UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

SEGUIMIENTO DE OBJETOS EN VÍDEO A LARGO PLAZO

PROYECTO FIN DE CARRERA

INGENIERÍA DE TELECOMUNICACIÓN

BORJA MAZA VARGAS

JULIO 2016

SEGUIMIENTO DE OBJETOS EN VÍDEO A LARGO PLAZO

Autor: Borja Maza Vargas
Tutor: Juan Carlos San Miguel Avedillo
Ponente: José María Martínez Sánchez

email: {borja.maza@estudiante.uam.es, juancarlos.sanmiguel@uam.es, josem.martinez@uam.es}

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2016

Resumen

En este proyecto se propone un análisis del seguimiento de objetos en secuencias de largo plazo. Recientemente se han incrementado los desarrollos de algoritmos de seguimiento enfocados a vídeos de corta duración. Sin embargo ante la constante evolución de los trackers y del aumento en el tiempo de las secuencias de vídeo en el uso diario, surge la necesidad de estudiar mecanismos que mejoren los ya desarrollados y la creación de nuevos algoritmos destinados para tratar especialmente con secuencias de largo plazo.

El objetivo principal del proyecto ha sido el estudio, diseño y evaluación de un algoritmo que combinase otros trackers desarrollados previamente tanto en secuencias de corto como de largo plazo. Para ello, en primer lugar se ha realizado un estudio del estado del arte en referencia al seguimiento de objetos, con especial atención en el caso de vídeos de largo plazo. Después el trabajo se centra en la selección y descripción de los algoritmos de seguimiento escogidos para evaluar y comparar el conjunto de vídeos de este proyecto. Una vez se han estudiado estos trackers, se avanza al diseño e implementación de un algoritmo de fusión. Esta propuesta pretende examinar el comportamiento de la combinación de algoritmo en el marco del largo plazo.

Finalmente se procede a la evaluación del nuevo algoritmo sobre numerosas secuencias a fin de poder realizar una comparación con los algoritmos individuales. Debido a que las secuencias de largo plazo presentan un gran número de problemas, los resultados obtenidos en su mayoría por todos los trackers son relativamente bajos.

Palabras clave

Seguimiento de objetos en vídeo, frame, ground-truth, algoritmos de seguimiento, bounding box, objeto de interés, largo plazo.

Abstract

In this master thesis an analysis of tracking objects in long-term sequences is proposed. Recently, the development of video tracking algorithms has been focused on short videos. However, the need to operate for long times (e.g. 24/7 video-surveillance) have increased need to study mechanisms to improve and update existing tracking algorithms for their use in long-term sequences.

The main aim of the project is the study, design and evaluation of an algorithm that combines other trackers sequences previously developed both short and long term. For this objective, first it has conducted a study of the state of art related to object tracking, focused on the case of long-term videos. After, this project focuses on the selection and description of the chosen tracking algorithms to evaluate and compare the set of videos of this project. Once these trackers have been studied, a fusion algorithm is implemented which examines the behavior of the combination of algorithm under the long-term framework.

Finally this project proceeds to evaluate the combination-based algorithms on numerous sequences in order to make a comparison with the individual algorithms. Because long-term sequences have a large number of problems, the performance obtained mostly by all trackers are relatively low.

Keywords

Video tracking, frame, ground-truth, tracker, bounding box, target, long-term.

Agradecimientos

A día de hoy recuerdo perfectamente el primer día que entré en la facultad, el día en el que aquellas paredes parecían gigantes dispuestos a derrotarte en cualquier momento. Sin embargo, tras varios años, ahora al verlas creo que se han quedado pequeñas y es el momento de salir a por retos y desafíos más grandes.

En primer lugar, me gustaría agradecer a mi tutor Juan Carlos San Miguel la oportunidad brindada de realizar este proyecto fin de carrera y toda su colaboración, consejos y rápidas respuestas en todo momento en la realización del mismo.
También quisiera dar las gracias a José María Martínez por ayudarme en la decisión de elegir un proyecto del grupo VPULab, en dónde siempre hay alguien dispuesto a echarte una mano con una sonrisa.

Por supuesto, quiero dar las gracias a mis compañeros durante esta etapa en la universidad. Compañeros no sólo de una carrera, sino de mañanas, de tardes, de findes, de bibliotecas, de partidos, de fatigas, de alegrías y de disgustos, pero sobre todo, de risas. Gracias a todos. Aunque muchos se merecen su espacio aquí, sería injusto no hacer una mención especial a Zazo, Foguet, Tolosana y Fran. Siempre pendientes, sin pedir nunca nada a cambio y sobre todo, excelentes personas dentro y fuera de la universidad.

No podría olvidarme de la gente que me rodeó durante mi estancia en tierras nórdicas. Sin lugar a dudas uno de los mejores semestres de mis últimos veinticinco años. Jag skulle vilja tacka alla människor om universitetet, lärare och alla internationellt elever stöd och sympati, särskilt spanjorerna från Madrid och dess omgivningar. Där jag fick upplevelser som jag minns alltid. Tack så mycket Sverige!

Aunque lejos quedan ya los años en el instituto y colegio, por suerte hay muchas personas excepcionales que siguen estando cerca. Comenzando por Dani, Álvaro, Alber, Mikel, Merlo, Clara y finalizando con Jorge y Palazón, sin duda me faltaría espacio en esta memoria para describir lo grandes que sois pareja del 4 de mayo!

Ya que he cogido carrerilla, agradecer a todas las personas del Club Balonmano Alcobendas su dedicación y apoyo. Válvula fundamental de escape todos estos años para cargar las pilas tras los largos días de Teleco.

Me guardo para el final el agradecimiento más personal posible. El de mi familia, es tan grande y somos tan pocos que va para todos, pero sobre todo a mis padres, siempre apoyándome y creyendo en mí cuando no lo hacía ni yo mismo. Siempre anteponiendo mis estudios y demás situaciones antes que las suyas propias. Y por supuesto a mi hermano, capaz de contagiarme su risa por cualquier tontería hasta tener que meternos en habitaciones diferentes para dejar de reír.

Muchas gracias a todos.

Borja Maza Vargas
Julio 2016

Índice general

Índice de figuras

Índice de tablas

Acrónimos

LT Long – Term

GT Ground – Truth

RGB Red, Green, Blue

HOG Histograma de Gradientes Orientados

FDA Frame Detection Accuracy

SFDA Sequence Frame Detection Accuracy

CLE Center Location Error

CTR Correct Track Ratio

LTT Long Term Tracking

TLD Tracking – Learning Detection

GMM Gaussian Mixture Model

SVM Support Vector Machine

LT-FLO Long Term Feature Less Object

MS Mean – Shift

CBWH Corrected Background Weighted Histogram

COT Color Tracker

AFS Active Feature Selection

AEST Active Examples Selection Tracker

MEEM Multiple Experts using Entropy Minimization

MIL Multiple Instance Learning

LapRLS Laplacian Regularized Least Squares

SPLTT Self – Paced Long – Term Tracking

LT-CT Long – Term Correlation Tracking

Capítulo 1

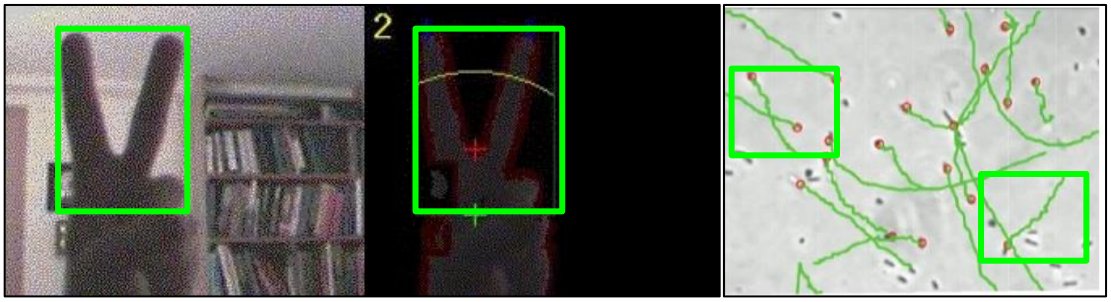
Introducción

1.1. Motivación

En la actualidad, cada vez es más frecuente encontrarse con cámaras de vídeo en cualquier lugar. Cada una de éstas puede tener una aplicación final diferente pero en la gran mayoría domina el uso del seguimiento de objetos en vídeo (*tracking*). Con el paso del tiempo ha ido creciendo el número y los lugares donde se emplean cámaras con este objetivo. Zonas con una gran concentración de personas u objetos, tales como áreas aeroportuarias, estaciones de tren o grandes carreteras son algunos ejemplos. Pero no solamente en el ámbito de seguimiento de vehículos, también se encuentran en centros comerciales, en parkings como medida de vigilancia y seguridad, o en numerosas situaciones médicas, como pueden ser diagnósticos realizados a través del seguimiento de células o moléculas. En este contexto cobra especial interés el análisis automático, que facilita y ayuda a manejar la gran cantidad de información que resulta de cada distinto suceso que puede darse. Así puede aparecer en desafíos basados en imágenes, tanto como en análisis de comportamientos o navegación autónoma.

A continuación se muestran algunos ejemplos de aplicaciones donde se emplea de manera notoria algoritmos de seguimiento de objetos (ver figura 1.1). La aplicación final marca en muchas ocasiones los bloques que formarán al algoritmo así como su funcionamiento y complejidad. Además pueden existir trackers que funcionen correctamente en distintas aplicaciones, o combinaciones de estos que produzcan mejores resultados en una misma aplicación.

a) b) c)



d) e) f)

Figura 1.1: Ejemplos de aplicaciones donde se emplea el seguimiento de objetos. Los rectángulos verdes indican el objeto de interés: a) Información de tráfico, b) Vídeo vigilancia – seguridad [27], c) Eventos deportivos, d) y e) Interacción humano – ordenador [28], f) Investigaciones médicas [1].

Los sensores de imagen, que se logran mejorar rápida y continuadamente, y el espectacular aumento de la potencia computacional han provocado la creación de complejos algoritmos para desarrollar y mejorar la eficiencia del seguimiento de objetos en vídeo. No obstante, el diseño de estos algoritmos (*trackers)* y su correspondiente implementación es una tarea complicada, debido a que el número de factores que pueden influir en el objeto, ya sea haciendo que se pierda o dando lugar a errores, es muy grande. Algunos de estos elementos pueden ser los cambios de iluminación del entorno, cambios de escala, las similitudes entre objetos, diferentes tipos de oclusiones del objeto de interés, cambios de pose o situaciones de motion blur. [1] (ver Figura 1.3).

Figura 1.2: Varias imágenes donde se muestran casos de seguimiento de objetos correcto (imagen izquierda), seguimiento erróneo (imagen central) y una imagen donde el objeto se ha perdido y no aparece en el plano visual de la cámara (imagen derecha).

El rectángulo rojo indica la zona de búsqueda del objeto, el rectángulo azul corresponde a la zona estimada por el algoritmo de seguimiento (*tracker*) y el rectángulo verde marca el área de la posición correcta del objeto, que se anota de forma manual (*ground-truth*).

Recientemente se han iniciado las investigaciones sobre el seguimiento de objetos durante largas etapas de tiempo, es decir, a largo plazo (*Long-Term*). Esta opción abre un gran abanico de posibilidades, con diversos campos de estudio y aplicación, como en sistemas de vídeo-vigilancia, en medicina, en la interacción entre personas y máquinas o como herramienta forense [1, 2, 3]. Debido a los avances tecnológicos constantes, y al afán del ser humano por almacenar, observar y tratar toda la mayor cantidad de información posible, surge la necesidad del seguimiento a largo plazo y su constante evolución. Grandes cantidades de datos son analizados, transportados o modificados en bancos, hospitales o desde los propios hogares, y además durante mucho tiempo. Por eso este contexto acapara tantos frentes abiertos y presenta un escenario complejo pero al mismo tiempo, con un gran número de oportunidades por investigar.

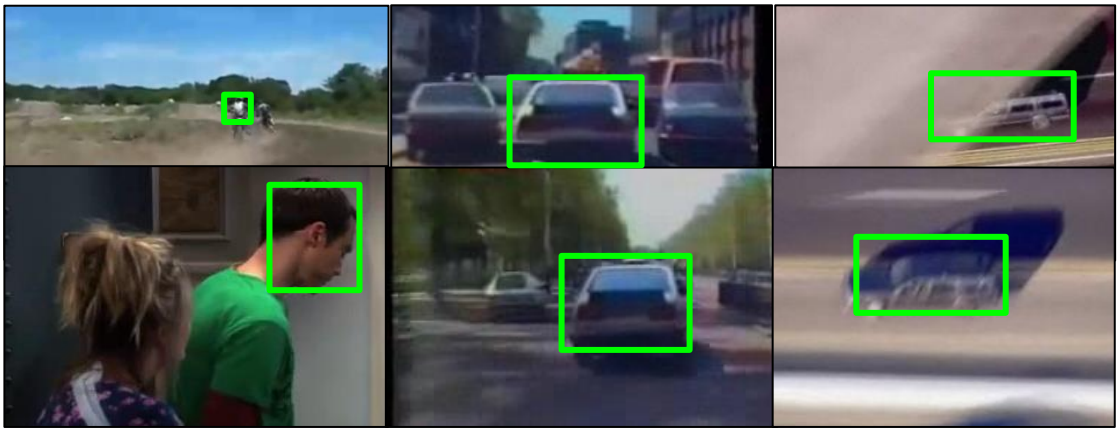


Figura 1.3: Ejemplos de diversos problemas presentes en distintas situaciones reales. Se muestra en el rectángulo de color verde el objeto seguido. De izquierda a derecha y de arriba a abajo: cambios de escala, similitud de objetos, oclusión parcial del objeto de interés, cambios de pose, cambios de iluminación y efecto motion blur.

Para llevar a cabo el estudio y desarrollo a largo plazo es conveniente dar una definición de este término. Sin embargo, no existe todavía una definición concreta, ya que al tratarse de una línea reciente de investigación, no hay un acuerdo cerrado que lo defina.

El seguimiento de objetos en vídeo hasta el día de hoy, se ha enfocado mayoritariamente en secuencias de corto plazo, pudiendo englobar este periodo de tiempo a aquellas con un número total de *frames*[1] por debajo de 1000. Muchos de los algoritmos ya existentes presentan un comportamiento aceptable en este tipo de vídeos, pero sus resultados con secuencias más largas todavía están por estudiarse o no son demasiado buenos. Además, en la actualidad cada vez se trabaja y se analiza con una mayor cantidad de información, y por consiguiente, con secuencias de mayor duración.

Por tanto, la motivación de este proyecto es contribuir al área de los vídeos de largo plazo, un campo de estudio relativamente nuevo, en el cual el número de estudios hasta la fecha no es muy amplio, en comparación con el de vídeos a corto plazo. A su vez, otro factor motivante del proyecto es el de comparar y analizar los resultados de

[1] De aquí en adelante se usarán los términos frames, fotogramas o imágenes indistintamente, para referirse a cada una de las imágenes instantáneas que componen los posteriores vídeos empleados.

algoritmos existentes en secuencias de corto plazo, con los resultados de estos mismos algoritmos con los nuevos vídeos de largo plazo, ya que la importancia de este ámbito del seguimiento de objetos continuará incrementando su importancia en los próximos años.

1.2. Objetivos

El objetivo principal de este Proyecto Fin de Carrera (PFC) es el diseño, implementación y evaluación de técnicas de actualización progresiva del modelo de objeto de algoritmos de seguimiento de objetos. Para ello se tienen en cuenta los siguientes sub-objetivos:

- Estudio del estado del arte actual. Se analizarán algunos planteamientos de trackers especialmente diseñados para vídeos a largo plazo.

- Planteamiento del problema a resolver llevando a cabo un análisis comparativo de varias técnicas, tanto para secuencias a corto plazo como para secuencias a largo plazo.

- Selección de una serie de algoritmos y posterior evaluación en varios conjuntos distintos de *dataset* (conjunto de datos utilizados, en este caso vídeos).

- Evaluación de las técnicas empleadas. Se pondrá especial atención en aquellas técnicas que presenten un mejor comportamiento para secuencias de vídeo a largo plazo.

1.3. Organización de la memoria

La memoria está formada por los siguientes capítulos:

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

- Capítulo 2: *Estado del arte*. Estudio del estado del arte del seguimiento de objetos a largo plazo.

- Capítulo 3: *Algoritmos seleccionados*. Descripción de los distintos trackers escogidos para su posterior evaluación.

- Capítulo 4: *Implementación*. Implementación de mecanismos para largo plazo.

- Capítulo 5: *Experimentación*. Exposición de los resultados obtenidos con los algoritmos seleccionados.

- Capítulo 6: *Conclusiones y trabajo futuro*.

Capítulo 2

Estado del arte

En este capítulo se estudia el estado del arte relacionado con el seguimiento de objetos en vídeos a largo plazo. En primer lugar se realiza una introducción al seguimiento de objetos, así como sus etapas y problemas más característicos (Sección 2.1). Después se estudia más en profundidad el seguimiento de objetos (*tracking*[2]) a largo plazo (Sección 2.2) para finalizar analizando las secuencias que se pueden considerar adecuadas para su uso en tracking a largo plazo (Sección 2.3).

2.1. Video tracking

2.1.1. Introducción

El seguimiento de objetos es uno de los campos con mayor crecimiento en los últimos años en el área de la visión computacional, y además presenta una espectacular progresión futura debido a que en la actualidad, el flujo de información que se recoge de cámaras, fotografías y vídeos es cada vez mayor. Esto está siendo posible por dos motivos fundamentalmente. El primero es que las características de las cámaras, tales como los sensores de imagen o la calidad de sus píxeles están mejorando rápida y continuamente. Los sensores de imagen de las cámaras son los elementos que permiten que el vídeo o la foto se vean en las mejores condiciones posibles. En este sentido, muchos sensores consiguen eliminar el efecto de objetos en movimiento o capturar la instantánea con mayor brillo o luminosidad para un mejor visionado. Compuestos por un gran número de píxeles, que provocan que las resoluciones sean cada vez mejores. El segundo es que el coste computacional de la evaluación de los algoritmos de seguimiento de objetos es cada vez más bajo, a causa de la también evolución de los ordenadores y sus características [1, 7].

[2] De aquí y en el resto del proyecto, se usarán indistintamente los términos tracking o video tracking para referirse a seguimiento de objetos.

Por ello, si se asume la definición de video tracking como el proceso por el cual se estima la localización de uno o más objetos usando una o más cámaras, y teniendo en cuenta los avances mencionados anteriormente, no resulta aventurado afirmar que este área de estudio e investigación está en plena ebullición de nuevos algoritmos y nuevos desafíos a los que enfrentarse.

2.1.2. Problemas más característicos

En este apartado se estudian algunos de los problemas más representativos que aparecen en el seguimiento de objetos [1, 4, 7]. Algunos han ido apareciendo a lo largo de las distintas etapas, pero a continuación se definirán más en profundidad (ver Figura 1.2).

- Cambios de apariencia: el objeto de interés puede sufrir diversas modificaciones a lo largo de la secuencia, como pueden ser:
    - Cambios de escala.
    - Cambios de pose (rotaciones, deformaciones).

- Cambios de iluminación: en función de las condiciones del escenario, como el momento del día, el tiempo atmosférico, sombras.

- Oclusiones: cuando por diferentes circunstancias el objeto sufre una oclusión durante un periodo de tiempo. Se suelen clasificar en:
    - Parciales: cuando la oclusión solamente afecta a una parte del objeto.
    - Totales: cuando todo el objeto sufre este fenómeno.

- Similitud de objetos: cuando la apariencia del objeto de interés es muy parecida a la de otras formas de la imagen.

- Salida de plano: se produce cuando el objeto de interés sale de la imagen correspondiente.

- Motion blur: problema que aparece cuando la región del objeto es borrosa debido al movimiento de la camara o del objeto en cuestión.

- Ruido: dependerá de cada secuencia y de las técnicas empleadas en cada algoritmo.

Una opción de esquematizar todos estos problemas es según el siguiente diagrama.

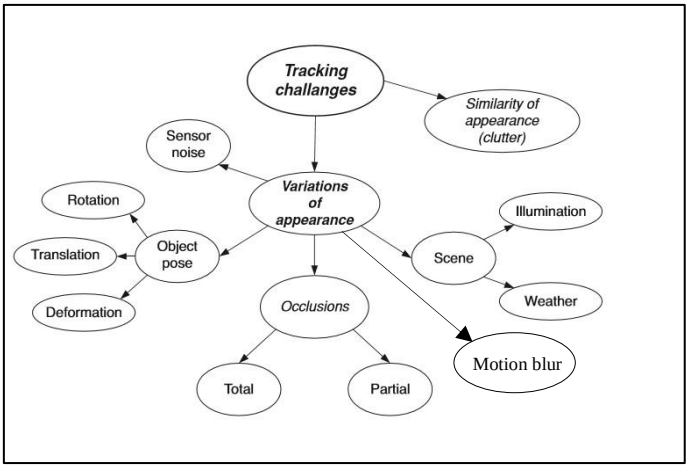


Figura 2.1: Esquema de los principales problemas del seguimiento de objetos.

2.1.3. Etapas

El creciente interés y el aumento de las investigaciones en este ámbito han provocado la aparición de muchos y muy variados algoritmos para lograr los mejores resultados posibles a la hora del seguimiento de objetos, en función de cual sea la aplicación final en cada caso. Además, al tratarse de un área totalmente abierta en la que no existe un criterio fijo o unas directrices establecidas de antemano, esto hace que el abanico de opciones sea muy elevado.

Estudiando y analizando diferentes situaciones y evoluciones con resultados satisfactorios para sus autores y para el mundo del video tracking en general, algunos de los cuales son los que aparecen en [4, 11, 12], podemos englobar o estructurar el proceso del seguimiento de objetos en vídeo en varias etapas (ver Figura 2.2).

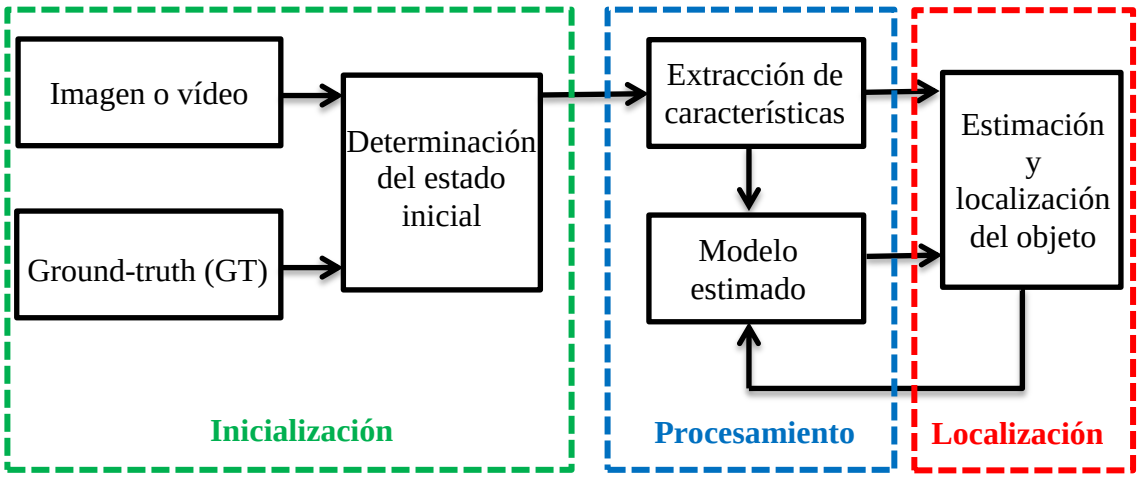


Figura 2.2: Diagrama de bloques de un seguidor de objetos en un ámbito genérico, generado a partir de [4, 11, 12, 13].

Inicialización

Comenzando con la parte de inicialización, las entradas para un algoritmo de seguimiento de objetos suelen ser en su mayoría las imágenes correspondientes a los distintos frames del vídeo, o el propio vídeo en cuestión. Esto acompañado del ground-truth (GT), que son las coordenadas correctas que indican donde se encuentra el objeto a seguir (ver Figura 2.3). Este GT se suele realizar manualmente, siendo una labor bastante tediosa en secuencias de larga duración como se verá en esta misma memoria más adelante. Además siempre debe estar disponible previamente a la ejecución del algoritmo. Al igual que las imágenes de las distintas secuencias, que pueden ser de formatos diferentes (las más empleadas son jpg y png), las coordenadas del ground-truth pueden indicar diferentes puntos, como las esquinas del objeto de interés o su centro y ancho y alto. Dentro del bloque denominado como determinación del estado inicial, aparecen varios pasos que pueden ser muy relevantes en el seguimiento de objetos.

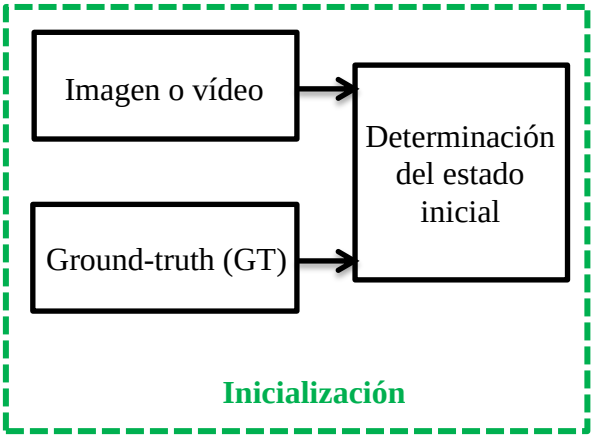
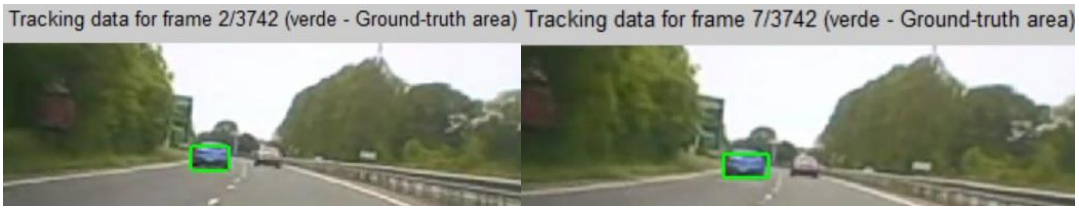


Figura 2.3: Diagrama de bloques de la fase de inicialización de un algoritmo de tracker genérico.



`[2 Nissan 268 111 34 22 0]` `[7 Nissan 271 107 37 20 0]`

`[14 Nissan 272 107 37 24 0]` `[18 Nissan 275 103 37 21 0]`

Figura 2.4: Ejemplos de varias imágenes de entrada con su correspondiente ground-truth. Los elementos indican el número de frame, una etiqueta descriptiva del objeto, las coordenadas representadas mediante x y w h, respectivamente, y finalmente la orientación.

Las etapas de la estimación inicial y del modelo inicial en ocasiones pueden desempeñar un papel muy importante en el funcionamiento y resultado del algoritmo en cuestión. Si tanto la estimación como el modelado no son lo suficientemente correctos, esto puede provocar que aparezcan errores en el seguimiento de objetos o resultados pobres relativamente pronto.

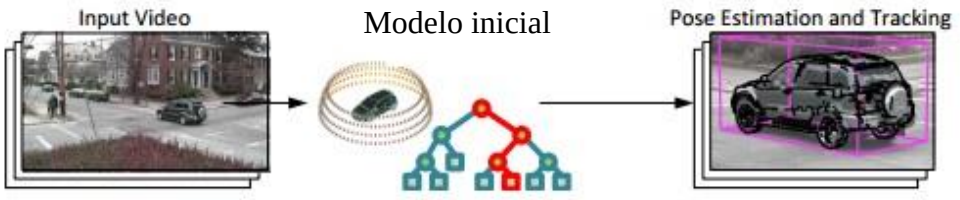


Figura 2.5: Ejemplo general de un tracker de seguimiento de objetos en su fase de inicialización. Extraído de [60].

Así pues, han aparecido en las últimas décadas, algoritmos que se concentran en la parte inicial del tracking, es decir, en el estudio y análisis de la inicialización del seguimiento del objeto, como pueden ser los descritos en [8] cuyo enfoque se concentra en la representación del objeto, o en cómo mejorar la plantilla o modelo del primer frame para así mejorar las siguientes, como se publica en [9]. Además hay investigaciones que tratan de mejorar trackers ya implementados, dotando de alteraciones al estado inicial del objeto en la primera imagen para conseguir unas determinadas características, como en el caso de [10], cuyo objetivo es un análisis más robusto. Otro ejemplo es el que se explica en [14], mejorando la estimación inicial de la pose pasando de 2D a 3D.

Procesamiento

Siguiendo el diagrama de la Figura 2.2, la siguiente etapa consta de dos grandes bloques, la extracción de características y el modelo estimado del objeto. Existen muchas formas de desarrollar cada bloque de esta fase, pero se ha empleado en esta memoria la que se enuncia en [4] por considerarse que muestra una visión global y que sus criterios son los más utilizados en el entorno del seguimiento de objetos.

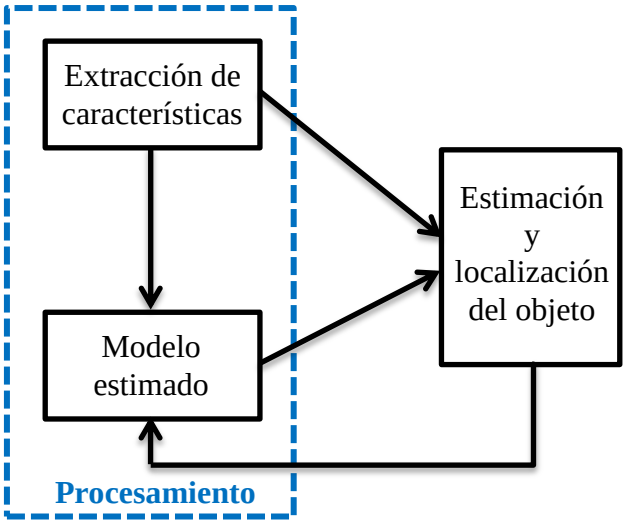


Figura 2.6: Diagrama de bloques de la fase de procesamiento de un algoritmo de tracker genérico.

El ajuste de las características, formado por las fases de extracción y proceso o selección de las mismas, puede tener un papel muy importante en el resultado final del tracker. Para la primera parte, es decir, la extracción de las características, se hace uso de una clasificación que engloba de manera sencilla muchos aspectos importantes. La extracción se divide en: por partes, o por características que representan la apariencia del objeto por completo.

La *extracción por partes* consiste en seleccionar pequeños objetos o regiones dentro del objeto y estudiar su relación. Entre los aspectos positivos de esta técnica están la no necesidad de usar características de objetos ocluidos o la posibilidad de clasificar las características como frente o fondo. Algunos de los mecanismos más utilizados son los filtros Haar, los valores RGB y los Histogramas de Gradientes Orientados (HOG). Los filtros Haar son muy usados por su invarianza a los cambios de brillo y por su bajo coste computacional. Esto es debido a que los filtros con bases Haar realizan una codificación que lo que calcula es la diferencia de intensidades en la imagen correspondiente, obteniendo así características que pueden ser contornos, puntos o líneas mediante la diferencia de contrastes entre una o más regiones [15, 16].
Mientras que con el uso de RGB y los histogramas HOG, se favorece la robustez del tracker, ya que hacen más fácil el manejo de pequeñas partes que puedan formar parte de pequeñas oclusiones o deformaciones [4].

La *extracción total* u *holística* de características representa la apariencia completa del objeto. Esto se puede obtener a través de una plantilla del objeto, por medio del color, de los histogramas de gradientes o por una combinación entre varias de estas técnicas. La parte positiva es que los histogramas toleran cambios de escala y rotaciones del objeto sin que la apariencia del modelo tenga que adaptarse a estas nuevas circunstancias, pero sí que es necesario que las plantillas se complementen con un algoritmo de actualización del modelo [4, 17, 18].

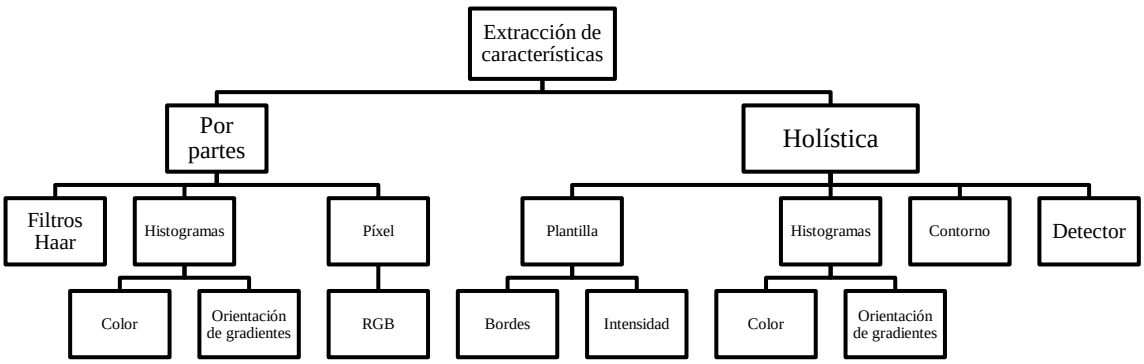


Figura 2.7: Esquema que muestra las distintas opciones de cada mecanismo de extracción de características. Generado a partir de [4].

La siguiente fase consiste en el proceso y/o la selección de las características extraídas anteriormente. En función del algoritmo de seguimiento seleccionado, puede producirse ambas partes o solamente la selección o el proceso. Por lo general, si se procesan las características, se emplea un pívot, siendo este elemento la apariencia inicial del objeto. Esto conlleva asumir que el rectángulo de coordenadas final en la primera imagen es el correcto y que la apariencia del fondo y del objeto mantiene similitudes con la inicial [4, 9].

Por otra parte, la selección de características se lleva a cabo cuando existe un conjunto de características cuya composición se actualiza con cada frame de acuerdo a su efectividad en el instante anterior. En este paso se asume que la posición estimada por el tracker en el frame anterior es la correcta. Un aspecto fundamental de la selección de características es conseguir que ciertos clasificadores actúen como selectores de características. Esto ocurre en los trackers de carácter discriminativo, que son aquellos que separan la apariencia del objeto con el fondo de la imagen [4, 20, 21].

La última fase de la etapa de procesamiento consiste en el modelo de apariencia del objeto. Resulta un bloque fundamental en la resolución final del algoritmo, por ello, muchos trackers centran su atención en esta etapa. Además suele ir seguida de una actualización del modelo de apariencia, tema en el que actualmente se está poniendo el foco de atención de este campo para conseguir importantes mejoras y progresos.

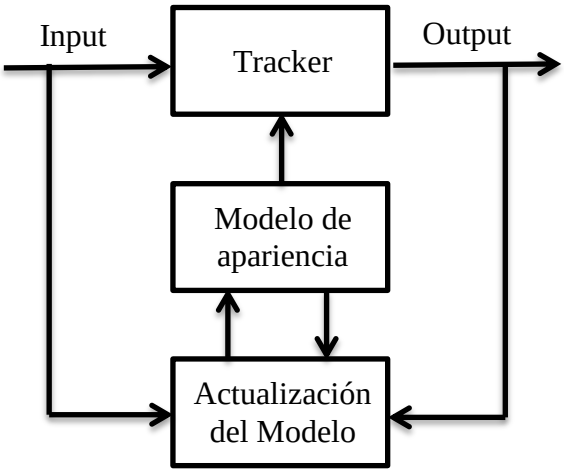


Figura 2.8: Diagrama de bloques de las distintas secciones en el procesamiento del modelo de apariencia del objeto. Extraído de [1].

Como se puede observar en la Figura 2.8, la salida del tracker puede ser empleada a su vez para adaptar el modelo del objeto de interés a las condiciones del escenario en el que se encuentre en ese instante [1]. Sin embargo existe cierto riesgo ante esta situación, ya que es más probable que aparezca cierta decadencia en la información del modelo debido a la amplificación del error del tracker causado por el bucle de realimentación de la actualización (ver Figura 2.9).

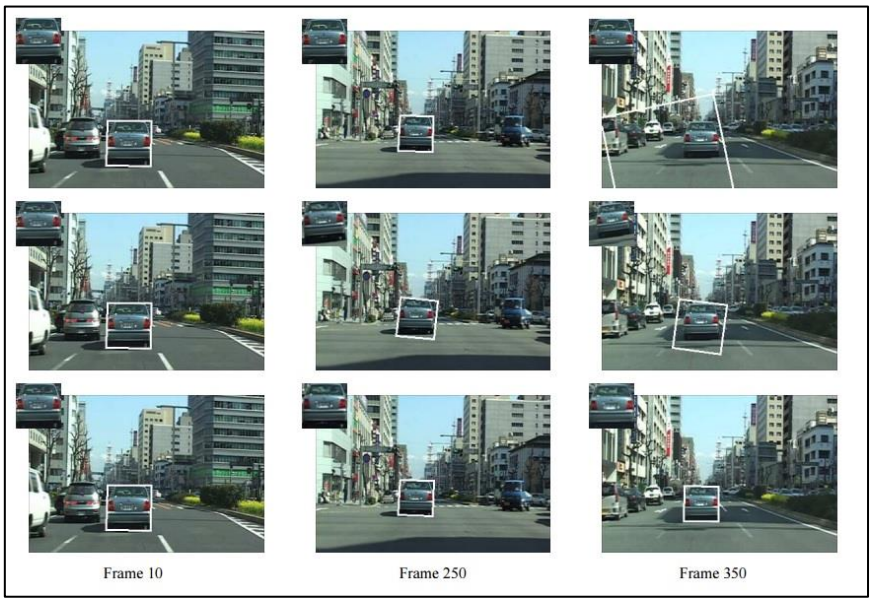


Figura 2.9: Comparación de una secuencia usando distintas estrategias de actualización. En la primera fila no se produce actualización y finalmente el tracker es erróneo. En la segunda fila hay una actualización cada frame y al final se puede observar cierto drifting[3]. En la tercera fila, además de la actualización en cada frame hay una fase que corrige el drifting. El tracker es correcto en la secuencia completa. Extraído de [9].

Localización

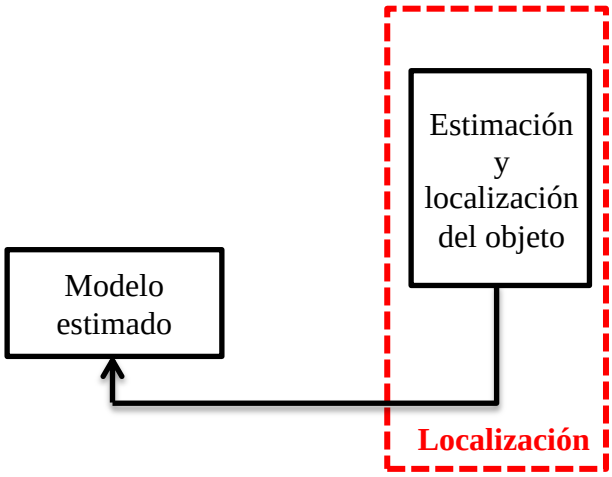


Figura 2.10: Diagrama de bloques de la fase de estimación y localización de un algoritmo de tracker genérico.

La última etapa es la estimación o localización del objeto de interés, y su consiguiente forma de representación. Si bien la posición va a depender de todos los procesos y factores descritos en las fases anteriores, la representación final puede depender de uso de la aplicación finalmente. En la mayoría de ocasiones, un rectángulo de coordenadas (comúnmente denominado como bounding box[4]) es usado para delimitar la posición de

[3] Drifting: Fenómeno común en el video tracking, que consiste en que el objetivo se desplaza gradualmente lejos de la plantilla establecida [24], provocando un mayor error con el paso del tiempo.
[4] De aquí y en adelante se usarán los términos rectángulo de coordenadas y bounding box indistintamente.

la región o del objeto de interés. Pero pueden aparecer otras representaciones como las siguientes:

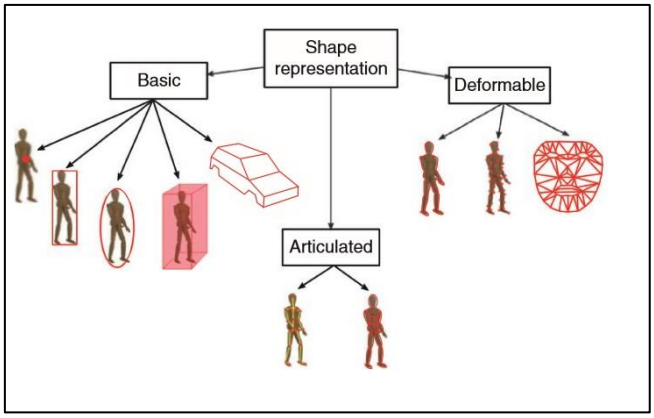


Figura 2.11: Distintas formas de representar al objeto de interés. De izquierda a derecha: formas básicas (punto, rectángulo, elipse, cubo, contornos), formas articuladas y contornos deformables. Extraído de [1].

2.1.4. Métricas de evaluación

En esta sección se van a mostrar algunos de los métodos existentes para evaluar los algoritmos de seguimientos de objetos. Entre los más usados y extendidos a nivel global se encuentran el FDA (y SFDA) [4, 31], el CLE [30, 31] y el CTR [4, 61].

- FDA (Frame Detection Accuracy): Este método calcula el solapamiento entre el ground-truth (GT) y la salida del sistema como una relación de la intersección entre dos objetos y la unión entre ellos. La suma de todos los solapamientos es normalizada por encima del promedio del número del GT y los objetos detectados.

$$FDA(t) = \frac{Overlap\ Ratio}{\left[\frac{N_G^{(t)} + N_D^{(t)}}{2}\right]} \quad (1) \quad \text{Siendo, } Overlap\ Ratio = \sum_{i=1}^{N_{mapped}^{(t)}} \frac{\left|G_i^{(t)} \cap D_i^{(t)}\right|}{\left|G_i^{(t)} \cup D_i^{(t)}\right|} \quad (2)$$

Ecuación (1): Fórmula del FDA. Ecuación (2): Fórmula Overlap o solapamiento.

Dónde: - $N_G^{(t)}$: Objetos del ground-truth.
- $N_D^{(t)}$: Objetos detectados.
- $N_{mapped}^{(t)}$: Número de objetos mapeados en el frame t.
- $G_i^{(t)}$: Indica el objeto del GT i^{th} en el frame t^{th}.
- $D_i^{(t)}$: Indica el objeto detectado i^{th} en el frame t^{th}.

La fórmula anterior calcula el FDA para un frame determinado. Para obtenerlo de la secuencia completa, se emplea la SFDA (Sequence Frame Detection Accuracy), que no es más que calcular la FDA a lo largo de todos los frames de la secuencia y normalizado por el número de frames de la secuencia, donde al menos existe un GT o un objeto detectado.

$$SFDA = \frac{\sum_{t=1}^{t=N_{frames}} FDA(t)}{\sum_{t=1}^{t=N_{frames}} \exists\, (N_G^{(t)}\ OR\ N_D^{(t)})} \quad , (3)$$

Ecuación (3): Fórmula del SFDA.

- CLE (Center Location Error): Se define como la distancia Euclídea entre el centro de las coordenadas del objeto y el centro del ground truth. Se puede considerar también calcular la media de esta formulación para tener una medida concreta del resultado final.

$$CLE = \sqrt{((centerPOS_1) - (centerGT_1))^2 + ((centerPOS_2) - (centerGT_2))^2} \ , (4)$$

Ecuación (4): Fórmula para calcular el CLE.

Siendo: $centerPOS_1$: primera coordenada del centro de la posición del objeto.
$centerPOS_2$: segunda coordenada del centro de la posición del objeto.
$centerGT_1$: primera coordenada del centro del ground-truth.
$centerGT_2$: segunda coordenada del centro del ground-truth.

- CTR (Correct Track Ratio): Porcentaje de frames de una secuencia en la que el tracker detecta correctamente al objeto de interés. Es conveniente definir previamente un umbral para delimitar qué porcentaje del objeto se considera como correcto, en la mayoría de los casos se utiliza un umbral igual a $d_k \geq 0.7$. Se puede emplear en combinación con la media del SFDA de forma que así se puede elegir el tracker que mejor se ajuste a los requisitos de la aplicación. Para hallar el valor del CTR, se parte del solapamiento definido por la puntuación Dice, d_k.

$$CTR = \%\ frames\ (k)\ con\ d_k\ \geq umbral \ , (5) \quad \text{Siendo, } d_k = \frac{2\left|B_{x_k} \cap B_{x_k GT}\right|}{\left|B_{x_k}\right| + \left|B_{x_k GT}\right|} \ , (6)$$

Ecuación (5): Fórmula para calcular CTR. Ecuación (6): Fórmula para la puntuación d_k.

2.2. Tracking a largo plazo

2.2.1. Definición y origen

En secciones anteriores se ha explicado en detalle en qué consistía el seguimiento de objetos en vídeo. La diferencia de este apartado es el concepto de largo plazo, una categoría introducida hace relativamente poco tiempo. Por ello, en esta sección se intentará dar una definición más precisa de esta reciente situación en el seguimiento de objetos.

Los primeros estudios del seguimiento de objetos a largo plazo, de ahora en adelante, también denominado *Long-Term Tracking (LTT)*, fechan del año 2010 aproximadamente. Además fue un nacimiento paulatino, ya que la idea seguía siendo el seguimiento de objetos pero en problemas que a diferencia de las investigaciones anteriores, si eran problemas cuya duración en el tiempo era importante [32]. Así pues, el verdadero interés por el LTT surgió con el estudio de Kalal et al. [33], que propuso posiblemente el primer tracker a largo plazo, el Tracking-Learning Detection (TLD). A partir de entonces se publicaron otros análisis en los que el seguimiento de objetos a largo plazo tenía especial relevancia. Pero en todos ellos, existe un denominador común. Se da por hecho la definición de largo plazo y se pasa a explicar cómo afecta o cómo se puede solucionar en un determinado problema o tracker, según corresponda.

Por esta razón es posible enunciar que no existe una definición exacta de esta nueva línea de investigación, del seguimiento de objetos a largo plazo. Resulta muy complejo establecer los límites, dónde empezaría y donde acabaría un tracking para categorizarlo de corto o largo plazo. Sin embargo, por lo estudiado en el estado del arte de este PFC, se propone una posible definición para el LTT, de forma que pueda servir de ayuda en el futuro para clasificar de manera más sencilla los trackers a corto plazo y los trackers a largo plazo.

Se puede definir por tanto, el tracking a largo plazo como aquel seguimiento de objetos cuyas secuencias se extienden a lo largo del tiempo, donde las características de los objetos pueden variar considerablemente debido a la aparición de múltiples problemas. Así pues el algoritmo debe aprender y adaptarse a lo largo del tiempo. En la práctica, secuencias por encima de 1000 frames se consideran largo plazo.

2.2.2. Enfoques y problemas

En esta sección se aborda con más detalle los diferentes enfoques que se han hecho sobre el seguimiento de objetos a largo plazo hasta la fecha, así como los problemas más comunes que causa el largo plazo. Algunos de los más destacados son el método de detección, cómo se consigue localizar el objeto de interés en función del entorno en el que esté o con las causas que lo puedan afectar [5]; la inicialización del entorno de seguimiento, ya que la posición del objeto en relación al fondo de la imagen puede causar que los resultados del algoritmo sean menos favorables [4]; la detección de oclusiones, tanto parciales o totales, estáticas o dinámicas [5, 6] y la actualización del modelo, en función del uso final de la aplicación.

Se parte del desarrollo del tracker TLD [33], que se puede considerar como el tracker base o de referencia para investigaciones posteriores en el seguimiento a largo plazo. Proporciona una teórica primera definición del LTT: aquel video que se procesa con una tasa de frames y cuyo proceso debería ejecutarse indefinidamente. Además se indica el problema fundamental del seguimiento a largo plazo, que es la detección del objeto cuando reaparece en el campo de visión de la cámara. Sin embargo, los inconvenientes no terminan aquí, ya que se pueden agravar por el hecho de que el objeto puede cambiar su apariencia y esto deja en el aire la relevancia de la apariencia inicial del objeto. También indica que un buen tracker LT debería controlar cambios de escala y de iluminación, oclusiones parciales y operar en tiempo real.

Otro aspecto a destacar es que parte de dos enfoques bien diferenciados (ver Figura 2.12), que son el seguimiento por un lado, y por el otro, la detección.

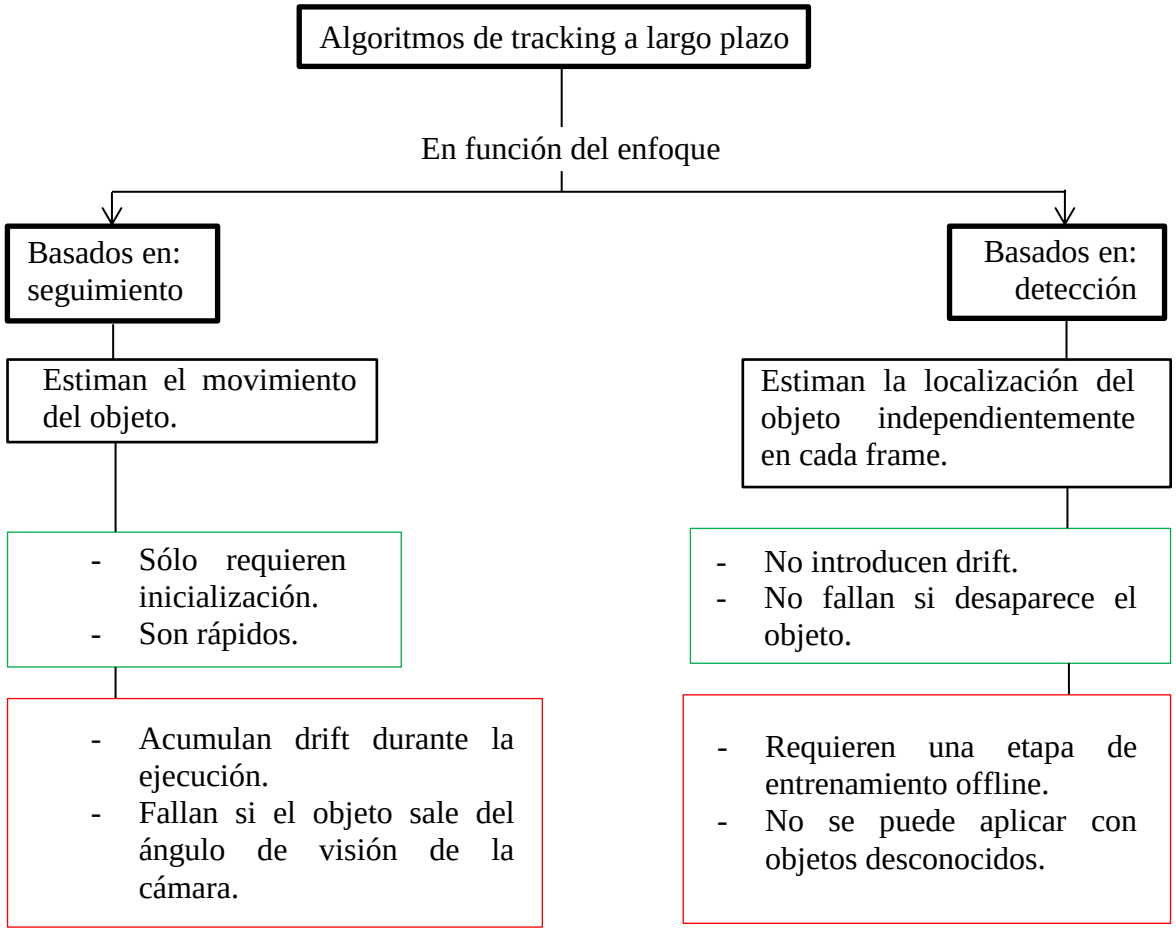


Figura 2.12: Esquema distintos enfoques trackers a largo plazo. Los rectángulos verdes indican las ventajas de esa aproximación mientras que lo contenido en el rectángulo rojo muestra los inconvenientes de cada uno de ellos. Generado a partir de [33].

Vistas ambas situaciones, ya se puede considerar el punto de partida del algoritmo TLD, que es la simultaneidad de los dos enfoques, de manera que un tracker de seguimiento puede proporcionar ligeros datos de entrenamiento para un detector a lo largo del tiempo, y un tracker de detección puede reinicializar un algoritmo y así minimizar los errores. Este nuevo algoritmo consiste en dividir el tracker en tres partes: seguimiento,

aprendizaje y detección. En primer lugar el algoritmo sigue al objeto de interés frame a frame. Después se realiza un aprendizaje en relación al error del detector y se actualiza para evitar errores en el futuro. Y el detector localiza todas las apariencias que se han observado y si son incorrectas, las actualiza. Pero también expone una serie de limitaciones, como son que el algoritmo sólo entrena al detector, mientras que la parte de tracking se mantiene fija, o que no se produce una sustracción del fondo para obtener supuestos mejores resultados.

Otros se centran en el común problema a largo plazo de las oclusiones. Según se propone en [32], mediante dos vías de información se busca un método que sea capaz de manejar oclusiones que se produzcan durante largos periodos de tiempo. La primera vía se lleva a cabo mediante la estrategia de sustracción del fondo y la segunda a través de la estimación de varias GMMs. La técnica de sustracción del fondo consiste en que las imágenes de la escena sin objetos que puedan ser considerados como intrusos, muestran un comportamiento regular que puede ser bien descrito por un modelo estadístico. Si se tiene un modelo estadístico de la escena, un objeto intruso puede ser detectado mediante la detección de las partes de la imagen que no encajan en el modelo (ver Figura 2.13). Continuando por esta misma línea, en [2] se da especial importancia al aprendizaje del modelo de apariencia y a la detección, empleando conjuntos de ejemplos negativos, algo no muy habitual en tracking, para enfrentarse al problema de las oclusiones y cambios de escala.

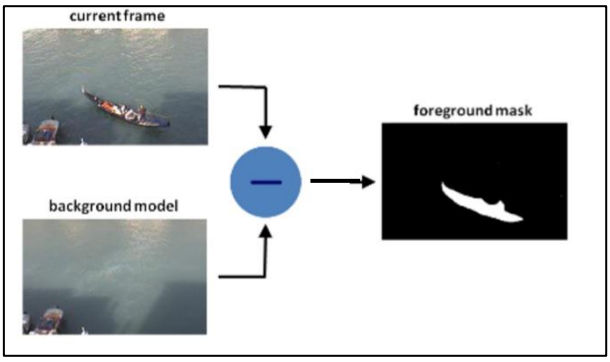


Figura 2.13: Esquema funcionamiento de la técnica de sustracción del fondo. Generado a partir de [45].

Muchas propuestas consideran un aspecto fundamental del Long-Term Tracking el bloque del modelo de aprendizaje y su actualización. Por ello, y por otros muchos factores, existen numerosos mecanismos para desarrollar la estimación del modelo y su consiguiente actualización, que, según indican en [4], algunos de ellos son los que se describen con más detalle a continuación:

- Último modelo: consiste en usar el modelo estimado en el último frame como modelo del siguiente frame y así sucesivamente. El problema de esta estrategia reside en que es una técnica muy propensa al drifting y por ello es necesario realizar adaptaciones adicionales para reducirlo, como se establece en [9].

- Ranking: consiste en almacenar las muestras o clasificadores más eficaces hasta una cantidad fijada, añadiendo siempre una más después de procesar cada imagen [4]. Puede provocar un gran aumento del coste computacional en función de la longitud de la secuencia.

- Blending: las muestras extraídas en el frame actual se mezclan con las anteriores. Es más estable que las estrategias definidas anteriormente, pero también es propenso a acumular drift, debido a que la inclusión de muestras erróneas no puede ser mitigado después.

- GMM (Gaussian Mixture Model): este método emplea varias distribuciones Gaussianas, útiles en entornos complejos para establecer el modelo del fondo especialmente. Suele obtener buenos resultados, pero su coste computacional es alto e introduce cierta cantidad de ruido en función del número de distribuciones escogidas [22, 25].

- Combinación: la estrategia que se establece [23] es la de realizar la colaboración entre, por un lado, un clasificador que asigna mayor peso al primer plano que al fondo (SDC), con un método basado en histogramas que toma la información de pequeños trozos de un entorno pensado para manejar oclusiones.

En [3, 4] coinciden en que una buena estrategia para la actualización del modelo es mediante clasificadores SVM (Support Vector Machine) [34] y sitúan al tracker STRUCT (Structured Output Tracking with Kernels) [35] como el que mejores resultados proporciona. El algoritmo STRUCK hace uso de filtros Haar (explicados en la sección 2.1.3.), concretamente de seis tipos diferentes de filtros, para la extracción de características. Además otra notoria ventaja es que se trata de un tracker capaz de identificar distintas apariencias del objeto a lo largo del tiempo, y sin descartar las antiguas. Esto hace que ayude a prevenir la aparición de drift durante el seguimiento, que podría tener lugar si la información pasada es descartada. A su vez, al desarrollar un mecanismo de aprendizaje estructurado, a través de SVM estructurados en lugar de clasificadores SVM, ha hecho posible obtener mejores resultados, salidas con menos ruido y facilitar la incorporación de mapas de características.

Los enfoques más recientes estudian problemas del LTT más allá de las oclusiones, como son cambios de apariencia debido a deformaciones, movimientos bruscos, o salidas del objeto de interés de la imagen. Además introducen nuevas estrategias para solventar estos problemas tales como la re-detección del objeto cuando el tracker falla o la combinación de métodos desarrollados en estudios anteriores [36]. También se proponen numerosas mejoras al primer algoritmo de largo plazo, el TLD. Varios trackers que siguen esto son el LT-FLO (Long Term Feature Less Object) [6]. El enfoque de este estudio se concentra en varias contribuciones. La primera de ellas es que el modelo del objeto se aprende de forma que no hay supervisión ninguna. El segundo punto importante es la propuesta de una estrategia de re-detección, que identifica cuando un tracker falla o un objeto desaparece y emplea el modelo online de apariencia, junto con el clasificador aprendido para volver a localizar al objeto. De esta forma se aporta de mayor estabilidad a largo plazo, lo que unido a su resistencia al drift, representa un marco adecuado para LTT con oclusiones totales. Todos estos procesos dan lugar al primer tracker a largo plazo basado en bordes, que se denomina LT-FLO (ver Figura 2.14. y 2.15).

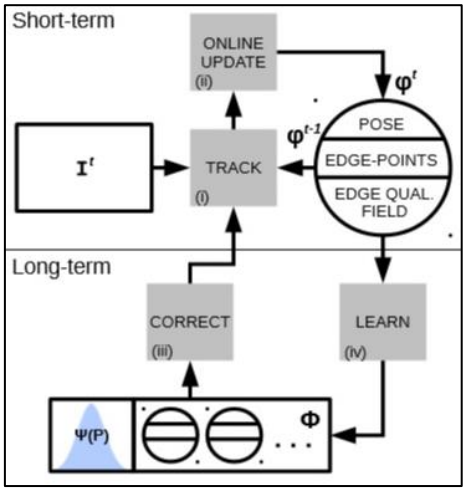


Figura 2.14: Diagrama de bloques del algoritmo del tracker LT-FLO. Extraído de [6].

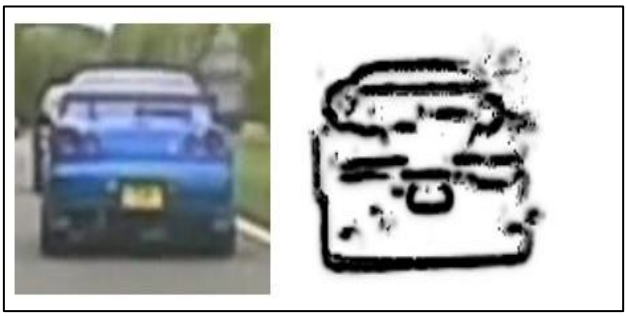


Figura 2.15: Ejemplo de una imagen y su calidad de bordes tras varios frames de tracking. Extraído de [37].

Recientemente, como mejora de este último, se ha desarrollado trackers basados en técnicas que emplean esquinas virtuales [37] para tratar de solucionar problemas de luminosidad cambiante u objetos de baja textura, desafíos que en muchos casos a día de hoy, son razón de fallo de muchos trackers a largo plazo.

2.3. Resumen del análisis a largo plazo

A continuación se muestra la Tabla 2.1. que contiene algunas de las características más importantes que se han mencionado en numerosos artículos en la sección anterior. El objetivo es agrupar toda la información posible de cara a que su consulta sea más rápida y visual.

Estudio	*Basados en*	*Inicialización*	*Modelo de apariencia*	*Detección de oclusiones*	*Reducción drifting*
[2]	DET	*	*****	***	***
[3]	DET	**	****	****	**
[4]	DET	***	*****	***	**
[5]	DET	*	***	****	**
[6]	SEG/DET	*	***	***	**
[9]	SEG	**	**	*	****
[22]	SEG	**	***	*	*
[23]	SEG/DET	**	***	****	***
[25]	SEG/DET	***	***	*	*
[32]	DET	*	****	*****	*
[33]	SEG/DET	***	***	***	**
[35]	SEG/DET	***	**	**	**
[36]	SEG	**	**	***	***
[37]	SEG/DET	***	****	***	**

Tabla 2.1: Resumen de las características en relación a la documentación estudiada en el Capítulo 2 de esta memoria. El criterio decidido va del rango de * a *****, donde la atención y desarrollo de esa característica o etapa va en orden ascendente de 1 a 5*. SEG = Seguimiento, DET = Detección.

Capítulo 3

Trackers seleccionados

En este capítulo se proponen una serie de trackers que han sido seleccionados para evaluar posteriormente un conjunto de secuencias a corto plazo y un conjunto de secuencias a largo plazo (explicadas en el Capítulo 5), de tal forma que se pueda comparar su funcionamiento a la vista de los resultados y sacar determinadas conclusiones. Se propone una pequeña introducción para realizar una primera situación global de cada tracker seleccionado. Cada algoritmo, consta de dos secciones en este capítulo. La primera de ellas consiste en la descripción del mismo, mientras que la segunda refleja el funcionamiento en este PFC del tracker correspondiente.
El código de los dos primeros trackers, MS y CBWH respectivamente, ha sido proporcionado por el VPU-Lab. Para el resto de los algoritmos, se ha obtenido el código de la página web de cada autor, que se encuentra públicamente disponible.

3.1. Introducción

En un algoritmo de seguimiento visual se suelen distinguir dos componentes principales. Por un lado, la rama de representación y localización del objeto, que está reñida con los cambios de apariencia en el mismo, y por otra parte, el filtrado y asociación de datos y características, que se suele enfrentar a todo lo relacionado con aspectos dinámicos del objeto a seguir, basándose en conocimientos previos y evaluando diferentes hipótesis. La manera en la que se combinan estas dos componentes, depende mucho de la aplicación final, ya que en algunos casos será más importante el peso que tenga la representación del objeto mientras que otros habrá que centrar más la atención en el movimiento de lo que se siga. Se han seleccionado un total de seis algoritmos diferentes con los que trabajar en este proyecto, que son los siguientes:

- Mean – Shift (MS): concentra su atención en la componente de la representación y localización del objeto. Intenta maximizar un función de densidad de probabilidad.

- Corrected Background Weighted Histogram (CBWH): se trata de una versión mejorada del algoritmo Mean – Shift, empleando la información del fondo para realizar una transformación de la representación del modelo de objeto exclusivamente.

- Color Tracker (COT): algoritmo que tiene como componente principal la característica del color, en sus distintas formas y atributos.

- Active Feature Selection (AFS): es un tracker discriminativo que propone un nuevo método de aprendizaje de sus clasificadores para aumentar su robustez y eficiencia.

- Active Examples Selection Tracker (AEST): presenta otro mecanismo de aprendizaje con el que se espera conseguir resultados más favorables y un tracker más robusto.

- Multiple Experts using Entropy Minimization (MEEM): detalla una nueva estrategia, basada en Support Vector Machine (SVM), para lograr un modelo de apariencia más eficiente.

3.2. Mean – shift (MS)

3.2.1. Descripción

El tracker Mean – Shift es un algoritmo muy empleado en el seguimiento de objetos debido a su simplicidad y eficiencia. Desde que se introdujo en el entorno de la visión computacional, se ha usado para solucionar problemas de segmentación, filtrado de imágenes y tracking, entre otros. La motivación para su selección ha sido comprobar cómo se sigue comportando actualmente uno de los trackers más usados y analizar lo que sucede al ejecutarse con secuencias largas, donde están presentes números desafíos para este tipo de algoritmos, como continuos cambios de escala, oclusiones totales o similitud de colores, que pueden suponer una gran losa para la información de color que pueda manejar en situaciones de agrupamiento.

Los puntos principales de este tracker son las representaciones tanto del objeto de interés como de las regiones candidatas, mediante un histograma de color, y la obtención de una función de semejanza entre ambas representaciones. Cuanto más suave sea esta función, mejor se podrá aplicar el método de la optimización del gradiente y por lo tanto, mucho más rápido será la localización del objeto en relación con la estrategia de búsqueda exhaustiva [17, 19]. Para medir la semejanza entre el modelo del objeto y el modelo candidato, Mean – Shift utiliza el coeficiente Bhattacharyya (ver Ecuación 7 y 8), que define la distancia entre las dos distribuciones discretas del siguiente modo.

Siendo d(y) la distancia entre ambas distribuciones p y q, y el coeficiente Bhattacharyya, entre p y q, ρ:

$$d(\mathbf{y}) = \sqrt{1 - \rho\left[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}\right]}\,, \tag{7}$$

$$\hat{\rho}(\mathbf{y}) \equiv \rho\left[\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}\right] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y})\hat{q}_u}\,, \tag{8}$$

Ecuaciones 7 y 8: Distancia entre dos distribuciones mediante el Coeficiente de Bhattacharyya [17].

El objetivo es minimizar la distancia d en la ecuación 7, o lo que es lo mismo, maximizar el coeficiente ρ de la ecuación 8.
Dado un conjunto de puntos, que puede ser una distribución de píxeles mediante un histograma de las proyecciones del fondo, y una pequeña ventana (que puede ser circular, cóncava…) el objetivo es ir moviendo la ventana al área de mayor número de puntos (o al área de mayor densidad de píxeles). Esto se describe gráficamente a continuación:

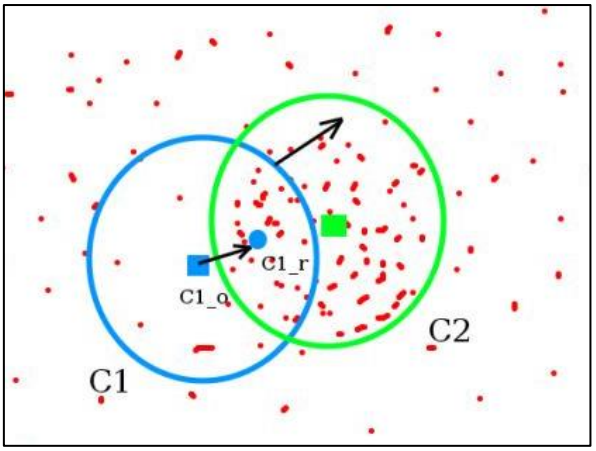


Figura 3.1: Ejemplo de funcionamiento de Mean - Shift. La ventana inicial corresponde a C1 en color azul, con centro en el rectángulo azul. El primer paso es calcular un nuevo centroide dentro de la ventana para ver si hay una concentración de puntos mayor en otro punto de la misma. Efectivamente esto ocurre, y el nuevo centro pasa a ser el cuadrado azul. Comienza la iteración del algoritmo. Se repite este proceso hasta que el nuevo centroide sea el punto con mayor distribución de píxeles de toda la imagen. En la imagen, esto corresponde al círculo verde C2. Extraído de [42].

A pesar de que este tracker mejoró en cuanto a valores de robustez, es modulable y posee una implementación sencilla, también presenta una serie de limitaciones importantes. Cuando las características del objeto de interés también están presentes en el fondo, tiende a generar mínimos locales, produciendo interferencias entre el fondo y

el objeto. Si existen oclusiones constantes, se debe emplear un filtro de movimiento más sofisticado, y también podría mejorarse si, conociéndose el movimiento del objeto frame a frame, se inicializase en múltiples localizaciones para mejorar los resultados de la función de semejanza.

3.2.2. Funcionamiento

El tracker Mean – Shift empleado en este proyecto recibe como entrada la secuencia de imágenes correspondientes en forma de vídeo, un archivo de ground-truth de la secuencia, y la configuración establecida anteriormente, de acuerdo a las características deseadas. Su implementación se ha basado en lo descrito en [17].

Tras calcular el centro del primer frame, inicializa el modelo del objeto. Halla el histograma RGB del modelo de objeto. Una vez realizado esta primera fase de inicializaciones, compara el histograma del modelo con el histograma del frame candidato. Calcula un nuevo centroide y lo normaliza. Después obtiene el nuevo histograma del modelo de objeto para conseguir de esta forma las posiciones de la nueva localización. Por último reinicia el modelo candidato. Se repite este proceso para cada frame de la secuencia.

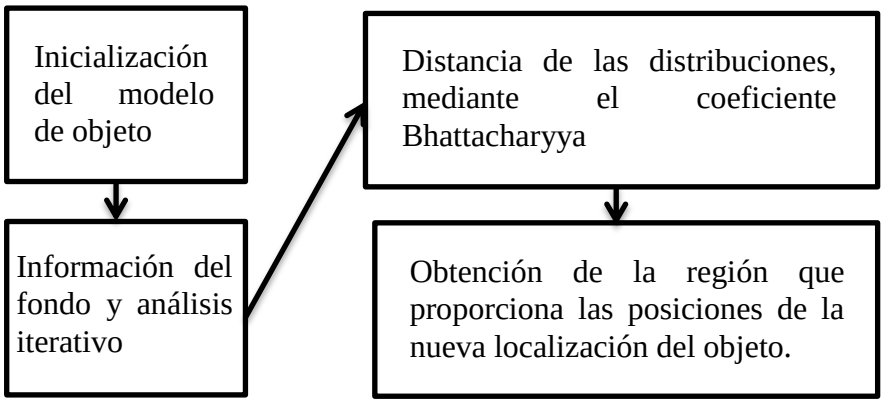


Figura 3.2: Diagrama de bloques que resume el funcionamiento del tracker MS.

3.3. Corrected Background Weighted Histogram (CBWH)

3.3.1. Descripción

Este tracker se trata de una versión mejorada del algoritmo Background Weighted Histogram (BWH), que pretendía a su vez, mejorar algunos de los problemas que tiene el tracker MS descrito en este mismo PFC en la Sección 3.2. Se ha seleccionado este algoritmo porque se trata de una mejora del primer tracker escogido, y dado que en secuencias de corto plazo supuestamente tendrá mejores resultados, existe curiosidad por observar y estudiar qué sucede con los vídeos de larga duración. De esta manera, se puede determinar si el algoritmo es válido para largo plazo o bien, se queda en una versión mejorada solamente para seguimiento en periodos de tiempo cortos.

La información del fondo es importante, entre otras razones, porque algunas de las características del objeto se encuentran también presentes en el fondo de la imagen, y su relevancia para la localización del objeto disminuye. Además en muchas ocasiones, resulta difícil separar exactamente el objeto del fondo, y su modelo puede contener características del fondo también. Al mismo tiempo, un uso incorrecto de la información del fondo puede afectar a la parte de la selección de escala, y esto puede llevar a que sea imposible determinar la escala apropiada del objeto. La iteración del algoritmo MS es invariante a la transformación de escala de los pesos, y por esta razón BWH no consigue mejorarlo debido a que transforma la representación del modelo de objeto y del modelo candidato.

Por este motivo surge el tracker CBWH, para conseguir realmente esa mejora del algoritmo Mean – Shift, y para ello propone un nuevo mecanismo de transformación. Partiendo de la representación discreta (histograma) del fondo en el espacio de características, como:

$$\{\hat{o}_u\}_{u=1\cdots m} \quad (\text{with} \quad \sum_{i=1}^{m} \hat{o}_u = 1) \qquad (9)$$

Y asumiendo que ô* es el valor mínimo distinto de cero, los coeficientes o pesos

$$\left\{v_u = \min\left(\hat{o}^* / \hat{o}_u, 1\right)\right\}_{u=1\cdots m} \qquad (10)$$

son usados para definir la transformación entre la representación del modelo de objeto y del modelo de candidato. Es en este punto donde se introduce la novedad del algoritmo. La Ecuación (10) se emplea para realizar la transformación del modelo de objeto solamente. Lo que esto significa es que se reducen las características salientes del fondo solo en el modelo de objeto pero no en el modelo de candidato, como sucede en BWH.
Se define una nueva fórmula de ponderación:

$$w_i'' = \sqrt{\hat{q}_{u'}' / \hat{p}_{u'}(\mathrm{y})} \qquad (11)$$

Donde $\hat{q}_{u'}'$ y $\hat{p}_{u'}$ son el modelo de objeto y el modelo de candidato, respectivamente.

Mediante un sencillo proceso de derivación se obtiene:

$$w_i'' = \sqrt{C'/C} \cdot \sqrt{v_{u'}} \cdot w_i \qquad (12)$$

Como la constante $\sqrt{C'/C}$ es un factor de escala que no influye en el proceso de seguimiento, se puede omitir y simplificar la Ecuación (12) de manera que el resultado final es:

$$w_i'' = \sqrt{v_u} w_i \qquad (13)$$

La Ecuación (13) muestra claramente la relación existente entre el peso calculado usando la representación del objeto (w_i) y el peso calculado mediante la información del fondo (w_i''). Si el color del punto i en la región del fondo destaca, el correspondiente valor de v_u es pequeño y por lo tanto la ponderación del punto en la Ecuación (13) disminuye y su relevancia para la localización del objeto queda reducida también.

Además es necesario actualizar el modelo de fondo dinámicamente para dotar al tracker de mayor robustez frente a los problemas que pueden surgir. Para ello, en primer lugar se calculan las características del fondo en el primer frame. Después, mediante los coeficientes Bhattacharyya se define la semejanza entre el modelo de fondo antiguo y el actual. Si este valor está por debajo de un umbral, esto quiere decir que hay cambios considerables en el fondo, y por tanto se actualiza tanto el modelo de fondo como el coeficiente. Finalmente se realiza la transformación del modelo de objeto empleando la Ecuación (13) y el coeficiente actualizado.

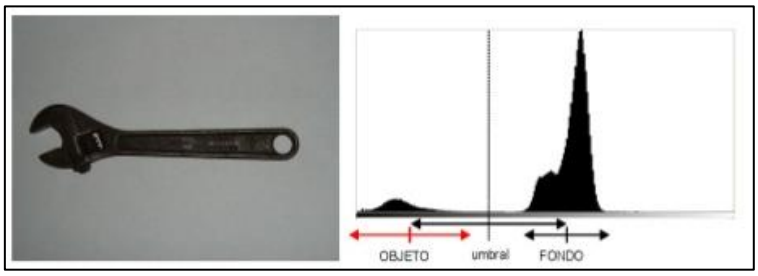


Figura 3.3: Ejemplo de un objeto representado mediante su histograma frente – fondo correspondiente. Extraído de [62].

3.3.2. Funcionamiento

El funcionamiento del tracker CBWH es exactamente el mismo que para el tracker MS (Sección 3.2.2), con una única novedad [40]. Inicializa el modelo de objeto, y tras ello, también el modelo del fondo. Y antes de comparar el histograma del modelo de fondo con el del frame actual, transforma el modelo del objeto con el modelo del fondo, siendo el resultado de esta transformación el que finalmente será comparado en el actual (ver Figura 3.3. y Figura 3.4).

Figura 3.4: Varios frames de una secuencia cuyo algoritmo de seguimiento es el BWH. Como se puede apreciar, en el frame 25 ya se deja de detectar al objeto. Extraído de [40].

Figura 3.5: Varios frames de una secuencia cuyo algoritmo de seguimiento es el CBWH descrito en esta sección. En este caso, desde el frame inicial y hasta el frame 115, el tracker sigue al objeto correctamente. Extraído de [40].

3.4. Color Tracker (COT)

3.4.1. Descripción

Color Tracker nace con el claro objetivo de mejorar el funcionamiento del seguimiento de objetos visual. Pone de manifiesto, que a diferencia de lo que pasa en el tracking, que la mayoría de los progresos se han limitado a transformaciones de espacios de colores simples, en el campo del reconocimiento y detección de objetos, la combinación de características de colores complejos con la luminosidad han dado mejores resultados. Sin embargo, esto no es tarea fácil debido a numerosos problemas que aparecen como sombras, cambios de iluminación, degradaciones… La selección de este algoritmo prosigue la línea iniciada por MS y por CBWH, la de trackers que han mostrado buenos resultados para ciertas condiciones, pero que tratan de seguir progresando con nuevos desafíos. Además, uno de los aspectos que más llama la atención de este algoritmo, es que su desarrollo se centra en algo tan básico y a la vez tan importante como es el color. Una característica que puede hacer que algoritmos que se basen en ello consigan grandes resultados, pero que a su vez, puede volverse en contra y generar problemas complejos de tratar. De ahí viene la motivación por este algoritmo, la de conocer y estudiar diversas maneras para seleccionar las cualidades del color de la mejor forma posible para el seguimiento de objetos en vídeo a largo plazo.

A pesar de que varias características del color han mostrado muy buenos resultados en la detección y reconocimiento de objetos, en muchos trackers el uso de la información del color se ha limitado a simples transformaciones. Quizás un motivo de esta circunstancia puede ser la dificultad que implica sacar el máximo provecho de la información del color, ya que propiedad puede sufrir variaciones de luminosidad, sombras, cambios por la cámara y geometría del objeto, entre otros.

Para llevar a cabo este tracker, se ha partido del algoritmo CSK [43, 44]. Este sistema emplea una estrategia de muestreo intenso mientras muestra que el proceso de coger sub-ventanas en un frame induce a una estructura circular. Por su comportamiento competitivo y por ser uno de los algoritmos más rápidos se ha elegido como punto de partida. El tracker CSK, a partir del objeto de una parte de la imagen, emplea un clasificador que aprende mediante la técnica de mínimos cuadrados. La clave para su alta velocidad es que exprime la estructura circular que aparece por la suposición periódica de la parte de la imagen.

Para incorporar la información del color, COT extiende el algoritmo CSK a características del color multi-dimensionales, definiendo un núcleo apropiado. Las características extraídas de una parte de la imagen se representan por la función x:

$$x : \{0,\ldots,M-1\} \times \{0,\ldots,N-1\} \rightarrow R^D$$

dónde x (m, n) es un vector D-dimensional que consiste en los valores de características en la posición (m, n). En el algoritmo CSK, un parte de la imagen en escala de grises es pre-procesada por la multiplicación de esa parte por una ventana Hann. Para el caso del tracker COT, se sigue el mismo procedimiento para cada canal de características. La elección de las características del color es fundamental en este tipo de algoritmos. Otro aspecto que hay que tener en cuenta, es que cada color lleva asociado un nombre. Así pues, se ha concluido que en inglés existen 11 términos básicos para referirse a todos los colores. Esto conlleva a que cada color genera un mapeo RGB que produce una representación en 11 dimensiones. COT realiza una normalización proyectando los nombres de los colores en una base ortonormal de un subespacio de 10 dimensiones. Sin embargo, el coste computacional se convierte en un problema al tener que analizar tantas dimensiones. Para ello, con COT se propone también una técnica de reducción adaptativa de las dimensiones que guarda la información útil mientras va reduciendo el número de dimensiones de color. La fórmula para llegar a este caso, es encontrar una reducción adecuada de las dimensiones mapeando el frame actual, y minimizando la función de coste que sigue la forma:

$$\eta_{\text{tot}}^{p} = \alpha_p \eta_{\text{data}}^{p} + \sum_{j=1}^{p-1} \alpha_j \eta_{\text{smooth}}^{j}. \qquad (14)$$

Siendo η_{data}^{p} datos que depende sólo del frame actual y η_{smooth}^{j} un término de igualdad con el frame j. Estos dos términos son controlados por la ponderación mediante $\alpha_1, \ldots, \alpha_p$.

Mediante esta serie de mecanismos, se ha conseguido mantener la precisión existente en otros algoritmos anteriores, incluso mejorando en algunos casos, mientras se procesa con un ratio de más de 100 frames por segundo. Esto lo convierte especialmente adecuado para aplicaciones en tiempo real como la vídeo vigilancia y seguridad.

3.4.2. Funcionamiento

Aunque la inicialización de este tracker no es exactamente igual a la de los dos anteriores (MS y CBWH, Secciones 3.2.1. y 3.3.2. respectivamente), se produce de manera muy similar. Tras ajustar los distintos parámetros de la configuración que rige COT, el algoritmo busca una secuencia de imágenes y su ground-truth. Además en este caso es necesario también que exista un archivo indicando el primero y el último frame del vídeo seleccionado.

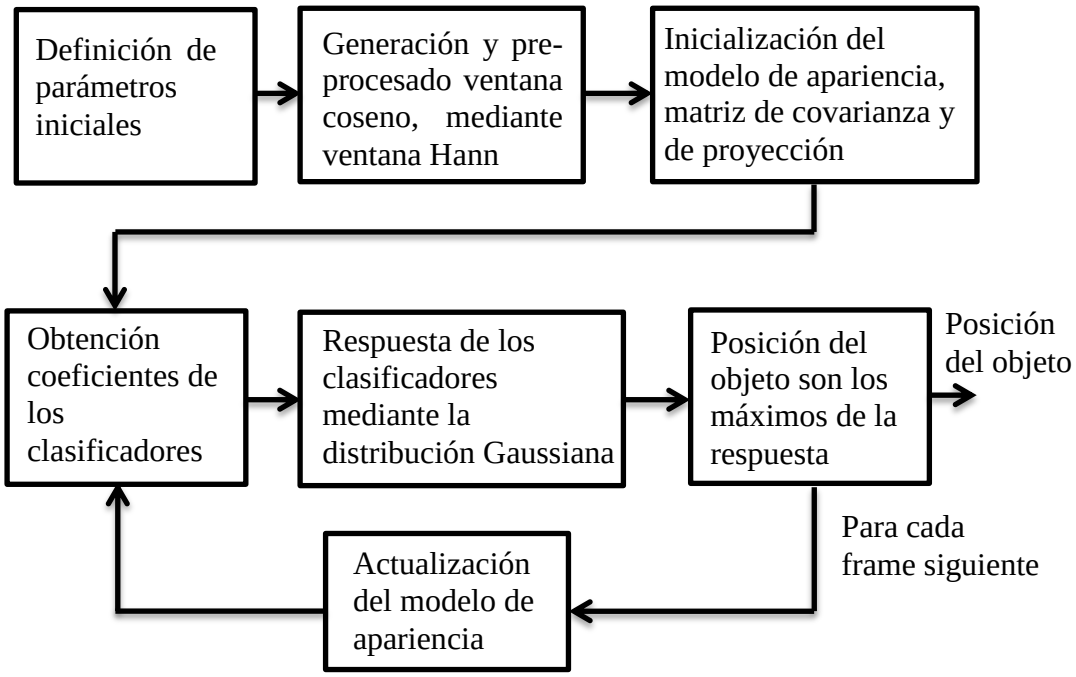


Figura 3.6: Diagrama de bloques que resume el funcionamiento del tracker COT.

El funcionamiento consiste en extraer un mapa de características con el que se entrena a los clasificadores. Inicializa el modelo de apariencia, la matriz de covarianza y la matriz de proyección. Con estas matrices, se obtiene la proyección de nuevas características, que son nuevos coeficientes para los clasificadores. Se calcula la respuesta a estos clasificadores mediante una Gaussiana (con desviación σ) (Figura 3.7) y la nueva localización del objeto se consigue con los valores máximos de la respuesta. Para el resto de la secuencia se repite el proceso actualizando el modelo de apariencia.

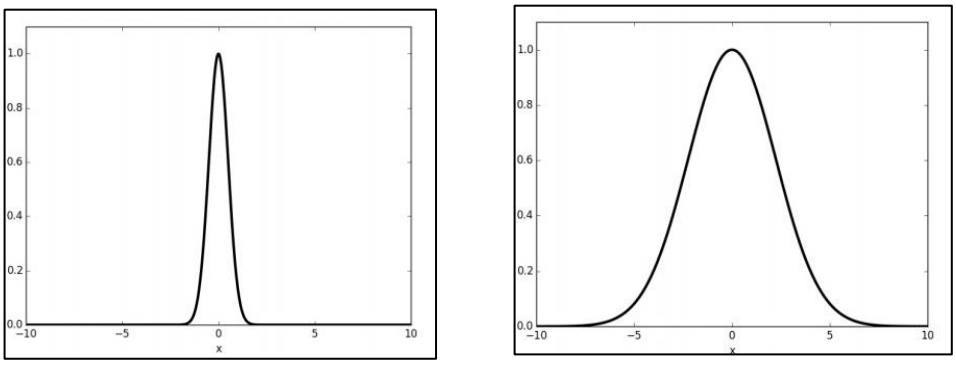


Figura 3.7: Varias representaciones de posibles distribuciones Gaussianas empleadas en el funcionamiento del tracker COT [50].

3.5. Active Feature Selection (AFS)

3.5.1. Descripción

El algoritmo AFS centra su atención en la parte, como su nombre indica, de la selección y extracción de las características. A pesar de que ya se han realizado trabajos de mejora en este módulo, el tracker AFS propone seguir avanzando y mejorando ciertos aspectos. Hasta este punto, muchos de los problemas de otros algoritmos relacionados con esto, era que cuantas más características eran seleccionadas, la probabilidad de que fuesen menos discriminativas, era también más alta, y esto puede derivar en problemas de drift. Tras seleccionar algoritmos que se centraban en la parte más inicial de todo el proceso de seguimiento, una de las razones para elegir el tracker AFS fue que pone su énfasis de mejora en otro aspecto distinto a los anteriores. En este caso para conseguir un modelo de apariencia del objeto discriminativo, desarrolla una nueva forma de actualizar los clasificadores. Además se ha probado que este algoritmo responde bien a largos cambios de apariencia, otro punto que se podrá comprobar con las secuencias de largo plazo de este proyecto.

El tracker Active Feature Selection (AFS) toma como origen el algoritmo MIL (Multiple Instance Learning). Este sistema se propone como una posible mejora al problema del drifting provocado por la pérdida de precisión en la detección del objeto y en las muestras extraídas. Para ello emplea muestras tanto positivas como negativas, y selecciona algunas características mediante la maximización de la función de probabilidad. Sin embargo, esta técnica de selección puede aportar al objeto características con menor información. Además para construir un clasificador lo suficientemente discriminativo es necesario aportar un amplio número de características, lo que hace que el coste computacional aumente. Y al seleccionar muchas características, es más probable que haya algunas que sean menos discriminativas, por lo que esto degradaría la actuación del algoritmo a lo largo del tiempo.
Para tratar de poner una solución a esta serie de desafíos que deja MIL, se propone el tracker AFS. Se inspira en el método de aprendizaje activo [46] que consiste en obtener un algoritmo que sea capaz de lograr una mayor precisión con un número menor de etiquetas, si se le permite escoger los datos de los cuales tiene que aprender.

Hay dos componentes importantes en el tracker AFS, que son, cómo detectar la localización del objeto en el nuevo frame y el otro es como actualizar el clasificador.

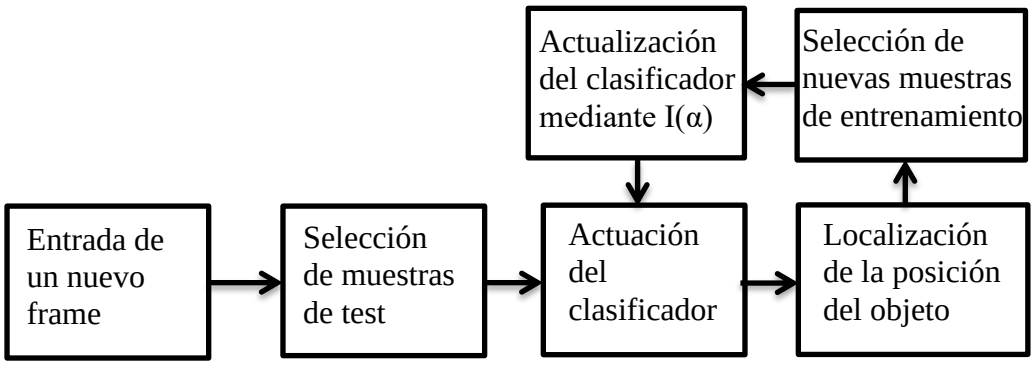


Figura 3.8: Diagrama de bloques que resume el funcionamiento del tracker AFS. Generado a partir de [30].

El planteamiento que realiza AFS para actualizar el clasificador es optimizar una función de información Fisher en lugar de la función de probabilidad. Para la deducción del clasificador H(x), se necesitan estimar sus correspondientes parámetros de ponderación α. Según la inecuación de Cramer – Rao [51], muestra que para cualquier elemento de estimación imparcial t_n de α, basada en n independientes e idénticas muestras distribuidas de la probabilidad $p(y|\alpha)$, la covarianza de t_n debería satisfacer que $cov(t_n) - \frac{1}{n} I(\alpha)^{-1}$ es una matriz definida no negativa, donde $I(\alpha)$ es la matriz de información Fisher [51] definida como:

$$I(\alpha) = -\int p(y|\alpha) \frac{\partial^2}{\partial \alpha^2} \log p(y|\alpha) dy \qquad (15)$$

La matriz de información de Fisher representa un conjunto de incertidumbre del modelo de clasificación, el cual a menudo se usa en el método de aprendizaje activo [46]. Minimizando la traza de la matriz de información de Fisher, tr $(I(\alpha))$, se pueden seleccionar las características que contienen mucha más información que las que se seleccionan con el criterio de MIL debido a que el criterio del algoritmo de AFS maximiza la incertidumbre de las características seleccionadas. Así solo se necesita escoger un número pequeño de clasificadores débiles, que son más discriminativos que los empleados en MIL. La proporción aproximada es de 15 sobre para un conjunto de 50 en AFS, mientras que en MIL se seleccionarían 50 candidatos de unos 250.

3.5.2. Funcionamiento

El tracker AFS necesita conocer el estado inicial del objeto antes de comience su ejecución. Una vez indicado esto, inicia el ajuste y configuración de los correspondientes parámetros. Estima la plantilla de características y la plantilla de muestras, que por un lado calculará las muestras positivas y por el otro las negativas. A continuación se extraen las características, por medio de un filtro Haar (Figura 3.8). Se obtienen las muestras positivas y negativas mediante los clasificadores y se propone la localización del objeto en el primer frame. Tras esto, los clasificadores pasan a una etapa de entrenamiento. Para el siguiente frame, se calculan los nuevos clasificadores. Se realiza una combinación lineal entre los débiles y los más fuertes. Una vez hecha esta combinación, se detectan y se guardan las posiciones del objeto de interés. Y finalmente se actualizan las características y los clasificadores, para repetir así el proceso con todos los frames restantes de la secuencia.

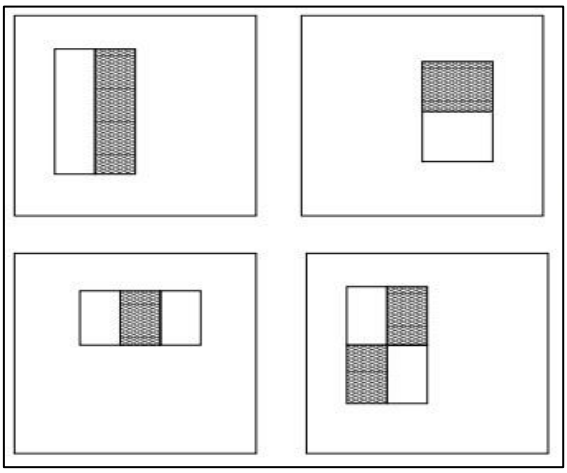


Figura 3.9: Ejemplos de filtros Haar usados para seleccionar distintas regiones de características. Extraído de [52].

3.6. Active Example Selection Tracker (AEST)

3.6.1. Descripción

Partiendo de la división entre trackers generativos, aquellos que representan un objeto en un espacio particular de características, y los trackers discriminativos, que son los que realizan una clasificación binaria entre el objeto y el fondo, el algoritmo AEST concentra sus esfuerzos en estos últimos, y más concretamente, en el aprendizaje de un clasificador que sea capaz de capturar cambios de apariencia en el seguimiento de objetos.
La motivación de este tracker se puede englobar en la del tracker anterior, AFS, ya que dentro del amplio espectro de opciones en el seguimiento de objetos, ambos están focalizados en la parte concreta del aprendizaje del clasificador, si bien AEST emplea otra alternativa para ello. Supone interesante entonces ver cuál es la diferencia final del resultado entre todos los trackers seleccionados, y especialmente, entre el algoritmo AFS y el AEST.

Para empezar a tratar el problema del entrenamiento de las muestras o ejemplos, proponen una selección activa de ejemplos que automáticamente seleccione los ejemplos con mayor información para el aprendizaje del clasificador [48].

Basado en esta estrategia, presentan un marco nuevo para los trackers discriminativos en el cual emparejan los objetivos de la selección de ejemplos con el del aprendizaje del clasificador. Para aportar robustez al aprendizaje, emplean Laplacian Regularized Least Squares (LapRLS) [47]. De esta forma puede usarse los datos sin etiquetar, que pueden ser recogidos fácilmente para clarificar el clasificador (ver Figura 3.9). Por otro lado, al emplear una estrategia de etiquetado conservadora, la adaptabilidad del tracker aumenta. Con estos dos mecanismos se consigue dotar al algoritmo de mayor efectividad y robustez.

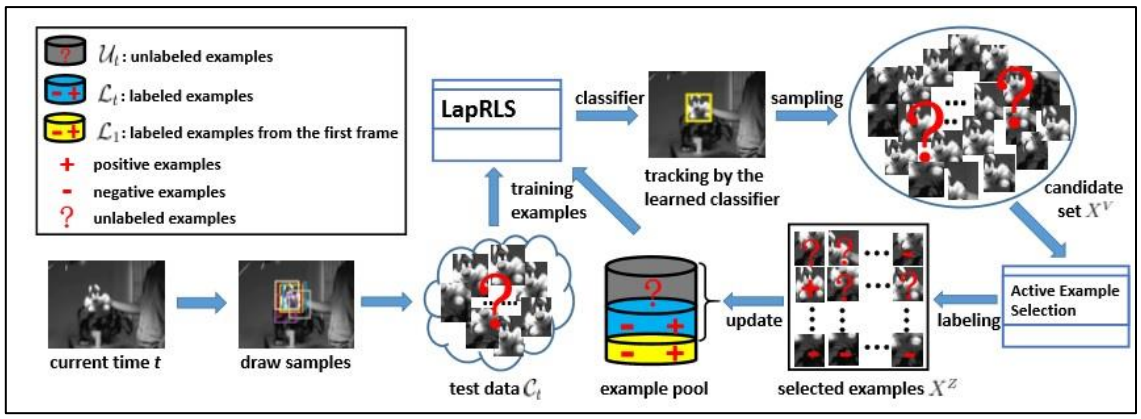


Figura 3.10: Esquema general de un algoritmo que emplea AEST. LapRLS es empleado para el aprendizaje de un clasificador que es capaz de aprovechar tanto los datos etiquetados como los no etiquetados durante el tracking. La etapa de AEST se introduce entre el muestreo y el etiquetado. Extraído de [48].

Por lo tanto, se exponen a continuación los distintos pasos por los que pasa el algoritmo descrito en [48] durante su ejecución de la secuencia o vídeo:

- Entrada: secuencia de imágenes $o_1, \dots, o_T$, una bounding box inicial en el primer frame para indicar el objeto de interés, regularización de los parámetros λ_1y λ_2; una función kernel K.
- Salida: resultados del tracking $\{s_t\}_{t=1}^{T}$

1) Inicialización: $s_1$es estado del objeto inicial; L_1y U_1 son los ejemplos etiquetados y sin etiquetar del primer frame, respectivamente.
2) Para $t = 2, \dots, T$ se hace:
3) Genera muestras s_t^i , extrae características x_t^i , conjunto de datos de test: $C_t = \{x_t^i\}_{i=1}^{N_s}$
4) Conjunto de entrenamiento de etiquetado $(X^L, y) \leftarrow L_t \cup L_1$, conjunto de entrenamiento de sin etiquetar $X^U \leftarrow U_t \cup C_t$, función de aprendizaje f_t con el uso de LapRLS.
5) Selecciona el resultado del tracking s_t , usando $s_t = \arg\max p(o_t|s_t^i)$
6) $X^V \leftarrow n$ ejemplos sin etiquetar generados por muestreo, $X^{Z'} \leftarrow X^L$, $X^V \leftarrow X^V \cup X^L$, selecciona m ejemplos X^Z mediante AEST;

7) Estima las etiquetas de X^Z como se describe en la sección III-C de [48], $L_t U_t \leftarrow X^Z$, actualiza el conjunto de ejemplos de entrenamiento.
8) Finaliza el bucle.

3.6.2. Funcionamiento

Al igual que sucedía con el tracker AFS (Sección 3.5.2.), en primer lugar AEST requiere conocer el estado inicial del objeto, el número de muestras, el formato del frame de entrada y algunos parámetros de las distribuciones (controlan aspectos como el tamaño y la escala de las muestras candidatas). Después de ajustar una amplia gama de parámetros, inicializa el etiquetado y los datos sin etiquetar. Calcula los parámetros para la realización de las LapRLS. A continuación carga un nuevo frame y procesa mediante un filtro los candidatos. Una vez obtenido esto, calcula una serie de medidas entre los ejemplos y los nuevos candidatos, para pasar a la realización del modelo del objeto y estimar de esta forma la nueva localización del objeto. Para finalizar, recoge los ejemplos y actualiza la colección de éstos. Vuelve a iniciar el proceso desde el punto que cargaba un nuevo frame para el resto de la secuencia. Durante la ejecución de este algoritmo, se ha podido comprobar que tiene un coste computacional muy alto en comparación con los vistos anteriormente en este capítulo.

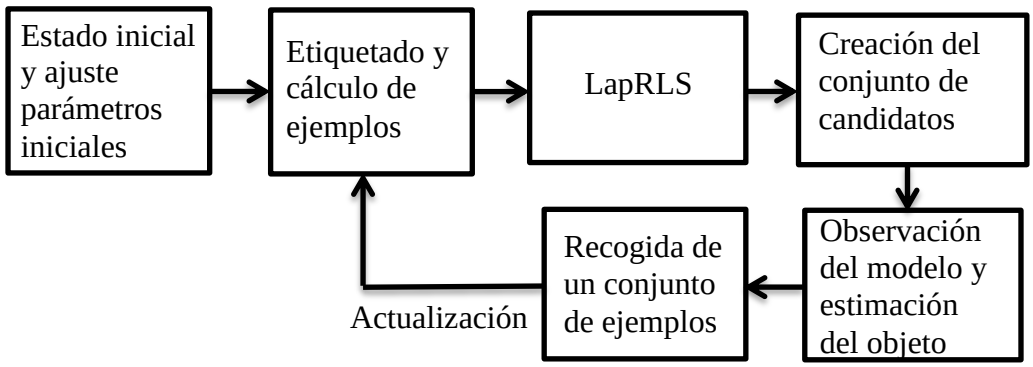


Figura 3.11: Diagrama de bloques que resume el funcionamiento del tracker AEST.

3.7. Multiple Experts using Entropy Minimization (MEEM)

3.7.1. Descripción

El tracker MEEM nace con el objetivo claro de reducir los problemas de drifting en los trackers online y los problemas que surgen a la hora de actualizar el modelo del objeto, cuando se dispone de la posición inicial del objeto y de observaciones previas, dentro de un entorno de seguimiento y detección. En una gran mayoría de estos trackers, se pretende tener en cuenta cambios de apariencia en el objeto con la actualización del modelo, pero este proceso también puede generar problemas de drift.

Esto ocurre por fallos del tracker, oclusiones o un mal alineamiento de las muestras de entrenamiento que pueden llevar a una actualización del modelo errónea. Algunas propuestas para solventar este desafío han sido incorporar la plantilla del primer frame o un conocimiento previo en el proceso de actualización del modelo [9]. Otros algoritmos [23] han desarrollado una especie de mecanismo de censura por el cual se evita una actualización del modelo cuando se cumplen ciertos criterios. Sin embargo el inconveniente principal de esto es que cuando ese mecanismo falla, los trackers pierden la capacidad de evolucionar porque el modelo de objeto solo puede evolucionar hacia delante sin posibilidad de corregir los errores pasados.
Los principales motivos de la selección de este algoritmo son que se trata de un tracker relativamente novedoso, que emplea una base para el clasificador, SVM muy usada en el seguimiento de objetos y que además ha sido estudiado en nuevos dataset de secuencias con desafíos comunes en tracking.

Lo que se propone con el algoritmo MEEM [49] es una fórmula capaz de corregir los efectos de malas actualizaciones del modelo después de que ello suceda. Es aquí donde se introduce el marco del seguimiento multi-expertos, constituido por un tracker discriminativo y sus frames de formación. El mejor experto es seleccionado en base a un criterio de mínima pérdida. Este criterio es desarrollado mediante la función de optimización de la entropía.
Para deducir una adecuada función de pérdida, se emplea una formulación que fue originalmente desarrollada por el problema de aprendizaje del semi-supervisado etiquetamiento parcial (PLL) [53]. En [53], el problema PLL se soluciona con un marco de referencia MAP que maximiza la posterior probabilidad del modelo parametrizado por θ,

$$\mathcal{C}(\theta, \lambda; \mathcal{L}) = L(\theta; \mathcal{L}) - \lambda H_{emp}(Y|X, Z; \mathcal{L}, \theta), \tag{16}$$

Donde $L(\theta; \mathcal{L})$es la probabilidad de los parámetros θ del modelo, y $H_{emp}(Y|X, Z; \mathcal{L}, \theta)$ es la entropía empírica condicional de las clases de etiquetas condicionadas por los datos de entrenamiento y por los posibles conjuntos de etiquetas. El parámetro escalar λ controla la compensación entre la probabilidad y la técnica anterior.
En el caso del tracker MEEM, para usar el criterio de mínima entropía se propone una nueva formulación del problema de seguimiento por detección en un conjunto de múltiples ejemplos PLL. La definición para la entropía final queda:

$$H(\mathbf{y}|\mathbf{x}, \mathbf{z}; \theta_E) = \sum_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \mathbf{z}; \theta_E) \log P(\mathbf{y}|\mathbf{x}, \mathbf{z}; \theta_E) \tag{17}$$

Cada uno de los distintos parámetros quedan descritos en [49].

Además el tracker MEEM se basa en el algoritmo de SVM (support vector machine) (Figura 3.12), el cual utiliza un prototipo establecido para gestionar las muestras de entrenamiento, y una técnica de mapeo de características explícita para lograr una eficiente actualización del modelo.

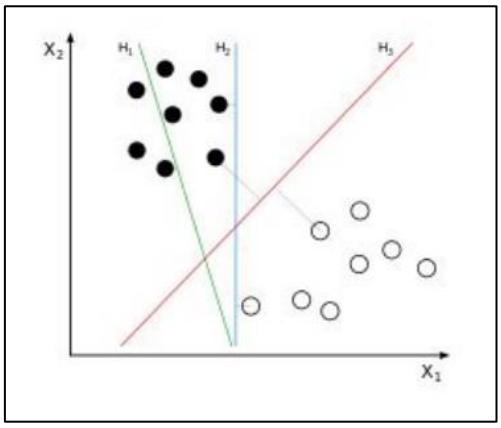


Figura 3.12: Clasificación según el algoritmo SVM. H_1 no separa las clases. H_2 sí que lo hace pero con un margen pequeño. H_3 los clasifica con el margen máximo. Extraído de [54]. Mejores resultados en color.

3.7.2. Funcionamiento

Este algoritmo arranca cargando el primer frame de la secuencia aunque es necesario indicar que como parámetro se debe pasar la posición inicial del objeto, el formato que tenga la imagen y la carpeta donde se encuentren almacenadas todas las imágenes. Después de inicializar sus parámetros correspondientes, genera las imágenes de características. A continuación empieza el seguimiento mediante el tracker SVM. Una vez hecho esto, se procesa el tracker de multi-expertos, calculando el criterio de la entropía mencionado anteriormente, para hallar de esta forma la posición del objeto. Finalmente se actualiza el clasificador SVM para que esté disponible para el siguiente frame.

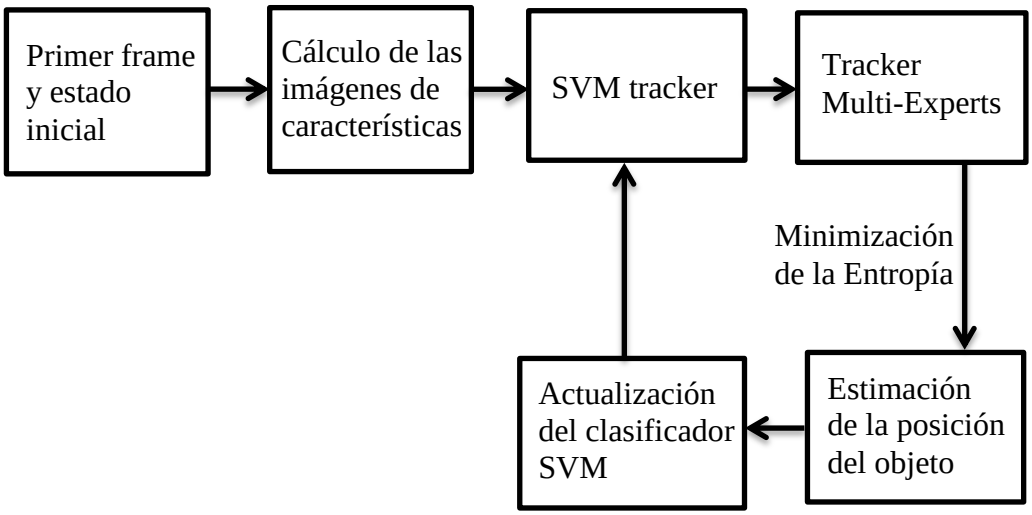


Figura 3.13: Diagrama de bloques que resume el funcionamiento del tracker MEEM.

3.8. Resumen de los algoritmos seleccionados

A continuación se muestra la Tabla 3.1. que contiene algunas de las características más importantes que se han mencionado en los trackers descritos en la sección anterior. El objetivo es englobar toda la información posible de cara a que su consulta sea más rápida y visual.

Tracker	*Tipo*	*Características*	*Modelo*	*Mecanismo de adaptación*	*Ventajas*	*Inconvenientes*
MS	DET	Representación del objeto y de las regiones candidatas	Información del fondo más proceso iterativo	Coeficiente de Bhattaharyya	Sencillo. Eficiente	Drifting. Manejo de oclusiones
CBWH	DET	Versión de mejora de MS. Sólo transforma modelo del objeto.	Información del fondo y actualización del mismo.	Coeficiente de Bhattacharyya	Reduce interferencias de fondo. Robustez ante mal inicio.	Drifting
COT	PRO	Uso de las propiedades del color como base para el tracking	Clasificador y características multidimensionales del color	Técnica de reducción adaptativa de las dimensiones de los atributos del color	Rapidez. Competitividad frente a cambios de luz, deformaciones.	Cambios de escala. Salidas del objeto del plano visual. Baja resolución.
AFS	PRO	Especial atención en extracción y selección de características	Modelo de apariencia discriminativo. Clasificador.	Optimización de la función de información de Fisher.	Manejo de cambios largos de apariencia. Precisión	Mayor coste computacional.
AEST	PRO	Nuevo marco de aprendizaje del clasificador con los ejemplos.	Aprendizaje activo. Modelo de objeto y observación	Selección de ejemplos con más información por LapRLS	Manejo de drift. Efectivo	Elevado coste computacional
MEEM	PRO	Corrección de malas actualizaciones del modelo, tras suceder	Multi-expert tracker más lineal SVM tracker	Minimización del criterio de la entropía	Ajuste correcto del drifting	Cambios de escala

Tabla 3.1: Resumen de las características más importantes de los trackers seleccionados y descritos en el Capítulo 3 de esta memoria. DET = Determinista, PRO = Probabilista.

Capítulo 4

Algoritmo propuesto

En este capítulo se propone un algoritmo para el seguimiento de objetos en vídeos, que puede ser empleado tanto con secuencias de corto plazo como con vídeos a largo plazo. Su principal objetivo es proporcionar una alternativa de combinación de los algoritmos seleccionados para conseguir mejorar el rendimiento final para cada secuencia. Para ello se emplea un mecanismo de fusión de los trackers, como se detallará en próximos apartados de este capítulo. En primer lugar se proporciona una breve introducción a los mecanismos de fusión de trackers existentes en la actualidad (sección 4.1). Posteriormente se explica una visión general del algoritmo propuesto (sección 4.2), para finalizar describiendo los pasos realizados para su implementación (sección 4.3).

4.1. Marco de fusión de trackers

Actualmente, dado el frecuente uso del tracking en un gran número de situaciones y aplicaciones de la vida real, son muchos los mecanismos y avances que han ido apareciendo para lograr disminuir los inconvenientes que puedan aparecer o para mejorar lo ya establecido. Sin embargo, al tratarse de un campo tan abierto y con tantas posibilidades, aún queda mucho por solucionar y estudiar.

Cada tracker funciona correctamente para diferentes secuencias, en función de sus características más fuertes y sus debilidades, y en función de los distintos problemas que pueda acarrear el vídeo correspondiente. Un algoritmo considerado como bueno en un determinado conjunto de secuencias puede generar un resultado desfavorable en un vídeo en el cual, otro algoritmo calificado como malo en ese conjunto de secuencias, produce un buen resultado. De este caso, entre muchos otros, es posible derivar la teoría de que combinando las fortalezas de una serie de algoritmos se pueden evitar, o al menos en el peor de los casos, reducir los inconvenientes de éstos para lograr mejores resultados en los distintos vídeos empleados. Esto es lo que se denomina a partir de este momento como fusión de trackers o algoritmos. A pesar de ser una estrategia relativamente nueva, ya han aparecido diversas técnicas de fusionado para obtener mejores rendimientos, como se describen en [55, 56 y 57].

Una primera clasificación de las técnicas de fusión de algoritmos, según se propone en [55], puede dividirse en activos o pasivos, en función de si se emplean o no, mecanismos de realimentación de los propios trackers de seguimiento. Los algoritmos de fusión activos necesitan estrategias de seguimiento que tengan en cuenta tipo de enfoques. Algunos de estos trackers son PROST [57] y VTS [59]. A medida que aumente el número de algoritmos, la fusión activa puede volverse muy compleja debido a que el número de especificaciones a considerar también se ve incrementado.
Por otro lado, en lo referido a la fusión pasiva, es aquella en la que se trabaja con algoritmos arbitrarios cuya duración dependerá de las salidas que sean compatibles con la propuesta de la fusión. En este contexto se desarrolló el tracker descrito en [58] y cuyo objetivo fundamental era mantener la entrada necesaria del usuario al algoritmo lo más pequeña posible.

Otras formas de separar los mecanismos de fusión de trackers son en función de su arquitectura y en dos niveles de fusión diferentes [1]. Respecto a su arquitectura, se puede diferenciar fusión de algoritmos en paralelo o secuencial. En el primer caso, cada tracker se ejecuta independientemente y el resultado de la fusión es el algoritmo que más precisión obtenga, o en su caso, la mejor combinación de varios. Por su parte, la arquitectura secuencial se realiza uno a uno, es decir, primero actúa un algoritmo que si consigue un buen resultado, este resultado se establece como el resultado de la fusión.
La clasificación por niveles se rige por fusión de trackers o por fusión de medidas [56]. El primer caso corresponde a las fusiones de algoritmos en las que se trabaja con las salidas obtenidas de cada tracker, teniendo en cuenta un modelo de características individual. Mientras que la fusión de medidas emplea múltiples características, que son combinadas internamente por los diferentes algoritmos, como por ejemplo, que tengan que analizar el color y los bordes. Para ello se pueden basar en métodos como filtrado de partículas, el modelo de apariencia o en el modelo de observación.

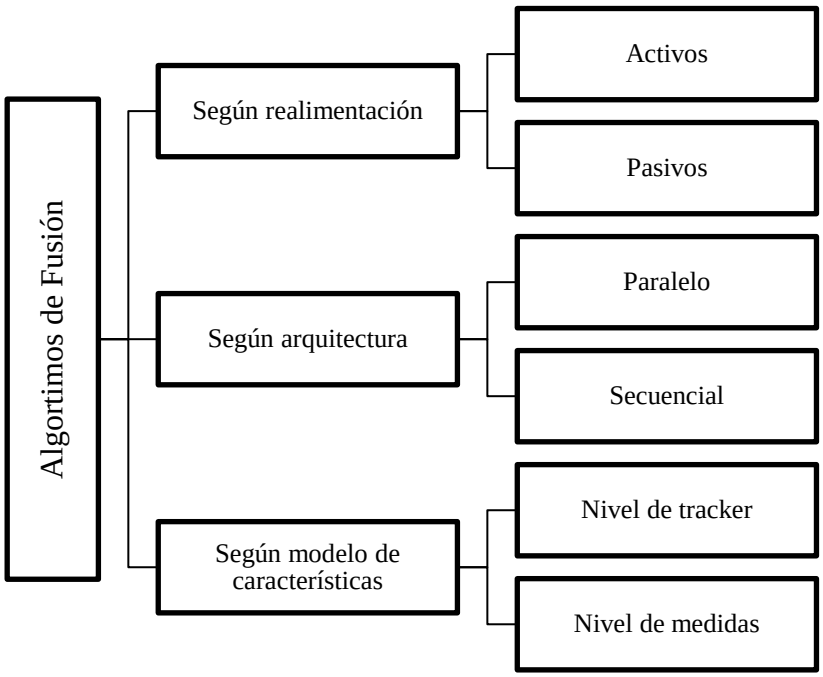


Figura 4.1: Esquema de las diferentes clasificaciones de los algoritmos de fusión.

4.2. Estructura del algoritmo propuesto

Se propone realizar un algoritmo de seguimiento de objetos en vídeo que proporcione un mejor rendimiento y resultados más favorables a los obtenidos con los distintos trackers estudiados en este PFC, basándose principalmente en el mecanismo de fusión de trackers [55]. Dentro del gran abanico de posibilidades existente, se ha optado por el mecanismo de emplear las salidas resultado de los diferentes trackers, y más concretamente, el bounding box resultado. Se ha escogido este método por su simplicidad en relación a los elementos de entrada y porque los diferentes algoritmos seleccionados son independientes. Es posible realizar la combinación con el número de trackers que se quiera, en función de las características o problemas de la aplicación final.

Sigue un esquema formado por tres etapas (ver Figura 4.2): en la primera de ellas se almacenan los bounding box (bbox) individuales de cada tracker; en la segunda se lleva a cabo la combinación o fusión entre ellas según el proceso establecido, que se desarrolla en profundidad en la siguiente sección 4.3.2; en la tercera fase se procede a estimar el resultado más óptimo del bounding box obtenido de la combinación.

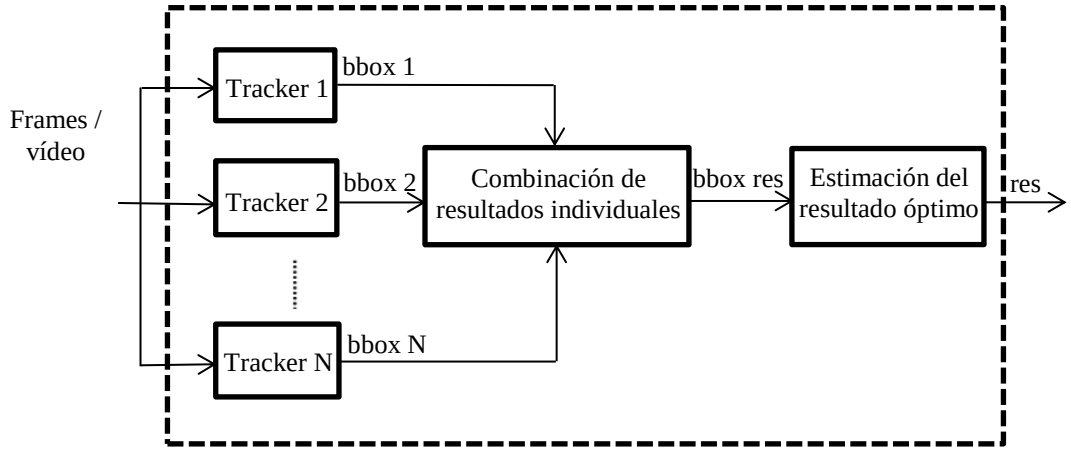


Figura 4.2: Diagrama de bloques del algoritmo propuesto.

En el resto de este capítulo se describe cada etapa del tracker propuesto mencionada anteriormente para estudiar su desarrollo y funcionamiento ayudado por ejemplos gráficos. Se hace especial hincapié en la segunda etapa (combinación), puesto que es la fase fundamental de la implementación realizada.

4.3. Desarrollo del algoritmo propuesto

4.3.1. Almacenamiento de los bounding box

En la primera etapa del algoritmo se almacenan los bounding box de cada tracker de forma individual. Las entradas iniciales son los frames correspondientes a la secuencia de vídeo deseada. Tras ejecutarse en los N trackers diferentes, se obtienen N salidas diferentes, que serán las entradas en la siguiente etapa de combinación. Cada resultado del tracker consiste en N bounding box $b_{i,j}$, i ∈ [1…N], una por cada frame i de la secuencia.

Es importante en esta etapa guardar únicamente los valores de los bounding box resultado de cada frame en cada tracker, y no los datos del resultado global del tracker, ya que hay algoritmos como el caso de AEST que además de mostrar el bounding box también representan el centro del objeto al mismo tiempo. Por eso, este enfoque solo se centra en los bounding box resultado.

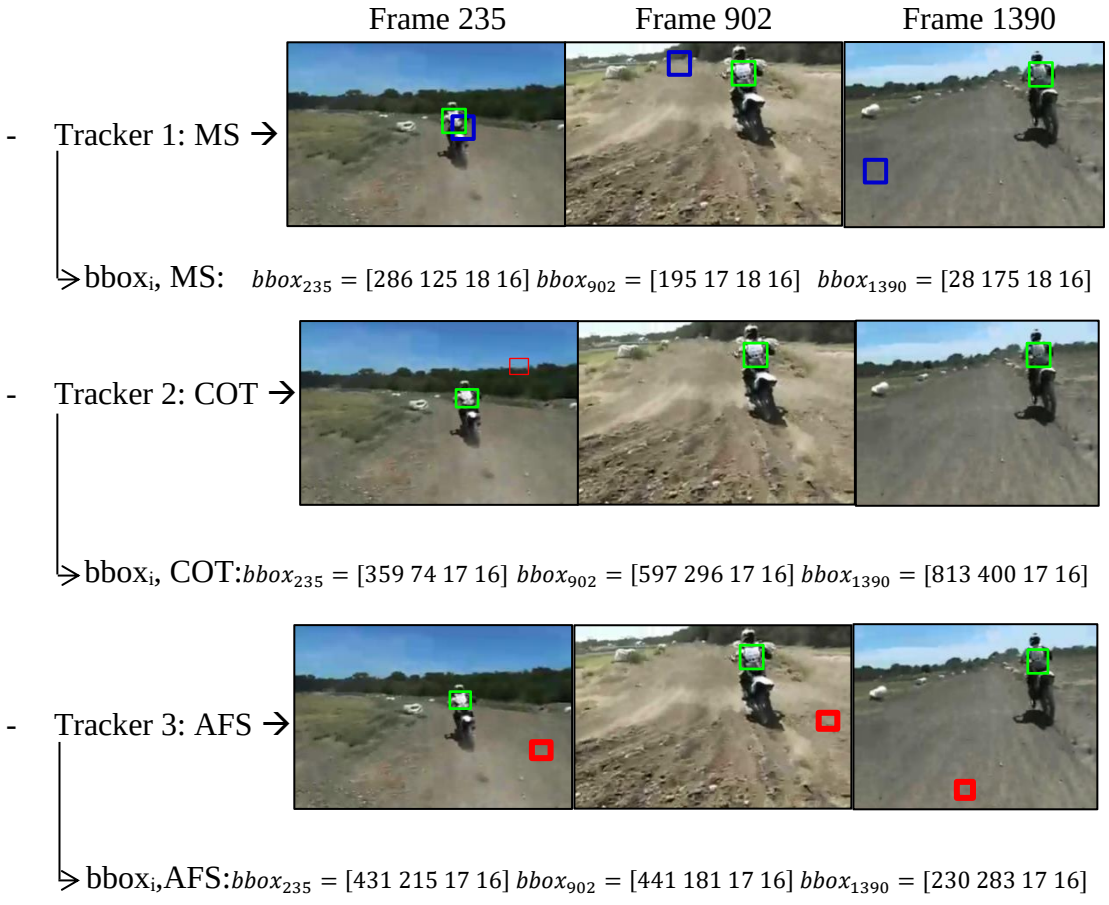


Figura 4.3: Ejemplo de funcionamiento de la primera etapa de almacenamiento a partir de los frames de una secuencia de largo plazo. Cada tracker muestra su bbox correspondiente del color y forma que tiene definido por su algoritmo (el rectángulo azul para MS y el rectángulo rojo para los trackers COT y AFS). En verde se muestra el ground-truth de las secuencias en cada frame. Cuando no aparece el bounding box quiere decir que los resultados no siguen correctamente al objeto en ese frame (caso de COT para los frames 902 y 1390).

4.3.2. Mecanismo de combinación

La segunda etapa consiste en el desarrollo y la integración de los distintos algoritmos para lograr avances y un rendimiento más favorable de todos los posibles trackers como combinación para cada una de las distintas secuencias estudiadas. Como ya se ha descrito en la sección 4.1. de este capítulo, existen múltiples técnicas y posibilidades a la hora de realizar la fusión de los algoritmos.

En el algoritmo que aquí se propone, partiendo de la base realizada en [55], consta de las siguientes etapas.

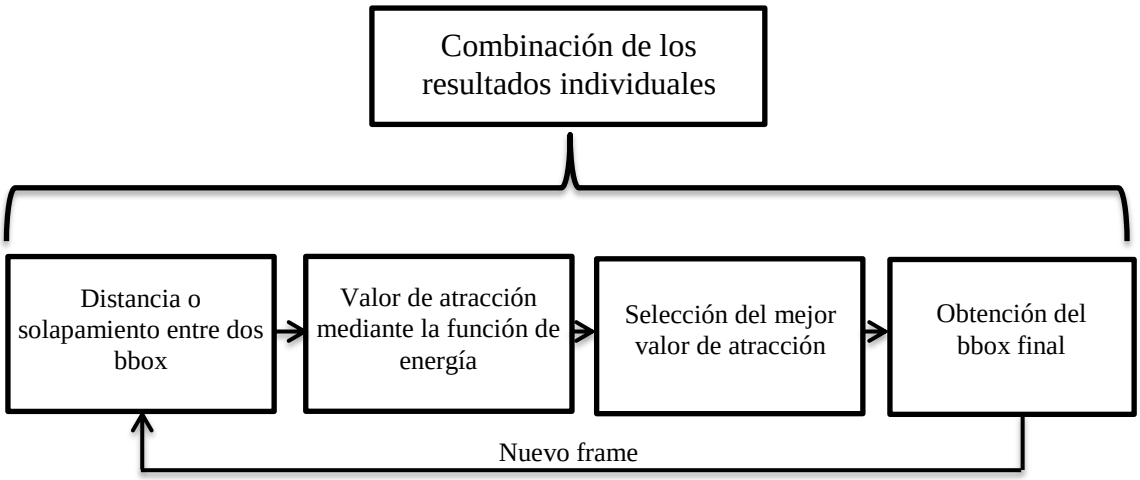


Figura 4.4: Diagrama de bloques del desarrollo del mecanismo de combinación de trackers.

Una vez almacenados cada uno de los bounding box de cada tracker para una secuencia particular, el primer paso del mecanismo de combinación consiste en hallar la distancia (d) o solapamiento (so) entre dos bbox, b y c. Si se escoge la distancia entre dos bounding box, la fórmula que se debe emplear es:

$$d(b,c) = \left\| \left(d_x(b,c), d_y(b,c), d_w(b,c), d_h(b,c) \right)^T \right\|_2 , \qquad (18)$$

Desarrollando cada componente de la ecuación (23) se obtiene:

$$d(b,c) = \left\| \left(2\frac{c_x - b_x}{c_w + b_w}, 2\frac{c_y - y}{c_h + b_h}, 2\frac{c_w - b_w}{\alpha(c_w + b_w)}, 2\frac{c_h - b_h}{\alpha(c_h + b_h)}\right)^T \right\|_2 , \qquad (19)$$

Ecuación 19: Fórmula para calcular la distancia de cada componente entre dos bounding box [55]. Las dimensiones son x e y, posición del objeto, la anchura w, y la altura h. Además, α es una constante que determina la influencia del factor de escala en la distancia.

El empleo de la Ecuación 18 se puede observar en la siguiente imagen (Figura 4.5).

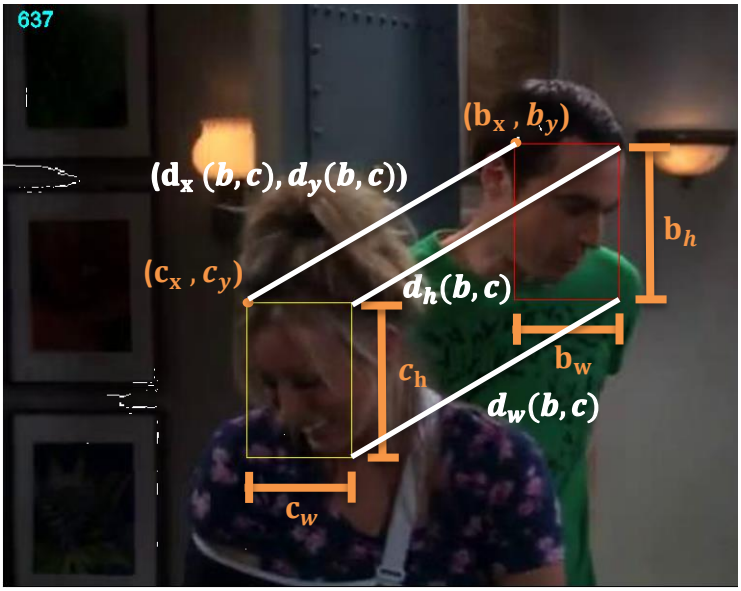


Figura 4.5: Ejemplo gráfico de la Ecuación 18 para un frame cualquiera de una secuencia en particular. Concretamente se corresponde con el frame 637 de la secuencia de largo plazo Sitcom. En este ejemplo se calcularía la distancia entre el bbox de color rojo, b (tracker COT) y el de color amarillo, c (tracker MEEM). Las componentes b_x, b_y, c_x y c_y corresponden al punto que indica la estimación del objeto de cada bbox, b_w y c_w representan la anchura (w) respectivamente, mientras que b_h y c_h equivalen a la altura (h) de cada uno de sus bbox. Esto se realiza con todos los bounding box de cada algoritmo para cada frame.

Por otro lado, para el cálculo del solapamiento entre dos bbox de dos trackers seleccionados se ha procedido de una forma similar al cálculo del FDA (Frame Detection Accuracy) detallado en el Capítulo 2 de esta memoria, salvo con la diferencia de que el solape se calcula entre dos bounding box resultado y no entre una salida y el ground-truth. Así pues, esto queda reflejado en las siguientes ecuaciones:

$$FDA(BB1, BB2) = \frac{Overlap\ Ratio}{\left\lceil \frac{N_{BB1}^{(t)} + N_{BB2}^{(t)}}{2} \right\rceil}, \quad (20)$$

$$\text{Siendo, } Overlap\ Ratio = \sum_{i=1}^{N_{mapped}^{(t)}} \frac{\left| BB_1^{(t)} \cap BB_2^{(t)} \right|}{\left| BB_1^{(t)} \cup BB_2^{(t)} \right|}, \quad (21)$$

Ecuación (20): Fórmula del FDA. Ecuación (21): Fórmula Overlap o solapamiento.

Dónde: - $N_{BB1}^{(t)}$: Objetos detectados en el bounding box 1.
- $N_{BB2}^{(t)}$: Objetos detectados en el bounding box 2.
- $N_{mapped}^{(t)}$: Número de objetos mapeados en el frame t.
- $BB_1^{(t)}$: Indica el bounding box resultado del tracker 1 en el frame t^{th}.
- $BB_2^{(t)}$: Indica el bounding box resultado del tracker 2 en el frame t^{th}.

En la implementación desarrollada se realiza este solapamiento entre cajas, de forma que se cubran todas las posibles combinaciones entre los bbox de todos los trackers, para poder garantizar en futuros pasos la selección más adecuada y que proporcione los mejores resultados en cada frame.

El siguiente paso consiste en obtener los valores de la función de atracción (o de energía) definida mediante la ecuación que se describe a continuación. Para inicializar el desarrollo, se hallan los valores de atracción para cada tracker, de forma que se obtendrán seis resultados distintos.

$$a_i(c) = \sum_{j \in M} \frac{1}{d(b_{i,j},c)^2 + \sigma}, \quad (22)$$

Ecuación 22: Fórmula de la función de atracción para una bbox candidata c, en un frame i. Siendo j los bbox resultados de cada tracker, M los distintos algoritmos seleccionados y σ una constante que no sólo evita la atracción infinita cuando la distancia es cero, sino que también reduce el incremento de la atracción cuando es cercana a cero. El valor de sigma se ajusta de manera independiente, en función de las condiciones y características de cada implementación. Extraída de [55].

En el desarrollo presentado aquí, se busca aquel solapamiento que sea mayor. Por lo tanto, al sustituir en la ecuación 22 la distancia por el solapamiento, cuanto mejor sea el solape, menor será el valor de atracción, que será uno de los objetivos que se quiere lograr. En el caso de trabajar con distancias, el valor de atracción deseado sería el mayor, ya que interesaría trabajar con aquellos trackers que produjeran una distancia menor. Tras hallar cada correspondiente suma de los resultados de un tracker en relación a los demás, se selecciona aquel algoritmo que proporcione el menor valor de $a_i(b)$, como se ha explicado antes. Este será el tracker que pasará a ser b_i en la ecuación 22, y que servirá de referencia para el primer bounding box candidato c.

El último bloque de la implementación comienza con la obtención del bbox c. En este punto existen muchas formas de conseguir nuevos candidatos como pueden ser mediante la media de los dos mejores bbox, por majority voting [58], mediana entre todos los algoritmos, de forma aleatoria limitado por un rango establecido previamente… En este proyecto se ha optado por esta última posibilidad, porque esta opción permite disponer de múltiples parámetros con los que se puede ir variando de forma individual o en conjunto, como pueden ser el valor de la constante σ, la posición (x, y) del objeto, su centro, la anchura o la altura. Además al dotarle de aleatoridad a la fusión, aunque limitada dentro de unos márgenes coherentes a las condiciones de la combinación de los algoritmos, resulta interesante ver cuál será el comportamiento final de la fusión ante el gran conjunto de desafíos que se le pueden presentar y como de favorable será la combinación realizada.
En este caso en particular, el primer criterio que se ha establecido para generar el bounding box candidato para cada frame se basa en generar nuevas posiciones x e y, mediante sencillas operaciones de la coordenada x con el ancho, y de la coordenada y con la altura. Se calcula un nuevo centro desplazado y escalado varios píxels. Estas nuevas posiciones sirven de referencia y se emplean en la creación aleatoria de las nuevas coordenadas del bbox candidato, tanto para x como para y. Una vez obtenido este bounding box para un determinado frame, se vuelve a emplear la ecuación 22 para

obtener de nuevo el mejor valor de atracción posible entre la bbox candidata y el resto de bounding boxes de los demás trackers.

Para finalmente ir almacenando los mejores valores que formen el bounding box final de la fusión, se emplea la selección por gradiente. Este método consiste en ir actualizando el valor de la atracción obtenido en cada iteración del algoritmo, de la siguiente manera. Si el valor actual es menor que el valor obtenido en la iteración anterior, se guarda ese valor de $a_i(c)$. Si por el contrario el resultado es mayor, entonces se descarta y se conserva el valor anterior $a_{i-1}(c)$. De esta forma, se va a ir consiguiendo mejores resultados para el valor de atracción con el paso de las iteraciones, y se va evitar a su vez los casos en los que la generación aleatoria resultase más desfavorable. Por último se vuelve a comprobar cuál es el algoritmo con que se produce mayor solape para generar un nuevo bbox candidato para la siguiente iteración, y repetir el proceso descrito anteriormente.

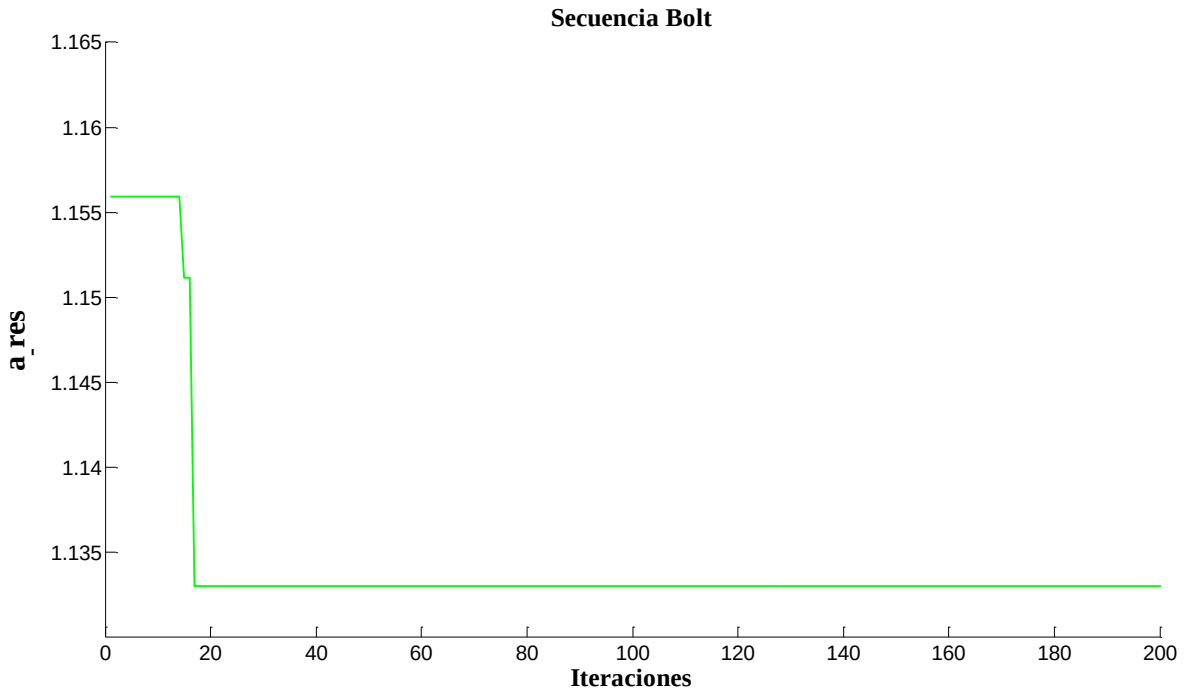


Figura 4.6: Ejemplo del resultado para la secuencia de corto plazo Bolt, del valor de atracción final para cada iteración en función de la técnica de selección por gradiente. En la gráfica se ve como a partir de una determinada iteración (18) los valores obtenidos en el resto son mayores que los anteriores, y por eso siempre se almacena el resultado anterior que es el que obtiene mejores actuaciones, de ahí que para el resto de la secuencia el valor de la atracción permanezca constante.

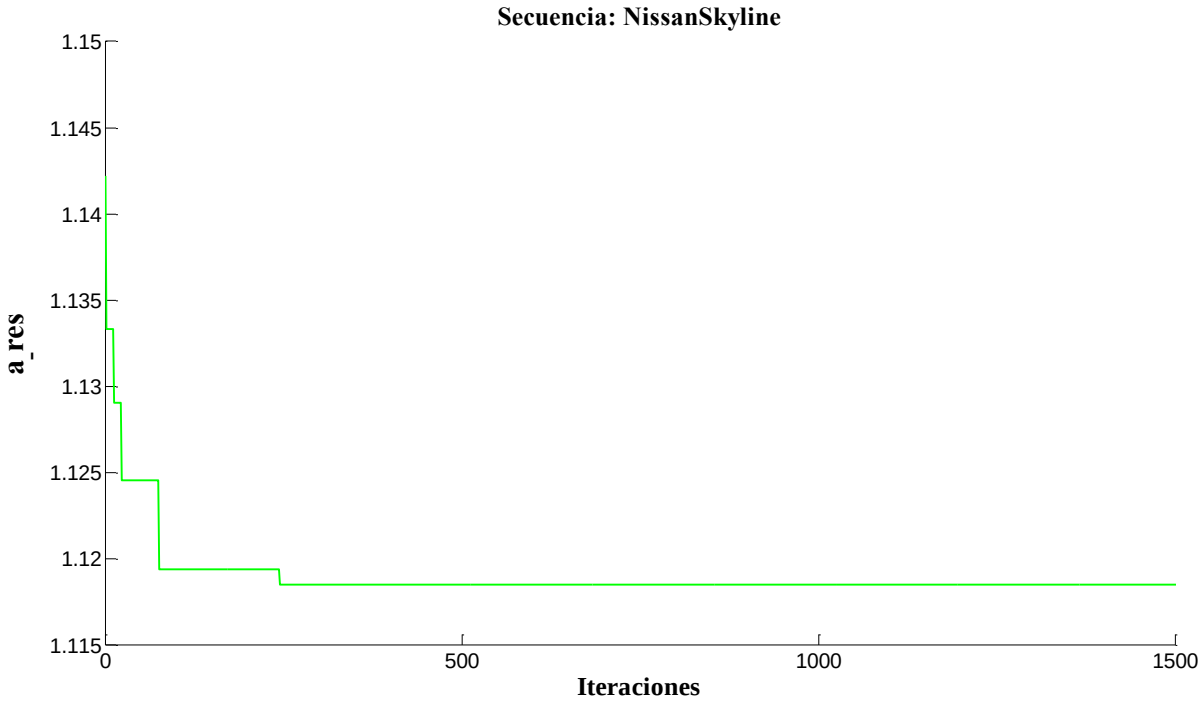
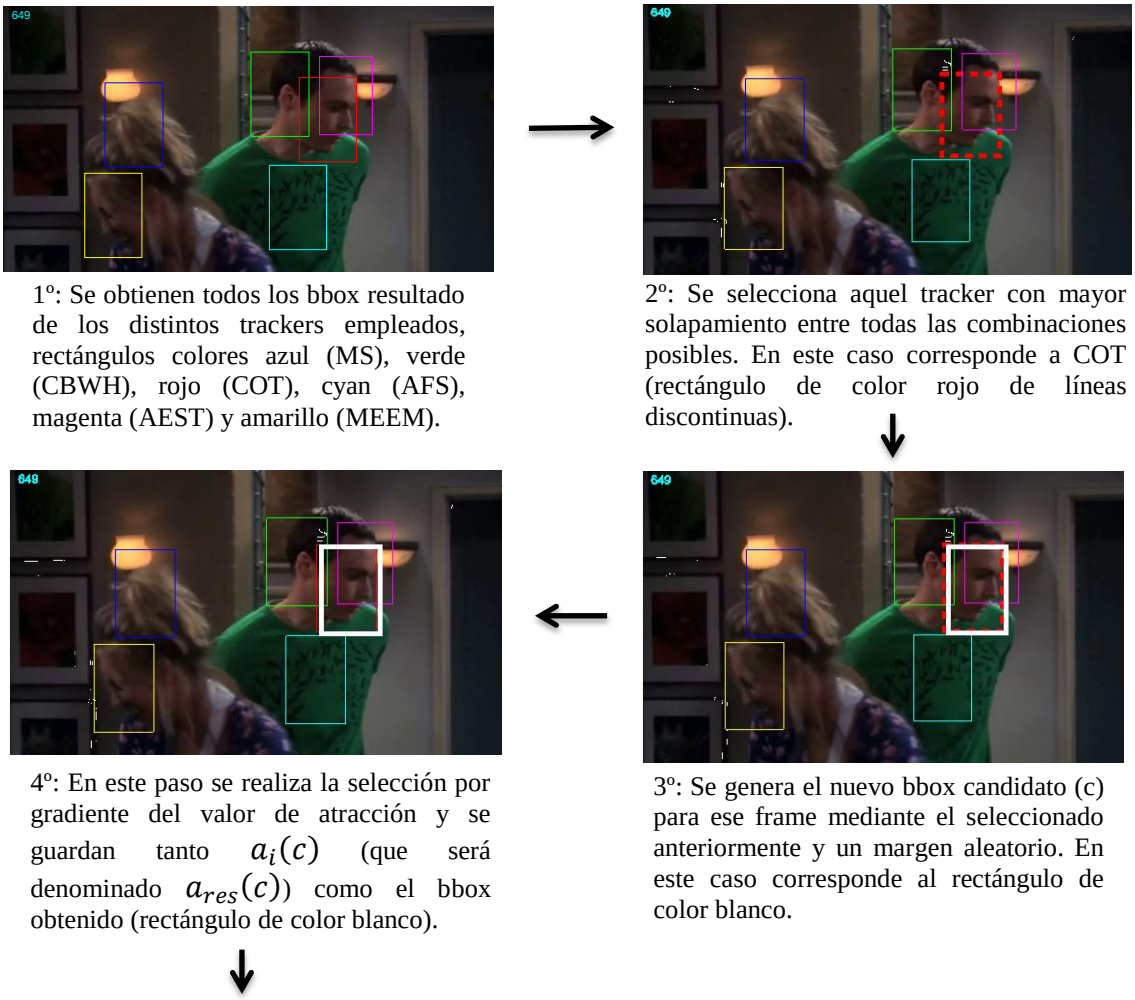


Figura 4.7: Ejemplo del resultado para la secuencia de largo plazo NissanSkyline, del valor de atracción final para cada iteración en función de la técnica de selección por gradiente. En la gráfica se ve como a partir de una determinada iteración (245) los valores obtenidos en el resto son mayores que los anteriores, y por eso siempre se almacena el resultado anterior que es el que obtiene mejores actuaciones, de ahí que para el resto de la secuencia el valor de la atracción permanezca constante.

A continuación se muestra un resumen en forma de ejemplos gráficos para la secuencia de largo plazo Sitcom, de los pasos del funcionamiento del algoritmo propuesto, para facilitar la comprensión del mismo.



1º: Se obtienen todos los bbox resultado de los distintos trackers empleados, rectángulos colores azul (MS), verde (CBWH), rojo (COT), cyan (AFS), magenta (AEST) y amarillo (MEEM).

2º: Se selecciona aquel tracker con mayor solapamiento entre todas las combinaciones posibles. En este caso corresponde a COT (rectángulo de color rojo de líneas discontinuas).

3º: Se genera el nuevo bbox candidato (c) para ese frame mediante el seleccionado anteriormente y un margen aleatorio. En este caso corresponde al rectángulo de color blanco.

4º: En este paso se realiza la selección por gradiente del valor de atracción y se guardan tanto $a_i(c)$ (que será denominado $a_{res}(c)$) como el bbox obtenido (rectángulo de color blanco).

5º: Para el siguiente frame (650), se vuelve a calcular el mejor tracker (mayor solape) entre el rectángulo blanco del frame anterior y el resto. Y se obtiene de nuevo el mejor algoritmo con el que hallar después el nuevo bbox candidato. En este caso vuelve a ser COT, representado por el rectángulo de color rojo de líneas discontinuas.

Se realiza el mismo proceso hecho desde el paso 1º para cada frame de la secuencia.

Figura 4.8: Explicación mediante ejemplos visuales del desarrollo del algoritmo propuesto.

Los resultados para cada secuencia de la implementación realizada se expondrán en el Capítulo 5 de la presente memoria.

Capítulo 5

Resultados experimentales

En este capítulo se muestran los resultados experimentales obtenidos con el algoritmo propuesto en el Capítulo 4. Las pruebas se han realizado tanto para las secuencias clasificadas de corto plazo, como para las de largo plazo, con el fin de comparar su rendimiento en unas y otras. Los experimentos ejecutados se han realizado en un procesador Intel® Core™ i5 de 4ª generación con 4GB de RAM.
El capítulo está dividido en seis apartados: en el primero de ellos, 5.1. se presentan las distintos datasets disponibles para realizar los experimentos; acto seguido en el apartado 5.2. se indican los trackers sobre los que se ha ejecutado el algoritmo y la sección 5.3. especifica la métrica llevada a cabo en todos ellos. Después se muestran los resultados obtenidos para las secuencias a corto plazo, apartado 5.4. A continuación se repite el proceso para el conjunto de vídeos a largo plazo, apartado 5.5. Finalmente, en la sección 5.6. se presentan los experimentos conseguidos mediante el algoritmo propuesto.

5.1. Datasets disponibles

Para llevar a cabo la evaluación del algoritmo propuesto se ha empleado el dataset público LTDT 2014, *Long Term Detection and Tracking*, (http://www.micc.unifi.it/LTDT2014) [38] enfocado especialmente para el largo plazo, ya que las seis secuencias que contiene superan los 1000 frames de duración. El objetivo es el de estimular la investigación en el seguimiento de uno o múltiples objetos durante largos periodos de tiempo [38]. Según sus consideraciones, una secuencia de largo plazo es aquella cuya duración es de al menos 2 minutos (25 – 30 fps), siendo más óptimas aquellas que se aproximen a los 10 minutos. Entre las características del dataset se encuentra que las oclusiones por encima del 50% y y las salidas del plano de rotación mayores de 90% se considera al objeto de interés no visible y se denota en su correspondiente groud-truth como “NaN”.

Se ha escogido este conjunto de secuencias porque a diferencia de las secuencias cortas, que existen varios sitios con muchas opciones donde elegir, a día de hoy no hay muchos

lugares que hayan agrupado varios conjuntos de secuencias con vistas a largo plazo. Sin embargo en este congreso si se ha hecho así con las seis secuencias escogidas. Entre los problemas que pueden aparecer se encuentran:

- Oclusiones del objeto de interés, tanto totales como parciales. Además se incluyen en esta categoría las salidas del plano visual del objeto.

- Cambios de escala producidos porque que el objeto varía su distancia a cámara haciendo que aumente o disminuya.

- Cambios de apariencia, ya sea por posibles sombras o movimientos propios del objeto.

- Similitud de objetos, sobre todo en los vídeos de seguimiento de vehículos en carretera.

- Cambios de iluminación, ya que el objeto entra en zonas de diferente luminosidad de forma gradual o variable.

El conjunto de las seis secuencias seleccionadas del LTDT seleccionadas para su evaluación con los distintos trackers es el formado por Motocross, NissanSkyline, Sitcom, Volkswagen, Carchase y LiverRun, siendo esta última la secuencia más larga con la que se ha trabajado en este ámbito hasta la fecha (ver Tabla 5.1 para obtener la resolución y el número de frames de cada una). En la Figura 5.1 se pueden ver varios frames de cada secuencia. El bounding box verde indica el objeto de interés.

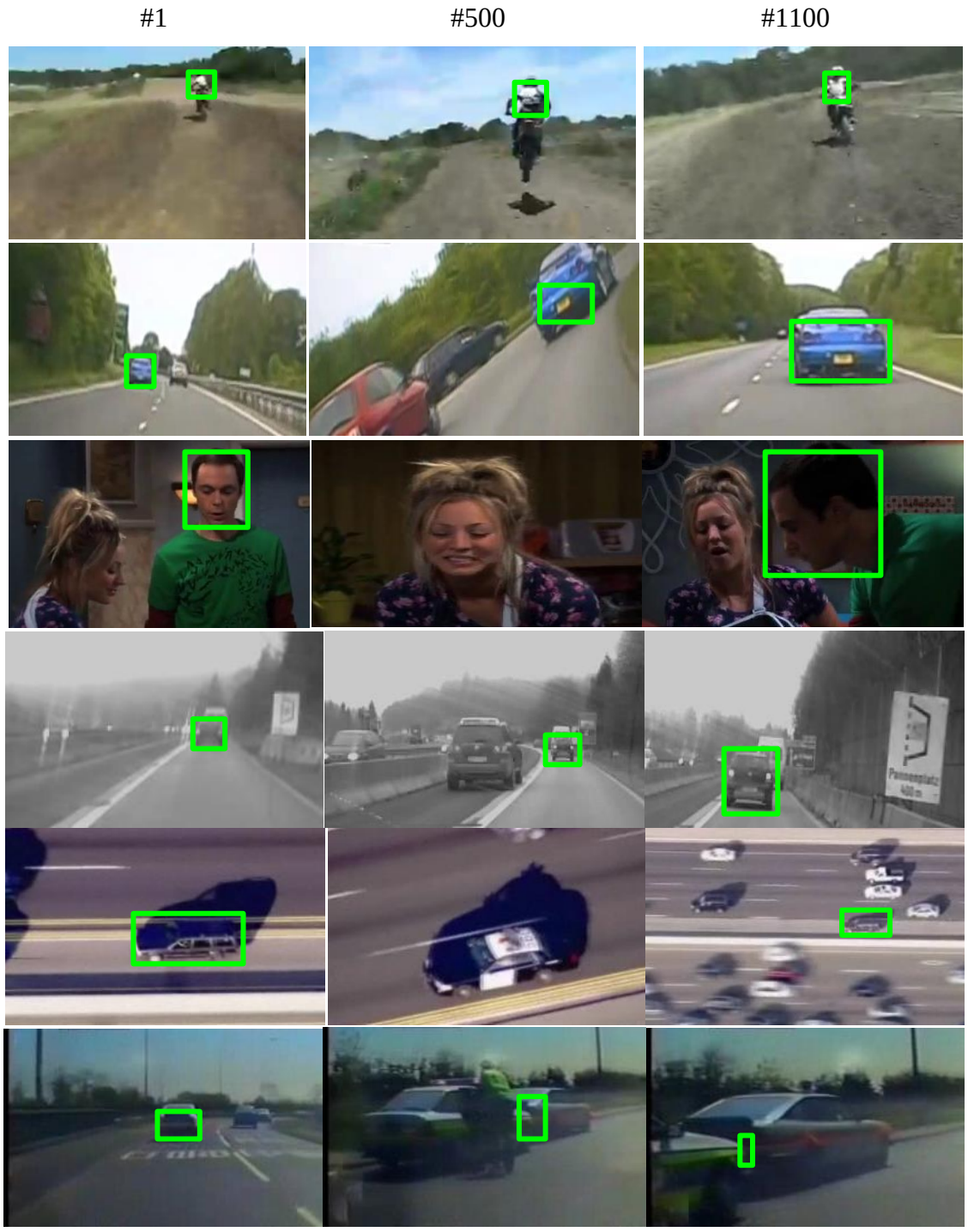


Figura 5.1: Varios frames (1, 500 y 1100 en todos los casos) de las secuencias que conforman el dataset de largo plazo. Por filas de arriba a abajo: Motocross, NissanSkyline, Sitcom, Volkswagen, Carchase y LiverRun.

5.1.1. Problemas más característicos

A continuación se detallan los problemas más comunes del seguimiento de objetos encontrados tras estudiar las secuencias escogidas anteriormente (ver Tabla 5.1). El criterio decidido va del rango de * a *****, donde la dificultad va en orden ascendente de 1 a 5*.

Secuencia	Duración (frames)	Oclusiones		Cambios de escala	Apariencia	Similitud objetos	Iluminación	Motion blur
		Parciales	Totales					
Motocross	1811	**	**	***	*	**	*	**
NissanSkyline	3742	*	*	**	*	****	***	***
Sitcom	3898	***	****	*	****	*	***	*
Volkswagen	4474	**	*	**	**	*****	****	**
Carchase	9886	**	****	***	**	*****	***	****
LiverRun	29597	**	**	*	*	****	**	**

Tabla 5.1. Problemas más característicos de las secuencias de largo plazo seleccionadas. En verde se muestra el caso más fácil, en rojo el más costoso de la característica de entre todas las secuencias.

A la vista de la anterior tabla, se puede concluir que la secuencia que más desafíos presenta es Carchase, y la que menos problemas supone es NissanSkyline. Resulta interesante mencionar también al vídeo LiverRun, que a pesar de ser el que mayor duración presenta, no sufre tantos inconvenientes como otros.

Además se ha empleado otro conjunto de secuencias para la realización de la parte de corto plazo. Las seis secuencias escogidas se encuentran públicamente en [26]. Son secuencias cortas, de menos de 500 frames de duración, pero donde también aparecen los problemas mencionados anteriormente. Las seleccionadas son las siguientes:

Nombre	*Resolución*	*Frames*
Deer	704x400	71
Skiing	640x360	81
Matrix	800x336	100
Coke	640x480	291
Bolt	640x360	350
Soccer	640x360	392

Tabla 5.2: Ejemplos de secuencias consideradas para tracking a corto plazo. Dataset extraído de [26].

A continuación se detallan los problemas más comunes del seguimiento de objetos encontrados tras estudiar las secuencias de corto plazo escogidas anteriormente (ver Tabla 5.2). El criterio decidido va del rango de * a *****, donde la dificultad va en orden ascendente de 1 a 5*.

Secuencia	Duración (frames)	Oclusiones		Cambios de escala	Apariencia	Similitud objetos	Iluminación	Motion blur
		Parciales	Totales					
Deer	71	-	-	*	*	***	*	**
Skiing	81	-	-	**	**	*	*	*
Matrix	100	*	*	**	***	****	****	**
Coke	291	***	**	*	**	*	*	*
Bolt	350	-	-	*	*	***	*	*
Soccer	392	***	*	*	**	****	*	*

Tabla 5.3. Problemas más característicos de las secuencias de corto plazo seleccionadas. En verde se muestra el caso más fácil, en rojo el más costoso de la característica de entre todas las secuencias

Tras la realización de la Tabla 5.3. se puede deducir que la secuencia con más inconvenientes es sin lugar a dudas Matrix. Esto se debe fundamentalmente a que se producen variaciones en la luminosidad de vídeo, además de que abundan objetos muy similares en el frente y en el fondo durante toda la secuencia.
Por otra parte, las secuencias que presentan un menor número de problemas son Bolt, Deer y Skiing, que no sufren de ninguna oclusión del objeto de interés durante la duración de cada una. Salvo por ciertas similitudes en el caso de Deer y Bolt, para el resto de campos analizados no se ven afectados.

Figura 5.2: Varios frames de las secuencias que conforman el dataset de corto plazo. Por filas de arriba abajo y de izquierda derecha: Deer (frame 35), Skiing (frame13), Matrix (frame 69), Coke (frame 165), Bolt (frame170) y Soccer (frame274). En todas ellas el rectángulo verde corresponde al objeto de interés.

5.2. Trackers empleados

El algoritmo propuesto surge como una combinación de seis trackers:

- *Mean – Shift* (MS) [17, 19].
- *Corrected Background Weighted Histogram* (CBWH) [40].
- *Color Tracker* (COT) [43, 44, 50].
- *Active Feature Selection* (AFS) [30, 46, 51, 52].
- *Active Example Selection Tracker* (AEST) [47, 48].
- *Multiple Experts using Entropy Minimization* (MEEM) [49, 53, 54].

Todos ellos se encuentran implementados en Matlab. Además en los trackers AFS, AEST y MEEM se ha empleado la librería pública de tratamiento de imágenes OpenCV[5] para compilar ciertos archivos necesarios para el correcto funcionamiento del algoritmo.

El código de los dos primeros trackers ha sido facilitado por el VPU-Lab. Para el resto de los algoritmos, el código se ha obtenido de la web de los autores correspondientes donde se encuentran disponibles.

5.3. Método de evaluación

Para llevar a cabo la evaluación del rendimiento del algoritmo propuesto, se ha decidido utilizar el mismo sistema aplicado para la evaluación individual de los trackers seleccionados. Como ya se ha comentado en el Capítulo 4, las estrategias de evaluación empleadas han sido el Center Location Error (CLE) así como el Sequence Frame Detection Accuracy (SFDA). La primera técnica calcula la distancia entre el centro de la posición predicha del objeto y el centro de la posición del ground-truth. Cuanto mayor sea la distancia, mayor será la desviación del tracker. Obviamente, el objetivo es conseguir que la puntuación CLE de cada ejecución sea la menor posible. El segundo mecanismo evalúa el solapamiento entre el ground-truth y la salida del sistema como una relación de la intersección entre dos objetos y la unión entre ellos. Los mejores resultados serán aquellos que consigan un coeficiente de SFDA mayor, siendo el máximo valor de solapamiento posible 1.

[5] http://opencv.org/downloads.html

La fórmula del Center Location Error es la siguiente:

$$Center\ Location\ Error = \sqrt{(centerposres_{\mathrm{i}} - centerposGT_{\mathrm{i}})^2 - (centerposres_j - centerposGT_j)^2}\ , \quad (23)$$

$$CLE\ (en\ píxels) = media\ (Center\ Location\ Error)\ , \quad (24)$$

Ecuación 23: Fórmula para calcular el Center Location Error entre el centro de la posición resultado del objeto y el GT.

Conviene hacer referencia a una adaptación realizada a la hora del cálculo y la representación del valor CLE en las secuencias de largo plazo dado que en todos los vídeos del dataset de largo plazo hay un número considerable de frames donde no aparece el objeto de interés, y cuyo ground-truth viene definido como NaN (Not a Number). Se comprobó inicialmente que el valor del CLE en estos frames era 0, es decir, el mejor posible, lo que provocaba falsos positivos y resultados finales muy alejados del valor real. Para solucionar este inconveniente, se ha eliminado el cálculo en los frames que su ground-truth tuviera alguna coordenada NaN, y para representarlo se utilizado la interpolación entre el último frame sin NaN y el siguiente.

Para el caso del SFDA, se define como:

$$SFDA = \frac{\sum_{t=1}^{t=N_{frames}} \frac{\sum_{i=1}^{N_{mapped}^{(t)}} \frac{\left|G_i^{(t)} \cap D_i^{(t)}\right|}{\left|G_i^{(t)} \cup D_i^{(t)}\right|}}{\left\lceil \frac{N_G^{(t)} + N_D^{(t)}}{2} \right\rceil}}{\sum_{t=1}^{t=N_{frames}} \exists\ (N_G^{(t)}\ OR\ N_D^{(t)})}\ , \quad (25)$$

Ecuación 25: Fórmula para calcular el valor SFDA del solapamiento producido entre el ground-truth y la salida del sistema.

Siendo: - $N_G^{(t)}$: Objetos del ground-truth.
- $N_D^{(t)}$: Objetos detectados.
- $N_{mapped}^{(t)}$: Número de objetos mapeados en el frame t.
- $G_i^{(t)}$: Indica el objeto del GT i^{th} en el frame t^{th}.
- $D_i^{(t)}$: Indica el objeto detectado i^{th} en el frame t^{th}.

En los siguientes apartados se presentan un conjunto de figuras que representan los rendimientos obtenidos de cada secuencia, tanto de corto como de largo plazo, para los dos mecanismos de evaluación. La estructura seguida consiste en por cada vídeo mostrar sus correspondientes gráficas del cálculo del CLE y del SFDA respectivamente, para después indicar las características más representativas de cada una de ellas.

5.4. Resultados del corto plazo

En primer lugar se ha procedido a la ejecución de los algoritmos seleccionados en el conjunto de secuencias de corto plazo, de tal manera que se tiene un punto inicial desde donde empezar a comparar posteriormente con los vídeos a largo plazo. Teóricamente los resultados en este apartado deberían ser mejores que en el de largo plazo, porque al ser vídeos más cortos, los problemas de seguimiento también tienen una duración menor.

El formato empleado es representar en una misma gráfica correspondiente a cada vídeo, el conjunto de los seis algoritmos seleccionados y descritos en el Capítulo 3, para poder realizar de este modo una mejor comparación entre cada tracker.
El eje de abscisas corresponde con la duración de la secuencia en frames, mientras que la puntuación obtenida queda reflejada en el eje de ordenadas, en el caso del CLE y en el caso del coeficiente SFDA.

- *Secuencia: Deer (71 frames)*

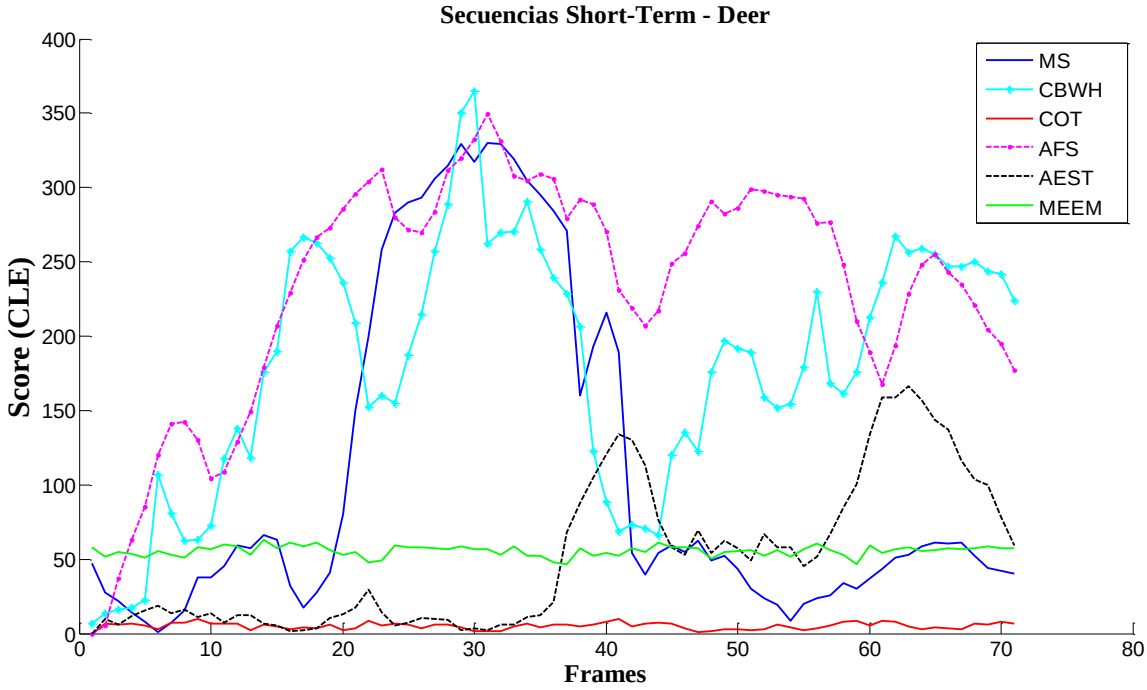


Figura 5.3: Gráfica resultados CLE de los seis trackers para la secuencia Deer.

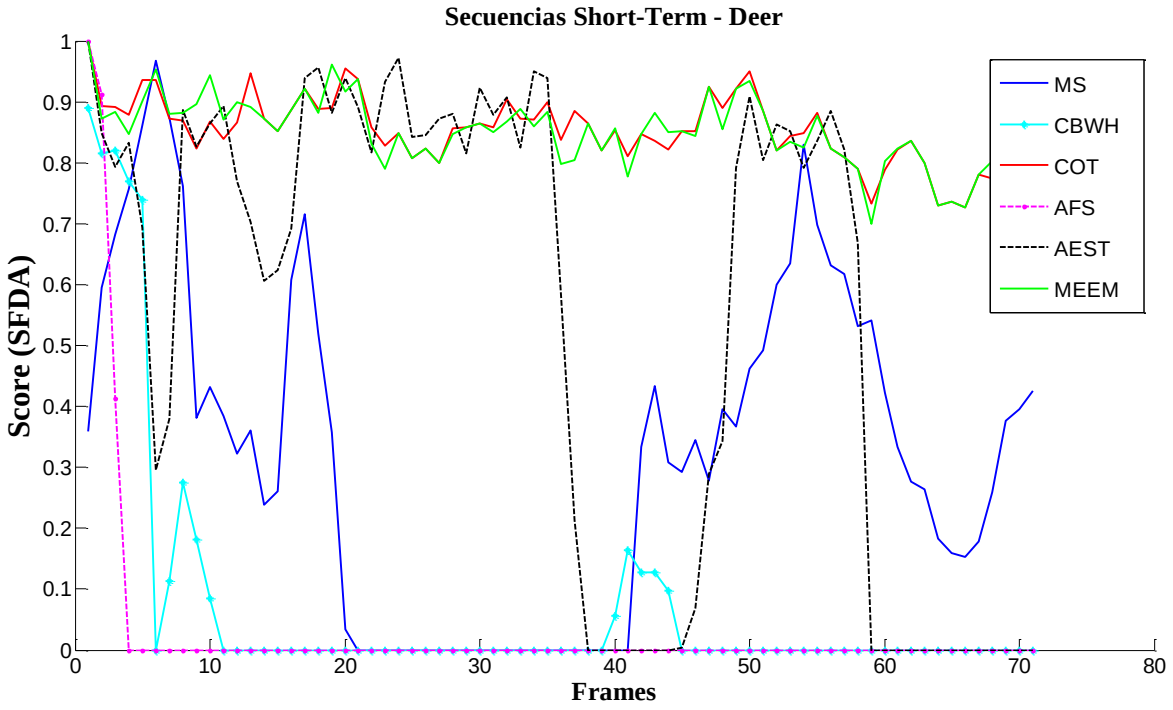


Figura 5.4: Gráfica resultados SFDA de los seis trackers para la secuencia Deer.

Los resultados obtenidos para la secuencia Deer mostrados en la Figuras 5.3. y 5.4. indican de forma clara que el mejor comportamiento corresponde al algoritmo COT. Para el caso del CLE presenta un buen rendimiento, constante y siempre por debajo de los 10 píxels, obteniendo una media final de 5,1 píxels. Otro tracker que inicializa de forma correcta y muestra un resultado aceptable, sobre todo durante la primera mitad de la secuencia es AEST, que finaliza con un CLE medio de 51,2 píxels. Hay que destacar también respecto de la evaluación CLE, el aspecto que presentan tres algoritmos, MS, CBWH y AFS en torno al frame 30. En todos ellos, el resultado es un pico que corresponde con valor muy alto y por tanto un resultado desfavorable.

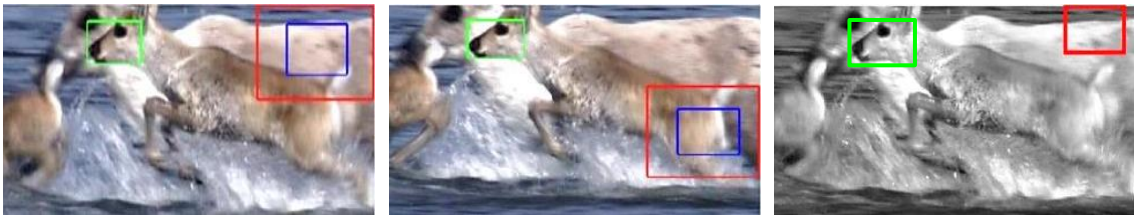


Figura 5.5: Ejemplos de varios frames que presentan un alto valor del CLE coincidiendo con el pico de la representación gráfica. De izquierda a derecha, algoritmos MS, CBWH y AFS en los frames 31, 30 y 31 respectivamente. En el caso del tracker AFS, el rectángulo rojo corresponde a los resultados estimados del objeto, y no al área de búsqueda como ocurre con MS y CBWH. El ground-truth queda representado por el rectángulo verde en todos los casos.

Respecto a la gráfica correspondiente a la evaluación SFDA, se distinguen de manera notoria tres grupos. Hay dos trackers, COT y MEEM cuyos resultados son favorables a lo largo de la secuencia, siempre con un coeficiente por encima de 0,7. Después, se encuentran AEST y MS, con un comportamiento bastante más irregular, alternando frames con buenos resultados con otros donde no hay solapamiento. Y por último, el grupo formado por los algoritmos AFS y CBWH, que claramente obtienen los peores resultados desde los primeros compases del vídeo.

Figura 5.6: Ejemplos de varios frames que presentan un alto valor del SFDA. De izquierda a derecha, algoritmos MEEM, y COT en los frames 20 para ambos casos. En verde el rectángulo del ground-truth y en azul la estimación del algoritmo.

- *Secuencia: Skiing (81 frames)*

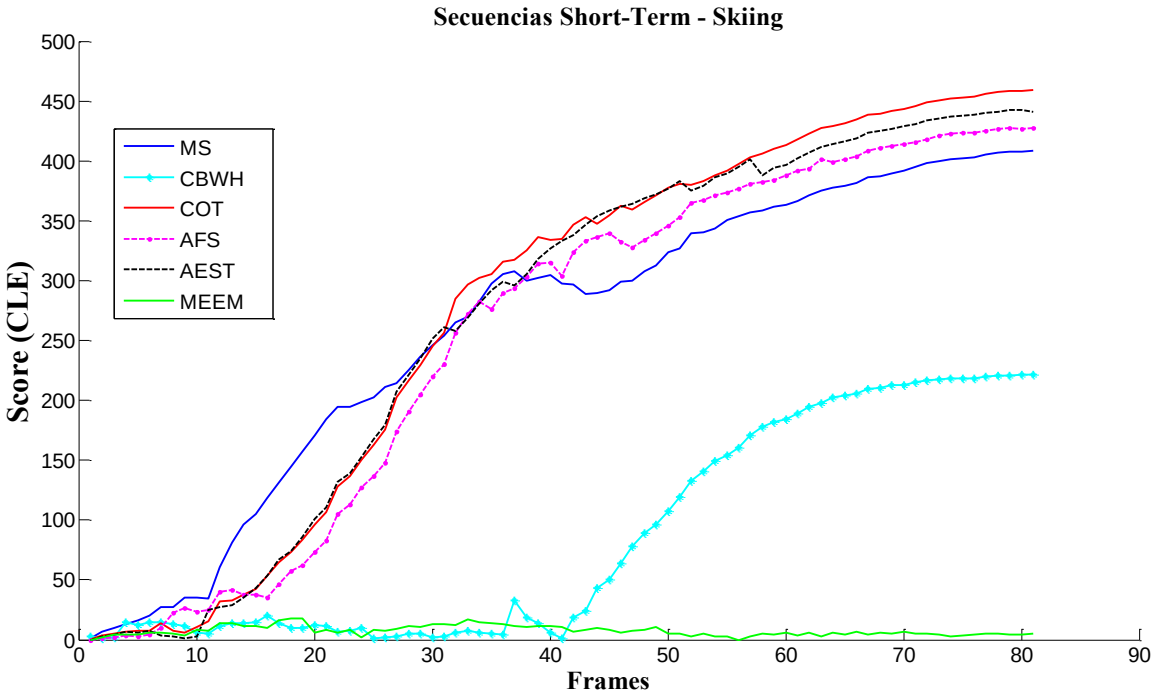


Figura 5.7: Gráfica resultados CLE de los seis trackers para la secuencia Skiing.

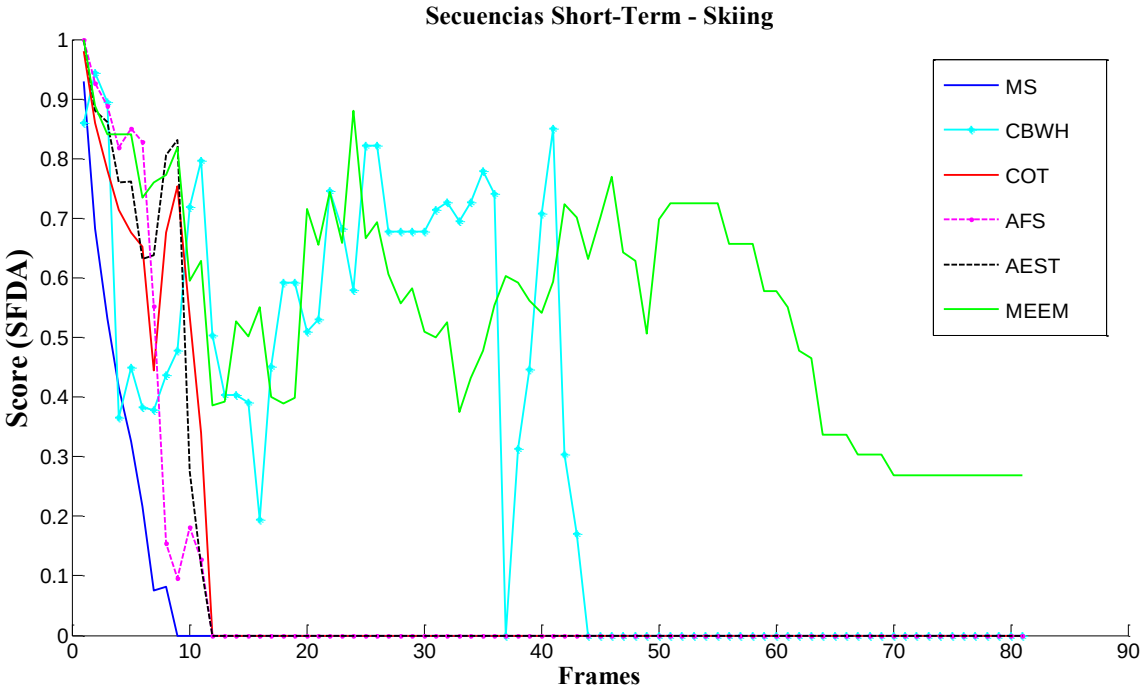


Figura 5.8: Gráfica resultados SFDA de los seis trackers para la secuencia Skiing.

La Figura 5.7. muestra varios aspectos destacables. En primer lugar, se puede observar como la mayoría de los algoritmos (MS, COT, AFS y AEST) presentan unos resultados bastante similares. Inicializan de forma adecuada pero a partir del frame 12 pierden al objeto y en el resto de la secuencia no logran recuperarlo. Sus valores medios finales del CLE son muy parecidos entre ellos (entre los 250 y 280 píxels). Los otros dos trackers restantes producen mejores resultados, siendo MEEM el más destacado, con valores por que no superan los 20 píxels y un CLE final de 7 píxels. Durante la primera mitad de la secuencia compiten MEEM y CBWH, sin embargo a partir del frame 43, éste último comienza a alejarse del objeto, si bien en menor medida que los cuatro primeros trackers.

Figura 5.9: Ejemplos donde los distintos trackers comienzan a perder al objeto. De izquierda a derecha se muestran los frames 12, 12 y 43 para los trackers MS, AFS y CBWH respectivamente. El rectángulo verde representa el ground-truth y los rectángulos azules para MS y CBWH, y el rojo para AFS, la posición estimada del objeto.

Conviene destacar en este vídeo la mejoría apreciada respecto del algoritmo CBWH respecto a su predecesor MS, reduciendo en más de la mitad el valor final del CLE.

En cuanto a la Figura 5.8. el análisis es bastante similar al realizado con la gráfica del CLE. Se produce la rápida disminución del coeficiente SFDA para los cuatro algoritmos indicados antes, quedándose CBWH y MEEM como los únicos que siguen obteniendo valores distintos de cero a lo largo del vídeo. Finalmente es MEEM quien consigue ser el algoritmo que presenta solapamiento entre el bounding box del ground-truth y el bounding box resultado, y aunque su valor se va reduciendo, obtiene una media final más que aceptable de 0,55, sobre todo teniendo en cuenta los pobres resultados del resto de los algoritmos.

- *Secuencia: Matrix (100 frames)*

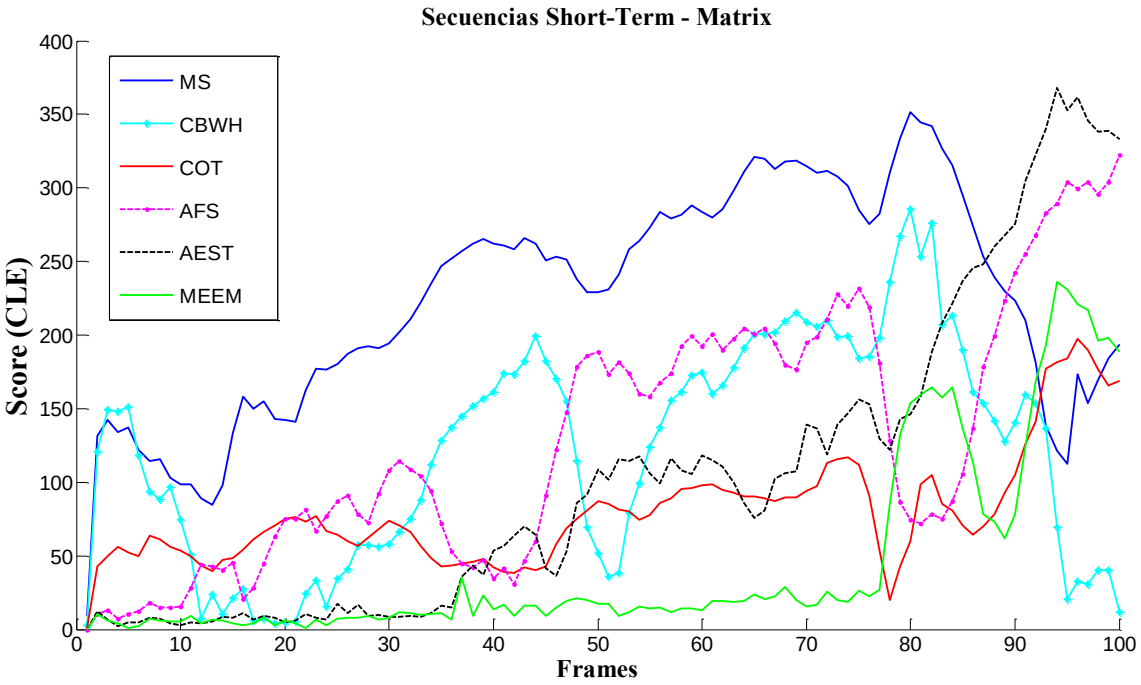


Figura 5.10: Gráfica resultados CLE de los seis trackers para la secuencia Matrix.

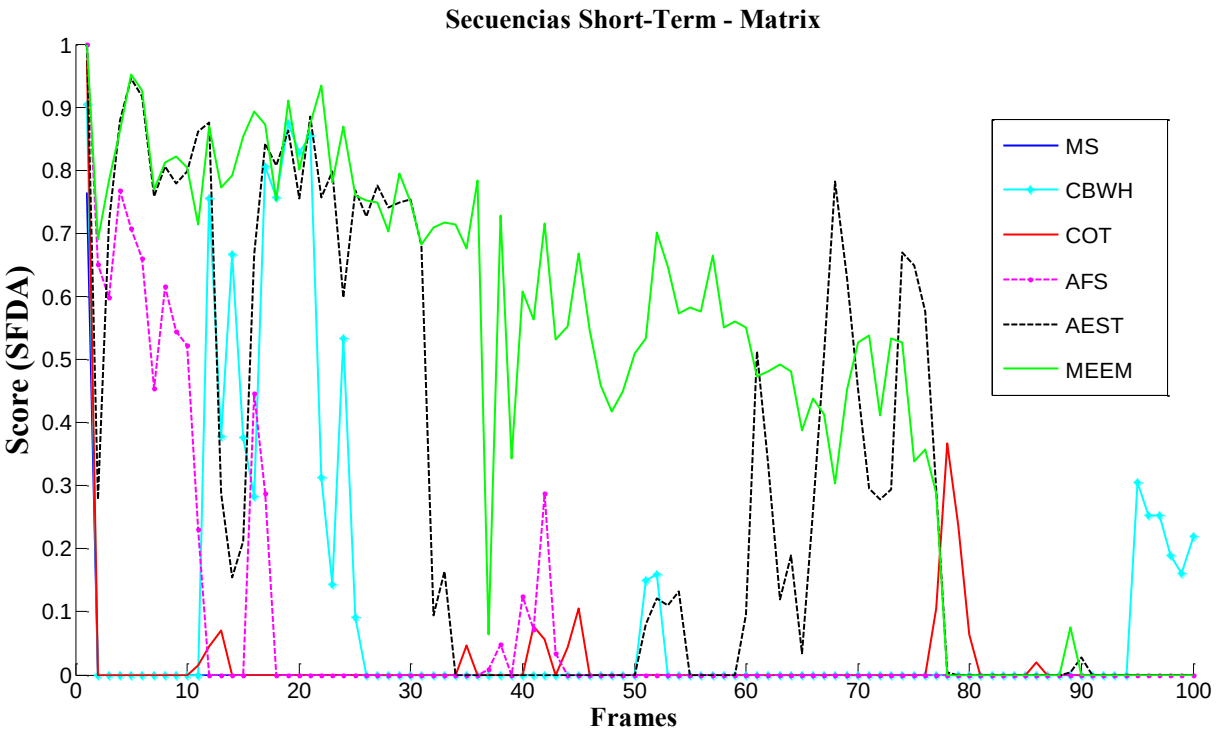


Figura 5.11: Gráfica resultados SFDA de los seis trackers para la secuencia Matrix.

La secuencia Matrix presenta varias dificultades que provocan que los resultados de los algoritmos analizados no sean demasiados buenos. Las características del objeto están presentes en el fondo durante todo el vídeo, hay similitudes de apariencia y rápidos cambios de pose.
Esto lleva a que en la Figura 5.10. se aprecie un comportamiento irregular como tónica general en todos los trackers. Además los algoritmos MS y CBWH fallan tras el inicio, y aunque CBWH vuelve a seguir el objeto durante unos frames, no consiguen buenos resultados finales. De nuevo MEEM y en este vídeo COT, obtienen los mejores resultados, siendo los únicos que logran estar por debajo de los 100 píxels de media final.

En relación a la gráfica del coeficiente de SFDA, los tres algoritmos más destacados son MEEM, AEST y CBWH. El primero de los tres obtiene los mejores resultados, una media de 0,49, aunque también contiene dos fuertes descensos (frames 37 y 78). Después AEST, cuyo SFDA medio es 0,3, y que alterna, incluso consecutivamente, fuertes descensos con rápidas subidas.

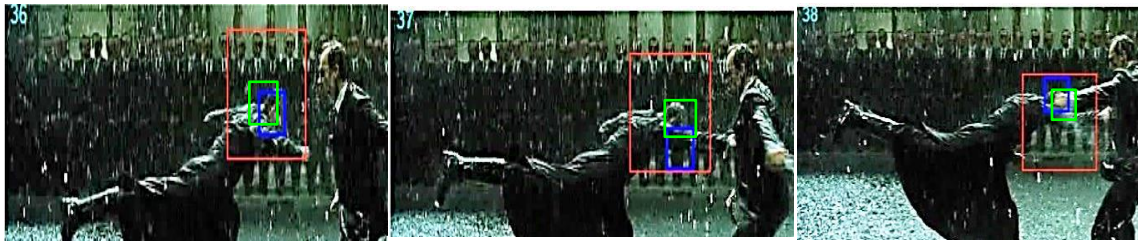


Figura 5.12: Ejemplo del solapamiento (SFDA) del algoritmo MEEM para la secuencia Matrix durante los frames 36, 37 y 38. En rojo el área de búsqueda, en azul el resultado de la estimación y de verde el ground-truth. Esta serie de frames consecutivos corresponden a uno de las amplias fluctuaciones que se pueden apreciar en la gráfica del SFDA. Debido a un cambio brusco de pose del objeto, el algoritmo también varía bruscamente su área de solapamiento.

- *Secuencia: Coke (291 frames)*

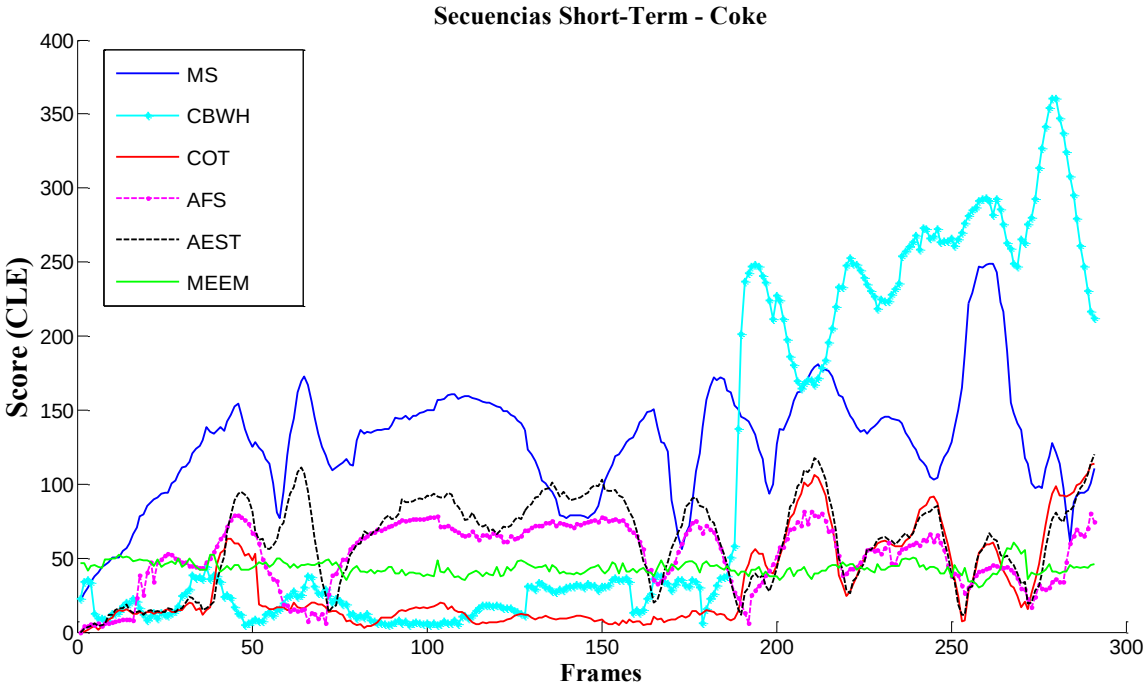


Figura 5.13: Gráfica resultados CLE de los seis trackers para la secuencia Coke.

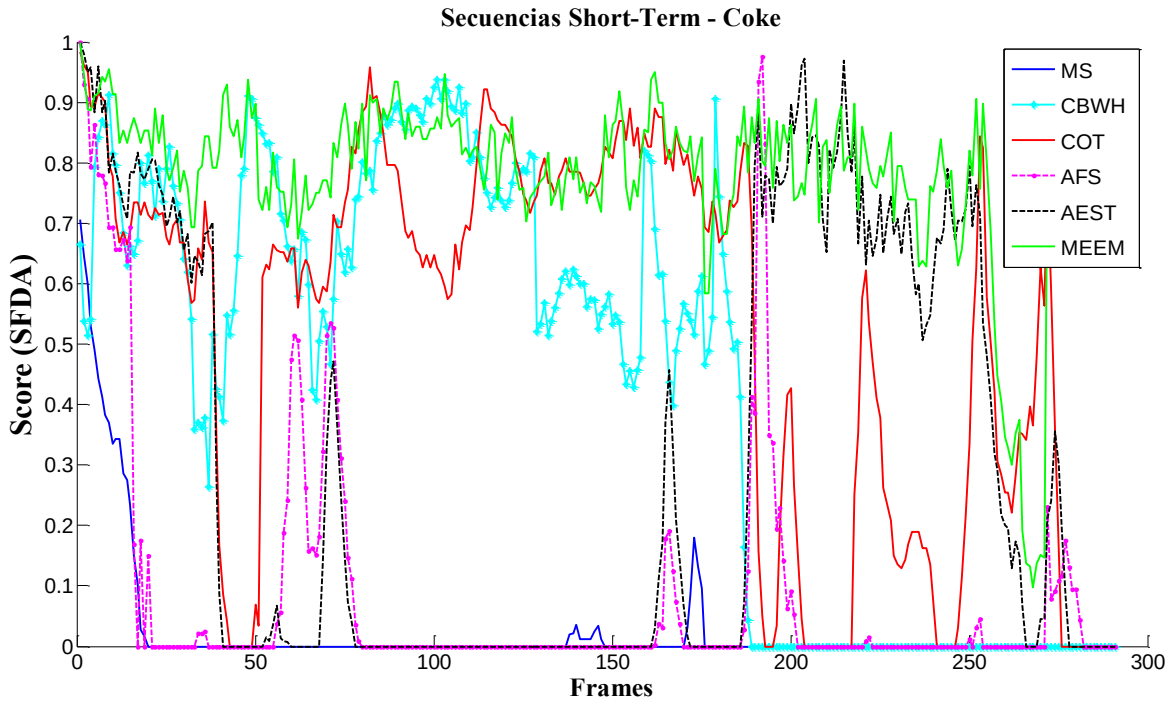


Figura 5.14: Gráfica resultados SFDA de los seis trackers para la secuencia Coke.

Tras un análisis global de la secuencia Coke, se aprecia rápidamente que es la secuencia en la que los distintos algoritmos han obtenido mejores resultados hasta el momento. Además hay una menor diferencia entre ellos, sin que ninguno refleje extraordinarios valores pero tampoco resultados muy negativos. Esto puede ser consecuencia de que este vídeo presenta un menor número de desafíos con respecto a otros del dataset evaluado. En Coke aparece una oclusión total y alguna parcial, de corta duración, como problemas más representativos.

Respecto a la técnica de evaluación CLE, los algoritmos AFS y AEST muestran resultados muy similares, algo que comparten con el tracker COT a partir del frame 189, momento en el cual se produce la oclusión total del objeto. Esto provoca que COT aumente su valor CLE. Lo mismo le sucede al algoritmo CBWH pero con resultados más negativos, ya que tras la oclusión no logra volver a seguir al objeto correctamente.

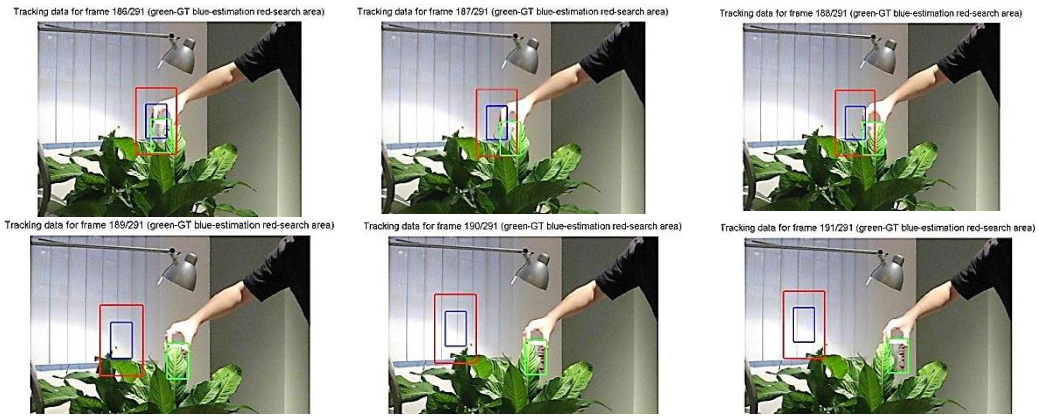


Figura 5.15: Varios frames consecutivos (de 186 al 191) en los que tiene lugar la oclusión total del objeto. En este caso se muestran los resultados del tracker CBWH.

En la Figura 5.14. se observa el comportamiento constante del algoritmo MEEM prácticamente hasta el final de la secuencia. Además también se encuentra el momento en el que tracker CBWH deja de seguir al objeto debido a la oclusión (frame 189), ya que el coeficiente SFDA en su caso desciende de 0,5 a 0 rápidamente. El resto de algoritmos están representados mediante diversas fluctuaciones, correspondiendo éstas a las distintas oclusiones parciales que sufre el objeto de interés a lo largo de la secuencia.

- *Secuencia: Bolt (350 frames)*

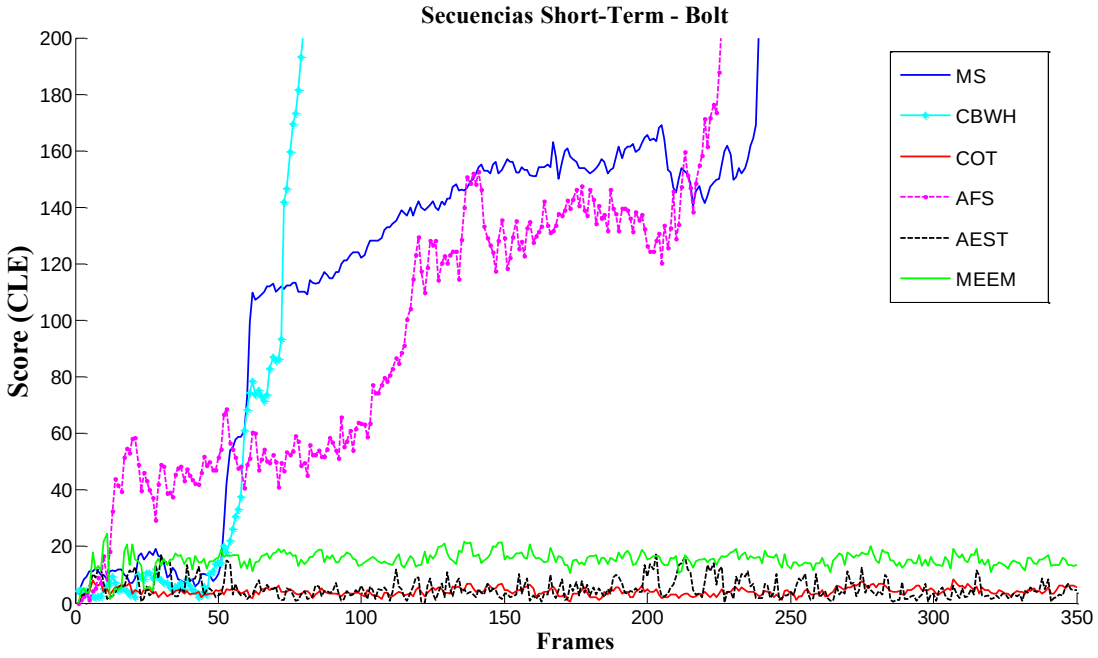


Figura 5.16: Gráfica resultados CLE de los seis trackers para la secuencia Bolt.

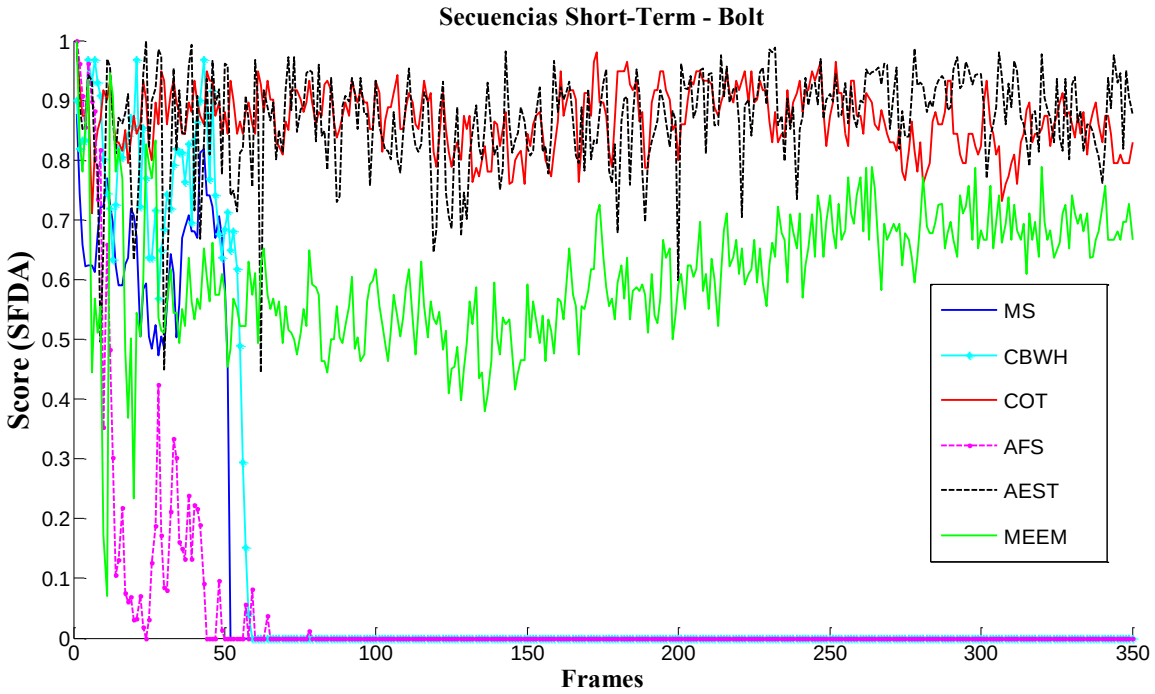


Figura 5.17: Gráfica resultados SFDA de los seis trackers para la secuencia Bolt.
En esta secuencia, para el mecanismo de evaluación del Center Location Error, se ha editado la representación del eje Y (Score) para que se puedan apreciar mejor los valores más bajos, ya que son los más representativos.

La valoración de esta secuencia resulta interesante ya que presenta una gran disparidad entre los diferentes trackers. Además los resultados tanto de la evaluación CLE como por medio del mecanismo de SFDA muestran las mismas consideraciones de manera clara. Por un lado se observan tres algoritmos que funcionan muy bien, siguiendo al objeto en todo momento y obteniendo resultados muy satisfactorios. Son COT, AEST y MEEM. Sin embargo, también hay que mencionar la mala actuación de MS, CBWH y AFS que sólo consiguen seguir al objeto durante los primeros frames de la secuencia. Esto es debido a que, para MS y AFS el tracker falla debido a la similitud entre los objetos, y en el caso de CBWH, tras varios frames en el inicio siguiendo correctamente, poco a poco va perdiendo el objeto de interés y no es capaz de aproximarse ni de recuperarlo.

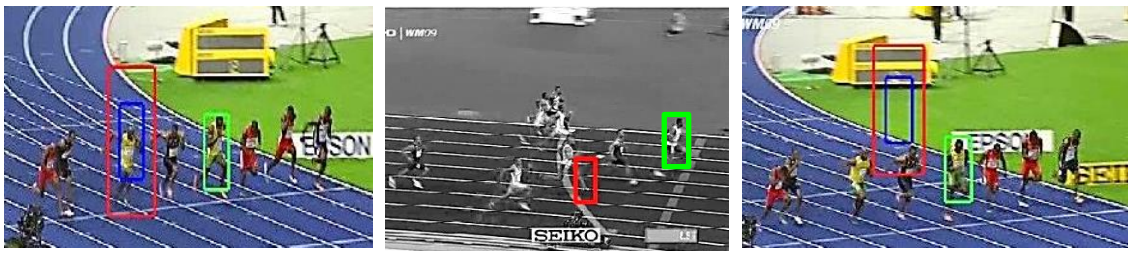


Figura 5.18: Ejemplos de frames donde fallan los trackers MS (frame 61), AFS (frame 210) y CBWH (frame 70). Los rectángulos azules para MS y CBWH indican el bounding box resultado. En el tracker AFS esto se muestra mediante el rectángulo rojo. En los tres algoritmos el ground-truth está representado mediante el rectángulo verde.

Como se ha mencionado anteriormente, en los casos de MS y AFS, el tracker falla debido a que se queda con otro objeto similar al objeto de interés. Para la última imagen, algoritmo CBWH, cuando deja de seguir al objeto no se detiene en otro similar, si no que cada vez se aleja más del objeto y sus resultados empeoran, como se ha comprobado en las Figuras 5.16 y 5.17.

- *Secuencia: Soccer (392 frames)*

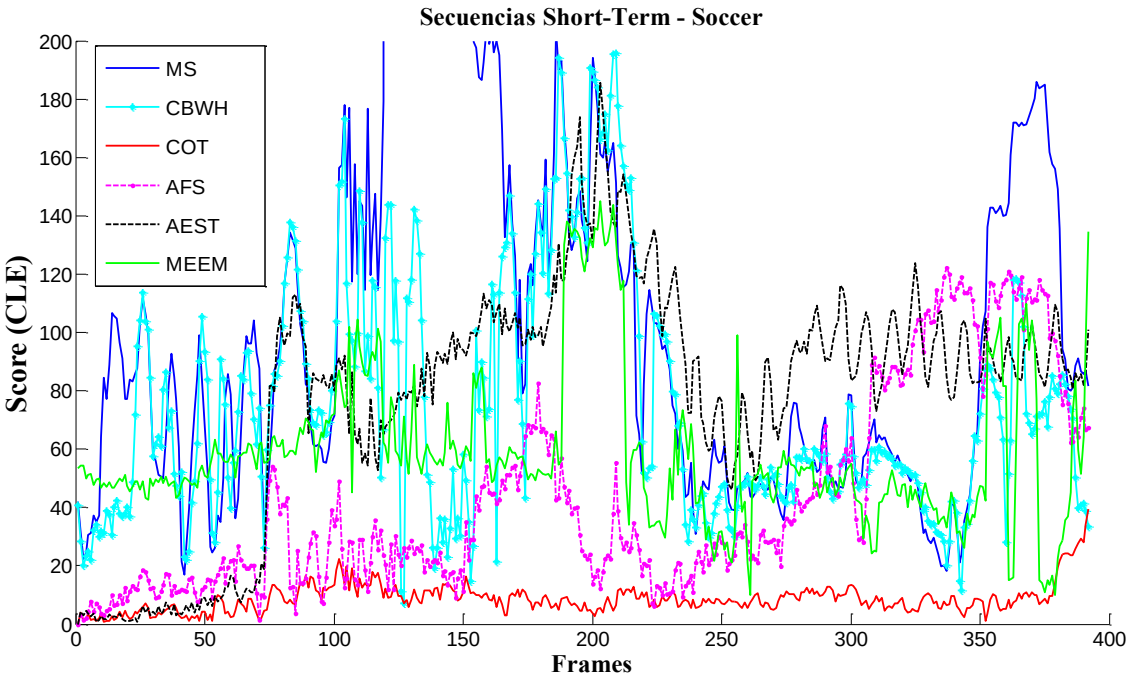


Figura 5.19: Gráfica resultados CLE de los seis trackers para la secuencia Soccer.

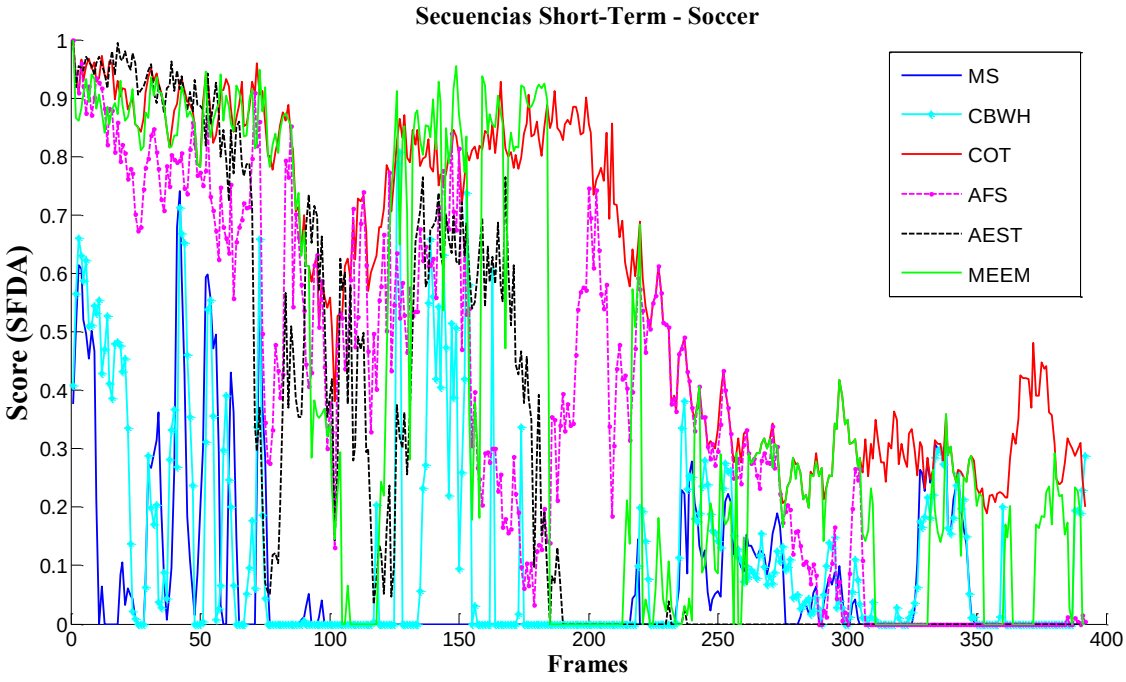


Figura 5.20: Gráfica resultados SFDA de los seis trackers para la secuencia Soccer.

La secuencia Soccer presenta como gran problema que las características del objeto están presentes en el fondo de la imagen a lo largo de todo el vídeo, así como varias oclusiones parciales y la similitud del objeto en la parte central de la secuencia.
Observando la gráfica de resultados mediante el mecanismo CLE, se deriva que el mejor algoritmo para esta secuencia corresponde a COT. Presenta un comportamiento bueno durante prácticamente todos los frames, siendo además muy constante. Para el caso del tracker AFS, empieza en la misma línea que COT, pero sufre cuando aparecen pequeñas oclusiones parciales durante varios frames (comienzan en torno al frame 150).
Otro de los puntos característicos que se observan en la Figura 5.19. son los picos aparecidos en torno a la mitad de la secuencia (frame 200) en los algoritmos MS, CBWH, AEST y MEEM. Esto se debe a que en esos frames aparece el problema de la similitud del objeto de interés con otro situado muy cerca y al mismo tiempo pequeñas oclusiones parciales, y los algoritmos confunden el objeto.

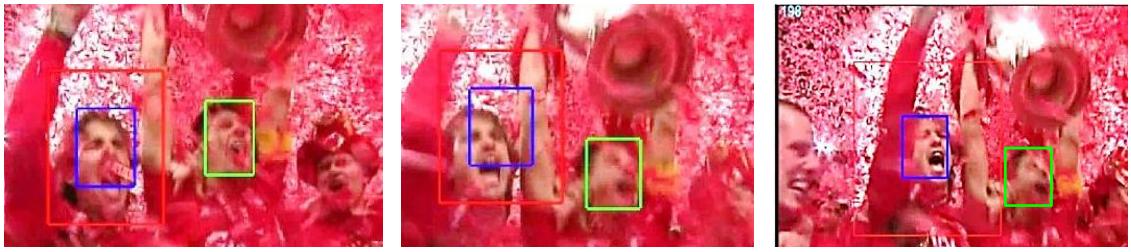


Figura 5.21: Ejemplos de varios trackers que confunden el objeto de interés debido a la similitud y a las pequeñas oclusiones parciales. De izquierda a derecha, algoritmo MS (frame 195), algoritmo CBWH (frame 197) y algoritmo MEEM (frame 198). El rectángulo azul indica la posición estimada del objeto por cada algoritmo, mientras que el rectángulo rojo el área de búsqueda y el verde corresponde al correspondiente ground-truth.

Respecto a los resultados obtenidos mediante la evaluación del coeficiente SFDA, destaca el buen inicio de todos los algoritmos salvo MS y CBWH. También se refleja el problema de la similitud en torno al frame 200, sobre todo para los casos de MEEM y AEST, descendiendo sus valores del SFDA de manera abrupta.

Finalmente en las siguientes tablas se muestran los valores de CLE (la media) y SFDA (media) para cada tracker en cada secuencia, así como la media total de cada algoritmo.

Tracker\Secuencia	*BOLT*	*COKE*	*SKIING*	*SOCCER*	*DEER*	*MATRIX*	*Media total*
MS	163	128	260	106	107	224	165
CBWH	330	102	**86**	75	180	121	149
COT	**4**	**31**	275	**8**	**5**	**79**	**67**
AFS	165	52	255	**43**	232	130	146
AEST	**5**	61	268	79	**51**	104	95
MEEM	15	**43**	**7**	59	55	**45**	**37**

Tabla 5.4: Resultados del Center Location Error medio de los tracker seleccionados para las secuencias de corto plazo. Los dos mejores valores de cada secuencia se representan en verde (1º) y rojo (2º).

Tracker\Secuencia	*BOLT*	*COKE*	*SKIING*	*SOCCER*	*DEER*	*MATRIX*	*Media total*
MS	0.093	0.025	0.040	0.072	0.319	0.007	0.092
CBWH	0.124	0.434	**0.306**	0.121	0.074	0.102	0.193
COT	**0.869**	**0.526**	0.091	**0.593**	**0.852**	0.022	**0.492**
AFS	0.041	0.088	0.079	0.378	0.032	0.080	0.116
AEST	**0.867**	0.299	0.093	0.291	0.541	**0.301**	0.398
MEEM	0.606	**0.776**	**0.551**	**0.412**	**0.850**	**0.495**	**0.615**

Tabla 5.5: Resultados del SFDA medio de los tracker seleccionados para las secuencias de corto plazo. Los dos mejores valores de cada secuencia se representan en verde (1º) y rojo (2º).

5.5. Resultados del largo plazo

A continuación se exponen una serie de gráficas en las que se muestran las puntuaciones CLE y SFDA obtenidas de cada secuencia para cada tracker. El formato empleado es representar en una misma gráfica correspondiente a cada vídeo, el conjunto de los seis algoritmos que se describieron en el Capítulo 3 de este PFC. El eje de abscisas corresponde con la duración de la secuencia en frames, mientras que la puntuación obtenida queda reflejada en el eje de ordenadas.
Además tras cada figura, se realiza un análisis con las consideraciones más características fruto del resultado de dicho algoritmo en esa secuencia determinada, actuando de la misma forma que se ha hecho en la sección 5.4.

Para las secuencias de largo plazo, en algunas ocasiones se ha editado la representación del eje de abscisas (Score) y del eje de ordenadas (Frames) para que se puedan apreciar mejor los valores más bajos, ya que son los más representativos (en el caso del método de evaluación CLE). Además, como la mayoría de las secuencias contienen una gran cantidad de valores “NaNs” que afectaban a los resultados finales de los algoritmos, se han suprimido estos valores a la hora de representarlos en las figuras. Se ha realizado la interpolación entre el último y el primer valor antes y después del frame que contiene algún dato “NaNs”. Es por este motivo que las gráficas tienen una duración menor a la indicada en la Tabla 5.1.

Se ha considerado esta opción para disponer de una gráfica que permita de este modo un análisis más óptimo.

- *Secuencia: Motocross (1368 frames)*

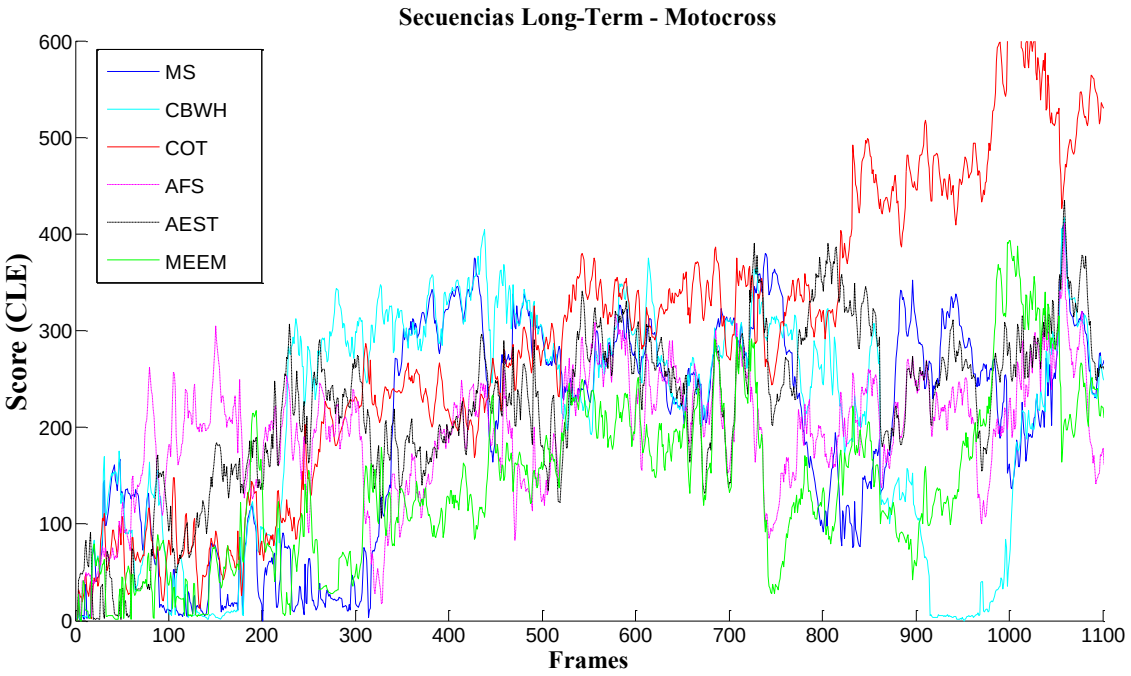


Figura 5.22: Gráfica resultados CLE de los seis trackers para la secuencia Motocross.

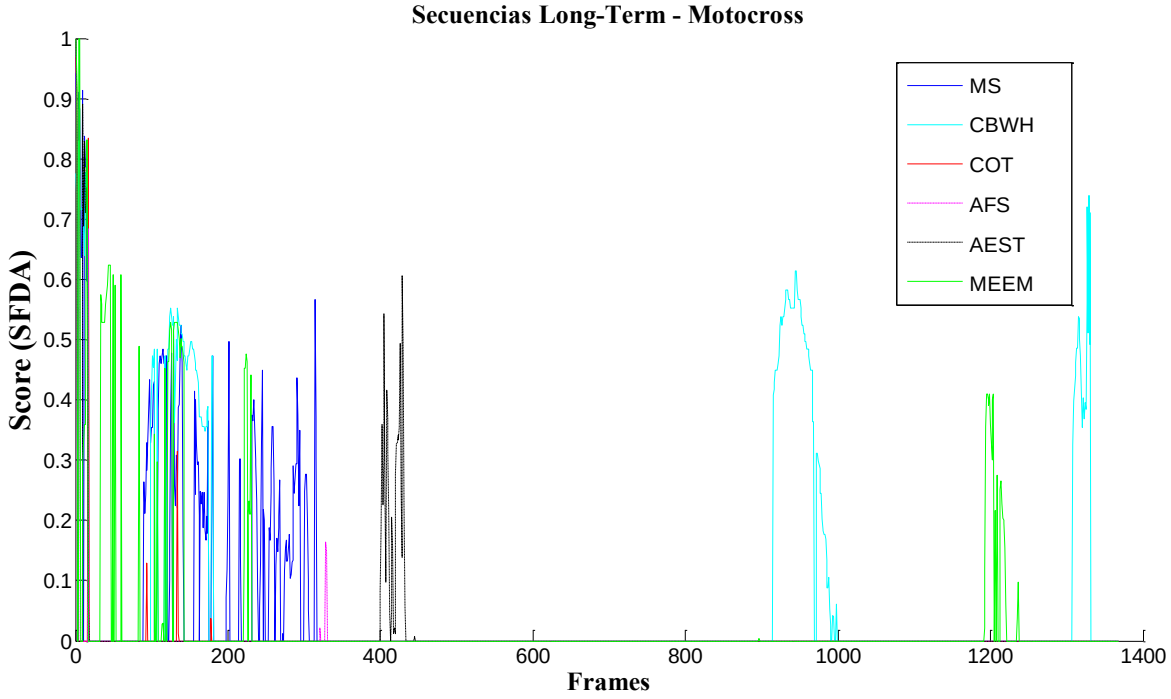


Figura 5.23: Gráfica resultados SFDA de los seis trackers para la secuencia Motocross.

Como cabría esperar en primera instancia, los resultados de los algoritmos en esta primera secuencia de largo plazo son bastante desfavorables. Se trata de una secuencia con constantes cambios de escala, oclusiones totales del objeto y salidas del plano de la cámara. Todo esto hace que los trackers se enfrenten a muchos desafíos durante un largo periodo de tiempo además.
Es por esto que en la Figura 5.22 se observa un comportamiento muy irregular como tónica general de los algoritmos, con constantes subidas y bajadas debido a los inconvenientes mencionados anteriormente.
Resulta más representativa en esta ocasión la técnica de evaluación del coeficiente SFDA, ya que presenta un marco más claro donde poder clasificar mejor los resultados de los algoritmos. Se pone de manifiesto en la Figura 5.23 que tras los 400 primeros frames, y tras producirse ya las salidas y posteriores retornos del objeto al plano de la cámara, ningún tracker consigue localizar al objeto de nuevo. Tan sólo se ven tres atisbos al final de la secuencia, que corresponden a los dos picos del algoritmo CBWH (frames 945 y 1329) y al tracker MEEM (frame 1204), pero con una duración de unas pocas decenas de frames.

Figura 5.24: Ejemplos de los resultados de distintos trackers durante varios frames para la secuencia Motocross. De izquierda a derecha y de arriba abajo: tracker CBWH (frame 945), tracker AEST (frame 376), tracker COT (frame 231) y tracker AFS (frame 397). Como se puede observar en las imágenes, exceptuando el tracker CBWH que consigue recuperar el seguimiento al objeto, el resto de algoritmos fallan y no vuelven a encontrar al objeto. En verde está representado el ground-truth y en rojo la estimación del objeto por parte de cada tracker.

- *Secuencia: Sitcom (3103 frames)*

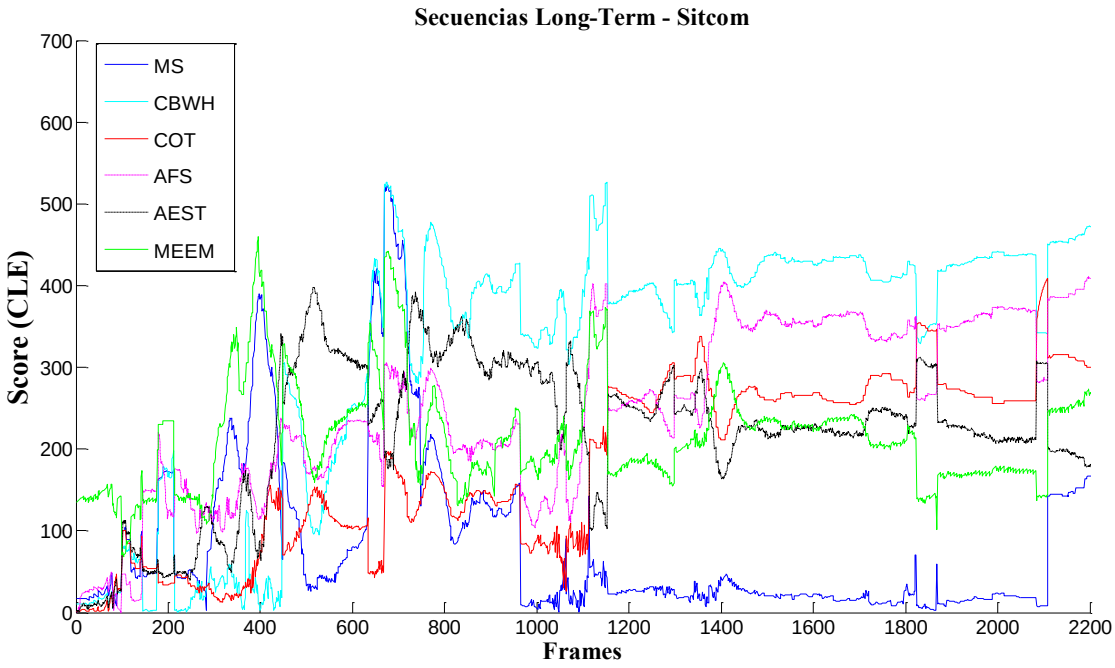


Figura 5.25: Gráfica resultados CLE de los seis trackers para la secuencia Sitcom.

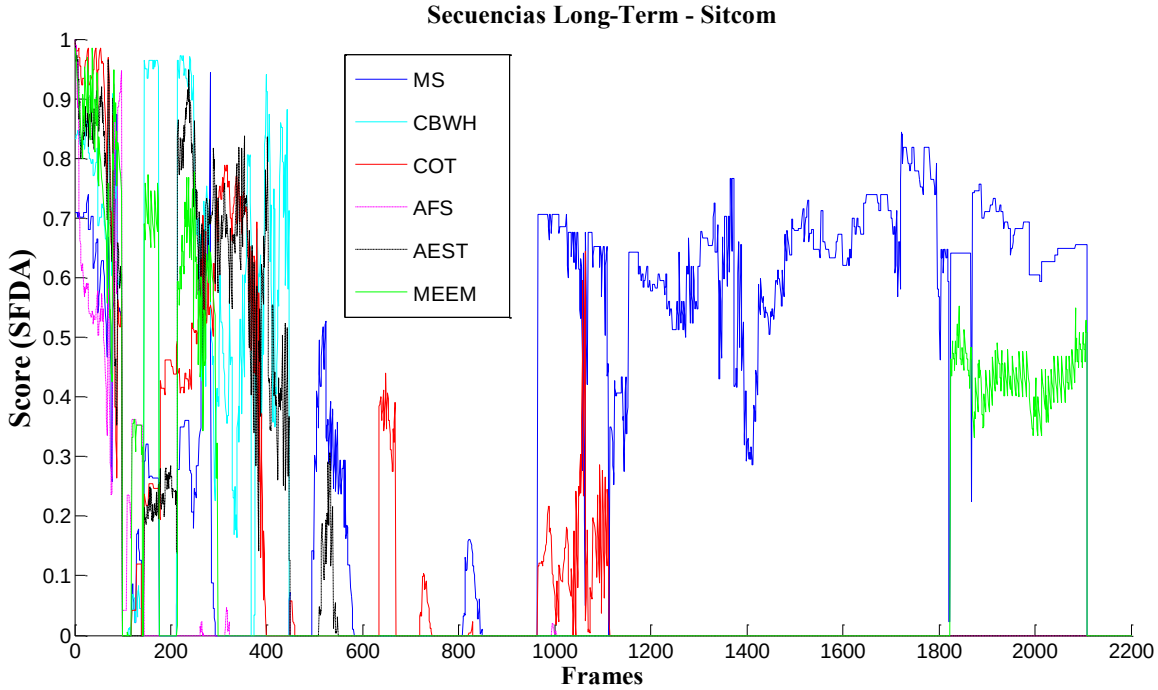


Figura 5.26: Gráfica resultados SFDA de los seis trackers para la secuencia Sitcom.

En el caso de ambas gráficas de los coeficientes CLE y SFDA (Figuras 5.25 y 5.26) se ha editado la representación del eje de los Frames, dado que a partir de los 2200 frames la mayoría de los algoritmos no obtenían ningún valor de solapamiento. Además así es posible apreciar ligeramente mejor la parte del inicio de la secuencia, donde todos los trackers están presentes.

La secuencia Sitcom presenta como principal dificultad las salidas de la imagen del objeto de interés. Al analizar sus resultados, destaca en primer lugar que el algoritmo que consigue un mejor rendimiento es el MS, siendo éste el más básico de todos. Un motivo puede ser que a pesar de tratarse de un vídeo de largo plazo, no cuenta con demasiados problemas. Además otros trackers como CBWH, AFS y MEEM responden de la misma forma que MS a los problemas que surgen durante la secuencia (por ejemplo de los frames 2100 a 2400, que se produce una salida de la imagen por parte del objeto de interés). A pesar de que el valor final CLE del tracker MS es de 59 píxels, para el resto de algoritmos los resultados se encuentran lejos de ser favorables, todos ellos superando la barrera de los 200 píxels. Es importante también mencionar el comportamiento "contrario" que reflejan los trackers entre sí, mejorando unos cuando empeoran los otros en los mismos frames (en la gráfica este suceso aparece a partir del frame 1200). Esto es debido a que algunos confunden el objeto de interés con otro objeto de similares características, y cuando éste desaparece vuelven a localizar el correcto.
Respecto a los resultados del método de evaluación SFDA, como era de esperar, afirma la buena posición del tracker MS en relación al resto, con un coeficiente SFDA de 0.4. Además es reseñable que el tracker COT alterna frames de solapamiento con otros que no, durante los 1000 frames, mientras que otros como AFS, AEST y CBWH fallan por completo antes de llegar a los primeros 500 frames.

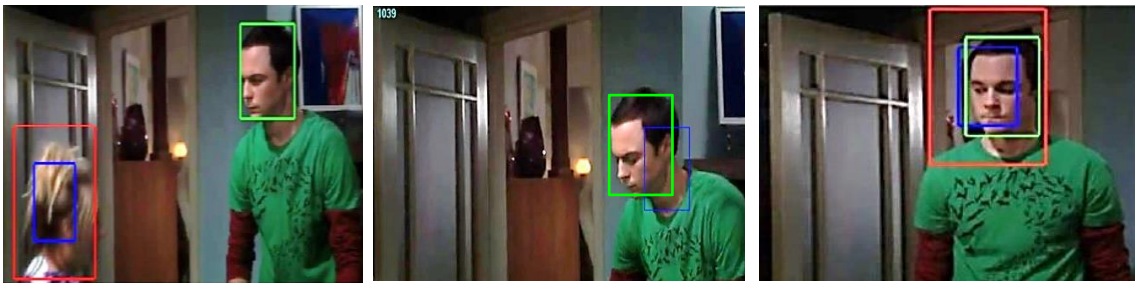


Figura 5.27: Ejemplos de los resultados de distintos trackers durante varios frames para la secuencia Sitcom. De izquierda a derecha: tracker CBWH (frame 1016), tracker COT (frame 1039) y tracker MS (frame 2031). Como se puede observar en las imágenes, el tracker CBWH no sigue correctamente al objeto, el algoritmo COT empieza a alejarse del mismo, mientras que MS sí que mantiene el seguimiento al objeto durante el resto de la secuencia. En verde está representado el ground-truth y en azul la estimación del objeto por parte de cada tracker.

- *Secuencia: NissanSkyline (3712 frames)*

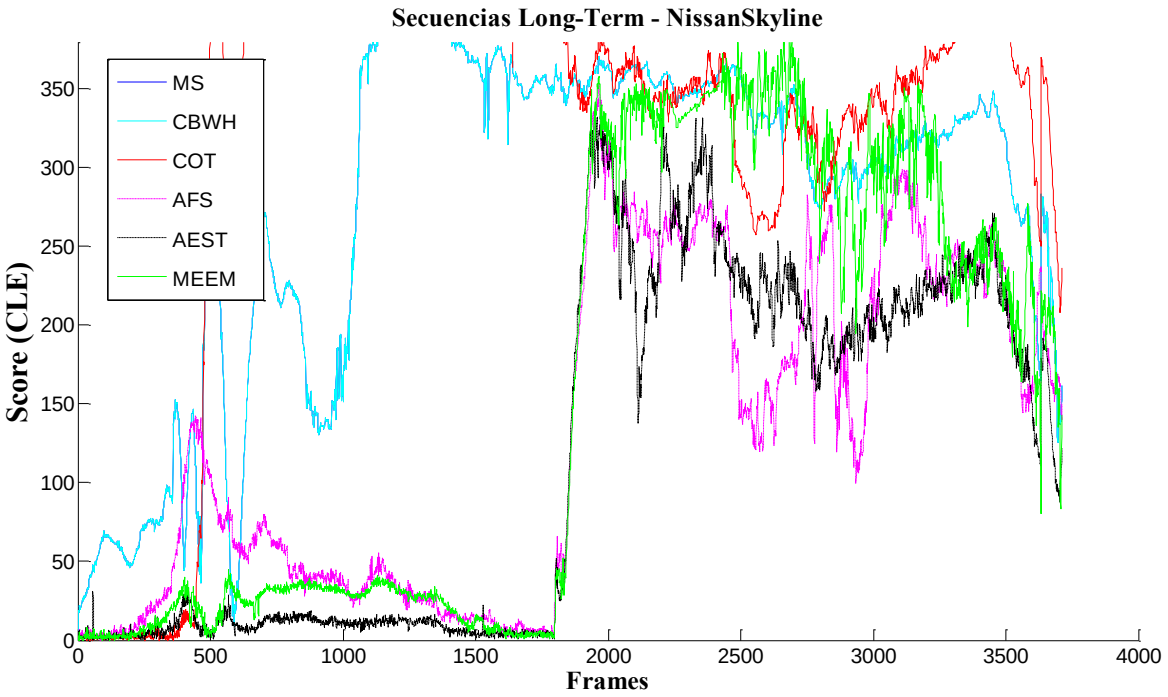


Figura 5.28: Gráfica resultados CLE de los seis trackers para la secuencia NissanSkyline.

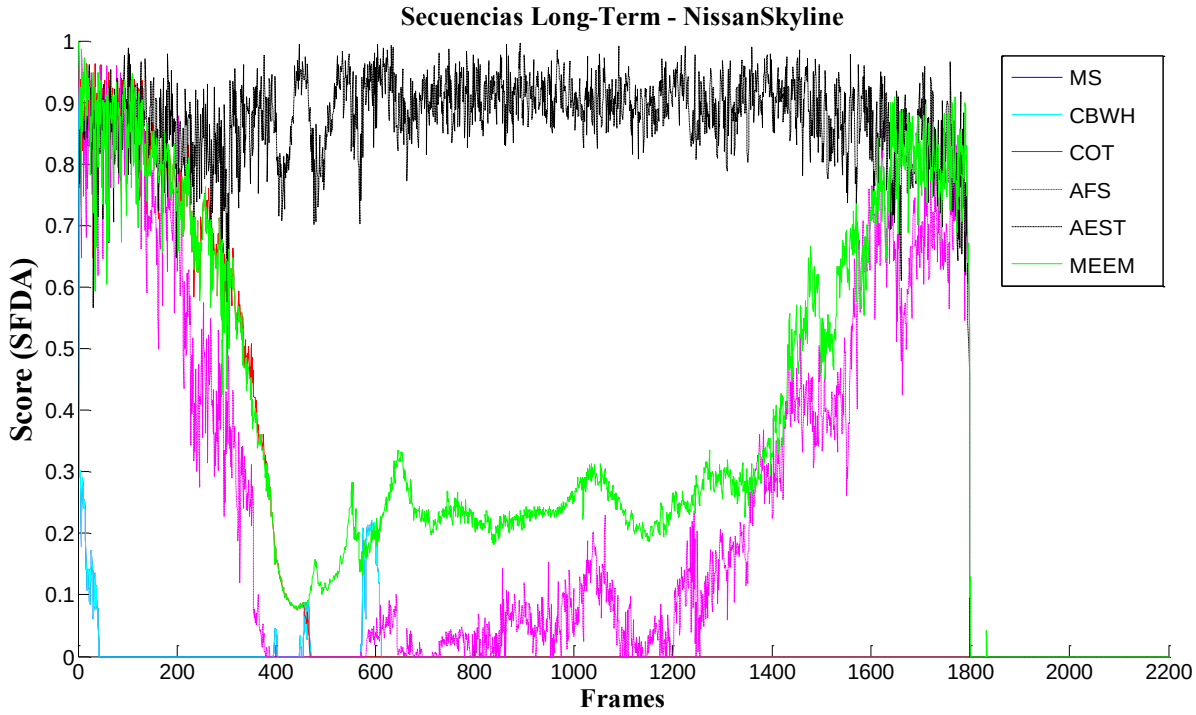


Figura 5.29: Gráfica resultados SFDA de los seis trackers para la secuencia NissanSkyline.

En el caso de la gráfica del coeficiente SFDA (Figura 5.29) se ha editado la representación del eje de los Frames, dado que a partir de los 1900 frames la mayoría de los algoritmos no obtenían ningún valor de solapamiento. Además así es posible apreciar ligeramente mejor la parte del inicio de la secuencia, donde todos los trackers están presentes momentáneamente.

Tal y como se indicaba en la Tabla 5.1, la secuencia NissanSkyline es la que presenta a priori, un menor número de dificultades. La más reseñable es la oclusión total del objeto que tiene lugar en torno al frame 1843. Observando los resultados obtenidos por los algoritmos seleccionados, hay que señalar que en este caso MS y CBWH se comportan de la misma forma (la gráfica de color azul correspondiente a MS coincide con la de CBWH), perdiendo rápido al objeto y obteniendo resultados desfavorables. Por otro lado, hay tres trackers que reflejan una mejor actuación, AFS, AEST y MEEM. Es muy significativo, y se aprecia en las gráficas de las dos formas de evaluación, donde tiene lugar la oclusión total del objeto, produciendo un salto muy brusco en los resultados. Estos algoritmos consideran las oclusiones, pero en esta secuencia, no consiguen volver a localizar el objeto pasados varios frames, de ahí que sus valores finales CLE sean altos (por encima de 100 píxels) y demasiado bajos para el SFDA (por debajo de 0.5).

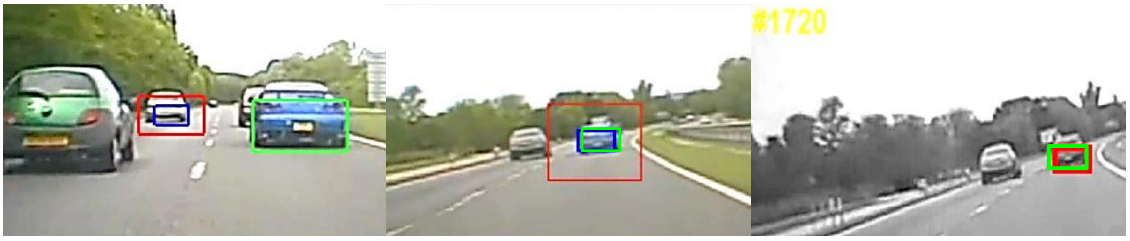


Figura 5.30: Ejemplos de los resultados de distintos trackers durante varios frames para la secuencia NissanSkyline. De izquierda a derecha: tracker MS (frame 883), tracker MEEM (frame 1699) y tracker AFS (frame 1720). Como se puede observar en las imágenes, el tracker MS no sigue correctamente al objeto, mientras que MEEM y AFS sí que mantienen el seguimiento al objeto correctamente hasta que se produzca la oclusión. En verde está representado el ground-truth y en azul (MS y MEEM) y en rojo (AFS), la estimación del objeto por parte de cada tracker.

Figura 5.31: Varios frames que muestran la oclusión que sufre el objeto y el resultado de aplicar el algoritmo AEST. De izquierda a derecha, instantes antes de que se produzca la oclusión, el tracker sigue correctamente al objeto (frame 1794); momento en el que se produce la oclusión total (frame 1843); el objeto vuelve a ser visible tras la oclusión pero el tracker ya no lo sigue correctamente (frame 1874). En verde se muestra el ground-truth y el rectángulo rojo indica la estimación del objeto.

- *Secuencia: Volkswagen (4330 frames)*

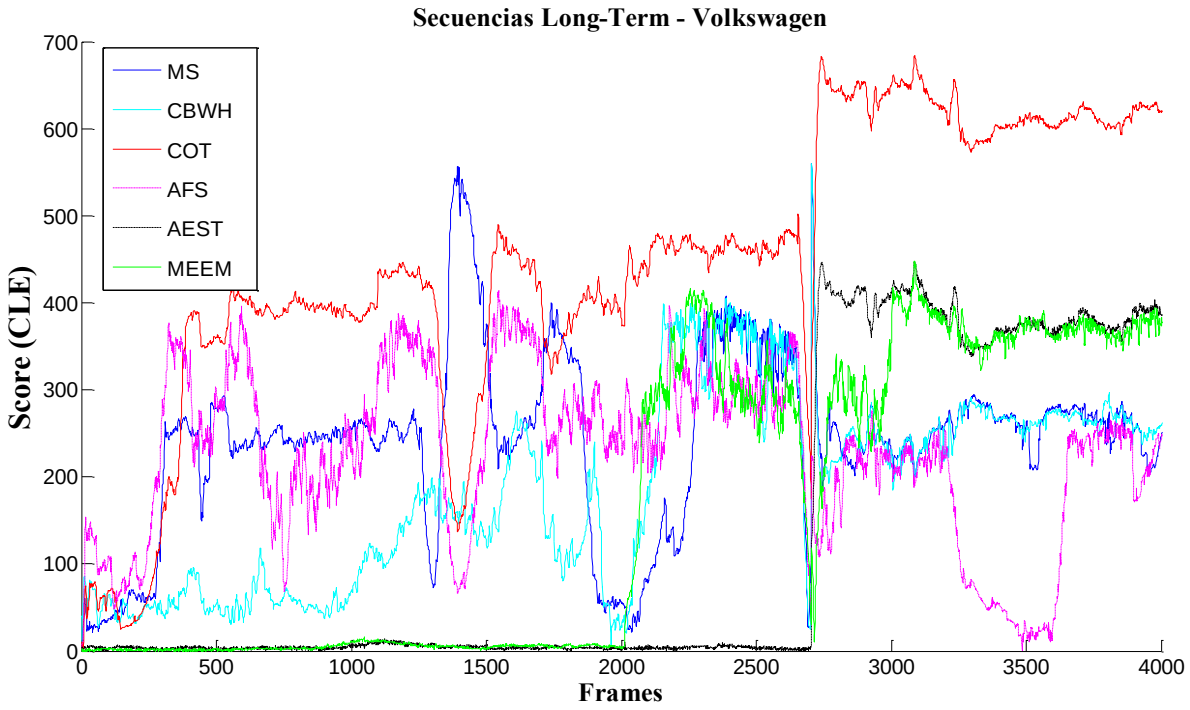


Figura 5.32: Gráfica resultados CLE de los seis trackers para la secuencia Volkswagen.

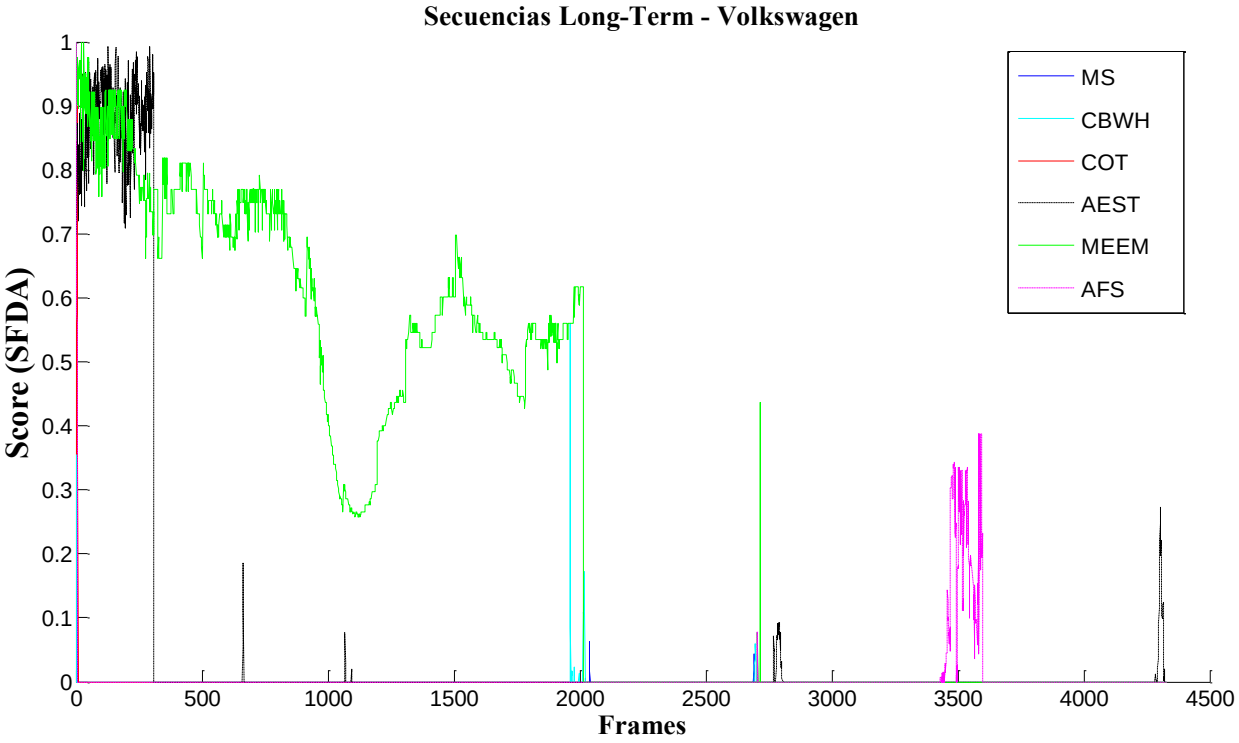


Figura 5.33: Gráfica resultados SFDA de los seis trackers para la secuencia Volkswagen.

La secuencia Volkswagen presenta como principales problemas la similitud constante entre el objeto de interés y el resto de elementos del vídeo, así como la iluminación y color de la secuencia. Además también trata el problema de la salida del objeto del plano de la cámara durante varios frames consecutivos, siendo este desafío un reto que provoca que los algoritmos de seguimiento seleccionados no consigan recuperar al objeto salvo en contadas excepciones.
Analizando los resultados mostrados en las Figuras 5.32 y 5.33, se deduce rápidamente que el algoritmo con mejor rendimiento es MEEM. Conviene explicar esta decisión, ya que si se observa únicamente la gráfica de la evaluación CLE, indica que el valor medio final del tracker AEST está por debajo del de MEEM, y por lo tanto sería mejor. Sin embargo, al observar los valores del coeficiente SFDA para ambos algoritmos, se aprecia como el solapamiento de AEST desciende bruscamente ya en el inicio de la secuencia (frame 309), mientras que para MEEM este descenso se produce en la mitad de la secuencia (frame 2014). Además el valor final SFDA de MEEM es mayor que el del tracker AEST. Esta diferencia se debe a una razón, y es que el tracker AEST en esta secuencia sigue correctamente al objeto durante los primeros 300 frames para luego dejar de seguirle pero sin alejarse en exceso, por lo que el valor del CLE no aumenta considerablemente como les sucede a otros trackers. En cambio, en la evaluación del solape entre las cajas, al no detectar el objeto ni una parte de él, el solape es nulo y se produce el descenso brusco obtenido en su correspondiente gráfica.
Otros rasgos a destacar son el pico (en la evaluación SFDA) o el descenso (en la evaluación CLE) mostrado por el tracker AFS alrededor de los 3500 frames cuando fallaba desde el inicio prácticamente, y los malos resultados como podía suponerse del algoritmo COT, cuya característica principal para el modelo es el color.

Figura 5.34: Ejemplos de los resultados del tracker MEEM durante varios frames para la secuencia Volkswagen. De izquierda a derecha, en el frame 1511 el algoritmo sigue correctamente al objeto; en el frame 2033 confunde el objeto de interés con la señal y a partir de ese momento y para el resto del vídeo el algoritmo falla. El rectángulo verde corresponde al ground-truth y el azul a la posición estimada del objeto.

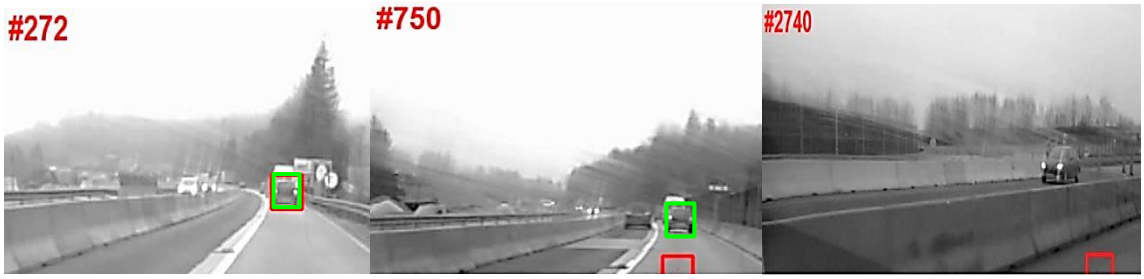


Figura 5.35: Ejemplos de los resultados del tracker AEST durante varios frames para la secuencia Volkswagen. De izquierda a derecha, en el frame 272 el tracker sigue de forma clara al objeto; en el frame 750 ya falla el algoritmo, pero su posición está relativamente cerca del objeto; y también aparece un frame donde se ha producido la salida del objeto del plano de la imagen. El rectángulo rojo muestra los resultados del tracker AEST y el verde es el ground-truth del vídeo en cuestión.

- *Secuencia: Carchase (8660 frames)*

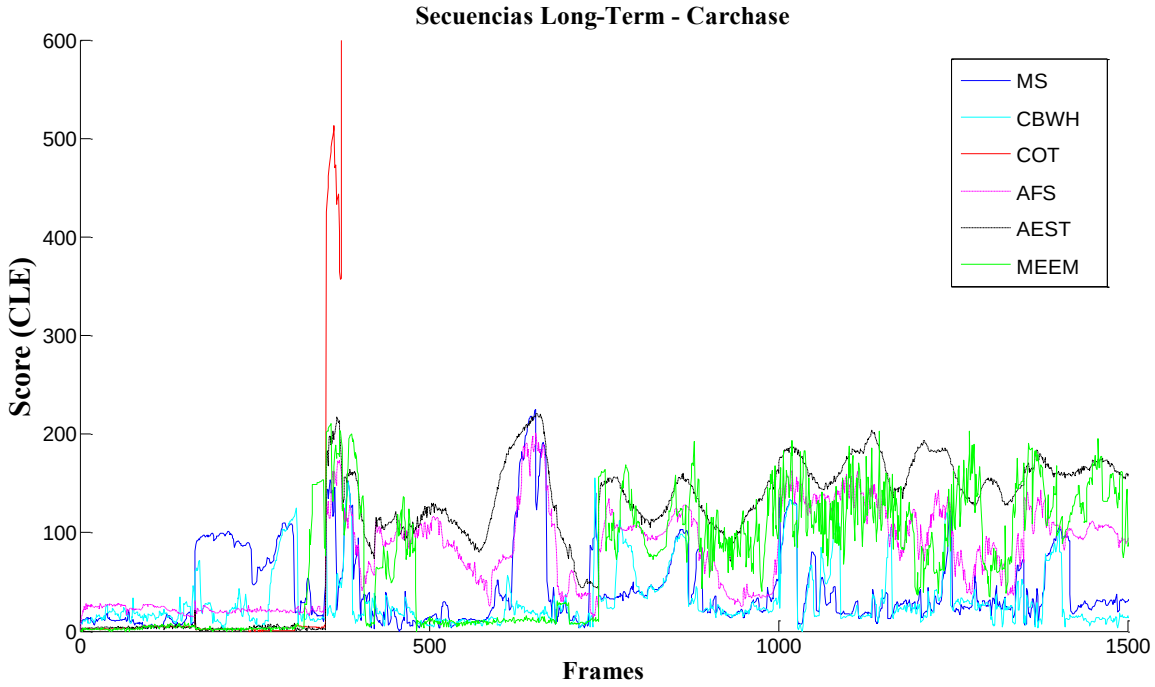


Figura 5.36: Gráfica resultados CLE de los seis trackers para la secuencia Carchase.

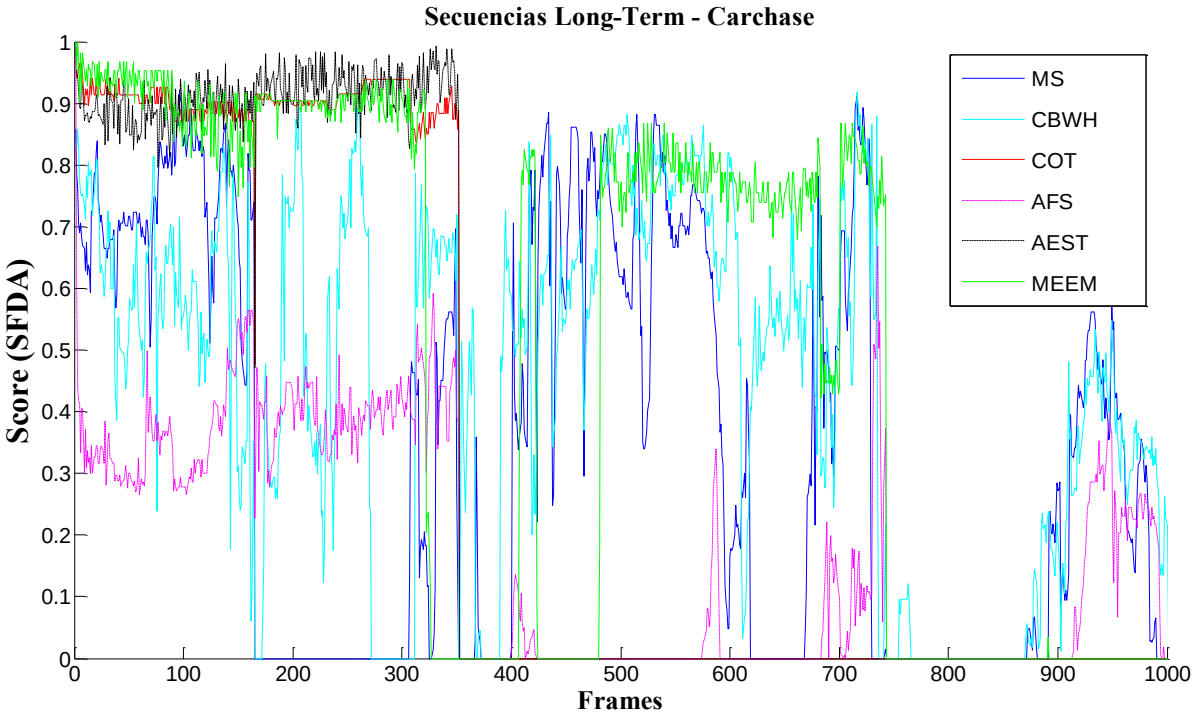


Figura 5.37: Gráfica resultados SFDA de los seis trackers para la secuencia Carchase.

En el caso de la gráfica del coeficiente CLE (Figura 5.36) se ha editado la representación del eje del Score, dado que el tracker COT proporciona unos resultados muy altos (CLE = 1500). A partir del frame 1000, el comportamiento de todos los trackers no sufre grandes alteraciones.

En el caso de la gráfica del coeficiente SFDA (Figura 5.37) se ha editado la representación del eje de los Frames, para poder destacar mejor a los trackers que alcancen los mejores valores. Además así es posible apreciar ligeramente mejor la parte del inicio de la secuencia, donde todos los trackers están presentes momentáneamente.

La secuencia Carchase presenta varios desafíos que la convierten en el vídeo con más problemas para los algoritmos seleccionados para el largo plazo. Oclusiones, similitud entre objetos o cambios de escala, son alguno de ellos.
Exceptuando el tracker COT mencionado anteriormente, el resto de algoritmos se comportan de forma parecida, donde predominan los ascensos y descensos bruscos durante todo el vídeo. Para deducir quiénes son los que obtienen los mejores rendimientos, es preferible analizar la Figura 5.37, donde dominan los colores azules, que se corresponden con los trackers más sencillos, MS y CBWH curiosamente. Sin embargo su coeficiente SFDA final es bajo y no alcanza el valor de 0,3 en ninguno de los casos. Se puede concluir que ninguno de los seis trackers analizados consigue resultados satisfactorios para esta secuencia.

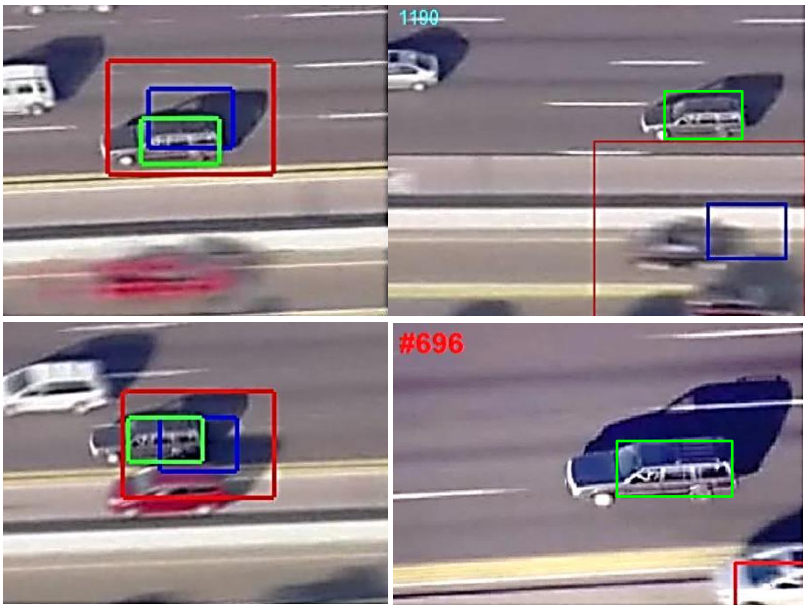


Figura 5.38: Ejemplos de los resultados de distintos trackers durante varios frames para la secuencia Carchase. De izquierda a derecha y de arriba a abajo: tracker MS (frame 1371), tracker MEEM (frame 1190), tracker CBWH (frame 1418) y tracker AEST (frame 696). Como se puede observar en las imágenes, los trackers MS y CBWH siguen correctamente al objeto durante gran parte de la secuencia, mientras que MEEM y AEST debido los diversos problemas que plantea el vídeo desde el inicio no mantienen el seguimiento del objeto. En verde está representado el ground-truth y en azul (MS, MEEM y CBWH) y en rojo (AEST), la estimación del objeto por parte de cada tracker.

- *Secuencia: LiverRun (28658 frames)*

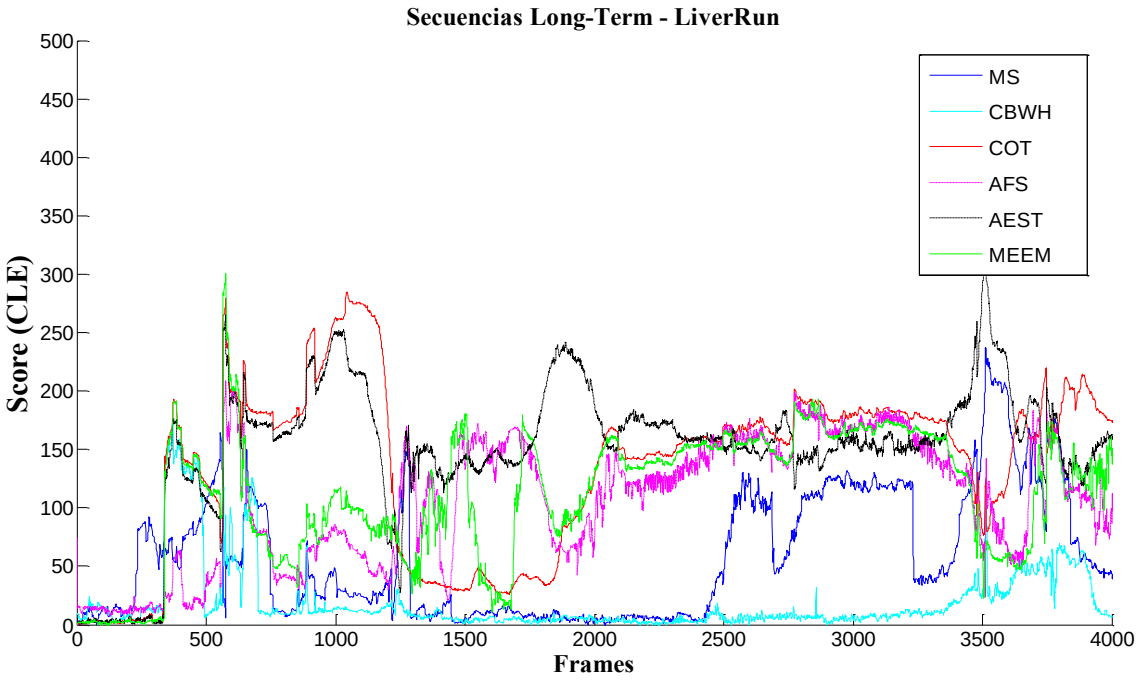


Figura 5.39: Gráfica resultados CLE de los seis trackers para la secuencia LiverRun.

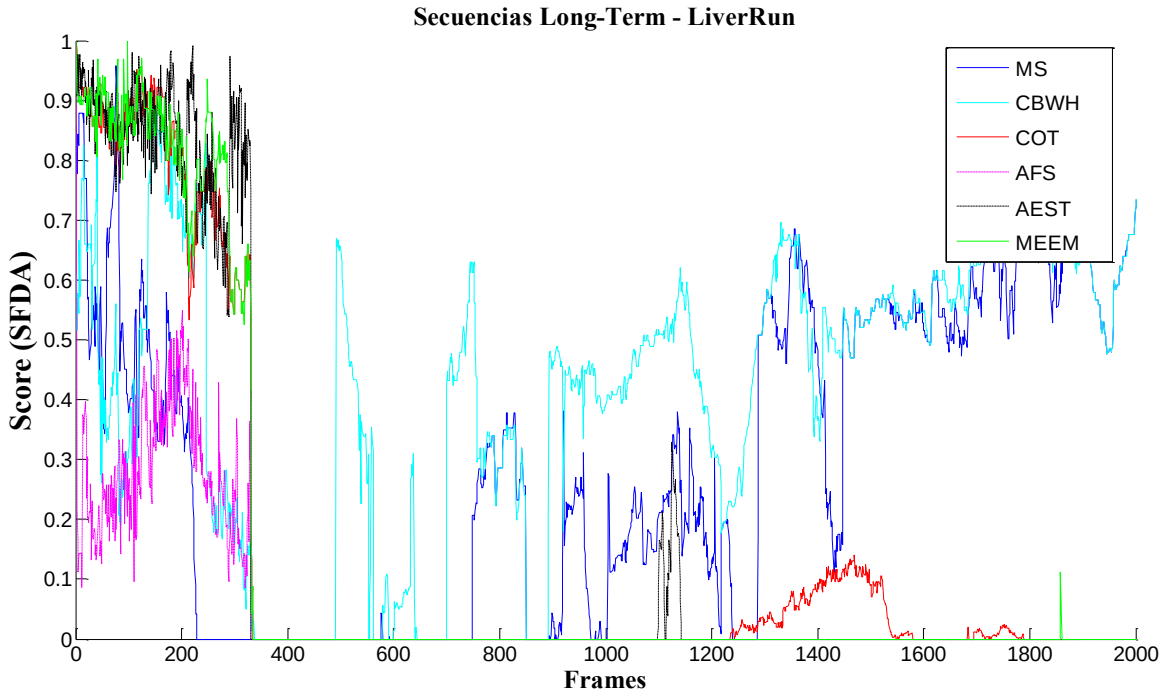


Figura 5.40: Gráfica resultados SFDA de los seis trackers para la secuencia LiverRun.

Al tratarse de una secuencia tan larga, en ambas gráficas se ha editado la representación del eje de los Frames, para poder destacar mejor a los trackers que alcancen los mejores valores. Además así es posible apreciar ligeramente mejor la parte del inicio de la secuencia, donde todos los trackers están presentes momentáneamente.

El análisis de la secuencia LiverRun resulta bastante similar al realizado para la secuencia Carchase, si bien hay que destacar que en este caso, el inicio de todos los trackers es bastante bueno, hasta aproximadamente el frame 338, momento en el que se produce la primera oclusión total del objeto. Tras esto, el tracker que mejor consigue volver a seguir al objeto es CBWH, seguido de MS y AFS. Estos tres algoritmos obtienen un valor CLE final por debajo de los 100 píxels. Sin embargo, en la gráfica de evaluación SFDA, se ve que los valores de solapamiento siguen siendo bajos en todos los casos.
Por tanto, se puede concluir que, con unas características similares a las de la secuencia Carchase, pero con el triple de duración, los algoritmos obtienen resultados ligeramente mejores a los del vídeo Carchase.

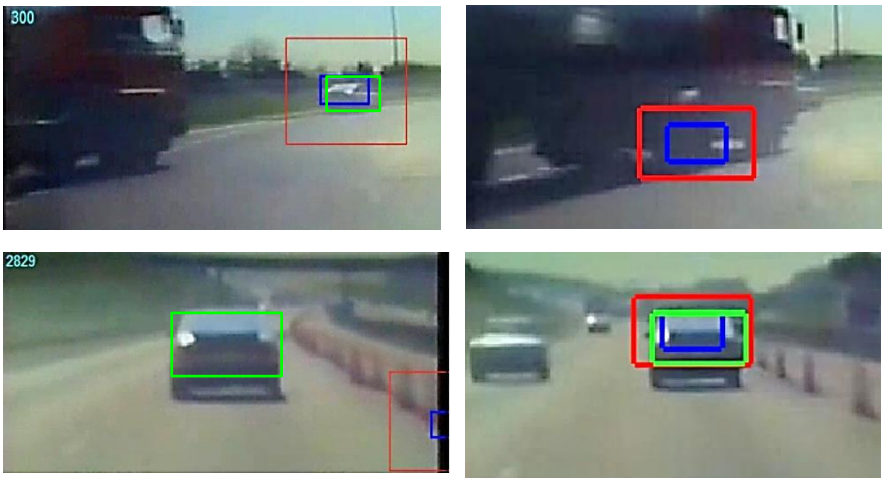


Figura 5.41: Ejemplos de los resultados de distintos trackers durante varios frames para la secuencia LiverRun. De izquierda a derecha y de arriba a abajo: tracker MEEM (frame 300), tracker MS (frame 339), tracker MEEM (frame 2829) y tracker CBWH (frame 3076). Como se puede observar en las imágenes, en el inicio, antes justo de la oclusión (imagen segunda), MEEM sigue correctamente al objeto, después MEEM ya no es capaz de seguirlo bien y en la última imagen, CBWH sí mantiene un buen seguimiento del objeto. En verde está representado el ground-truth y en azul la estimación del objeto por parte de cada tracker.

Finalmente en las siguientes tablas se muestran los valores de CLE (la media) y SFDA (media) para cada tracker en cada secuencia, así como la media total de cada algoritmo.

Tracker\Secuencia	*Motocross*	*Nissan*	*Sitcom*	*Volkswagen*	*Carchase*	*LiverRun*	*Media total*
MS	162	282	**59**	239	**63**	**78**	**147**
CBWH	**162**	282	273	190	**63**	**56**	171
COT	380	333	232	452	14992	195	318(*)
AFS	202	**130**	283	231	114	99	176
AEST	241	**117**	227	**142**	145	173	174
MEEM	**145**	157	**214**	**176**	116	124	**155**

Tabla 5.6: Resultados del Center Location Error medio de los tracker seleccionados para las secuencias de largo plazo. Los dos mejores valores de cada tracker se representan en verde (1º) y rojo (2º).

(*) El cálculo de la media total del algoritmo COT se ha realizado con cinco secuencias, ignorando su resultado en el vídeo Carchase, debido que su valor se aleja demasiado del resto de rangos de resultados.

Tracker\Secuencia	*Motocross*	*Nissan*	*Sitcom*	*Volkswagen*	*Carchase*	*LiverRun*	*Media total*
MS	**0.034**	0.004	**0.398**	0.001	**0.222**	**0.095**	**0.125**
CBWH	**0.061**	0.004	0.079	0.001	**0.269**	**0.227**	0.106
COT	0.002	0.077	0.080	0.001	0.036	0.009	0.034
AFS	0.003	0.137	0.023	0.008	0.022	0.027	0.036
AEST	0.011	**0.428**	0.079	**0.063**	0.037	0.014	0.105
MEEM	0.029	**0.208**	**0.117**	**0.288**	0.060	0.045	**0.124**

Tabla 5.7: Resultados del SFDA medio de los tracker seleccionados para las secuencias de largo plazo. Los dos mejores valores de cada secuencia se representan en verde (1º) y rojo (2º).

Una vez que se han obtenido los resultados de los trackers para cada una de las secuencias, tanto de corto como de largo plazo, es conveniente realizar un breve análisis de los mismos, sobre todo en relación al comportamiento en el objetivo principal, que es el dataset de largo plazo.

Empezando por las últimas tablas conseguidas (Tablas 5.6 y 5.7), se distingue que de forma sorprendente, el tracker que mejor resultados presenta es Mean – Shift (MS), curiosamente el más básico de todos, seguido de MEEM. Después aparecen CBWH y AEST con valores muy similares entre ellos y cerrando el ranking se encuentran AFS y COT respectivamente. Esta clasificación pone de manifiesto que al no estar ningún tracker enfocado especialmente al largo plazo, cualquiera de ellos puede obtener buenos resultados en un vídeo concreto, mientras que para la siguiente secuencia de largo plazo generar valores negativos. Esta situación es relativamente común en el escenario del seguimiento de objetos y se hace aún más notoria en el ámbito del largo plazo. También es necesario apreciar que los valores finales conseguidos por los seis algoritmos no son demasiados buenos, con medias muy altas en el caso del CLE (> 100 píxels) y unos coeficientes SFDA excesivamente bajos (< 0.150) , por lo que el seguimiento de objetos en vídeos de largo plazo de estos algoritmos para este conjunto de secuencias todavía tiene mucho por mejorar.

Enlazando este estudio con las Tablas 5.4 y 5.5 de las secuencias de corto plazo, se observa que los resultados finales mejoran considerablemente. En esta parte, el comportamiento sí refleja algo esperado, siendo MS el que peores datos consigue. De nuevo MEEM ocupa las posiciones de cabeza y esta vez con resultados positivos, con un CLE medio de 37 píxels y un SFDA por encima de 0.5, concretamente 0.615. Además otros dos algoritmos, COT y AEST también consiguen valores bastante favorables en ambas evaluaciones. Queda claro que la longitud del vídeo, y todos los desafíos que conlleva, influye de manera notoria en los resultados de los algoritmos que se utilicen para el seguimiento de objetos.

A continuación se muestra una tabla para el conjunto de secuencias de largo plazo en la que se muestran los frames iniciales en los que el tracker consigue seguir al objeto de manera relativamente correcta. Se han definido las siguientes condiciones: se considera un seguimiento correcto todos los frames cuyo valor SFDA sea > 0; se estima que el algoritmo ha perdido el objeto si presenta más de 20 frames consecutivos con un resultado de SFDA = 0. Esta última condición es para evitar posibles casos en los que el objeto salga del ángulo de visión de la imagen en los primeros frames, provocando esto que no refleje el comportamiento real del algoritmo para el resto del vídeo.

Tracker\Secuencia	*Motocross*	*Nissan*	*Sitcom*	*Volkswagen*	*Carchase*	*LiverRun*	*Media total*
MS	16	42	**175**	6	164	228	105
CBWH	16	42	**175**	1	271	**336**	140
COT	**17**	467	**398**	8	**352**	**334**	263
AFS	**17**	386	143	7	**352**	330	206
AEST	**17**	**1799**	98	**308**	**352**	330	**484**
MEEM	**61**	**1800**	**175**	**2013**	**326**	**334**	**785**

Tabla 5.8: Resultados de los frames iniciales en los que los tracker seleccionados para las secuencias de largo plazo obtienen un SFDA > 0. Los dos mejores valores de cada secuencia se representan en verde (1º) y rojo (2º).

Estudiando la Tabla 5.8 se aprecia que el tracker que tiene un mecanismo de inicialización mejor es el MEEM, superando con amplia mayoría al resto de algoritmo analizados. Además de estos resultados se pueden derivar dos afirmaciones importantes. En primer lugar, queda clara la evidencia del gran progreso que se está llevando a cabo en el desarrollo de algoritmos de seguimiento de objetos. Se ve en este ejemplo que a medida que las investigaciones en este campo continúan creciendo, los trackers mejoran sus técnicas y mecanismos de inicialización obteniendo mejores resultados. En el caso representado en este PFC, MS consigue un valor medio muy por debajo del resto. CBWH lo mejora, pero no obtiene una cantidad significativa para el largo plazo. En el polo opuesto se encuentran los trackers MEEM y AEST, que sí logran resultados más destacados.
La otra consideración importante, a tenor de estos valores con los obtenidos en las Tablas 5.6 y 5.7, donde MS obtenía buenas cifras, es que este algoritmo se recupera más veces y mejor que el resto de trackers, ya que siendo el que peores inicios presenta, acaba logrando mejores resultados que el resto.

5.6. Resultados del algoritmo propuesto

A continuación se exponen los resultados en los vídeos más representativos obtenidos para el algoritmo de fusión desarrollado en el Capítulo 4. Otras secuencias de corto plazo quedan representadas en el Anexo A de este PFC. Se ha representado el valor SFDA a lo largo de los frames de forma análoga a las secciones 5.4 y 5.5 de este capítulo.
Para cada secuencia se muestran los resultados en el primer frame de la misma, varios frames intermedios y el bbox de la combinación en el último frame. El rectángulo obtenido en todas las secuencias aparece representado por el color negro (salvo en los vídeos Matrix, Carchase y LiverRun que se ha dibujado de color blanco para facilitar su apreciación).

Además al final de esta sección se presenta una tabla con los valores del coeficiente SFDA obtenido por el bbox resultado en cada una de las secuencias del algoritmo desarrollado.

Secuencias de largo plazo

- *Secuencia: Sitcom (3898 frames)*

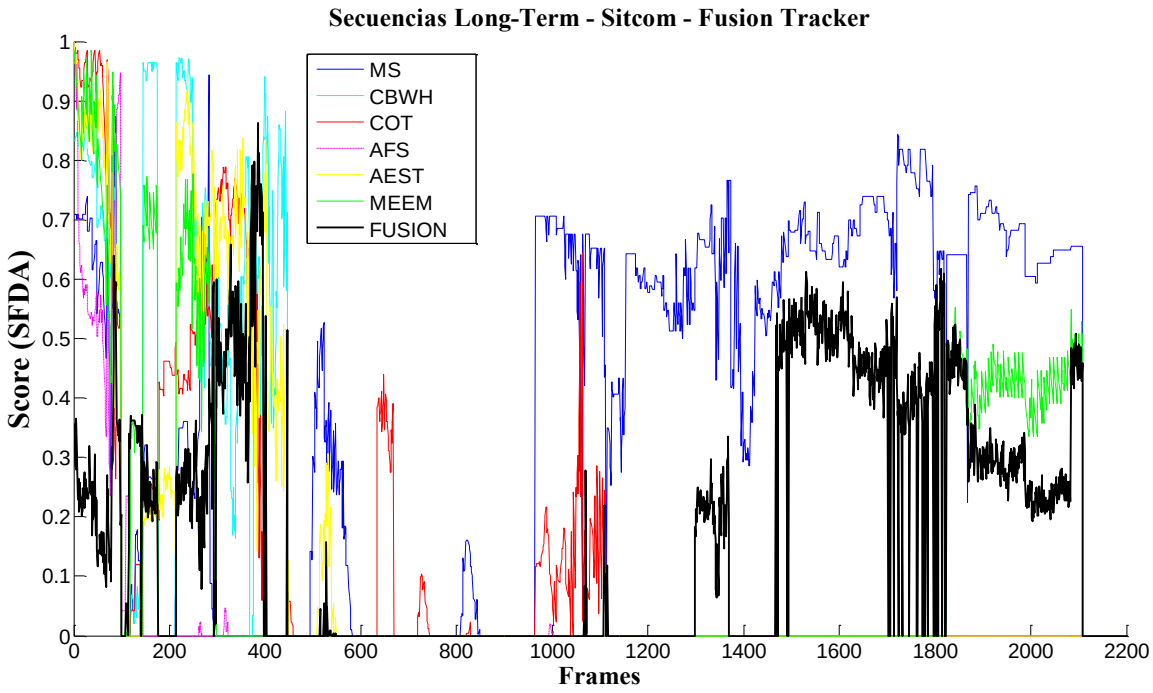


Figura 5.42: Gráfica resultados SFDA de los seis trackers más el algoritmo propuesto (color negro) para la secuencia Sitcom.

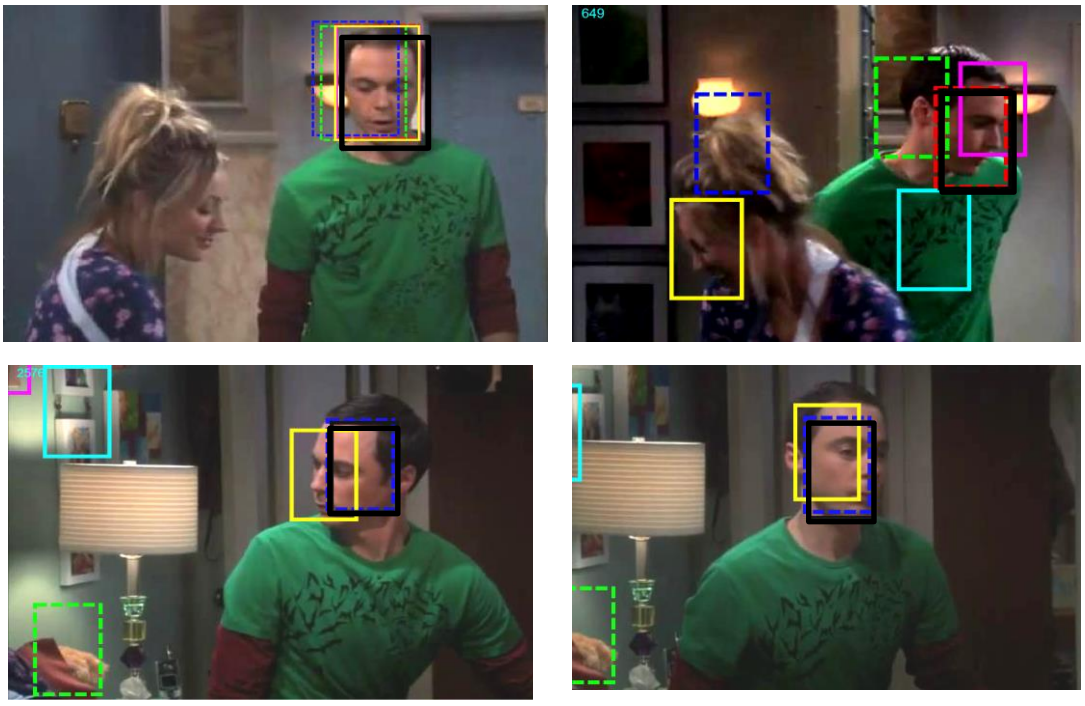


Figura 5.43: Representación del bbox resultado (color negro) de la combinación de trackers durante los frames 1, 649, 2576 y 3898 de la secuencia Sitcom. El resto de rectángulos representan a MS (azul de líneas discontinuas), CBWH (verde de líneas

discontinuas), COT (rojo de líneas discontinuas), AFS (cyan), AEST (magenta) y MEEM (amarillo).

El funcionamiento del algoritmo propuesto presenta para este vídeo de largo plazo un buen comportamiento durante todo el tiempo, actualizando correctamente su posición en función de los trackers que presenten mayor solapamiento para cada frame. Esto se pone de manifiesto en los frames mostrados en la Figura 5.43 ya que en el frame 649 los trackers representados por los rectángulos verde, rojo y magenta son los que obtienen mejores resultados. Sin embargo con el paso del tiempo, sus resultados empeoran y son los bbox amarillo y azul los que presentan un mejor comportamiento, y con ellos, el rectángulo negro del algoritmo de fusión.

- *Secuencia: Carchase (9886 frames)*

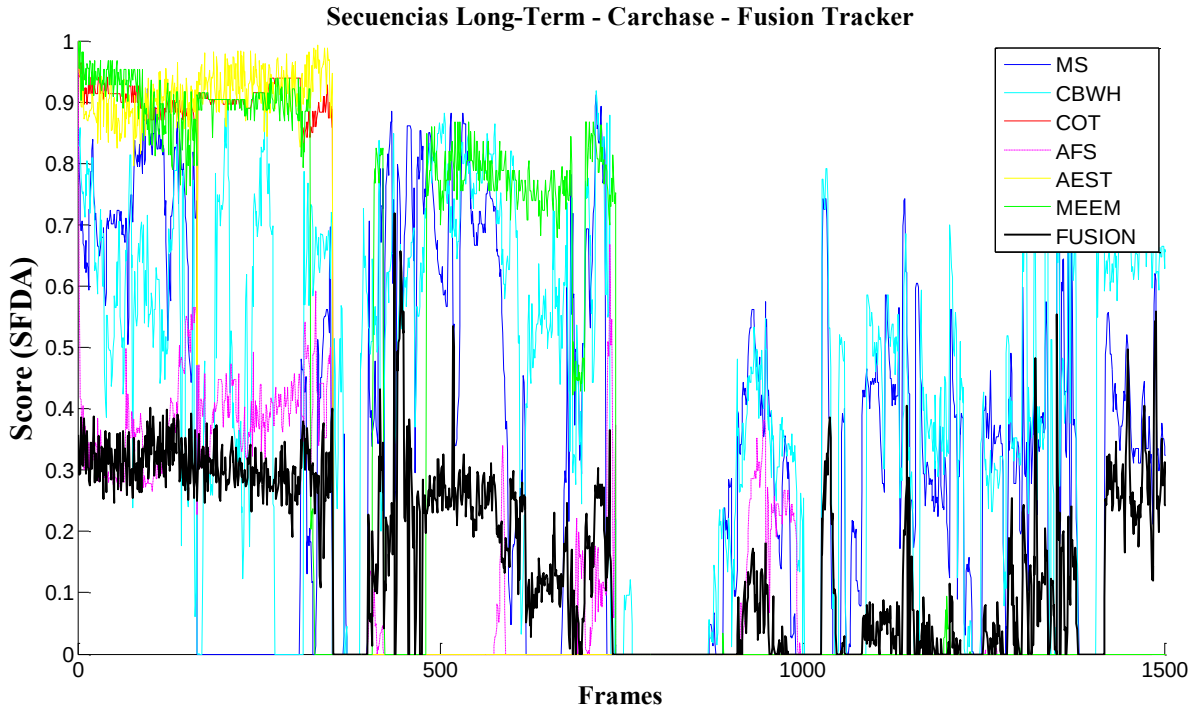


Figura 5.44: Gráfica resultados SFDA de los seis trackers más el algoritmo propuesto (color negro) para la secuencia Carchase.

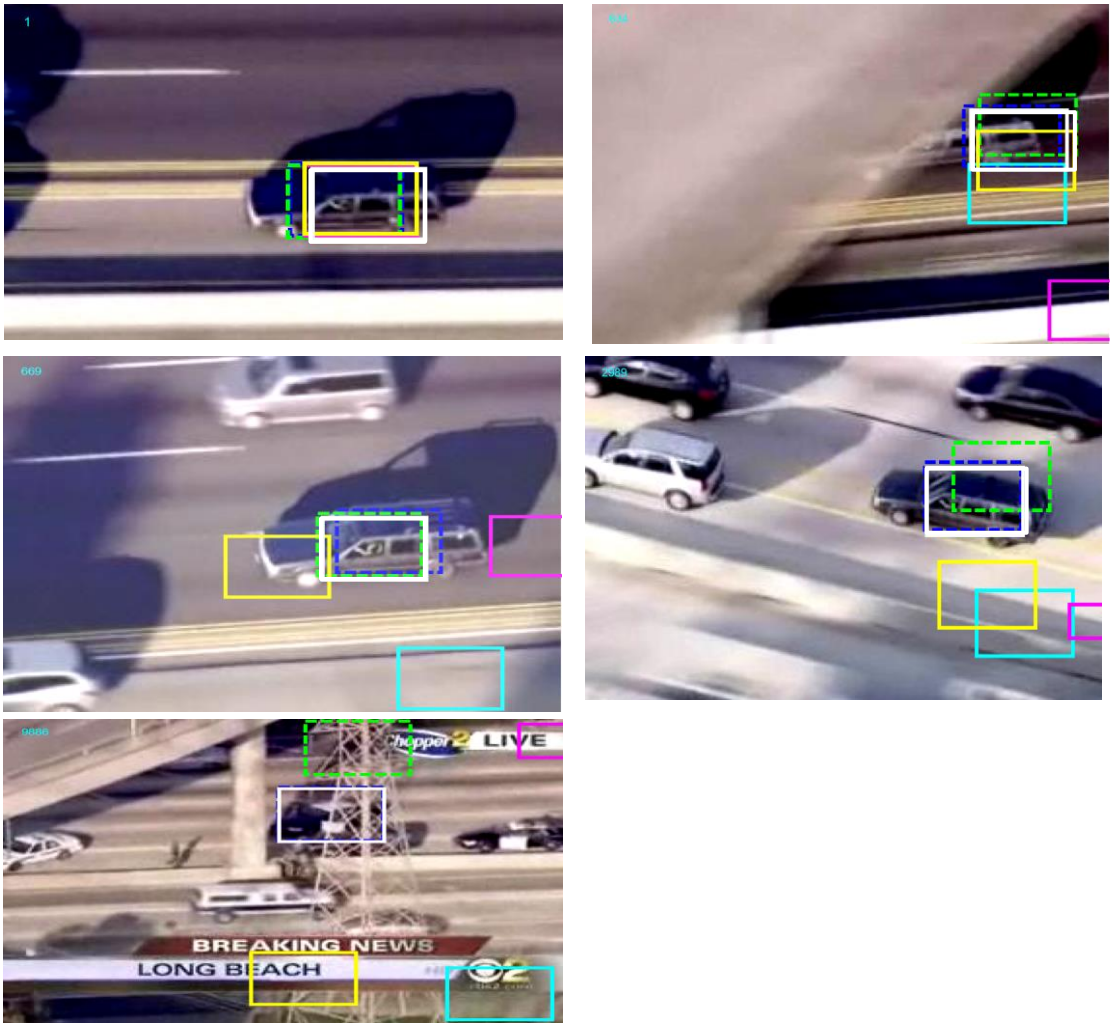


Figura 5.45: Representación del bbox resultado (color blanco) de la combinación de trackers durante los frames 1, 634, 669, 2989 y 9886 de la secuencia Carchase. El resto de rectángulos representan a MS (azul de líneas discontinuas), CBWH (verde de líneas discontinuas), COT (rojo de líneas discontinuas), AFS (cyan), AEST (magenta) y MEEM (amarillo).

Se han representado varios frames intermedios del vídeo de largo plazo Carchase puesto que cabe destacar varias situaciones a considerar. Entre los frames número 634 y 669 tiene lugar una oclusión total del objeto. En el frame 634 se puede ver como justo antes de que se produzca este problema el algoritmo implementado sigue al objeto (rectángulo blanco). Poco después de que termine la oclusión, el resultado continúa siendo favorable ya que a diferencia de otros algoritmos que han perdido el objeto, el bbox de la fusión sigue detectándolo y siguiéndolo.
Por otra parte, también se ha mostrado otro frame intermedio (2989) en el que se presenta un caso en el que se produce un solapamiento de trackers dos a dos. Como se explicó en el Capítulo 4, el algoritmo propuesto buscará aquellos que consigan un solapamiento mayor. En este ejemplo, el solape entre los bounding box de color azul y verde es mayor que el de color amarillo y cyan, y por ello el rectángulo blanco correspondiente a la fusión de trackers presenta ese resultado.

- *Secuencia: LiverRun (29597 frames)*

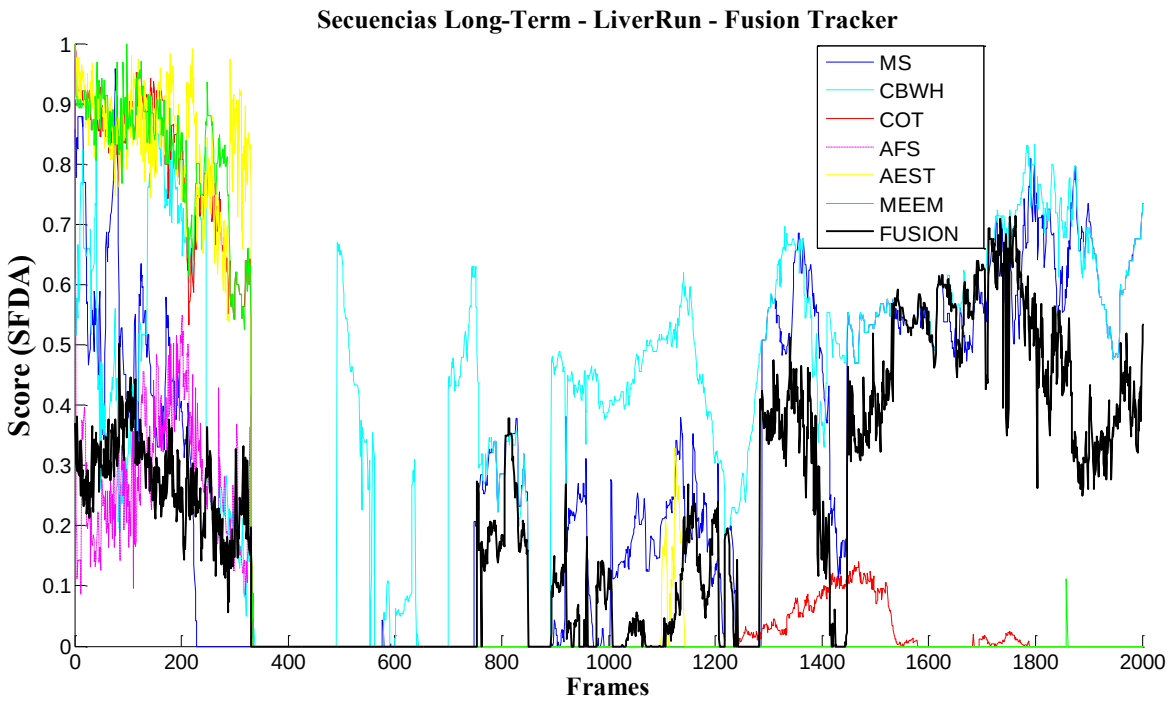


Figura 5.46: Gráfica resultados SFDA de los seis trackers más el algoritmo propuesto (color negro) para la secuencia LiverRun.

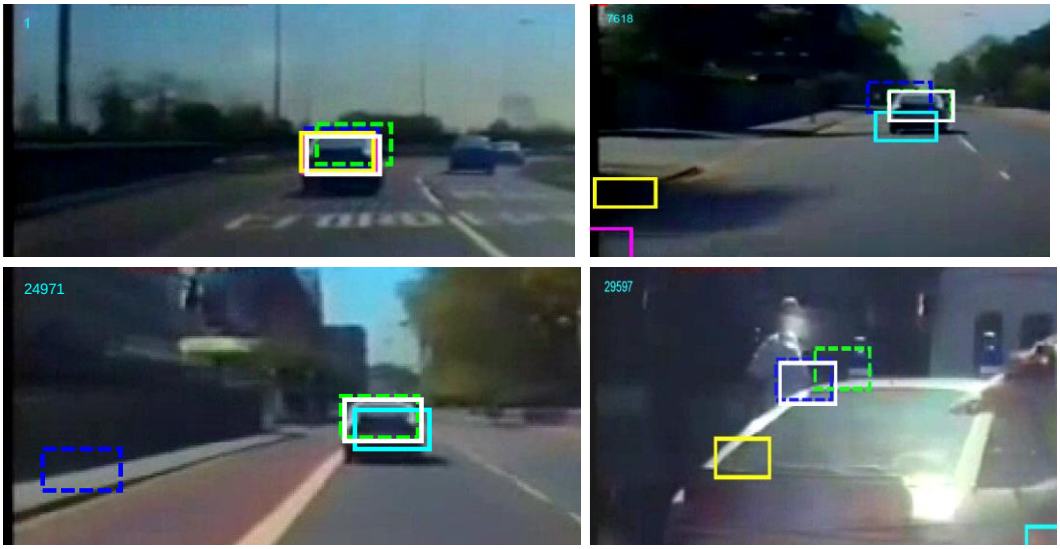


Figura 5.47: Representación del bbox resultado (color blanco) de la combinación de trackers durante los frames 1, 7618, 24971 y 29597 de la secuencia LiverRun. El resto de rectángulos representan a MS (azul de líneas discontinuas), CBWH (verde de líneas discontinuas), COT (rojo de líneas discontinuas), AFS (cyan), AEST (magenta) y MEEM (amarillo).

En los ejemplos mostrados para la secuencia más larga seleccionada, LiverRun, se comprueba que el algoritmo propuesto sigue presentando buenos resultados con el paso del tiempo, debido a que otros trackers también actúan correctamente. Se puede observar que tanto en el frame 7618 como en el 24971, ya casi al final de la secuencia, el bounding box de la combinación sigue con el objeto de interés aunque otros algoritmos que en otras secuencias presentaban buenos números (rectángulos azul y amarillo en Sitcom) no lo hacen.

A continuación se muestran en las siguientes tablas los valores del coeficiente SFDA (media) para el algoritmo propuesto para cada secuencia de corto y de largo plazo, así como su media total. Aparecen también los valores de los demás algoritmo para facilitar su comparación y análisis.

Tracker\Secuencia	*Bolt*	*Coke*	*Skiing*	*Soccer*	*Deer*	*Matrix*	*Media total*
MS	0.093	0.025	0.040	0.072	0.319	0.007	0.092
CBWH	0.124	0.434	**0.306**	0.121	0.074	0.102	0.193
COT	**0.869**	**0.526**	0.091	**0.593**	**0.852**	0.022	**0.492**
AFS	0.041	0.088	0.079	0.378	0.032	0.080	0.116
AEST	**0.867**	0.299	0.093	0.291	0.541	**0.301**	0.398
MEEM	0.606	**0.776**	**0.551**	**0.412**	**0.850**	**0.495**	**0.615**
FUSIÓN	0.321	0.292	0.145	0.193	0.292	0.103	0.224

Tabla 5.9: Resultados del SFDA medio del algoritmo desarrollado como fusión de trackers para las secuencias de corto plazo. Los dos mejores valores de cada secuencia se representan en verde (1º) y rojo (2º).

Tracker\Secuencia	*Motocross*	*Nissan*	*Sitcom*	*Volkswagen*	*Carchase*	*LiverRun*	*Media total*
MS	**0.034**	0.004	**0.398**	0.001	**0.222**	**0.095**	**0.125**
CBWH	**0.061**	0.004	0.079	0.001	**0.269**	**0.227**	0.106
COT	0.002	0.077	0.080	0.001	0.036	0.009	0.034
AFS	0.003	0.137	0.023	0.008	0.022	0.027	0.036
AEST	0.011	**0.428**	0.079	0.063	0.037	0.014	0.105
MEEM	0.029	**0.208**	0.117	**0.288**	0.060	0.045	**0.124**
FUSIÓN	0.028	0.044	**0.195**	**0.212**	0.109	0.058	0.108

Tabla 5.10: Resultados del SFDA medio del algoritmo desarrollado como fusión de trackers para las secuencias de largo plazo. Los dos mejores valores de cada secuencia se representan en verde (1º) y rojo (2º).

Los resultados obtenidos por el algoritmo derivado de la fusión de los trackers muestran un comportamiento final parejo a los trackers CBWH y AEST, que ocupan las posiciones intermedias del conjunto de algoritmos seleccionados. Según se ha ido viendo a lo largo de esta memoria y en relación a su ejecución en los vídeos escogidos, el algoritmo de fusión presenta resultados acordes a sus características y funcionamiento. Por una parte mejora a varios algoritmos, ya que en su ejecución busca la menor distancia o el mayor solapamiento posible entre trackers. Sin embargo, en ciertas secuencias ocurre que la mayor parte de los algoritmos no siguen correctamente al objeto, siendo un único algoritmo el que actúa adecuadamente. Si el resto de trackers se encuentran muy juntos, esto provocará el algoritmo de fusión también derive resultados desfavorables, y de ahí que no logre los mejores resultados de los seis algoritmos.

Finalmente destacar también que se han llevado a cabo varias alternativas para encontrar finalmente el bbox resultado más óptimo entre las opciones barajadas. En primer lugar se comenzó empleando el valor del parámetro σ de la Ecuación 22 proporcionado en [55], siendo de 0.03. Sin embargo, pronto se comprobó que al disponer de seis trackers y de doce secuencias (seis de ellas de largo plazo que suman un total de 53378 frames) los valores finales de a_res tendían a ser extremadamente altos y poco representativos. Por esta razón, se decidió usar un valor de $\sigma = 5$ que proporciona resultados más razonables.

Por otra parte, se probaron tres maneras distintas de obtener el bbox candidato en cada uno de los frames de las secuencias. En la siguiente imagen se muestra un ejemplo del bbox candidato obtenido según cada posibilidad. En primer lugar se optó por desplazar la posición y escalar el ancho y alto del bbox resultado de los algoritmos. Sin embargo esto producía demasiada aleatoridad y podía dar lugar a bbox de tamaños muy reducidos o demasiado grandes, derivando en solapamientos o distancias engañosos. Después se descartó escalar la anchura (w) y la altura (h), desplazando la posición (x, y) en función de los valores de su ancho y alto, respectivamente. Si bien esta opción proporcionaba bbox de tamaños precisos, cuando los rectángulos eran demasiado grandes también generaba desplazamientos amplios, y la posición del bbox candidato podía aparecer muy alejada del objeto y de los demás algoritmo. Por lo tanto, siguiendo esta línea se decantó por desplazar la posición de forma más limitada, también en función de su ancho y altura, pero esta vez un valor de w/10 para la coordenada x, y un valor de h/10 para la coordenada y.

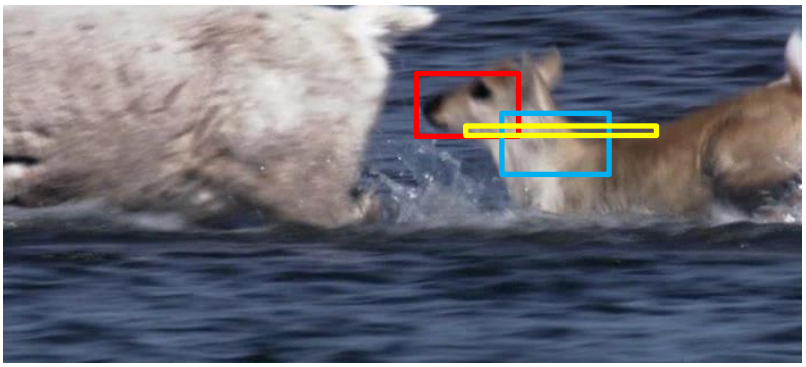


Figura 5.48: Ejemplo de los rectángulos finales obtenidos en la secuencia Deer para cada variante. El rectángulo amarillo corresponde al primer caso (desplazamiento de la posición (x,y) y escalado de w y h). El de color cyan surge de aplicar aleatoridad con un desplazamiento de la posición (x, y) en función de w/2 y h/2. Y el rectángulo rojo es el resultado de variar la posición (x, y) en función de w/10 y h/10. Como se puede observar, esta última variante (bounding box rojo) es la que mejor resultado consigue y la que se ha llevado a cabo a lo largo del proyecto.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este proyecto se ha llevado a cabo un análisis del seguimiento de objetos en un conjunto de secuencias de largo plazo, apoyado por varios vídeos de menor duración, considerados de corto plazo.
Para empezar, se comenzó realizando un detallado estudio del arte del seguimiento de objetos, de donde se obtuvieron los problemas y desafíos más comunes que suelen aparecer cuando se aborda el campo del *video-tracking*. También se observó con detenimiento las distintas etapas en las que todo algoritmo de seguimiento de objetos por lo general, puede estar dividido: Inicialización, Procesamiento y Localización del objeto de interés.
Posteriormente, siguiendo con el estudio del arte, se hizo especial hincapié en el seguimiento de objetos en vídeo de largo plazo, investigando su definición y diferentes enfoques realizados hasta la fecha. Tras examinar varias aproximaciones, se realizó una clasificación y comparación de los aspectos más característicos de los enfoques estudiados en la literatura (Tabla 2.1).

Acto seguido, se pasó al análisis y selección de los algoritmos que posteriormente serían ejecutados en las distintas secuencias de vídeos. Debido al gran número de trackers existentes para el seguimiento de objetos de corto plazo, y a la reciente aparición de algoritmos que intentan evitar los problemas presentas en largo periodos de tiempo, se decidió seleccionar seis trackers que fueran desde una perspectiva más básica, como el algoritmo MS, a otros más sofisticados como el MEEM, que tratan de evitar algunos problemas del seguimiento de objetos. Para ello se llevó a cabo un análisis con las características y atributos más significativos de cada uno de los trackers seleccionados. Para finalizar con esta parte, se generó una comparativa con las ventajas e información más relevante de todos los algoritmos empleados (Tabla 3.1). También se trabajó un tiempo con algunos algoritmos enfocados para vídeos de larga duración, pero su inicialización y puesta en marcha no era del todo correcta en la mayoría de las secuencias por lo que se decidió tomar otras alternativas.

Tras haber estudiado a lo largo del proyecto el gran número de problemas que aparecen durante el seguimiento de objetos, así como el gran abanico de posibilidades de este campo en cuanto a la mejora y rendimiento de los trackers, se llevó a cabo un análisis de las distintas técnicas y mecanismos más recientes de combinación y fusión de trackers. Por ello, se decidió desarrollar un algoritmo basado en [55] que combinase las diferentes salidas, almacenadas como bounding box, de cada tracker para cada una de las secuencias empleadas. El objetivo era conseguir que el comportamiento de esa fusión de algoritmos obtuviese mejores resultados que la mayoría de los trackers empleados cuando actuaban de manera individual.

Finalmente, se evaluaron los trackers seleccionados con cada una de las secuencias escogidas, tanto de corto como de largo plazo. De esta forma resulta más fácil de analizar las diferencias de un mismo algoritmo en función de la duración del vídeo. Para ello se realizaron dos técnicas de evaluación diferentes. Por un lado, mediante el valor del Center Location Error (CLE) y por otra parte con el coeficiente del Sequence Frame Detection Accuracy (SFDA). Para cada una de la secuencias se concluyen cuáles son los dos trackers que presentan los mejores resultados, independientemente del mecanismo de evaluación. Además en esta parte se exponen de manera gráfica las conclusiones de la combinación de algoritmos desarrollada, en la que se refleja la mejoría de resultados en cada secuencia respecto a la mayoría de los resultados de los algoritmos obtenidos individualmente.

Después del análisis de los resultados y del estudio global del proyecto, se llega a la conclusión de que a pesar de que el marco de seguimiento de objetos en vídeos a largo plazo es muy extenso, el gran número de investigaciones recientes está provocando una clara mejoría en los resultados finales. Probablemente el enfoque de los nuevos algoritmos irá más orientado al largo plazo y existirán un mayor número de posibilidades con las que seguir al objeto de forma correcta a lo largo del tiempo.

6.2. Resumen del trabajo realizado

A continuación se muestra en una tabla, a modo de recapitulación, las distintas etapas o fases que se han ido realizando durante este proyecto. Cada etapa lleva consigo un modo de implementación y una breve descripción de la misma. Conviene definir previamente los modos de implementación que se emplearán en este resumen, que son implementado, adaptado y propio. Se define la implementación como la aplicación de métodos o técnicas para la puesta en funcionamiento de algo. La adaptación consiste en la realización de pequeños ajustes para su adecuación a las características del trabajo. Finalmente aquello catalogado como propio es la fase en la que las acciones realizadas no se han visto influenciadas por terceras partes.

Etapa	*Implementación*	*Descripción*	*Fuente*
Ejecución de trackers	Adaptada	Ajuste de las características de cada tracker según las necesidades de los dataset	[17, 19, 30, 40, 43, 48, 49]
Gestión bbox resultado	Propia	Unificación bajo el mismo formato y forma de los bbox resultados para sus usos posteriores.	[–]
Cálculo distancia/solape	Adaptada	Adaptación del mecanismo de solapamiento estudiado a los requisitos propuestos	[4, 55]
Cálculo valor de atracción	Implementada	Implementación en Matlab para poder poner en marcha el algoritmo propuesto	[55]
Mecanismo de optimización	Implementada	Implementación en Matlab para poder conseguir los mejores resultados posibles dentro las limitaciones de los algoritmos empleados	[55, 58]
Técnicas de búsqueda de optimización	Propia	Realización de varias estrategias con el fin de encontrar aquella que proporcionase los resultados más óptimos de todas las opciones	[–]

Tabla 6.1: Resumen por etapas del trabajo desarrollado.

6.3. Trabajo futuro

El escenario del seguimiento de objetos, y más en particular, el del estudio a largo plazo continúa siendo un problema abierto. Muchos son los desafíos que tienen que tratar los algoritmos de seguimiento en sus futuras líneas de investigación. Sin embargo, cada vez son más los estudios que ponen su foco de atención en este ámbito computacional.
Tras la realización de este proyecto aparecen varias líneas futuras de investigación, como pueden ser las siguientes:

- Ampliación de la información de la fase de inicialización. Muchos de los algoritmos seleccionados, obtienen resultados desfavorables tras varias decenas de frames. Sería deseable que en la etapa de inicio se tuvieran más posibilidades o se abordasen más tipos de características para obtener mejores estimaciones y modelos iniciales, para que no se produjesen malos resultados que pudieran afectar al resto de la secuencia.

- Desarrollo del uso de un *pívot* para el largo plazo. Muchos trackers emplean una referencia, o pívot, en su etapa de selección y procesamiento de características. De aquí se deriva este nuevo camino y su posterior estudio para adaptar esta técnica a lo largo del tiempo.

- Creación de una técnica de reinicio del algoritmo. Para evitar pérdidas del objeto constantes o malos resultados en largos periodos de tiempo, una supuesta solución podría ser diseñar un mecanismo de reinicio del tracker pasados una cierta cantidad de frames. De esta manera algoritmos que cada vez se alejaban más del objeto, podrían volver a tener oportunidades de reportar resultados aceptables al final de la secuencia.

- Ponderación de algoritmos y nuevas técnicas de fusión. Tras la realización del sistema de combinación de trackers, se ha llegado a la conclusión de que cuando cada tracker está en un punto alejado del objeto, sin solapamiento o a la misma distancia, el algoritmo mejoraría los resultados mediante una ponderación previa de los demás trackers. De esta forma, en situaciones difíciles, se reduciría el riesgo de obtener resultados negativos que afectasen a su rendimiento en el resto del vídeo. Además la línea de fusión de trackers continúa abierta, ya que las combinaciones y posibilidades crecen con cada nuevo tracker y sus correspondientes características.

- Estudio y desarrollado de trackers enfocados a largo plazo. Llevar a cabo un análisis más ampliado de esta reciente línea de trackers, como pueden ser los algoritmos Self Paced Long-Term Tracking (SPLTT) [2] o Long-Term Correlation Tracking (LT-CT) [36], para que puedan ser ejecutados en una mayor cantidad de vídeos de larga duración y comparados respecto a otros trackers sin este determinado enfoque.

Bibliografía

[1] E. Maggio and A. Cavallaro, Video Tracking: Theory and Practice. Hoboken, NJ: Wiley, 2011.

[2] James Steven Supančič III Deva Ramanan, "Self-paced learning for long-term tracking", in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[3] Yang Hua, Karteek Alahari, and Cordelia Schmid, "Occlusion and Motion Reasoning for Long-Term Tracking", in: *13th European Conference Computer Vision (ECCV)* , 2014.

[4] Samuele Salti, Andrea Cavallaro, Luigi Di Stefano, "Adaptive Appearance Modeling for Video Tracking: Survey and Evaluation", in: *IEEE Transactions on Image Processing (TIP)*, 2012.

[5] Quanfu Fan, Sharath Pankanti, Lisa Brown, "Long-term Object Tracking For Parked Vehicle Detection", in: *11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2014.

[6] Karel Lebeda, Simon Hadfield, Jiri Matas, Richard Bowden, "Long-Term Tracking Through Failure Cases", in: *IEEE Conference on Computer Vision Workshops (ICCVW)*, 2013.

[7] Yilmaz, A., Javed, O., and Shah, M. 2006. "Object tracking: A survey. ACM *Comput. Sur*v, 2006.

[8] B. D. Lucas and T. Kanade. " An Iterative Image Registration Technique with An Application to Stereo Vision", in: *7th Intl Joint Conf on Artificial Intelligence (IJCAI),*1981.

[9] I. Matthews, T. Ishikawa, and S. Baker. The Template Update Problem.

[10] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, "Object Tracking Benchmark".

[11] M. Prieto, M. Marufo, L. Di Matteo, R. Verrastro, A. Hernández, J. Gomez, C. Verrastro, "Algoritmo de seguimiento de objetos en imágenes mediante reconstrucción iterativa de histograma entiempo real".

[12] Alok K. Watve, "A Seminar On Object tracking in video scenes".
[13] Yigithan Dedeoglu, Moving Object Detection, Tracking and Classification for smart video surveillance, 2004.

[14] Mykhaylo Andriluka, Stefan Roth, Bernt Schiele, Monocular 3D Pose Estimation and Tracking by Detection.

[15], Marta Lucía Guevara, Julián David Echeberry, William Ardila Urueña, "Detección de rostros en imágenes digitales usando clasificadores en cascada, Scientia et Technica, Junio de 2008.

[16] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded Up Robust Features".

[17] Dorin Comaniciu, Visvanathan Ramesh, Peter Meer , " Kernel-Based Object Tracking", in: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[18] Stan Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms", in: *IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, June 1998.

[19] Robert Collins , CSE486, Penn State; Lecture 29: "Video Tracking: Mean-Shift".

[20] Xiaoyu Wang , Gang Hua, and Tony X. Han, "Discriminative Tracking by Metric Learning".

[21] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning".

[22] Abhishek Kumar Chauhan, Prashant Krishan, "Moving Object Tracking using Gaussian Mixture Model and Optical Flow", in: *International Journal of Advanced Research in Computer Science and Software Engineering*, 2013.

[23] Wei Zhong Dalian, Huchuan Lu Dalian, Ming-Hsuan Yang , "Robust Object Tracking via Sparsity-based Collaborative Model".

[24] Jiyan Pan and Bo Hu, "Robust Object Tracking against Template Drift".

[25] Katharina Quast, André Kaup, "AUTO GMM-SAMT: An Automatic Object Tracking System for Video Surveillance in Traffic Scenarios", in: *EURASIP Journal on Image and Video Processing,* 2010.

[26] Visual Tracker Benchmark ver 1.1 (submitted to PAMI),Test sequences.

[27] http://www.adcis.net/en/Applications/List-Of-Examples/Motion-Tracking.html

[28] Stephen C. Crampton and Margrit Betke, Finger Counter: “A Human-Computer Interface”.

[29] Vasant Manohar, Padmanabhan Soundararajan, Harish Raju, Dmitry Goldgof, Rangachar Kasturi, and John Garofolo, “Performance Evaluation of Object Detection and Tracking in Video”.

[30] Kaihua Zhang, Lei Zhang, Ming-Hsuan Yang and Qinghua Hu, “Robust Object Tracking via Active Feature Selection”, in: *IEEE transactions on circuits and systems for video technology, vol. 23, no. 11*, November 2013.

[31] Min Yang, Yuwei Wu, Mingtao Pei, Bo Ma and Yunde Jia, “Coupling Semi-supervised Learning and Example Selection for Online Object Tracking”, in: *Asian Conference on Computer Vision (ACCV)*, 2014.

[32] Vasilis Papadourakisa, Antonis Argyrosa, “Multiple objects tracking in the presence of long-term occlusions”.

[33] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, “Tracking-Learning-Detection”, in: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 6, NO. 1,* JANUARY 2010.

[34]http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm. html

[35] Sam Hare, Amir Saffari, Stuart Golodetz, Vibhav Vineet, Ming-Ming Cheng, Stephen L. Hicks and Philip H. S. Torr, “Struck: Structured Output Tracking with Kernels”, in: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 2014.

[36] Chao Ma, Xiaokang Yang, Chongyang Zhang and Ming-Hsuan Yang, “Long-term CorrelationTracking”.

[37] Karel Lebeda, Simon Hadfield, Jirı Matas and Richard Bowden, “Texture-Independent Long-Term Tracking Using Virtual Corners”.

[38] http://cvisioncentral.com/wp-content/uploads/formidable/LTDT2014-CFP-last.pdf

[39] http://cmp.felk.cvut.cz/~vojirtom/dataset/

[40] Jifeng Ning, Lei Zhang , David Zhang and Chengke Wu, “Robust meanshift tracking with corrected background weighted histogram”.

[41] Yizong Cheng, “Mean Shift, Mode Seeking, and Clustering”, in: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 17, NO, 8,* AUGUST 1995.

[42] http://docs.opencv.org/3.1.0/db/df8/tutorial_py_meanshift.html#gsc.tab=0

[43] Martin Danelljan , Fahad Shahbaz Khan, Michael Felsberg , Joost van de Weijer, “Adaptive Color Attributes for Real-Time Visual Tracking”.

[44] Joao F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista, “Exploiting the Circulant Structure of Tracking-by-detection with Kernels”.

[45]http://docs.opencv.org/3.1.0/d1/dc5/tutorial_background_subtraction.html#gsc.tab=0

[46] Burr Settles, “Active Learning Literature Survey”, 2010.

[47] Mikhail Belkin, Partha Niyogi and Vikas Sindhwani, “Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples”, in: *Journal of Machine Learning Research 7 (2006) 2399-2434*.

[48] Min Yang, Yuwei Wu, Mingtao Pei, Bo Ma and Yunde Jia, “Online Discriminative Tracking with Active Example Selection”.

[49] Jianming Zhang, Shugao Ma and Stan Sclaroff, MEEM: “Robust Tracking via Multiple Experts Using Entropy Minimization”.

[50] http://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/svms/RBFKernel.pdf

[51]http://coltech.vnu.edu.vn/~thainp/books/Wiley_-_2006_-_Elements_of_Information_Theory_2nd_Ed.pdf

[52] https://hashtagyoco.wordpress.com/

[53] Yves Grandvalet and Yoshua Bengio, “Semi-supervised Learning by Entropy Minimization”.

[54] Ms. G.Anusha and Mrs. E. Golden Julie, “Improving the Performance of Video Tracking Using SVM”, in: *International Journal of Engineering Trends and Technology (IJETT) – Volume 11 Number 3,* May 2014.

[55] Christian Bailer, Alain Pagani and Didier Stricker, “A Superior Tracking Approach: Building a strong Tracker through Fusion”.

[56] Rafael Martín Nieto, “ON THE FUSION OF SINGLE-TARGET VIDEO OBJECTS TRACKING ALGORITHMS”, Trabajo Fin de Máster.

[57] Jakob Santner, Christian Leistner, Amir Saffari, Thomas Pock and Horst Bischof, “PROST: Parallel Robust Online Simple Tracking”.

[58] Christian Bailer, Alain Pagani and Didier Stricker, “A user supported tracking framework for interactive video production”.

[59] Junseok Kwon and Kyoung Mu Lee, “Tracking by Sampling Trackers”.

[60] Michael Hodlmoser , Branislav Micusik, Marc Pollefeys, Ming-Yu Liu and Martin Kampel, “Model-Based Vehicle Pose Estimation and Tracking in Videos Using Random Forests”, in: 2013 *International Conference on 3D Vision - 3DV*, 2013.
[61] Tahir Nawaz, Fabio Poiesi, Andrea Cavallaro, “Assessing Tracking Assessment Measures”.

[62] http://es.slideshare.net/jcbp_peru/el-histograma-una-imagen-digital

Apéndice A

Resultados del algoritmo de fusión para secuencias de corto plazo

En este apéndice se muestran algunos ejemplos gráficos de los resultados del algoritmo de fusión desarrollado en este proyecto para varias secuencias de corto plazo. Como sucedía con los algoritmos estudiados de manera individual, el algoritmo de fusión actúa mejor en vídeos de menor duración.

Secuencias de corto plazo

- *Secuencia: Skiing (81 frames)*

Figura A.1: Representación del bbox resultado (color negro) de la combinación de trackers durante los frames 1, 16, 22, 41 y 81 de la secuencia Skiing.

Lo más representativo de este vídeo es que como la mayoría de los algoritmos pierden el objeto en el frame 16 (sólo dos ellos lo siguen correctamente) el solapamiento es mayor entre los otros trackers, y fruto de ello el rectángulo negro no consigue un buen resultado en este frame. Sin embargo, como se puede observar en el frame 22, mientras que los algoritmos que habían perdido al objeto siguen sin mejorar sus resultados, el tracker resultante de la fusión consigue recuperar al objeto ya que el solapamiento es mayor entre los algoritmos que detectan al objeto correctamente. Sucede lo mismo en el frame 41.

- *Secuencia: Coke (291 frames)*

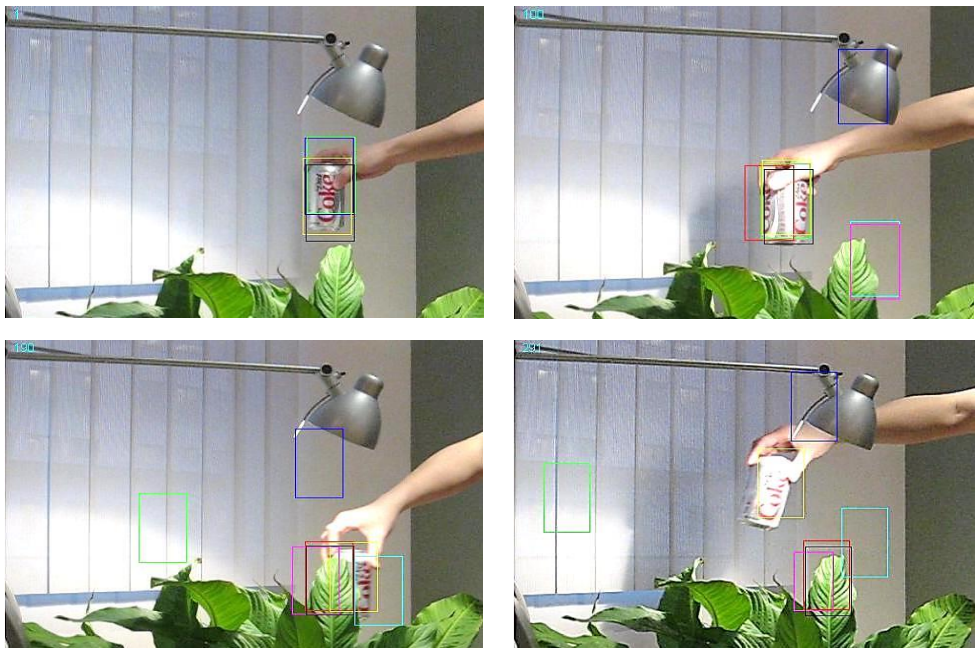


Figura A.2: Representación del bbox resultado (color negro) de la combinación de trackers durante los frames 1, 100, 190 y 291 de la secuencia Coke.

En el caso del vídeo Coke, lo más característico sucede al final del mismo. Se observa que durante gran parte del tiempo el resultado del algoritmo propuesto proporciona buenos resultados, también durante la oclusión parcial (frame 190), pero que sin embargo tras la oclusión total del final de la secuencia no es capaz de terminar con el objeto de interés ya que la mayoría de los algoritmos solapan donde se ha producido ese problema y no son capaces de recuperarlo (excepto el tracker MEEM, rectángulo de color amarillo).

- *Secuencia: Bolt (350 frames)*

Figura A.3: Representación del bbox resultado (color negro) de la combinación de trackers durante los frames 1, 106, 220 y 350 de la secuencia Bolt.

En esta secuencia destaca el buen comportamiento conseguido por el algoritmo desarrollado como fusión de los trackers desde el inicio y hasta el fin de la misma. En los frames intermedios (frames 106 y 220) se puede apreciar como algunos algoritmos empiezan a empeorar sus resultados (rectángulos verde, azul y cyan) mientras que el bbox de color negro sigue correctamente con el objeto.

Apéndice B

Presupuesto

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 2.200 €
- Material de oficina..... 150 €
- Total de ejecución material..... 2.350 €

2) Gastos generales

- 16 % sobre Ejecución Material..... 376 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material..... 141 €

4) Honorarios Proyecto

- 1200 horas a 15 € / hora..... 18000 €

5) Material fungible

- Gastos de impresión..... 50 €
- Encuadernación..... 200 €

6) Subtotal del presupuesto

- Subtotal Presupuesto..... 21117 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto ..... 4434.5 €

8) Total presupuesto

- Total Presupuesto..... 25551,5 €

Madrid, Julio de 2016

El Ingeniero Jefe de Proyecto

Fdo.: Borja Maza Vargas
Ingeniero de Telecomunicación

Apéndice C

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un *estudio del seguimiento de objetos en vídeo a largo plazo*. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

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

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

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

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la

contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa otro concepto.

Condiciones particulares

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

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.