



# Análisis Automático del Rendimiento de los Jugadores en Partidos de Tenis

Izaguirre, Agustín Emilio

Tutor: Dra. María Juliana Gambini

Instituto Tecnológico de Buenos Aires (ITBA)

2020



# Resumen

El problema de seguimiento de objetos en secuencias de imágenes y en videos tiene múltiples aplicaciones en distintos aspectos, como el estudio del medio ambiente, el estudio del comportamiento de animales, la robótica, la seguridad y el diagnóstico médico, entre otras.

Una de las aplicaciones que posee el seguimiento de objetos, que se desarrolló en los últimos años, es la posibilidad de detectar acciones en eventos deportivos, en tiempo real, lo que resulta muy útil para colaborar en la tarea del árbitro y también para analizar el rendimiento de los jugadores.

Los diferentes eventos deportivos donde se pueden utilizar son eventos de fútbol, vóley, handball, etc.

En este trabajo, se aplica un método de seguimiento de objetos en secuencias de imágenes para detectar resultados de jugadas en partidos de tenis y realizar un análisis estadístico del rendimiento deportivo de cada jugador.

# Índice general

<b>1. Introducción</b>	<b>3</b>
<b>2. Estado del Arte</b>	<b>5</b>
2.1. Modelado del objeto . . . . .	5
2.2. Detección del objeto de interés . . . . .	12
2.3. Seguimiento del objeto . . . . .	16
2.4. Problemas en el seguimiento de objetos . . . . .	25
2.5. YOLO: Real-Time Object Detection . . . . .	28
<b>3. Seguimiento de Jugadores en Eventos Deportivos</b>	<b>31</b>
3.1. Detección de la cancha . . . . .	31
3.2. Seguimiento de la pelota . . . . .	40
3.3. Detección de cambios de movimiento de la pelota . . . . .	44
3.4. Análisis del rendimiento del jugador en el video . . . . .	48
<b>4. Resultados</b>	<b>49</b>
<b>5. Conclusiones</b>	<b>71</b>

# Capítulo 1

## Introducción

A medida que avanza la tecnología, se ha incrementado el uso de cámaras y el almacenamiento de datos en forma digital. Por esta razón se incrementa la necesidad de analizar automáticamente los datos capturados por las cámaras. Uno de los procesos fundamentales para lograrlo es el seguimiento de objetos en video.

Las imágenes y videos presentes en el mundo digital, combinados con la potencialidad que ofrecen las computadoras actuales, han facilitado la creación de aplicaciones sofisticadas en el área de procesamiento de imágenes y videos. Una de estas aplicaciones es el conteo de jugadas, aciertos y desaciertos para realizar estadísticas en eventos deportivos. Este análisis permite evaluar el rendimiento de un jugador, técnica que se utiliza especialmente en jugadores profesionales.

Los seres humanos reconocemos fácilmente gran cantidad de objetos, sujetos a diferentes condiciones de iluminación, distintos tamaños o escalas e incluso si se encuentran rotados, trasladados u obstruidos. Sin embargo esta tarea no es sencilla de realizar desde el punto de vista informático, porque los métodos automáticos de seguimiento fallan frente a cambios en iluminación, posición del objeto y borroneo, entre otros problemas [21].

Un objeto de la imagen es cualquier elemento de interés dentro de la escena, que puede ser detectado. En un ambiente dinámico tanto los objetos como el fondo pueden variar.

El seguimiento de objetos en videos consiste en detectar un conjunto de

objetos en el cuadro inicial, identificando a cada uno y estimar la posición de los mismos a medida que el video avanza en el tiempo.

El seguimiento de objetos puede ser destinado a numerosas aplicaciones. En la actualidad se utilizan técnicas de seguimiento de objetos en video en tiempo real, permitiendo el desarrollo de nuevos videojuegos basados en realidad aumentada. Puede utilizarse también sistemas de seguridad automatizados, diagnóstico médico, asistencia en cirugías y monitoreo ambiental, entre otras.

El objetivo de este trabajo es realizar automáticamente conteos de jugadas, aciertos y desaciertos de un jugador de tenis. Para lograr este objetivo, es necesario realizar el seguimiento de un jugador, de la pelota, de la cancha y de la red a lo largo de la secuencia de imágenes.

El trabajo está compuesto de la siguiente manera: en el Capítulo 2, se presenta el estado del arte para el seguimiento de los objetos en secuencias de imágenes. En el Capítulo 3, se describen los métodos utilizados para la detección de la cancha, el seguimiento de los jugadores y la pelota y el cálculo de las estadísticas. En el Capítulo 4 se comentan los resultados obtenidos utilizando los métodos previamente mencionados. Para finalizar, el Capítulo 5 contiene las conclusiones del trabajo realizado.

# Capítulo 2

## Estado del Arte

El objetivo de este capítulo es realizar un resumen de los métodos que se utilizan para el seguimiento de objetos en videos, dar a conocer los algoritmos que los implementan y las ventajas y desventajas de su utilización.

Los principales problemas en el seguimiento de objetos son el modelado del objeto, cambios en la apariencia y forma del objeto, movimientos repentinos, cambios en la iluminación, sombra y reflexiones, ruido y oclusión.

La arquitectura típica del seguimiento de objetos se basa en tres módulos interdependientes:

- Modelado del objeto
- Detección del objeto
- Seguimiento del objeto

A continuación se detallan los métodos utilizados en cada módulo [28, 14].

### 2.1. Modelado del objeto

El modelado del objeto es un aspecto importante para el seguimiento de los objetos. Una de las mayores dificultades encontradas para modelar un objeto es encontrar una apropiada descripción visual que permita distinguir el objeto del fondo y de los demás objetos.

En un escenario de seguimiento, un objeto puede ser definido como cualquier cosa que resulte de interés para un posterior análisis. Como por ejemplo

autos en la ruta, burbujas en el agua, pelota en un partido de tenis, entre otros.

Los objetos a seguir pueden ser representados por sus formas y apariencias.

Para modelar el objeto, además de su representación también pueden tenerse en cuenta las características del mismo.

Seleccionar las características correctas del objeto a seguir es de suma importancia, ya que de esta manera los objetos pueden ser distinguidos fácilmente. La propiedad más importante que se desea que cumpla la característica seleccionada, es su unicidad con respecto al espacio donde se realiza el seguimiento.

Muchos algoritmos de seguimiento suelen utilizar una combinación de métodos de representación del objeto junto con una selección de características del mismo para modelar el objeto de interés

En esta sección se analizan los métodos utilizados para modelar el objeto de interés.

## **Representación del objeto**

### **Representación por puntos**

El objeto se representa con un solo punto, indicando alguna característica importante del mismo, por ejemplo su centro de masa ó utilizando un conjunto de puntos. Esta representación es aceptable para objetos que ocupan un espacio pequeño dentro de la imagen.

En la Figura 2.1(a) se observa como se representa al objeto con un solo punto en el centro de masa, mientras que la Figura 2.1(b) muestra como puede representarse un objeto utilizando un conjunto de puntos.

Uno de los beneficios de representar al objeto como un punto es que es más fácil de procesar y aplicar en algoritmos complicados.



(a) Utilizando centro de masa.

(b) Utilizando conjunto de puntos.

Figura 2.1: Representación por puntos utilizando el centroide y un conjunto de puntos.

### Formas geométricas primitivas

La representación de puntos no logra capturar todas las posibles dinámicas del objeto, como por ejemplo rotación, escalado, etc. Se necesitan de formas geométricas para este tipo de problemas, las formas más populares son el rectángulo, cuadrado, círculo y elipse.

En la Figura 2.2 se muestra cómo se representan los objetos utilizando un rectángulo y una elipse.

Estas formas geométricas son más apropiadas para representar simples objetos rígidos y con métodos adaptativos también pueden usarse para objetos no rígidos.

El movimiento para estas representaciones es modelado por una transformación de traslación, afín u homografía.



(a) Utilizando un rectángulo.



(b) Utilizando una elipse.

Figura 2.2: Representación por figuras geométricas utilizando un rectángulo y una elipse.

### Modelos de formas articuladas

Las formas articuladas son usadas para seguimiento de objetos de interés que estén divididos en porciones y cada una de ellas estén descritas individualmente.

Los objetos articulados están compuestos por partes, las cuales están unidas por articulaciones formando un cuerpo. Por ejemplo el cuerpo humano está formado por el torso, las piernas, la cabeza, los pies y está conectado por articulaciones. La Figura 2.3 muestra un ejemplo de un cuerpo formado por un conjunto de elipses. La relación entre las partes esta controlada por modelos de movimiento cinemático como por ejemplo el ángulo en las articulaciones y los grados de libertad que puedan tener las mismas.

Para poder representar un objeto articulado se puede modelar las partes que constituyen el cuerpo con cualquier forma geométrica primitiva y a las articulaciones con las intersecciones entre éstas.

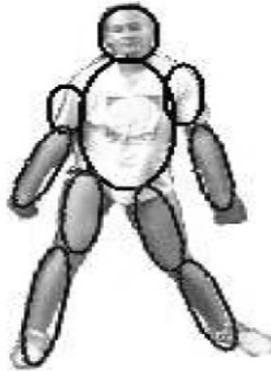


Figura 2.3: Representación utilizando modelos de formas articuladas.

### Modelos esqueléticos

Este modelo es comunmente utilizado como una representación de formas para el reconocimiento de objetos. Un modelo esquelético puede ser utilizado para modelar tanto objetos rígidos como objetos articulados.

Podemos definir al esqueleto como un conjunto de articulaciones dentro del objeto que describe las dependencias y define las restricciones entre la representación y las partes. La Figura 2.4 muestra un ejemplo de un modelo esquelético. Puede observarse que cada componente del cuerpo se representa con un segmento de recta.

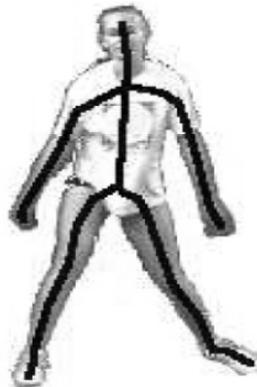


Figura 2.4: Representación utilizando modelos esqueléticos.

### **Silueta del objeto**

La silueta, también llamada gotas, es una máscara de bits densa formada por un solo cuerpo no disjunto que representa al objeto de interés. La Figura 2.5 muestra un ejemplo de esta forma de representación.

Esta manera es importante para algunas aplicaciones, como por ejemplo para métodos de substracción de fondo donde se identifican los objetos que están en primer plano utilizando el cambio que se produce de un cuadro a otro del video.



Figura 2.5: Representación utilizando la silueta del objeto.

### **Contorno**

En esta representación se utiliza el borde o límite de un objeto, tomado como una curva cerrada y continua. Provee un manera de representar un objeto por medio del contorno del mismo. La Figura 2.6 muestra un ejemplo. Esta forma de representación es útil para representar objetos no rígidos.



Figura 2.6: Representación utilizando el contorno del objeto.

## Características del objeto

### Modelado de color

El color es una de las característica más importantes para describir un objeto en una imagen. Sin embargo el color aparente de un objeto es influenciado principalmente por dos factores físicos, la iluminación y la reflexión de la superficie del objeto. Para representar el color hay varios modelos posibles, tales como RGB (Red, Green, Blue) y HSV (Hue, Saturation, Values) entre otros.

### Modelado de forma

La forma de un objeto puede ser representada como un conjunto de puntos de control. Esta forma de modelar el objeto es utilizada para seguir al mismo a partir de sus límites. Además a diferencia del modelado de color es menos sensible a cambios en la iluminación

### Modelado de textura

La textura es una medida de la variación de una superficie, a partir de la ella se definen cualidades como la suavidad o regularidad. Requiere más procesamiento que el color. Las representaciones de texturas se pueden clasificar en dos clases, las estructurales y las estadísticas. Hay varios descriptores de

texturas como GLCM's (Gray-Level Concurrences Matrices) [9], medidas de texturas Law's, Steerable pyramids entre otros.

### **Modelado de movimiento**

La detección de movimiento es una parte esencial para el sistema de visión humano. El flujo óptico representa el movimiento como vectores de desplazamiento que definen el movimiento de cada pixel en una región determinada. Algunos métodos son el de Horn y Schunk [11], el método propuesto por Lucas-Kanade [17], entre otros.

## **2.2. Detección del objeto de interés**

Todo método de seguimiento requiere de algún mecanismo de detección del objeto, ya sea en cada frame o únicamente en el primer frame.

Un enfoque muy utilizado para la detección de objetos es usar solamente la información contenida en cada frame, como si fuera una imagen estática. Aunque también existen métodos de detección que utilizan la información computada en una secuencia de frames. Esta información temporal computada suele resaltar las regiones que sufren cambios en frames consecutivos.

Determinadas las regiones donde se encuentra el objeto en la imagen, la tarea de seguimiento busca encontrar una correspondencia entre frames consecutivos, para poder obtener un seguimiento del mismo y en esta tarea es esencial la detección del objeto.

En esta sección se analizan los métodos utilizados para detectar el objeto.

### **Detectores de puntos de interés o puntos clave**

Los detectores de puntos son utilizados para encontrar pixels de interés dentro de la imagen. Es deseable que estos sean resistentes a cambios de la iluminación, rotación y escala. Los puntos de interés se usan mucho en contextos de detección de movimiento, cámaras stereo y seguimiento.

Algunos métodos son el algoritmo de Harris [10] o el método de SIFT (Scale-Invariant Feature Transform)[16]. La Figura 2.7 muestra puntos cla-

ve o de interés encontrados en una figura, utilizando el método de Harris (Figura 2.7(a)) y el método de SIFT (Figura 2.7(b)).



(a) Método de Harris.



(b) Método SIFT.

Figura 2.7: Detección de puntos clave con los métodos de Harris y SIFT.

## Substracción del fondo

En este caso la detección del objeto puede lograrse construyendo una representación de la imagen a la que llamamos modelo de fondo. A partir del modelo de fondo podemos encontrar desviaciones del fondo en los siguientes cuadros del video. Cualquier cambio significativo en una región de la imagen del modelo de fondo significa que esos pixels corresponden un objeto moviéndose como muestra la Figura 2.8.



(a) Imagen original.



(b) Imagen con substracción del fondo.

Figura 2.8: Método de substracción del fondo.

Sin embargo si el modelo de fondo no fue modelado o actualizado correc-

tamente, el método se hace vulnerable a cambios en la iluminación o cambios estructurales del fondo. Es por esta razón que el algoritmo debe ser robusto ante cambios en el ambiente y capaz de manejar cambios en la iluminación.

Los píxeles de las regiones que sufren los cambios son marcados para ser procesados posteriormente. Es posible utilizar un algoritmo de componentes conectadas para obtener las regiones conectadas que se corresponden con los objetos. Algunos métodos son el propuesto por Wren [27] y Frame differencing [13].

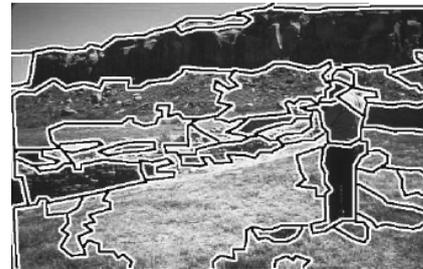
## Clasificación

El objetivo de los algoritmos de clasificación de imágenes, es particionar la imagen en regiones perceptualmente similares. Cada algoritmo de segmentación trata de resolver dos problemas, el criterio por el cual realizar una buena partición y el método para lograr una partición eficiente. Algunos métodos son Mean-Shift Clustering [6], Segmentación de la imagen usando particiones de grafos y Contornos activos [25].

Se puede observar en la Figura 2.9 la partición obtenida por el método de Mean-Shift Clustering.



(a) Imagen original.



(b) Imagen con Mean-Shift Clustering.

Figura 2.9: Método de clasificación.

## Aprendizaje supervisado

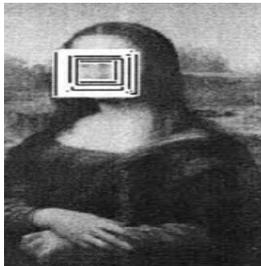
Para la detección del objeto se requiere que se aprendan distintas vistas del objeto automáticamente a partir de un conjunto de ejemplos y por medio de un mecanismo de aprendizaje supervisado. Los mecanismos de aprendizaje

supervisado, basándose en un conjunto de prueba, generan una función que mapea entradas a los resultados deseados.

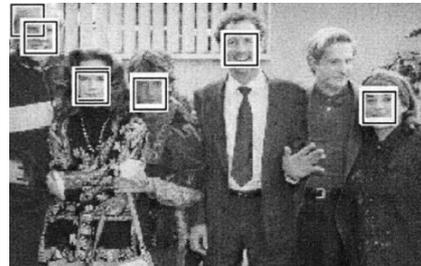
En el contexto de detección de objetos, los ejemplos para el aprendizaje están compuestos por pares de características del objeto y una clase de objeto asociada .

Algunos métodos utilizados son Adaptive boosting [1], Support Vector Machine [7], entre otros.

Se puede observar en la Figura 2.10 la aplicación del método de Support Vector Machine para la detección de objetos incluso en pinturas como la Gioconda.



(a) SVM aplicado a el cuadro de "La Gioconda".



(b) SVM aplicado a imágenes de personas.

Figura 2.10: Método de Support Vector Machine (SVM).

## Remoción de sombras

Luego de la substracción del fondo obtenemos pixels que pueden ser tanto de objetos como de sombras. Esto sucede porque los pixels de sombras son detectados como distintos del fondo. La Figura 2.11 muestra el efecto producido al aplicar el método de remoción de sombra.



(a) Imagen original.



(b) Imagen luego de remoción de sombras.

Figura 2.11: Método de remoción de sombras.

La sombra ocurre cuando hay un objeto entre la fuente de iluminación y la superficie donde está el objeto. Las sombras son de dos tipos, propias (cuando son producidas por el mismo objeto debido a que se observa la parte del objeto en la dirección opuesta a donde proviene la luz) y proyectada (cuando en la superficie del objeto se ve la sombra producida por otro objeto).

### 2.3. Seguimiento del objeto

El objetivo del seguimiento de objetos en videos consiste en generar la trayectoria de un objeto en el tiempo, basándose en la posición del mismo en cada frame del video. A su vez también se puede obtener la región completa que ocupa el objeto en cada instante de tiempo.

La tarea de establecer una relación de correspondencia entre las instancias de los objetos en los distintos frames puede realizarse tanto de manera conjunta como separada.

En el primer caso la región del objeto y su correspondencia son estimadas mediante un proceso que consiste en actualizar iterativamente la posición del objeto y la información de la región obtenida en frames anteriores. Mientras que cuando se realiza de manera separada, las regiones posibles del objeto son obtenidas mediante algún algoritmo de detección del objeto. Luego el algoritmo de seguimiento es el que se encarga de establecer la correspondencia entre los distintos frames.

Independientemente del enfoque elegido, los objetos son representados con

los modelos explicados en las secciones anteriores. El modelo elegido establece restricciones sobre el movimiento o deformación que el objeto puede sufrir durante su seguimiento. Por ejemplo si se elige modelar al objeto mediante la representación del objeto por un punto, no se podrá detectar ninguna deformación del mismo.

En esta sección se analizan los métodos utilizados para seguir el objeto.

## Seguimiento de punto

Los objetos detectados en frames consecutivos son representados por puntos y la asociación de los puntos está basada en el estado del objeto en el frame anterior. El estado del objeto podría incluir la posición y el movimiento.

Este enfoque requiere de un mecanismo externo encargado de realizar la detección del objeto en cada frame.

De esta manera el seguimiento del objeto puede formularse como la correspondencia de los objetos detectados (representados por puntos) en frames consecutivos. Sin embargo la correspondencia de puntos es un problema complicado, especialmente en la presencia de oclusiones, malas detecciones del objeto y frames donde existan muchas salidas y entradas de objetos.

Los métodos de correspondencia de puntos pueden dividirse en dos categorías, los métodos determinísticos y los métodos estadísticos..

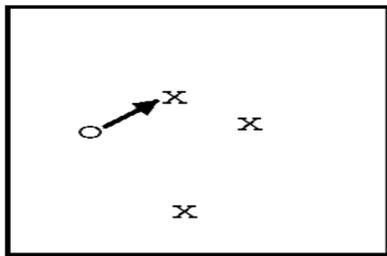
### Métodos determinísticos

Definen un costo para asociar un objeto en un frame  $t-1$  con un único objeto en el frame  $t$ . La minimización del costo del objeto se formula como un problema de optimización combinatoria.

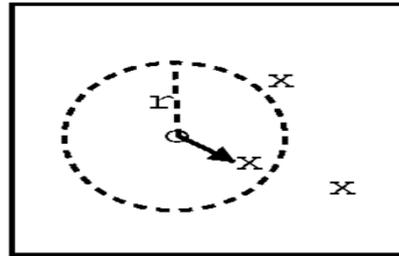
La correspondencia de costos suele definirse usando una combinación de las siguientes restricciones:

- **Proximidad:** El objeto no cambia tanto su posición entre dos frames consecutivos. Como se ve en la Figura 2.12(a), el objeto se asocia al objeto más cercano en el frame siguiente.

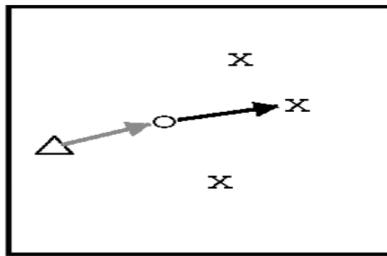
- **Velocidad máxima:** La velocidad del objeto tiene una cota superior. Como se ve en la Figura 2.12(b) el objeto es asociado con el objeto que se encuentre dentro del círculo punteado en el proximo frame. De esta manera la cota de la velocidad queda representada por el radio de dicho círculo.
- **Pequeños cambios de velocidad:** El objeto no cambia drásticamente de velocidad en dos frames consecutivos. Como se ve en la Figura 2.12(c) se asocia al objeto con un objeto en el frame siguiente que posee la velocidad más similar.
- **Rigidez:** La distancia entre puntos dentro de un objeto permanece constante. Como se ve en la Figura 2.12(d) los objetos mantienen constante su distancia.
- **Movimiento común:** Restringe que la velocidad de los objetos dentro de una vecindad pequeña sea similar. Como se ve en la Figura 2.12(e) los objetos poseen velocidades similares. Esta restricción es útil para objetos representados por múltiples puntos.
- **Proximidad uniforme:** Es una combinación de proximidad y pequeños cambios de velocidad.



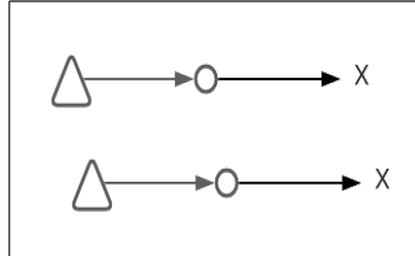
(a) Proximidad.



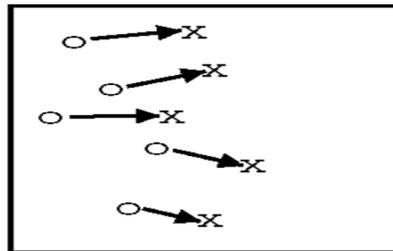
(b) Velocidad máxima.



(c) Pequeños cambios de velocidad.



(d) Rigidez.



(e) Movimiento común.

Figura 2.12: Restricciones para la correspondencia de costos. Las X representan a los objetos en el frame siguiente, los  $\Delta$  representan a los objetos en el frame anterior y los círculos representan al objeto en el frame actual. Las flechas grises representan la velocidad del objeto en el frame anterior mientras que las flechas negras la velocidad del objeto en el frame actual.

Estas restricciones no son específicas de los métodos determinísticos y podrían también ser utilizadas en métodos estadísticos.

Una solución que produzca una correspondencia uno a uno (Figura 2.13(b)) entre todas las posibles asociaciones (Figura 2.13(a)) puede obtenerse a partir de la utilización de métodos de asignación óptima como Hungarian algorithm

[15] o métodos de búsqueda greedy.

En la Figura 2.13 se muestra como se establece una correspondencia uno a uno a partir de todas las asociaciones posibles.

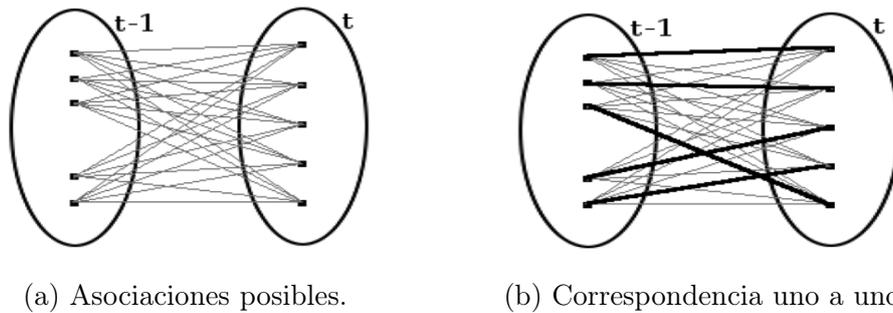


Figura 2.13: Correspondencia de puntos.

### Métodos estadísticos

Como las medidas obtenidas por sensores de video contienen ruido y los objetos en movimiento pueden sufrir perturbaciones aleatorias. Los métodos estadísticos solucionan este problema teniendo en cuenta tanto las medidas como las incertezas del modelo durante la estimación del estado del objeto.

Estos métodos usan como enfoque el espacio estado (*state space*) para modelar las propiedades del objeto como la posición, velocidad y aceleración. Las medidas consisten en la posición del objeto en la imagen, que a su vez es obtenida mediante algún método de detección.

Estos métodos también pueden usarse para estimar el estado de cualquier sistema que varíe con el tiempo.

Si consideramos a un objeto moviéndose en la escena, la información que representa un objeto está definida por la secuencia de estados  $X_t, t \geq 1$  y el cambio en el estado está gobernado por la ecuación:

$$X(t) = X(t - 1) + W(t)$$

donde  $W(t)$  es el ruido blanco en el estado  $t$ .

Luego la relación entre el estado y las medidas es obtenida utilizando  $X(t)$  y  $N(t)$ , donde  $N(t)$  es ruido blanco independiente de  $W(t)$ . El objetivo es la estimación del estado  $X(t)$  dadas todas las medidas hasta el momento,

o lo que es equivalente construir la pdf (Probability Density Function) de  $X(t)$  dadas las medidas.

Para la estimación del estado de un único objeto se suelen usar filtros Kalman [19] o filtros de partículas [24].

### Filtros Kalman

Son usados para estimar el estado de un sistema lineal donde se asume que el estado inicial del objeto y el ruido tiene distribución gaussiana.

El filtrado está compuesto de dos pasos, la predicción (se usa el estado del modelo para predecir el nuevo estado y la covarianza del mismo) y la corrección (utiliza las observaciones actuales para actualizar el estado del objeto).

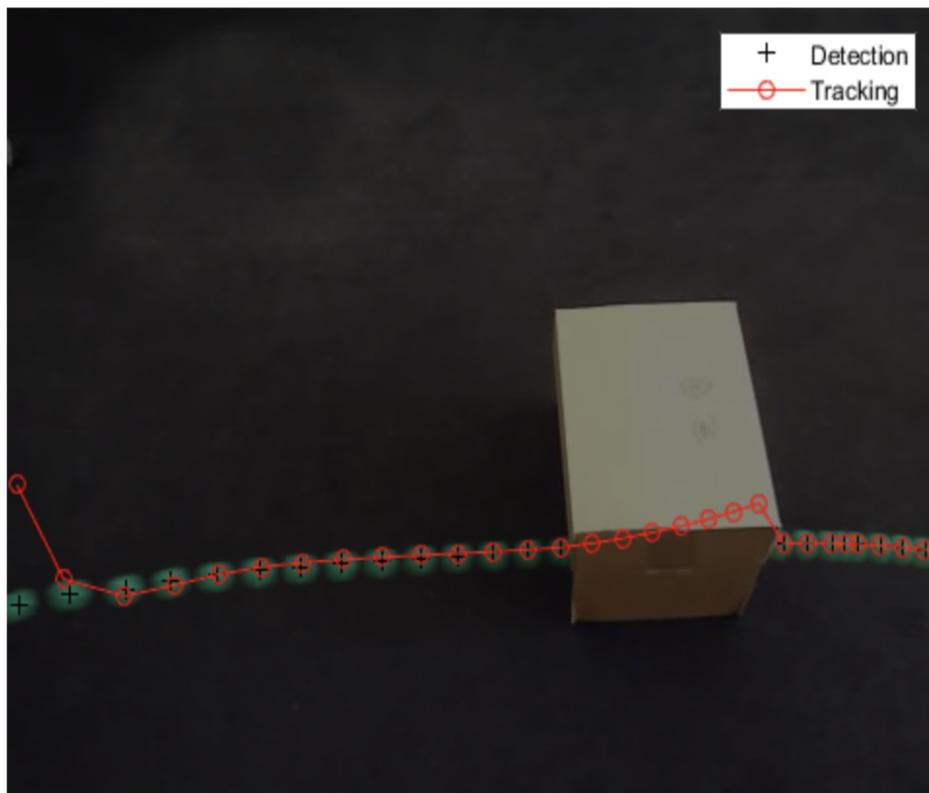


Figura 2.14: Seguimiento con filtros Kalman.

En la Figura 2.14, se muestra la aplicación de filtros de Kalman en el

seguimiento de la trayectoria de una pelota en un video. Los puntos reales de la trayectoria son los señalizados con +, mientras que los detectados son los puntos rojos. Se puede observar como con la corrección se logra mejorar el seguimiento de la pelota hasta que comienza la oclusión de la caja. Luego de la misma vuelve a obtenerse un error en el seguimiento que se soluciona nuevamente con la corrección.

### Filtros de partículas

Cuando el estado inicial o el ruido del sistema no tienen distribución Gaussiana los filtros Kalman no resultan eficientes. En los filtros de partículas la probabilidad condicional del estado en un tiempo  $t$ , se calcula con un conjunto de muestras, donde cada una tiene un peso y el peso representa la importancia de la muestra.

El esquema de muestreo más común es el esquema de muestreo de importancia que consiste en:

1. **Selección:** Se seleccionan  $N$  muestras aleatorias del estado en el tiempo anterior. Para el estado inicial, pueden tomarse las muestras del estado inicial con pesos equiprobables.
2. **Predicción:** Por cada muestra seleccionada  $S_{t-1}^n$  se genera una muestra nueva usando la función  $S_t^n = f(S_{t-1}^n, W_{t-1}^n)$ , donde  $W_{t-1}^n$  es un error gaussiano con media cero y  $f$  es una función no negativa.
3. **Corrección:** Se calculan los pesos correspondientes a las nuevas muestras.

Con las nuevas muestras  $S_t$  se puede calcular la nueva posición del objeto. Además de seguir las mejores partículas, un muestreo adicional es utilizado para remover las muestras con poco peso. Como se vé en la Figura 2.15 se remueven los puntos más lejanos al globo (los de menor peso) y se concentran sobre él.



Figura 2.15: Seguimiento de objetos con filtro de partículas.

En cambio, para múltiples objetos usando filtros Kalman o filtros de partículas, se requiere asociar de manera determinística las medidas con su correspondiente objeto antes de aplicar los filtros.

El método más sencillo para resolver este problema de correspondencia es el de utilizar el enfoque del vecino más cercano. Sin embargo si los objetos están cerca entre sí, es más probable que se produzca una asignación incorrecta, la misma puede causar que el filtro no converja. Existen muchas maneras estadísticas de resolver este problema, las más utilizadas son Joint Probability Data Association Filter o Multiple Hypothesis Tracking (MHT) [23].

## Seguimiento del núcleo

El seguimiento del núcleo es realizado computando el movimiento del objeto, que puede ser representado por el movimiento de una región del objeto de un frame al siguiente. El movimiento del objeto suele ser un movimiento paramétrico (traslación, entre otros).

Dividimos a estos métodos de seguimiento en dos categorías basándonos en la representación de la apariencia utilizada.

## Basados en plantillas y densidad de modelos de apariencia

Son muy usados por su simplicidad y bajo costo computacional. A su vez, éstos se dividen en dos categorías basándose en si el objeto es seguido individualmente o de manera conjunta.

## Modelos de apariencias de múltiples vistas

Estos modelos representan la información del objeto a partir de la observación más reciente. Los objetos pueden aparecer en diferentes vistas y si la vista del objeto cambia drásticamente durante el seguimiento, el modelo de apariencia puede dejar de ser válido y puede perderse el objeto seguido. Para solucionar este problema, diferentes vistas del objeto pueden ser aprendidas de manera offline para luego ser usadas en el seguimiento.

## Seguimiento de la silueta

Los objetos pueden tener formas complejas y muchas veces no pueden ser descritos por simples figuras geométricas. Los métodos basados en siluetas proveen una forma precisa de la descripción de la forma del objeto. El objetivo es encontrar la región del objeto en cada frame a partir de un modelo del objeto obtenida en frames anteriores.

Dividimos a estos métodos de seguimiento en dos categorías:

### Figuras coincidentes

La búsqueda es realizada computando la similaridad del objeto con el modelo generado en el frame anterior. Se asume que la silueta solo se traslada entre frame y frame, por lo que no puede aplicarse al movimiento de objetos no rígidos. El modelo del objeto es actualizado en cada frame que el objeto es localizado, de esta manera permite tolerar cambios en la apariencia del objeto.

### Evolución del contorno

A diferencia del método de figuras coincidentes, los métodos de evolución del contorno hacen evolucionar el contorno inicial del objeto a su nueva posición en el frame actual. Para poder realizar esta evolución del contorno entre frames, se requiere que parte del objeto en el frame actual se superponga con su posición en el frame anterior.

Se realizan desde dos perspectivas: usando modelos del estado del espacio para modelar la figura y movimiento del contorno, o directamente evolucionar

nando el contorno del objeto minimizando la energía del contorno, usando técnicas de minimización directas como el gradiente descendente.

## 2.4. Problemas en el seguimiento de objetos

### Oclusión

La oclusión ocurre cuando parte (oclusión parcial) o la totalidad (oclusión total) del objeto a seguir es escondida por otro objeto. La oclusión parcial es más difícil de detectar, debido a que para ello se debe poder distinguir un cambio en la apariencia del objeto ocluído [8]. Se distinguen tres tipos de oclusión.

#### Oclusión propia

Ocurre cuando una parte del objeto tapa a otra, suele ocurrir cuando el objeto a seguir es articulado. Como se puede ver en la Figura 2.16, el hombre de la imagen tapa parte de su cara con su propia mano.



Figura 2.16: Oclusión propia.

#### Oclusión entre objetos

Ocurre cuando dos objetos seguidos se tapan entre si. Como se ve en la Figura 2.17 el hombre más cercano oculta al más lejano.

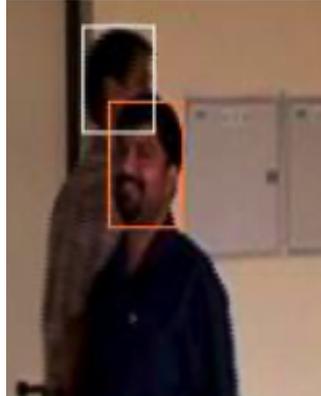


Figura 2.17: Oclusión entre objetos.

### Oclusión con el fondo

Ocurre cuando una estructura del fondo tapa el objeto a seguir. Como se muestra en la Figura 2.18 la pared oculta en parte al hombre que se está siguiendo.

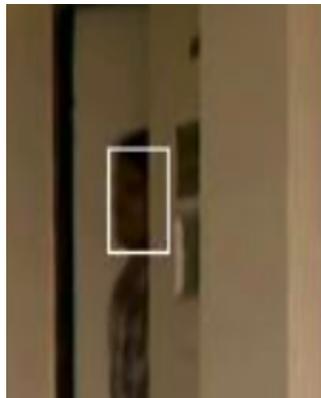


Figura 2.18: Oclusión con el fondo.

### Manejo de la oclusión

Para resolver los problemas de oclusión total se modela el movimiento del objeto a seguir mediante modelos de dinámica lineal o no lineal. De esta manera puede predecirse la posición del objeto hasta que el mismo vuelva a reaparecer.

Otras maneras utilizadas para solucionar el problema de la oclusión total son proyectar la silueta del objeto y flujos ópticos.

## Cambios en la iluminación

Los cambios producidos en la iluminación sobre los objetos son una de las mayores dificultades que se encuentran cuando se estudia el seguimiento de un objeto.

Debido a los cambios de posición de un objeto en relación a una fuente de luz natural o artificial, se produce una gran variación en la iluminación del mismo. Se puede observar en la Figura 2.19 los cambios de iluminación que pueden sufrir los objetos.

Encontrar propiedades que sean independientes de la iluminación y que aparezcan en todas las imágenes de un objeto particular es difícil. Sin embargo podemos utilizar medidas que sirven para casi todas las condiciones (las llamadas casi-invariantes)[26].

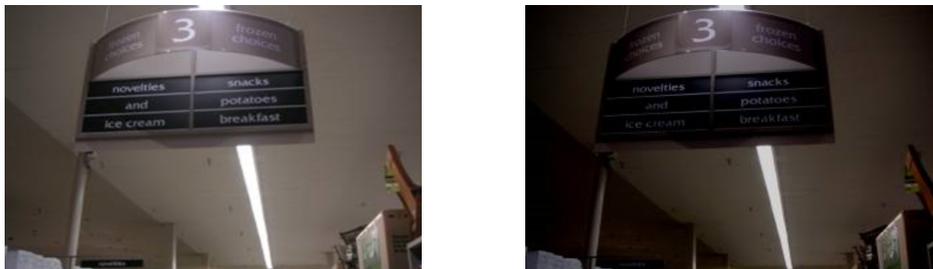


Figura 2.19: Mismo objeto con cambios en la iluminación.

Algunas de estas medidas son la posición del borde, que es insensible a cambios en la iluminación en objetos poliedros (sin superficies suaves) [2], la distribución de la dirección del gradiente de la imagen es un casi-invariante probabilístico [5].

## Ruido

El ruido en imágenes es un problema frecuente que puede ser causado por insuficiente iluminación, cámaras de baja calidad, compresión de la imagen,

entre otros factores [18]. En la Figura 2.20 se ve como el ruido puede afectar una imagen.



(a) Imagen original.



(b) Imagen con ruido Gaussiano.

Figura 2.20: Efecto del ruido sobre una imagen.

Algunos métodos para reducir el ruido en las imágenes son filtros lineales suaves, difusión anisotrópica, Non-local means [3], filtros no lineales, la transformada de Wavelet [20], entre otros.

## 2.5. YOLO: Real-Time Object Detection

Se aplica toda la imagen a una simple red neuronal. La red neuronal divide la imagen en regiones y predice límites con cajas asignando a cada región una probabilidad. Las cajas son pesadas por las probabilidades de la región que contienen. [22].

Este modelo tiene varias ventajas sobre otros sistemas clasificadores. En primer lugar observa la imagen completa al momento de realizar la clasificación por lo que las predicciones usan todo el contexto de la imagen. Las predicciones las hace basandose en una única red neuronal, a diferencia de otros sistemas como R-CNN donde se requieren miles para una única imagen. Por este motivo este método es rápido.

El problema de aplicar este método a un partido de tenis es que únicamente logra detectar la pelota y al jugador que se encuentra más cercano a la cámara mientras que al jugador lejano lo confunde con las personas del fondo. Se puede observar en la Figura 2.21 como se detectan todas las personas presentes en la imagen (las de la tribuna, los jugadores, el árbitro, etc).

El método utiliza un umbral que representa la confianza de que el objeto sea detectado correctamente. Por default utiliza el valor de 0.25 para este umbral. Si se aumenta dicho umbral no logra detectar al jugador lejano, debido a que no lo hace con la suficiente confianza. En la Figura 2.22 se observa que al aumentar el umbral del método, las personas que se encuentran más lejanas a la cámara no son detectadas.

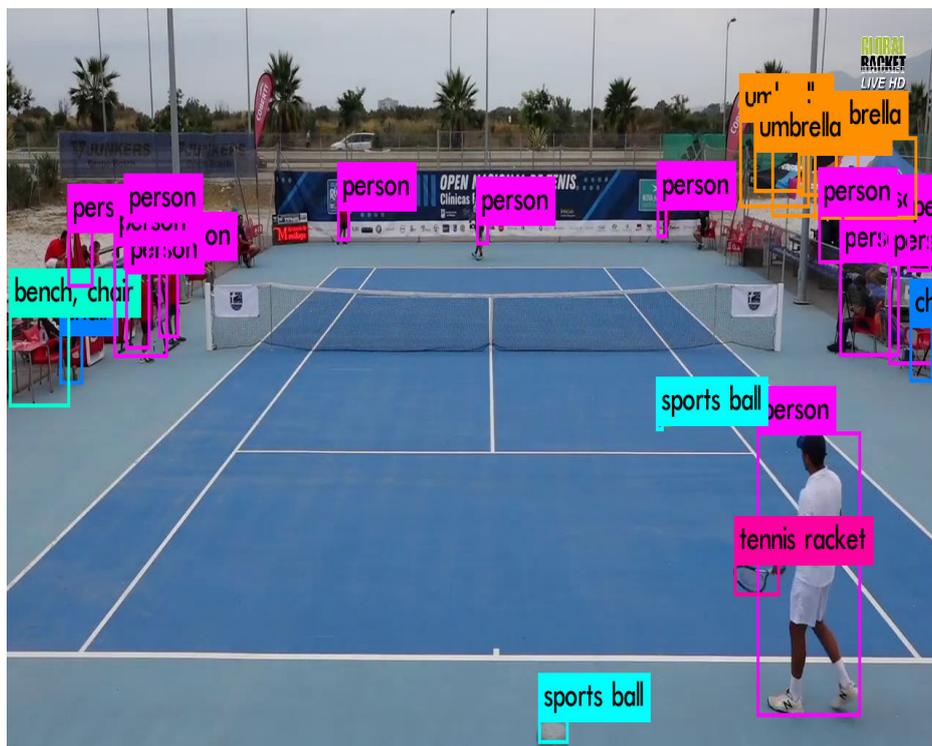


Figura 2.21: Método YOLO aplicado a imagen de un partido de tenis.



Figura 2.22: Método YOLO aplicado a la imagen de un partido de tenis con un umbral más alto.

## Capítulo 3

# Seguimiento de Jugadores en Eventos Deportivos

El objetivo de este capítulo es detallar el procedimiento utilizado para seguir a un jugador en videos de partidos de tenis.

### 3.1. Detección de la cancha

#### Idea

La detección de la cancha consiste en determinar en cada frame del video donde se encuentran las líneas que enmarcan los límites de la cancha. Es decir las líneas que delimitan donde es válido que pique la pelota.

Como la cámara va a estar fija, no es necesario realizar este proceso todo el tiempo. Alcanza con una vez, incluso podría realizarse antes de que comience el partido.

Para detectar la cancha se usó la transformada de Hough [12] al resultado de aplicar el metodo de Canny [4] a la imagen original, ya que es capaz de encontrar formas parametrizables dentro de una imagen, considerando regiones globales de pixels.

## Detector de Bordes de Canny

A continuación se explicará la idea general del detector de bordes de Canny. El método de Canny es utilizado para encontrar bordes en imágenes, es el detector de bordes óptimo de acuerdo a los siguientes criterios:

- **Buena detección:** Implica minimizar la probabilidad tanto de falsos positivos y falsos negativos.
- **Buena Localización:** Los bordes detectados deben estar lo más cerca posible de los bordes reales.
- **Restricción de respuesta simple:** Cada borde debe ser identificado por un único punto.

El detector de bordes de Canny se implementa en distintas etapas:

1. **Suavizamiento y diferenciación:** Primero se suaviza la imagen antes de realizar cualquier detección de bordes. El éxito del método depende del tamaño del filtro utilizado en este paso, ya que en función del mismo, será más o menos sensible al ruido.
2. **Obtener la dirección perpendicular al borde:** En este paso se analizan únicamente las direcciones vertical y horizontal. Generando una nueva imagen binaria con las mismas dimensiones que la original donde el valor de cada pixel representa su magnitud de borde.
3. **Estimación del ángulo perpendicular al borde:** A partir de los resultados obtenidos en el paso anterior, se estima el ángulo de la dirección perpendicular al borde discretizando los posibles valores del ángulo de la siguiente manera  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  y  $135^\circ$
4. **Supresión de no máximos:** Una vez determinada la dirección del borde, se analizan los vecinos del punto que identifica al borde en dicha dirección. Si alguno de sus vecinos tiene una magnitud de borde mayor, se elimina el punto como borde. En esta etapa se busca eliminar los puntos que identifiquen al mismo borde dejando únicamente al de mayor magnitud de borde.

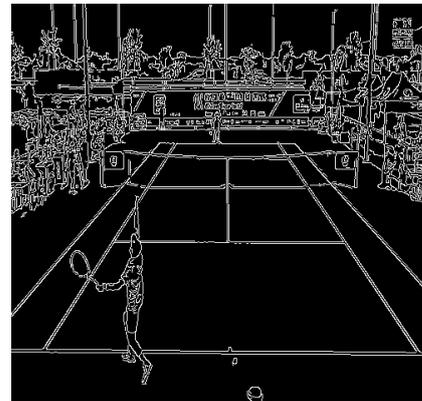
5. **Umbralización con histéresis:** En esta etapa se toman dos umbrales,  $u1$ ,  $u2$  con  $u1 < u2$ . se eliminan los puntos con una magnitud de borde menor a  $u1$  y se mantienen los que poseen una magnitud de borde mayor a  $u2$ . Para los puntos cuya magnitud de borde esta entre  $u1$  y  $u2$  se mantienen únicamente si estan conectados con algun borde. El objetivo de esta etapa es eliminar los bordes falsos.

En esta primera etapa se calculan todos los bordes de cada frame del video y luego este resultado es utilizado por la transformada de Hough.

En la Figura 3.1 se pueden observar todos los bordes obtenidos a partir de la imagen original.



(a) Imagen original.



(b) Bordes obtenidos usando Canny.

Figura 3.1: Aplicación del detector de bordes de Canny sobre una imagen.

## Transformada de Hough

A continuación se explicará la idea del método de la transformada de Hough para detectar líneas.

Dada una imagen binaria (blanco-negro), si consideramos el punto  $(xi, yi)$  blanco, entonces por ese punto pasan infinitas rectas

$$yi = a * xi + b$$

para diversos valores de  $a$  y  $b$ .

Hough propone considerar  $\mathbf{a}$  y  $\mathbf{b}$  como variables y contar la cantidad de puntos  $(\mathbf{x}_i, \mathbf{y}_i)$  que cumplen con la ecuación y luego mediante votación quedarse con las más votadas.

Se subdivide el espacio de parámetros  $(\mathbf{a}, \mathbf{b})$  en lo que se denomina el *array acumulador*. Los parámetros  $a_{min}$ ,  $a_{max}$ ,  $b_{min}$ ,  $b_{max}$  se eligen a priori y son los rangos esperados de la pendiente y la ordenada a origen respectivamente. También se eligen a priori la cantidad de elementos en los que se discretizan estos intervalos.

En la Figura 3.2 se muestra la discretización del espacio en  $\mathbf{a}$  y  $\mathbf{b}$

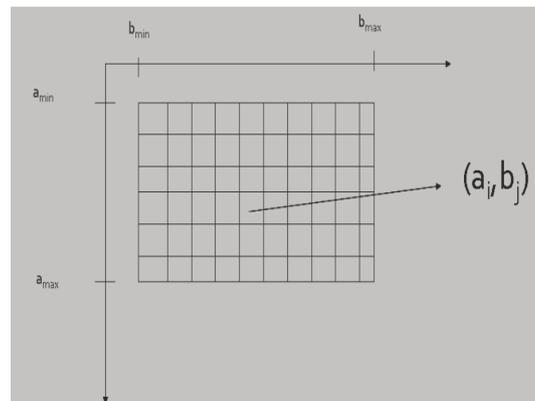


Figura 3.2: Discretización del espacio en  $\mathbf{a}$  y  $\mathbf{b}$ .

### Matriz Acumulador

La matriz acumulador  $\mathbf{A}$  tiene la misma dimensión que la que se utilizó en la discretización del espacio. De esta manera la celda  $\mathbf{A}(i, j)$  se corresponde a la coordenada del espacio con parámetros  $(a_i, b_j)$ . La matriz acumulador  $\mathbf{A}$  comienza con el valor de  $\mathbf{0}$  para todas sus celdas.

### Algoritmo

Para cada elemento  $(a_i, b_j)$  y para cada pixel  $(x_k, y_k)$  blanco de la imagen se busca verificar si se cumple la siguiente propiedad:

$$|b_j - x_k \cdot a_i - y_k| < \epsilon$$

Si se cumple la propiedad, entonces:

$$A(i, j) = A(i, j) + 1$$

De esta manera si al final del ciclo  $\mathbf{A}(i, j) = c$ , significa que existen  $c$  pixels de la imagen que satisfacen la ecuación de la recta con pendiente  $\mathbf{ai}$  y ordenada al origen  $\mathbf{bi}$ .

El último paso consiste en buscar los máximos y quedarnos con las rectas correspondientes a esas ecuaciones.

## Implementación

Para la implementación se usaron las funciones provistas por *openCv* *HoughLines* y *HoughLinesP*

Estas funciones, a diferencia del algoritmo descrito, que usa como parámetros a la pendiente  $\mathbf{a}$  y la ordenada a origen  $\mathbf{b}$ , se representa a las rectas utilizando coordenadas polares (Ver Figura 3.3).

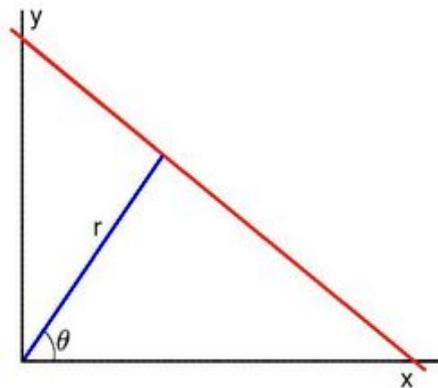


Figura 3.3: Representación de la recta usando coordenadas polares.

De esta manera la ecuación de la recta se expresa mediante la ecuación:

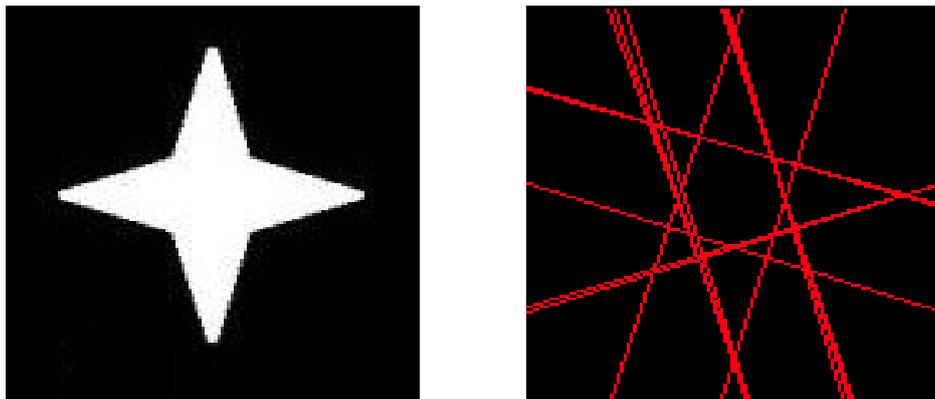
$$y = \left(-\frac{\cos(\theta)}{\sin(\theta)}\right) * x + \left(\frac{r}{\sin(\theta)}\right)$$

Primero veremos los parámetros utilizados para *HoughLines*:

```
1 cv.HoughLines(imagen, lineas, rho, theta, umbral)
```

- *imagen*: Imagen de 8-bits binaria.
- *líneas*: Bordes obtenidos por algún detector de bordes como *Canny*
- *rho*: La discretización de distancia del acumulador medida en pixels.
- *theta*: La discretización del ángulo del acumulador medida en radianes.
- *umbral*: Mínima cantidad de votos que debe poseer una línea para ser devuelta.

En la Figura 3.4 se puede observar el resultado de aplicar *HoughLines* a una imagen.



(a) Imagen original.

(b) Líneas obtenidas de HoughLines.

Figura 3.4: Aplicación de HoughLines sobre una imagen.

La otra función es *HoughLinesP* que es una optimización de *HoughLines*, no toma todos los puntos en consideración. Solo toma un subconjunto aleatorio de los mismos que es suficiente para su detección, requiriendo una disminución del threshold. También como se puede ver en la Figura 3.5 las líneas encontradas por la función tienen un comienzo y un final.

```

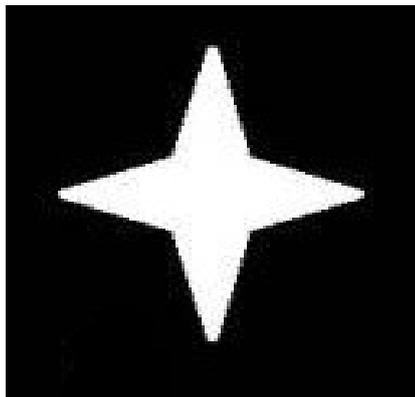
1         cv.HoughLinesP (imagen, lineas, rho, theta, umbral,
2                           minLongitudLinea, maxLEspacioLinea)

```

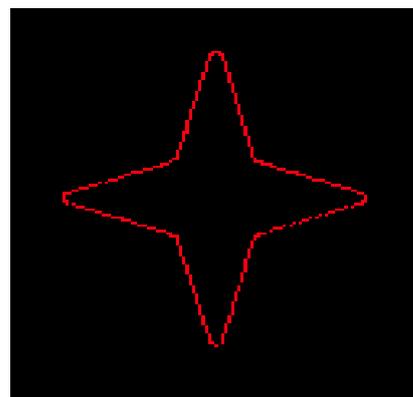
- *imagen*: Imagen de 8-bits binaria.

- *líneas*: Bordes obtenidos por algún detector de bordes como *Canny*
- *rho*: La discretización de distancia del acumulador medida en pixels.
- *theta*: La discretización del ángulo del acumulador medida en radianes.
- *umbral*: Mínima cantidad de votos que debe poseer una línea para ser devuelta.
- *minLongitudLinea*: Mínima longitud de línea a detectar. Desecha todos los segmentos de longitud menor a la especificada por este parámetro.
- *maxLEspacioLinea*: Máximo espacio permitido entre puntos de una línea para ser considerados parte del mismo segmento.

En la Figura 3.5 se puede observar el resultado de aplicar *HoughLinesP* a una imagen.



(a) Imagen original.



(b) Líneas obtenidas de *HoughLinesP*.

Figura 3.5: Aplicación de *HoughLinesP* sobre una imagen.

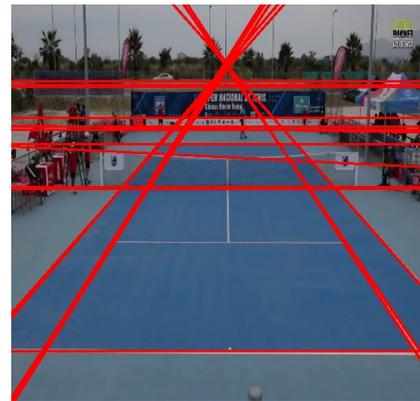
A continuación veremos los resultados obtenidos de aplicar las funciones de *HoughLines* (Ver Figura 3.7) y *HoughLinesP* (Ver Figura 3.8) a una imagen de una cancha de tenis (Ver Figura 3.6).



Figura 3.6: Imagen original de la cancha de tenis utilizada para detectar las líneas.

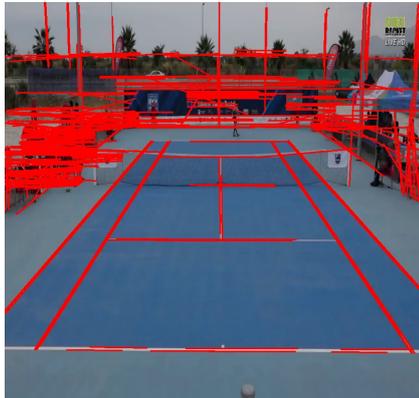


(a) Aplicación de HoughLines con umbral bajo.



(b) Aplicación de HoughLines con umbral alto.

Figura 3.7: Aplicación de HoughLines usando distintos umbrales.



(a) Aplicación de HoughLinesP con umbral bajo.



(b) Aplicación de HoughLinesP con umbral alto.

Figura 3.8: Aplicación de HoughLinesP usando distintos umbrales.

## Problemas

A pesar de que el método de la transformada de Hough encuentra las líneas de la cancha, también encuentra otras líneas que no nos son de interés.

Como se puede ver en la Figura 3.7 y en la Figura 3.8 además de las líneas de la cancha se detectan otras líneas provenientes del fondo que queremos ignorar.

A partir de la detección inicial, afinando las líneas e ignorando las del fondo obtenemos las líneas que se observan en la Figura 3.9



Figura 3.9: Líneas de la cancha ajustadas.

## 3.2. Seguimiento de la pelota

### Idea

El seguimiento de la pelota consiste en poder localizar la posición de la pelota en cada frame del video. Para poder realizar estadísticas basadas en los movimientos de la misma.

### Método de Intercambio de Pixels

Para el seguimiento de la pelota se utiliza el método de Intercambio de Pixels [25] que es una mejora del método de Contornos Activos. Se utiliza este método por que permite a partir de una curva inicial (dada por el usuario y contenida en el objeto a seguir) ir modificándola hasta que el borde de la misma se ajuste al del objeto de interés.

El metodo de Intercambio de Pixels se utiliza en seguimiento de videos ya que permite reducir el costo computacional y aplicarse a imágenes con mucho ruido.

### Algoritmo

Se parte de considerar la imagen como si estuviera formada únicamente por dos elementos (el objeto de interés y el fondo).

Bajo esa consideración se parte de la curva inicial aportada por el usuario, debe ser una curva cerrada contenida en el objeto de interés.

A partir de esa curva se obtiene el contorno de la misma separada en dos listas de pixels: *Lin* (el contorno interno de la curva) y *Lout* (contorno externo de la curva). De esta manera quedan determinadas tres regiones en la imagen:

- **El objeto:** Conformado por el conjunto de pixels encerrados por *Lin*.
- **El borde del objeto:** Conformado por el conjunto de pixels que son parte de *Lin* o de *Lout*.
- **El fondo:** Conformado por el conjunto de los pixels restantes.

En la Figura 3.10 se observan las tres regiones de la imagen.

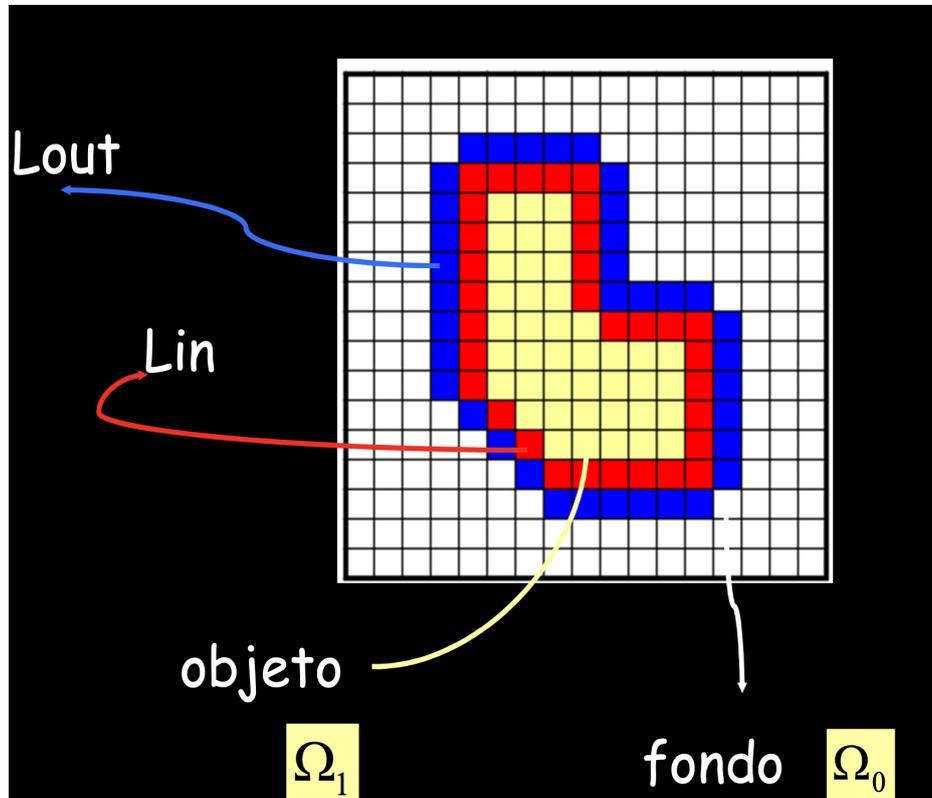


Figura 3.10: Separación de la imagen en las tres regiones a partir de la curva inicial.

Se toma una matriz  $\phi$  de las mismas dimensiones de la imagen que posee los valores descriptos en la ecuación 3.1

$$\phi(x) = \begin{cases} 3 & \text{si } x \in \text{fondo} \\ 1 & \text{si } x \in L_{out} \\ -1 & \text{si } x \in L_{in} \\ -3 & \text{si } x \in \text{objeto} \end{cases} \quad (3.1)$$

Como se puede inferir de la ecuación 3.1  $L_{in}$  representa los puntos interiores del objeto que tienen por lo menos un vecino exterior y  $L_{out}$  representa los puntos exteriores al objeto que tienen por lo menos un vecino interior.

Como criterio de vecindad se utiliza la *vecindad de von Neumann*, también conocida como *4-vecinos*. Este criterio de vecindad considera como

vecinos a los pixels que se encuentren conectados directamente ya sea en la dirección vertical o en la dirección horizontal.

Se elije una funcion  $Fd$  para determinar si un pixel pertenece o no al objeto, donde  $Fd$  cumple con la ecuación 3.2.

$$Fd(x) = \begin{cases} x \in \text{objeto} & \text{si } Fd(x) \geq 0 \\ x \notin \text{objeto} & \text{si } Fd(x) < 0 \end{cases} \quad (3.2)$$

En nuestro caso se usó como  $Fd$  una función basada en el color del objeto, sin embargo podría usarse una función que se base en otras características como aplicar un banco de filtros que modelen texturas, señales visuales, etc.

### Pasos del algoritmo

1. Indicar la región inicial con un rectángulo dentro del objeto de interés, definir  $L_{in}$  y  $L_{out}$  y calcular el color promedio del objeto.
2. Para cada  $x$  en  $L_{out}$ , si  $Fd(x) \geq 0$ , borrar  $x$  de  $L_{out}$  y agregarlo a  $L_{in}$  y agregar a  $L_{out}$  los vecinos de  $x$  que sean parte del fondo.
3. Se eliminan los pixels de  $L_{in}$  que se transformaron en parte del objeto (y se los marca como parte del mismo), es decir los pixels en  $L_{in}$  que no tienen ningun vecino  $L_{out}$ .
4. Para cada  $x$  en  $L_{in}$ , si  $Fd(x) < 0$ , borrar  $x$  de  $L_{in}$  y agregarlo a  $L_{out}$ , agregar también a  $L_{in}$  los vecinos de  $x$  que sean parte del objeto.
5. Se eliminan los pixels de  $L_{out}$  que se transformaron en parte del fondo(y se los marca como parte del mismo), es decir los pixels en  $L_{out}$  que no tienen ningún vecino  $L_{in}$ .

Los últimos pasos del algoritmo se aplican hasta que no haya cambios o se alcance el número de iteraciones especificadas.

El algoritmo descrito se aplica a una imagen. Para extenderlo a un video, primero se descompone al video en frames y se aplica el algoritmo al primer frame del video. A partir del segundo frame, se saltea el paso 1 del algoritmo, y se utiliza como  $L_{in}$  y  $L_{out}$  los resultantes de aplicar el algoritmo al frame anterior.

## Corrección del algoritmo

El éxito de la aplicación del algoritmo a videos se basa en la idea de que el objeto de interés no varíe demasiado su posición ni su color en dos frames consecutivos, ya que si no hay intersección entre los bordes del objeto de interés en dos frames consecutivos se pierde el rastro del mismo.

En un partido de tenis, la pelota se mueve a mucha velocidad y por esta razón sucede que para algunos frames consecutivos los bordes de la pelota son disjuntos. Para solucionar este problema se busca a la pelota en un entorno de cincuenta pixels con respecto a los bordes de la misma ( $L_{in}$  y  $L_{out}$ ) en el frame anterior.

Como se puede observar en la Figura 3.11 y la Figura 3.12 utilizando el método de intercambio de pixels, se logra seguir la pelota correctamente entre dos frames consecutivos.



Figura 3.11: Posición de la pelota en el frame inicial.



Figura 3.12: Posición de la pelota en el siguiente frame.

### 3.3. Detección de cambios de movimiento de la pelota

Para poder detectar el movimiento de la pelota, es necesario conocer la posición de la pelota en mas de un frame consecutivo.

#### Cálculo de la velocidad de la pelota

Para obtener la velocidad de la pelota en un frame dado, partimos del supuesto que conocemos la posición de la pelota tanto en el frame anterior como en el actual.

Primero se calcula la posición del centro de masa de la pelota en un frame utilizando la siguiente fórmula:

$$r\vec{c}m = \frac{\sum_{x \in (Lin \cup Lout)} x}{|Lin \cup Lout|}$$

Luego a partir de la posición del centro de masa de la pelota en dos frames consecutivos, podemos hallar la dirección de la velocidad de la pelota. Como solo nos importa la dirección lo calculamos de manera unitaria utilizando la ecuación:

$$velocidad_{\vec{v}} = \frac{r\vec{c}m_f - r\vec{c}m_i}{|r\vec{c}m_f - r\vec{c}m_i|}$$

De esta manera podemos calcular la velocidad de la pelota en cada frame, a excepción del frame en que se pone en movimiento.

En las Figura 3.13 y Figura 3.14 se puede observar la velocidad de la pelota en dos frames consecutivos.



Figura 3.13: Velocidad de la pelota en un frame.



Figura 3.14: Velocidad de la pelota en el frame siguiente.

Los cambios de la velocidad en la pelota pueden darse por dos razones:

- Pique de la pelota contra el piso, red o algún elemento del fondo.
- Golpe de un jugador sobre la pelota.

Es importante distinguir estos casos para poder interpretar correctamente las acciones del juego.

Primero hay que detectar cuando se produce un cambio de velocidad importante sobre la pelota. Se asume como un cambio de velocidad importante cuando la dirección de la pelota cambia de sentido y para lograr eso se usó la siguiente fórmula:

$$proyeccion = velocidad_{n-1} \cdot velocidad_n$$

Si la proyección es negativa significa que se produjo un cambio en el sentido de la velocidad de la pelota.

## Golpe sobre la pelota

Para que el cambio de velocidad de la pelota sea detectado como un golpe de un jugador deben cumplirse las siguientes dos condiciones:

- La proyección de la velocidad final de la pelota sobre el vector que va del jugador que se esta siguiendo al rival debe ser positiva.
- El jugador debe estar en un entorno cercano a la pelota.

Para poder realizar estos cálculos también es necesario seguir la posición del jugador y para ello aplicamos la misma técnica que se realizó con la pelota. Como se observa en la Figura 3.15 se sigue tanto a la pelota como al jugador más cercano a la cámara.



Figura 3.15: Seguimiento de la pelota con su velocidad y del jugador.

## Pique de la pelota

Si la velocidad de la pelota cambió su sentido y no cumple las condiciones para ser un golpe del jugador, se la considera como un pique. Queda por determinar entonces si el pique de la pelota fue un pique válido o inválido y para ello se utiliza una matriz binaria (unos y ceros)  $\mathbf{M}$ .

La matriz binaria es del mismo tamaño que la imagen que se está analizando y se inicializa utilizando como información los límites detectados de la cancha. Como la matriz es de las mismas dimensiones que la imagen se corresponde la celda  $\mathbf{M}[i, j]$  con el pixel  $(i, j)$  de la imagen (Ecuación 3.3).

$$M[i, j] = \begin{cases} 0 & \text{si } (i, j) \in \text{cancha} \\ 1 & \text{si } (i, j) \notin \text{cancha} \end{cases} \quad (3.3)$$

## 3.4. Análisis del rendimiento del jugador en el video

Para analizar el rendimiento del jugador en el video, primero se selecciona un fragmento del video y luego con el programa se enumeran las acciones detectadas por el mismo y se las compara con las acciones realmente ejecutadas.

Para poder llevar a cabo este proceso, primero es necesario tener un rango de frames que contengan acciones (pique de pelota, pegada de jugador, golpe a la red, entre otros).

Una vez obtenido ese intervalo se analizan dichos frames con el programa y se comparan los resultados obtenidos por el programa con los que se ven en el video.

# Capítulo 4

## Resultados

En este capítulo se expondrán las acciones detectadas en el video mediante la ejecución del programa y su contrastación con la acción real del mismo.

Para obtener los resultados se analizaron diversos fragmentos del video que contenían una acción particular, como pueden ser la del pique de la pelota dentro de la cancha, el golpe del jugador, entre otras.

En la Figura 4.1 puede observarse el seguimiento de la pelota, mientras la misma pica dentro de la cancha y es detectada correctamente.



(a) Imagen 1.



(b) Imagen 2.

Figura 4.1: Imagen 1 y 2 del seguimiento de un pique de la pelota dentro de la cancha detectado.



(c) Imagen 3.



(d) Imagen 4.

Figura 4.1: Imagen 3 y 4 del seguimiento de un pique de la pelota dentro de la cancha detectado.



(e) Imagen 5.



(f) Imagen 6.

Figura 4.1: Imagen 5 y 6 del seguimiento de un pique de la pelota dentro de la cancha detectado.



(g) Imagen 7.



(h) Imagen 8.

Figura 4.1: Imagen 7 y 8 del seguimiento de un pique de la pelota dentro de la cancha detectado.



(i) Imagen 9.



(j) Imagen 10.

Figura 4.1: Imagen 9 y 10 del seguimiento de un pique de la pelota dentro de la cancha detectado.



(k) Imagen 11.



(l) Imagen 12.

Figura 4.1: Imagen 11 y 12 del seguimiento de un pique de la pelota dentro de la cancha detectado.



(m) Imagen 13.



(n) Imagen 14.

Figura 4.1: Imagen 13 y 14 del seguimiento de un pique de la pelota dentro de la cancha detectado.



(ñ) Imagen 15.

Figura 4.1: Imagen 15 del seguimiento de un pique de la pelota dentro de la cancha detectado.

En la Figura 4.2 puede observarse el seguimiento de la pelota, mientras la misma es golpeada por el jugador y dicha acción detectada correctamente.



(a) Imagen 1.



(b) Imagen 2.

Figura 4.2: Imagen 1 y 2 del seguimiento de un golpe a la pelota detectado.



(c) Imagen 3.



(d) Imagen 4.

Figura 4.2: Imagen 3 y 4 del seguimiento de un golpe a la pelota detectado.



(e) Imagen 5.



(f) Imagen 6.

Figura 4.2: Imagen 5 y 6 del seguimiento de un golpe a la pelota detectado.



(g) Imagen 7.



(h) Imagen 8.

Figura 4.2: Imagen 7 y 8 del seguimiento de un golpe a la pelota detectado.



(i) Imagen 9.



(j) Imagen 10.

Figura 4.2: Imagen 9 y 10 del seguimiento de un golpe a la pelota detectado.



(k) Imagen 11.



(l) Imagen 12.

Figura 4.2: Imagen 11 y 12 del seguimiento de un golpe a la pelota detectado.



(m) Imagen 13.



(n) Imagen 14.

Figura 4.2: Imagen 13 y 14 del seguimiento de un golpe a la pelota detectado.



(ñ) Imagen 15.



(o) Imagen 16.

Figura 4.2: Imagen 15 y 16 del seguimiento de un golpe a la pelota detectado.



(p) Imagen 17.



(q) Imagen 18.

Figura 4.2: Imagen 17 y 18 del seguimiento de un golpe a la pelota detectado.



(r) Imagen 19.



(s) Imagen 20.

Figura 4.2: Imagen 19 y 20 del seguimiento de un golpe a la pelota detectado.



(t) Imagen 21.



(u) Imagen 22.

Figura 4.2: Imagen 21 y 22 del seguimiento de un golpe a la pelota detectado.



(v) Imagen 23.



(w) Imagen 24.

Figura 4.2: Imagen 23 y 24 del seguimiento de un golpe a la pelota detectado.



(x) Imagen 25.

Figura 4.2: Imagen 25 del seguimiento de un golpe a la pelota detectado.

En la Tabla 4.1 se exponen los resultados luego de analizar las partes del video que contenían las diferentes acciones que se esperaban detectar.

Acción	Detectada
Pique de pelota dentro de la cancha	Si
Golpe del jugador	Si
Pelota golpea la red	No
Punto por doble pique	No
Saque del jugador	No

Tabla 4.1: Resultados del programa sobre la detección de las acciones que afectan la pelota.

# Capítulo 5

## Conclusiones

En este último capítulo se expondrán las conclusiones del trabajo realizado. A semejanza de un resumen final, se hará una recopilación del trabajo y la investigación llevada a cabo durante la realización del proyecto y se revisarán los resultados globales obtenidos tras las pruebas del capítulo anterior.

En la primer fase del proyecto nos planteamos el objetivo de diseñar un sistema de seguimiento de objetos en video, a efectos de evaluar el rendimiento de un jugador de tenis. El estudio se realizó con las limitaciones de requerir de una cámara fija en uno de los lados de la cancha, desde la cual se aprecie en perspectiva la totalidad de la misma. Este requerimiento a su vez trajo el inconveniente de que la pelota sea difícil de seguir cuando la misma se posicione lejos de la cámara, debido a su pequeño tamaño, los cambios de iluminación y la oclusión de la red o el jugador.

La dificultad de poder seguir a la pelota durante varios frames consecutivos, impidió poder realizar un análisis automático del rendimiento. En cambio el proceso debe realizarse de manera manual, una vez que se pierde la pelota, volver a comenzar el proceso desde el frame en el cual se perdió la misma.

En cuanto a los resultados obtenidos por el trabajo, dada la dificultad del problema, pensamos que podría disminuirse el error agregando herramientas de aprendizaje automático supervisado para el seguimiento de varios objetos al mismo tiempo.

También sería posible analizar videos que tuvieran más cámaras, apor-

tando de esta manera nuevas perspectivas para el movimiento de la pelota en la cancha aunque también agregaría dificultad al problema y costo computacional.

La dificultad del problema es muy grande debido al reducido tamaño de la pelota y su gran velocidad. Como trabajos futuros, tenemos la idea de aplicar métodos de aprendizaje automático para la detección de la pelota en el video.

# Bibliografía

- [1] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995.
- [2] Peter N Belhumeur and David J Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):245–260, 1998.
- [3] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Non-local means denoising. *Image Processing On Line*, 1:208–212, 2011.
- [4] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- [5] Hansen F Chen, Peter N Belhumeur, and David W Jacobs. In search of illumination invariants. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 254–261. IEEE, 2000.
- [6] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.
- [7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

- [8] Pierre F Gabriel, Jacques G Verly, Justus H Piater, and André Genon. The state of the art in multiple object tracking under occlusion in video sequences. In *Advanced Concepts for Intelligent Vision Systems*, pages 166–173, 2003.
- [9] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.
- [10] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.
- [11] BKP Horn. Schunk. b. g. *Determining optical flow*, 1981.
- [12] Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.
- [13] Ramesh Jain and H-H Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):206–214, 1979.
- [14] Anand Jalal and Vrijendra Singh. The state-of-the-art in visual object tracking. *Informatika (Slovenia)*, 36:227–248, 01 2012.
- [15] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [16] Tony Lindeberg. Scale invariant feature transform. 2012.
- [17] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.
- [18] S Milyaev and I Laptev. Towards reliable object detection in noisy images. *Pattern Recognition and Image Analysis*, 27(4):713–722, 2017.
- [19] Anurag Mittal and Larry S Davis. M 2 tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

- [20] Quan Pan, Lei Zhang, Guanzhong Dai, and Hongai Zhang. Two denoising methods by wavelet transform. *IEEE transactions on signal processing*, 47(12):3401–3406, 1999.
- [21] Kalpesh R Ranipa and Kiritkumar Bhatt. Illumination condition effect on object tracking: A review. *Global Journal of Computer Science and Technology*, 2014.
- [22] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.
- [23] Donald Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979.
- [24] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2003.
- [25] Yonggang Shi and William Clement Karl. Real-time tracking using level sets. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 34–41. IEEE, 2005.
- [26] Rufin Vogels and Irving Biederman. Effects of Illumination Intensity and Direction on Object Coding in Macaque Inferior Temporal Cortex. *Cerebral Cortex*, 12(7):756–766, 07 2002.
- [27] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Paul Pentland. Pfnder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):780–785, 1997.
- [28] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: a survey. *acm comput surv. ACM Comput. Surv.*, 38, 12 2006.