UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

SEGUIMIENTO DE OBJETOS BASADO EN MÚLTIPLES CARACTERÍSTICAS

Guillermo Luna Aguado
Tutor: Juan Carlos San Miguel Avedillo
Ponente : Jose María Martínez Sánchez

JUNIO 2017

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

SEGUIMIENTO DE OBJETOS BASADO EN MÚLTIPLES CARACTERÍSTICAS

AUTOR: Guillermo Luna Aguado
TUTOR: Juan Carlos San Miguel Avedillo

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2017

Trabajo parcialmente financiado por el gobierno español bajo el proyecto TEC2014-53176-R (HA-Video)

Resumen (castellano)

Este trabajo de fin de grado tiene como objetivo principal el seguimiento de objetos en secuencias de video basándonos en sus caracteristicas de color y de forma. Para ello es necesario diseñar un algoritmo de seguimiento, que localice un objeto a través del tiempo dada su posición inicial y estime su posición en instantes posteriores.

En este TFG se hará uso del filtro de partículas clásico basado en sus características de color en el espacio RGB. El primer paso fue hacer compatible el algoritmo con otros espacios de color, y se escogió HSV como alternativa. Posteriormente se llevó a cabo la implementación de dos artículos [2] y [3] para añadir una característica adicional al filtro de partículas inicial, extrayendo así más información acerca del objeto. Una característica que complementa la información de color a la perfección es la orientación del objeto, por lo que extraemos dicha información haciendo uso de histogramas de gradiente [4].

Este TFG también incluye una técnica novedosa [3] que saca provecho de la posición de las partículas generadas. Además de detectar el objeto, extrae información del área que le rodea, discriminando así entre lo que sería el objeto y lo que pertenecería al fondo de la imagen. Las partículas que proporcionen información acerca del objeto tendrán más importancia que aquellas que la proporcionen acerca del fondo.

Una vez tenemos el resultado del tracker se comparan con anotaciones manuales de la posición del objeto (ground-truth) mediante métricas de evaluacion, en la que la eficacia del algoritmo vendrá dada por el área de solape entre el resultado del tracker y el ground-truth.

Por último, se comparan los resultados arrojados por el filtro de partículas inicial, basado en histogramas de color; y el filtro de partículas modificado para extraer información adicional de la forma y orientación del objeto, así como de información de lo que rodea al objeto. Se podrá observar cierta mejoría en los resultados, ya que cuanta más información del objeto tenga el tracker, mejor podrá detectarlo y estimar futuras posiciones.

Palabras clave (castellano)

Seguimiento, filtro de partículas, multi-característica, Bhattacharyya, histogramas de color, histogramas de gradiente, fusión, ground-truth, adaptativo,

Abstract (English)

This final degree thesis has as main goal, the video-tracking of objects based in their color and shape features. This requires the use of a tracking algorithm, that locate the target over time, given his initial position, and estimates his later positions.

In this FDT is going to be use the classic particle filter based in their color features of the RGB space. The first step, was to make the current algorithm compatible with another color space, and HSV was chosen as alternative. Subsequently, took place the implementation of two papers [2] and [3] in order to add an aditional feature to the initial particle Filter, extracting more information about the target. One feature that complements color information very well is orientation, so we extract that information using gradient histograms [4].

This FDT includes a novel technique [3] that benefits from the position of the generated particles too. Apart from detecting the target, it extracts surrounding area information, discriminating against what corresponds to the target and what corresponds to the image background. Particles that provide information about the target will have mor importance that those that provide information about the background.

Once we have the result of the tracker, it will be compared with manual annotations about target's postition (ground-truth) through evaluation metrics, where effectiveness is provided by the overlapping area between the tracker result and the ground-truth.

Finally, the results thrown by the initial particle Filter, based in color histograms; and the particle filter modified to extract aditional information about target's shape and orientation, as well as its surrounding information, will be compared. It will be posible to see slight improvement in the results, because for the more information has the tracker about the target, the better the tracker will detect the target and estimate future postions.

Keywords (inglés)

Tracking, particle filter, mutifeature, Bhattacharya, color histogram, gradient histogram, fusion, ground-truth, adaptative

Agradecimientos

En primer lugar, agradecer a Juan Carlos que me haya acogido y guiado en este proyecto, he aprendido mucho acerca del video-tracking y los algoritmos de tracking y sin duda sin él no hubiera sido posible. También a todos los que hicieron posible que pueda entregar este trabajo al que tanto esfuerzo he dedicado.

En segundo lugar, agradecer todos aquellos que me acompañaron este tiempo y que hicieron de esta carrera una cuesta muy larga pero llena de momentos inolvidables, es imposible pensar en teleco y no acordarme de todos vosotros: Dimitri, Álvaro, Miki, Dani, Juanlu, Héctor, Hugo, Alex, Manu, Dass, Iván y Diego gracias por todo chavales.

¡A los que comprenden mi locura y comparten mí pasión y al Cholo Simeone, con ellos aprendí que, si se cree y se trabaja, se puede!

Y por último a mi familia y amigos que han confiado siempre en mí y me han apoyado cuando más negro veía las cosas.

A todos, gracias!

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

El seguimiento de objetos en una secuencia de video, conocido como video-tracking, es el proceso de estimar a través del tiempo la localización o posición de uno o más objetos de interés en dicha secuencia, como por ejemplo, el seguimiento de personas o de caras como se muestra en la figura 1.1.

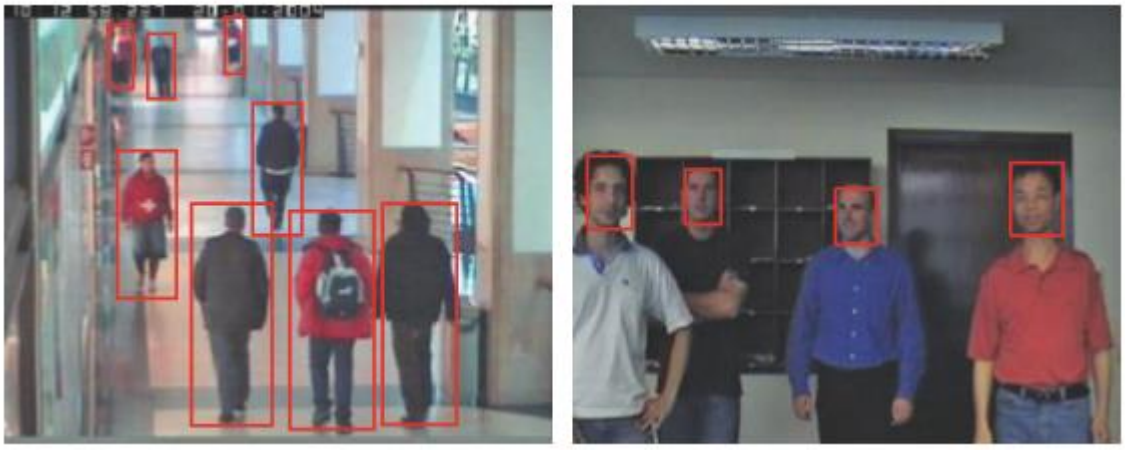


Figura 1.1: Ejemplos de tracking. En la imagen de la izquierda: detección y seguimiento de personas, y en la derecha: detección y seguimiento de caras [1].

Los avances en calidad y resolución de imágenes han hecho posible que el video-tracking sea utilizado en numerosas aplicaciones, como, por ejemplo, en el ámbito de la producción multimedia y realidad aumentada, en aplicaciones médicas e investigación psicológica, en vigilancia, en robótica y vehículos no tripulados, etc. Sin embargo, que esté tan extendido no quiere decir que sea una tarea sencilla de llevar a cabo; relacionar un objeto con su representación en determinados instantes de una secuencia puede llegar a ser una tarea muy complicada ya que, en ocasiones, intervienen factores ajenos al objeto, como cambios de iluminación, de escala, y de orientación; la presencia de elementos en la escena con caracteristicas similares a las del objeto que estamos siguiendo(Figura 1.2: apartado b)), o incluso oclusiones que hagan que el objeto quede ocluido totalmente o parcialmente (Figura 1.2: apartado a)).

a)

b)

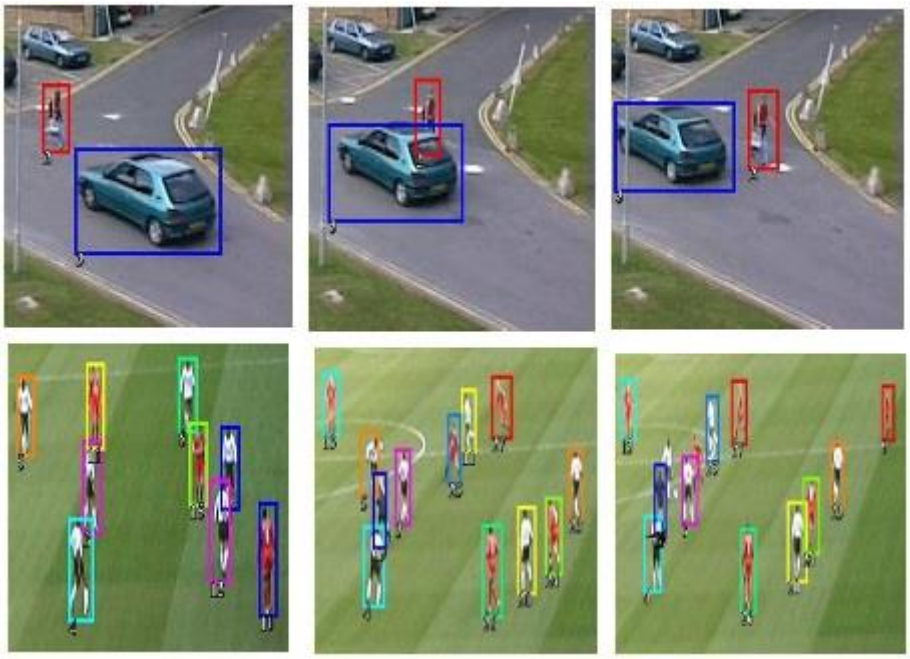


Figura 1.2: ejemplos de distintos problemas que podemos encontrarnos en seguimiento de objetos: a) siendo la persona con un recuadro rojo el objeto de nuestro tracker, tenemos un ejemplo de oclusión parcial producida por el coche. En la secuencia b) tenemos un ejemplo de clutter, donde seguir uno de los objetivos presentes en la escena, un jugador en concreto se hace muy complicado debido a la presencia de elementos muy similares, ya que hay jugadores incluso con la misma equipación [10]

Es por ello necesario implementar algoritmos que sean capaces de hacer frente a estos desafíos de una forma eficiente, sin perder el objeto independientemente de que intervengan o no los factores antes comentados.

1.2 Objetivos

El principal objetivo de este proyecto es el seguimiento de objetos en secuencias de video basándonos en diferentes caracteristicas: color y orientación. Para llevar a cabo esta tarea ha sido necesario el estudio previo de las bases del video-tracking, así como del algoritmo utilizado, el filtro de partículas.

Para alcanzar este objetivo, por tanto, estructuraremos el trabajo de la siguiente manera:

- Estudio del estado del arte: adquirir los conocimientos necesarios acerca del seguimiento de objetivos.
- Estudio del filtro de partículas: una vez conozcamos las bases del seguimiento de objetos nos centraremos en el estudio del filtro de partículas.

- Implementación: se llevará a cabo la implementación de los artículos [2] y [3] tras su estudio y se modificará un filtro de partículas dado según las directrices de dichos artículos.
- Evaluación: finalmente se evaluarán los resultados del filtro de partículas obtenido tras las modificaciones y se comparan los resultados con el filtro de partículas del que se partía.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 1. Introducción: Motivación, objetivos y organización de la memoria.
- Capítulo 2. Estado del arte: explicación de los fundamentos del video-tracking y del algoritmo conocido como filtro de partículas.
- Capítulo 3. Desarrollo: algoritmo de partida, implementación de métodos propuestos en los papers y problemas encontrados.
- Capítulo 4. Resultados: Resultados obtenidos y conclusiones: métricas, dataset, resultados.
- Capítulo 5. Trabajo futuro.

2 Estado del arte

2.1 Introducción

En este capítulo se explicará de forma resumida en que consisten principalmente el video-tracking y el filtro de partículas, tracker sobre el cual está basado este proyecto.

Para entender mejor el video-tracking, se hará una breve descripción del mismo. A continuación, se llevará a cabo la formulación del problema, trátese de un seguimiento de un único objeto o de varios, y de los distintos tipos de algoritmos que podemos encontrar. Este trabajo de fin de grado se centrará en seguimiento de un único objeto mediante un algoritmo automatizado. También se hablará de la extracción de caracteristicas: de bajo, medio y alto nivel; y de la representación de objetos, que pueden llevarse a cabo mediante la representación de forma o de apariencia.

Una vez se hayan explicado los fundamentos básicos del video-tracking, se explicará el funcionamiento del filtro de partículas. Tras una breve descripción del tracker, se explicará la fusión de caracteristicas y la forma de medir la fiabilidad de estas. Además, se añadirá una descripción del ground-truth y de las métricas de localización y de clasificación que pueden ser utilizadas, que ayudaran a entender las figuras que muestren resultados del tracker.

Además, al final de este trabajo puede encontrar una tabla que le facilitara información acerca de las variables que aparecerán a lo largo de todo el escrito.

2.2 Seguimiento de objetos

2.2.1 Introducción

El video tracking es el proceso de estimar a través del tiempo la localización o posición de uno o más objetos de interés. El video-tracking se va a enfrentar a diferentes desafíos, sin embargo, los mayores desafíos son aquellos relacionado con la similitud de apariencia entre el objeto (clutter) y los demás objetos de la escena, y el cambio de apariencia de el mismo (debidos a cambios de pose, iluminación cambiante, ruido, u oclusiones).

Tal y como se muestra en la figura 2.1, para diseñar un video-tracker se necesita, principalmente, definir el método de extracción de la información relevante, definir una representación para codificar la forma y apariencia de un objeto, definir un método de propagar el estado del objeto a través del tiempo, definir una estrategia para manejar objetivos que aparecen y desaparecen de la escena (track management), y la forma de extraer los metadatos del estado del objeto en una forma compacta.

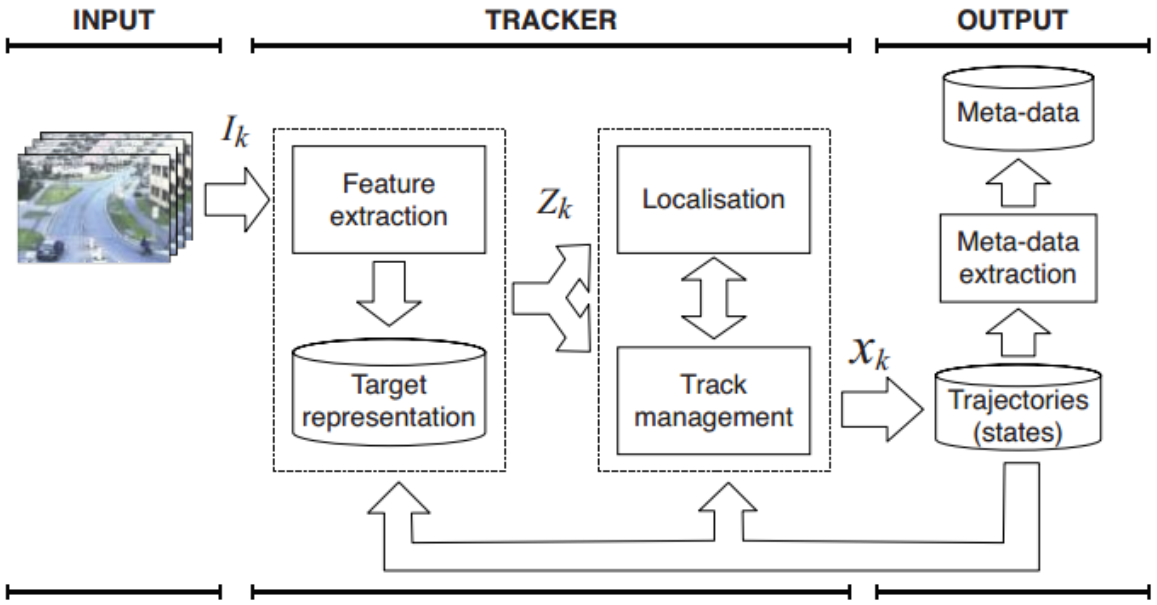


Figura 2.1: Etapas del video-tracking. [1]

2.2.2 Formulación del problema

Seguimiento de un único objeto: sea $I = \{I_k: k \in N\}$ que representa los frames de la secuencia de video, con $I_k \in E_I$ siendo la frame k, definida en E_I, el espacio de todas las posibles imágenes.

La estimación de una serie de tiempo seria: $x = \{x_k: k \in N\}$. Los vectores $x_k \in E_s$ son los estados del objeto, y E_s es el espacio de estados. x es conocido también como trayectoria del objeto en E_s. La observación que genera un objeto se encuentra codificada en $z_k \in E_0$ (espacio de características, resalta información relevante).
La información que podríamos encontrar en x_k podría ser, información sobre la localización del objeto y su forma, información de la apariencia del objeto, o información de la variación temporal de forma o apariencia.

EJEMPLO: $x_k = (u_k, v_k, h_k, w_k, \theta_k)$: estado con forma de elipse donde u y v indican la posición del centro, h la altura, w la anchura y θ la rotación en sentido de las agujas del reloj.

El seguimiento puede llevarse a cabo mediante tres clases diferentes de algoritmos de video tracking: manual, automatizado, e interactivo. Con algoritmo manual, el tracking es implementado directamente por el usuario, y se utiliza principalmente cuando hay mucha precisión, por ejemplo, en la definición de los bordes.

Con el algoritmo automatizado utilizamos información a priori sobre los objetos que se encuentran codificados por algún algoritmo. Se utiliza sobre todo para inicializar un tracker y\o apoyar el estado de estimación a lo largo del tiempo.

Los algoritmos interactivos son utilizados para compensar las carencias de los dos anteriores. Se precisa la interacción del usuario en determinadas fases del proceso de

tracking. Es utilizado en aplicaciones tag-and-track, cuando un operador inicializa manualmente un objeto de interés que entonces es seguido por el algoritmo de tracking.

2.2.3 Extracción de caracteristicas [1]

Características de bajo nivel (color, gradiente, movimiento):

- Color: CIELAB, CIELUV, RGB, YIQ, YUV, YCbCr, HSL, HSI.
- Invariantes de color fotométricas: Matiz y Saturación, RGB Normalizado, Modelo c1c2c3, Modelo l1l2l3.
- Gradiente y derivados: Operador Sobel, selección de escala, tratamiento de ruido, Laplacianos, movimiento.

Características de nivel medio (bordes, esquinas, regiones):

- Bordes.
- Puntos y regiones de interés: detector de esquinas Moravec, detector de esquinas Harris, detección a múltiples escalas.
- Regiones uniformes.

Características de alto nivel (objetos):

- Modelos de background: los algoritmos de sustracción de background usan la suposición de cámara estática para aprender un modelo de variación de intensidad de los pixeles del fondo (background). La estimación y actualización del modelo de background puede realizarse coleccionando estadísticas del valor del pixel a través del tiempo. Los métodos básicos basado en esta técnica tienden a fallar cuando la apariencia del fondo varia significativamente (nubes tapando el sol). Otro de los problemas viene dado por su inadecuación intrínseca a tratar interacciones entre objetos (proximidad y oclusiones).

- Modelos de objeto: modelan la apariencia de una clase predefinida de objetivos. Los enfoques más generales de la detección de objetos están basados en el entrenamiento de un conjunto de clasificadores que usan imágenes que contienen muestras representativas de la clase. Se dice entonces que un objeto ha sido detectado cuando se generan respuestas en una región de la frame a las características aprendidas.

2.2.4 Representación de objetos [1]

Representación de forma: la representación de la forma del objeto que se va a seguir, y siguiendo el esquema mostrado en la figura 2.2, puede llevarse a cabo de tres maneras diferentes:

- Modelos básicos: Aproximación de punto, aproximación de área (bounding-box), aproximación de volumen.
- Modelos articulados.
- Modelos deformables: para modelar objetos de los que no disponemos de información previa, y que pueden sufrir deformaciones que no serían bien modeladas por modelos básicos. Para modelar estos objetos tendríamos modelos de fluidos, contornos y modelos de distribución de puntos.

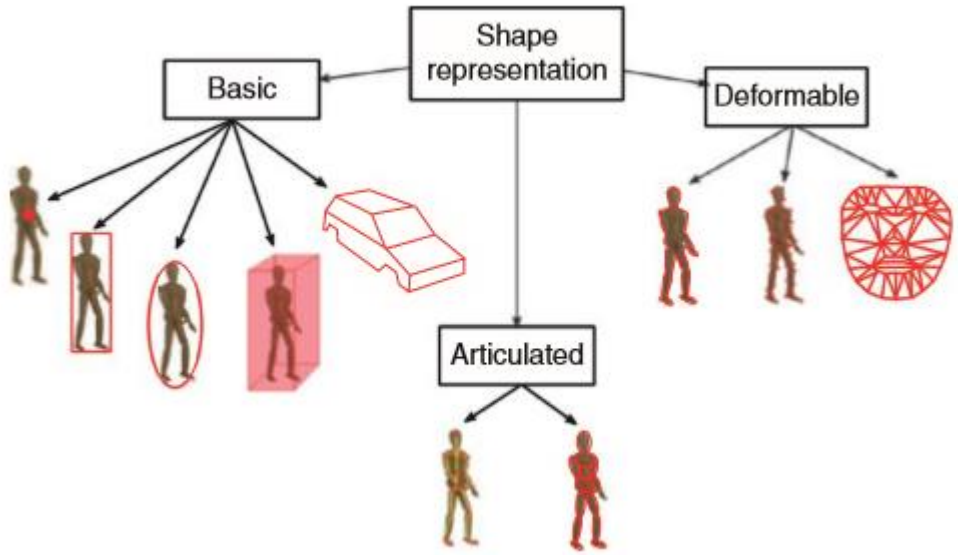


Figura 2.2: Ejemplos de representación de objetos. [1]

Representación de apariencia: modelo de la proyección esperada de la apariencia del objeto en el plano de la imagen. La representación de la apariencia del objeto puede ser específica, es decir, no tiene por qué ser generalizada para todos los objetos de la misma clase. Una representación de apariencia suele estar emparejada con una función, que estima la probabilidad de que un objeto este en un estado particular. Una de las características de estas funciones es su "suavidad" (smoothness) ante pequeñas variaciones de posición y forma del objeto. Además, facilita la labor del tracker de localización y reduce la probabilidad de que el estimador converja a soluciones menos óptimas. Podremos representar la apariencia mediante el uso de plantillas, histogramas (de color, orientación, o estructurales), haciendo frente a los cambios de apariencia (estrategias de actualización del modelo).

2.3 Filtro de Partículas [1]

2.3.1 Introducción

Un filtro de partículas, a grandes rasgos, es un algoritmo que mediante la generación de partículas nos va a permitir seguir un determinado objeto. A estas partículas se les van a ir otorgando "pesos", según los cuales se van a ir muestreando y actualizando dichas partículas, quedándonos solo con las que mejor ajustan nuestro algoritmo.

Y ¿cómo sabemos que las partículas ajustan adecuadamente nuestro algoritmo? Iremos comparando los resultados obtenidos con los resultados esperados.

Matemáticamente, siendo z_k los datos que sabemos, y x_k los estados que queremos estimar, suponemos que x_k depende de x_{k-1} y z_k depende solo de z_k, la recursión está totalmente determinada por la ecuación g_k:

$z_k = g_k (x_k, n_k) ; \ x_k = f_k (x_{k-1}, m_{k-1})$ (2.4)

Cuando las funciones g_k y f_k no son funciones de variación de tiempo lineales, podemos considerar una solución basada en la integración Monte Carlo.

Las densidades $p_{k|k} (x_k \mid z_{1:k})$ están aproximadas con una suma de L_k deltas de Dirac centradas en $\{ x_k^{(i)} \}^{Lk}$ de la forma:

$p_{k|k} (x_k \mid z_{1:k}) \approx \sum^{Lk} w_k^{(i)} \delta(x_k - x_k^{(i)})$ (2.5)

donde $w_k^{(i)}$ son los pesos asociados a cada partícula, quedando:

$$w_k^{(i)} \propto \frac{w_{k-1}^{(i)}}{a_{k-1}^{(i)}} \cdot \frac{g_k\left(z_k \middle| x_k^{(i)}\right) \cdot f_{k|k-1}\left(x_k^{(i)} \middle| x_{k-1}^{(i)}\right)}{q_k\left(x_k^{(i)} \middle| x_{k-1}^{(i)}, z_k\right)} \qquad (2.6)$$

donde $\{a_{k-1}^{(i)} \}^{Lk-1}$ es la función de remuestro: esta función define la probabilidad de cada partícula $x_k^{(i)}$ de generar una nueva en tiempo k; y $q_k (x_k^{(i)} \mid x_{k-1}^{(i)} , z_k)$ es la función de muestreo de importancia que implica que el peso $w_k^{(i)}$ dependa del estado anterior $x_{k-1}^{(i)}$.

La estimación del estado se realiza tomando la estimación máxima a posteriori (partícula con más peso) o calculando la esperanza de la partícula pesada con la siguiente ecuación:

$$E[x_k|z_{1:k}] \approx \frac{1}{L_k} \sum_{i=1}^{L_k} w_k^{(i)} \cdot x_k^{(i)} \qquad (2.7)$$

Un filtro de partículas puede tratar funciones de densidad multimodales y además recuperarse de oclusiones breves. Sin embargo, el resultado del tracking depende mucho de los parámetros elegidos, que dependen del contenido de la imagen. Es más, el número de partículas requeridas por el modelo de PDF subyacente incrementa exponencialmente con la dimensionalidad del espacio estado, aumentando los costes computacionales. La eficiencia del PF depende de hecho de la distribución de muestras en el espacio estado.

2.3.2 Fusión [1]

Cuando hablamos de fusión nos referimos al resultado de fusionar distintos algoritmos o de distintas características. Por eso es importante determinar su tipo y numero, su importancia relativa y el mecanismo de fusión de información. Para realizar esta fusión existen dos estrategias diferentes: fusión a nivel de tracker y fusión a nivel de medidas.

Fusión a nivel de tracker: se utiliza esta estrategia cuando por ejemplo se tienen múltiples algoritmos independientes de condensación en cada característica de la representación del objeto. Las salidas de los algoritmos independientes pueden usarse como observables de una cadena de Markov.

Fusión a nivel de medidas: esta estrategia se utiliza sin embargo cuando fusionamos múltiples características a nivel de medición, las medidas son combinadas internamente por el algoritmo de tracking. La fusión sigue un procedimiento que dé más importancia a los coeficientes estables. En un filtro de partículas puede usarse, por ejemplo, para fusionar múltiples características multimodales asumiendo independencia condicional de las características dado el estado. En esta aproximación la contribución de la característica se mantiene constante y la adaptividad recae en el muestreo que descarta las partículas con baja probabilidad. Si podemos asumir independencia inter-feature (entre características), la contribución de cada característica puede tomarse multiplicando las probabilidades y seleccionando pesos basados en la distancia entre tracking resultado de cada característica y el resultado de tracking global. Cada peso es usado como exponente de su probabilidad correspondiente.

2.3.2.1 Fusión de características en el Filtro de Partículas

Fusión de probabilidades (factorización no adaptativa de probabilidades): esta fusión se lleva a cabo en la etapa de actualización o *update*. El PF puede manejar múltiples caracteristicas N_f en el nivel de probabilidad. Suponiendo que queremos evaluar una probabilidad $g_{k,j}(z_k \mid x_k)$ en tiempo k por cada característica j ($j = 1 \ldots N_f$), una solución es comparar la probabilidad general como una combinación lineal de las probabilidades de caracteristicas únicas como:

$$g_k(z_k|x_k) = \sum_{j=1}^{Nf} \alpha_{k,j} \cdot g_{k,j}(z_k|x_k) \qquad (2.8)$$

Donde $\alpha_{k,j}$ es un coeficiente mezcla.

Re-muestreo de multi-caracteristicas: se realiza en la etapa de re-muestreo. Cuando se utiliza un muestreo sistemático multinomial (partículas muestreadas proporcionalmente a su peso): $a = w$; y por tanto $w_k^{(i)}$ seria proporcional a g_k. Esto quiere decir que los pesos son proporcionales a las probabilidades del vector de observación, y que en el caso de multi-característica, las partículas se dibujan proporcionalmente a los de la probabilidad mezclada. Como la evaluación de la fiabilidad de cada característica requiere un conjunto de partículas que precisamente representen todos los componentes dela mezcla definida, es apropiado introducir una estrategia de muestreo multi-característica como la siguiente:

$$a_k = \sum_{I=1}^{Nf} \beta_{k,j} \cdot g_{k,j}(z_k | x_k^{(i)}) \qquad (2.9)$$

Donde $\beta_{k,j} = \begin{cases} \alpha_{k,j} ; & \alpha_{k,j} > V/N_f \\ 0 ; & resto \end{cases}$.

El umbral V > 0 previene que todas las partículas sean dibujadas de la distribución de probabilidad de una única característica si los pesos se descompensan.

2.3.3 Fiabilidad de las características [1]

La fiabilidad de una característica cuantifica su habilidad o capacidad de representar un objeto basado en su apariencia tan bien como la capacidad de separar el objeto del fondo y de otros objetos.

Distancia a la media: la distancia a la media se basa en la diferencia entre el estado determinado por la partícula con máxima probabilidad de fusión y el promedio de probabilidades de una determinada característica sobre el conjunto de partículas.

Distancia al centroide: la fiabilidad de una característica puede también estimarse basándose en el nivel de concordancia entre cada característica y el resultado general del tracker. La contribución de cada característica es una función de la distancia euclídea $\tilde{E}_{k,j}$ entre en centro del mejor estado estimado de la característica *j* y el centro del estado obtenido combinando las características usadas en los resultados de fiabilidad en *k-1*.

Incertidumbre espacial: para computar la probabilidad como en la ecuación de fusión de probabilidades, estimamos la mezcla de coeficientes basada en la fiabilidad de cada característica. Pesamos la influencia de cada característica basada en su incertidumbre espacial. La incertidumbre espacial de una característica depende de la forma de su probabilidad, y para facilitar la tarea del estimador de estado, su probabilidad debería ser: suave, unimodal, es decir, que posea un único pico; e informativa alrededor del máximo, es decir, con superficies no planas alrededor del pico.

Sin embargo, su probabilidad puede presentar múltiples picos y en el peor de los casos, su máximo local puede estar cerca de la posición predicha del objeto. Este tipo de característica cuando se compare con otra que tenga máximos definidos, es más incierta espacialmente. Para estimar la incertidumbre espacial, analizaremos los valores propios de la matriz de covarianza $C_{k,j}$ de las partículas $x_k^{(i)}$ pesados por la probabilidad y computados para cada característica *j* en tiempo *k*. La incertidumbre queda finalmente relacionada con una hiper-elipse teniendo los valores propios como semi-ejes. Cuanto mayor sea el hiper-volumen, mayor la incertidumbre de la característica correspondiente al estado del objeto.

$$C_j = \begin{bmatrix} \frac{\sum_{i=1}^{L_k} l_j(u^{(i)}, v^{(i)})(u^{(i)} - \bar{u})^2}{\sum_{i=1}^{L_k} l_j(u^{(i)}, v^{(i)})} & \frac{\sum_{i=1}^{L_k} l_j(u^{(i)}, v^{(i)})(u^{(i)} - \bar{u})(v^{(i)} - \bar{v})}{\sum_{i=1}^{L_k} l_j(u^{(i)}, v^{(i)})} \\ \frac{\sum_{i=1}^{L_k} l_j(u^{(i)}, v^{(i)})(u^{(i)} - \bar{u})(v^{(i)} - \bar{v})}{\sum_{i=1}^{L_k} l_j(u^{(i)}, v^{(i)})} & \frac{\sum_{i=1}^{L_k} l_j(u^{(i)}, v^{(i)})(v^{(i)} - \bar{v})^2}{\sum_{i=1}^{L_k} l_j(u^{(i)}, v^{(i)})} \end{bmatrix} \quad (2.10)$$

Donde $x_k^{(i)}$ se representa mediante coordenadas de un espacio 2D $x = (u^{(i)}, v^{(i)})$, y l_j sustituye a $g_{k,j}(z_k | x_k)$. Ver tabla 2.1.

2.4 Ground-truth y métricas

El ground-truth representa el resultado de tracking esperado. Un ground-truth puede referirse a diferentes propiedades de exactitud de los resultados de tracking: la presencia de la proyección del objeto en la imagen en un pixel concreto en un tiempo determinado (ground-truth a nivel de pixel), el valor del estado del objeto o subconjunto de parámetros del estado en un tiempo concreto (ground-truth a nivel de estado). El ground-truth a nivel de pixel esta aproximado casi siempre por un cuadro delimitador o una elipse alrededor del área del objeto. En el contexto de computer visión, los datos de ground-truth incluyen un conjunto de imágenes y un conjunto de etiquetas, y un modelo definido para el reconocimiento del objeto. Las etiquetas son añadidas tanto manualmente como automáticamente mediante un análisis de la imagen, dependiendo de la complejidad del problema.

Para crear un dataset de ground-truth [9], deberíamos tener en cuenta las siguientes tareas principales:

- Diseño del modelo (Model Design): el modelo define la composición de los objetos, por ejemplo, fuerza, localización, etc. Debe ser coherente con la imagen y ajustarse al problema propuesto devolviendo resultados coherentes.
- Conjunto de entrenamiento (Training Set): se recopila y etiqueta un conjunto para trabajar con el modelo y que contiene imágenes y características tanto positivas como negativas (que generan falsas coincidencias).
- Conjunto de testeo (Test Set): se recopila un conjunto de imágenes para someterlas al Training Set, para verificar la exactitud del modelo y predecir los resultados correctos.
- Diseñar el clasificador (Classifier design): Esto se construye para cumplir con las metas de la aplicación de velocidad y precisión, incluyendo la organización de datos y optimizaciones de búsqueda para el modelo.
- Entrenar y probar (Training and Testing): Este trabajo se realiza utilizando varios conjuntos de imágenes para comprobar el ground-truth.

Los datasets del ground-truth [9] se podrían dividir en las siguientes categorías: **producidos sintéticamente** (mediante modelos generados por ordenador), **producidos de forma real** (secuencia de video o imágenes diseñada y producida), **seleccionados de forma real** (imágenes seleccionadas de fuentes existentes), **anotación automatizada**(se usan análisis de características y métodos de aprendizaje para extraer características), **anotados manualmente** (localización de características y de objetos hechas por un experto), y **combinados** (combinan categorías de las explicadas anteriormente).

Hay muchas variables que intervienen en la composición de un ground-truth, y una de esas variables son las métricas. Las métricas se definen para medir y registrar los resultados necesarios creando un dispositivo de prueba y ejecutando el algoritmo contra el dataset, como por ejemplo falsos positivos y falsos negativos [1]. Estas se pueden dividir en:

Métricas de localización:

Una forma sencilla de comparar objetivamente diferentes resultados de video-tracking es computar la distancia entre la estimación del estado y el estado de ground-truth de cada frame. Hay que tener especial cuidado cuando comparamos áreas de representación de errores o cuando estamos evaluando trackers multi-objetos.

- Resultados de un único objeto: una medida simple del error entre el estado estimado x y el estado de ground-truth $\tilde{x}$ es la distancia euclídea, $d_2(.)$, definida como:

$$d_2(x,\tilde{x}) = \sqrt{(x-\tilde{x})'(x-\tilde{x})} = \sqrt{\sum_{i=1}^{D}(x_i-\tilde{x}_i)^2}; \quad (2.11)$$

  Donde D es la dimensión del espacio estado E_s. Ver Tabla 2.1.

- Lidiar con el sesgo de perspectiva: el problema aparece cuando usamos métricas de distancia en estimaciones de estados posicionales como los producidos por trackers elípticos y basados en cuadros delimitadores. Como la puntuación del error del centroide no depende del tamaño de objeto, una estimación del error de unos cuantos pixeles de un error cercano a la cámara contara lo mismo que un error del mismo objeto situado alejado de la cámara. Sin embargo, cuando un objeto está alejado de la cámara su proyección en el plano de la imagen el relativamente más pequeño que cuando está cerca. Además, pequeños errores posicionales en el campo lejano pueden corresponder a estimaciones trackeadas que no comparten ningún pixel con el ground-truth. Para solucionar este problema, podemos computar el error del centroide normalizado por el tamaño del objeto. (Ver Tabla 2.1 por si necesita saber el significado de alguna variable).

$$x = (u, v, w, h, \theta); \quad (2.13)$$

$$\tilde{x} = (\tilde{u}, \tilde{v}, \tilde{w}, \tilde{h}, \tilde{\theta}); \text{ (ground-truth)} \quad (2.14)$$

$$e_u(x,\tilde{x}) = \frac{cos\tilde{\theta}(u-\tilde{u})-sin\tilde{\theta}(v-\tilde{v})}{\tilde{w}}; \qquad (2.15)$$

$$e_v(x,\tilde{x}) = \frac{sin\tilde{\theta}(u-\tilde{u})+cos\tilde{\theta}(v-\tilde{v})}{\tilde{h}}; \qquad (2.16)$$

$$e(x_k,\tilde{x}_k) = \sqrt{e_u(x_k,\tilde{x}_k)^2 + e_v(x_k,\tilde{x}_k)^2}; \qquad (2.16)$$

En la tabla 2.1 encontremos un ejemplo explicando los distintos valores de las ecuaciones anteriores.

Métricas de clasificación:

Otra aproximación para evaluar un tracker es valorar su actuación como un problema de clasificación. Considerando la avaluación de la actuación del tracker como un problema de clasificación, los resultados podrán ser valorados como verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN), o falsos negativos (FN). Un TP se define como un punto de ground-truth que está dentro del bounding-box de un objeto detectado y seguido por el algoritmo de tracking. Un FP es un objeto que se detecta y se sigue pero que no tiene correspondencia con ningún punto del ground-truth. TN se define como el conjunto de puntos en los que no hay detección del objeto ni ground-truth. Y FN son los puntos del ground-truth que no están dentro del bounding-box producido por el algoritmo de tracking. En la figura 2.3 se ve un ejemplo donde un vehículo no está siendo seguido correctamente, ya que el bounding box del tracker deja fuera el punto del ground-truth produciendo por ello un FN en el punto correspondiente al ground-truth, y un FP donde coloca el bounding-box. En la parte inferior de la figura 2.3 vemos TPs ya que el punto de ground-truth queda dentro del bounding-box.

Figura 2.3: ejemplos de falsos positivos FP, falsos negativos FN y verdaderos positivos TP. [11]

3 Desarrollo

3.1 Algoritmo inicial

El punto de partida de este proyecto es un filtro de partículas clásico basado en histogramas de color. Este algoritmo está dividido en cuatro etapas principales, que pueden identificarse fácilmente en la figura 3.1:

- Inicialización: en esta etapa se generan aleatoriamente un número determinado de partículas sobre el plano de la imagen. En nuestro caso utilizaremos 600 partículas.
- Predicción: en esta etapa se modificará el valor de las partículas añadiéndoles una pequeña perturbación que nos ayude a estimar mejor el siguiente estado del objeto.
- Actualización: en esta etapa se calcula el peso de cada partícula en función de la distancia al modelo de referencia. Aquellas que se asemejen más tendrán un peso mayor, y aquellas más distintas un peso menor.
- Estimación: en esta etapa se genera un conjunto nuevo de partículas que servirán de estimación a priori del estado de siguiente.

Después de la etapa de estimación y con el nuevo conjunto de partículas, se realiza la etapa de predicción, seguida de la de actualización, repitiéndose en bucle hasta que se acabe la secuencia de video analizada.

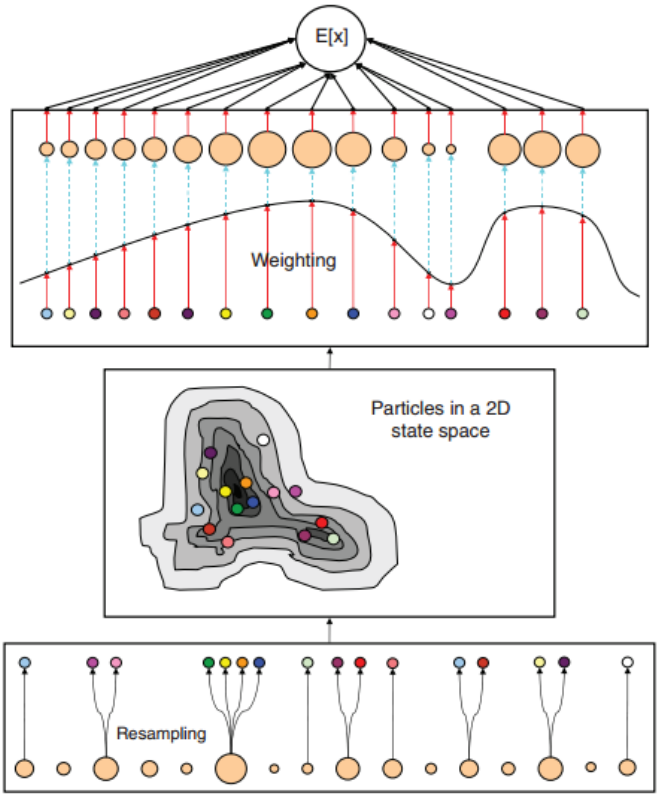


Figura 3.1: en esta figura vemos de abajo a arriba, que tras la estimación del estado anterior (ultima imagen), las partículas más parecidas al modelo generan una predicción del siguiente estado (imagen del medio), y después se actualizan los pesos de nuevo de acuerdo a la distancia con el modelo (imagen superior), generando por tanto otra estimación (E(x)).[1]

El código proporcionado lleva a cabo un seguimiento de objetos basado en sus características de color, mediante el uso de histogramas de color (figura 3.1). Estos histogramas se utilizan por su invariancia a la rotación y al escalado, su robustez a oclusiones parciales del objeto, y por la reducción de datos, lo que conlleva un cálculo más eficiente.

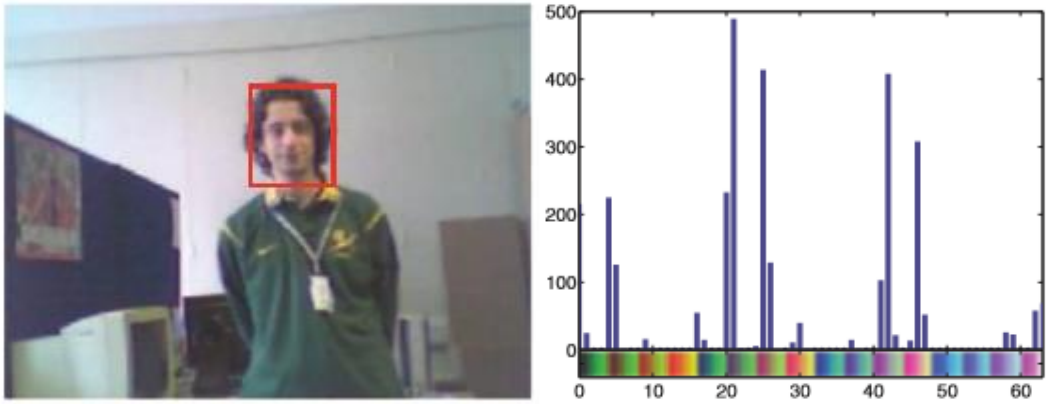


Figura 3.2: Ejemplo de histograma de color.[1]

3.2 Adaptative Multifeature Tracking in a Particle Filtering Framework

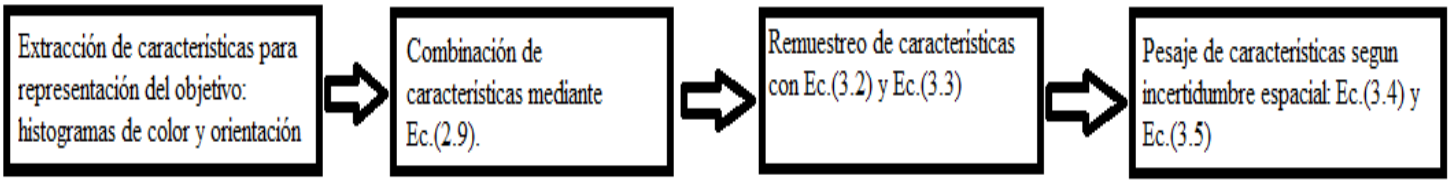


Figura 3.3: Diagrama de bloques del artículo "Adaptative Multifeature Tracking in a Particle Filtering Framework".

El título de esta sección se corresponde con el primer artículo a implementar [2] y su diagrama de bloques se corresponde con la figura 3.3. En él se propone combinar varias características en un solo filtro de partículas pesando sus contribuciones utilizando una medida de fiabilidad derivada de la distribución de las partículas en el espacio estado. Las características en las que se basan el algoritmo propuesto de este articulo son el color y la orientación. Por tanto, se añadirá a la característica de color que ya teníamos en nuestro algoritmo inicial, la de orientación, y al igual que hacíamos con la característica de color, la orientación vendrá determinada por histogramas de direcciones de gradiente que representaran la forma y los bordes internos del objeto [4].

La combinación de caracteristicas se llevará a cabo como ya se explicó en la sección 2.3.2.1 Fusion de caracteristicas en filtro de partículas. En este artículo, sin embargo, propone hallar la probabilidad $p(z_t|x)$ de cada una de las características, color y orientación, mediante la ecuación:

$$p_m(z_t|x) = e^{-(d(f(x), q) / \sigma * \sigma)} \qquad (3.1)$$

donde hallamos la distancia de Bhattacharyya entre f(x) (candidato) y q (el modelo). El valor de σ modela el ruido en las medidas de cada característica.

El remuestreo se llevará a cabo de forma muy similar también a como ya se explicó, añadiendo simplemente algunos matices, ya que con la función de remuestreo anterior, cuando el algoritmo degeneraba, la mayoría de las partículas se remuestreaban únicamente de una característica ignorando otros componentes de la probabilidad fusionada. Por ello se propone modificar la ecuación (2.9) para utilizar en su lugar las ecuaciones (3.2) y (3.3):

$$a_k^{(i)} = \sum_{j=1}^{Nf} \beta_{k,j} \cdot g_{k,j}(z_k | x_k^{(i)}) \qquad (3.2)$$

$$\beta_{k,j} = \begin{cases} \alpha_{k,j}\,; & \alpha_{k,j} > T \\ 0\,; & resto \end{cases} \qquad (3.3)$$

La ecuación (3.2) es muy similar a su predecesora, pero la (3.3) introduce un nuevo término "T", que define el límite inferior de partículas remuestreadas de cada característica.

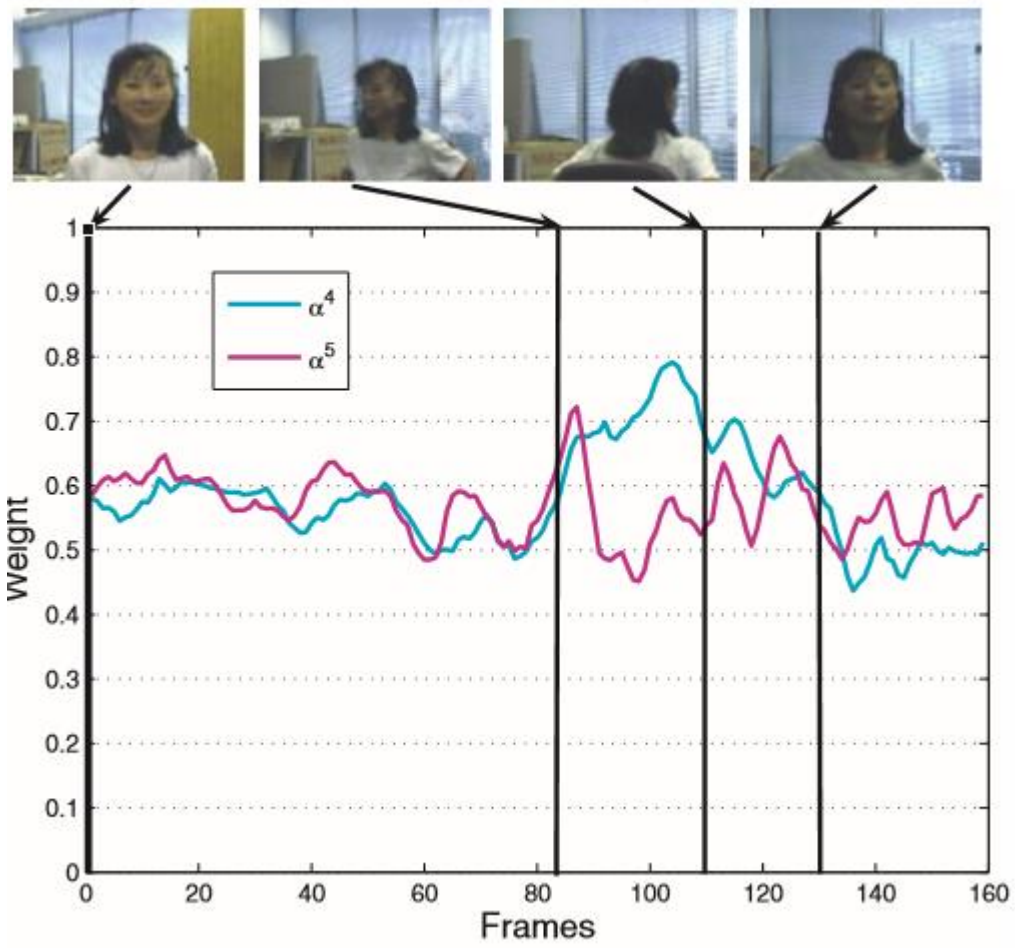


Figura 3.4: Ejemplo de pesos de los histogramas de orientación del filtro para la combinación adaptativa de características.[1]

Finalmente, los pesos de cada característica se asociarán según su incertidumbre espacial, de la forma que se explicó en la sección 2.3.3. Para calcular la fiabilidad de las características y partiendo de la matriz de covarianza de la sección antes mencionada, la incertidumbre espacial se halla según las ecuaciones (3.4) y (3.5):

$$U_{m,t} = \sqrt[D]{\prod_{k=0}^{D} \lambda_{m,t}^{(k)}} = \sqrt[D]{\det(C_{m,t})} \qquad (3.4)$$

$$\gamma_{m,t}^{4} = \frac{1}{U_{m,t}} \qquad (3.5)$$

Para suavizar variaciones temporales $\gamma^4_{m,t}$ pasa por un filtrado temporal:

$$\alpha^4_{m,t} = \iota\, \alpha_{m,t-1} + (1 - \iota)\, \gamma^4_{m,t}$$

donde "ι" determina la velocidad de actualización de $\alpha_{m,t}$, cuanto más bajo sea más rápido se actualizara. En la figura 3.4, se muestra un ejemplo de cómo serán los pesos asociados a la incertidumbre espacial calculada. También aparece $\alpha^5_{m,t}$, otra medida de incertidumbre espacial, que no usaremos en ningún momento.

El algoritmo del filtro de partículas adaptativo propuesto mejora la flexibilidad de la representación y al combinar histogramas de color y de gradiente genera mejores resultados en general que los algoritmos que utilizan una única característica. Además, este modelo puede extenderse a un mayor número de características.

3.3 Continuously Adaptive Data Fusion and Model Relearning for Particle Filter Tracking With Multiple Features

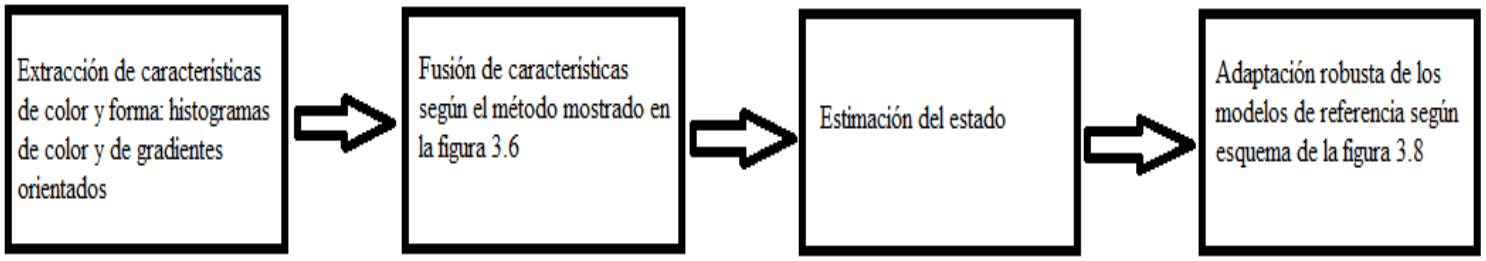


Figura 3.5: Diagrama de bloques del artículo "Continuously Adaptive Data Fusion and Model Relearning for Particle Filter Tracking With Multiple Features".

Como en el apartado anterior, el título da la sección también se corresponde con el del segundo paper [3] y cuyo diagrama de bloques podemos ver en la figura 3.5. Este artículo propone un nuevo método de tracking utilizando filtros de partículas, que proporcione una óptima habilidad de discriminar fondos variables. Además, se muestra como actualizar los modelos de cada característica mientras seguimos al objeto de forma continua y robusta. Tanto la fusión de probabilidades como los parámetros de actualización de los modelos se adaptan robustamente en cada frame extrayendo información contextual. Para implementar

este artículo será necesario implementar previamente el primer artículo, ya que se basa en un tracker que combina modelos de color y de forma.

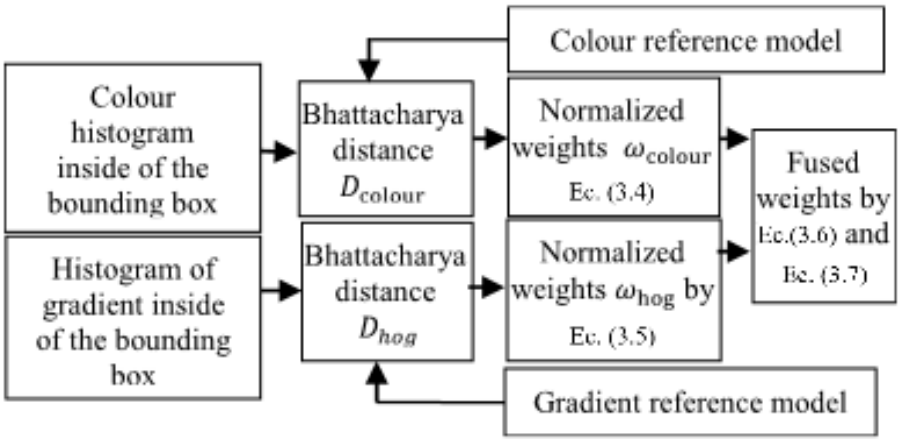


Figura 3.6: Diagrama de bloques del método de fusión de características. [3]

La fusión adaptativa online de múltiples modelos de características va a llevarse a cabo de forma similar a la comentada en el artículo anterior, pero siguiendo del esquema de la figura 3.6. Primero se hallarán las distancias, D_{color} y D_{hog}, entre modelo, H_{ref}, y candidato, H_{cand}, de cada una de las características, y posteriormente se hallarán los pesos asociados a estas de la siguiente forma, w_{color} y w_{hog}:

$$D_{color}\left(H_{ref}^{color}, H_{cand}^{color}\right) = \sum_{h=1}^{M} \sqrt{H_{ref}^{color}(\zeta) . H_{cand}^{color}(\zeta)} \qquad (3.2)$$

$$D_{hog}\left(H_{ref}^{hog}, H_{cand}^{hog}\right) = \sum_{h=1}^{M} \sqrt{H_{ref}^{hog}(\zeta) . H_{cand}^{hog}(\zeta)} \qquad (3.3)$$

$$\omega_{color}^{(i)} = \frac{1}{\sqrt{2\pi}\,\sigma_{color}} exp\left\{D_{color}^2 / 2\sigma_{color}^2\right\} \qquad (3.4)$$

$$\omega_{hog}^{(i)} = \frac{1}{\sqrt{2\pi}\,\sigma_{hog}} exp\left\{\frac{D_{hog}^2}{2\sigma_{hog}^2}\right\} \qquad (3.5)$$

Para dar más importancia a las partículas que representan mejor el objeto frente a las que no discriminan entre fondo y objeto, cogemos el valor máximo de los coeficientes de Bhattacharyya de las ecuaciones (3.2) y (3.3) y calculamos un factor global de pesaje μ_d:

$$\mu_d = \frac{\sigma_{color}^{\omega} D_{color}^{max}}{\sigma_{hog}^{\omega} D_{hog}^{max} + \sigma_{color}^{\omega} D_{color}^{max}} \qquad (3.6)$$

quedando la fusión de características de la forma:

(3.7)

$$\omega^{(i)} = \mu_d \omega^{(i)}_{color} + (1 - \mu_d)\omega^{(i)}_{hog}$$

usando información contextual para actualizar continuamente el factor de pesaje durante el tracking asegurando una óptima discriminación del modelo combinado de características.

Otra de las novedades de este artículo es la actualización robusta online de los modelos de referencia de cada característica mediante la introducción de información del fondo de la imagen. Esto se conseguirá ensanchando el bounding-box un factor τ (se sigue la recomendación del paper y se fijará en 1.2), de forma que queda como se muestra a continuación en la figura 3.7

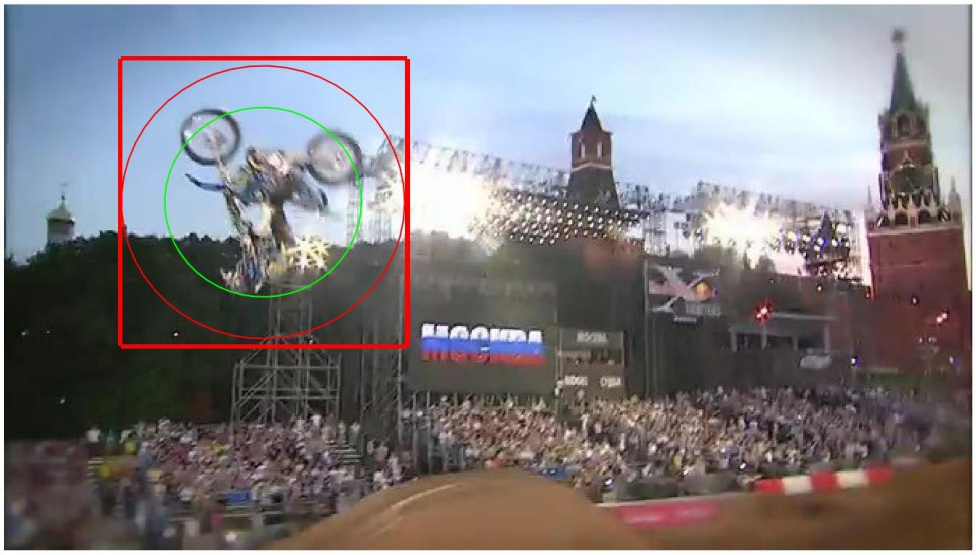


Figura 3.7: frame de la secuencia “Motocross_1”, donde se muestra en verde la elipse de ground-truth, y en rojo la elipse y el bounding-box ensanchado un factor τ.

Sobre este nuevo bounding-box (en rojo en figura 3.7), generamos un modelo de apariencia (histograma) para cada característica y hallamos el histograma del fondo, H_b, de la siguiente manera:

$$H_b(\zeta) = \frac{A_{f+b}H_{f+b}(\zeta) - A_f H_f(\zeta)}{A_{f+b} - A_f} \qquad (3.8)$$

Donde A_f es el área de la elipse del frente, y A_{f+b} el área de la elipse anteriormente comentada y mostrada gráficamente en la figura 3.7, que tomara informacion del frente y del fondo.

A continuación, se actualizarán los modelos de referencia anteriormente usados en las ecuaciones (3.2) y (3.3) siguiendo el esquema de la figura 3.8. Para ello, se define un peso de reaprendizaje, c_u (Ec. 3.9), que nos indicara cuan seguro es utilizar el primer plano para actualizar el modelo de referencia; cuanto mayor sea dicho valor, más seguro. Para actualizar el modelo de referencia habrá que aplicar las siguientes ecuaciones, una para cada característica:

$$c_u = 1 - e^{-\lambda_c (H_f(\zeta)/H_b(\zeta))} \tag{3.9}$$

$$\hat{H}_{ref}^{color}(\zeta) = \left(1 - c_u^{color}\right) H_{ref}^{color}(\zeta) + c_u^{color} H_f^{color}(\zeta) \tag{3.10}$$

$$\hat{H}_{ref}^{hog}(\zeta) = \left(1 - c_u^{hog}\right) H_{ref}^{hog}(\zeta) + c_u^{hog} H_f^{hog}(\zeta) \tag{3.11}$$

Donde λ_c es un parámetro de regulación, H_f el histograma del frente de la imagen, el foreground, H_b, el histograma del fondo de la imagen, el background, H_{color} y H_{hog} los histogramas de color y gradiente previamente calculados, y $\hat{H}_{ref}$ es el modelo actualizado de color y gradiente respectivamente.

El esquema para actualizar los modelos de referencia es mostrado en la figura 3.8.

Figura 3.8: Diagrama de bloques de reaprendizaje del modelo de referencia. [3]

4 Integración, pruebas y resultados

4.1 introducción

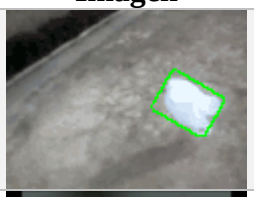
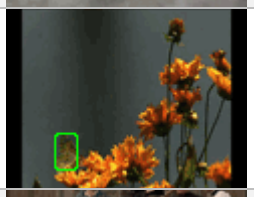
Tras la implementación de los papers recomendados en los que se recomendaba la fusión de probabilidades de color y gradiente, y una adaptación online de los modelos de referencia de ambas características se han obtenido mejores resultados en videos donde el objeto cambiaba la iluminación o hay clutter con el fondo.

Importante: A partir de este momento, nos referiremos al tracker del que partíamos como **Tracker 1** y el tracker que hemos obtenido tras las implementaciones anteriormente explicadas **Tracker 2**.

4.2 Dataset

Para la obtención de los resultados que se mostraran a continuación, se ha utilizado el dataset VOT de 2015. Este dataset cuenta con 60 secuencias, de las que se proporcionan las frames de cada secuencia y un archivo de ground-truth para cada una, cuyo resultado sería el recuadro verde que se puede ver en la figura 4.1 sobre cada objeto a seguir, especificado en la última columna de dicha figura.

Estas serían algunas de las secuencias que podemos encontrar:

Secuencia	Imagen	Descripción
bag		En esta secuencia el objeto al que se debe q seguir es una bolsa blanca.
butterfly		En esta secuencia el objeto al que se debe seguir es una mariposa
godfather		En esta escena se tiene que seguir el tocado blanco que lleva una persona.
motocross_1		En esta escena se debe seguir la moto que realiza acrobacias en el aire.
octopus		En esta escena se debe seguir un pulpo que esta sobre la arena.

pedestrian_2		En esta escena se debe seguir a la persona que va andando por la calle.
Soccer_1		En esta escena se debe seguir la cara de Steven Gerrard.
tiger		En esta escena el objeto a seguir es un tigre de juguete.

Figura 4.1: ejemplo de secuencias que podemos encontrar en el dataset VOT2015.

4.3 Métricas

Se utilizarán las métricas de clasificación para evaluar el rendimiento del tracker (sección 2.4). Al fin y al cabo, estas métricas nos sirven para calcular la proporción de solape existente entre el bounding-box del ground-truth y el del tracker, correspondiendo esta a la región que ocupan los TN sobre el conjunto de todos los positivos (TP + FP).

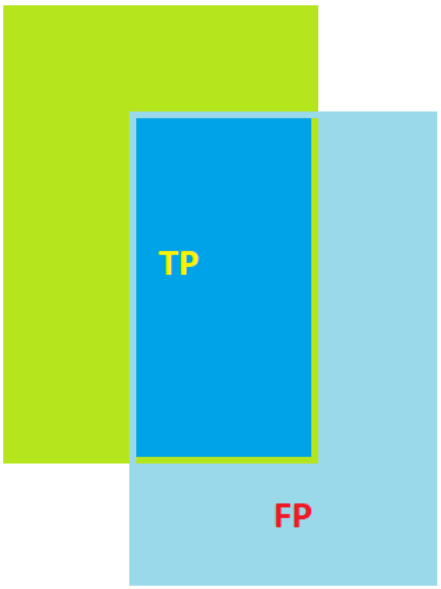


Figura 4.2: Representación gráfica del solape entre bounding-boxes. El área de color verde correspondería al ground-truth, y el área de color verde al tracker que utilicemos. El área que queda coloreada de azul oscuro corresponde al área de solape entre ambos.

Como puede apreciarse en la figura 4.2, el área que corresponde a los TP sería el área donde nuestro tracker acierta la posición del objeto. Para hallar la eficacia solo habría que calcular que proporción del área correspondiente al tracker ocupan los TP, para ello hacemos la operación antes comentada: TP/(TP + FP).

Hay que tener en cuenta que cuando los dos recuadros se separen completamente se producirá un reinicio del algoritmo que tras consultar en un template la posición del objetivo colocara ahí el estado estimado por el tracker, lo que producirá un máximo en la gráfica de solape, ya que el ground-truth y el resultado del tracker serán el mismo un instante.

4.4 Resultados

A continuación, se mostrarán diversos ejemplos donde se comparan los resultados de ambos trackers, describiendo si las condiciones de tracking son buenas o no y como actuaran el Tracker 1 y el Tracker 2 ante las dificultades que presente cada secuencia.

4.4.1 Comparación de resultados del Tracker 1 y Tracker 2: media y varianza.

Como resultados globales del tracking, se muestra a continuación en la tabla 4.1 una comparativa entre la media y varianza del solape que arrojan el tracker 1 y el tracker 2, y los proporcionados por VOT2015, utilizando un algoritmo de seguimiento DSST.

	Bag	Ball1	Ball2	Basket	Birds1	Birds2	Blanket	Bmx	Bolt1	Bolt2
V	0.09±0.1	0.1±0.08	0.1±0.13	0.1±0.05	0.05±0.04	0.1±0.3	0.14±0.15	0.15±0.2	0.15±0.09	0.1±0.08
T1	0.58±0.01	0.50±0.11	0.24±0.30	0.56±0.08	0.35±0.19	0.29±0.10	0.75±0.22	0.35±0.66	0.72±0.12	0.65±0.15
T2	0.60±0.02	0.41±0.10	0.23±0.24	0.60±0.08	0.39±0.20	0.32±0.11	0.74±0.22	0.35±0.68	0.52±0.18	0.61±0.21

	Bttfly	Car2	Crossg	Dino	Fer	Fish1	Fish2	Fish3	Godfat	Gym1
V	0.05±0.05	0.05±0.02	0.06±0.05	0.05±0.06	0.05±0.05	0.03±0.02	0.07±0.04	0.05±0.03	0.1±0.07	0.1±0.08
T1	0.63±0.31	0.57±0.03	0.54±0.11	0.62±0.31	0.45±0.15	0.58±0.13	0.41±0.15	0.48±0.13	0.45±0.10	0.68±0.09
T2	0.65±0.40	0.50±0.16	0.55±0.44	0.45±0.21	0.53±0.21	0.47±0.20	0.56±0.21	0.62±0.07	0.59±0.14	0.48±0.12

	Gym2	Gym3	Gym4	Hand	Icesk1	Icesk2	Leaves	March	matrix	Soccer1
V	0.09±0.08	0.12±0.1	0.15±0.2	0.05±0.17	0.05±0.09	0.09±0.1	0.05±0.09	0.03±0.05	0.04±0.06	0.02±0.04
T1	0.45±0.09	0.63±0.19	0.47±0.43	0.64±0.05	0.36±0.14	0.56±0.03	0.61±0.02	0.23±0.34	0.50±0.13	0.42±0.26
T2	0.63±0.24	0.47±0.55	0.65±0.08	0.46±0.20	0.49±0.05	0.68±0.02	0.19±0.29	0.49±0.12	0.42±0.24	0.47±0.09

Tabla 4.1: media y varianza para 30 secuencias de VOT2015, donde se comparan los resultados entre los resultados aproximados de VOT2015 (V), Tracker1 (T1), y el Tracker2 (T2), respectivamente.

4.4.2 Ejemplos de secuencias

A. Secuencia "Butterfly"

En esta secuencia se ve una mariposa de color amarillo volando junto a unas flores naranjas a velocidad reducida y sobre un fondo oscuro. Este es un ejemplo de detección y seguimiento favorable, ya que no vamos a tener problemas en separar el plano del fondo del plano frontal, y al no hacer movimientos muy rápidos y bruscos no perderemos el objetivo en ningún momento.

El resultado del tracker 1 podemos verlo en la figura 4.3:

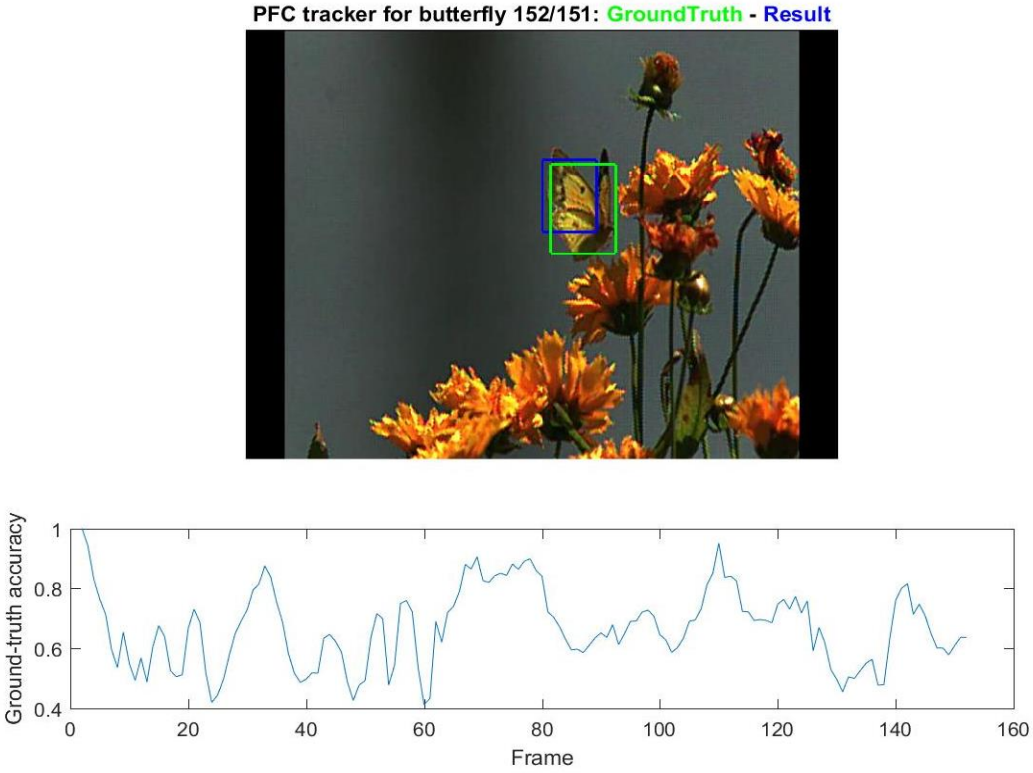


Figura 4.3: Resultados totales de similitud del resultado del tracker 1 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.65 ± 0.28.

Si añadimos las modificaciones mencionadas anteriormente al Tracker 1 obtendríamos el Tracker 2, mostrado en la figura 4.5, que arrojaría para este fichero de video los siguientes resultados:

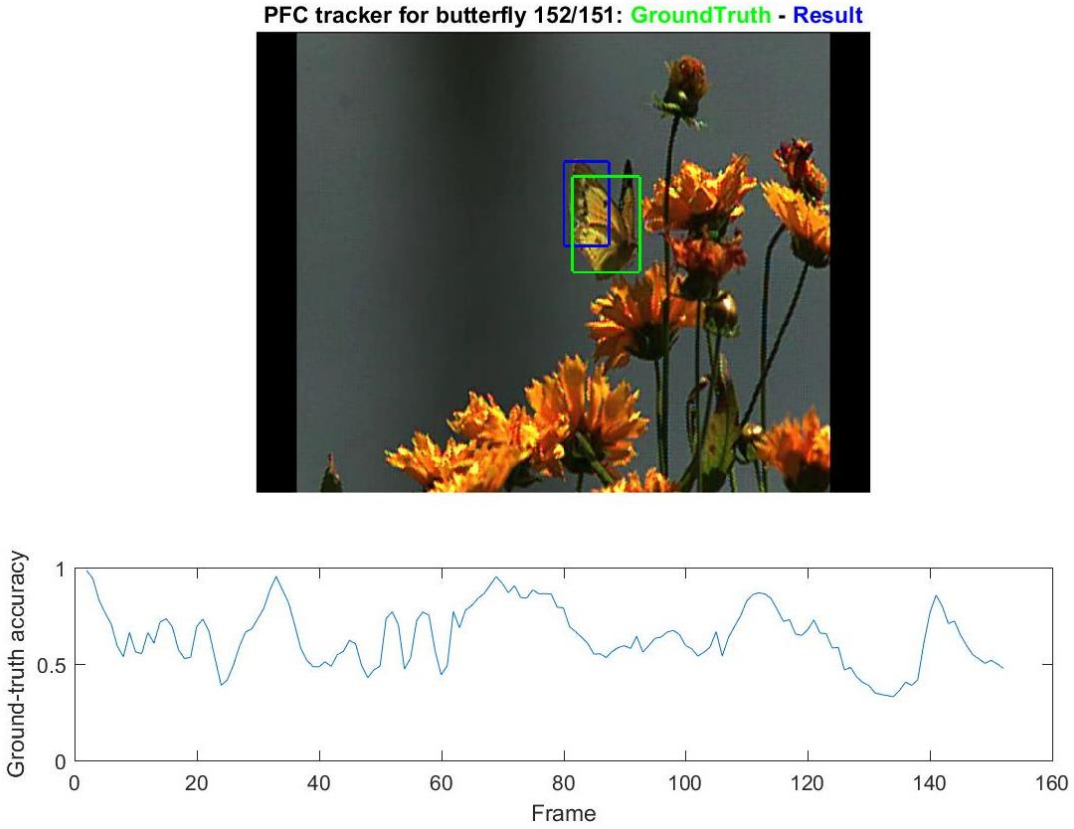


Figura 4.4: Resultados totales de similitud del resultado del tracker 2 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.65 ± 0.40.

Como se puede observar en las figuras 4.3 y 4.4, los resultados de ambos trackers son casi idénticos, pudiendo producirse variaciones debido a la característica aleatoria de las partículas.

B. *Secuencia "Soccer1"*

En esta secuencia se muestra a un jugador de fútbol celebrando la consecución de un título rodeado de sus compañeros de equipo y de otros elementos típicos de estas celebraciones, como el confeti que en un determinado acaba cubriendo casi al jugador por completo.

En esta secuencia el tracker se enfrenta a dos problemas fundamentales: la semejanza del objeto, Gerrard en este caso, con el resto de sus compañeros de equipo; y el confeti, que llega a ser muy abundante en una escena concreta de la secuencia y tapa la cara del Steven.

A continuación, se mostrarán los cómputos globales de esta secuencia por los dos trackers presentados, en las figuras 4.5 y 4.6 respectivamente.

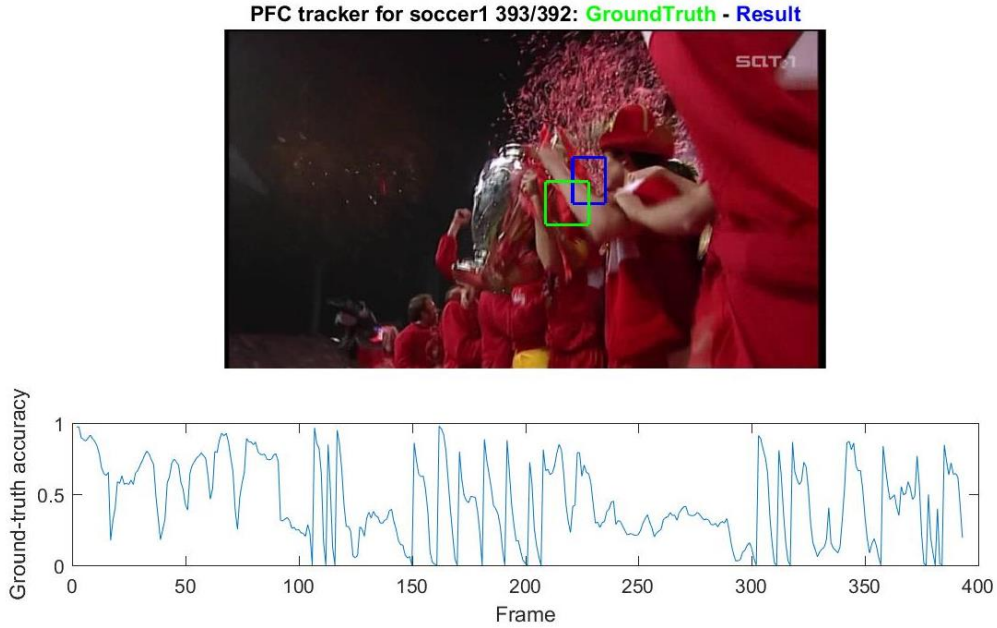


Figura: 4.5: Resultados totales de similitud del resultado del tracker 1 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.42± 0.26.

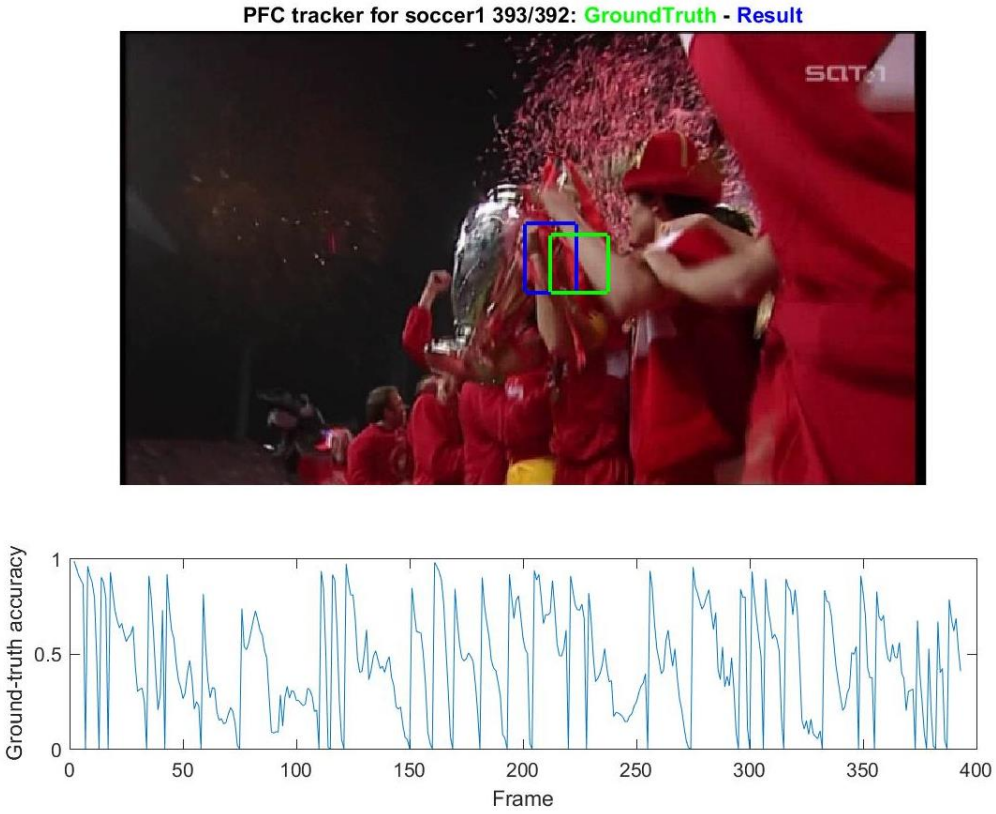


Figura: 4.6: Resultados totales de similitud del resultado del tracker 2 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.47± 0.09.

Unas secuencias que presenta tantos picos, nos indica que el bounding-box resultado del tracker se ha separado muchas veces del ground-truth, es decir, que ha perdido de vista el objetivo numerosas veces. En esta secuencia tiene sentido que aparezca este resultado ya que es una secuencia en la que el objeto se pierde de vista mucho hasta para el ojo humano.

C. *Secuencia "Iceskater2"*

En esta secuencia se verá a una pareja de patinadores en una competición. El objeto del seguimiento será la chica, que quedará ocluida por el chico a la hora de realizar algunas piruetas. En esta secuencia nos encontraremos con cambios en la forma, orientación y escala del objeto a seguir, desafíos que un filtro de partículas clásico debería solucionar sin demasiados problemas. Sin embargo, el color de la valla publicitaria y la presencia de otra persona van a hacer que el tracker 1 sufra alguna que otra confusión cuando la pareja se separa. Justo para eso se ha diseñado el tracker 2, y es lo que se ve cuando comparamos los resultados globales de solape en las figuras 4.7 y 4.8 respectivamente, el tracker 2 arroja mejores resultados.

La mejoría puede verse tanto gráficamente, como vemos en dichas figuras, como numéricamente como se observa en la tabla 4.1.

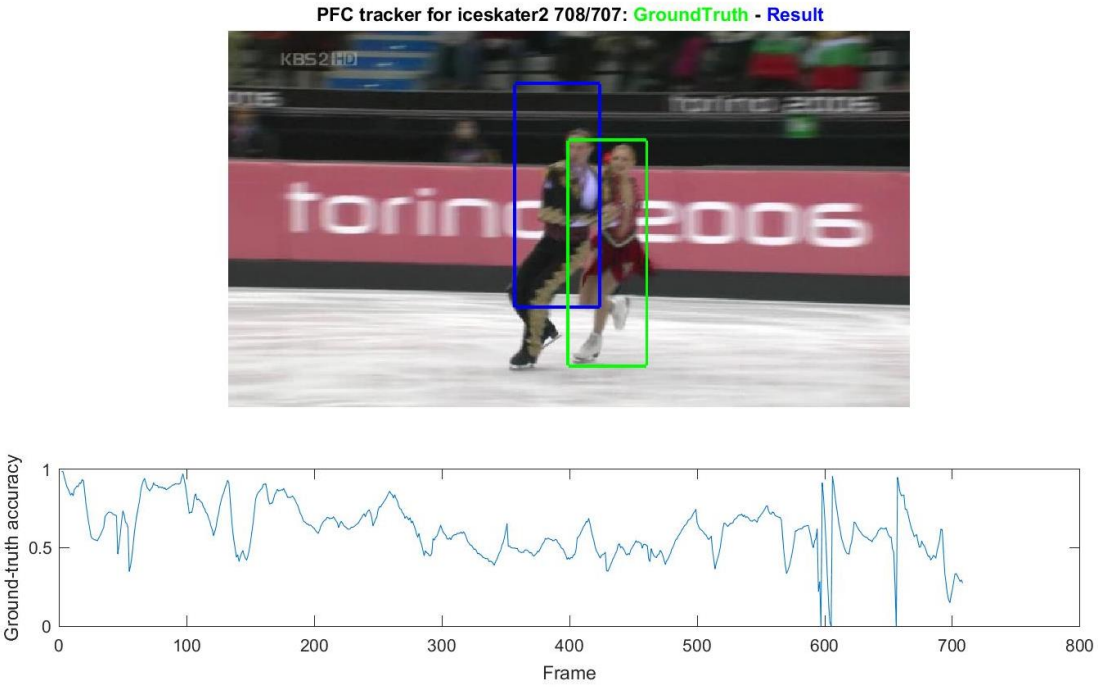


Figura: 4.7: Resultados totales de similitud del resultado del tracker 1 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.68 ± 0.02.

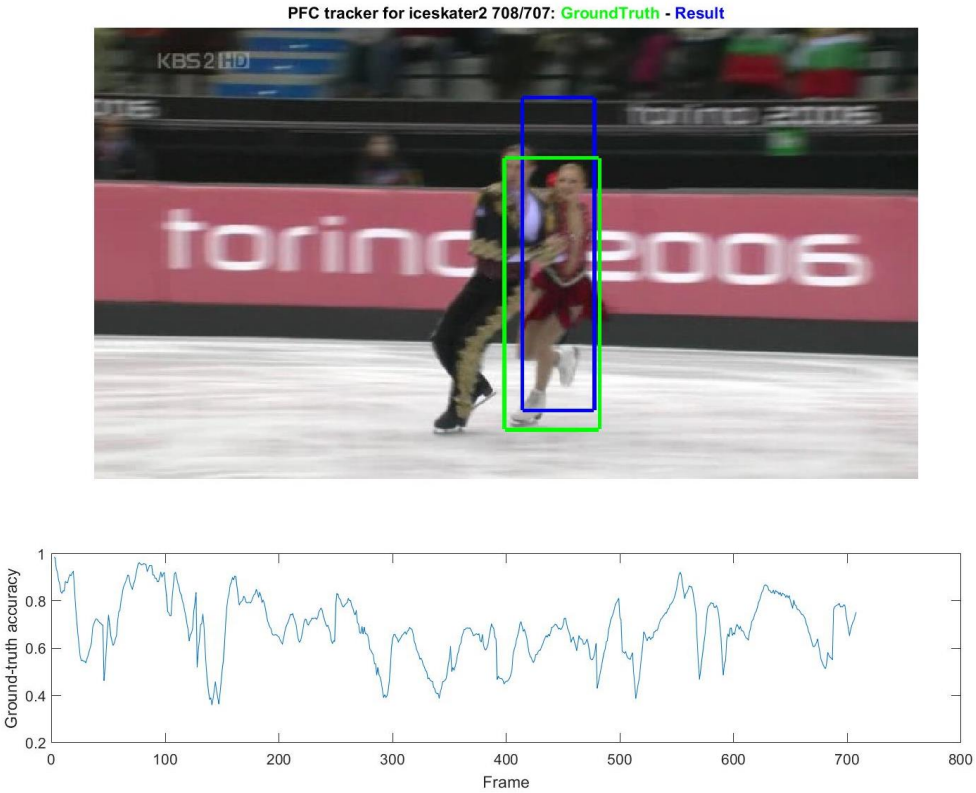


Figura: 4.8: Resultados totales de similitud del resultado del tracker 2 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.68 ± 0.02.

D. *Secuencia "Fish3"*

En esta secuencia el objeto de seguimiento es un pez de color amarillo nadando en un arrecife de coral de colore vivos y rodeado de más peces que podrían provocar confusiones en el tracker. En esta secuencia tenemos un pequeño factor de clutter, ya que algunos corales tienen un color similar al pez, y tenemos alguna oclusión parcial que pueden crear problemas.

En este caso el resultado del tracker 2 es también superior al del tracker 1, teniendo este último que reiniciarse numerosas veces a partir de la mitad de la secuencia como se muestra en la figura 4.9. El tracker 2, sin embargo, no pierde nuca de vista el objeto y lo mantiene en todo momento localizado arrojando una proporción de solape de 0.62 de media como puede consultarse en la tabla 4.1, su resultado grafico se ve en la figura 4.10.

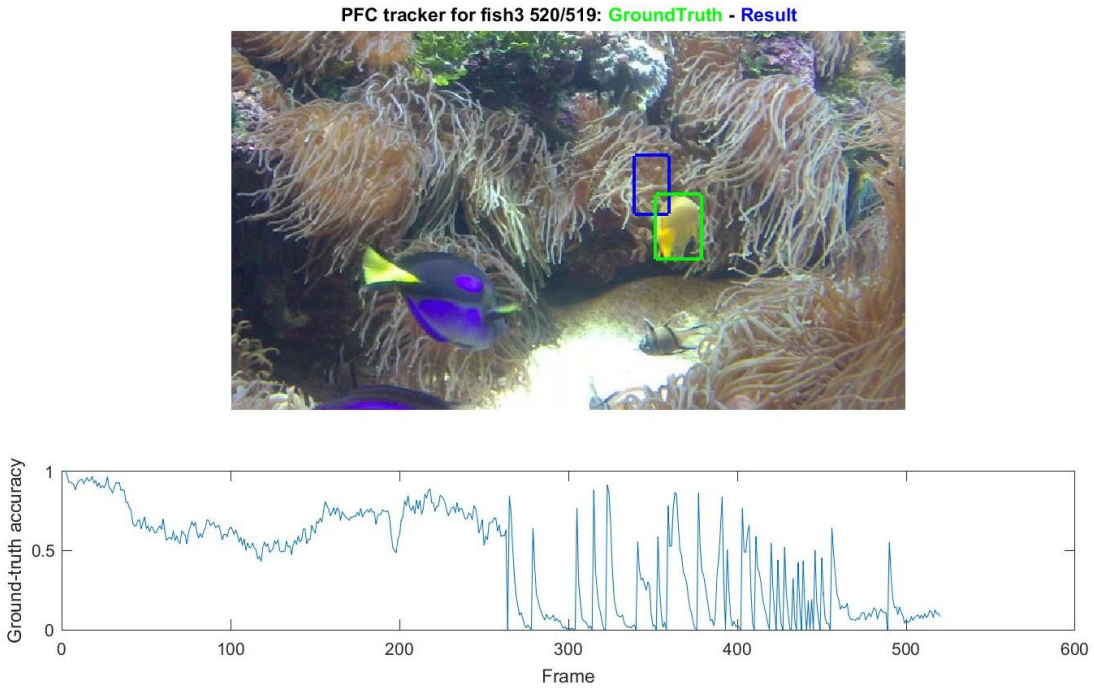


Figura: 4.9: Resultados totales de similitud del resultado del tracker 1 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de .

Tal y como comentábamos antes, a partir de la mitad de la secuencia, el tracker 1 pierde constantemente de vista el objeto de seguimiento, provocando esa forma característica que vemos en la gráfica de solape situada bajo la secuencia.

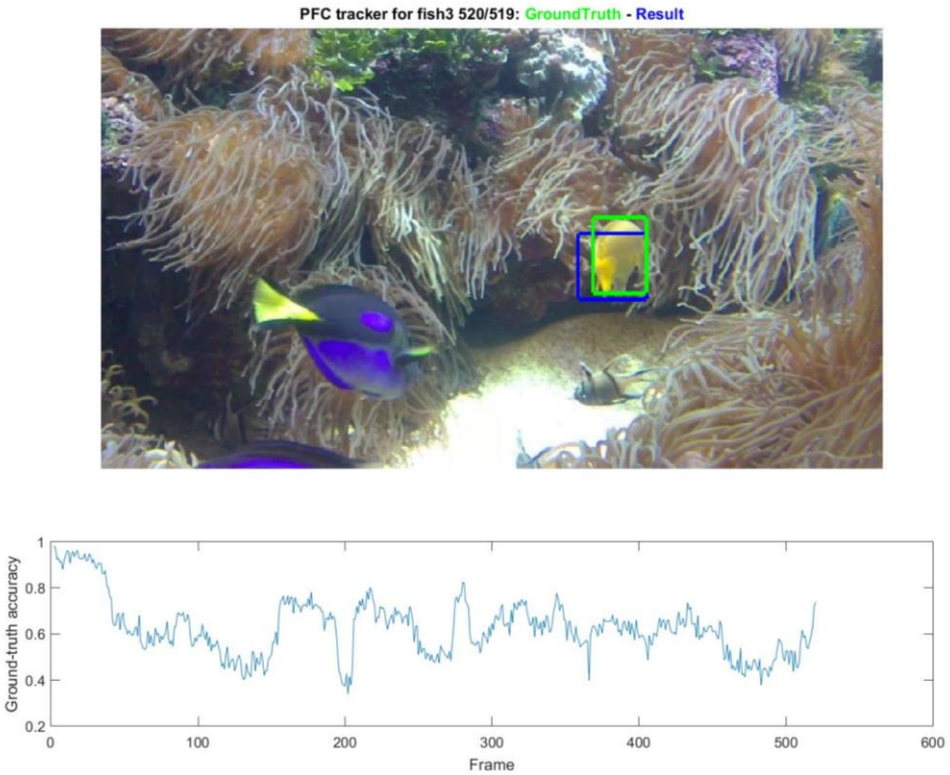


Figura: 4.10: Resultados totales de similitud del resultado del tracker 2 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.62±0.07.

En esta imagen, sin embargo, el tracker no pierda de vista el objeto y no tiene la necesidad de reiniciarse, el resultado es mejor tanto gráficamente como numéricamente.

E. *Secuencia "Matrix"*

En esta secuencia el objeto de seguimiento es la cara de uno de los personajes de la película cinematográfica *Matrix.* En esta secuencia, los principales problemas son el movimiento del objeto y su semejanza con los rostros de las personas que hay en el background. Para esta secuencia los dos trackers muestran resultados semejantes como se aprecian en las figuras 4.11 y 4.12 donde se ven los resultados globales de solape de ambos trackers respectivamente.

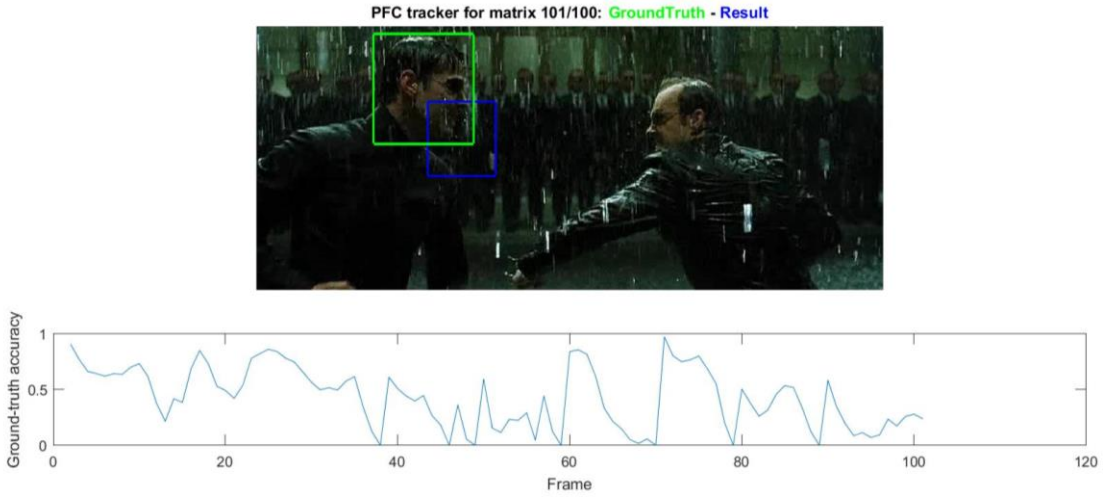


Figura: 4.11: Resultados totales de similitud del resultado del tracker 1 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.50±0.13.

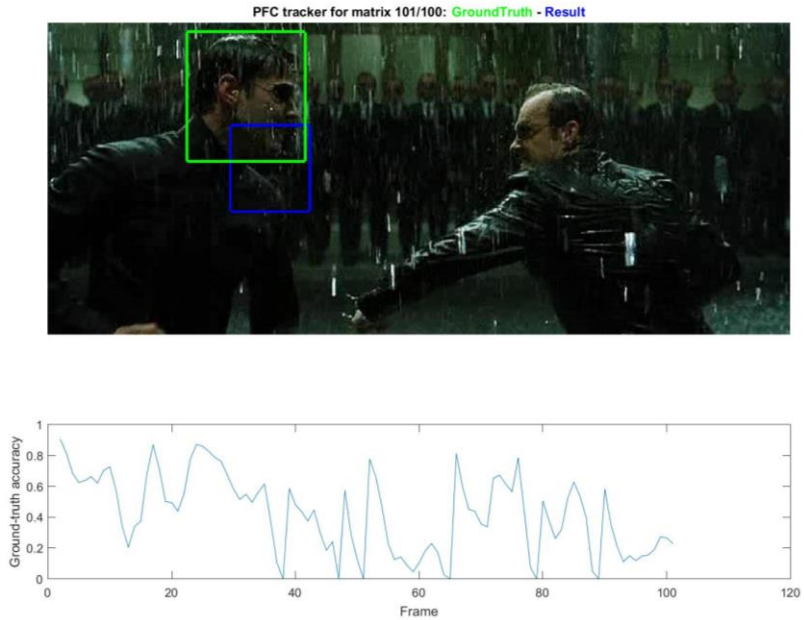


Figura: 4.12: Resultados totales de similitud del resultado del tracker 2 con el ground-truth. El recuadro verde corresponde al ground-truth, y el azul corresponde al resultado de nuestro tracker. En la gráfica de abajo podemos ver el porcentaje normalizado de solape entre ambos, siendo igual a 1 cuando el solape es máximo y 0 cuando no existe solape entre los recuadros. En este caso la media y varianza del solape es de 0.42±0.24.

Prácticamente se reinician el mismo número de veces incluso, por eso los resultados son tan similares.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

El objetivo de este trabajo de fin de grado era incorporar al filtro de partículas basado en características de color, otros espacios de color además del RGB sobre el que estaba implementado, como por ejemplo el HSV; pero sobre todo la extracción de información de información de la orientación del objetivo y de información que nos ayude a discriminar entre el plano frontal y el plano del fondo.

La información de la orientación del objetivo se obtiene gracias a los histogramas de gradiente [4], y complementa muy bien a la información de color cuando se producen cambios en la iluminación de la escena o cuando hay clutter con el fondo. Aunque la mejora no es abismal, sí que es apreciable en los resultados del Tracker 2, cuando incluíamos en el cálculo de los pesos de las partículas la información proporcionada por los ya mencionados histogramas de gradiente.

Otra de las medidas que nos han ayudado a mejorar los resultados es la que nos ayuda a discriminar entre objetivo y el fondo, mediante el uso de un bounding-box ensanchado. Al hacerlo para cada una de las características la información es mayor y por tanto los resultados mejoran también, aunque como ya he mencionado anteriormente no de manera muy exagerada.

Para ver de forma más clara esta mejoría sería necesario ejecutar el algoritmo un número de veces elevado y hacer un promedio de los datos obtenidos, aunque esto conlleva un elevado coste computacional, y es necesario ejecutar varios procesos en paralelo con el fin de rebajar dicho coste.

5.2 Trabajo futuro

Considerando que las técnicas explicadas anteriormente y aplicadas al tracker inicial han supuesto una mejora apreciable, el objetivo del trabajo futuro será dar con la combinación correcta de los parámetros de tracking, como pueden ser las medidas de ruido de cada una de las caracteristicas, el valor con el que ensanchamos el bounding-box del tracker, o el parámetro de regulación de reaprendizaje de pesos que se llevaba a cabo en la sección 3.3.3; con la intención de hacer más notoria la diferencia entre ambos trackers.

Este objetivo también podemos alcanzarlo introduciendo más características que puedan darnos información adicional del objetivo o mediante la fusión de distintos trackers que nos den más robustez frente a oclusiones, cambios de escala o deformaciones del objetivo.

Referencias

[1] E. Maggio, and A. Cavallaro, “Video Tracking: Theory and Practice”, John Wiley & Sons, Ltd, 2011.

[2] E. Maggio, F. Smerladi, and A. Cavallaro, “Adaptative Multifeature Tracking in a Particle Filtering Framework”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, No. 10, October 2007.

[3] J. Xiao, R. Stolkin, M. Oussalah, and A. Leonardis, “Continuously Adaptive Data Fusion and Model Relearning for Particle Filter Tracking With Multiple Features”, IEEE Sensors Journal, vol. 16, April 15, 2016.

[4] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, 'Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection,' In: 12th International IEEE Conference On Intelligent Transportation Systems, 2009, St. Louis, 2009. V. 1. P. 432-437.

[5] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking.” IEEE Transactions on Signal Processing. 50 (2). p 174—188

[6] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, Antonio Torralba. "HOGgles: Visualizing Object Detection Features" International Conference on Computer Vision (ICCV), Sydney, Australia, December 2013.

[7] Mohamed Bécha Kaâniche, “Tracking HoG Descriptors for Gesture Recognition”, IEEE Member and François Brémond INRIA Sophia Antipolis - Mediterranean Research Center – PULSAR Project 2004

[8] P. Fieguth, and D. Terzopoulos, “Color-based tracking of heads and other mobile objects at video frame rates”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 21–27, 1997

[9] Scott Krig, “Computer Vision Metrics: Survey, taxonomy, and analisys”, Apress Open, 2014.
[10] M. Souded, and F. Brémond, “An Object Tracking in Particle Filtering and Data Association Framework, Using SIFT Features”, in: International Conference on Imaging for Crime Detection and Prevention (ICDP), London, United Kingdom, November 2011.
[11] J. Black, T. Ellis, and P.L. Rosin, “A novel method for video tracking performance evaluation”, Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS), 2003.

Glosario

TFG	Trabajo de Fin de Grado
FDT	Final Degree Thesis
PDF	Probability density function
PF	Particle Filter
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
DSST	Discriminative Scale Space Tracker

Anexo 1: Tabla de referencia de símbolos

En la siguiente tabla se indicarán los distintos símbolos que aparecen en la memoria y su significado, con el fin de hacer más fácil la comprensión al lector:

Símbolo	Significado
$\mathbf{E_i}$	Espacio de imágenes: espacio de las frames de una secuencia de video.
$\mathbf{E_s}$	Espacio de estados: espacio de los estados del objeto.
$\mathbf{E_o}$	Espacio de características: espacio de las observaciones de un objeto.
$\mathbf{x_k}$	Estado de un objeto en el instante de tiempo "k".
$\mathbf{x_k = (u_k, v_k, h_k, w_k, \theta_k)}$	Estado con forma de elipse donde u y v indican la posición del centro, h la altura, w la anchura y θ la rotación en sentido de las agujas del reloj.
$\tilde{x}$	Estado del ground-truth.
$\mathbf{z_k}$	Medidas del objeto en el instante de tiempo "k".
$\mathbf{w_k^{(i)}}$	Peso asociado cada partícula "i" en el instante "k".
Índices "k" y "t"	Tiempo.
Índice "i"	Numero de partícula.
Índices "j" y "m"	Característica, se omite si solo hay una.
$\mathbf{q_k}$	Función de muestreo de importancia.
$\mathbf{q_k (x_k^{(i)} \mid x_{k-1}^{(i)} , z_k)}$	Función de muestreo de importancia que implica que el peso $w_k^{(i)}$ dependa del estado anterior $x_{k-1}^{(i)}$.
$\mathbf{g_k}$	Probabilidad.
H	Histograma
$\mathbf{H_{color}}$	Histograma de color.
$\mathbf{H_{hog}}$	Histogramas de gradiente.
$\mathbf{D_{color}}$	Distancia entre histogramas de color.
$\mathbf{D_{hog}}$	Distancia entre histogramas de gradiente.
$\mathbf{w_{color}}$	Pesos asociados a la característica de color.
$\mathbf{w_{hog}}$	Pesos asociados a la característica de orientación/gradiente.
$\mathbf{\sigma_{color}}$	Modela el ruido de la característica de color
$\mathbf{\sigma_{hog}}$	Modela el ruido de la característica de gradiente.
A	Área.
Índice "f"	Indica pertenencia al *foreground.*
Índice "b"	Indica pertenencia al *background.*
Índice "f+b"	Indica pertenencia a un área formada por *foreground* y *background.*

Tabla 2.1: Símbolos y sus significados.