = [1, 1, \dots, 2, 2].$$

Este resultado supondría un error en el etiquetado, ya que, el jugador  $J1$  comienza con la etiqueta 1 en el tiempo  $t = 1$  y a lo largo del tiempo cambia su etiqueta a 2. Sin embargo, el etiquetado del jugador  $J10$  es,

$$q(J10) = [10, 10, \dots, 10, 10].$$

En este caso las etiquetas se mantienen iguales durante toda la trayectoria por lo que el etiquetado sería correcto. El error en el etiquetado del jugador  $J1$  es uno muy común donde los sistemas de seguimiento cambian las etiquetas en algún instante, este error es precisamente uno de los problemas que se desea corregir asignando una etiqueta inequívoca a lo largo de todo el tiempo para evitar coincidencias.

## Análisis de Resultados

En este capítulo vamos a mostrar, analizar y comparar los resultados obtenidos con el sistema de seguimiento respecto al mismo sistema con un método de verificación de coincidencias en los etiquetados de los jugadores. Para poder evaluar el sistema utilizaremos los datos del etiquetado manual y se calculará el error obtenido antes y después de utilizar el método mencionado. Dicho error será calculado como el número de veces en que se le asigna una etiqueta equivocada a algún jugador en cada fotograma.

### 5.1 Cálculo del Umbral $\lambda$

El primer paso antes de la simulación del algoritmo implementado es el cálculo del umbral relacionado con la probabilidad de transición que se presenta en la sección 4.5.2. Para escoger este umbral se simuló el sistema con 1200 fotogramas variando el umbral desde 0,1 hasta 1,0 en intervalos de 0,1. A continuación se presenta la Matriz de confusión para  $\lambda = 0,1$ ,  $\lambda = 0,3$  y  $\lambda = 0,5$ .

		Resultado del Etiquetado Mediante el Algoritmo de Viterbi									
		1	2	3	4	5	6	7	8	9	10
Etiqueta Real del Jugador	1	1194	3	0	0	1	0	1	0	1	0
	2	279	911	5	2	0	0	0	0	1	2
	3	1	2	1190	4	0	1	1	0	1	0
	4	147	1	6	1038	3	1	1	1	1	1
	5	443	0	1	3	746	4	1	0	1	1
	6	509	2	1	5	3	675	4	0	1	0
	7	561	1	0	0	0	8	626	4	0	0
	8	0	0	1	0	0	0	6	1186	7	0
	9	346	0	0	0	0	0	0	5	843	6
	10	686	0	2	0	0	1	4	1	11	495

Tabla 1 - Matriz de Confusión para  $\lambda = 0,1$

		Resultado del Etiquetado Mediante el Algoritmo de Viterbi									
		1	2	3	4	5	6	7	8	9	10
Etiqueta Real del Jugador	1	1192	5	0	0	1	0	1	0	1	0
	2	4	1185	5	2	0	1	0	0	1	2
	3	1	2	1190	4	0	1	1	0	1	0
	4	2	1	5	1184	3	1	1	1	1	1
	5	4	0	1	4	1181	5	2	0	1	2
	6	0	2	1	7	4	1180	5	0	1	0
	7	104	2	2	1	0	9	1076	5	0	1
	8	0	0	1	0	0	0	6	1186	7	0
	9	0	1	0	0	1	0	0	7	1181	10
	10	0	2	2	0	3	22	4	2	14	1171

Tabla 2 - Matriz de Confusión para  $\lambda=0,3$

		Resultado del Etiquetado Mediante el Algoritmo de Viterbi									
		1	2	3	4	5	6	7	8	9	10
Etiqueta Real del Jugador	1	1192	5	0	0	1	0	1	0	1	0
	2	4	1185	5	2	0	1	0	0	1	2
	3	1	2	1190	4	0	1	1	0	1	0
	4	2	1	5	1184	3	1	1	1	1	1
	5	4	0	1	4	1181	5	2	0	1	2
	6	0	2	1	7	4	1180	5	0	1	0
	7	0	2	2	1	0	9	1179	6	0	1
	8	0	0	1	0	0	0	6	1186	7	0
	9	0	1	0	0	1	0	0	7	1181	10
	10	0	2	2	0	3	2	4	2	14	1171

Tabla 3 - Matriz de Confusión para  $\lambda=0,5$

La matriz de confusión es una herramienta de visualización donde cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Se representan los resultados de esta manera ya que estas facilitan ver si el sistema se está confundiendo entre clases. Como se puede apreciar en las Tablas 1-3, a medida que se aumenta el umbral, las equivocaciones en el etiquetado de los jugadores disminuyen. Por ejemplo, si comparamos en las tres tablas la fila 7 de la columna 1, veremos que con el umbral en  $\lambda = 0,1$  el

número de veces que le asigna al jugador 7 la etiqueta de jugador 1 es de 561. Mientras que con el umbral en  $\lambda = 0,3$  es de 104 y con  $\lambda = 0,5$  es de 0 veces. Para calcular el error de etiquetado simplemente se suman los elementos en la matriz que no están en la diagonal (Errores) y se divide entre la suma de todos los elementos de la matriz (12000). Los valores de error de esta simulación se encuentran en la Tabla 1 y gráficamente en la Figura 11.

		Fotogramas: 1 - 1200									
Umbral de la Probabilidad de Transición ( $\lambda$ )		0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
Error de Etiquetado		25,8%	4,40%	2,28%	1,90%	1,42%	1,42%	1,42%	1,42%	1,42%	1,42%

Tabla 4 - Porcentaje de Error Dependiendo del Umbral  $\lambda$

Como se puede ver en los datos de la Tabla 4 y la Figura 11, El error comienza en un 25,8% con un umbral de  $\lambda = 0,1$  y disminuye hasta un error de 1,42% con un umbral de 0,5 en adelante. Con estas pruebas se puede decidir de una manera experimental el valor del umbral que nos proporcionara el menor error de etiquetado. De este punto en adelante se utilizara un umbral de  $\lambda = 0,5$  para simular el sistema.

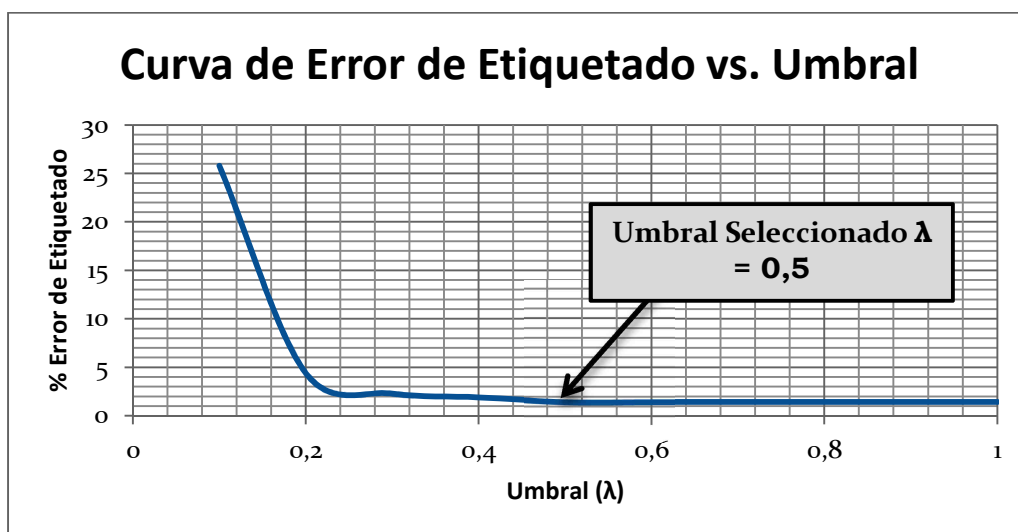


Figura 11 - Curva de Error al cambiar el Umbral ( $\lambda$ )

## 5.2 Simulación del Sistema Inicial

Luego de obtener el valor del umbral, se procedió a simular el sistema para analizar detalladamente los resultados obtenidos. Como se mencionó en la sección anterior, al simular el algoritmo con un umbral de  $\lambda = 0,5$  se obtuvo un error de 1,42%. Este porcentaje de error supone un fallo de un total de 171 etiquetas para todos los jugadores en 1200 fotogramas. Además de los resultados globales, podemos analizar los resultados para cada jugador individualmente. En la Tabla 5 se presentan los fallos encontrados particularmente por cada jugador.

Fotogramas: 1 - 1200										
Jugador	1	2	3	4	5	6	7	8	9	10
Fallos en etiquetado	8	15	10	16	19	20	21	14	19	29
Porcentaje de Error	0,67%	1,25%	0,83%	1,33%	1,58%	1,67%	1,75%	1,16%	1,58%	2,42%

Tabla 5 - Resultados detallados del Etiquetado inicial

Como se puede observar, el jugador que más dificultad tuvo el algoritmo de etiquetar fue el número 10 ya que lo etiquetó incorrectamente 29 veces en una secuencia de 1200 fotogramas, esto se puede deber a que el jugador número 10 está constantemente moviéndose alrededor del terreno de juego y en alguna ocasión su probabilidad puede ser parecida a la de otro jugador. Sin embargo, si se toma en cuenta que el error en el etiquetado es de solo un 2,42% individual y de un 0,242% global, se puede decir que el Algoritmo de Viterbi, con el valor del umbral escogido, realiza la labor del etiquetado de forma eficiente y ha sido un método acertado para resolver el problema planteado.

Para poder apreciar los resultados visualmente, se creó un código en Matlab que obtiene las imágenes reales del partido y les superpone la localización del jugador con un cuadro rojo alrededor de este punto. Además, muestra las etiquetas de los jugadores obtenidas por el algoritmo en cada fotograma y nos permite evaluar los resultados de una manera



diferente. A continuación, en las Figuras 12 – 14 se pueden apreciar las imágenes de varios instantes del partido de fútbol que representan los resultados del algoritmo.

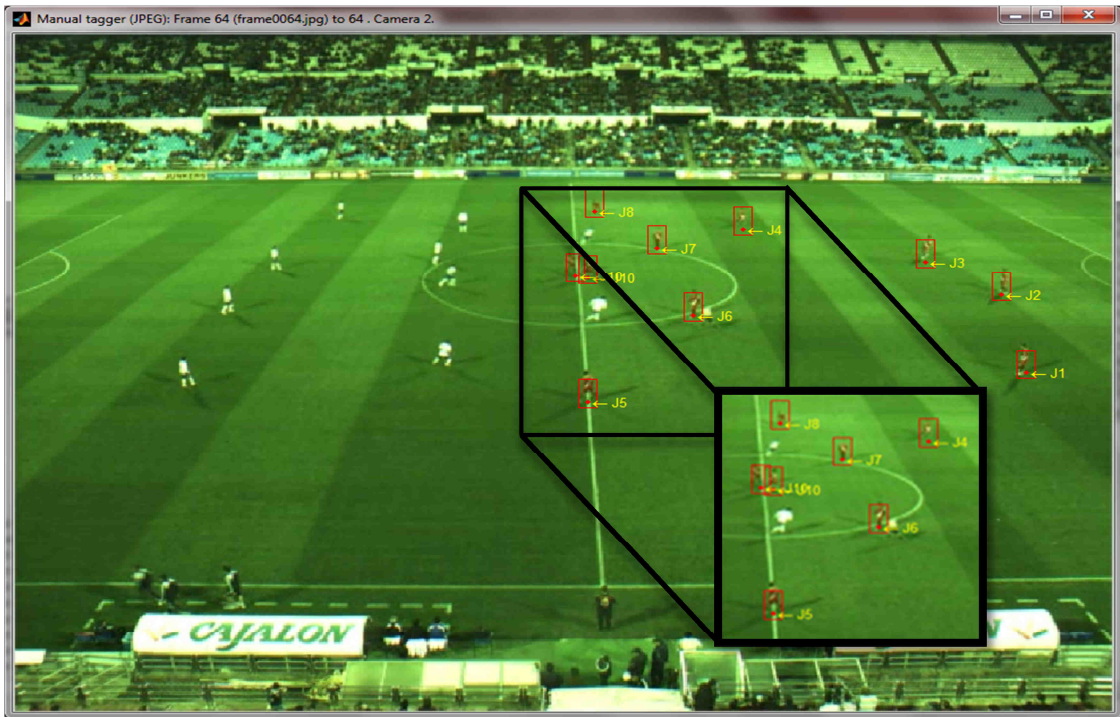


Figura 12 - Resultado de Etiquetado Inicial para el fotograma 64

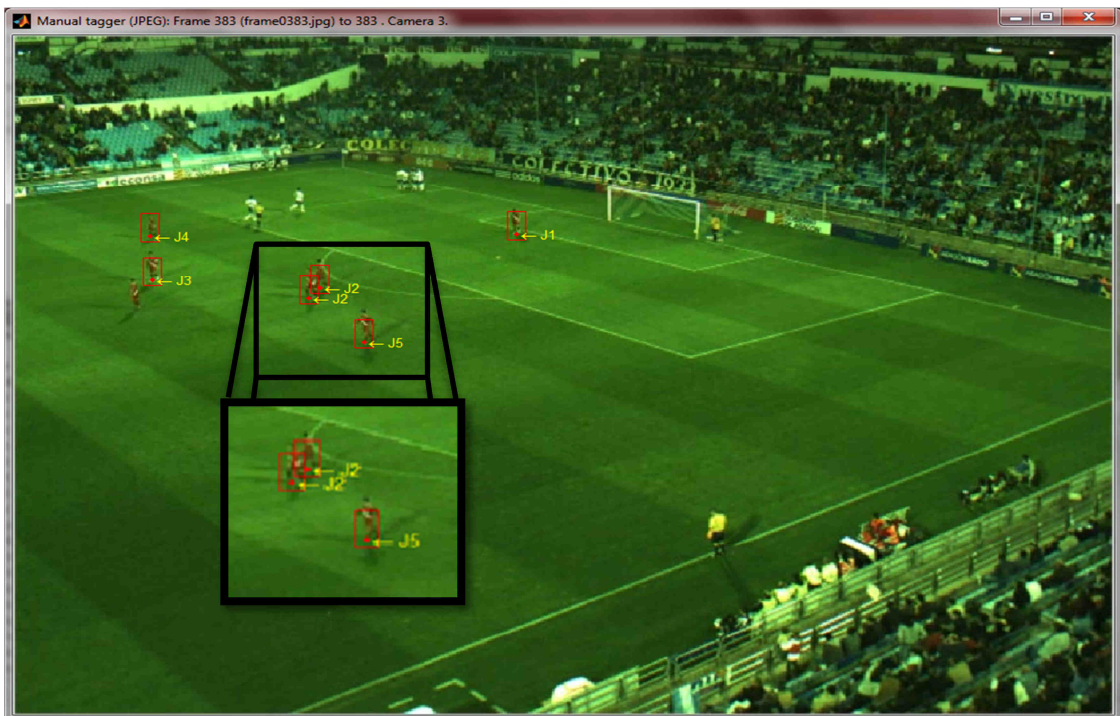


Figura 13 - Resultado de Etiquetado Inicial para el fotograma 383

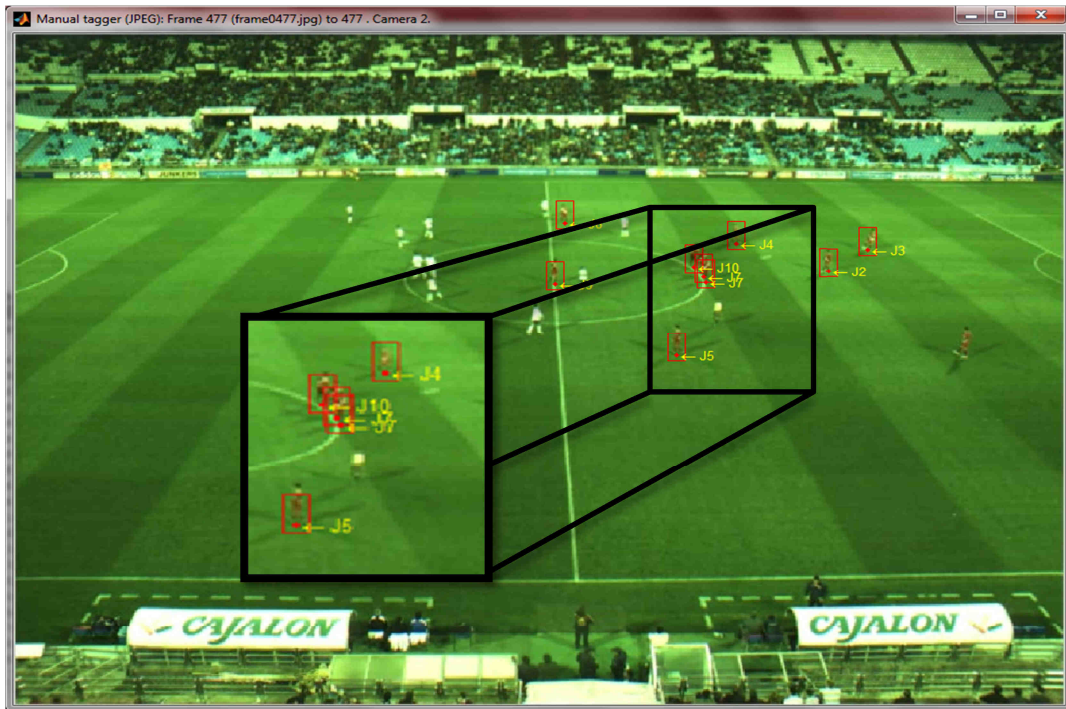


Figura 14 - Resultado del Etiquetado Inicial para el fotograma 477

En las figuras anteriores se pueden observar algunos de los errores comunes que se obtuvieron en el etiquetado. Como se ve en las imágenes, el error que encontramos con el algoritmo es que coinciden las etiquetas en más de un jugador debido a las oclusiones que ocurren durante el partido. Por ejemplo, en la Figura 12 (fotograma 64) se repite la etiqueta del jugador 10 un error común al estar ambos jugadores tan cerca el uno del otro, mientras que en la Figura 13 (fotograma 383) se repite la etiqueta del jugador 2 y finalmente en la Figura 14 (fotograma 477) se repite el jugador 7 ya que un jugador está ocultando totalmente al otro.

### 5.3 Método de Corrección de Etiquetas

---

A pesar de que los resultados obtenidos anteriormente producen un error muy bajo y aceptable, en esta sección se presenta una técnica para eliminar el error en el etiquetado mediante una alteración del Algoritmo de Viterbi. Para continuar, debemos recordar que el Algoritmo de Viterbi obtiene la mejor trayectoria de los estados en una secuencia. Para esto utiliza la variable  $\delta_t(j)$  que representa la probabilidad de los estados a través

de una trayectoria. En la implementación del algoritmo este término almacena los valores de la probabilidad de estados para todos los fotogramas en la secuencia. Para explicar cómo funciona el método, consideremos la situación del fotograma presentado en la Figura 12. Para facilitar el proceso llamaremos a los jugadores que se encuentran con la misma etiqueta el Jugador A y el Jugador B. Como se mencionó anteriormente, en el fotograma 383, el resultado del Algoritmo de Viterbi asignó a los dos jugadores la etiqueta del jugador 2. Esto se debe a que en el fotograma 383 el valor de  $\delta_t(j)$  para el jugador A y B es el siguiente:

$\delta_t(j)$ del Jugador A			$\delta_t(j)$ del Jugador B		
J	Fotograma 383	Max	J	Fotograma 383	Max
1	3,42E-10		1	4,39E-10	
2	1,86E-07	1,86E-07	2	1,29E-07	1,29E-07
3	8,34E-09		3	6,20E-09	
4	1,70E-10		4	1,91E-10	
5	1,30E-08		5	4,72E-09	
6	1,85E-07		6	6,96E-08	
7	8,06E-09		7	3,78E-09	
8	2,30E-12		8	1,41E-12	
9	6,15E-11		9	2,63E-11	
10	8,42E-11		10	5,18E-11	

Tabla 6 - Tabla de Valores de  $\delta_t(j)$  para el fotograma 383

El Algoritmo de Viterbi llega a su solución buscando el máximo de los valores de  $\delta_t(j)$  para el fotograma y el índice de este valor es el número de la etiqueta del jugador. En este caso, como se ve en la Tabla 6, ambos jugadores (A y B) tienen la mayor probabilidad de tener la etiqueta 2 y por eso se repite este valor en el resultado. Gracias a que el valor de  $\delta_t(j)$  es almacenado para todos los fotogramas de la secuencia, se utilizarán estos valores para resolver las repeticiones. Cuando se obtengan los resultados del Algoritmo de Viterbi, se verifican los estados en cada fotograma para asegurar que no hay una repetición de etiquetas. Si sucede que hay una repetición de etiquetas, se toma el valor de  $\delta_t(j)$  para ese fotograma, el

fotograma siguiente y el anterior y se suman los valores de probabilidad de estos. Luego de sumar los valores de  $\delta_t(j)$  nuevamente se busca el máximo y el índice de este. Finalmente, se verifica que no haya repeticiones nuevamente, si no las hay se le asignan los nuevos estados a los jugadores. Sin embargo, si hay repeticiones, se le vuelve a sumar más fotogramas anteriores y siguientes a  $\delta_t(j)$  y comienza el proceso de nuevo hasta que se resuelva el error. Si volvemos al ejemplo del fotograma 383 y hacemos el procedimiento mencionado, el resultado es el siguiente:

J	$\delta_t(j)$ del Jugador A			Sum	Max
	Fotograma 382	Fotograma 383	Fotograma 384		
1	2,69E-10	3,42E-10	1,34E-10	7,45E-10	
2	1,19E-07	1,86E-07	9,60E-08	4,01E-07	
3	4,29E-09	8,34E-09	6,81E-09	1,94E-08	
4	2,20E-10	1,70E-10	1,78E-10	5,68E-10	
5	1,02E-08	1,30E-08	5,11E-09	2,83E-08	
6	2,42E-07	1,86E-07	1,95E-07	6,22E-07	6,22E-07
7	6,34E-09	8,06E-09	6,81E-09	2,12E-08	
8	1,81E-12	2,30E-12	2,18E-12	6,29E-12	
9	4,84E-11	6,15E-11	6,43E-11	1,74E-10	
10	6,63E-11	8,42E-11	7,90E-11	2,30E-10	

Tabla 7 -  $\delta_t(j)$  del Jugador A en los fotogramas 382 - 384

J	$\delta_t(j)$ del Jugador B			Sum	Max
	Fotograma 382	Fotograma 383	Fotograma 384		
1	3,96E-10	4,39E-10	1,16E-10	9,50E-10	
2	1,92E-07	1,29E-07	9,12E-08	4,12E-07	4,12E-07
3	3,87E-09	6,20E-09	3,71E-09	1,38E-08	
4	2,83E-10	1,91E-10	1,35E-10	6,08E-10	
5	4,26E-09	4,72E-09	1,27E-09	1,03E-08	
6	9,47E-08	6,96E-08	4,50E-08	2,09E-07	
7	3,41E-09	3,78E-09	2,39E-09	9,58E-09	
8	1,27E-12	1,41E-12	9,14E-13	3,59E-12	
9	2,37E-11	2,63E-11	1,86E-11	6,85E-11	
10	4,67E-11	5,18E-11	3,36E-11	1,32E-10	

Tabla 8 -  $\delta_t(j)$  del Jugador B en los fotogramas 382 - 384

Se han tomado los valores de  $\delta_t(j)$  para un fotograma antes y un fotograma después y se ha sumado su valor de probabilidad. Seguidamente se ha calculado el índice del valor máximo para obtener la nueva etiqueta que se le asigna a los Jugadores A y B. Como aparece en la Tabla 7, la etiqueta del Jugador A ha cambiado de ser la numero 2 a la numero 6 y en la Tabla 8 la etiqueta del Jugador B se ha mantenido en la numero 2. Por lo tanto, en este ejemplo solo con sumar los valores de  $\delta_t(j)$  anteriores y siguientes se pudo resolver el problema de la coincidencia de etiquetas. Este proceso hará que el sistema adquiera todavía más fiabilidad y resolverá los problemas de etiquetado de objetos con mayor precisión. Con esta técnica implementada se procede a simular nuevamente el sistema para comparar los resultados y verificar su eficacia.

#### 5.4 Simulación del Sistema Final

Después de haber resuelto los problemas de las coincidencias en las etiquetas se procedió a simular el sistema completo con los mismos parámetros de las simulaciones anteriores. La matriz de confusión final se presenta en la Tabla 9.

		Resultado del Etiquetado Mediante el Algoritmo de Viterbi									
		1	2	3	4	5	6	7	8	9	10
Etiqueta Real del Jugador	1	1200	0	0	0	0	0	0	0	0	0
	2	0	1199	0	1	0	0	0	0	0	0
	3	0	0	1200	0	0	0	0	0	0	0
	4	0	1	0	1199	0	0	0	0	0	0
	5	0	0	0	0	1200	0	0	0	0	0
	6	0	0	0	0	0	1200	0	0	0	0
	7	0	0	0	0	0	0	1200	0	0	0
	8	0	0	0	0	0	0	0	1200	0	0
	9	0	0	0	0	0	0	0	0	1200	0
	10	0	0	0	0	0	0	0	0	0	1200

Tabla 9 - Matriz de confusión de la simulación del sistema final

Fotogramas: 1 - 1200										
Jugador	1	2	3	4	5	6	7	8	9	10
Fallos en etiquetado	0	1	0	1	0	0	0	0	0	0
Porcentaje de Error	0%	0,084%	0%	0,084%	0%	0%	0%	0%	0%	0%

Tabla 10 - Resultados detallados del Etiquetado Final

Al comparar la Tabla 9 con la Tabla 3 se puede apreciar claramente la diferencia que ha hecho el método de corrección de etiquetas. El sistema ha pasado de tener 171 errores de etiquetado a solo 2 errores lo que significa un error final de 0,0167%, un valor insignificante si se compara con el error anterior de 1,42%.

En el análisis detallado de los fallos de los jugadores, que se presenta en la Tabla 10, se ve como las únicas etiquetas en las cuales se equivocó el sistema fue en una etiqueta del jugador 4 y en una etiqueta del jugador 2. Al investigar el fallo cometido, resulta que en el fotograma 304 se intercambian las etiquetas de ambos jugadores, sin embargo, el mismo Algoritmo de Viterbi corrige este fallo ya que el método busca la mejor trayectoria global por lo que en el próximo fotograma ya está corregido su estado. A continuación, en las Figuras 15 – 17 se presentan las imágenes de las mismas situaciones mostradas anteriormente con las etiquetas incorrectas para probar la corrección del método final. Como se ve claramente en las imágenes, el método añadido al algoritmo ha corregido los errores que se obtenían anteriormente y muestra un etiquetado correcto de los jugadores en el terreno de juego. En el próximo capítulo se ofrecerán conclusiones al trabajo hecho y se mencionaran posibles trabajos futuros.

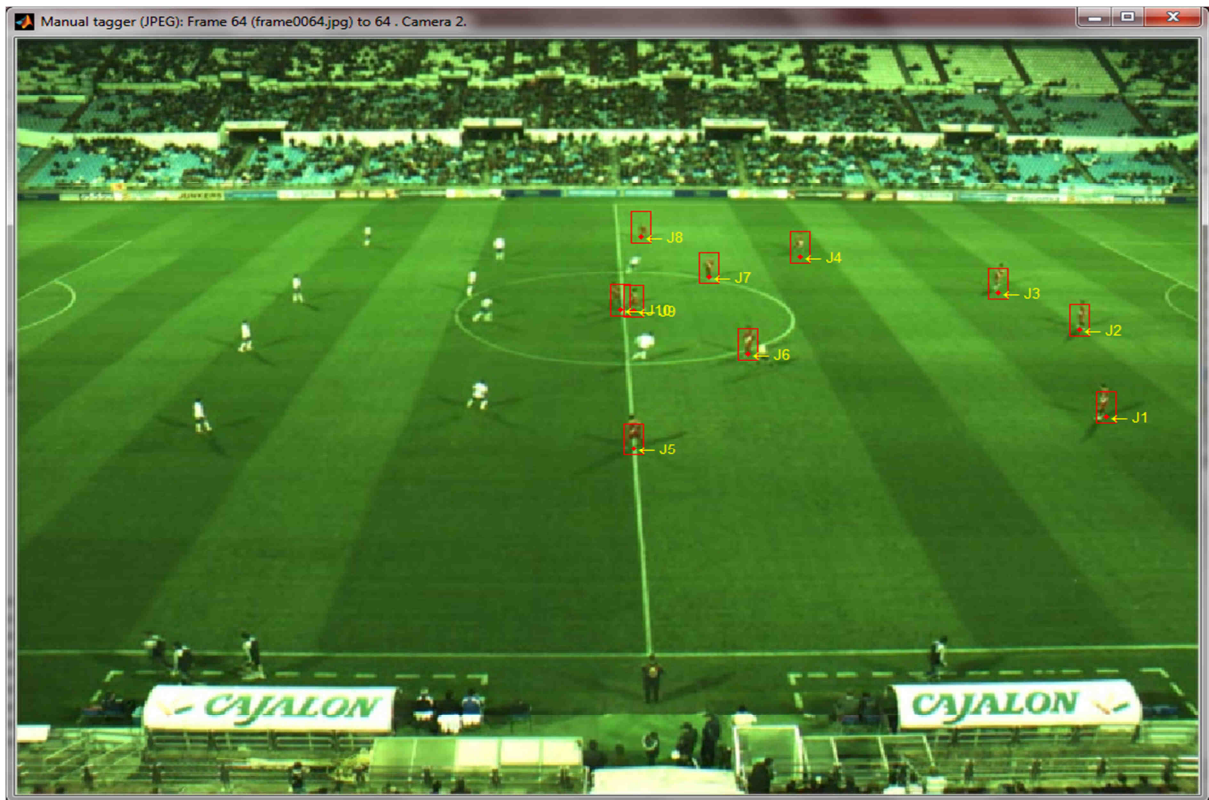


Figura 15 - Resultado de Etiquetado Final para el fotograma 64



Figura 16 - Resultado de Etiquetado Final para el fotograma 383

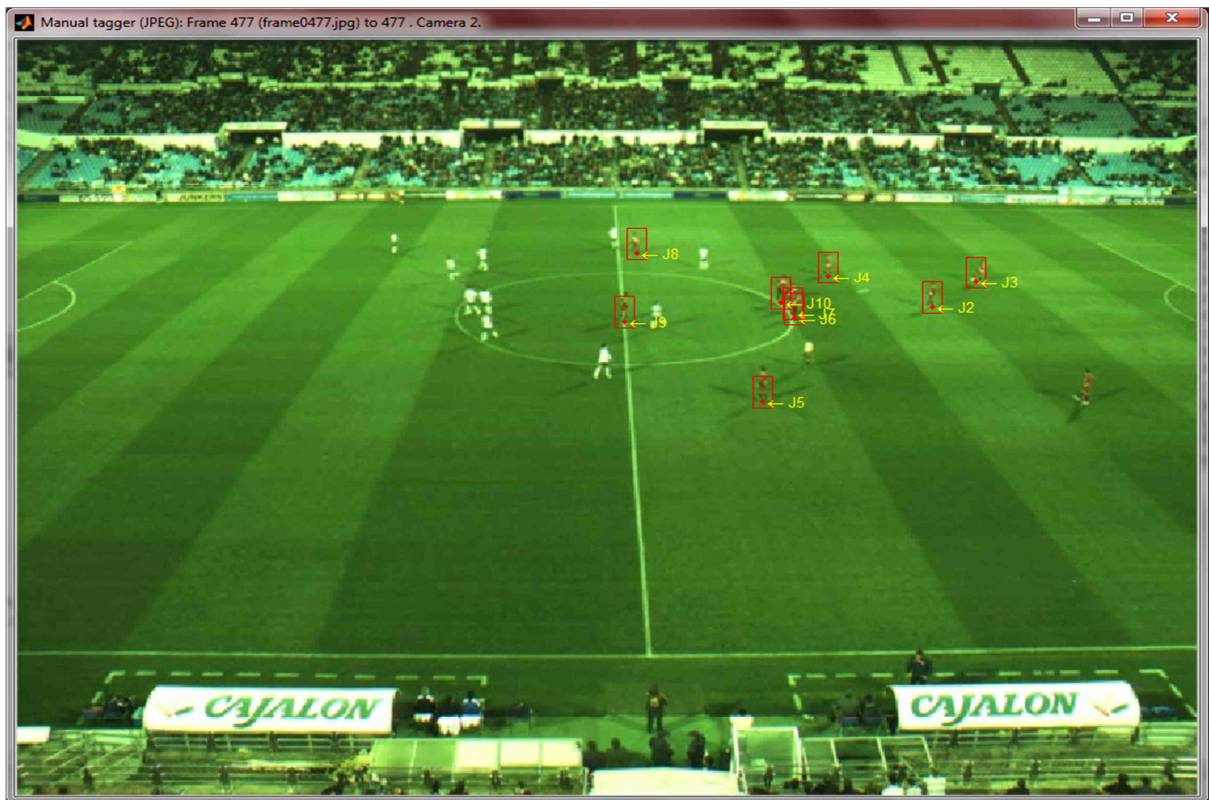


Figura 17 - Resultado de Etiquetado Final para el fotograma 477



## Conclusiones

---

En este Trabajo Fin de Master se ha presentado un método para solucionar el problema del seguimiento de múltiples objetos mediante técnicas de visión por computador y análisis de probabilidades. Para validar el sistema se ha aplicado el método a las imágenes de un partido de fútbol, ya que por la naturaleza del deporte, existen muchos instantes donde los objetos en la imagen se ocluyen parcial o totalmente y presentaría un entorno útil para verificar la fiabilidad del sistema. En el capítulo 5 se pueden observar los resultados de las simulaciones del algoritmo.

El sistema inicial, con la implementación del Algoritmo de Viterbi, se simuló para un total de 1200 fotogramas aproximadamente 10 min. En esta simulación se obtuvieron 171 etiquetas incorrectas de las 12000 etiquetas que se asignan, lo que corresponde a un error de 1,425%. Aun teniendo un error tan bajo, se exploró una manera de hacer que el sistema sea más robusto y resolviera mejor las oclusiones. Para esto se alteró el Algoritmo de Viterbi añadiendo un método de corrección de coincidencias demostrado en la sección 5.3. El cambio en el algoritmo supuso una disminución en el error total del 1,42% al 0,0167% lo que equivale a solo 2 errores en las etiquetas.

Con estos resultados se demuestra que el Algoritmo de Viterbi, con algunas alteraciones, es un método idóneo para resolver problemas complicados de oclusiones parciales y totales para obtener las trayectorias de los jugadores a través de la secuencia de imágenes. El porcentaje de error resultante muestra que el sistema es robusto y fiable para la situación dada

y deja la puerta abierta para aplicarlo a otras situaciones de seguimiento de objetos.

El sistema propuesto ha logrado el objetivo principal planteado en este trabajo, encontrar un método capaz de seguir objetos en una secuencia de imágenes asignándole a cada uno una etiqueta inequívoca durante toda una secuencia. De la misma manera se han alcanzado los objetivos intermedios:

- Encontrar o desarrollar un algoritmo de seguimiento de múltiples objetos.
- Minimizar la tasa de error en el seguimiento, teniendo en cuenta situaciones como cruces y oclusiones.

Un objetivo no mencionado, pero aun así muy importante, es obtener los menores tiempos de ejecución del sistema posibles. Este parámetro es significativo ya que te asegura que has implementado el método de la manera más eficiente posible y permite posibilidad de ejecutar el sistema en tiempo real. El sistema, ejecutado en el entorno de Matlab con un procesador AMD E-350 @ 1.60 GHz y 4,00 GB de memoria RAM, tiene un tiempo de ejecución de 3.825s, un valor normal debido a la complejidad de los cálculos.

## ***6.1 Propuestas de Desarrollo Futuro***

---

El algoritmo de seguimiento planteaba un gran problema principal, los cruces y oclusiones que se producen entre jugadores dentro del campo de fútbol. En especial en jugadas en las que hay una gran acumulación de personas en una zona del campo. Un posible trabajo futuro sería intentar aumentar la eficacia del algoritmo implementado utilizando más puntos de vista que los disponibles en las imágenes utilizadas en este trabajo, esto significaría utilizar un mayor número de cámaras. Además, sería interesante poder hacer el seguimiento mediante visión por computador, no por la posición de los jugadores en el campo, sino por la silueta de los números de las camisetas lo que necesitaría instalar cámaras de una mayor resolución en muchos puntos del campo. Otra propuesta para un trabajo futuro podría ser la implementación del sistema en tiempo real, recibiendo las imágenes del campo mientras el partido se está disputando y poder analizar las trayectorias de los jugadores. Esta propuesta tendría la posibilidad de convertirse en un producto comercial ya que los equipos deportivos utilizan dispositivos tecnológicos para mejorar su juego. Sin embargo, esto significaría que el método hay que pasarlo del lenguaje de Matlab a otro lenguaje capaz de hacer los cálculos más rápido.

# Anexo I: Cálculo de la matriz de Homografía

---

Existen dos razones por la que es recomendable realizar un seguimiento sobre el plano del suelo en vez de sobre la imagen:

- Mejorar la precisión de seguimiento de los jugadores, ya que se reduce el efecto del error perspectivo en la imagen.
- Poder realizar un seguimiento de objetos a partir de los datos obtenidos de forma simultánea por varios sensores a la vez, uniéndolos a continuación las diferentes observaciones en un plano común.

Para trasladar la información obtenida en la imagen al plano aplicaremos una transformación proyectiva. Como aproximación asumiremos que no existe ninguna distorsión creada por la lente de la cámara. Un punto en el plano proyectivo representará un rayo que pasa por el origen en el espacio 3D, tal y como podemos ver en la Figura 18.

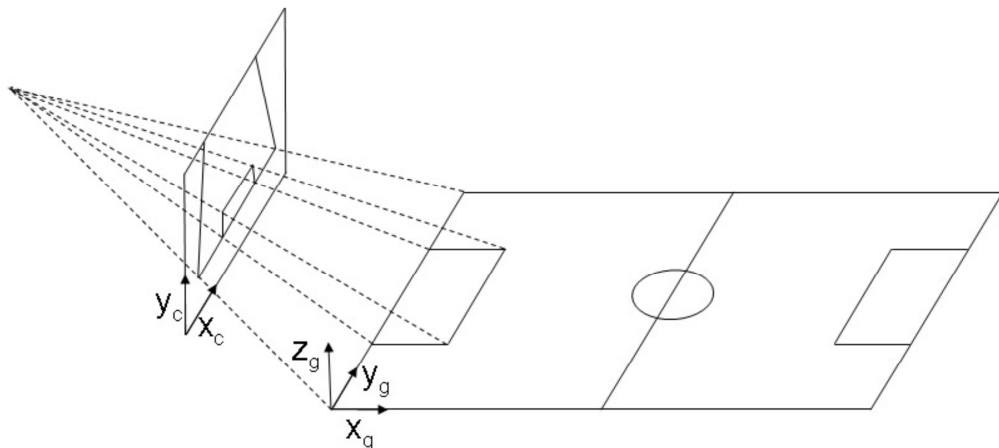


Figura 18 - Ejemplo gráfico de la transformación de coordenadas de la imagen al plano

La transformación proyectiva entre dos planos puede ser representada como una transformación lineal donde,  $p_2 = T_{21}p_1$ . Si queremos volver al sistema de referencia anterior podremos usar como matriz de transformación  $T_{21}^{-1}$ , de manera que la transformación también sea lineal.

Si dicha transformación es representada en coordenadas cartesianas, los resultados no serán lineales. Sean:

$p_c = (x_i, y_i, 1)^T$ : Las coordenadas del punto  $i$  en el sistema de referencia de la cámara.

$p_g = (x_i^g, y_i^g, 1)$ : Las coordenadas homogéneas asociadas a las coordenadas del punto  $i$  en el sistema de referencia del plano del suelo  $(x_i^g, y_i^g)$ .

$T_{gc}$ : La transformación proyectiva para las coordenadas  $p_c$  y  $p_g$ .

Para obtener dicha transformación proyectiva, para cada par  $i$  de puntos se tomará esta ecuación:

$$(\lambda_i x_i^g, \lambda_i y_i^g, \lambda_i)^T = (T_{gc}(x_i, y_i, 1))^T$$

Si la desarrollamos:

$$\begin{pmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i^g x_i & -x_i^g y_i & -x_i^g \\ 0 & 0 & 0 & x_i & y_i & 1 & -y_i^g x_i & -y_i^g y_i & -y_i^g \end{pmatrix} t = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Donde  $t = (t_{11} \ t_{12} \ t_{13} \ t_{21} \ t_{22} \ t_{23} \ t_{31} \ t_{32} \ t_{33})^T$  es un vector formado por los elementos de la matriz de homografía  $T_{gc}$ . Si usamos cuatro pares de puntos (En nuestro caso, por ejemplo, podríamos tomar los cuatro puntos asociados a los extremos del campo que se pueden ver cada cámara - por cada cámara tendremos una homografía), de forma que tres de ellos no sean colineales, podremos construir una matriz  $M$  de  $8 \times 9$  elementos, donde  $Mt = 0$ . De esta manera, la solución  $t$  corresponde al vector propio asociado con el menor valor propio (en este caso el valor propio nulo) de la matriz  $MTM$ ; el cuál puede ser calculado mediante una descomposición en valores singulares de la matriz  $M$ . Una vez hallemos la matriz de homografía  $T_{gc}$ , podremos realizar su transformación inversa para obtener la homografía en el sistema de referencia deseado.



# Bibliografía

---

- [1] J. Black, T. Ellis, and P. Rosin, "Multi-View Image Surveillance and Tracking," in *IEEE Workshop on Motion and Video Computing*, 2002.
- [2] A. Mittal and L. Davis, "M2tracker: a Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene," *Computer Vision and Image Understanding*, vol. 51(3), pp. 189–203, 2003.
- [3] S. Iwase and H. Saito, "Parallel Tracking of All Soccer Players by Integrating Detected Positions in Multiple View Images," in *International Conference on Pattern Recognition*, August 2004, pp. 751–754.
- [4] M. Xu, J. Orwell, and G. Jones, "Tracking Football Players With Multiple Cameras," in *International Conference on Image Processing*, October 2004, pp. 2909–2912.
- [5] D. R. Magee, "Tracking multiple vehicles using foreground, background and motion models," *Image and Vision Computing*, vol. 22, no. 2, pp. 143–155, February 2004.
- [6] J.J. La Viola, "A comparison of unscented and scented Kalman filtering for estimating quaternion motion," In *Proc. American Control Conference*, pp. 2435-2440 (2003).
- [7] E.A. Wan and R. Van der Merwe. "The unscented Kalman filter for nonlinear estimation," In *Proc. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC)*, pp.153-158, Lake Louise, AB, Canada, (2000).
- [8] J. Vermaak, A. Doucet, and P. Perez, "Maintaining Multimodality Through Mixture Tracking," in *International Conference on Computer Vision*, October 2003, pp. 1110–1116.
- [9] K. Smith, D. Gatica-Perez, and J.-M. Odobez, "Using Particles to Track Varying Numbers of Interacting People," in *Conference on Computer Vision and Pattern Recognition*, 2005.
- [10] Z. Khan, T. Balch, and F. Dellaert, "Mcmc-Based Particle Filtering for Tracking a Variable Number of Interacting Targets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805–1918, 2005.
- [11] C. Yang, R. Duraiswami, and L. Davis, "Fast multiple object tracking via a hierarchical particle filter," in *International Conference on Computer Vision*, 2005.
- [12] T. Mauthner, M. Donoser, and H. Bischof, "Robust Tracking of Spatial Related Components," in *International Conference on Pattern Recognition*, 2008.
- [13] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking," in *European Conference on Computer Vision*, May 2004.
- [14] P. Willet, Y. Ruan, and R. Streit, "The PMHT: Its problems and some solutions," in *IEEE Transactions on Aerospace and Electronic Systems*, vol.38 no.3, pp.738-754 (Jul.2002).
- [15] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and H. Wensheng, "Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions," in *Conference on Computer Vision and Pattern Recognition*, June 2006, pp. 666–673.
- [16] Wu, B., Nevatia, R., "Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors," in *International Journal of Computer Vision* (2007)

- [17] C. Huang, B. Wu, and R. Nevatia, "Robust Object Tracking by Hierarchical Association of Detection Responses," in *European Conference on Computer Vision*, 2008, pp. 788–801.
- [18] P. Nillius, J. Sullivan, and S. Carlsson, "Multi-Target Tracking - Linking Identities Using Bayesian Network Inference," in *Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2187–2194.
- [19] R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
- [20] J. Wolf, A. Viterbi, and G. Dixon, "Finding the Best Set of K Paths Through a Trellis With Application to Multitarget Tracking," in *Aerospace and Electronic Systems*, IEEE Transactions on, vol. 25, no. 2, pp. 287–296, March 1989.
- [21] H. Jiang, S. Fels, and J. Little, "A Linear Programming Approach for Multiple Object Tracking," in *Conference on Computer Vision and Pattern Recognition*, 2007, pp. 744–750.
- [22] L. Zhang, Y. Li, and R. Nevatia, "Global Data Association for Multi-Object Tracking Using Network Flows," in *Conference on Computer Vision and Pattern Recognition*, 2008.
- [23] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," in *IEEEASSP Mag.*, pp. 4-16, Jan. 1986.
- [24] L. R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. IEEE*, vol. 77, no. 2, pp. 257-286, Feb. 1989.
- [25] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," in *IEEE Trans. Information Theory*, vol. IT-13, pp. 260-269, Apr. 1967.
- [26] G.D. Forney, "The Viterbi algorithm," in *Proc. IEEE*, vol. 61, pp. 268-278, Mar. 1973.
- [27] J. Martínez-del-Rincón, E. Herrero, J. Raúl, C. Orrite, C. Medrano, M. Montañés, "Multicamera sport player tracking with Bayesian estimation of measurements," in *Optical Engineering*, Apr. 2009