

Universidad de La Habana  
Facultad de Matemática y Computación



# Detección de la Zona de Strike

Autores:

**Aldo Javier Verdesia Delgado  
Darian Ramón Mederos**

Tutor:

**José Alejandro Mesejo Chiong**

Trabajo de Diploma  
presentado en opción al título de  
Licenciados en Ciencia de la Computación

Fecha

Noviembre de 2022

Para ti; del niño más lindo del mundo

**Aldo Javier Verdesia Delgado**

Se lo dedico a mis abuelos, a mi madre, a mi hermano, a mi novia por apoyarme siempre y en especial a mi padre que sé que me apoya donde quiera que esté.

**Darian Ramón Mederos**

# Agradecimientos

Agredecer a toda aquella persona que en su momento formó parte de estos años apoyándome y alentándome a no rendirme. A todo mi grupo 12 y allegados por siempre aportarme sus ayudas y formar parte de esta etapa. Un agradecimiento especial a mis tres amigos y compañeros de trabajos, que fueron parte de mi desarrollo. También agradecer a todos mis familiares, en especial a mis padres por estar ahí para mí en todo momento y a mi novia por el apoyo, la ayuda y la motivación para que este sueño fuera posible.

**Aldo Javier Verdesia Delgado**

Le agradezco a todas las personas que de una forma u otra me ayudaron, no solo en el desarrollo de la tesis sino a lo largo de estos años en la carrera. Le quiero agradecer, a mi familia por su apoyo inalcanzable a pesar de todas las dificultades y a mi novia por siempre apoyarme y ayudarme en los malos momentos.

**Darian Ramón Mederos**

De parte de ambos le agradecemos profundamente a nuestro tutor el profesor José Alejandro Mesejo, por su dedicación al desarrollo de la tesis y su consejo siempre oportuno. Además a los profesores que nos han impartido conocimientos en estos años y que han dejado su huella en nosotros a todos muchas gracias.

**Aldo Javier Verdesia Delgado y Darian Ramón Mederos**

# Opinión del tutor

En declaraciones al periodista Oscar Sánchez Serra, publicadas en el diario Granma el 9 de marzo de este año en el artículo titulado “*Por la zona de strike pasa el espectáculo y la calidad de la pelota cubana*”, Luis Daniel del Risco, en aquel entonces jefe de Regla y Arbitraje de la Comisión Nacional de Béisbol aseguraba «Un árbitro, en home, decide entre 190 y 200 lanzamientos, aproximadamente, y el margen de error es de cinco a siete envíos para que sea un buen trabajo, siempre que estos errores, o error, no incidan en el resultado del juego, pues entonces no se consideraría una buena labor».

Estas declaraciones de del Risco ponen de relieve la importancia del trabajo arbitral en nuestro pasatiempo nacional y la necesidad de proporcionar a los árbitros y la audiencia televisiva de una herramienta para ayudar a los primeros a perfeccionar su labor y a los segundos a hacer apreciaciones menos parcializadas. Como requisito fundamental para la herramienta se estableció el de “trabajar con lo que hay” dadas las limitaciones tecnológicas que padecemos. Otras ligas de béisbol, como la MLB, disponen de varias cámaras de alta velocidad y hasta radares para determinar la trayectoria de la bola de béisbol en el campo de juego.

Con el desafío de disponer solamente de “la cajita” y lograr algo al menos similar a lo que presentan las trasmisiones de la MLB acometieron Aldo Javier Verdesia Delgado y Darian Ramón Mederos su Trabajo de Diploma. Para ello asimilaron técnicas del procesamiento de imágenes y de la visión computacional que demuestran la capacidad de ambos para extraer conocimientos adquiridos en el transcurso de la carrera. Esto lo lograron bajo mínima supervisión, en los plazos requeridos y mostrando, además de constancia, profundidad en el aprendizaje. Son estas las características necesarias que deben mostrar los egresados de Ciencia de la Computación de nuestra Facultad y que los destacan donde quiera que desempeñen su futura labor profesional.

Los resultados obtenidos por Aldo Javier Verdesia Delgado y Darian Ramón Mederos son excelentes. Teniendo en cuenta que la TV cubana aún no logra transmitir todos los partidos de béisbol en alta definición, que las cámaras que enfocan el home plate no siempre están a una altura sobre el terreno óptima y que la base home muchas veces dista del color blanco el por ciento de éxito del software logrado es excepcional. Considero que ambos, también jugadores del equipo de béisbol de la Facultad que este

año lograron el segundo lugar en los Juegos Caribes, merecen la máxima calificación en este su ejercicio de culminación de estudios.

Dr. José Alejandro Mesejo Chiong  
La Habana, 24 de Noviembre de 2022.

# Resumen

El béisbol es considerado el deporte nacional de Cuba, apasiona a muchos y deja indiferentes a pocos. Las transmisiones de los eventos realizados de este deporte son recibidas con gran agrado por el público y con el pasar de los años estas han ido en aumento.

En el presente documento se propone la realización de un software que enriquezca las transmisiones televisivas, permitiendo mostrarle al espectador la trayectoria de un lanzamiento y la zona de Strike del bateador. El trabajo describe técnicas clásicas de procesamiento de imágenes como son la sustracción de fondo y la eliminación de ruido para el mejorado de imágenes. Se aborda el tema de la detección de objetos en movimiento a alta velocidad y se describen las trayectorias que estos realizan analizándolas en los diferentes ejes. También se presentan algoritmos de predicción y elección de trayectorias.

Se propone también un algoritmo para determinar la zona de Strike que depende de varios procesos como filtrado por intensidad, detección del bateador mediante detección de movimiento, detección de segmentos candidatos a home en una imagen binaria, detección de la pelota de béisbol mediante detección de círculos, búsqueda de contornos en imágenes binarias , suavizado de funciones mediante filtro de suavizado.

El documento refleja una serie de experimentos que analizan la fiabilidad del software y los errores más comunes que aparecen en la predicción tanto de la trayectoria como en la obtención de las líneas que limitan la zona Strike.

# Abstract

Baseball is considered the national sport of Cuba, it excites many and leaves few indifferent. The transmissions of the events carried out in this sport are received with great pleasure by the public and over the years these have been increasing. This document proposes the creation of software that enriches television broadcasts by showing the viewer the trajectory of a pitched baseball and the batter's strike zone. The work describes classical image processing techniques such as background subtraction and noise removal for image enhancement. The subject of movement detection of objects at high speed is addressed and the trajectories that they describe, analyzing them in the different axes, are computed. Trajectory prediction algorithms are also presented, as well as methods for choosing them.

An algorithm to determine the strike zone is also proposed. This algorithm depends on several processes such as intensity filtering, hitter detection by motion detection, detection of home plate candidate segments in a binary image, baseball detection by circles, contour search in binary images and function smoothing by using smoothing filter.

The document reflects a series of experiments that prove the reliability of the software and the most common errors that appear in both the prediction of the trajectory and in obtaining the lines that limit the strike zone.

# Índice general

<b>Introducción</b>	<b>1</b>
<b>1. La determinación de la Zona de Strike</b>	<b>5</b>
1.1. Sistemas computacionales novedosos para seguir la bola y determinar su paso por la ZdS . . . . .	5
<b>2. Detección de la trayectoria de la bola</b>	<b>8</b>
2.1. Pre-procesamiento de las SMV . . . . .	8
2.1.1. Des-entrelazado . . . . .	9
2.1.2. Filtros de eliminación de ruido y suavizado . . . . .	9
2.1.3. Sustracción del fondo . . . . .	11
2.2. Detección de Blobs . . . . .	14
2.2.1. Filtro por área . . . . .	15
2.2.2. Filtro de circularidad . . . . .	15
2.2.3. Filtro de compacidad . . . . .	15
2.2.4. Filtro por convexidad . . . . .	16
2.2.5. Filtro por Inercia . . . . .	16
2.2.6. Conclusiones de la detección de blobs . . . . .	18
2.3. Análisis de Distribución . . . . .	18
2.3.1. Exploración de trayectorias . . . . .	19
2.3.2. Identificación de trayectorias . . . . .	19
2.3.3. Búsqueda de trayectoria aplicando algoritmo RANSAC . . . . .	20
2.3.4. RANSAC sobre XDI y YDI . . . . .	21
2.3.5. Obtención de la trayectoria y candidatos perdidos . . . . .	22
<b>3. Determinación de la Zona de Strike</b>	<b>24</b>
3.1. Detección del Home Plate . . . . .	24
3.1.1. Determinación de ancho de home plate . . . . .	25
3.1.2. Determinación de movimiento en Caja de bateo . . . . .	26
3.1.3. Detección de candidatos . . . . .	27
3.1.4. Identificación entre candidatos . . . . .	27

3.2. Detección de caja de bateo . . . . .	29
3.3. Detección de contorno del bateador . . . . .	29
3.3.1. Movimiento de píxeles . . . . .	30
3.3.2. Cambio de intensidad de colores . . . . .	31
3.4. Localización de puntos claves del bateador . . . . .	32
3.5. Determinación de puntos de la zona de strike . . . . .	33
<b>4. Detalles de Implementación y Experimentos</b>	<b>35</b>
4.1. Detalles de la aplicación . . . . .	36
4.1.1. Algoritmo de reconocimiento de trayectorias . . . . .	37
4.1.2. Algoritmo de determinación de zona de strike . . . . .	38
4.2. Experimentos y Resultados . . . . .	40
4.2.1. Conjunto de pruebas . . . . .	40
4.2.2. Experimentación . . . . .	40
4.2.3. Errores comunes en detección de trayectoria de la bola . . . . .	41
4.2.4. Valoración de los resultados de la detección de la trayectoria .	42
4.2.5. Errores comunes en determinación de la zona de strike . . . . .	43
4.2.6. Valoración de los resultados de la determinación de la zona de strike . . . . .	43
<b>Conclusiones</b>	<b>50</b>
<b>Recomendaciones</b>	<b>51</b>
<b>Bibliografía</b>	<b>53</b>

# Introducción

Desde sus inicios las transmisiones de juegos de béisbol por televisión han lograron una gran audiencia. El 17 de mayo de 1939 se transmitió el primer juego de béisbol por TV cuando Princeton y Columbia jugaron un partido de la Ivy League de Estados Unidos [Garcia 2021]. El sábado 11 de agosto de 1951 se logra televisar a todo color por primera vez un partido del béisbol de la MLB, a través de la estación WCBS-TV de la ciudad de Nueva York [Garcia 2021]. La primera transmisión de costa a costa de un juego de béisbol de la Major League Baseball (MLB) a color ocurrió el 1 de octubre de 1951 cuando la NBC televisó la victoria de los Brooklyn Dodgers sobre los New York Giants [Garcia 2021].

En Cuba las trasmisiones televisivas de béisbol se iniciaron oficialmente el 31 de Octubre de 1950 por Unión Radio TV en el Canal 4 [Tv-Historias 2011]. Sin embargo ya días antes se habían hecho trasmisiones parciales con gran éxito desde el Stadium del Cerro (hoy Estadio Latinoamericano). Desde la salida al aire de su programación ininterrumpida el 18 de diciembre de 1950, la segunda televisora cubana, CMQ TV, Canal 6, compartió con el Canal 4, las transmisiones televisivas desde el Stadium del Cerro [Tv-Historias 2011]. Desde entonces ‘la pelota’ ha ocupado un espacio prominente en la TV cubana.

El béisbol es un deporte que implica una gran cantidad de estadísticas, que a menudo se muestran durante las transmisiones televisivas para informar los niveles de rendimiento de los jugadores. Con la llegada de las nuevas tecnologías como la televisión digital con la alta definición, los avances en visión computacional, en procesamiento de imágenes y el Big Data, la cantidad y los tipos de datos utilizados en las transmisiones han aumentado cada año [Zheng y C.-Y. Chen 2022]. Los aficionados al béisbol aumentan su experiencia de entretenimiento y carga cognitiva con estas nuevas informaciones.

En un juego de béisbol una polémica habitual de los fanáticos (al menos en el contexto cubano) es determinar si un lanzamiento del *Lanzador* es *strike* o *bola*, independientemente de la decisión arbitral (vea [Núñez 2018] y [Pichardo 2022]). Un lanzamiento es *strike* si pasa por la *Zona de Strike* (ZdS) y con lanzamiento definimos el recorrido que hace la bola desde que es lanzada por el Lanzador hasta que llega al Receptor. La ZdS es aquella área sobre el home cuyo límite superior es una línea

horizontal en el punto medio entre la parte alta de los hombros y la parte alta del pantalón del uniforme del bateador, y el nivel bajo es una línea en la parte hueca debajo de las rodillas. La ZdS será determinada dada la posición del bateador al momento que este está preparado para batear un lanzamiento, como se muestra en la figura 1 (vea [FCB 2020]).

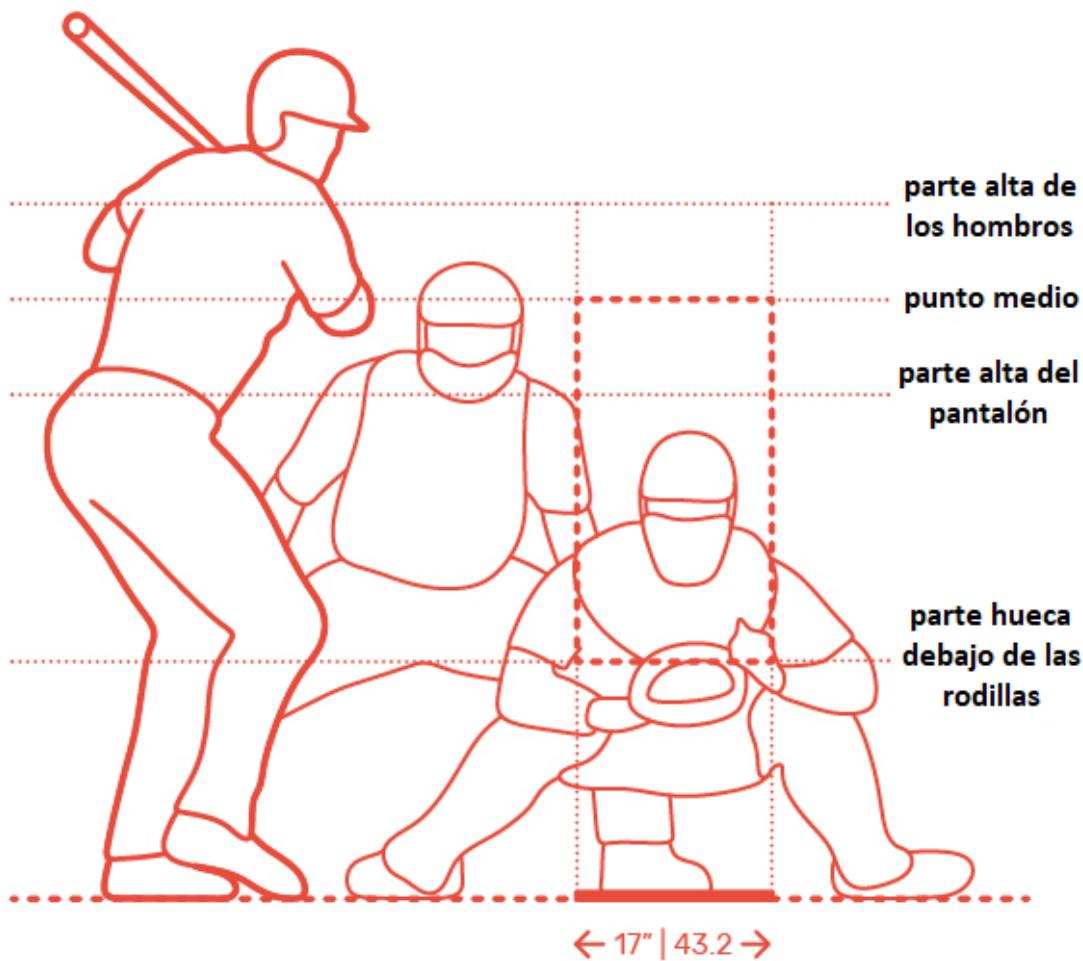


Figura 1: Definición de la Zona de strike según las reglas del béisbol.

Según [FCB 2020] la base de home deberá estar marcada por una plancha de caucho blanco de cinco lados. Deberá ser un cuadrado de 17 pulgadas con dos de las esquinas removidas de modo que un lado sea de 17 pulgadas de largo, y los dos lados adyacentes sean de 8 y media pulgadas y los dos lados restantes de 12 pulgadas, formando un ángulo que se denomina punta. Debe ser colocado en el terreno con

la punta sobre la intersección de las líneas que se extienden desde el home hacia la primera y la tercera base; con el borde de 17 pulgadas de frente a la placa de pitcheo y los dos bordes de 12 pulgadas (30,48 cm) coincidiendo con las líneas de primera y tercera base. Los bordes superiores del home deben estar biselados y la base deberá fijarse en el terreno a nivel de la superficie de la tierra.

Los árbitros mediante su apreciación deben decidir entonces si un lanzamiento que viene a una velocidad entre 75 y 95 millas aproximadamente pasa por encima del home a una altura determinada, lo cual es complejo. El árbitro es una persona y como tal puede tener una equivocación. Este problema no lo podemos solucionar pero sí posible entregar al televíidente una ayuda visual para que estos puedan determinar si el lanzamiento pasa por la ZdS. Sistemas como el descrito anteriormente para el enriquecimiento visual de las transmisiones ya son de uso común por las compañías de televisión que transmiten la MLB (a saber ESPN y TBS). Estas compañías con cámaras adicionales y dedicadas específicamente a la tarea (vea el capítulo 1 para una descripción más amplia de este sistema) logran imponer sobre la transmisión usual un rectángulo transparente que determina la ZdS con gran precisión.

En Cuba a pesar de contar ya con transmisiones de béisbol en alta definición aún no se dispone de un sistema similar. Una dificultad para importar alguno de los ya establecidos radica en su alto costo. Sin embargo empleando solo las secuencias monoculars de video (SMV) que transmite la TV se puede desarrollar un sistema que permita en muchos casos determinar si un lanzamiento es bola o strike. Es esa la tarea que se abordará en el presente trabajo.

#### **Objetivo general:**

- Desarrollar un software que detecte automáticamente la trayectoria de una pelota al ser lanzada por un pitcher y determine la Zona de Strike en una secuencia monocular de video.

#### **Objetivos específicos:**

- Analizar las más relevantes investigaciones anteriores relacionadas con los problemas a abordar.
- Implementar varias técnicas para el procesamiento de videos e imágenes.
- Determinar la trayectoria de la bola de béisbol en SMV.
- Determinar la Zona de Strike en SMV a partir de la posición del bateador y la ubicación de la base home.
- Evaluar la calidad del software.

El desarrollo del presente Trabajo de Diploma se fundamenta a partir de la siguiente hipótesis: dadas las características particulares de la pelota, su movimiento, forma y aspecto, así como las características de la base home es posible encontrar la trayectoria del lanzamiento así como las líneas que delimitan la Zona de Strike en SMV transmitidas por la TV.

Para demostrar la validez de la anterior hipótesis el Trabajo de Diploma se estructura de la siguiente forma: En el primer capítulo se abordan investigaciones y softwares relacionados con nuestros objetivos que existen en el mundo. El segundo capítulo contiene la descripción de la tarea de determinar la trayectoria de la bola de béisbol en la secuencia de imágenes (cuadros) del lanzamiento. En el tercer capítulo se describen los procesos que se realizan para lograr detección de los 4 puntos que determinan el rectángulo de la Zona de Strike. El cuarto capítulo se destina a explicar importantes detalles de la implementación del sistema propuesto. Muestra el funcionamiento para varias escenas tomadas de la televisión cubana y se hace un estudio estadístico para evaluar la efectividad del algoritmo presentado. La tesis finaliza presentando las conclusiones obtenidas como resultado de la investigación además de las recomendaciones para trabajos futuros. Al final del trabajo se muestran las referencias bibliográficas consultadas.

# **Capítulo 1**

## **La determinación de la Zona de Strike**

La insatisfacción (justificada o injustificada) de los fanáticos beisboleros cubanos con los conteos de strikes o bolas tiene larga data y ha sido reflejada por la prensa escrita (vea por ejemplo [Núñez 2018]). Recientemente el destacado árbitro cubano Luis César Valdés declaró «El tema de los conteos es difícil porque no se ejercita, no existe un sitio donde uno diga voy a contar bolas y strike, esto sólo se hace en los colegios que hay en Japón, México y las Grandes Ligas» [Pichardo 2022].

Con el propósito de contribuir a aliviar (o avivar) esta polémica en el presente capítulo se presentan sistemas que se emplean en otras ligas de béisbol para capacitar a los árbitros e informar a los espectadores durante la transmisión de un partido de béisbol. También se detalla la solución que proponemos limitada por el empleo de secuencias monoculars transmitidas por la TV cubana.

### **1.1. Sistemas computacionales novedosos para seguir la bola y determinar su paso por la ZdS**

El primer software utilizado para obtener las velocidades y trayectorias de las pelotas de béisbol lanzadas fue PITCHf/x [Fast 2010]. Este sistema, que hizo su debut en la posttemporada de la Major League de Béisbol (MLB) de 2006, fue instalado en todos los estadios de la MLB. Los datos del sistema fueron utilizados por los organismos de radiodifusión para mostrar una representación visual del terreno de juego y si un lanzamiento entró o no en la ZdS, también se utilizó para determinar el tipo de lanzamiento, como una recta, una curva o un slider. PITCHf/x utiliza tres cámaras montadas permanentemente en el estadio para rastrear la velocidad y la ubicación de una pelota de béisbol lanzada desde el montículo del lanzador hasta el

plato con una precisión superior a una milla por hora y una pulgada.

La MLB utilizó los datos de PITCHf/x en su Sistema de Evaluación de Zona que se utilizó para calificar y proporcionar retroalimentación a los árbitros. Los analistas sabermétricos señalan que la precisión de los árbitros mejoró después de que esta tecnología se introdujo en la MLB.

Desde 2017 PITCH/f fue reemplazado en la Major League Baseball(MLB) por [TRACKMAN 2022]. Es un sistema que utiliza la patente Optically Enhanced Radar Tracking (OERT) que provee a la MLB un sistema de radares de la más alta gama de la industria para evaluación y desarrollo de jugadores. Proporciona información del juego en vivo, de cada lanzamiento y cada batazo puliendo la experiencia de los aficionados en las transmisiones de televisión, además de la oferta de entretenimiento a los aficionados.[TRACKMAN-LMB 2022]

El sistema mide la trayectoria completa y la velocidad de giro de pelotas de béisbol lanzadas y bateadas a través de una cámara de "óptica de alta velocidad" que puede proporcionar 1,000 cuadros por segundo. Esto no solo se traduce en una gran cantidad de datos que se recopilan detrás del montículo, sino también en una alta calidad de datos que se pueden comunicar fácilmente al jugador durante la instrucción. Track-Man es utilizado para la evaluación y desarrollo de jugadores por todos los equipos de ligas mayores y ligas menores en los Estados Unidos [TRACKMAN-Function 2022].

Ball-Strike [Ball-Strike(ABS) 2022] es un software que se está utilizando en la actualidad en las ligas menores de béisbol en Estados Unidos, software que está en fase de prueba. Su objetivo es muy parecido al objetivo de nuestro producto porque detecta si un lanzamiento es strike o bola pero no solo funciona como aporte televisivo en este caso sino que está influyendo directamente en el juego de béisbol. Al principio de cada juego cada equipo tiene 3 oportunidades para reclamar un lanzamiento. Cada vez que se reclama un lanzamiento se utiliza el software, si la reclamación prospera entonces el equipo no pierde una de sus oportunidades, en otro caso pierde una oportunidad de reclamación. Una reclamación solo puede hacerla el bateador, el catcher o el pitcher, sin otra ayuda lo cual impide que se puedan apoyar en una transmisión televisiva para saber cuando reclamar o no. Hay informaciones de posible utilización en venideros años en la MLB por lo que al parecer el software ha tenido un impacto positivo para el béisbol.

La MLB no es la única que utiliza la tecnología TrackMan, en otras ligas de primer nivel como la Liga Japonesa de Béisbol Profesional (NPB) y la Korean Baseball Organization (KBO) un gran número de equipos profesionales cuentan con ella. La Liga Mexicana de Béisbol(LMB) anunció recientemente que será la primera Liga profesional de béisbol en Latinoamérica con el sistema Trackman. Este acuerdo traerá beneficios directos para los aficionados, transmisiones de TV, desarrollo y evaluación de jugadores [TRACKMAN-LMB 2022]. Otras ligas importantes como la liga canadiense de béisbol (CAN-AM) y la liga italiana de Béisbol(IBL) no utilizan softwares

ni tecnologías de este tipo.

Todos los sistemas mencionados anteriormente emplean más de una cámara y hasta radares para lograr el seguimiento preciso de la bola de béisbol. Empleando secuencias monoculares de vídeo el presente trabajo tiene como precedentes los de [Bas-Martínez 2014] y [González-Maqueira 2015] en los cuales se trata de detectar la trayectoria de la bola.

El trabajo [Bas-Martínez 2014] es un primer acercamiento al análisis de videos de béisbol. Se trata de una aplicación que logra la detección y el seguimiento de la pelota en un lanzamiento. En él se explican algunas propiedades de las imágenes digitales, procedimientos para el procesamiento de imágenes y se muestran técnicas para la detección de objetos en movimiento, la determinación de los candidatos a bola y la identificación de la trayectoria .

En el trabajo [González-Maqueira 2015] se propuso un software para automatizar el proceso de detección de la trayectoria de una pelota al ser lanzada por el pitcher y al igual que en [Bas-Martínez 2014] se describen técnicas para la determinación de los candidatos a bola y la identificación de la trayectoria. Estas técnicas pueden ser de ayuda para el desarrollo del presente trabajo.

Los trabajos [H.-T. Chen, Tsai y col. 2009] y [H.-S. Chen y col. 2007] realizados por el equipo de investigadores formados por Chia-Hsin Chan, Ya-Ching Chang, Yee-Choy Chean, Hsuan-Shen Chen y Yi-Wen Chen, describen detalladamente procesos computacionales de determinación de home, de las cajas de bateo, de la silueta del bateador y de los puntos de rodillas, caderas y hombros. Incluso se muestra más de una forma de realizar algunos de dichos procesos. Es un antecedente importante para el desarrollo de este trabajo porque el proceso completo que describen fue probado por ellos a través de una muestra de videos y tiene una efectividad de un 98 por ciento, lo cual es una garantía para utilizar dichos procesos en la implementación del software.

# **Capítulo 2**

## **Detección de la trayectoria de la bola**

Analizando detalladamente la posición que toma la bola de béisbol en cada cuadro de la SMV en su trayecto hacia el plato, se destacan una serie de características fundamentales. Con ayuda de estas características se puede detectar y determinar la trayectoria original de la bola desde su salida de la mano del lanzador hasta la mascota del receptor de forma automática. En el presente capítulo se explica el procedimiento a seguir para la obtención de la trayectoria correcta de la bola que permite determinar su posición al pasar por encima de home. Este procedimiento se divide en varios pasos los cuales posibilitan el cumplimiento del objetivo. Primeramente se hace un pre-procesamiento de las imágenes buscando detallar mejor los objetos contenidos en estas. Luego se procede a detectar posibles candidatos a bolas mediante la detección de manchas sobre las imágenes. Finalizando con la aplicación algoritmos de tipo Ransac para encontrar dentro de los posibles candidatos, las bolas que contienen a la trayectoria original del lanzamiento.

### **2.1. Pre-procesamiento de las SMV**

Las imágenes en ocasiones pueden presentar anomalías, debido a la calidad de la cámara, la inexperiencia del camarógrafo o la situación de la toma, que impiden la visualización efectiva de ciertos objetos que aparecen en ella. En el presente trabajo se opera con SMV tomadas de transmisiones televisivas de juegos de béisbol nacionales, realizadas en diferentes provincias y horarios por lo que estas difieren en calidad, claridad y resolución.

Teniendo en cuenta lo anterior sobre cada cuadro (imagen) de la SMV se emplea un proceso de mejora, con técnicas del pre-procesamiento de imágenes, buscando convertirlos en cuadros óptimos para la detección de la trayectoria del lanzamiento y

de la ZdS. Este proceso se describe a continuación.

### 2.1.1. Des-entrelazado

En el mundo de la televisión existe una técnica, motivada por las antiguas pantallas de tubo, para duplicar la velocidad de los fotogramas contenidos en un video sin consumir ancho de banda extra. Esta técnica conocida como *entrelazado de video* permite mejorar la percepción de movimiento y eliminar las sensaciones de parpadeo entre imágenes [Wikipedia contributors 2022]. La TV cubana aún emplea esta técnica en sus transmisiones por lo que las SMV tomadas en este trabajo la presentan

El entrelazado de video tiene una relación directa con la calidad con que se disponen en los cuadros de una secuencia los objetos que se mueven a una alta velocidad. Es este el caso de la bola de béisbol cuya velocidad promedio varía entre las 75 y 90 mph. Un video entrelazado produce una deformación entre cuadros consecutivos de los objetos en rápido movimiento (figura 2.1a). Dada esta situación se hace necesario aplicar a las SMV un algoritmo de des-entrelazado para minimizar las deformaciones de la bola y obtener una imagen más nítida de la misma. El algoritmo utilizado para el proceso de des-entrelazado fue el algoritmo de Yadiff el cual basa su funcionamiento en verificar los píxeles del cuadro anterior, actual y siguiente para completar el campo faltante usando métodos adaptativos locales [Merényi s.f.] del paquete de programas FFMPEG [Wikipedia 2022] dando como resultado que la bola mantenga su aspecto circular (figura 2.1b)



(a) Entrelazado

(b) Des-entrelazado

Figura 2.1: Comparación de entrelazado y des-entrelazado

### 2.1.2. Filtros de eliminación de ruido y suavizado

El ruido en una imagen digital es la variación aleatoria del brillo o el color de un píxel debido a la aportación lumínica de la escena y el dispositivo de entrada. Las

imágenes pueden contener ruidos provocados por diversas fuentes, como los sensores ópticos, eléctricos, mecanismos de apertura en cámaras fotográficas, y también debido a la transmisión de dichas imágenes a través de un canal físico, según se define en [Rivas 2003]. De manera general se puede representar un ruido en un píxel por la ecuación

$$I(x,y) = I_0(x,y) + n \quad (2.1)$$

donde  $(x,y)$  son las coordenadas del píxel,  $I(x,y)$  es el color o nivel de brillo del píxel en la imagen con ruido,  $I_0(x,y)$  es lo correspondiente en la imagen sin ruido y  $n$  es el ruido aleatorio. El nivel o cantidad de ruido en una imagen no tiene manera definitiva para ser detectado mediante algoritmos, usualmente solo es perceptible por la apreciación humana.

Debido a la posible presencia de ruido a cada cuadro de la SMV se le aplica un proceso de reducción de ruido buscando imágenes de mejor calidad para el resto del proceso. Existen diferentes tipos de algoritmos destinados a la reducción de ruido en imágenes entre los que destacan los filtros lineales, los no lineales y los temporales. Los filtros se aplican píxel a píxel y no solo tienen en cuenta la intensidad propia sino también la intensidad de los píxeles de una vecindad o de otras regiones de la imagen. Los filtros lineales utilizan la convolución de una imagen con una máscara de convolución. Los filtros no lineales realizan operaciones no lineales con los píxeles de una vecindad de un píxel y los filtros temporales analizan varias imágenes de la misma escena tomadas en diferentes instantes de tiempo. [Cepero Ruz y col. 2010] Entre los distintos filtros posibles se seleccionaron dos, el filtro de mediana y el filtro gaussiano, los cuales contribuyen a tener una mayor calidad de la imagen a analizar.

### Filtro de Mediana (figura 2.2b)

Dada una imagen  $f(i,j)$ , el procedimiento consiste en generar una nueva imagen  $g(i,j)$  cuya intensidad para cada píxel es la mediana del conjunto de píxeles que forman su entorno de vecindad. De esta manera se homogeneizan los píxeles de intensidad muy diferentes con respecto a la de los vecinos. Este filtro es efectivo cuando se tiene presencia de ruido aleatorio. La operación consiste en analizar una matriz de píxeles de  $N \times N$  y reemplazar el píxel central por el valor de la mediana de todos ellos y repetir el algoritmo recorriendo toda la imagen [Cepero Ruz y col. 2010].

### Filtro Gaussiano(figura 2.2c)

Este filtro de tipo lineal elimina el ruido de manera eficiente y emborrona menos los bordes que los otros filtros pertenecientes a la misma categoría. El valor de cada píxel es el resultado de promediar con distintos pesos los valores vecinos a ambos lados de dicho píxel. Para calcular la influencia de cada píxel del entorno de vecindad se utiliza la distribución gaussiana. [Cepero Ruz y col. 2010]

$$f(x) = ae^{\left(\frac{(x-b)^2}{2c^2}\right)} \quad (2.2)$$

$f(x)$  resulta en un número real entre 0 y 1 que representa el porcentaje de influencia mientras que  $a$ ,  $b$  y  $c$  son constantes reales. Usualmente  $a = 1$ , y  $x$  se remplaza por la distancia entre el píxel que está siendo modificado y el píxel del que se está tomando el valor;  $c$  es la varianza del ruido.

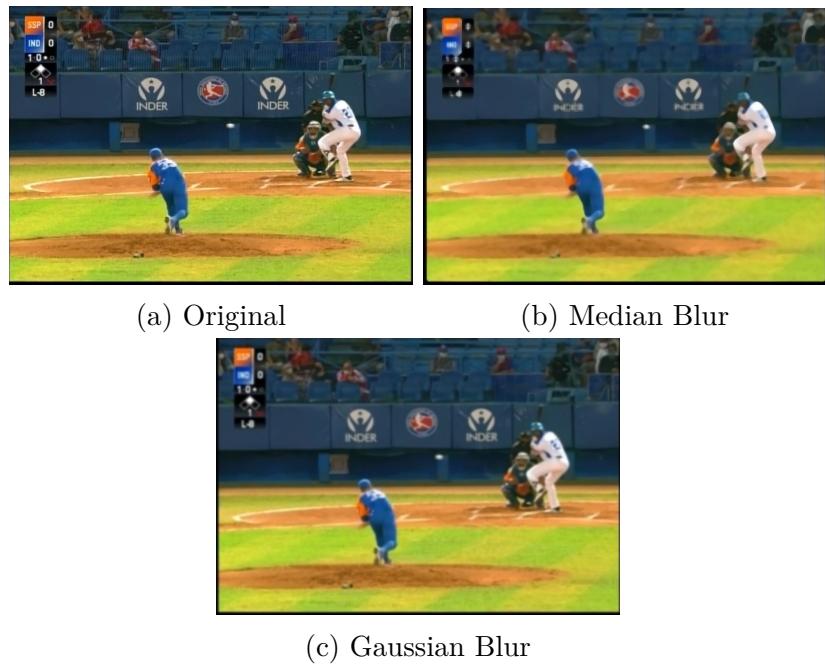


Figura 2.2: Comparación de filtros

### 2.1.3. Sustracción del fondo

Evidentemente en una SMV de un lanzamiento la bola de beisbol es un objeto en movimiento que no forma parte del fondo de la imagen. Es por ello que un paso esencial de la determinación de la trayectoria de la bola consiste en la sustracción del fondo en cada uno de los cuadros de la SMV.

Como se define en [Escobar Tafurt 2016] la sustracción de fondo se refiere al proceso en el cual un cuadro, usualmente de una secuencia de video, es analizado con el fin de determinar si cada píxel es parte del fondo o del frente. Por ello también se le conoce como detección de primer plano. Existen diversos algoritmos de sustracción de fondo pero la mayoría de ellos siguen un flujo definido por Cheung y Kamath [Cheung y Kamath 2005], pasando por los siguientes pasos.

- Pre-procesamiento: Aplicar técnicas de procesamiento de imágenes para cambiar la entrada por una que pueda ser procesada.
- Modelación del fondo: Utilizado para mantener el fondo fijo.
- Detección del primer plano: Es donde se aplica lo conocido como sustracción de fondo.
- Validación de los datos: Utilizado para eliminar falsos positivos y validar la calidad del resultado.

Para un trabajo óptimo, el proceso de sustracción de fondo se basa en varios criterios fundamentales como es que la cámara siempre esté estática, no existan cambios abruptos de iluminación, fondo visible sin objetos ocultos, entre otros. No todos estos criterios se cumplen en las grabaciones televisivas de béisbol pero existen métodos que permiten resultados bastante aceptable para el desarrollo del software.

Durante la realización de la investigación se valoraron distintos algoritmos de susstracción de fondo cada uno con un conjunto de características distintivas y diferentes entre si. A continuación se presentan los más relevantes entre los analizados.

### Diferencia de cuadros

El método se basa en tomar dos imágenes consecutivas de la SMV y calcular la diferencia del valor de intensidad entre cada píxel de cada una. Se define un valor umbral que determina si el píxel es fondo o frente. El algoritmo calcula la diferencia de la  $k$ -esima imagen  $I_k$  de la secuencia con la imagen  $I_{k-1}$  correspondiente al frame  $k - 1$  [Bouwmans y col. 2014]. En OpenCV se tiene una implementación simple de esta técnica. Consiste en ir recorriendo los cuadros que conforman el video y aplicando la función `absdiff` al cuadro  $N$  y al cuadro  $N - 1$ . Esta función calcula los cambios en los píxeles de una imagen a otra

### Mixture Of Gaussians (MOG)

Es un algoritmo de segmentación de fondo y frente basado en mezcla gaussiana. Una característica destacada de este algoritmo es que selecciona el numero apropiado de distribuciones gaussianas para cada píxel.

Este algoritmo presenta una mejor adaptación en escenarios cambiantes por la iluminación ambiente realizando una selección apropiada del número de distribuciones gaussianas por cada píxel. En OpenCV se muestra la función `createBackgroundSubtractorMOG2()` la cual trabaja con una imagen y devuelve otra con el fondo sustraído.

## Comparación de los algoritmos

Los dos algoritmos brevemente presentados anteriormente se muestran robustos y eficientes para la solución del problema de detección de objetos en movimientos a alta velocidad. Existen muchos otros que sin embargo no cumplen completamente los requisitos para la solución del problema presentado sobre las SMV de transmisiones de béisbol de la TV cubana [Bouwmans y col. 2014].

Para la selección de los algoritmos de sustracción del fondo que se utilizarán en nuestro software se tuvieron en cuenta criterios imprescindibles a cumplir para lograr una solución factible. Primero se prioriza la velocidad de procesamiento. Debido a la cantidad de cuadros que deben ser analizados y la necesidad de proporcionar una solución en pocos segundos un algoritmo veloz es imprescindible. Por otra parte también se valora la sensibilidad de los algoritmos a los cambios, tanto de iluminación, de movimiento de la cámara, así como la cantidad de objetos en movimiento.

Las imágenes 2.3 y 2.4 muestran la aplicación de los algoritmos anteriormente mencionados y del algoritmo ViBE (Visual Background Extraction) [Barbuzza y col. 2017] que es uno de los más recientes y efectivos pero demasiado lento para nuestros propósitos. Debido a la variación de los horarios de los partidos de béisbol se realizó primero una muestra con un partido grabado en el horario diurno (figura 2.3) y otro realizado en el horario nocturno (figura 2.4)

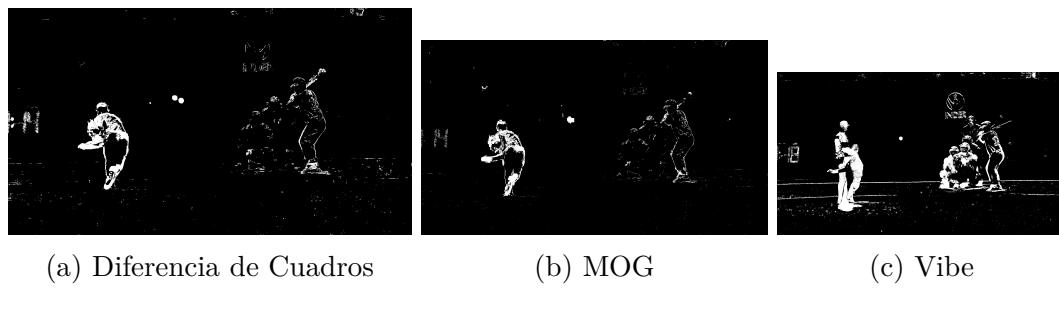


Figura 2.3: Comparación de algoritmos de sustracción de fondo en horario diurno

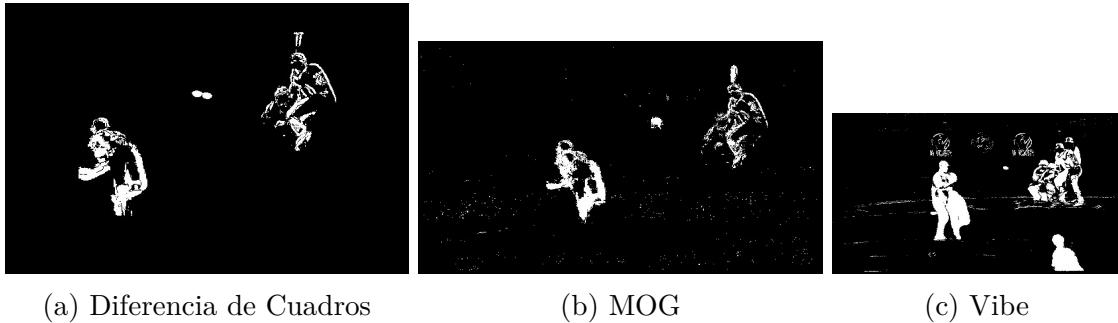


Figura 2.4: Comparación de algoritmos de sustracción de fondo en horario nocturno

Los algoritmos de diferencia de cuadros y MOG aunque difieren un poco, sobresalen sobre los otros analizados durante la investigación. En la figura 2.3 se aprecia como aparece un poco de ruido en la imagen debido a los cambios de iluminación natural, efecto que no se ve reflejado en la figura 2.4. El algoritmo de diferencias de cuadros es bastante eficiente y requiere del menor tiempo de procesamiento. El algoritmo MOG muestra un tiempo aceptable de procesamiento y destaca bastante dada su capacidad de adaptarse a ambientes que no son los ideales para ejecutar el proceso de sustracción de fondo, tiene una fácil adaptabilidad a los cambios de iluminación, así como logra evadir los movimientos leves de la cámara producidos por no estar esta siempre estática en las transmisiones de la TV cubana.

## 2.2. Detección de Blobs

Una vez realizada la sustracción de fondo las imágenes de la SMV del lanzamiento se binarizan de forma tal que los píxeles del fondo tienen valor de intensidad cero (negro) y los objetos en movimiento valor máximo de intensidad (blanco). Se asume que en cada cuadro de la SMV binarizada aparece la bola como un conjunto simplemente conexo de píxeles en blanco.

Como se define en [Paguay Paguay y Urgilés Ortiz 2012] los blobs (gota o mancha en inglés) son regiones de píxeles más luminosos (o más oscuros) que los de los alrededores y que forman una “mancha” aislada del resto de la imagen. En nuestro contexto estos blobs corresponden a objetos en movimiento en la SMV original. La aplicación del método de detección de blobs a cada cuadro (figura 2.6) de la SMV nos permite obtener una distribución de puntos (centro del blob) que constituyen los candidatos a ser la bola de béisbol en el cuadro (vea figura 2.7).

En OpenCV se tiene una herramienta la cual facilita el trabajo con la detección de manchas. Conocida como `SimpleBlobDetector`, esta función utiliza diversos parámetros para obtener las manchas sobre la imagen. Utiliza varios filtros y pone límites

inferiores y superiores haciendo que los píxeles que formen un blob tenga que estar comprendidos entre estos para ser validados

Como objetivo tenemos entonces determinar el blob correspondiente al conjunto de píxeles que comprende la bola de béisbol. En una imagen de la SMV binarizada mediante sustracción de fondo la detección de blobs da como resultado un conjunto de cardinalidad bastante grande (vea figura 2.7). Se requiere reducir este conjunto mediante características especiales del objeto que se esta localizando, en nuestro caso la bola de béisbol. La aplicación de varios filtros y otras restricciones permite eliminar algunas regiones identificadas como blobs por el algoritmo de detección y de las cuales se conoce con anterioridad que no pueden formar parte del conjunto solución.

### 2.2.1. Filtro por área

La bola de béisbol presenta un tamaño menor a otros objetos en movimiento en el cuadro como son el lanzador, el receptor o incluso el arbitro. No obstante es mayor que la empuñadura del bate o alguna letra publicitaria parpadeante. Teniendo esta consideración se puede obtener que una mancha con un área mayor o menor de ciertos umbrales específico debe ser descartada eliminando así candidatos a bola que solo dificultaran la búsqueda de la solución. Para el proceso de detección de bolas en imágenes con resolución 1920x1080 se busca un área comprendida entre 400 y 1200.

### 2.2.2. Filtro de circularidad

Al ser la bola de béisbol un objeto que se mueve a alta velocidad puede presentar una deformación en alguno de los cuadros. No obstante siempre mantendrá al menos una forma elíptica y bastante cercana al círculo. En OpenCV se toma como un circulo perfecto un valor de  $c = 1$ , la bola de béisbol por lo dicho anteriormente no tendrá esta perfección pero no existirá una deformidad demasiado amplia por lo que se propone un valor de  $c = 0,8$

### 2.2.3. Filtro de compacidad

Un objeto que no es la bola puede aún pasar los filtros de área y circularidad debido a su tamaño aceptable y su relación de aspecto adecuada. Por esta razón, el filtro de compacidad se emplea para eliminar aquellos objetos con un grado de compacidad  $D_c$  inferior a un umbral  $T_c$ . Los objetos con  $D_c$  baja se filtrarían mientras que los objetos con  $D_c$  alta se conservarían. El grado de compacidad  $D_c$  se define mediante

$$D_c = \frac{A_{obj}}{A_{bb}} \quad (2.3)$$

donde  $A_{obj}$  es el área del objeto (número de píxeles que componen el blob) y  $A_{bb}$  el área del la caja con lados alineados con los ejes en la cual se encuentra inscrito (bounding box en inglés). En OpenCV se propone declarar una límite inferior y superior de intensidad y los umbrales que no estén en este rango serán descartados como posibles candidatos. Para el caso de la búsqueda de la bola se propone un umbral mínimo de 200 y máximo de 500

#### 2.2.4. Filtro por convexidad

Debido a los cambios veloces de posición, la cámara puede perder parte de la bola al detectar un blob, teniéndose solo una porción de un circulo o una elipse, haciendo que una potencial bola no se detectara. Este filtro flexibiliza que se puedan tener manchas las cuales presenten porciones faltantes dentro de la circunferencia que lo delimita

#### 2.2.5. Filtro por Inercia

Este filtro no solo considera la forma circular clásica de la bola sino que permite aceptar una variación de su forma de manera alargada.

En la figura 2.5 se muestran las características visuales de cada filtro.

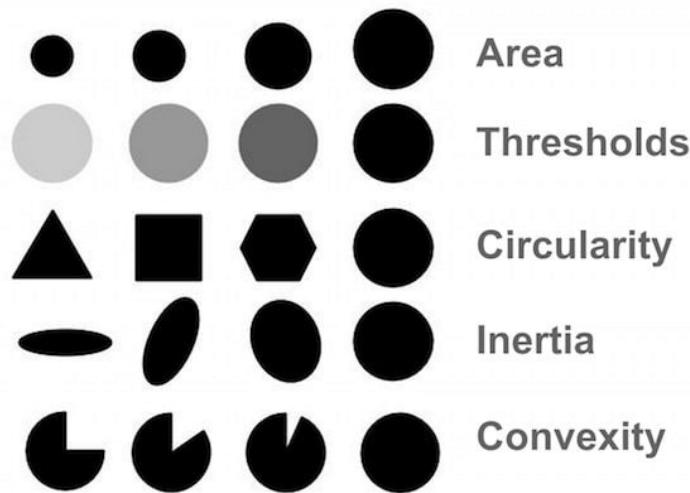


Figura 2.5: Muestra visual de los diferentes filtros

La utilización de filtros como los anteriores suele no ser suficiente, analizando en cuestión de términos de ubicación se encuentra con que la bola siempre se un detectara

en cuadros continuo o muy cercanos por lo que un blob detectado en cuadro donde en su epsilon-vecindad no exista otro puede ser eliminado. Los puntos aislados son considerados no relevantes, al analizar el lanzamiento este se encuentra comprendido en una región , delimitada por el lanzador y el receptor, por lo que un punto que tenga sus coordenadas demasiada alejada del resto queda también eliminado .

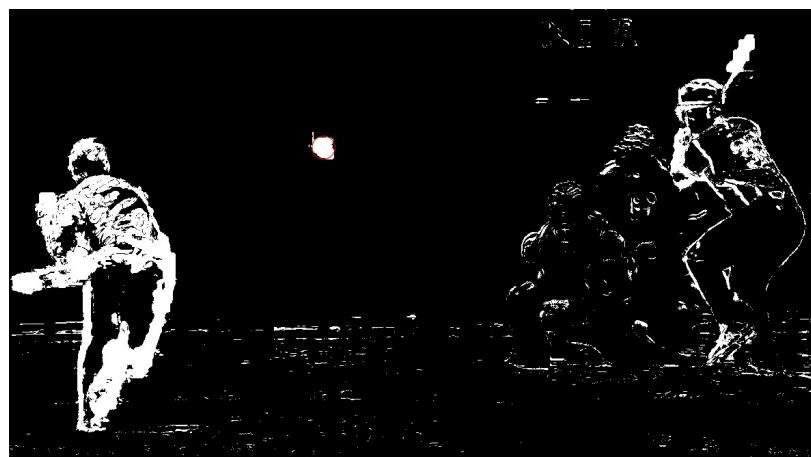


Figura 2.6: Detección de manchas en un cuadro de video

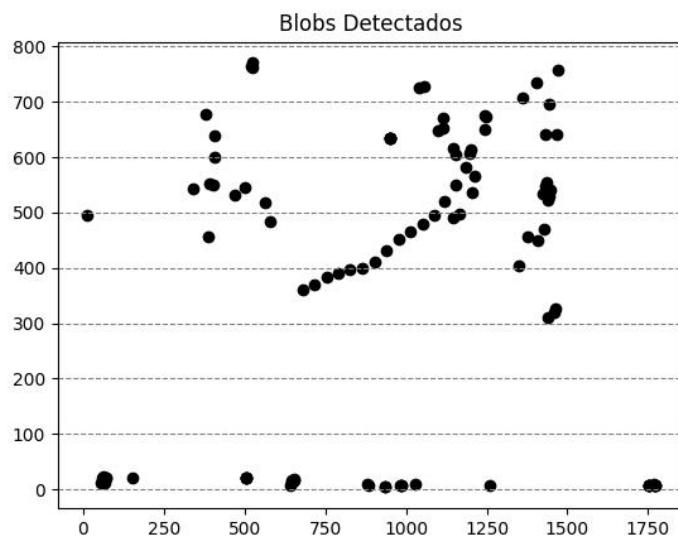


Figura 2.7: Representación de Blobs detectados

### 2.2.6. Conclusiones de la detección de blobs

El proceso de obtención y filtrado de blobs en las imágenes ya procesadas mediante sustracción de fondo es bastante veloz debido a que un lanzamiento no está compuesto por demasiados cuadros. Las operaciones de eliminación de puntos innecesarios o filtrado tampoco trabajan sobre un conjunto demasiado grande por lo cual este proceso también se ejecuta rápidamente. En definitiva el algoritmo de detección y filtrado de blobs es uno bastante simple y efectivo para la detección de candidatos a bola de béisbol. Este algoritmo es bastante adecuado a las necesidades de encontrar la trayectoria de la bola de béisbol en transmisiones televisivas con suficiente velocidad.

Luego de la obtención y filtrado de blobs se tiene una selección bastante reducida de puntos en cada cuadro considerados relevantes para la obtención de la trayectoria completa del lanzamiento. No obstante ciertos factores influyen en que no se tengan un único punto en cada cuadro con la ubicación precisa de la pelota o que en alguno no aparezca el correspondiente a la misma. Por ejemplo, la aproximación al receptor produce una mezcla de los píxeles de la bola con los de la mascota, también el paso de la bola por una región que tenga de fondo un bateador con uniforme blanco (caso bastante común en los lanzamientos de un lanzador derechos contra bateadores zurdos) entre otras cuestiones del ambiente. Dadas estas carencias el algoritmo de detección y filtrado de blobs se complementa con un método para encontrar en el conjunto total de puntos en cada uno de los cuadros los únicos correspondientes a la bola y hasta lograr obtener los faltantes (de existir esta situación).

## 2.3. Análisis de Distribución

En una escena de lanzamiento, la trayectoria de la bola describe una curva casi parabólica independientemente que sea una bola rápida o “recta” en argot del béisbol. Al analizar por separado el movimiento la vertical y la horizontal de la pelota se tiene que en la dirección vertical esta describe una parábola debido a la gravedad y en la horizontal una recta. Basándose en estas características en [H.-T. Chen, H.-S. Chen y col. 2008] se propone un análisis de distribución 2D para explorar posibles trayectorias.

En la (figura 2.8) se ilustra el análisis de distribución de candidatos. La distribución de candidatos se presenta por separado para las coordenadas  $x$  y  $y$  de los puntos candidatos. La X-Distribución se obtiene al tomar en el eje de las abscisas el número del cuadro y en el de las ordenadas la coordenada  $x$  del punto candidato (figura 2.8a). De igual manera se obtiene la Y-Distribución mediante la coordenada  $y$  del punto candidato (figura 2.8b).

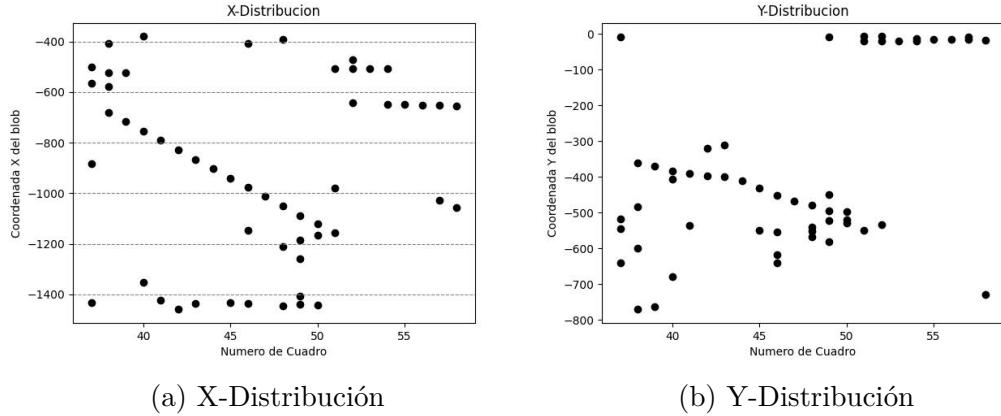


Figura 2.8: Representación de X-Distribución y Y-Distribución

### 2.3.1. Exploración de trayectorias

El procedimiento se basa en la utilización de las distribuciones anteriormente descritas, para generar todas las posibles trayectorias y luego obtener la verdadera. Basado en el principio que en la bola genera rectas en la X-Distribución y paráolas en la Y-Distribución se busca los puntos que cumplan con ambas descripciones simultáneamente. El primer paso es vincular los candidatos con los vecinos más cercanos, dado que se buscan rectas y paráolas se utilizan las ecuaciones [H.-T. Chen, H.-S. Chen y col. 2008]

$$\begin{aligned} y &= mx + n, \\ y &= ax^2 + bx + c. \end{aligned}$$

Mediante estas ecuaciones se trata de predecir la posición de la bola en el siguiente cuadro. Partiendo de estimados de los valores de  $\{m, n, a, b, c\}$  se analizan los puntos del cuadro  $k$  con los del  $k+1$ . Si se tiene que dos puntos no cumplen las ecuaciones, es decir al sustituir en ellas se obtiene un valor mayor a uno fijado, no se considera que el punto  $k+1$  pertenece al grupo de candidatos de la trayectoria que se está evaluando. En el caso contrario el conjunto aumenta al incorporar el punto  $k+1$  como parte de la trayectoria y se actualizan las ecuaciones para nuevos valores de  $\{m, n, a, b, c\}$ . El proceso termina cuando existe un número de puntos restantes menos que un valor prefijado.

### 2.3.2. Identificación de trayectorias

Terminada la exploración se tiene una serie de conjuntos  $X_i \subset XDI$  y  $Y_j \subset YDI$  donde  $X_i$  y  $Y_j$  son conjuntos que conforman alguna trayectoria factible en la X-

Distribución y en la Y-Distribución respectivamente [Chu y col. 2006]. Estos conjuntos son filtrados y así de las trayectorias potenciales se escoge la real del lanzamiento. Por cada conjunto se han mantenido los valores  $\{m, n, a, b, c\}$  que mejor ajustan las trayectorias. Los siguientes criterios se utilizan para eliminar trayectorias falsas:

- **Longitud de la trayectoria:** La distancia del lanzador al receptor es de aproximadamente 18 m y la velocidad promedio de un lanzamiento varía entre 33 m/s (75 mph) y 40 m/s (90 mph) por lo que la bola demora menos de medio segundo en recorrer la distancia. Dado que las transmisiones de TV se tienen a 30 cuadros por segundo una trayectoria con más de 15 puntos (equivalentes a 15 cuadros) puede considerarse como falsa.
- **Eliminación de candidatos no aislados:** La bola normalmente debe aparecer en una secuencia de cuadros muy cercanos unos a otros ya que aunque es posible no detectar una bola en un cuadro (por que se superponga con el bateador u otra causa) en los siguientes cuadros debe aparecer. Por ello una trayectoria que tenga un espacio muy grande también queda eliminada.

Después de la eliminación, quedan muchas menos trayectorias potenciales. Para cada trayectoria restante, calculamos la longitud de candidatos consecutivos a bolas aisladas. La trayectoria con la mayor longitud de candidatos aislados consecutivos se finaliza y se extrae como la trayectoria de la pelota.

### 2.3.3. Búsqueda de trayectoria aplicando algoritmo RANSAC

Volviendo al principio que en la X-distribución y en la Y-Distribución la trayectoria se percibe como una recta y como una parábola respectivamente, el objetivo se resume en ajustar dichas funciones simultáneamente sobre los candidatos en las distribuciones. Para ello se presenta otra vía de solución basada en algoritmos tipo RANSAC.

El Random sample consensus (**RANSAC**) es un método iterativo para realizar ajuste de parámetros a un modelo matemático sobre un conjunto de datos que contiene valores atípicos. Es un algoritmo no determinista en el sentido de que produce un resultado razonable solo con una cierta probabilidad, que se incrementa a medida que se permiten más iteraciones. El algoritmo fue publicado por primera vez por Fischler y Bolles en 1981 [Fischler y Bolles 1981]. Este método busca la mejor aproximación que cumple con el modelo bajo análisis. Para ello se selecciona aleatoriamente un subconjunto de puntos y se utilizan para estimar los parámetros iniciales del modelo. Una vez definidos, el algoritmo evalúa el conjunto de puntos restantes para ver si coinciden con los parámetros ya definidos, esta coincidencia tiene un grado de tolerancia

establecida previamente. Los parámetros resultantes se obtienen de la estimación que contenga la mayor cantidad de puntos coincidentes.

### 2.3.4. RANSAC sobre XDI y YDI

Para encontrar la recta que mejor ajusta en XDI se utiliza el algoritmo RANSAC pero con ciertas modificaciones que permiten hacerlo más eficiente. La selección de puntos no pasa a ser de manera aleatoria sino de forma secuencial. Sean  $(i, x_{ij})$  y  $(k, x_{kl})$  los puntos a partir de los cuales obtendremos la recta, estos deben cumplir con:

- $k > i$  como se busca secuencialmente el segundo punto será de un cuadro posterior, si fuera del anterior ya se hubiera analizado,
- $x_{kl} > x_{ij}$  Durante el estudio de varias distribuciones se ha comprobado que la recta siempre posee pendiente positiva si se trabaja con las coordenadas de los píxeles originales

Al tener seleccionados dos puntos iniciales, estos se utilizan para determinar la ecuación de la recta correspondiente. Se obtiene

$$m = \frac{x_{kl} - x_{ij}}{k - i}, n = \frac{kx_{ij} - ix_{kl}}{k - i}.$$

Teniendo los valores iniciales de la ecuación iteramos por el conjunto de puntos buscando aquellos que se ajusten a la recta. Un punto  $(x_z, y_z)$  perteneciente al conjunto de puntos de XDI se considera perteneciente a la recta si cumple que

$$\frac{|ax_z + y_z + b|}{\sqrt{x_z^2 + y_z^2}} \leq \epsilon \quad (2.4)$$

Si la cardinalidad del conjunto formado por los puntos seleccionados como ajustados está en el rango predeterminado se procede a calcular la trayectoria parabólica en YDI, sino se seleccionan otros puntos.

Para la selección de los tres puntos iniciales que representan la ecuación de la parábola se tienen en cuenta solo los que fueron ajustados a la recta en el paso anterior. Seleccionados los tres puntos se ajusta la ecuación parabólica sucesivamente a los puntos pertenecientes al conjunto de la recta. Si ambas funciones tienen una cardinalidad igual, se considera como posible sino se repite el proceso desde el inicio. En la (figura 2.9) se muestra el resultado final de la selección de la recta y la parábola que describen la trayectoria real del lanzamiento

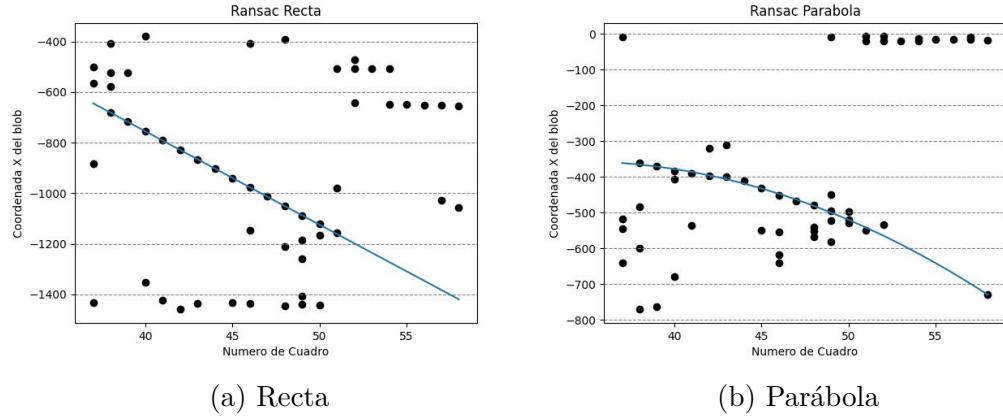


Figura 2.9: Representación de las funciones que describen el lanzamiento

### 2.3.5. Obtención de la trayectoria y candidatos perdidos

Al terminar el proceso se obtiene dos conjuntos de puntos  $XDI$  y  $YDI$ , de igual cardinalidad que comparten el valor de las abscisas respectivamente, así como los parámetros para la ecuación de la recta y de la parábola. Sean los puntos  $X_1(i, j) \in XDI$  y  $X_2(i, k) \in YDI$  que comparten el mismo cuadro en X-Distribución y en Y-Distribución, entonces el punto en la trayectoria original tendría las coordenadas  $X_p(j, k)$ . De esta manera para cada punto de los conjuntos se obtiene la trayectoria definitiva (figura 2.10).

En caso de no existir algún cuadro donde aparezca un punto candidato, al tener las ecuaciones de la recta y la parábola se pueden obtener las coordenadas  $x$  y  $y$  en ese cuadro mediante interpolación lineal o cuadrática respectivamente.

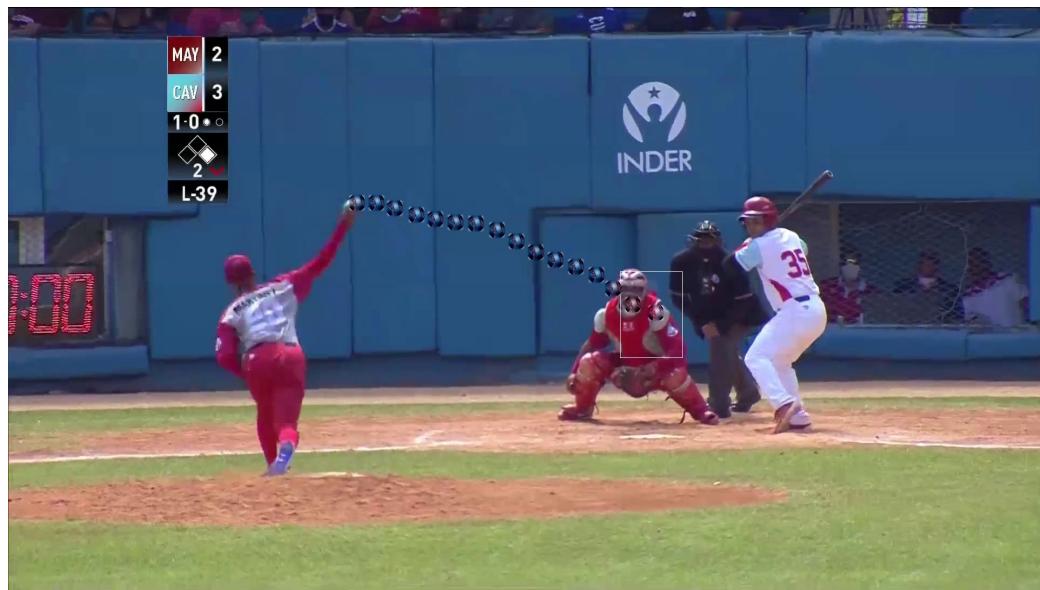


Figura 2.10: Trayectoria del Lanzamiento

# Capítulo 3

## Determinación de la Zona de Strike

Al tener la trayectoria de la bola el siguiente paso es detectar la ZdS que se presentará como un rectángulo que ubicado sobre la cara anterior del home plate (la base home). Para saber donde se ubicarán exactamente las cuatro esquinas del rectángulo en cada cuadro de la secuencia es necesario llevar a cabo varios procesos que serán explicados en los siguientes epígrafes.

### 3.1. Detección del Home Plate

El primer paso consiste en la detección de la base home en los cuadros de la SMV para trazar las líneas verticales que delimitan la ZdS. Se asume que el home plate es de color blanco y al menos parcialmente visible y cercano a los pies del bateador. Para resaltar dicha zona se experimentó primeramente con varios filtros de substracción de fondo como los mencionados en el capítulo anterior pero se llegó a la conclusión de que estos no eran efectivos para la detección. Se optó por la siguiente técnica: se lleva la imagen a escala de grises (para tener una mejor percepción de la intensidad de cada píxel) y se aplica una técnica de umbral. Es decir, dado un umbral el valor de intensidad de un píxel se lleva a los valores 0 o 255. Es de 255 cuando el píxel en la escala de grises tiene intensidad mayor al umbral fijado. En la figura 3.1a se observa un ejemplo de un cuadro original y en la figura 3.1b se muestra el cuadro luego de ser filtrado. La función que se utiliza es `threshold` de la librería cv2 (nombre del wrapper de OpenCV para Python).

El home plate se muestra en forma de segmento de línea horizontal corta de color cercano al blanco y se encuentra alrededor del centro de la imagen. Basado en estas propiedades visuales se realiza el proceso de detección de home expuesto en [H.-T. Chen, Tsai y col. 2009]. Se utiliza un umbral de intensidad alto para que otros píxeles con intensidad media no interfieran al identificarlo.

Teniendo los píxeles de mayor intensidad que delimitan el home plate se quiere



Figura 3.1: Cuadro de video antes y después de ser binarizado

encontrar su parte frontal la cual en la imagen será el segmento del contorno de home que más abajo se encuentra. En un principio el ancho del home plate se establecía con un valor predeterminado pero se llegó a la conclusión de que, al ser este un aspecto importante para el alcance del software y la variabilidad del emplazamiento de las cámaras en diferentes estadios del país, es mejor estimarlo. Aunque no se puede determinar este ancho en la imagen con exactitud sin referencias extra, sí se puede estimar proporcionalmente a partir del diámetro de la bola. Por ello el primer paso consiste en estimar la longitud de la bola así como la posición del bateador.

### 3.1.1. Determinación de ancho de home plate

Se conoce que la bola tiene un diámetro de 2.9 pulgadas y la cara frontal de home tiene una longitud de 17 pulgadas según [FCB 2020]. La idea es determinar el diámetro de la bola en píxeles al pasar por encima del home plate y con ello se podrá establecer una proporción y saber la longitud de la cara frontal del mismo.

Para saber el diámetro de la bola cuando pasa por encima de home se identificaban en un principio posibles círculos en cada uno de los cuadros de video que son procesados al detectar el movimiento en la caja de bateo. Para detectarlos primeramente a la imagen en escala de grises se le aplica el filtro `GaussianBlur` de suavizar los bordes y exista mayor probabilidad de delimitar objetos. Luego con la función `HoughCircles` de la librería `cv2` se detectan los posibles círculos en la imagen suavizada. Un ejemplo de la detección se muestra en la figura 3.2. En caso de que se encuentre algún posible círculo se debe verificar que este es la bola de béisbol, para eso se analiza la intensidad de color en algunos puntos del candidato.

Este proceso tiene el objetivo de determinar el diámetro de la bola independientemente del proceso de determinación de la trayectoria para acortar el tiempo de ejecución de la aplicación , pero dada la poca efectividad del mismo no se ejecuta en



Figura 3.2: Detección de la bola en uno de los cuadros de la secuencia analizada

la aplicación y se decide esperar por el diámetro de la bola resultante de la detección de la trayectoria.

### 3.1.2. Determinación de movimiento en Caja de bateo

Para detectar la caja de bateo donde se encuentra el bateador se parte de la idea de identificar el punto más alto del mismo durante la acción de batear. Este punto generalmente es el bate por la postura natural que adoptan los bateadores. Para ello se analizan cada uno de los cuadros del lanzamiento, desde que la bola es lanzada hasta que es recibida por el receptor.

Los cuadros ya se encuentran en escala de grises lo cual facilita la detección de movimientos. Tomando el primer cuadro como referencia, se compara con cada uno de los otros restantes y se guarda la diferencia absoluta entre la intensidad de dos píxeles en la misma posición en ambos cuadros. Luego se binariza dicha imagen de forma tal que si el valor de intensidad del píxel es mayor que 160 entonces se le asigna el nuevo valor de 255, caso contrario se le asigna el valor de 0. Esta técnica de umbral es realizada con la función `threshold` de la biblioteca `cv2`. Este proceso tiene el objetivo de que sean resaltados los píxeles donde haya una diferencia considerable de intensidad.

Luego entre los píxeles en movimiento encontrados se elige el píxel que más arriba se encuentre en la imagen y este será el píxel donde existe movimiento en la imagen a mayor altura. A partir de dicho píxel se determina un rango estimado de donde se encuentra el bateador que nos servirá para establecer su cajón de bateo y también para descartar algunos candidatos a home plate. Este proceso es efectivo porque como el bateador está en un primer plano su silueta tiende a tener mucha más intensidad de color que el fondo de la imagen .

### 3.1.3. Detección de candidatos

Partiendo de la imagen binarizada entonces se procede a buscar posibles candidatos a home plate. Para encontrar los candidatos se buscan todos los posibles segmentos que se pueden formar en la imagen con píxeles con intensidad de 255 y con una pendiente entre 0.03 y -0.03 y con una longitud entre el mínimo y máximo estimado anteriormente. Para buscar dichos candidatos se utiliza la función `HoughLinesP` de la librería `cv2`. Un ejemplo de identificación de posibles candidatos se muestra en la figura 3.3.



Figura 3.3: Detección de posibles candidatos a Home plate

### 3.1.4. Identificación entre candidatos

Para la detección se tiene un problema, los rectángulos que se dibujan en ambos costados de home para delimitar las cajas de bateo son creados con cal por lo que los píxeles que los identifican pueden tener un valor de intensidad igual o superior a la

de los píxeles del home plate. Estas líneas a veces parciales podrían interferir en la identificación de posibles candidatos a segmento de home plate. Por ello uno de los procedimientos previos a la identificación de home fue identificar el píxel más alto en movimiento o lo que es lo mismo el píxel del bate que permite localizar al bateador.

Teniendo la posición del bateador y la trayectoria de la bola en el lanzamiento, obtenida con las técnicas explicadas en el capítulo precedente, entonces se puede determinar si el bateador está en el cajón a la derecha o a la izquierda del home plate (es derecho o zurdo). Para determinar si es derecho o zurdo solo se comprueba si la bola en algún punto de su trayectoria se encuentra más a la derecha que el bateador. En caso de que así sea el bateador es zurdo, en otro caso el bateador es derecho.

Conociendo si el bateador es zurdo o derecho se pueden descartar primeramente candidatos a home que estén por detrás del bateador, demasiado lejos o demasiado cerca. Si hay un solo candidato entonces es el home. En caso de que haya más de un candidato el home será el que tenga más abajo en la imagen uno de sus puntos extremos (se asume que no existe un píxel candidato que se encuentre por debajo de uno de los frontales del home). Un ejemplo de la identificación de home se observa en la figura 3.4.



Figura 3.4: Identificación de home mostrada en cuadro en escala de grises

En caso que no queden candidatos a home después de este proceso se comienza el proceso de detección reduciendo el umbral de intensidad al binarizar la imagen. Este procedimiento se repite hasta que se obtenga una detección o hasta llegar a un valor de intensidad que permite asegurar que no existe buena visibilidad del home plate en la SMV bajo análisis.

### 3.2. Detección de caja de bateo

La caja (o cajón) de bateo solo se puede detectar si fue detectado el home plate. En caso de que así sea entonces se conoce, según [FCB 2020], que la caja de bateo mide 48 pulgadas de ancho y cada una se encuentra a 6 pulgadas de home. Al conocer el ancho en píxeles del home plate y cuantas pulgadas mide este en la realidad podemos saber a cuantos píxeles equivale una pulgada. Por lo tanto se puede estimar con certeza la ubicación exacta de cada uno de los cajones de bateo así como el ancho en píxeles de cada uno (vea [H.-T. Chen, Tsai y col. 2009]).

Como tenemos dos puntos de la línea frontal de home podemos determinar su pendiente mediante la ecuación

$$m = \frac{y_2 - y_1}{x_2 - x_1}. \quad (3.1)$$

De acuerdo a la pendiente hallada y conociendo a cuantos píxeles equivale una pulgada hallamos, para cada uno de los dos cajones de bateo, sus límites izquierdo y derecho. El objetivo de conocer los límites de los cajones es minimizar el espacio a analizar a la hora de detectar la silueta del bateador porque obligatoriamente el bateador debe estar completamente en uno de ambos cajones dependiendo si batea a la derecha o a la zurda. En la figura 3.5 se muestra la detección de las cajas de bateo para uno de los ejemplos analizados.



Figura 3.5: Cuadro que muestra la determinación de las cajas de bateo

### 3.3. Detección de contorno del bateador

Una vez determinada la línea frontal del home plate, que permite determinar las líneas verticales de la ZdS, lo siguiente es detectar la silueta del bateador para deter-

minar las líneas horizontales de la ZdS. El límite inferior de la ZdS (línea inferior) está determinado por las rodillas y el superior (linea horizontal superior) está determinado por el punto medio entre las caderas y los hombros como se explica en [FCB 2020] (vea 1). La postura del bateador hace que mediante su silueta sean detectables estos puntos.

Se implementaron dos formas de reconocer la silueta del bateador: movimiento de píxeles y cambio de intensidad de colores. Ambos se explican a continuación.

### 3.3.1. Movimiento de píxeles

A partir de la imagen de píxeles en movimiento, obtenida previamente, se crea la de aquellos más a la derecha en la caja de bateo que denotaremos por RMP. La coordenada  $(x, x_i)$ , de cada RMP forma una función discreta unidimensional  $R(i) = x_i$ , donde  $i$  es el índice de cada RMP a lo largo de la dirección y.

Se necesita continuidad espacial e ignorar los fragmentos. Para ello se utiliza el filtro de Savitzky-Golay (Schafer 2011) para suavizar los valores de la función  $R(i)$  y obtener la  $R_s(i)$ .

De manera similar, se crea una imagen de píxel en movimiento más a la izquierda (LMP) a partir de los píxeles en movimiento más a la izquierda (LMP) y la correspondiente función  $L(i)$ . Se aplica el filtro de suavizado y se obtiene  $L_s(i)$  que representa la parte izquierda del contorno del bateador. El proceso completo de detección de silueta se observa en la figura 3.6.

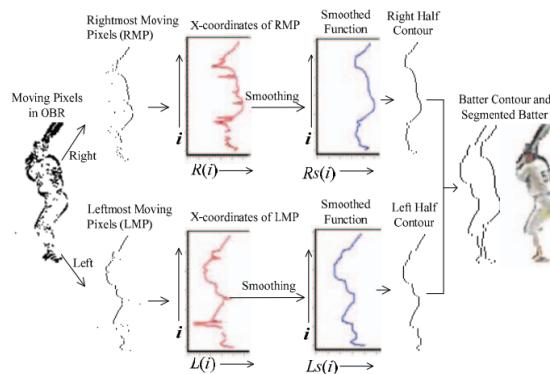


Figura 3.6: Descripción del proceso completo de detección de movimiento

Del bateador solo es necesario determinar uno de sus dos contornos de movimiento, el contorno derecho si el bateador es derecho y el contorno de la izquierda si el bateador batea a la izquierda. No se necesita el contorno delantero porque puede que el receptor

o el arbitro interfieran en la detección de movimiento y tampoco ofrece información relevante que no pueda ofrecer el contorno trasero.

### 3.3.2. Cambio de intensidad de colores

Cuando el uniforme del bateador es blanco la silueta o contorno se puede detectar sin problemas como se explica arriba. Si no es este el caso se realiza una binarización de la imagen mediante un umbral como ya se ha explicado anteriormente (vea la figura 3.7a). Luego se aplica un filtro de detección de bordes sobre la imagen binarizada con la función `findContours` de la biblioteca `cv2` de Opencv y como resultado se obtiene el conjunto de todos los contornos identificados en la imagen como se muestra en la figura 3.7b.

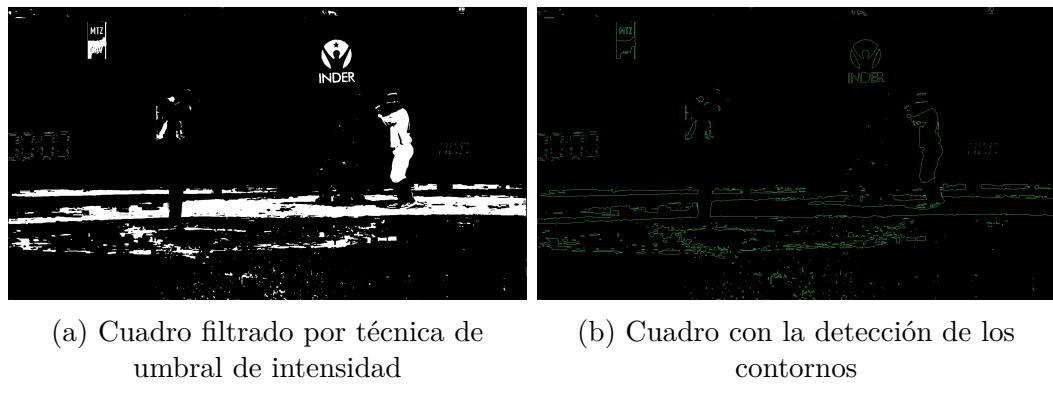


Figura 3.7: Proceso de detección de contornos

Después de identificar los contornos se determina cual es la silueta del bateador. Para ello se tiene en cuenta su ubicación dentro de la caja de bateo detectada determinada de acuerdo a la técnica explicada arriba (vea la figura 3.8).

La forma de detectar la silueta del bateador mediante intensidad de colores tiene la ventaja de la precisión de la silueta que se logra, sin embargo, esta técnica depende de la intensidad de los colores del uniforme del bateador y en los casos en que el uniforme tenga colores de poca intensidad o que los pantalones sean de un color y la camisa de otra es muy difícil identificar la silueta. Por lo tanto existen casos en los que no es funcional. En cuanto a la detección por movimiento de píxeles la precisión de la silueta es alta aunque no tanto como la de intensidad de colores pero funciona para la mayoría de clips analizados y por eso la consideramos fundamental. Por ello fue este último método el elegido para la detección de la silueta en el software.



Figura 3.8: identificación del contorno del bateador

### 3.4. Localización de puntos claves del bateador

Como se aborda en [H.-T. Chen, Tsai y col. 2009], para determinar los límites superior e inferior de la ZdS se tienen que ubicar los puntos característicos que son: la cadera, los hombros y rodillas sobre el contorno del bateador. Debido a la cinemática humana estos puntos presentan giros bruscos en el contorno del cuerpo que se extrae de acuerdo al método de la sección anterior. Por lo tanto, los puntos de extremos de curvatura del contorno son buenos candidatos para los puntos característicos.

Consideremos aquí un bateador diestro (se realiza un proceso similar para un bateador siniestro). Se calcula el NMC (mínimo negativo de curvatura) y los puntos de PMC (curvatura máxima positiva) del contorno de la espalda (la mitad derecha). El punto PMC más cercano al punto medio de la altura de la región segmentada se elige como el punto de la cadera. Luego, el punto NMC debajo de punto de la cadera se elige como el punto de la rodilla, como se muestra en la figura 3.9.

Para ubicar los hombros, ya que hay un cambio repentino de color entre la cabeza y el cuerpo del bateador, definimos el punto central estimado de la masa del batedor. Pasando por la masa desde abajo hacia arriba, y se calcule la diferencia de color entre píxeles adyacente. La primera diferencia de color saliente por encima de la cadera se elige como el punto del hombro.

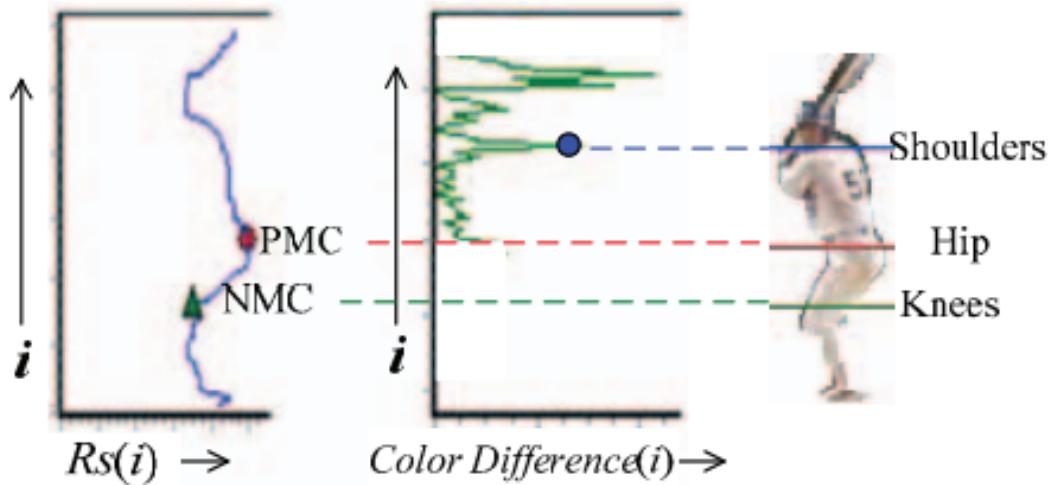


Figura 3.9: Descripción de la detección de hombros , rodillas y caderas

### 3.5. Determinación de puntos de la zona de strike

Teniendo los puntos de rodillas, caderas y hombros y el ancho de home se puede determinar la completamente la ZdS. Las líneas verticales de la ZdS son perpendiculares al home plate en sus puntos extremos y las líneas superior e inferior de la ZdS deben ser paralelas al home plate. La línea inferior la determina el punto de las rodillas y la superior el punto medio entre las caderas y los hombros. Como se tienen los 2 puntos extremos de la línea frontal de home se puede determinar la pendiente de esta mediante la ecuación

$$m = \frac{y_2 - y_1}{x_2 - x_1}. \quad (3.2)$$

Al tener la pendiente podemos determinar  $n$  (de la ecuación  $y = mx + n$ ) sustituyendo las coordenadas de uno de los dos puntos extremos del segmento en la ecuación de la recta. De esta forma se tiene la ecuación de la recta.

Luego se necesitan obtener las 2 rectas perpendiculares a los puntos extremos de home. Teniendo en cuenta que el producto de las pendientes de dos rectas perpendiculares es  $-1$  y que se tiene la pendiente del home plate entonces se puede obtener la pendiente de ambas rectas verticales. Con la pendiente de ambas rectas se puede obtener el valor de  $n$  correspondiente a cada una de ellas sustituyendo en la ecuación de la recta con las coordenadas de uno de los puntos que pertenecen a ellas, los puntos extremos de la línea frontal de home respectivamente.

Dado que dos líneas paralelas tienen igual pendiente entonces las líneas límite inferior y superior de la ZdS tienen la misma pendiente que la recta frontal del home

plate. El valor de  $n$  de la recta inferior se encuentra evaluando el punto de las rodillas y para obtener la  $n$  del límite superior se repite el procedimiento evaluando la función de la recta en un punto que se encuentra en una posición intermedia entre la coordenada  $y$  del punto de las caderas y la coordenada  $y$  de el punto de los hombros.

Los 2 puntos superiores del rectángulo los determinan la intersecciones de la línea horizontal superior con cada una de las 2 perpendiculares a home. Así mismo ocurre con los puntos inferiores del rectángulo, están determinados por la intersección de la línea horizontal inferior con las dos rectas perpendiculares a home. El resultado de determinar la ZdS para una de las SMV analizadas se muestra en la figura 3.10 .



Figura 3.10: Zona de strike

## Capítulo 4

# Detalles de Implementación y Experimentos

La implementación de este software se desarrolló en el lenguaje de programación Python. El entorno de desarrollo escogido para el trabajo fue el editor de código Visual Studio Code, versión 1.73.1 . Además, se utilizó la biblioteca especializada en visión por computadora OpenCV2, versión 4.5.5.64, la más empleada por desarrolladores de software para problemas de este ámbito.

Python es un lenguaje de propósito general, lo cual significa que puede ser usado en distintos tipo de aplicaciones. Es un lenguaje multiplataforma o sea que puede ejecutarse en tanto en Windows, como en GNU/Linux y en Mac OS X. Existen un gran número de bibliotecas especializadas desarrolladas en este lenguaje y muchas otras presentan wrappers para Python.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision según sus siglas en inglés. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en una gran cantidad de aplicaciones, y hasta 2022 se la sigue mencionando como la biblioteca más popular de visión computacional. La detección de movimiento, el reconocimiento de objetos, la reconstrucción 3D a partir de imágenes, son sólo algunos ejemplos de funcionalidades existentes en OpenCV.

Su popularidad se debe a que es:

- libre, publicada bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación,
- multiplataforma, para los sistemas operativos GNU/Linux, Mac OS X, Windows y Android, y para diversas arquitecturas de hardware como x86, x64 (PC), ARM (celulares y Raspberry Pi)

- bien documentada: la organización tiene una preocupación activa de mantener la documentación de referencia para desarrolladores lo más completa y actualizada posible, ejemplos de uso de sus funciones y tutoriales accesibles al público no iniciado en visión artificial, además de difundir y fomentar libros y sitios de formación.

A continuación explicaremos brevemente el funcionamiento de las funciones de OpenCV utilizadas durante el desarrollo de nuestro software. Se puede encontrar más información de esta biblioteca en [Mistry y Saluja 2016].

- **VideoCapture:** Permite trabajar con secuencias de video ya sea mediante la captura en vivo a través de una cámara o mediante un archivo de video,
- **imwrite:** Se utiliza para guardar una imagen en cualquier dispositivo de almacenamiento de acuerdo con un formato especificado,
- **imread:** Carga una imagen del archivo especificado. Si la imagen no se puede leer (debido a la falta de archivo, permisos incorrectos, formato no compatible o no válido), este método devuelve una matriz vacía,
- **cvtColor:** El método se utiliza para convertir una imagen de un espacio de color a otro. Hay más de 150 métodos de conversión de espacios de color disponibles en OpenCV,
- **imshow:** método que se utiliza para mostrar una imagen en una ventana. La ventana se ajusta automáticamente al tamaño de la imagen,
- **line:** método que se utiliza para dibujar una línea en cualquier imagen,
- **medianBlur:** Método de utilizado para la eliminación de ruido en imágenes,
- **threshold:** La función generalmente se usa para obtener una imagen de dos niveles (binaria) a partir de una imagen en escala de grises,
- **SimpleBlobDetector:** La función integrada en la biblioteca propone un algoritmo de detección de blobs (manchas) en una imagen.

## 4.1. Detalles de la aplicación

La aplicación se divide en dos funcionalidades fundamentales, la detección de trayectoria de la bola y la detección de la ZdS. A continuación cada funcionalidad se presenta en varias subsecciones.

### 4.1.1. Algoritmo de reconocimiento de trayectorias

La obtención de la trayectoria se divide en varios procesos necesarios para obtener de una secuencia de video original, otra con los píxeles de la posición de la bola resaltados. Se implementan una serie de clases las cuales son independientes y se encargan de cada funcionalidad.

#### ImageProcessor

La clase tiene como objetivo fundamental procesar las imágenes de entrada, mediante la aplicación de filtros de suavizados y sustracción de fondo. Presenta dos métodos principales.

- **Softened(self,frame)** :El metodo recibe como entrada un cuadro del video y ejecuta la función `medianBlur`, dando como resultado una imagen con el ruido eliminado como salida
- **BackSubstraccion(self,frame)**: Recibe como entrada la imagen sin ruido y le aplica el algoritmo MOG de sustracción de fondo, da como salida una imagen en escala de grises, donde solo se destacan los objetos en movimiento.

#### Blob Detector

La clase es la encargada del reconocimiento de manchas sobre cada imagen. Trabaja sobre los cuadros del video devolviendo como información tres listas, una con las coordenadas  $x$  de las manchas encontradas, otra con el valor de las coordenadas  $x$  y por último una con el número de cada cuadro. La clase se inicializa con los valores de los filtros comentados en la sección 2.1.2. Esta contiene dos métodos que se complementan en la obtención del resultado

- **FrameDetector(self,frame)**: El método recibe como entrada un cuadro del video y se auxilia de la función `SimpleBlobDetector`, dando como salida dos listas con las coordenadas X Y de las manchas detectadas en esa imagen
- **VideoDetector(self,video)**: Tiene como entrada el clip de video del lanzamiento, la función se encarga de separarlos en cuadros enumerados y de aplicar el método `FrameDetector` sobre cada uno. Tiene como salida las listas con las coordenadas y con los números de cuadro.

#### BlobFilter

La clase trabaja sobre las coordenadas de las manchas (blobs) así como las distribuciones XDI y YDI. Es la encargada de filtrar los puntos que cumplen con ciertas

características las cuales los hacen no pertenecer a la solución. La clase disminuye la cardinalidad de los conjuntos de manchas detectadas para mejorar la complejidad temporal de los próximos algoritmos.

### Trajectory

La clase se inicializa con las coordenadas de las manchas y los números de cuadros. Tiene como función principal obtener la lista con la trayectoria de la bola. La clase se encarga de analizar la X-Distribución y la Y-Distribución y de la aplicación del algoritmo RANSAC, así como la obtención de la recta y la parábola descritas en la sección 2.3. Determina conjuntos de iguales cardinalidad y obtiene la trayectoria final utilizando estos, así como en el cuadro donde aparecen. La clase incluye varias funciones.

- **DetectLine(self ,frame,x)** : El método se encarga mediante el algoritmo RANSAC de seleccionar dos puntos para generar la ecuación de una recta. Luego analiza los restante buscando coincidencias para dar la lista como salida
- **DetectParab(self,frame,y)** Método similar al anterior con el cambio que se trabaja solo con los puntos coincidentes con la recta
- **DetectTrajectory(self,frame,x,y)** Utiliza como entrada las coordenadas de los píxeles que son considerados bolas, después del proceso de la detección de la recta y la parábola y también los cuadros donde se generan. La función da como salida un diccionario donde las llaves son los cuadros y los valores las coordenadas ( $x,y$ ) de la mancha.

### VideoTrajectory

Luego de obtener un diccionario donde se describe en que cuadro aparece la bola de beisbol, así como donde, quedaría sustituir en el cuadro la imagen de una bola en dichos puntos. La clase se encarga de este proceso, recibe el video como entrada y al coincidir los cuadros con alguna de las llaves del diccionario genera un nuevo cuadro (imagen) donde aparece la bola añadida en la ubicación deseada. La clase genera un nuevo video con la secuencia del lanzamiento resaltada en forma de cometa.

#### 4.1.2. Algoritmo de determinación de zona de strike

Para la determinación de la ZdS es necesaria la realización de varios procesos.

## Detección de home

Para la detección de home se utilizan varias funciones :

- **convertir\_de\_video\_a\_cuadros**: Su función es leer un video y guardarla en forma de cuadros de video en una ruta específica.
- **detection\_zona\_de\_strike**: Es la función principal para la detección de home y se encarga de llamar a la función correspondiente para cada fase de la detección.
- **blackOrwhite**: Es la función encargada de binarizar una imagen en escala de grises mediante una técnica de umbral de intensidad utilizando la función threshold de la librería cv2 de Opencv.
- **identificarCandidatos**: Su función es determinar todos los posibles candidatos a línea frontal de home teniendo en cuenta para cada segmento candidato que debe cumplir con tener pendiente entre -0.3 y 0.3 y una longitud determinada .
- **detectar\_X\_Bate**: Su función es determinar las coordenadas del píxel más alto en movimiento.Para ello se analizan los cuadros de video desde que la bola es lanzada hasta que llega al receptor.
- **verifica\_comienzo\_de\_lanzamiento**: Esta función tiene como objetivo detectar círculos e identificar si es la bola de béisbol.
- **es\_derecho**: Es la función encargada de determinar si el bateador es derecho o zurdo conociendo cual es el píxel más alto en movimiento teniendo las coordenadas de la bola en la secuencia del lanzamiento.
- **determinar\_cajas\_de\_bateo**: Es la función encargada de determinar los límites de las cajas de bateo sabiendo la ubicación de home.
- **return\_diametro**: Retorna el diámetro de la bola teniendo el radio de esta.

## Determinación de la silueta del bateador

Para determinar la silueta del bateador se utiliza la función:

- **detectar\_funcion\_silueta**: Es la encargada de detectar la silueta del bateador. Primeramente se buscan los píxeles en movimiento a la derecha si el bateador es derecho o a la izquierda si el bateador es zurdo. Dichos píxeles forman una función que se suaviza ya que pueden detectarse falsos píxeles en movimiento. Para suavizarla se utilizó la función **savgol\_filter** (filtro de Savitzky-Golay)

de la librería `scipy`. Luego se buscan las rodillas, las caderas y los hombros del bateador. Para ello se detecta la cadera como el punto más a la izquierda de la función suavizada si el bateador es zurdo, o más a la derecha si el bateador es derecho. A partir del punto de la cadera buscamos los puntos de las rodillas y los hombros.

## 4.2. Experimentos y Resultados

La presente sección muestra los experimentos realizados con el fin de evaluar el software propuesto y sus resultados correspondientes. Se explica cada fase del proceso y se reflejan las consideraciones para la elección de un tipo de técnica sobre otra. El objetivo principal de la sección es obtener en que porcentaje la aplicación es funcional.

### 4.2.1. Conjunto de pruebas

El software está previsto para ser utilizado en transmisiones televisas de béisbol a lo largo del país. Con este fin se recopilaron grabaciones de 15 juegos diferentes, tanto de la Serie Nacional de Béisbol como de la Liga Élite. El conjunto presenta una gran variedad ya que se toman partidos en el horario nocturno y en el diurno, se escogieron diferentes equipos que presentan una indumentaria que difiere bastante y la presentan en diferentes estadios. Se tienen como sedes principales: el Estadio Latinoamericano parque donde realiza sus partidos como local el equipo de Industriales, el José Ramón Cepero perteneciente a la provincia de Ciego de Ávila y el Guillermón Moncada de la provincia de Santiago de Cuba.

De cada juego solo interesan los fragmentos de filmación donde el lanzador ejecuta un lanzamiento hacia el home plate y la bola es recibida por el receptor sin que el bateador ejecute un “swing”, por lo que el video total de los partidos se divide en clip de videos de a lo sumo 10 segundos. Otra consideración importante y que afecta directamente la ejecución del algoritmo es la variedad de resolución que existen entre los videos, algunos son tomados de transmisiones del canal alta definición y otros del canal Tele Rebelde. Por último para comparar nuestro programa con alguno ya existentes se añadieron al conjunto de prueba 3 videos de transmisiones de partidos de la MLB.

### 4.2.2. Experimentación

Como primer experimento (Tabla 4.1 ) se realiza una comparativa en los distintos algoritmos de sustracción de fondo abordados en el artículo. Se selecciona una muestra del conjunto total para ser analizada, resaltando que se tomaron videos de diferentes definiciones. El objetivo es valorar las técnicas en cuanto a la cantidad de candidatos

encontrados. No es conveniente que la cardinalidad del conjunto de candidatos sea demasiado alta ya que pueden contaminar la búsqueda de la solución pero tampoco debe quedar muy baja debido a la posibilidad de desechar bolas potenciales.

Tabla 4.1: Blob detectado en técnicas de sustracción de fondo

resolución	Cant Videos	MOG	D-Frame	VIBE
720	69	1426	1683	5754
1920	80	2986	3668	9136
1280	3	76	74	69

Los resultados mostrados en la tabla 4.1 muestran como el algoritmo MOG se mantiene estable en el análisis las diversas resoluciones, dando como resultado una cantidad de candidatos satisfactoria para le desarrollo del algoritmo. Otro experimento interesante es la evaluación del algoritmo para determinar cuanto difiere (distancia euclíadiana) la posición real de la bola en el cuadro con la detectada por el algoritmo. Primeramente se tomaron como base 7 videos de diferentes horarios, estadios y equipos. Se analizó manualmente las coordenadas de las bolas en las trayectorias de cada lanzamiento. Esto permitió una comparativa entre la posición real y la posición prevista por el algoritmo que se muestra en la tabla 4.2. Se aprecia un resultado favorable en la aplicación del algoritmo. La cantidad de bolas detectadas es mayor que las generadas permitiendo así que la desviación real sea menor y la medición del algoritmo sea mas acertada.

Tabla 4.2: Evaluación de la generación de bolas

Secuencia	Bolas Reales	Bolas Detectadas	Bolas Generadas	Dif-máxima	Dif-acumuladas
1	16	12	4	1.90	2.27
2	17	13	4	2.25	3.60
3	16	15	1	0.78	0.78
4	14	10	4	1.43	3.29
5	13	13	0	0	0
6	12	10	0	0	0
7	13	12	1	0.95	0,95

#### 4.2.3. Errores comunes en detección de trayectoria de la bola

Como se apreció en el apartado anterior el software no detecta la trayectoria exacta de la bola en el 100 por ciento de los videos. El algoritmo puede presentar fallas

debido a varios factores, pero algunos son más influyentes que otros. A continuación se describen los más comunes

- **Suciedad de la pelota:** La bola de béisbol es un objeto blanco y brillante sobresaliente en las imágenes, pero el transcurso del juego puede hacer que esta se ensucie perdiendo su color original y afectando a la detección 4.1.
- **La pelota se pierde en el uniforme:** Debido al ángulo de las cámaras, existe el momento en que la bola se superpone a la imagen del bateador, con los bateadores zurdos principalmente, y si el uniforme es blanco no se detecta 4.2.
- **Movimiento de la cámara:** Nuestros videos son tomados con la cámara del jardín central, una cámara que no es estática y que depende de un operario. Varios factores como el viento o la mala manipulación pueden hacer que la cámara se mueva de un lado a otro o de arriba abajo y esto hace que las coordenadas de los píxeles cambien y el algoritmo realice una mala estimación.
- **Mala calidad de los videos:** Aunque no es la mayoría, algunas transmisiones son realizadas por las emisoras locales y las imágenes son demasiado borrosas, haciendo que el proceso de eliminación de ruido falle, también existen escenas donde hasta la detección de la bola de forma manual es compleja, por lo que el algoritmo no arroja un buen resultado 4.1.

#### 4.2.4. Valoración de los resultados de la detección de la trayectoria

El software propuesto presenta un buen porcentaje de acierto en la detección de la trayectoria original del lanzamiento. En los cuadros donde no se obtienen la posición real de la bola, el algoritmo de estimación desarrollado da unas coordenadas bastante confiables, suministrando así conjuntos de videos con trayectoria resaltada satisfactorios. Otro análisis favorable es la corrección del error cuando la bola se superpone con el bateador. El desarrollo de los experimentos refleja una superioridad en cuanto a efectividad en videos tomados en parques donde la cámara presenta buen ángulo y poco movimiento, así como los de transmisiones en alta definición ya que se visualizan con mayor claridad los objetos en cada cuadro. Al comparar con programas desarrollados en otras partes del mundo, la posición final de la bola que muestra el software que se presenta en este documento está muy cercana a la de estos productos (que han sido probados extensamente, emplean varias cámaras dedicadas exclusivamente al análisis y se ejecutan con equipos de cómputo de altas prestaciones), dando una buena idea de la precisión obtenida

#### 4.2.5. Errores comunes en determinación de la zona de strike

La determinación de la Zona de Strike tampoco es posible en el ciento por ciento de los videos que conforman la colección de pruebas. El proceso puede fallar debido a problemas con la calidad de la imagen así como por situaciones que nuestro algoritmo no es capaz de controlar. Algunas de las fallas más comunes son:

- **Poca visualización de home:** Uno de los procesos que se realizan es determinar la ubicación de home. Si este no tiene visibilidad en la imagen por tener tierra encima (como se muestra en la figura 4.3) es muy difícil que sea detectado.
- **Movimiento de la cámara para determinar la posición del bateador:** Para determinar si en un píxel hubo movimiento debe haber un cierto cambio de intensidad en el mismo .Los píxeles de color blanco son los más intensos en la escala de grises y si la cámara se mueve puede ocurrir que se genere un falso positivo, como se observa en la figura 4.4.
- **Movimiento de la cámara:** El home plate es determinado fundamentalmente a partir de un cuadro del video, si hay movimiento de la cámara entonces el home plate aparece en diversas ubicaciones en el clip como se observa en la figura 4.5.
- **Mala calidad de los videos:** La mala calidad de un video puede hacer que home no tenga la suficiente visibilidad como se evidencia en la figura 4.1.

#### 4.2.6. Valoración de los resultados de la determinación de la zona de strike

Se escogió una variada muestra del conjunto de 100 videos de prueba y se evaluó con estos la exactitud de cada algoritmo implementado así como la determinación de la Zona de Strike.

##### Análisis de la determinación de home

Tabla 4.3: Resultados de la determinación de la posición del bateador

Videos	número de secuencias	Correctas	Incorrectas
MLB	2	2	0
Serie Nacional (Jose Ramón Cepero)	85	82	3
Serie Nacional(Latinoamericano)	13	13	0

Tabla 4.4: Resultados de la determinación de si el bateador es zurdo o derecho

Videos	número de secuencias	Correctas	Incorrectas
MLB	2	2	0
Serie Nacional (Jose Ramón Cepero)	85	85	0
Serie Nacional(Latinoamericano)	13	13	0

La determinación de home se divide en dos procesos. El proceso de determinación de la posición del bateador el cual como se observa en la tabla 4.3 tiene un muy buen porcentaje de exactitud (un 97 por ciento) y el proceso de determinar si el bateador es derecho o zurdo, algoritmo que funciona en el ciento por ciento de los videos de la muestra como se observa en la tabla 4.4.

Los 3 casos de la muestra de videos en los que la determinación de la posición del bateador es incorrecta fallan por detectar un píxel blanco como falso píxel en movimiento por la inestabilidad de la cámara.

Tabla 4.5: Resultados de la determinación de home

Videos	número de secuencias	Correctas	Incorrectas
MLB	2	2	0
Serie Nacional (Jose Ramón Cepero)	85	70	15
Serie Nacional(Latinoamericano)	13	12	1

Como se observa en los resultados estadísticos de la determinación de home en la tabla 4.5, el algoritmo tiene una precisión de un 84 por ciento, falla en 16 casos, entre los que:

- No fue detectada correctamente la posición del bateador en 3 de las secuencias de video.
- Hay demasiada intensidad por la iluminación del sol y esto hace que se detecten segmentos falsos candidatos a home plate en 12 videos.
- El home plate no tiene una buena visibilidad en uno de los videos .

### Análisis de la determinación de la silueta y puntos claves del bateador

La determinación de la silueta del bateador y puntos de rodillas, caderas y hombros por detección de movimiento funciona para todos los casos analizados como se observa en la tabla 4.6.

Tabla 4.6: Resultados de la determinación de la silueta y puntos de rodillas,caderas y hombros

Videos	secuencias	Correctas	Incorrectas
MLB	2	2	0
Serie Nacional (Jose Ramón Cepero)	85	85	0
Serie Nacional(Latinoamericano)	13	13	0

### Análisis de la determinación de la zona de strike

Como la determinación de la Zona de Strike se divide en la determinación de home plate y de la silueta del bateador junto a sus puntos claves entonces es la determinación de la zona de strike depende del éxito de ambos procesos. Como la determinación de la silueta tuvo un 100 por ciento de efectividad sobre la muestra tomada entonces el éxito de la determinación de la ZdS es igual a la del algoritmo de determinación del home plate. Es entonces correcta para el 84 por ciento de los casos analizados.

### Calidad General del software

Tabla 4.7: Resultados de la determinación de la trayectoria de la bola y Resultado general del software

Videos	secuencias	trayectoria efectiva	trayectoria incorrecta
MLB	2	2	0
Serie Nacional (Jose Ramón Cepero)	85	76	9
Serie Nacional(Latinoamericano)	13	9	4

El seguimiento de la trayectoria de la bola presenta un 87 por ciento de efectividad para la muestra como se observa en la tabla 4.7. El software funciona de forma efectiva cuando es exitosa tanto la determinación de la trayectoria de la bola como la de la ZdS, lo cual ocurre en el 74 por ciento de los casos analizados. Es este un buen resultado teniendo en cuenta la diversidad en la calidad de los clips analizados.

### Evaluación de los objetivos

Cada uno de los objetivos específicos que se propusieron en un principio fue completado con éxito. El objetivo general, de una detección automática sin asistencia, fue completado parcialmente porque el software resultante es semi-asistido ya que requiere al usuario como entrada el número de los cuadros del clip donde comienza

y termina el lanzamiento. Hay forma documentada de resolver este problema y será una recomendación para el desarrollo del software en un futuro.



Figura 4.1: La bola está sucia y la imagen tiene baja calidad



Figura 4.2: Bola perdida en el bateador



Figura 4.3: Muestra poca visibilidad de home



Figura 4.4: Muestra la mala detección de la posición del bateador



(a) Cuadro antes de moverse la cámara    (b) Cuadro luego de moverse la cámara

Figura 4.5: Se muestra el movimiento de la cámara

# Conclusiones

Dado que los lanzamientos son factores determinantes para la victoria o derrota de un equipo en un juego de béisbol, es conveniente para el personal profesional y la audiencia disponer de información adicional y precisa sobre los mismos. El presente trabajo es una intención de proporcionarla sin requerimientos de hardware adicionales a los ya utilizados para las trasmisiones de los partidos por la TV.

El seguimiento de la pelota y la detección de la Zona de Strike en los videos de trasmisiones de béisbol es una tarea desafiantre debido al pequeño tamaño de la pelota, la alta velocidad de esta y la dificultad para apreciar claramente el home plate en los mismos. En este trabajo se logra un seguimiento de la trayectoria utilizando las características físicas de la bola. También se obtiene una representación visual acertada del rectángulo que delimita la zona por donde debe pasar la pelota para ser declarada como strike. La evaluación del lanzamiento se realiza en un tiempo breve lo cual permite emplear nuestro software en las repeticiones entre lanzamiento y lanzamiento usuales en la TV.

El uso de técnicas simples pero correctamente estructuradas y fundamentadas empleadas en el trabajo, así como su validación mediante una diversa y amplia experimentación concluye con un software útil capaz de mejorar la calidad de las transmisiones del evento deportivo de mayor relevancia para Cuba.

# Recomendaciones

Durante el desarrollo de la investigación se determinaron ciertas cuestiones relevantes, que no pudieron abordarse a profundidad y son recomendables para contribuir al perfeccionamiento del software. A continuación algunas de ellas:

- **Detección automática del inicio y final de la trayectoria:** Implementar algoritmos que detecten en que cuadro de la SMV el lanzador suelta la bola y en cual el receptor la captura convertirá el sistema de asistido a automático posibilitando ejecutarlo en directo encima de la transmisión de la televisión nacional,
- **Mejoramiento de la detección de manchas:** La detección de manchas (blobs) se realiza con ayuda de las funciones de OpenCV como se explica anteriormente. Optimizar este proceso con un algoritmo diferente aumenta la cantidad de candidatos y evita la pérdida de posibles soluciones.
- **Implementación de otros algoritmos de reconocimiento de objetos:** Se recomienda la utilización de otras técnicas para la detección de objetos como son el uso de redes neuronales, para aumentar la exactitud de la trayectoria resultante.
- **Optimizar algoritmos de predicción:** La optimización de los algoritmos que calculan las posiciones de bolas perdidas en la trayectoria así como el uso de otras técnicas no aleatorias puede contribuir al aumento de la precisión del software.
- **Estabilización de la imagen:** El movimiento brusco de la cámara al grabar puede provocar que la Zona de Strike cambie de ubicación de cuadro a cuadro. Desarrollar un estabilizador de video aumentaría la precisión del software.
- **Implementación de otros algoritmos de detección de movimiento:** Recomendamos seguir investigando sobre como determinar la posición del bateador, aunque nuestra idea funciona para muchos casos la detección del bateador puede mejorar su precisión.

- **Determinación de Strike o Bola:** Luego de determinar la Zona de Strike y la trayectoria de la bola determinar el momento preciso del paso (o no) de la bola pasa por la ZdS contribuiría a perfeccionar la funcionalidad del software.

# Bibliografía

- Ball-Strike(ABS), A. (2022). *EL SISTEMA ABS CHALLENGE*. Consultado el 12 de octubre de 2022, desde <https://www.mlb.com/news/abs-system-pitch-timer-bigger-bases-tested-triple-a>. (Vid. pág. 6)
- Barbuzza, R., Fernández Esteberena, L., Dominguez, L., Perez, A., Rubiales, A. J., & D'Amato, J. P. (2017). Separación de sombras a los objetos detectados con sustracción de fondo en video. *Congreso Argentino de Ciencias de la Computación*, 23 (vid. pág. 13).
- Bas-Martínez, A. A. (2014). *Seguimiento de una pelota de beisbol en secuencias monoculares de video* (Trabajo de Diploma). Universidad de La Habana. La Habana, Cuba. (Vid. pág. 7).
- Bouwmans, T., Porikli, F., Höferlin, B., & Vacavant, A. (2014). *Background modeling and foreground detection for video surveillance*. CRC press. (Vid. págs. 12, 13).
- Cepero Ruz, R., Cabrera Rodríguez, C., & Armada Herrera, J. (2010). *Implementación de algoritmos de reducción de ruido en las imágenes* (thesis). (Vid. pág. 10).
- Chen, H.-S., Chen, H.-T., Tsai, W.-J., Lee, S.-Y., & Yu, J.-Y. (2007). Pitch-by-pitch extraction from single view baseball video sequences. *2007 IEEE International Conference on Multimedia and Expo*, 1423-1426 (vid. pág. 7).
- Chen, H.-T., Chen, H.-S., Hsiao, M.-H., Tsai, W.-J., Lee, S.-Y., y col. (2008). A trajectory-based ball tracking framework with visual enrichment for broadcast baseball videos. *Journal of Information Science and Engineering*, 24(1), 143-157 (vid. págs. 18, 19).
- Chen, H.-T., Tsai, W.-J., & Lee, S.-Y. (2009). Stance-based strike zone shaping and visualization in broadcast baseball video: Providing reference for pitch location positioning. *2009 IEEE International Conference on Multimedia and Expo*, 302-305. <https://doi.org/10.1109/ICME.2009.5202495> (vid. págs. 7, 24, 29, 32)
- Cheung, S.-C. S., & Kamath, C. (2005). Robust background subtraction with foreground validation for urban traffic video. *EURASIP Journal on Advances in Signal Processing*, 2005(14), 1-11 (vid. pág. 11).

- Chu, W.-T., Wang, C.-W., & Wu, J.-L. (2006). Extraction of baseball trajectory and physics-based validation for single-view baseball video sequences. *2006 IEEE International Conference on Multimedia and Expo*, 1813-1816 (vid. pág. 20).
- Escobar Tafurt, L. F. (2016). Evaluación de algoritmos de sustracción de fondo para conteo de personas (vid. pág. 11).
- Fast, M. (2010). What the heck is PITCHf/x. *The Hardball Times Annual, 2010*, 153-158 (vid. pág. 5).
- FCB. (2020). *Reglas Oficiales del Béisbol* (C. de Reglas y Arbitraje, Ed.). Federación Cubana de Béisbol. (Vid. págs. 2, 25, 29, 30).
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395 (vid. pág. 20).
- Garcia, H. (2021). *Historia de las transmisiones MLB: Primer partido TV-Color*. Consultado el 21 de octubre de 2022, desde <https://momentodeportivord.com/historia-de-las-transmisiones-mlb-primer-partido-tv-color/>. (Vid. pág. 1)
- González-Maqueira, E. (2015). *Detección de la trayectoria de una pelota al ser lanzada por el pitcher* (Trabajo de Diploma). Universidad de La Habana. La Habana, Cuba. (Vid. pág. 7).
- Merényi, D. (s.f.). Zlepení kvality videa z prmyslové kamery (vid. pág. 9).
- Mistry, K., & Saluja, A. (2016). An introduction to opencv using python with ubuntu. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol*, 1(2), 65-68 (vid. pág. 36).
- Núñez, A. (2018). *Strike cantado*. Consultado el 21 de octubre de 2022, desde <https://www.trabajadores.cu/20180131/strike-cantado/>. (Vid. págs. 1, 5)
- Paguay Paguay, A. R., & Urgilés Ortíz, P. R. (2012). *Recuperación de imágenes mediante extracción de blobs aplicando el operador Laplaciano de Gauss y el Kernel Gaussiano y desarrollo de un prototipo* (B.S. thesis). (Vid. pág. 14).
- Pichardo, P. (2022). «Los árbitros tienen que pasar de forma inadvertida sino...» expresó figura histórica del arbitraje cubano. Consultado el 21 de octubre de 2022, desde <https://swingcompleto.com/luis-cesar-valdes-arbitraje-cubano-serie-61-220309/>. (Vid. págs. 1, 5)
- Rivas, M. R. (2003). Sistema de pre-procesamiento de imágenes electrocardiográficas en telemedicina (vid. pág. 10).
- Schafer, R. W. (2011). What is a Savitzky-Golay filter? [lecture notes]. *IEEE Signal processing magazine*, 28(4), 111-117 (vid. pág. 30).
- TRACKMAN. (2022). *TRACKMAN Technical Especifications*. Consultado el 12 de octubre de 2022, desde <https://www.trackman.com/baseball/V3-Game-Tracking/tech-specs>. (Vid. pág. 6)
- TRACKMAN-Function. (2022). *TrackMan Baseball: La tecnología de seguimiento oficial para la fábrica de béisbol*. Consultado el 12 de octubre de 2022, desde <https://www.trackman.com/baseball/V3-Game-Tracking/tech-specs>.

- //www.baseballfactory.com/trackman-baseball-official-tracking-technology-baseball-factory/. (Vid. pág. 6)
- TRACKMAN-LMB. (2022). *LMB: La Liga Mexicana de Beisbol y TrackMan Inc. anuncian una alianza tecnológica de clase mundial*. Consultado el 12 de octubre de 2022, desde <https://www.milb.com/news/lmb-y-trackman-anuncian-alianza-tecnologica-de-clase-mundial>. (Vid. pág. 6)
- Tv-Historias. (2011). *Cuba, 1950: Las primeras imágenes de Béisbol en la Televisión iberoamericana*. Consultado el 21 de octubre de 2022, desde [https://www.tvcubana.icrt.cu/index.php?option=com\\_content&view=article&id=208:cuba-1950-las-primeras-imagenes-de-beisbol-en-la-television-iberoamericana5&catid=97&Itemid=64#:~:text=As%C3%AD%2C%20por%20azar%2C%20finalizando%20la,este%20deporte%20en%20Ibero%20Am%C3%A9rica](https://www.tvcubana.icrt.cu/index.php?option=com_content&view=article&id=208:cuba-1950-las-primeras-imagenes-de-beisbol-en-la-television-iberoamericana5&catid=97&Itemid=64#:~:text=As%C3%AD%2C%20por%20azar%2C%20finalizando%20la,este%20deporte%20en%20Ibero%20Am%C3%A9rica). (Vid. pág. 1)
- Wikipedia. (2022). FFmpeg — Wikipedia, La enciclopedia libre [[Internet; descargado 12-noviembre-2022]]. <https://es.wikipedia.org/w/index.php?title=FFmpeg&oldid=146401048>. (Vid. pág. 9)
- Wikipedia contributors. (2022). Deinterlacing — Wikipedia, The Free Encyclopedia [[Online; accessed 12-November-2022]]. <https://en.wikipedia.org/w/index.php?title=Deinterlacing&oldid=1117259664>. (Vid. pág. 9)
- Zheng, M.-C., & Chen, C.-Y. (2022). Types of Major League Baseball Broadcast Information and Their Impacts on Audience Experience. *Informatics*, 9(4), 82 (vid. pág. 1).