

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

TRABAJO FIN DE GRADO

Seguimiento de objetos basado en múltiples
algoritmos.

Eduardo Moreno de Pablos
Tutor: Juan Carlos San Miguel Avedillo
Ponente: José María Martínez Sánchez

Julio 2016

Seguimiento de objetos basado en múltiples algoritmos

AUTOR: Eduardo Moreno de Pablos

TUTOR: Juan Carlos San Miguel Avedillo



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

Resumen

El objetivo principal de este trabajo consiste en analizar el seguimiento de un objeto en una secuencia de vídeo mediante la utilización de un conjunto de algoritmos de búsqueda o seguimiento.

En primer lugar, se lleva a cabo un estudio inicial del seguimiento de objetos, haciendo un énfasis en los aspectos básicos de detección, búsqueda y seguimiento, así como en la combinación de los diversos algoritmos de búsqueda. Posteriormente, se procede a estudiar un conjunto de trabajos científicos (papers), que aunque se centren en el mismo tema, utilizan diferentes técnicas para lograr el objetivo deseado.

Después de esta etapa de aprendizaje teórico, se empieza a utilizar la herramienta Matlab con un conjunto de algoritmos y secuencias seleccionados, comparando la eficacia de los algoritmos mediante el error basado en anotaciones manuales (*ground-truth*), además de compararlos entre sí con una serie de medidas que estiman la fiabilidad de los algoritmos, o comparar un algoritmo consigo mismo en instantes de tiempo diferentes. Entre dichas medidas cabe destacar la distancia entre centros y el solapamiento entre áreas detectadas en la frame correspondiente. Dichas medidas nos otorgan una capacidad de visualizar que algoritmos son óptimos en una posible combinación de estos, y cuales no. Por último, para introducir una mayor firmeza en la posible selección de los algoritmos más óptimos de cara a una combinación de estos, se utilizan mapas de confianza. Estos mapas son áreas de un tamaño similar a las frames, las cuales muestran la probabilidad de que el objeto esté en cada punto de ese mapa.

Los resultados experimentales muestran las virtudes y defectos de cada algoritmo sobre el conjunto de secuencias seleccionadas utilizando las medidas anteriormente descritas. Una vez obtenidos los resultados provenientes de las medidas descritas anteriormente, se procede a realizar una fusión de algoritmos combinando aquellos con mayores puntuaciones.

Palabras clave

procesamiento de vídeo, seguimiento de objetos, fiabilidad en tiempo de ejecución, solapamiento espacial, distancia inter-centroide, mapa de confianza.

Abstract

The main objective of this thesis consists on tracking objects in video sequences using a set of search or tracking algorithms.

First, an initial study of existing works is carried out with emphasis on the basic aspects of detection, search, tracking and the combination of various search algorithms. Then, a set of scientific papers that are focused on the same topic have been studied. These papers use different techniques to achieve the desired objective.

After this theoretical learning stage, the Matlab tool is used to test a set of selected algorithms and sequences, comparing the effectiveness of the algorithms by the error based on ground-truth also to compare them with a series of measures, or comparing an algorithm himself in different time instants to estimate their reliability. These measures employ the distance between centers and the overlap between areas detected in the corresponding frame. These measures give us an ability to visualize which algorithms are optimal in a possible combination of these, and which are not. Finally, confidence maps are used to estimate results' stability in the possible selection of the optimal algorithms for their combination. These maps are extracted from each algorithm and represent areas of a size similar to the frames which show the probability that the object is located at each point of the map.

Experimental results show the strengths and weaknesses of each algorithm on the set of selected sequences using the proposed set of reliability measures. Once the results obtained from the measures described above, we proceed to combine the best algorithms as proposed.

Keywords

video processing, object tracking, reliability estimation, spatial overlap, inter-center distance, confidence map.

Agradecimientos

Quiero agradecer a Juan Carlos San Miguel por su labor como tutor, ya que gracias a él he aprendido una serie de conceptos de gran interés sobre el trabajo realizado, además de adquirir numerosos conocimientos respecto a diversas herramientas y programas desconocidos para mi hasta este momento. Además he de decir que se ha preocupado constantemente por mi aprendizaje y me ha ayudado a mejorar en gran medida la eficacia en la forma de organizarme.

También he de agradecer al profesorado de la carrera, el cual se ha esmerado y preocupado por mi formación en los diversos ámbitos de esta, así como el aprendizaje de mis compañeros.

Por último y no menos importante, mis agradecimientos a mis compañeros de clase, ya que sin ellos la carrera no hubiese sido lo mismo, y gracias a mi familia, en especial a mis padres, por el apoyo incondicional durante estos 4 años.

Eduardo Moreno de Pablos

Julio 2016

Índice general

Resumen	v
Abstract	vi
Agradecimientos	vii
1. Introducción.	1
1.1. Motivación.	1
1.2. Objetivos.	1
1.3. Estructura de la memoria.	3
2. Estado del arte.	5
2.1. Introducción	5
2.2. Seguimiento de objetos	5
2.2.1. Etapas	5
2.2.2. Problemas	7
2.2.3. Métodos de seguimiento de objetos	8
2.3. Selección de trabajos relacionados	10
2.3.1. Symbiotic Tracker Ensemble Toward A Unified Tracking Framework	10
2.3.2. A Superior Tracking Approach: Building a strong Tracker through Fusion	12
2.3.3. Tracker-Level Fusion for Robust Bayesian Visual Tracking	15
2.3.4. Randomized Ensemble Tracking	15
2.4. Conclusiones	17
3. Desarrollo	19
3.1. Esquema global	19
3.2. Algoritmos	19
3.2.1. Mean-shift Tracking	20
3.2.2. Robust Mean Shift Tracking with Corrected Background-Weighted Histogram	20
3.2.3. Particle Filter Color Tracker	21
3.2.4. Adaptive Color Attributes for Real-Time Visual Tracking	21
3.3. Métricas	21

3.3.1.	Solapamiento entre bounding boxes	22
3.3.2.	Distancia entre centros	23
3.3.3.	Score Likelihood	24
3.4.	Fusión	26
4.	Resultados experimentales	27
4.1.	Datasets	27
4.2.	Métricas de evaluación	29
4.3.	Resultados globales	29
4.4.	Resultados específicos	30
4.4.1.	Inter-tracker	30
4.4.2.	Intra-tracker	33
4.4.3.	Score Likelihood	35
4.4.4.	Fusión	36
5.	Conclusiones y trabajo futuro.	41
5.1.	Conclusiones.	41
5.2.	Trabajo futuro.	42
	Bibliografía	43

Índice de figuras

1.1. Ejemplos de aplicaciones de seguimiento de objetos.	2
2.1. Etapas principales del seguimiento de objetos.	6
2.2. Diferentes representaciones del objetivo.	6
2.3. Único algoritmo frente a varios problemas.	9
2.4. Varios algoritmos frente a varios problemas.	9
2.5. Arquitectura de fusión en paralelo.	10
2.6. Arquitectura de fusión secuencial.	11
2.7. Symbiotic	12
2.8. Diagrama de bloques algoritmo Superior Tracking Approach.	14
2.9. Distribución del coeficiente de incertidumbre.	16
2.10. Corrección del estado.	16
2.11. Randomized Ensemble Tracker.	17
3.1. Esquema global del desarrollo.	20
3.2. Funcionamiento del algoritmo Mean-shift.	21
3.3. Funcionamiento algoritmo Particle Filter Color Tracker.	22
3.4. Solapamiento entre bounding boxes.	23
3.5. Distancia entre centros.	24
3.6. Mapa de confianza.	25
3.7. Score Likelihood.	25
4.1. Inter-tracker secuencia I1_basic.	31
4.2. Inter-tracker secuencia I3_cars_basic_2.	32
4.3. Inter-tracker secuencia Singer1.	32
4.4. Intra-tracker secuencia I1_basic.	33
4.5. Intra-tracker secuencia I3_cars_basic_2.	34
4.6. Intra-tracker secuencia Singer1.	34
4.7. Score Likelihood secuencia I3_cars	35
4.8. Score Likelihood secuencia FaceOcc1.	36
4.9. Score Likelihood secuencia Rolling.	36
4.10. Comparación entre los algoritmos y la fusión secuencia I3_cars_basic_2.	37
4.11. Comparación entre los algoritmos y la fusión secuencia FaceOcc1.	38
4.12. Comparación entre los algoritmos y la fusión secuencia Bolt.	38
4.13. Comparación entre la fusión equitativa y la fusión ponderada secuencia FaceOcc1.	39

4.14. Comparación entre la fusión equitativa y la fusión ponderada secuencia	
Singer.	39

Índice de tablas

4.1. Desafíos que presentan los datasets.	27
4.2. Datasets utilizados.	28
4.3. Resultados promedios Distancia entre centros inter-tracker.	29
4.4. Resultados promedios Solapamiento inter-tracker.	29
4.5. Resultados promedios Distancia entre centros intra-tracker.	29
4.6. Resultados promedios Solapamiento intra-tracker.	30
4.7. Resultados promedios Score likelihood intra-tracker.	30
4.8. Resultados promedios Groundtruth error.	30

Capítulo 1

Introducción.

1.1. Motivación.

El seguimiento de objetos, como su nombre indica, consiste en estimar y localizar un objeto en una secuencia de vídeo. Para poder llevar esta tarea a cabo, se utiliza un algoritmo de búsqueda en concreto, o un conjunto de algoritmos, cada uno especializado en una posible adversidad en dicha secuencia, combinando la parte óptima de cada uno de ellos para poder detectar y seguir al objeto deseado en cada frame del vídeo. Por otra parte, este tipo de tarea alberga una gran complejidad debido al conjunto de técnicas que hay que utilizar para llevar a cabo el seguimiento, además de ser un proceso lento por la gran cantidad de información que incluye un vídeo. De cara a optimizar el seguimiento, existen diferentes técnicas. Una de ellas es la fusión de algoritmos, utilizando y combinando las características de los algoritmos con mejores resultados.

En este campo se ha estado trabajando en gran medida en los últimos años, y actualmente se encuentra en ámbitos como la vídeo-vigilancia (Figura 1.1), donde no hará falta una persona a cargo de observar que ocurre en la señal de vídeo capturada en tiempo real. Otros ejemplos serían las aplicaciones médicas, en las cuales será de interés la búsqueda y detección de los tejidos en el cuerpo humano, así como la localización y el seguimiento de una aguja en una operación.

1.2. Objetivos.

El objetivo de este proyecto consiste en analizar e investigar el comportamiento de los diversos algoritmos de búsqueda y seguimiento que se utilizan en diferentes secuencias de vídeo. Para llevar a cabo este proceso, se definen una serie de medi-



Figura 1.1: Ejemplos de aplicaciones de seguimiento de objetos. ([1][2])

das para poder comparar los algoritmos entre sí, y poder apreciar qué algoritmos son más interesantes a utilizar en una hipotética fusión de algoritmos a lo largo de una determinada secuencia, ya que cada algoritmo está especializado en un conjunto de adversidades en concreto. Con dicha fusión, complementaríamos entre sí los algoritmos con mejores resultados.

Por tanto, el objetivo principal se desglosa en varios objetivos generales:

1. Estudio y análisis teórico del estado del arte

Para poder empezar con este proyecto, es necesario documentarse acerca del seguimiento de objetos y sus partes fundamentales, donde cabe destacar qué tipo de algoritmos se utilizan, en que se basan dichos algoritmos, cómo consiguen detectar el objeto en una secuencia en concreto y como combinar un conjunto de algoritmos de búsqueda entre sí.

2. Ampliación de los conceptos teóricos

Tras un estudio del estado del arte de manera generalizada, se empieza a obtener información más detallada de un conjunto de papers sobre cómo llevar a cabo el seguimiento de los objetos, aprendiendo a diferenciar las técnicas utilizadas por dichos papers.

3. Propuesta e implementación

Al finalizar la parte de aprendizaje teórico de este proyecto, se empieza a utilizar código, observando los diferentes algoritmos a utilizar ya implementados y

comenzando a obtener datos a partir de estos a la vez que se sacan las primeras conclusiones. Posteriormente se propone un algoritmo basado en combinación.

4. Evaluación de las técnicas utilizadas

Una vez adentrado en el código, se comienza a utilizar más técnicas de diferente implementación para dar mayor soporte a las conclusiones obtenidas con anterioridad, así como observar qué algoritmos óptimos para una secuencia de vídeo en concreto, y cuales no. Por último, se extraen las últimas conclusiones sobre la utilización de varios algoritmos en seguimiento de objetos.

1.3. Estructura de la memoria.

La memoria de este proyecto se divide en los siguientes capítulos:

- Capítulo 1. Introducción: Motivación y objetivos del proyecto.
- Capítulo 2. Estado del arte : Qué es el seguimiento de objetos de manera más específica y comentarios sobre los papers estudiados.
- Capítulo 3. Desarrollo: Algoritmos y métricas a utilizar.
- Capítulo 4. Resultados experimentales: Datasets utilizados y resultados obtenidos de las métricas de la sección anterior.
- Capítulo 5. Conclusiones y trabajo futuro.
- Referencias y anexos.

Capítulo 2

Estado del arte.

2.1. Introducción

En esta sección se explica y se definen los fundamentos básicos del seguimiento de objetos, las etapas principales por las que está compuesto, los problemas principales que pueden aparecer al llevar a cabo esta tarea y diferentes métodos para acometer el seguimiento. Por otra parte, también se comentan y se ilustran aspectos básicos de la fusión de un conjunto de algoritmos, y cómo, mediante dicha fusión, se puede optimizar la robustez en el seguimiento de objetos. Además, se definen los diferentes tipos de arquitecturas de fusión.

Por último, se muestra las ideas y conceptos de diferentes artículos sobre como llevar a cabo el seguimiento de objetos de manera óptima.

2.2. Seguimiento de objetos

2.2.1. Etapas

El seguimiento de objetos es un campo de investigación de gran importancia, ya que tiene utilidad en diversos ámbitos. Consiste en detectar, reconocer y seguir un determinado objeto en una secuencia de imágenes. Habrá un escenario específico y el objetivo es no perder de vista el objeto a seguir. Las cámaras de vídeo capturan información sobre los objetos en forma de un conjunto de píxeles. El seguidor de objetos relacionara el aspecto del objeto de interés y el valor de los píxeles correspondientes.

El seguimiento de objetos se puede descomponer en cinco etapas principales (Figura 2.1):

- **Extracción de características:** En esta fase se define el método para extraer la información relevante del área de la imagen donde se encuentra el objeto.

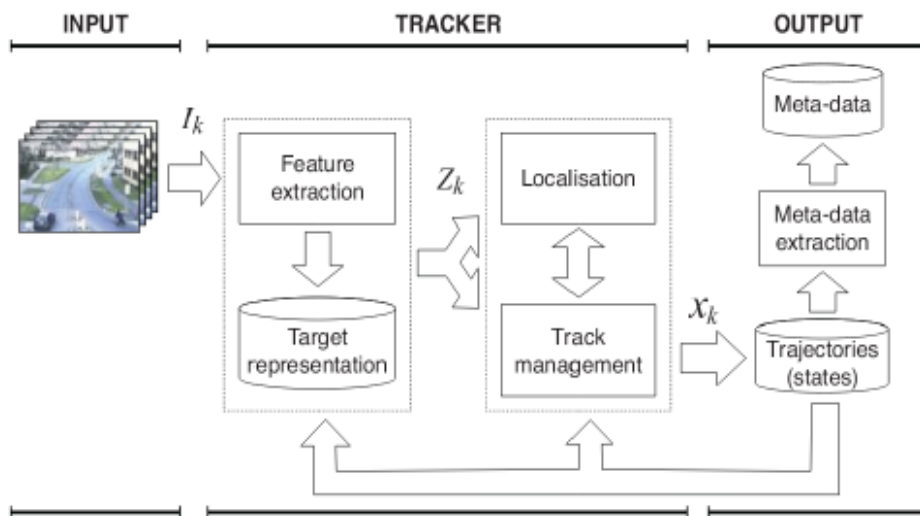


Figura 2.1: Etapas principales del seguimiento de objetos. [[3]]

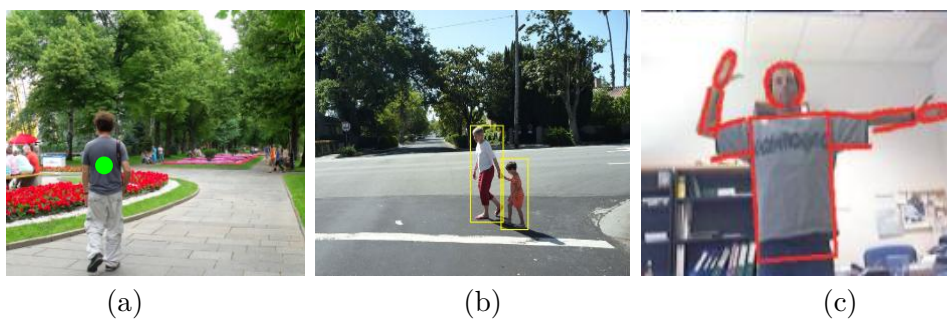


Figura 2.2: Diferentes representaciones del objetivo.

La extracción de características se puede dividir en 3 clases principales: Nivel bajo (e.g., color, gradiente), Nivel medio (e.g., bordes, esquinas) y Nivel alto (el objeto).

- Representación del objeto:** El objeto a seguir se puede representar mediante sus formas y apariencias. Generalmente se utilizan las siguientes representaciones de forma del objeto:
 1. Punto: El objeto está representado por un centroide, o por un conjunto de puntos (Figura 2.2 (a)). Se suele utilizar cuando el objeto a seguir ocupa un espacio pequeño en la imagen.
 2. Formas geométricas: el objeto se representa mediante un rectángulo, una elipse,

u otros tipos de formas geométricas primitivas (Figura 2.2 (b)). Son útiles para objetos rígidos simples.

3. Modelos articulados: aproxima la forma del objeto combinando una serie de rígidos modelos, como pueden ser puntos o áreas, para crear conexiones topológicas (Figura 2.2 (c)). Se puede utilizar para modelar objetos articulados y rígidos.
4. Modelos deformables: puede que el objeto a seguir incluya una serie de deformaciones que no se puedan modelar mediante una forma canónica. Este determinado problema se puede solucionar buscando regiones homogéneas en el objeto que se puedan representar, y a continuación seguir esas regiones. Por otra parte y de otra manera, se puede definir el contorno del objeto mediante una serie de puntos.
 - **Localización:** Se debe localizar el objeto a lo largo del tiempo, a partir de una posición inicial.
 - **Gestión del seguimiento:** Se define una estrategia en base a las apariciones o desapariciones del objeto en el plano.
 - **Extracción de meta-datos:** Se extraen y se almacenan anotaciones del vídeo y toda la información necesaria para el seguimiento.

2.2.2. Problemas

En ocasiones, no va a ser tan fácil llevar a cabo el seguimiento, ya que la relación entre el objeto y la proyección de su imagen es muy compleja y puede depender de diversos factores además de la posición del objeto. Por lo tanto, en este proceso en general pueden ocurrir una serie de problemas. Entre dichos problemas cabe destacar los cambios de iluminación, en los que cambiará el color del objeto a seguir y dificultará su seguimiento. Otros problemas pueden ser las deformaciones de este objeto, ya que aunque cambie el color de este, si mantuviese su forma, podríamos solventar ese problema. Más ejemplos de problemas son las rotaciones, traslaciones, ruido u oclusiones, entre muchos otros. En las oclusiones perdemos de vista el objeto durante un determinado tiempo. Pueden ser totales, donde el objeto de interés es totalmente tapado por otro objeto, o parciales, donde se llegan a ver partes del objeto a seguir en la escena. El ruido introduce una cierta distorsión en la imagen captada por el sensor de la cámara, afectando así al rendimiento del seguidor. Por suerte, la mayoría de problemas que podamos tener se pueden solventar.

2.2.3. Métodos de seguimiento de objetos

Hay dos métodos diferentes para llevar a cabo el seguimiento de objetos. El primer método, y menos eficaz, se *basa en desarrollar un único algoritmo*, lo más complejo que se desee, dependiendo del objetivo final. El aspecto negativo de este primer método consiste en que por muy complejo que sea el algoritmo, solo estará entrenado para unos tipos de problemas concretos, como los mencionados en la sección anterior. Por lo tanto, si aparecen otros tipos de problemas, el algoritmo no podrá solventarlos de la manera adecuada.

El segundo método es mucho más eficaz que el primero, ya que soluciona los defectos de este. *Consiste en el desarrollo de varios algoritmos*, cada uno entrenado en un determinado problema de seguimiento. Después se procede a la fusión del conjunto de algoritmos entrenados. De esta manera, se tienen cubiertos la mayoría de los posibles problemas a la hora de seguir el objeto, con unos algoritmos mucho más sencillos, pero relacionado entre sí.

Como se ha descrito anteriormente, en este trabajo de fin de grado, se dispondrá de un conjunto diverso de algoritmos, los cuales intentarán detectar únicamente un objeto en la secuencia de imágenes. Cada uno de estos algoritmos funciona relativamente bien en situaciones concretas, en las cuales han sido entrenados, como por ejemplo los problemas citados anteriormente [2.2.2], es decir, cambios de iluminación, oclusiones, rotaciones, cambios de escala, deformaciones, entre muchos otros. Como cada algoritmo está entrenado en una determinada situación, cuando se agrupan varios fenómenos como los descritos anteriormente, si disponemos únicamente de un solo algoritmo, tendremos dificultades ya que podrá solucionar el problema para el que ha sido entrenado, pero no los demás problemas (Figura 2.3).

Por otra parte, el uso de múltiples algoritmos mejora el rendimiento y la robustez de una aplicación de seguimiento de objetos, ya que aunque cada algoritmo esté entrenado en un problema específico, al utilizar varios algoritmos, y combinar sus propiedades, obtenemos una mayor solución a varios problemas que puedan aparecer, y por lo tanto, más eficacia a la hora del seguimiento del objeto. (Figura 2.4).

De forma generalizada, se pueden considerar dos tipos de arquitecturas de fusión. El primer tipo se denomina *arquitectura en paralelo* (Figura 2.5). En este tipo de arquitectura, cada algoritmo de seguimiento se ejecuta independientemente al resto,

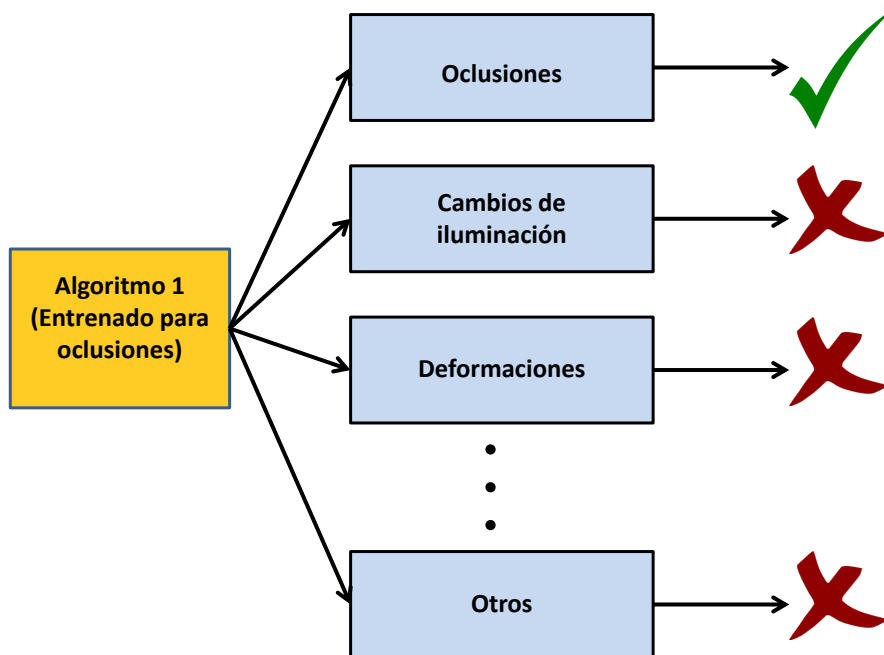


Figura 2.3: Único algoritmo frente a varios problemas.

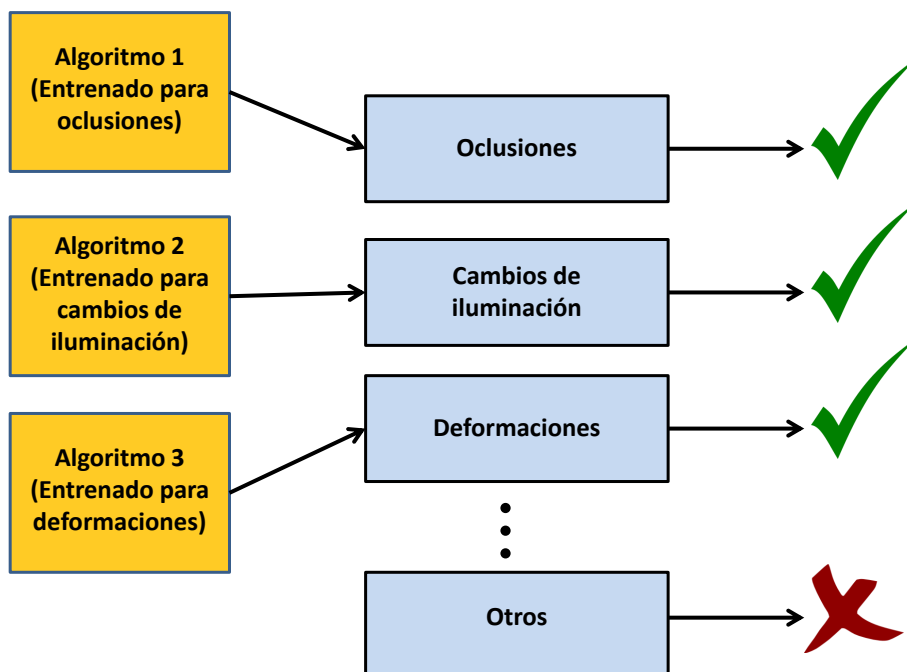


Figura 2.4: Varios algoritmos frente a varios problemas.

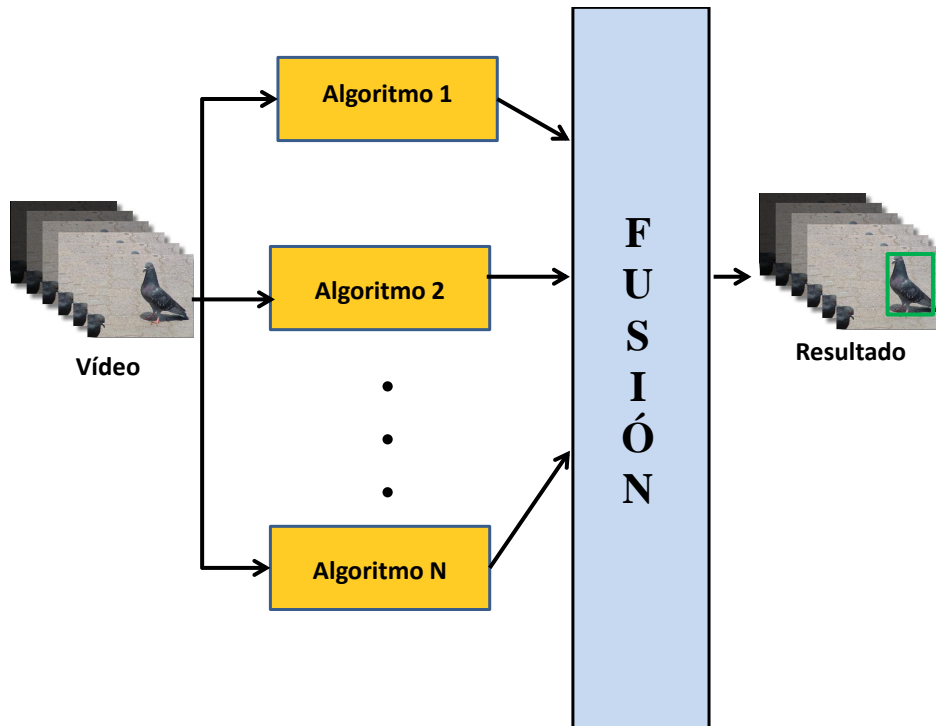


Figura 2.5: Arquitectura de fusión en paralelo.

y el resultado de la fusión es la mejor combinación posible entre los diversos algoritmos ejecutados.

El otro tipo de arquitectura a considerar se denomina *arquitectura secuencial o en cascada* (Figura 2.6). En esta arquitectura se aplica un método de refinamiento progresivo, es decir, se adapta el resultado a medida que se ejecutan los algoritmos de manera secuencial con las mejores estimaciones de cada uno de ellos.

2.3. Selección de trabajos relacionados

2.3.1. Symbiotic Tracker Ensemble Toward A Unified Tracking Framework

La mayoría de los métodos de fusión se centran en el análisis a nivel de pixel, lo cual genera un alto coste computacional y su potencial decae cuando se utilizan

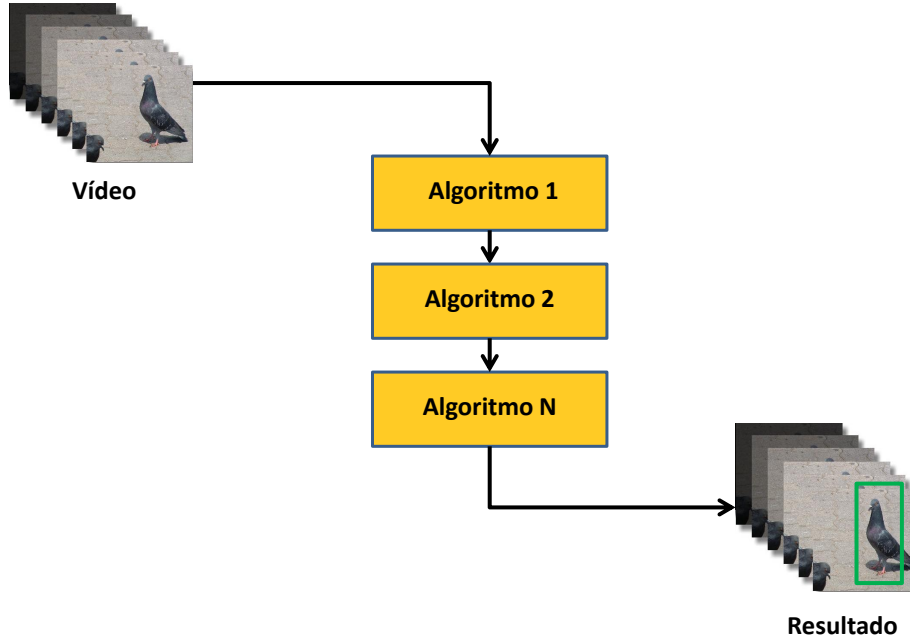


Figura 2.6: Arquitectura de fusión secuencial.

aplicaciones en tiempo real.

En este artículo [[4]] se genera un método de fusión común independiente de los trackers individuales, es decir, para la fusión se va a tratar únicamente las salidas de los trackers, no el mecanismo de estos. Además, todos los trackers se ejecutan en paralelo y el objetivo es encontrar la mejor combinación para la salida de estos.

Principalmente, en este paper se definen dos aspectos a considerar en el seguimiento de objetos (Figura 2.7). Uno de estos aspectos se denomina *consistency*, el cual se define como la variación de la eficacia del bounding box a la hora de seguir al objetivo, entre un frame en concreto, y su frame anterior. El otro aspecto se denomina *pair-wise tracker correlation*, y se estima mediante un proceso de propagación gráfico. El resultado final se genera gracias a una mezcla de modelo Gaussiano.

Para estimar el parámetro *consistency*, se debe comparar el resultado $R_{k,n-1}$ y $R_{k,n}$ mediante el método de correlación temporal para comprobar si el seguimiento se está haciendo correctamente, para un tracker k determinado, y para un frame n . Es decir, queremos comprobar la credibilidad de ese tracker.

La siguiente ecuación comprueba dicha credibilidad:

$$C_{k,n}^0 = \xi_k C_k + (1 - \xi_k) \Theta(R_{k,n-1}, R_{k,n}) C_{k,n-1} \quad [[4]]$$

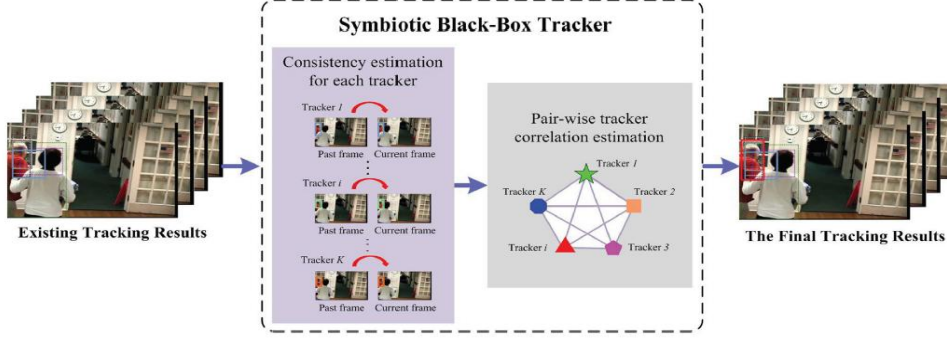


Figura 2.7: Esquema del algoritmo propuesto [[4]].

Siendo ξ_K el peso del tracker k , C_k el coeficiente de confianza del tracker k y $\theta(R_{k,n-1}, R_{k,n})$ el coeficiente de consistencia desde el frame $n-1$ hasta el frame n .

Para estimar el parámetro *pair-wise tracker correlation*, Se obtiene la correlación entre los resultados de los diferentes trackers, partiendo de las múltiples salidas de estos con el objetivo de obtener una salida conjunta óptima.

Se obtiene con la siguiente ecuación [[4]]:

$$\begin{cases} C_{1,n}^s = \eta_1 C_{1,n}^0 + \frac{(1-\eta_1)}{k-1} \sum_{i \neq 1} \Phi(R_{1,n}, R_{i,n}) C_{i,n}^{s-1} \\ C_{2,n}^s = \eta_2 C_{2,n}^0 + \frac{(1-\eta_2)}{k-1} \sum_{i \neq 2} \Phi(R_{2,n}, R_{i,n}) C_{i,n}^{s-1} \\ \vdots \\ C_{k,n}^s = \eta_k C_{k,n}^0 + \frac{(1-\eta_k)}{k-1} \sum_{i \neq k} \Phi(R_{k,n}, R_{i,n}) C_{i,n}^{s-1} \end{cases} \quad (2.1)$$

Siendo η_k un parámetro que determina el peso del coeficiente de credibilidad inicial. Los coeficientes nombrados anteriormente, θ y Φ , no son más que comparaciones entre las salidas de los resultados de los trackers. Para definir dichos coeficientes, se utilizan métricas como por ejemplo distancia entre centros, o solapamiento entre bounding boxes. Estas métricas se explicarán más adelante en el capítulo 3.

2.3.2. A Superior Tracking Approach: Building a strong Tracker through Fusion

El método propuesto por este artículo [[5]] se basa en localizar los bounding boxes producidos por los trackers utilizados, seleccionando aquellos que superen un umbral

previamente establecido. Después se realizará un promedio de los bounding boxes seleccionados.

Cómo el umbral depende en gran medida de cada secuencia, este paper se enfoca en campos de atracción, los cuales no necesitan umbral. En primer lugar, se obtienen las distancias entre un bounding boxes obtenidos, en las 4 dimensiones (x, y, ancho, alto). Dicha distancia se nombra como $d(b,c)$, siendo b y c dos bounding boxes.

En segundo lugar, se calcula el *attraction function* para un bounding box candidato (c) en el frame i .

$$a_i(c) = \sum_{j \in M} \frac{1}{d(b_{i,j},c)^2 + \sigma} \quad (2.2)$$

Siendo σ una constante, que además de evitar atracciones infinitas para distancias cero, es útil para la reducción de ruido. $b_{i,j}$ es el bounding box del tracker j en la frame i .

La función se atracción nombrada anteriormente, se modifica añadiendo los pesos de los trackers (w_j), ya que cada tracker rinde de manera diferente en cada secuencia.

$$a_i^w(c) = \sum_{j \in M} \frac{w_j^2}{d(b_{i,j},c)^2 + \sigma} \quad (2.3)$$

La nueva función se denomina *weighted attraction function*, la cual se puede normalizar de la siguiente manera:

$$\overline{a_i^w}(c) = \frac{a_i^w(c)}{\max_{b_{i,j} \in R_i} a_i^w(b_{i,j})}, \quad R_i := \{b_{i,1} \dots b_{i,M}\} \quad (2.4)$$

Por otra parte, se define la función de energía E_T . Se utilizar para ignorar posibles discontinuidades en la trayectoria de seguimiento. Como la trayectoria real no tiene discontinuidades, es deseable evitarla y encontrar una trayectoria continua.

Además, también se define la función p , diseñada para penalizar las desviaciones en la trayectoria continua que se está diseñando.

Para ir eliminando los trackers con los resultados menos favorables a medida que avanzan los frames, se utilizan los siguientes métodos:

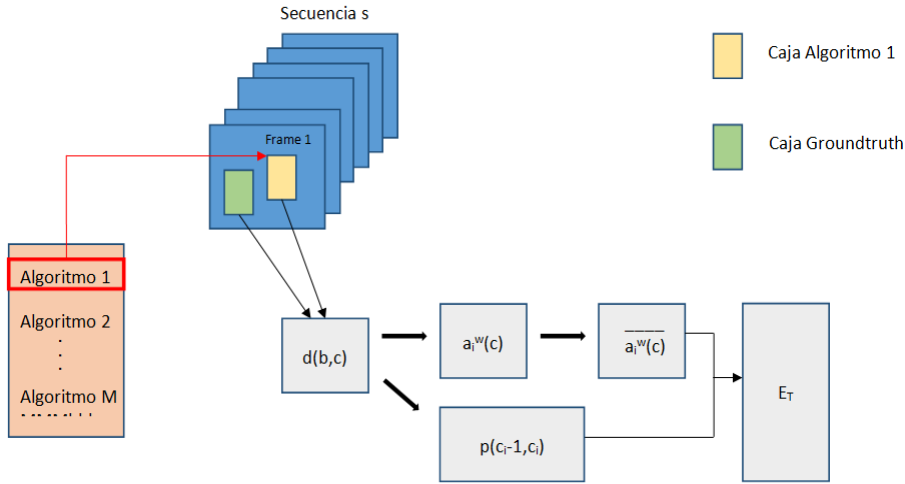


Figura 2.8: Diagrama de bloques algoritmo Superior Tracking Approach.

- The Local Approach:** En el seguimiento del objeto, se obtiene cierta ventaja al ir eliminando los resultados de seguimiento menos favorables, ya que pueden distorsionar y entorpecer el proceso de fusión. En concreto, este método elimina los algoritmos de seguimiento ineficientes por cada secuencia independientemente. Para este objetivo, se calcula el rendimiento P_j de cada resultado de seguimiento j en la secuencia. La ecuación del rendimiento es la siguiente:

$$P_j = \sum_{i \in N} a_i^w(b_{i,j}) \quad (2.5)$$

Después, se excluyen los peores resultados, es decir, los algoritmos con valores P_j más pequeños.

- The Global Approach:** Este nuevo método se utiliza para eliminar y descartar los algoritmos de seguimiento ineficientes, al igual que el método anterior. Este elimina los algoritmos indeseados para toda la secuencia a la vez. Para encontrar los candidatos a eliminar, primero se divide el conjunto de datos de entrenamiento en varias partes, y se realizan varios experimentos. Se comprueba la eficacia de los algoritmos en los ensayos y se eliminan permanentemente los que se hayan descartado en al menos 7 ensayos.

En la Figura 2.8 se representa el funcionamiento completo para seleccionar los algoritmos más eficientes para la fusión.

2.3.3. Tracker-Level Fusion for Robust Bayesian Visual Tracking

La novedad que incluye el método propuesto por este paper [[6]] se basa en la inclusión de una medida de fiabilidad online, es decir, en tiempo real, para identificar el nivel de calidad de cada tracker para su posterior fusión.

El primer paso consiste en estimar el estado, es decir, obtener el resultado de la búsqueda del objetivo. Después se procede a realizar una evaluación, donde se comprueba la calidad del seguimiento. En esta etapa, el algoritmo TLF se basa en la incertidumbre espacial de las partículas, cuya ecuación se muestra a continuación:

$$\widetilde{u}_k^i = \frac{\sqrt[2]{|\widehat{C}_k^i|}}{w_k^i h_k^i} \quad (2.6)$$

C es la matriz de covarianzas entre las partículas, w y h son el ancho y el alto del resultado obtenido, respectivamente, utilizados para normalizar la incertidumbre obtenida. la variable i corresponde al tracker utilizado.

El coeficiente de incertidumbre u muestra un valor cercano a cero y más o menos estable si el seguimiento se realiza correctamente, como ocurre en la Figura 2.9 con el algoritmo de color verde, todo lo contrario al algoritmo de color rojo, mostrando en este valores elevado y muy dispares a partir del frame 50.

Por último, se aplica una corrección mejorando el rendimiento del seguimiento mediante la fusión de los datos de los múltiples trackers utilizados. Dicha corrección se basa en comprobar los pesos de los resultados sobre cada píxel obtenidos a partir de los trackers a lo largo de la secuencia, reemplazando los pesos con valores por debajo de umbrales p establecidos por los resultados de los píxeles con pesos altos de otros trackers (Figura 2.10).

2.3.4. Randomized Ensemble Tracking

En la Figura 2.11 se muestra el diagrama de bloques que sigue este artículo. Este modelo empieza con un conjunto de clasificadores débiles C_i , un vector de pesos D sobre un espacio finito discreto, sobre el cual se utiliza una distribución de Dirichlet $Dir(D)$, y el dato de entrada x .

El método propuesto divide el dato de entrada en una rejilla de pequeñas parcelas para poder enviar las características de cada una de estas al clasificador débil correspondiente. Al mismo tiempo, se muestrea la distribución $Dir(D)$ obteniendo M muestras $D^{(1)}, D^{(2)}, \dots, D^{(M)}$. La fusión se halla en combinar Las M muestras con

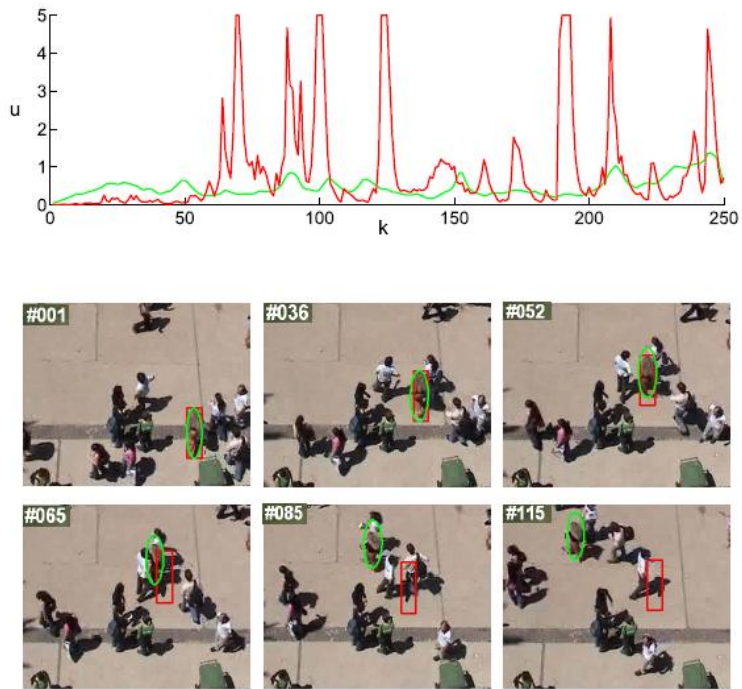


Figura 2.9: Distribución del coeficiente de incertidumbre. [[6]]

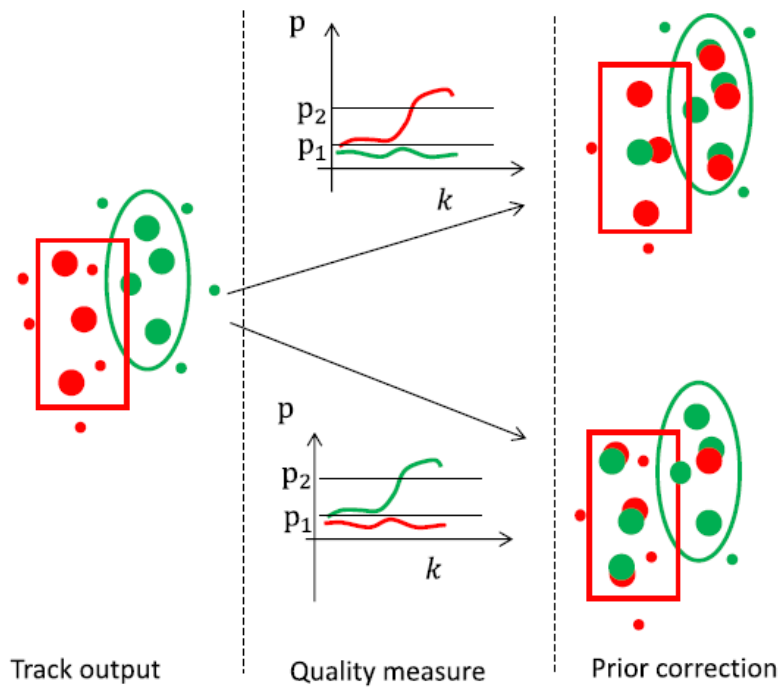


Figura 2.10: Corrección del estado. [[6]]

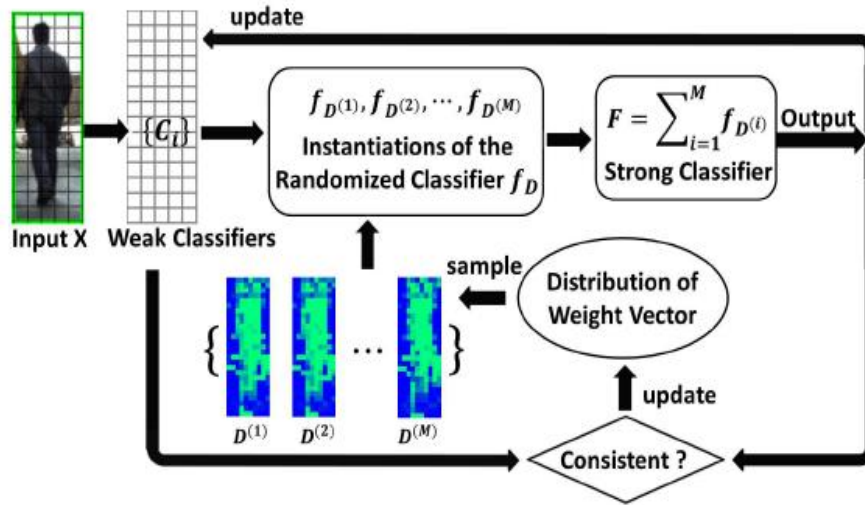


Figura 2.11: Randomized Ensemble Tracker .

las salidas de los clasificadores débiles obteniendo lo que se conoce como conjuntos de clasificadores débiles $f_{D^{(1)}}, f_{D^{(2)}}, \dots, f_{D^{(M)}}$.

Mediante estos conjuntos, como se puede apreciar en la figura 2.11, se calcula una aproximación F . Dicha aproximación se corresponde con salida esperada del método utilizado.

Para entradas posteriores, se recalcula la distribución $Dir(D)$ de acuerdo a la salida obtenida anteriormente.

2.4. Conclusiones

A lo largo de esta sección, se ha mostrado el proceso de aprendizaje sobre el seguimiento de objetos y temas relacionados. Se ha explicado de forma general el seguimiento de objetos, sus etapas principales, posibles problemas...

Por otra parte, se ha tratado el tema de la fusión del algoritmos, aclarando el motivo de su uso y las arquitecturas existentes.

Cuando se han argumentado los aspectos generales del tema tratado, se ha procedido a comentar y describir el conjunto de artículos leídos para profundizar en el tema, sobre todo en el aspecto de fusión, ya que este trabajo consiste en utilizar varios algoritmos de seguimiento. Dichos artículos principalmente se centran en seleccionar, entre el conjunto de algoritmos disponibles, aquellos en los que, mediante una serie de medidas, ofrecen mejores resultados. Además, sobre los algoritmos elegidos, se pueden corregir los posibles fallos que se produzcan a lo largo de la secuencia.

Este trabajo utiliza gran parte de la información de los papers comentados con anterioridad, en especial [2.3.1](#), ya que las métricas utilizadas en este artículo, como por ejemplo distancia entre centros o solapamiento entre cajas, son de gran utilidad de cara a comprobar que algoritmos están funcionando correctamente.

Capítulo 3

Desarrollo

En este capítulo se describen el conjunto de algoritmos utilizados en la tarea de seguimiento de objetos, así como las distintas métricas utilizadas para medir la eficacia de estos y poder elegir con cierto criterio en la última sección los algoritmos más adecuados de cara a una fusión de estos.

3.1. Esquema global

En la figura 3.1 se detalla el esquema general del desarrollo de este TFG. En primer lugar se utilizan los algoritmos de seguimiento obteniendo un conjunto de resultados. Después, sobre esos resultados se aplican un conjunto de medidas para poder evaluarlos. Para finalizar, se decide que algoritmos fusionar o combinar y cuales no observando las mediciones realizadas sobre los resultados de los algoritmos,

La fusión consta de dos fases: selección de algoritmos mediante métricas inter-tracker y combinación de algoritmos (equitativa o ponderada). La combinación de algoritmos ponderada puede beneficiarse de disponer de varias métricas tanto inter-tracker como intra-tracker, se pueden realizar diversas combinaciones entre estas de cara a obtener una fusión lo más óptima posible (Figura 3.1).

3.2. Algoritmos

No se entra en detalle sobre cada algoritmo utilizado. Se consideran *cajas negras* ya que el interés se muestra únicamente sobre el resultado obtenido de cada uno de ellos.

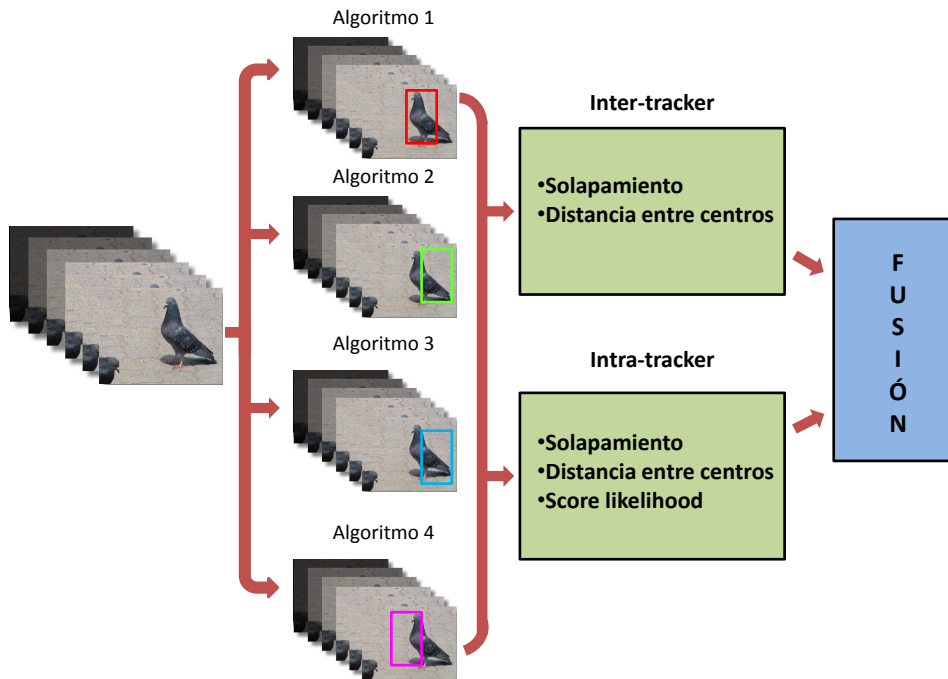


Figura 3.1: Esquema global del desarrollo.

3.2.1. Mean-shift Tracking

Dicho algoritmo se centra en encontrar y localizar modas en un conjunto de muestras de datos mediante el uso de la función de densidad de probabilidad (Figura 3.2). Como se puede apreciar en esta figura, se escoge en primer lugar un punto de referencia, para más tarde buscar en los modelos de los vecinos más cercanos el candidato que maximiza la función de densidad de probabilidad.

3.2.2. Robust Mean Shift Tracking with Corrected Background-Weighted Histogram

Este algoritmo se basa principalmente en el algoritmo Mean-shift [3.2.1], mejorando aspectos como la reducción de la interferencia del fondo en una imagen al localizar el objeto a seguir [[8]]. Por lo tanto, ofrece una convergencia y una precisión mayor que el algoritmo Mean-shift. Además, dicho algoritmo es robusto a posibles fallos de inicialización del objetivo.

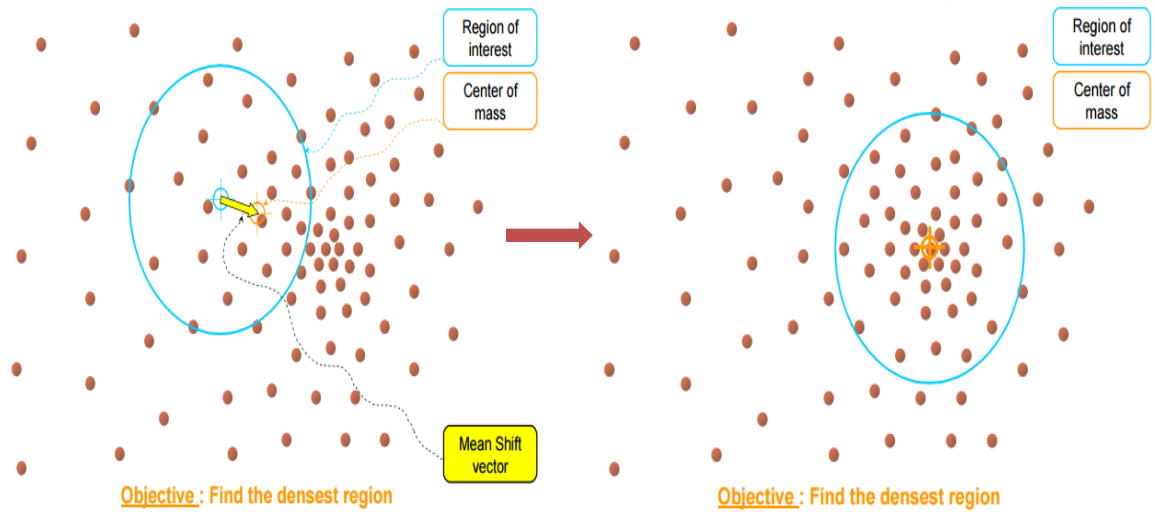


Figura 3.2: Funcionamiento del algoritmo Mean-shift.[7]

3.2.3. Particle Filter Color Tracker

El funcionamiento de este algoritmo se basa en mantener una distribución de la probabilidad en cada estado (escala, ubicación, etc.) del objeto a seguir. Dicha distribución se basa en un conjunto de muestras ponderadas, o partículas, las cuales se propagan en el tiempo usando una serie de ecuaciones conocidas como *Bayesian filtering equations*. Cada partícula representa una posible localización del objetivo, y para detectar la trayectoria del objeto en sí, se seleccionan las partículas con los pesos o medias más altas en cada frame.

3.2.4. Adaptive Color Attributes for Real-Time Visual Tracking

Este algoritmo parte de las ideas del algoritmo CSK, extendiéndolo mediante atributos de color, ya que sin estos atributos, se obtendrían unos resultados no del todo favorables en señales multicanal [[9]]. Además, existe la opción de poder seleccionar de manera previa la intensidad y el color en modelos de detección de personas, por ejemplo.

3.3. Métricas

El conjunto de métricas a utilizar se aplican a dos grupos: comparaciones inter-tracker e intra-tracker, es decir, comparaciones entre distintos algoritmos y comparaciones con el mismo algoritmo, respectivamente.

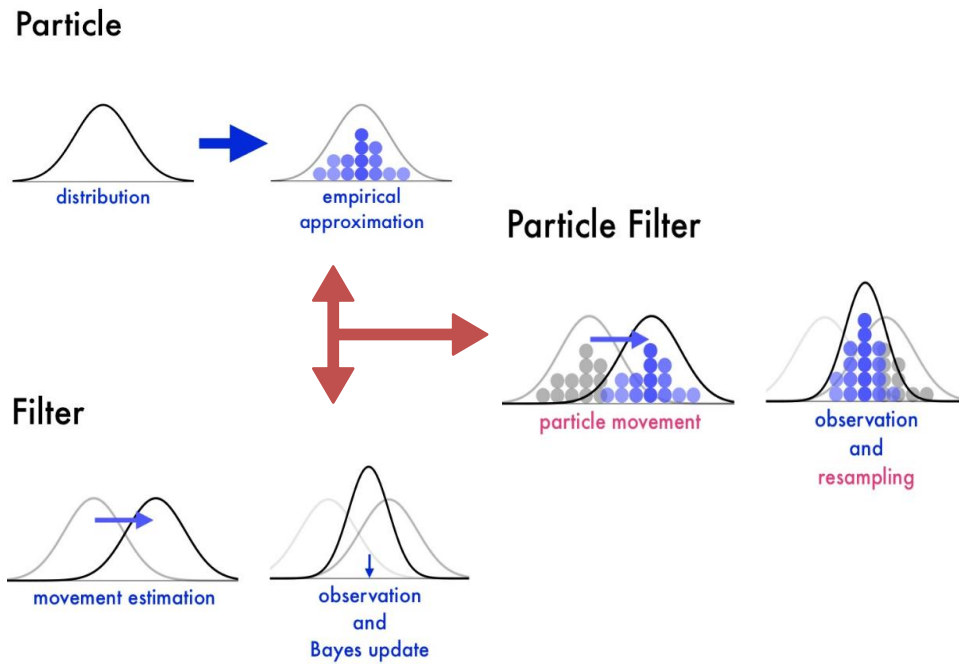


Figura 3.3: Funcionamiento algoritmo Particle Filter Color Tracker.

- **Métricas intra-tracker:** se mide en un mismo algoritmo, un frame en concreto y su frame anterior, para visualizar si hay desvíos del objeto a seguir fuera de lo normal, o posibles errores de seguimiento y detección.
- **Métricas Inter-tracker:** se mide entre diferentes algoritmos, en el mismo instante de la secuencia. Tiene un gran uso a la hora de apreciar que algoritmos funcionan de manera menos óptima frente al resto de algoritmos. Por lo tanto, se detectan los algoritmos menos eficientes para no usarlos en una hipotética fusión de algoritmos.

3.3.1. Solapamiento entre bounding boxes

Este tipo de métrica mide el grado de intersección de dos bounding boxes. Viene definida mediante la siguiente fórmula [[4]]:

$$F(R1, R2) = 1 - 2 \times \frac{Pr(R1, R2) \times Re(R1, R2)}{Pr(R1, R2) + Re(R1, R2)} \quad (3.1)$$

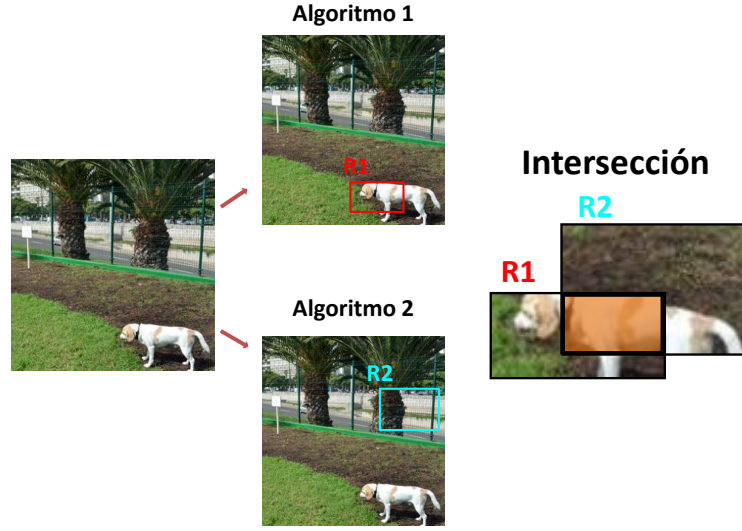


Figura 3.4: Solapamiento entre bounding boxes.

$$Pr(R1, R2) = \frac{A(R1 \cap R2)}{A(R2)} \quad (3.2)$$

$$Re(R1, R2) = \frac{A(R1 \cap R2)}{A(R1)} \quad (3.3)$$

Donde $R1$ y $R2$ son las áreas de los bounding boxes correspondientes, $Pr(R1, R2)$ es la intersección entre las dos áreas normalizada por el área de $R2$ y $Re(R1, R2)$ es la intersección entre las dos áreas normalizada por el área de $R1$.

El resultado será un número normalizado entre 0 y 1, siendo un solapamiento mayor con valores cercanos a 1.

3.3.2. Distancia entre centros

Este tipo de medida es muy utilizada en la evaluación del rendimiento de los trackers. Mediante esta métrica se calcula la distancia entre el centro de dos bounding boxes. Se utiliza la siguiente fórmula:

$$D(R1, R2) = \frac{d_{R1.R2}}{diagonal_{R1} + diagonal_{R2}} \quad (3.4)$$

Donde $d_{R1, R2}$ es la distancia entre el centro del bounding box $R1$ y el del bounding box $R2$. Se utilizan las diagonales de dichas bounding boxes para poder normalizar el resultado.

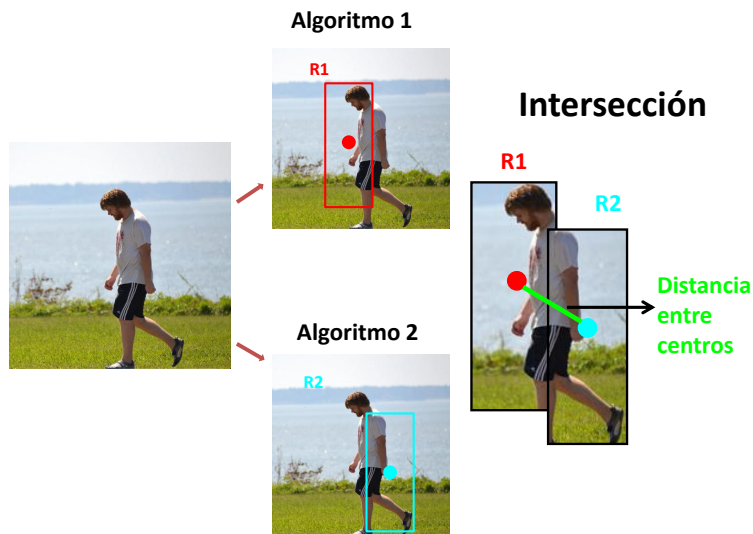


Figura 3.5: Distancia entre centros.

3.3.3. Score Likelihood

Mediante dicha medida, se comprueba lo distintos que son los algoritmos entre sí. Se empieza obteniendo un mapa de confianza de la secuencia para cada algoritmo (Figura 3.6). Un mapa de confianza simplemente es una imagen de un tamaño parecido o igual a la frame actual, donde se representan en cada partícula de esta imagen, una probabilidad de que el objeto a seguir se encuentre en esa posición. Su funcionamiento es similar a un mapa de calor. Posteriormente, se consigue de cada mapa de confianza, distintas áreas, una por cada algoritmo. dichas áreas corresponden con los bounding boxes de cada algoritmo.

Después, se realiza un promedio de la suma de las áreas obtenidas para cada mapa de confianza (Figura 3.7).

Con esta métrica se podrá observar lo diferentes que son los resultados de otros algoritmos frente al algoritmo del mapa de confianza obtenido.

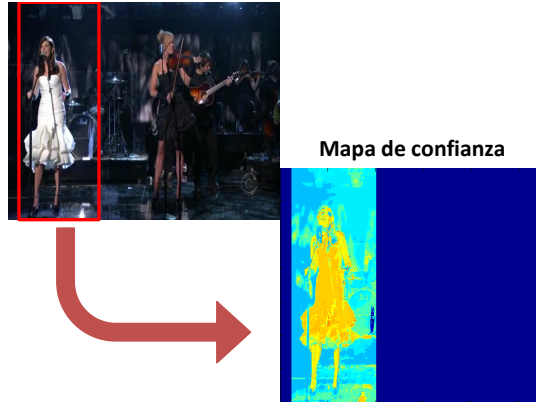


Figura 3.6: Mapa de confianza.

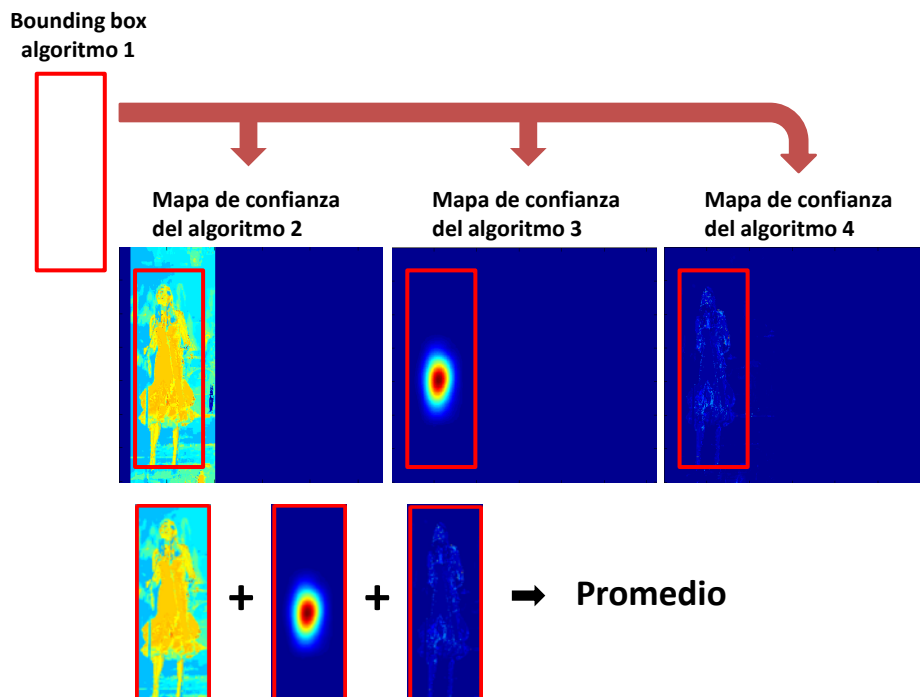


Figura 3.7: Score Likelihood.

3.4. Fusión

Después de obtener el conjunto de medidas descritas anteriormente, se procede a realizar una selección de los algoritmos con resultados más favorables de cara al seguimiento del objeto deseado.

En primer lugar, se realiza una selección sobre los resultados (ver gráficas siguiente capítulo) obtenidas por las métricas inter-tracker. En dichas gráficas, para un algoritmo en concreto, se realiza un promedio de todos los resultados donde se utilice dicho algoritmo $Al(i)$. Si el promedio de los resultados anteriores $M(Al(i))$ se encuentra por debajo de un determinado umbral a definir τ , el algoritmo se utiliza para la fusión de estos. Si el promedio se encuentra por encima del umbral τ , el algoritmo se descarta. Este paso se implementa para cada algoritmo.

Una vez que están seleccionados los algoritmos a fusionar, se procede a la fase de combinación de dichos algoritmos. En esta fase se pueden utilizar diferentes métodos. El método más sencillo es un promedio de los valores obtenidos por cada algoritmo seleccionado, como se puede apreciar en la siguiente ecuación:

$$Fe = \frac{\sum_{i=1}^N Al(i)}{N} \quad M(Al(i)) \leq \tau, \quad (3.5)$$

donde Fe es el resultado de la fusión equitativa, es decir, el promedio de los resultados de cada algoritmo $Al(i)$ seleccionado y N es el número de algoritmos.

El otro método, definido en la siguiente ecuación, consiste en un promedio ponderado, ponderando con un valor más alto aquellos algoritmos que tengan los mejores resultados en las métricas intra-tracker.

$$Fp = P(i) \sum_{i=1}^M Al(i), \quad M(Al(i)) \leq \tau, \quad (3.6)$$

donde Fp es la fusión ponderada, M es el número de algoritmos elegidos y $P(i)$ es el factor de ponderación correspondiente al algoritmo *elegido* $As(i)$, además se debe cumplir que $\sum_{i=1}^N P(i) = 1$.

Capítulo 4

Resultados experimentales

En este capítulo se presentan los resultados obtenidos durante el proyecto utilizando el conjunto de métricas utilizadas y los algoritmos de fusión propuestos, además del conjunto de datasets utilizados para llevar a cabo los experimentos.

El seguimiento de objetos llevado a cabo se ha realizado de forma offline, es decir, no se ha realizado en tiempo real.

4.1. Datasets

El conjunto de datasets que se han utilizado a lo largo de los experimentos provienen de las secuencias utilizadas en el Tracker Benchmark v1.0 [[10]]. Se ha seleccionado un total de 10 secuencias diferentes, atendiendo a los problemas habituales en el seguimiento de objetos [2.2.2], como pueden ser cambios de iluminación, oclusiones, ruido, etc. Las secuencias seleccionadas son las descritas en la Tabla 4.2. Entre todos los problemas que tienen que solventar los algoritmos de seguimiento, los más destacados se nombran en la Tabla 4.1.

Oclusiones	Deformaciones	Cambios de iluminación	Variaciones de escala
FaceOcc1, Bolt, Singer1, Walking.	Basketball, Bolt, Skiing, Walking.	Basketball, Rolling, Singer1, Skiing.	Singer1, Skiing, Walking

Tabla 4.1: Desafíos que presentan los datasets.

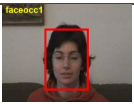



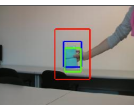





Nombre	Imagen	# frames	Descripción
FaceOcc1		100	Rostro de una mujer desplazándose alrededor de la cámara.
Basketball		267	En dicha secuencia el objeto a seguir es un jugador de baloncesto.
Bolt		73	En este vídeo el objeto a detectar y seguir es Bolt en una carrera.
I1_basic		200	Esta secuencia es muy simple ya que el objeto a seguir es un óvalo.
I2_basic		200	El objeto a seguir es un libro moviéndose alrededor la imagen.
I3_cars		200	En esta secuencia se puede ver un coche desplazándose de derecha a izquierda.
Rolling		164	En este vídeo se puede ver un motorista haciendo acrobacias.
Singer1		120	El objeto a detectar es una cantante sobre un escenario.
Skiing		81	En este vídeo se pretende detectar y seguir a un esquiador.
Walking		200	Esta secuencia consiste en una persona paseando por la calle.

Tabla 4.2: Datasets utilizados.

Comparación algoritmos	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Walking
MS-CBWH	.01±.002	.20±.005	.15±.004	.99±.001	.09±.007	.47±.109	.50±.041	.44±.062	.97±.011	.28±.039
MS-PFC	.34±.003	.34±.015	.45±.012	.85±.064	.41±.006	.60±.083	.36±.024	.67±.069	.92±.047	.39±.018
MS-ACA	.11±.001	.12±.003	.90±.043	.95±.035	.11±.001	.20±.024	.82±.075	.31±.019	.90±.058	.26±.012
CBWH-PFC	.35±.006	.23±.009	.53±.029	.00±.000	.36±.009	.66±.061	.61±.038	.82±.079	.95±.019	.36±.030
CBWH-ACA	.12±.002	.18±.007	.92±.025	.88±.035	.12±.005	.38±.060	.94±.015	.65±.128	.90±.036	.42±.039
PFC-ACA	.19±.004	.40±.001	.86±.073	.97±.012	.33±.002	.47±.039	.88±.059	.32±.001	.89±.050	.64±.034

Tabla 4.3: Resultados promedios Distancia entre centros inter-tracker.

Comparación algoritmos	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Walking
MS-CBWH	.03±.008	.30±.009	.33±.016	.04±.004	.17±.026	.10±.033	.58±.079	.50±.059	.04±.038	.39±.052
MS-PFC	.90±.006	.79±.003	.69±.005	.20±.128	.79±.0060	.54±.150	.68±.014	.75±.088	.09±.062	.69±.009
MS-ACA	.22±.001	.21±.012	.12±.070	.01±.006	.21±.0050	.37±.044	.13±.069	.46±.040	.07±.040	.53±.034
CBWH-PFC	.86±.038	.78±.005	.83±.026	.00±.000	.76±.010	.27±.127	.65±.118	.23±.128	.08±.063	.58±.057
CBWH-ACA	.24±.006	.28±.013	.15±.104	.03±.025	.23±.020	.10±.031	.14±.108	.44±.163	.12±.079	.61±.060
PFC-ACA	.89±.009	.79±.004	.15±.088	.02±.018	.69±.002	.75±.021	.15±.094	.69±.003	.13±.084	.51±.209

Tabla 4.4: Resultados promedios Solapamiento inter-tracker.

4.2. Métricas de evaluación

Para poder evaluar las métricas y comprobar si ofrecen buenos o malos resultados, se obtienen en primer lugar una gráfica del groundtruth error de cada algoritmo, es decir, como de diferente es el resultado del algoritmo respecto al resultado correcto.

4.3. Resultados globales

Para finalizar esta sección, se introducen las Tablas 4.3 hasta 4.8 , donde se muestran todos los resultados obtenidos, en formato media \pm varianza, utilizando las métricas comentadas con anterioridad. En la Tabla 4.8 Se puede observar que el error de groundtruth de la fusión equitativa es en general menor que los demás algoritmos. Por otra parte, el error de groundtruth de la fusión ponderada es menor que el de la fusión equitativa solamente en tres secuencias. Esto se debe a la insuficiencia de datos con los que elegir que algoritmos ponderan más.

Algoritmo	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Walking
MS	.07±.000	.03±.001	.06±.001	.03±.003	.01±.000	.01±.001	.04±.002	.03±.001	.13±.042	.04±.001
CBWH	.07±.000	.03±.002	.05±.002	.08±.016	.01±.000	.01±.001	.07±.010	.12±.052	.16±.031	.06±.007
PFC	.08±.000	.03±.001	.06±.000	.03±.001	.01±.249	.01±.694	.03±.001	.03±.000	.13±.012	.03±.001
ACA	.08±.070	.03±.001	.01±.275	.03±.002	.01±.180	.01±.085	.05±.012	.03±.515	.11±.013	.03±.000

Tabla 4.5: Resultados promedios Distancia entre centros intra-tracker.

Algoritmo	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Walking
MS	.14±.001	.05±.004	.08±.002	.07±.010	.01±.000	.03±.005	.07±.004	.10±.004	.17±.051	.09±.006
CBWH	.13±.001	.06±.006	.09±.004	.16±.037	.02±.001	.04±.007	.10±.017	.18±.052	.23±.053	.12±.018
PFC	.27±.096	.19±.016	.20±.003	.10±.005	.03±.000	.07±.002	.12±.009	.25±.006	.31±.032	.12±.009
ACA	.14±.000	.06±.004	.00±.000	.10±.025	.00±.592	.01±.000	.01±.002	.10±.001	.21±.028	.08±.003

Tabla 4.6: Resultados promedios Solapamiento intra-tracker.

Algoritmo	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Walking
MS	.69±.036	.08±.031	.69±.021	.07±.029	-.08±.069	.01±.014	.09±.034	.02±.010	.05±.030	.09±.024
CBWH	.69±.037	.09±.029	.48±.039	.00±.013	-.06±.092	.07±.018	.06±.057	.04±.033	.01±.030	.09±.019
PFC	.67±.027	.20±.024	.07±.031	.00±.000	.25±.053	.30±.162	.08±.036	.00±.000	.00±.000	.03±.010
ACA	.70±.032	.08±.031	.13±.073	.02±.015	.16±.029	.34±.046	.06±.035	.39±.031	.02±.013	.08±.034

Tabla 4.7: Resultados promedios Score likelihood intra-tracker.

4.4. Resultados específicos

Se ha desarrollado un *script* en el cual se recorre cada secuencia frame a frame aplicando el conjunto de métricas a utilizar para poder comparar los algoritmos utilizados en la tarea de seguimiento de objetos. Las siglas “MS” se refieren al algoritmo Mean shift, “CBWH” se corresponde con Robust Mean Shift Tracking with Corrected Background-Weighted Histogram, “PFC” con Particle Filter Color Tracker y “ACA” con Adaptive Color Attributes for Real-Time Visual Tracking.

El siguiente conjunto de gráficas obtenidas se dividen en comparación inter-tracker e intra-tracker [3.3] y Score Likelihood [3.3.3].

4.4.1. Inter-tracker

Dentro de la comparación inter-tracker, se obtienen dos gráficas para cada secuencia. La primera correspondiente al solapamiento entre bounding boxes y la segunda correspondiente a la distancia entre centros normalizada [3.3.1] [3.3.2].

A continuación podemos ver algunos ejemplos en las siguientes figuras.

Algoritmo	FaceOcc1	Basketball	Bolt	I1_basic	I2_basic	I3_cars	Rolling	Singer1	Skiing	Skiing
MS	.23±.004	.49±.013	.84±.076	.22±.001	.26±.014	.17±.005	.78±.048	.62±.086	.96±.020	.43±.030
CBWH	.20±.012	.61±.035	.99±.032	.24±.006	.35±.020	.28±.013	.82±.042	.75±.105	.95±.013	.73±.095
PFC	.19±.008	.50±.019	.28±.007	.66±.059	.35±.020	.22±.004	.77±.106	.54±.110	.91±.048	.16±.003
ACA	.08±.002	.19±.031	.97±.017	.00±.000	.16±.005	.91±.048	.86±.081	.47±.024	.90±.060	.16±.004
Fusión Equitativa	.11±.001	.35±.013	.91±.058	.16±.002	.21±.009	.13±.005	.79±.053	.49±.069	.90±.064	.23±.013
Fusión ponderada	.09±.001	.49±.030	.99±.004	.08±.001	.27±.015	.33±.009	.85±.032	.46±.073	.99±.004	.70±.115

Tabla 4.8: Resultados promedios Groundtruth error.

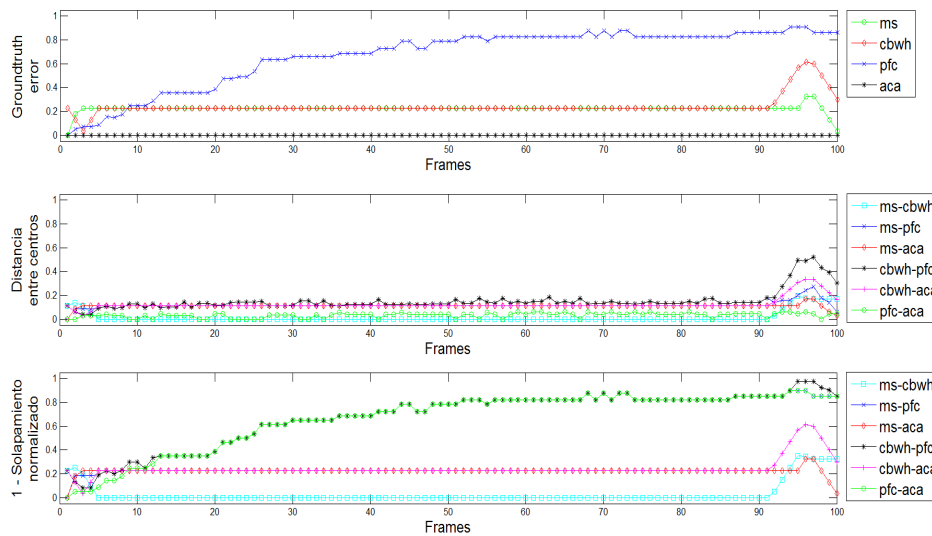


Figura 4.1: Inter-tracker secuencia I1_basic.

En la Figura 4.1 se puede apreciar como el algoritmo MS, CBWH y ACA tienen un error de groundtruth relativamente bajo. Esto se debe a que es un objeto fácil de seguir. Sin embargo PFC a medida que avanzan las frames, su error de groundtruth aumenta. Otro detalle a observar es el ligero aumento del error sobre la secuencia 96. Esto ocurre así ya que en dicha frame el objeto cambia de trayectoria. En las gráficas de distancia entre centros y solapamiento, se aprecia valores muy cercanos a cero para la comparación MS-CBWH. Esto es así porque CBWH se basa en MS con ligeras mejoras. También el algoritmo ACA obtiene buenos resultados como se puede comprobar en las comparaciones MS-ACA y CBWH-ACA. Por otra parte, se observan más valores altos en las comparaciones con el algoritmo PFC, al obtener este un seguimiento ineficiente.

En las Figuras 4.2 y 4.3 se pueden observar más ejemplos de otras secuencias.

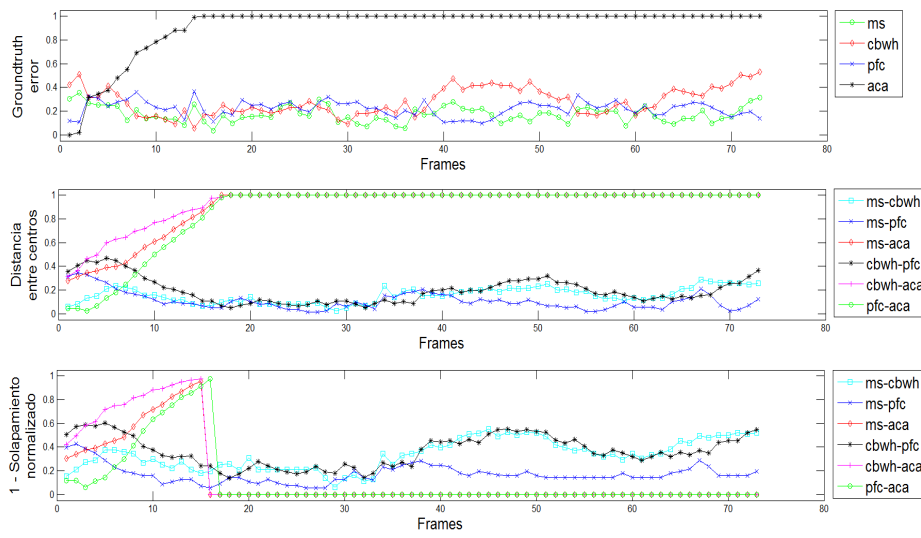


Figura 4.2: Inter-tracker secuencia I3_cars_basic_2.

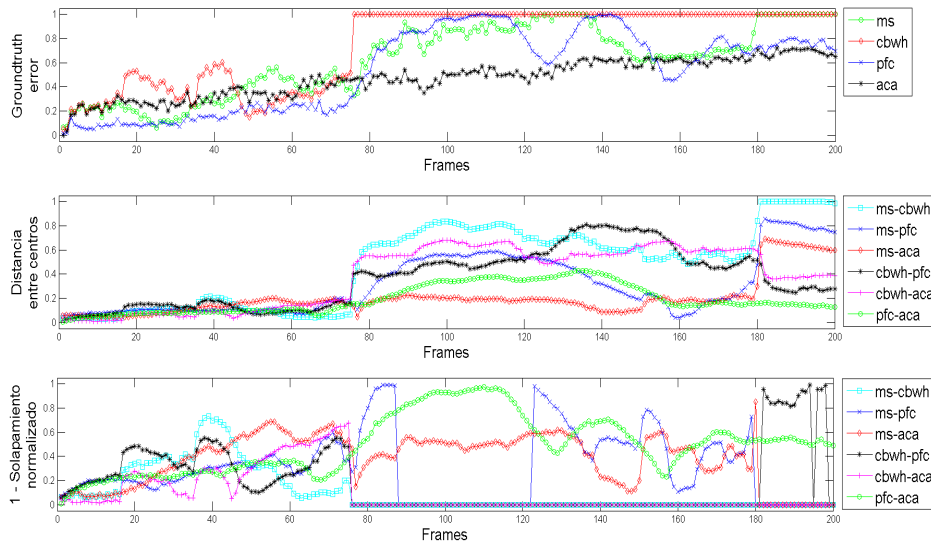


Figura 4.3: Inter-tracker secuencia Singer1.

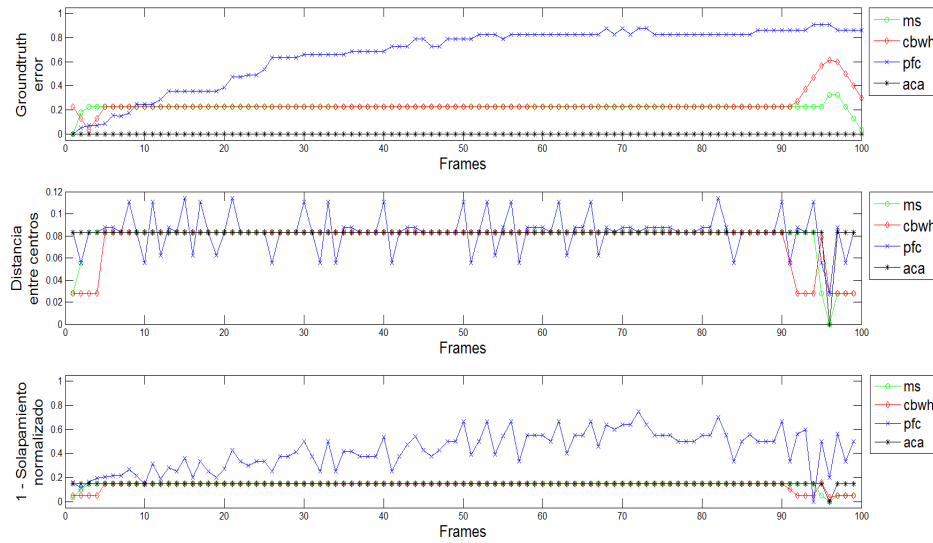


Figura 4.4: Intra-tracker secuencia I1_basic.

4.4.2. Intra-tracker

En la comparación intra-tracker, al igual que en la comparación inter-tracker, se obtiene una comparación respecto a la distancia entre centros y una comparación referente al solapamiento de los bounding boxes del algoritmo a analizar entre dos frames consecutivos.

En la Figura 4.4 se puede apreciar la inestabilidad del algoritmo PFC frente al resto de algoritmos, los cuales se mantienen estables y con valores cercanos a cero. Al apreciar valores atípicos con este tipo de medida podemos detectar que algoritmos funcionan de manera más o menos óptima.

En las Figuras [4.5] [4.6] se muestran otros ejemplos. En la secuencia *I3_cars* se puede visualizar en el algoritmo ACA, a partir del frame 8, valores atípicos, ya que decaen los valores de las métricas a cero y se mantienen a cero durante toda la secuencia.

Esta afirmación queda corroborada mediante la gráfica de groundtruth error, donde se puede ver como aumenta su valor hasta 1.

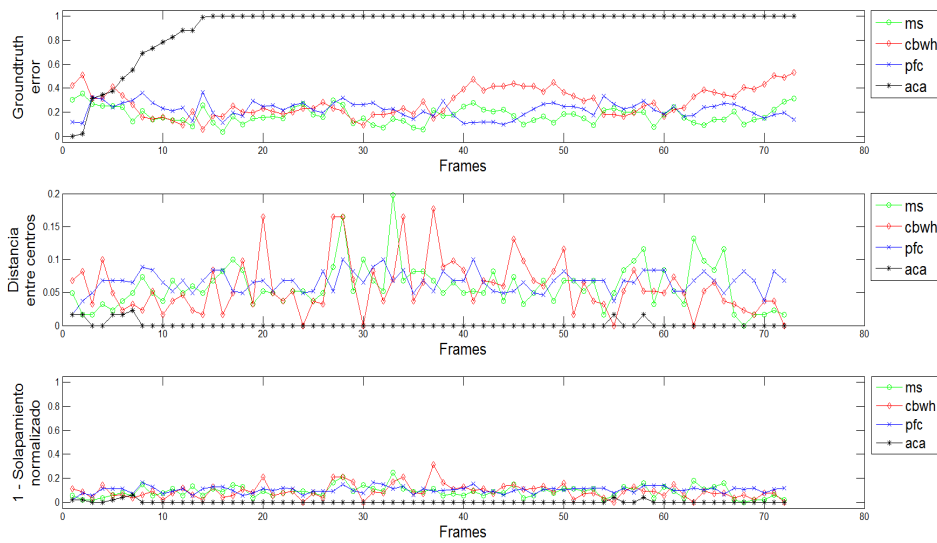


Figura 4.5: Intra-tracker secuencia I3_cars_basic_2.

En la siguiente figura el algoritmo CBWH presentan un pico extraño en el frame 75 de la secuencia Singer1. Efectivamente visualizando la gráfica de groundtruth error se puede apreciar un aumento creciente del error hasta 1. Lo que ha ocurrido es que hasta ese frame el algoritmo ha sido capaz de seguir al objeto pero por algún problema de los que pueden aparecer [2.2.2] ha perdido la pista al objeto.

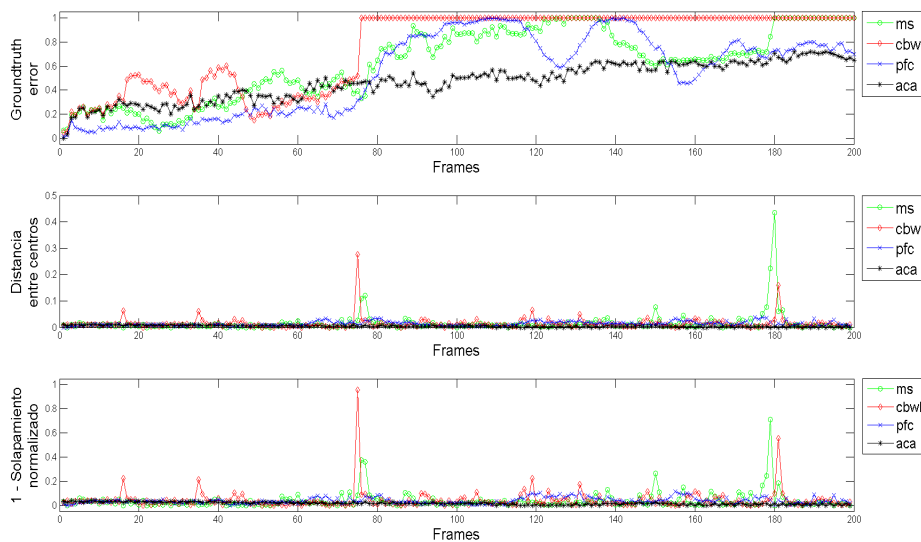


Figura 4.6: Intra-tracker secuencia Singer1.

4.4.3. Score Likelihood

A continuación se comentan los resultados obtenidos de la métrica Score Likelihood.

En la figura 4.7 se puede observar como el algoritmo PFC alcanza su máximo en el frame 8. Por lo tanto, en este frame, el bounding box resultante de este algoritmo detecta de manera notable el objeto en los mapas de confianza obtenidos por los demás algoritmos. A medida que avanzan los frames se puede comprobar como descende de manera brusca esta medida, ya que el resultado de este algoritmo se separa de los resultados de los demás. Un caso similar a este ocurre con el algoritmo ACA.

En general los demás algoritmos obtienen buenos resultados, ya que se parecen entre sí de manera notoria. Las gráficas (Figura 4.8) (Figura 4.9) muestran más ejemplos de esta métrica:

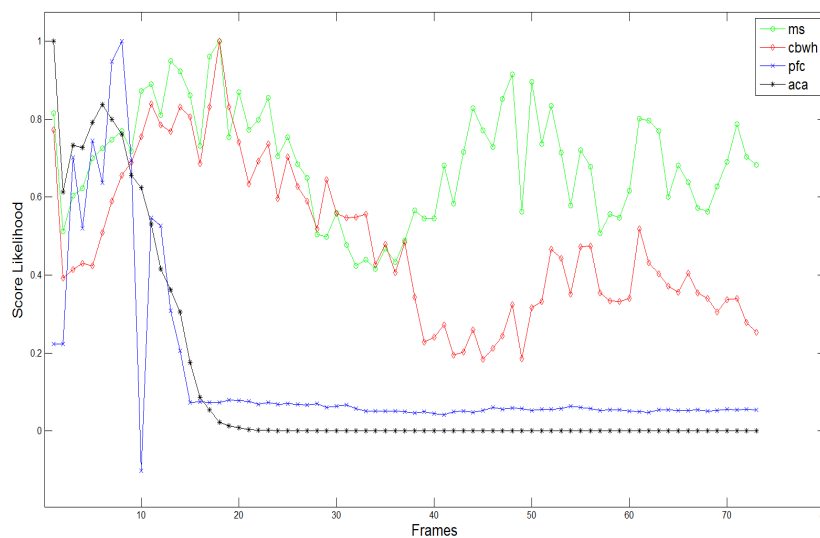


Figura 4.7: Score Likelihood secuencia I3_cars .

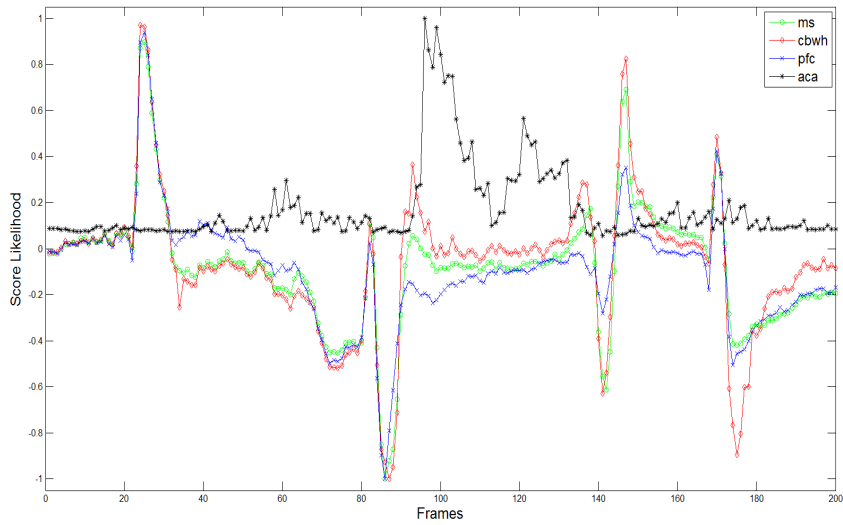


Figura 4.8: Score Likelihood secuencia FaceOcc1.

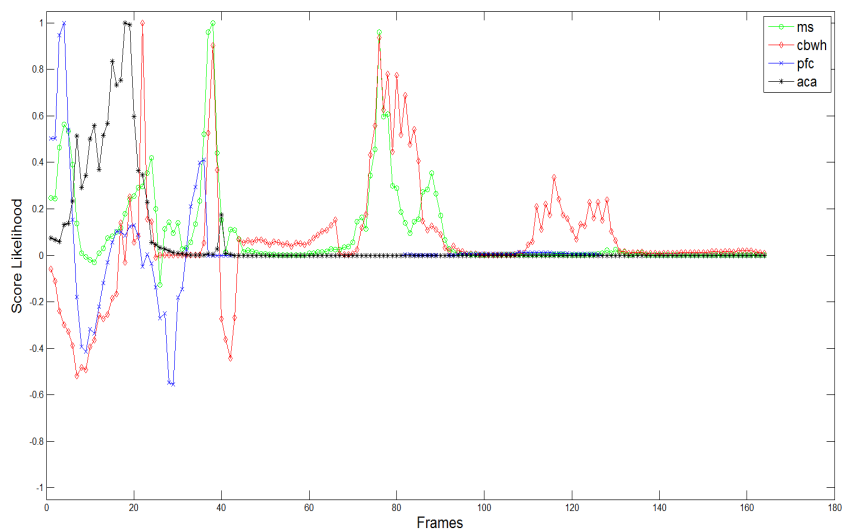


Figura 4.9: Score Likelihood secuencia Rolling.

4.4.4. Fusión

Al obtener los resultados de las métricas, el siguiente paso para establecer un seguimiento óptimo del objeto consiste en combinar lo mejor de cada algoritmo [3.4].

Se ha utilizado el solapamiento como métrica inter-tracker para la selección de algoritmos [3.3.1], con un umbral de .7. En primer lugar, en el proceso de combinación de algoritmos, se ha utilizado la fusión equitativa, donde no se necesita utilizar una

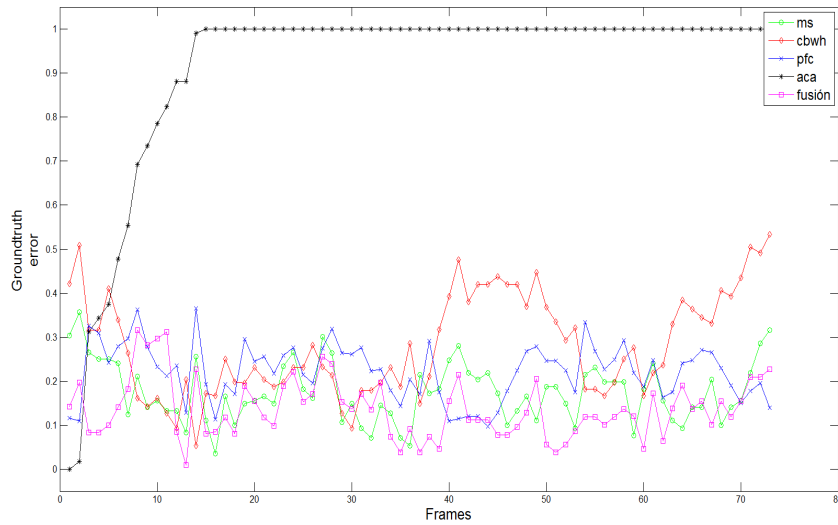


Figura 4.10: Comparación entre los algoritmos y la fusión secuencia I3_cars_basic_2.

métrica en concreto, ya que simplemente es un promedio de los valores de los resultados obtenidos por los algoritmos seleccionados. En la mayoría de las secuencias, se obtiene un error de groundtruth relativamente menor en comparación con los demás algoritmos en conjunto. En la figura 4.10 podemos ver como prácticamente en todos los frames obtenemos un error de groundtruth menor, mientras que en la figura 4.11 únicamente es superado por el algoritmo ACA.

Sin embargo, en otras secuencias, por ejemplo en la secuencia Bolt (Figura 4.12), tres de los cuatro algoritmos pierden la pista el objetivo a partir del frame 45, y al obtener un valor alto de solapamiento entre ellos mismos, se generan en la fusión unos valores altos de error de groundtruth.

De cara a mejorar la fusión, se incluye una segunda fase en la cual se ponderan los algoritmos según el valor obtenido a partir del coeficiente intra-tracker. Como medida intra-tracker se utilizó la distancia entre centros, sin obtener los resultados esperados ya que si un algoritmo falla en el seguimiento y se mantiene en un punto fijo a lo largo de los frames, obtendrá un valor óptimo en lo que se refiere a la distancia entre centros, por lo tanto será ponderado con un valor más alto.

Por consiguiente, se utilizó en la medida intra-tracker el Score Likelihood en vez de la distancia entre centros. Con dicho cambio se obtienen en algunas secuencias un error de groundtruth menor que la fusión equitativa, como se puede apreciar en las figuras 4.13 y 4.14. En lo que se refiere a la ponderación de algoritmos, el algoritmo con mejor resultado intra-tracker pondera un 60% del total. El segundo mejor algoritmo pondera un 25%, y si no se han descartado, los siguientes ponderan

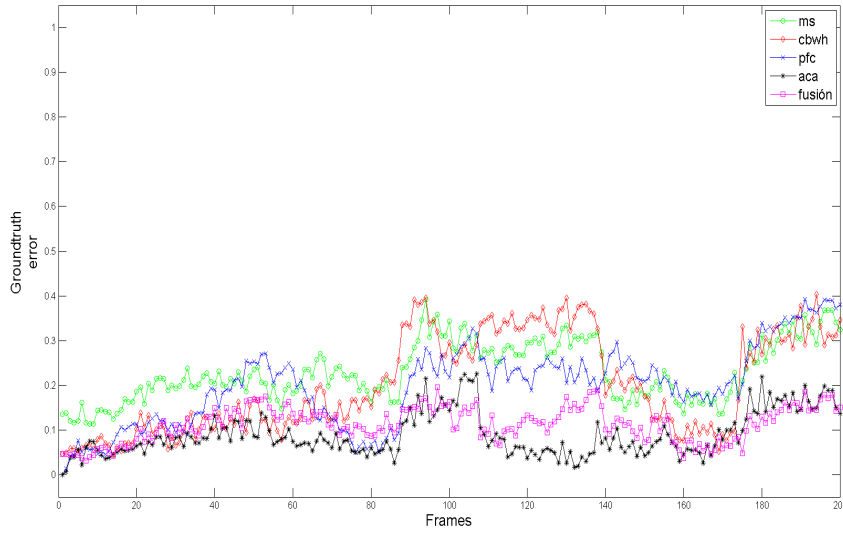


Figura 4.11: Comparación entre los algoritmos y la fusión secuencia FaceOcc1.

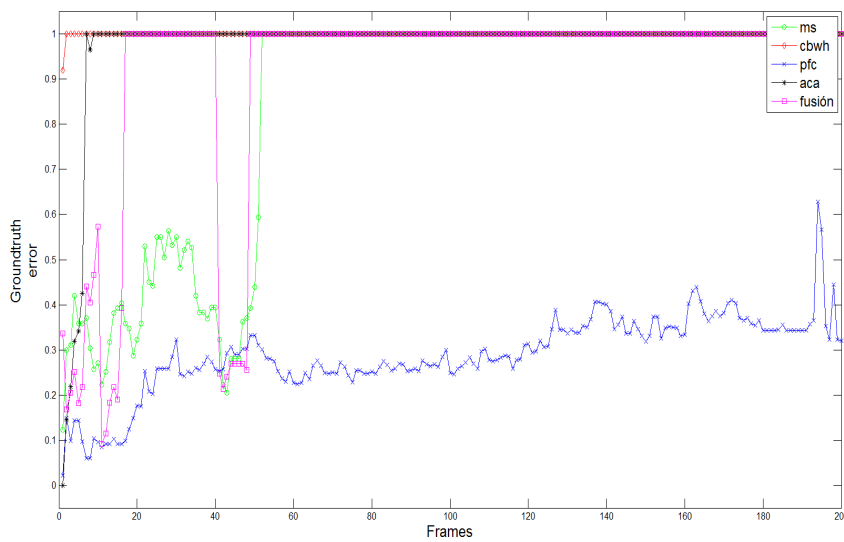


Figura 4.12: Comparación entre los algoritmos y la fusión secuencia Bolt.

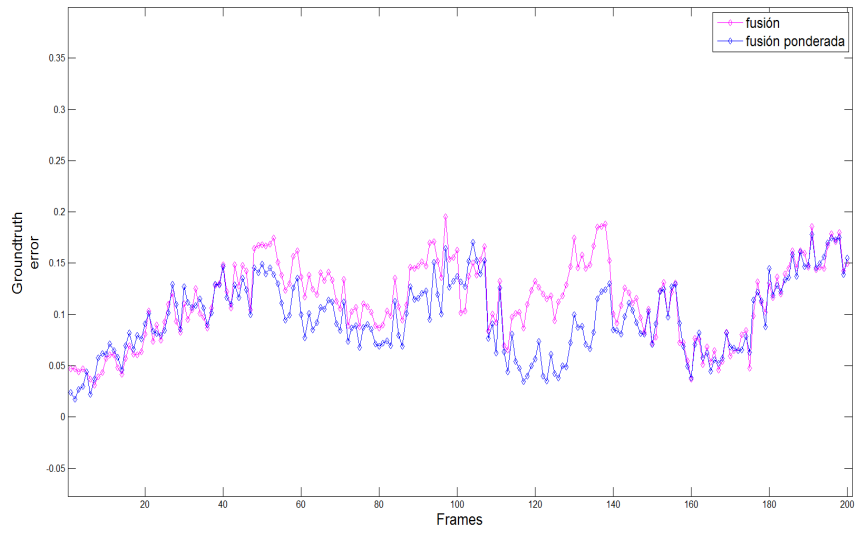


Figura 4.13: Comparación entre la fusión equitativa y la fusión ponderada secuencia FaceOcc1.

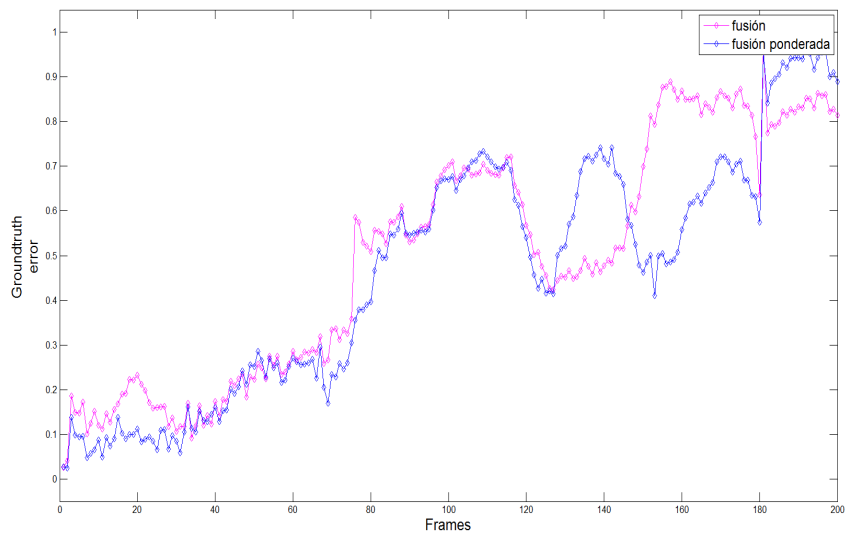


Figura 4.14: Comparación entre la fusión equitativa y la fusión ponderada secuencia Singer.

10 % y 5 % respectivamente.

Capítulo 5

Conclusiones y trabajo futuro.

5.1. Conclusiones.

El objetivo principal de este proyecto era analizar el comportamiento de un conjunto de algoritmos a la hora de detectar y seguir a un objeto en concreto en una secuencia. Además, gracias a este análisis, se ha podido definir una serie de pautas para poder combinar las mejores características de cada algoritmo. Este proceso se ha logrado de forma satisfactoria debido a un estudio previo sobre el seguimiento de objetos de forma general, para más adelante, gracias a la lectura de un conjunto de papers donde han desarrollado una serie de algoritmos de seguimiento, poder realizar un aprendizaje más detallado sobre este tema.

Una vez realizado el proceso de aprendizaje, se definieron tres tipos de métricas [3.3] para poder medir la eficacia de los algoritmos. Dichas métricas son el solapamiento entre bounding boxes, la distancia entre centros y el Score likelihood.

Por otra parte, el proceso de análisis de la fiabilidad de cada algoritmo se divide en dos partes: la primera corresponde a un análisis inter-tracker, es decir, un análisis entre los algoritmos. La segunda parte corresponde con un análisis intra-tracker, es decir, un análisis de cada algoritmo en distintos instantes de tiempo. En concreto se compara un frame y su frame anterior.

Al obtener los resultados provenientes de las métricas, se definen dos métodos de fusión [3.4]. Una fusión equitativa donde se realiza el promedio de los algoritmos seleccionados, y una fusión ponderada, en la cual cada algoritmo estará ponderado de diferente manera según el resultado que haya obtenido en la métrica intra-tracker. A mejor resultado, mayor ponderación.

Aunque hay algunas secuencias donde los algoritmos producen resultados ineficientes, en general funcionan de manera aceptable salvo en alguna secuencia donde se

presentan varios problemas de seguimiento [2.2.2], en las cuales funcionan de manera aceptable uno o dos algoritmos.

Respecto a la fusión, en general la fusión equitativa obtiene buenos resultados, ya que adquieren, en una gran parte de las secuencias utilizadas, un error de groundtruth menor que el error de groundtruth correspondiente a los algoritmos en sí.

Con la fusión ponderada se pretende mejorar los resultados de la fusión equitativa. Solo se consigue, aunque no en todas las secuencias, utilizando en la fase de ponderación la métrica Score likelihood.

5.2. Trabajo futuro.

Tras la realización de este proyecto aparecen varias líneas de trabajo futuro:

- Utilizar diferentes tipos de métricas, analizando y comparando sus resultados con los de las métricas utilizadas, para obtener diferentes puntos de vista de cara a la comprobación del funcionamiento de los algoritmos.
- Incluir un mayor número de algoritmos y secuencias para poder realizar experimentos más fiables y exhaustivos.
- Incluir nuevas modalidades de fusión intentando optimizar el error de groundtruth, ya que un modelo de fusión equitativo y un modelos ponderado, son insuficientes de cara a obtener los mejores resultados posibles.
- Implementar este proyecto en modo online, es decir, en tiempo real, para poder utilizarlo en sistemas que requieran este tipo de modalidad.

Bibliografía

- [1] ADCIS. Adcis seguimiento del movimiento aplicaciones ejemplos. [2](#)
- [2] Proyecto Hesperia. Deteccion y seguimiento de objetos en primer plano. [2](#)
- [3] E. Maggio y A. Cavallaro. *Video tracking, Theory and Practice*. Wiley, 2011. [6](#)
- [4] Longfei Zhang y Alexander Hauptmann Yue Gao, Rongrong Ji. Symbiotic tracker ensemble toward a unified tracking framework. 2014. [11](#), [12](#), [22](#)
- [5] Alain Pagani y Didier Stricker Christian Bailer. A superior tracking approach: Building a strong tracker through fusion. 2014. [12](#)
- [6] Andrea Cavallaro y Carlo S. Regazzoni Tewodros, A. Biresaw. Tracker-level fusion for robust bayesian visual tracking. 2015. [15](#), [16](#)
- [7] PSU R.Collins, CSE. *Mean-shift Tracking*. 2006. [21](#)
- [8] David Zhang y Chengke Wu Jifeng Ning, Lei Zhang. Robust mean shift tracking with corrected background-weighted histogram. 2010. [20](#)
- [9] Michael Felsberg y Joost van de Weijer Martin Danelljan, Fahad Shahbaz Khan. Adaptive color attributes for real-time visual tracking. 2014. [21](#)
- [10] Online object tracking: A benchmark, 2013. [27](#)

Glosario

MS - Meanshift

CBWH - Corrected Background-weighted histogram

