Hybrid Particle Filter with Color Histogram for Enhanced Robustness in Object Tracking

Lassaad Ayadi and Hatem Ghodbane
Mohamed Khider University, Biskra, Algeria
E-mail: lassaad.ayadi@univ-biskra.dz, h.ghodbane@univ-biskra.dz

Keywords: Tracking, detection, particle filter, color histogram, occlusion, image sequence

Received: July 20, 2025

In this paper, we describe a hybrid method for detecting and tracking moving objects in image sequences. Particle filters and color histograms are combined in the proposed method to address issues with occlusions, lighting variations, and object appearance. The goal of integrating these two techniques is to improve tracking robustness. Experiments conducted in the OTB 2013 and OTB 2015 databases show that our method, called PFHist, outperforms several existing trackers. It achieves success up to 80% in terms of overlap rate and 94% accuracy in terms of center location error, especially in cases of partial or total occlusions. Moreover, the RGB color space has been shown to be more efficient than the HSV space, and the use of a reduced number of particles (100) allows for better performance while reducing computational cost. Future work will focus on the automatic selection of the optimal color space and on extending the method to multi-object tracking.

Povzetek: Članek predstavi PFHist, hibridni sledilnik, ki združi delčne filtre in barvne histograme za robustno zaznavanje in sledenje premikajočih se objektov ob zakrivanju in spremembah osvetlitve.

1 Introduction

Object tracking is a difficult problem that arises in a large number of computer vision and image processing applications. This difficulty is accentuated in environments without constraints where the tracking system must adapt to the significant variability of objects, variations in brightness, partial or total occlusion, noise caused by the acquisition system, and motion detection issues.

Object tracking relies on the invariance properties of objects of interest. Invariance can concern the geometry of the scene or objects, the appearance of objects (i.e., photometry or color), or even kinematics (e.g., spatio-temporal constraints). The method used to track an object is strongly linked to the representation of that object and the features used to track it. Thus, the choice of the representation of the object and the characteristics depends on the context, the nature of the object, and the information that we want to obtain from the tracking.

The first object tracking algorithms were applied in the military field as soon as radar appeared. Since then, it has become at the heart of many applications such as video surveillance, gesture recognition, definition of visual interfaces, virtual reality, image compression, analysis of human or animal movement, robotics, etc. This requires increasingly efficient algorithms.

However, given the complexities that can be encountered, the performance of algorithms developed for object tracking remains limited. It should be noted that the majority of these algorithms are based on the Bayesian formalism; the latter was used to remedy the different complexities linked to the problem considered, such as errors or noise in measurements and the occlusion of objects compared to the positioning of the sensors.

In this domain, particle filtering tracking methods [1] [2] also called the sequential Monte Carlo method, have proven their validity. Particle filtering is a Bayesian approach in which the probability of the configuration of an object (its position, its scale), taking into account observations, is represented by a set of weighted samples called particles. This representation makes it possible to maintain several tracking hypotheses simultaneously, unlike algorithms based on a single representation, seeking to maximize a criterion using iterative optimization methods, and which are therefore more sensitive to specific errors due to the presence of ambiguities or rapid or erratic movements.

In the literature, several classifications of object tracking methods have appeared. The most adopted is that proposed by Yilmaz et al. in [3] the authors classified the tracking methods into three categories, namely point tracking, kernel tracking, and silhouette tracking (see table 1). Subsequently, other researchers [4][5] [6] chose this classification in their studies.

Methods based on particle filters have proven, in the literature, their validity in different applications, while methods based on histogram comparison have given acceptable results. The objective of this work is to propose a hybrid method for detecting and tracking moving objects in a sequence of images. The proposed method is inspired by the principle of particle filters; moreover, it is based on color

Table 1: Classification of object tracking methods

Category	Method	Principle	Advantages	Disadvantages
Point Tracking: Moving objects are represented by points	**Kalman Filter** [7][8] [9] [10]	– Based on the optimal recursive data processing algorithm – Composed of two phases, prediction and correction	– Manipulation of noises. – The solutions are optimal.	– Single object tracking. – The state variables follow the Gaussian distribution.
	Particle Filter [11] [12] [13]	– Generalization of Kalman filtering in which the state variable is no longer described by a Gaussian. – Implementation of a recursive Bayesian filter using Monte-Carlo simulations.	– Tracking multiple objects. – Well suited to complex object trajectories and occlusions.	– High computational complexity
	Multiple Hypothesis Tracking (MHT) [14] [15]	– Several frames are observed for better tracking.	– Tracking multiple objects. – Management of occlusion. – Calculation of optimal solutions.	– Very expensive in memory and calculation time.
Kernel Tracking : Objects are represented by a basic geometric shape (rectangle or ellipse, etc.). The estimated movement is generally parametric (translation, rotation, affine, etc.).	**Simple Template Matching** [16] [17] [18] [19] [20]	– The Initial Template is the content of the image within the bounding box in the first frame. – Find the part of the image most similar to the initial Template.	– Treatment of partial occlusion.	– Single object tracking. – Sensitive to changes in illumination. – Slowness due to exhaustive search on all or part of the image.
	Mean Shift Method [21] [22] [23] [24]	– Representation of the object by a histogram. – Non parametric method that iteratively maximizes the similarity in appearance between the weighted color histogram representing the object and the corresponding histogram representing the hypothetical position.	– Management of partial occlusion. – Considerably reduced object location estimation time.	– Single object tracking – Confusion if there are regions with similar colors
	Support Vector Machine (SVM) [25]	– Find the best hyperplane that separates two classes. – Positive examples are images of the object to follow, and negative examples are made up of all the things that are left.	– Management of partial occlusion.	– Single object tracking. – Requires physical initialization. – Requires a learning phase
	Layering Based Tracking [26]	– Modeling of the image by layers, one for the background and one layer for each object. – Each layer consists of the shape (ellipse), movement pattern (translation and rotation), and appearance of the layer.	– Tracking multiple objects. – Management of complete occlusion.	
	Principal component analysis (PCA) [27] [28] [29] [30]	– Construct a subspace representing the appearance of the object using PCA – Transform the image to the eigenspace by minimizing the equation which evaluates the difference between the image reconstructed by the eigenvectors and the input image. – Monitoring is carried out by iteratively estimating the transformation parameters	– Able to manage occlusion and illumination variations	– Sensitive to noise.
	Color histogram [23] [31] [32] [33]	– The object is represented by a histogram of colors. – The use of the Bhattacharyya coefficient makes it possible to detect the object of interest.	– Robust to geometric variations. – The method is quick	– The choice of distance for histogram comparison depends on the frame content
Silhouette Tracking : Used for complex shaped objects,estimate the silhouette of objects of interest for each frame of the video	**Shape Matching** [34] [35] [36]	– Calculation of the similarity between the object and the model generated from the previous image having the form of a "edge map". – At each image the object is updated so that it takes into account the change in appearance.	– Management of occlusion (Hough transformation techniques).	– Single object tracking
	Contour Matching [37] [38] [39] [40]	– Make an initial contour evolve towards the object of interest by calculating a speed of evolution. – The direction of evolution is given by the normal to the contour. – Determine a characteristic that allows you to differentiate the object from the background.	– Flexible for handling a variety of object shapes. – Segmentation of non-homogeneous regions.	– Ineffective if the images are noisy, or when the contour is not entirely inside or outside the region to be segmented. – Only able to segment regions that have sharp edges.

histograms. The goal was to take the benefit of the advantages of both methods in order to resolve the difficulties linked to false detection, especially in the case of partial or total occlusion, of objects having several colors, in the case of a resemblance between the object and background, and in the case of rapid movement or variation in illumination. In this paper, Section 2 presents the tracking methodology, where we detailed the two main stages of tracking, which are detection (Section 2.1) and object tracking (Section 2.2). Section 3 illustrates the experimental results, in which the quantitative results (Section 3.1) and the qualitative results (Section 3.2) are discussed. In Section 4 we close the study and discuss perspectives.

2 Tracking methodology

The tracking is carried out using a hybrid method, based on particle filter and color histogram. The idea realized in this work is inspired by the principle of particle filtering, in which we generate, at the beginning, randomly N particles; a particle in our method is a window of size (M * L) pixels. In each frame, we compare the histogram of each particle with that of the area selected at the beginning of the target object. If the distance between the two histograms is minimal, the object is detected. At this level, the particles having detected the object are kept, while the other particles are deleted. Our contribution is in the resampling step (Section 2.2.2), because it was necessary to generate new particles so that the object will be detected in the next frames, especially in the case of total occlusion. The hybridization of these two methods allowed us to benefit from their advantages. The proposed resampling method allowed us to detect objects after partial or total occlusion; in addition the use of color histograms allowed us to detect objects with multiple colors or objects with a color similar to the background color. In the following, the two main stages of the tracking process, namely detection and tracking itself, will be detailed.

2.1 The first step: detection

Detection is an important step because it allows finding the object of interest in each frame of the video sequence. At first, a region of the object of interest must be selected in the first frame, and the color histogram of this region is calculated. The proposed method consists of detecting moving objects, so in each frame and for each particle a comparison between the color histogram of this particle and the color histogram of the selected area is done, if the distance between the two histograms is minimal then the particle is marked otherwise it will be deleted. Finally the marked particles can indicate that the object is detected.

2.1.1 Histogram comparison

To compare two histograms (H1 and H2), you must choose a metric (distance d(H1,H2)) to express to what extent the two histograms correspond. There are different metrics, in this work two metrics are chosen to see which is the most suitable.

- The intersection distance: it is a simple but widely used measure which counts the number of bins where histograms overlap. This metric gives a value between 0 and the minimum number of samples in both histograms, with 0 indicating no overlap and the maximum value indicating perfect overlap [41]. The intersection distance is calculated by equation ((1)).

$$d\left(H_1, H_2\right) = \sum_I \min\left(H_1(I), H_2(I)\right) \tag{1}$$

- Bhattacharyya distance: It is a measure of similarity between two histograms. It is based on the Bhattacharyya coefficient, which measures the similarity of two probability distributions [41]. Equation (2) shows the bhattacharyya distance.

$$d\left(H_1, H_2\right) = \sqrt{1 - \frac{1}{\sqrt{\tilde{H}_1 \tilde{H}_2 N^2}} \sum_I \sqrt{H_1(I) \cdot H_2(I)}} \tag{2}$$

Where

$$\bar{H}_k = \frac{1}{N} \sum_J H_k(J) \tag{3}$$

N: is the total number of histogram bins.
For the intersection method, the higher the metric, the more accurate the match. However, for Bhattacharyya's method, the less the result, the better the match.

2.2 The second step: object tracking

Tracking object is to locate its position in each frame of the video sequence. The detection process must be repeated for each frame. As indicated in the previous paragraph, marked particles determine the target object;at this level, the eliminated particles must be regenerated in order to predict the position of the target in the next frame. Two sets of particles are generated:

- Particles close to the remaining particles (for the greatest weight particle we generate more particles), This allows tracking to continue when moving nearby.
- Random particles make it possible to detect the object if it is occluded in previous frames or if it is moved at high speed.

In the following, the main algorithm and the proposed method of resampling are detailed.

2.2.1 Main tracking algorithm based on particle filter and color histogram

The main algorithm describes the tracking process, whose principle is to read the video sequence frame by frame, for each frame detect the target object (section 2.1), in this case the object is marked. In order to detect the target object in the next frame a resampling method (section 2.2.2) is proposed to generate new particles.
Algorithm 1 summarizes this process:

$$Pc.x = \sum_{i=1}^{N} \frac{Pi.x}{N} \tag{4}$$

$$Pc.y = \sum_{i=1}^{N} \frac{Pi.y}{N} \tag{5}$$

where :
Pi: The particle i in the previous frame.
N: Number of particles in the previous frame.

2.2.2 Resampling strategy in the proposed method

According to the principle of particle filter, new particles must be generated at each frame, not only to replace the particles already deleted in the previous frame, but also, above all, to be able to detect the object in the following frames. The algorithm 2 summarizes our proposed method for resampling. The method proposed for regenerating deleted particles consists of treating three cases:

1. First case: some particles have been deleted, in this situation you have to look for the particle of the greatest weight to calculate the number of particles to generate around this particle. For the rest of the particles it is necessary to generate only 4 particles around each of them. At this level, if there are particles left to generate, they must be generated randomly to be able to detect the object if it is moved with a large distance.
2. Second case: no particle has been deleted, so the old particle positions must be modified to continue detection. In this case the new position of each particle will be calculated by adding a small random value to the old position, since it is assumed that the object will not be moved too far.

Algorithm 1 Main algorithm

Read the video sequence
while Not End of video **do**
  Read an image
  $NBParticle = 0$
  if first image **then**
    Generate N random particles
    Select an area of interest
    Calculate the histogram of the selected area (Hist 1)
  end if
  for each particle **do**
    Calculate the histogram of the current particle (Hist2)
    Calculate the distance between Hist1 and Hist2
    if $distance <= threshold$ **then** //The current particle is close to the selected area, it must be kept
      Calculate the weight of the particle;
      $NBParticle$= $NBParticle$ +1
    else//The current particle is far from the selected area, it must be deleted
      Delete current particle
    end if
  end for
  Display the current image
  if $NBParticle >= 1$ **then**// At least one particle has detected the target
    calculate the particle center// equation (4) and (5)
    mark the target in the current frame
  end if
  Resampling;
  Go to next image
end while

Algorithm 2 Resampling Algorithm

Calculate the number S of deleted particles
$S = N–NE$ // N: total number of particles // NE: number of existing particles
if $((S > 0) and (S < N))$ **then**
  Find the particle with the greatest weight
  Calculate the number NB of particle to generate around this particle, NB=(S* 25)/100
  Generate NB particles around the particle with the greatest weight //Generate 25% of particle around this particle
  for i=0 to NE **do**
    Generate 4 particles around particle i
  end for
  R= N – number of existing particles //R: number of particles remaining to be generated;
  if $(R >= 1)$ **then**
    Generate R particles randomly
  end if
else
  if (S= 0) **then** // No particle has been deleted
    Change particles positions // add a small random value to the old position, since it is assumed that the object will not be moved too far.
  else// (S=N) all particles have been deleted (the object is occluded)
    Calculate the predicted area around the center particle of the previous frame // we suppose that the target is occluded by another object
    Generate half of the particles in this area
    Generate the other half of the particles randomly
  end if
end if

3. Third case: all the particles have been deleted, that is to say, the object is completely occluded. Normally there is another object which has hidden the target object, and the latter will appear in the following frames, so we propose to calculate the predicted area in which we assume that the object will be moved. This area is a window calculated around the center particle Pc (Equations (4) and (5)) of the previous frame, and then we generate half of the particles in this area, and the other half of the particles randomly. in This way, the probability of finding the object in the following frames is increased.

2.2.3 Performance metrics

In the literature [42] [43][44], many performance metrics for object tracking algorithms are proposed. In this work, we have chosen the two metrics most used in related work.

– **Center location error (CLE)** The oldest way to measure performance is center location error (CLE). It is a common measurement [44] [45] [46][47][48] consisting of measuring the average distance between the centers of the predicted boxes $\{x_t^P\}_{t=1}^N$ and the ground truth boxes $\{x_t^G\}_{t=1}^N$, given by equation (6) .

$$\Delta_\mu \left(\Lambda^G, \Lambda^P\right) = \frac{1}{N}\sum_{t=1}^{N} \left\|x_t^G - x_t^P\right\| \qquad (6)$$

Where : Λ^G: ground truth annotation,
Λ^P: the predicted annotation of the tracker,
N : the number of frames in the sequence.
This distance does not always give a precise measure of performance, since a tracker can lose the target object in a few frames, whether in the case of occlusion or in the case of a bad prediction, in this situation the average distance (CLE) becomes very big. Generally the threshold used to compare the precision between different trackers is 20 pixels [49].
We proposed as a performance measure a **success rate** calculated according to the number of frames having a distance, between the centers of the predicted boxes and the ground truth, less than or equal to 20 pixels

and the total number of frames (equation (7))

$$\text{Success rate} = \frac{\left\| \left\{ \left\| x_t^G, x_t^P \right\| \leq 20 \right\}_{t=1}^{N} \right\|}{N} \tag{7}$$

- **Overlap ratio (VOR)**: The overlap ratio (VOR) [44], between the predicted box A_t^P and the ground truth box A_t^G is considered an important measure. It is defined as the ratio of the intersection and union areas of the boxes, given by equation (8):

$$\emptyset_t \left(\Lambda^{\mathrm{G}}, \Lambda^{\mathrm{P}} \right) = \{\emptyset_t\}_{t=1}^{N}, \quad \emptyset_t = \frac{A_t^G \cap A_t^P}{A_t^G \cup A_t^P} \tag{8}$$

To measure the performance of a tracker we divide the number of successful frames whose overlap ratio $\emptyset_t$ is greater than the given threshold τ (The threshold often used for evaluating tracking performance is 0.5 [20]) by the total number of frames (equation (9))

$$P_\tau \left(A^G, A^P \right) = \frac{\left\| \{ t \mid \emptyset_t > \tau \}_{t=1}^{N} \right\|}{N} \tag{9}$$

Where:
τ: denotes the recovery threshold.
N: denotes the number of frames.
P_τ : denotes the average overlap ratio. This rate takes into account the size of the bounding box, which makes it more precise than the center location error.

3 Experimental results

In this section we present the results obtained by applying our object tracking. We tested our method on the OTB 2013 and OTB 2015 databases. the sequences of images of this databases contain several difficulties such as Occlusion (OCC), Illumination variation(IV), Scale Variation (SV), Deformation (DEF), Motion Blur (MB), Fast Motion (FM), In-Plane Rotation(IPR), Out-of-Plane Rotation (OPR), the Out-of-View (OV), Background Clutters (BC), low resolution (LR). The proposed tracker in this work is based on a particle filter and a color histogram (PFHist). Initially, we randomly launch 100 particles, each particle is represented by a window of 100*100 pixels, and during processing, each particle takes the height and width of the window selected in the target object. This will allow us to compare the color histograms between this window and those of the particle correctly. For histogram comparison; we chose the intersection and Bhattacharyya methods, as they are the most widely used in most studies. In our experience, we noticed that the Bhattacharyya method yielded satisfactory results in most sequences chosen with a comparison threshold between 0.4 and 0.8, while in some sequences we were forced to choose the intersection method with a comparison threshold between 0.6 and 0.8 to obtain acceptable results.

3.1 Quantitative results

At first we tried to make a comparison between the Camshift tracker (which is used in different research works in this field) and the proposed tracker (PFHist). Since the color space influences the quality of the result, we have tested our tracker with the two color spaces HSV (Hue, Saturation, Value) and RGB (Red, Green and Blue) which are most used in computer vision.
Figure 1 shows the results obtained for tracking a person in a David3 sequence with difficulties (OCC, DEF, OPR, BC), this sequence contains 253 frames of 640*480 pixels.

The test of these three trackers allowed us to notice that:

- The Camshift tracker, at first, was able to track the target even in the case of partial occlusion, but we noticed that after almost complete occlusion it lost the target (frame 193).
- The PFHist tracker with the HSV color space was also able to follow the target but when it was occluded it lost it for a certain moment and then it was able to detect it, since the proposed particle resampling method allowed to generate particles everywhere.
- By implementing the same tracker (PFHist) with the RBG color space we noticed that it can detect the target object quickly after occlusion, the overlap ratio justifies this result.

Table 2 presents a comparison of our PFHist tracker with the results presented in [50] of the Meanshift and KaMS (Kalman Meanshift) trackers using the VOR (overlap ratio) metric. The maximum VOR in the tested sequences shows that PFHist has the best average performance (0.6677). We

Table 2: Comparison of the overlap ratio obtained by the Meanshift, KAMS trackers (KALMAN and MeanShift) [50] and that of our PFHist tracker

Sequence	Meanshift	KaMS	PFHist
David3	0.6721	0.6851	**0.7549**
Jogging	0.1589	**0.5097**	0.4593
Girl2	0.5338	0.6517	**0.7889**
Average	0.4549	0.6155	**0.6677**

also experimented with various video sequences with different resolutions and different challenges. The number of frames differs from one sequence to another; we tried to do the tests with two color spaces (HSV and RGB) and two numbers of particles (100 and 200). To measure performance we used the averages of the overlap ratio VOR and the success rate. The obtained results of this experience are summarized in table 3. Figure 2 and 3 shows, respectively, a comparison of the overlap ratio and success rate obtained with the two color spaces (HSV and RGB).

From the results obtained, we can notice that:

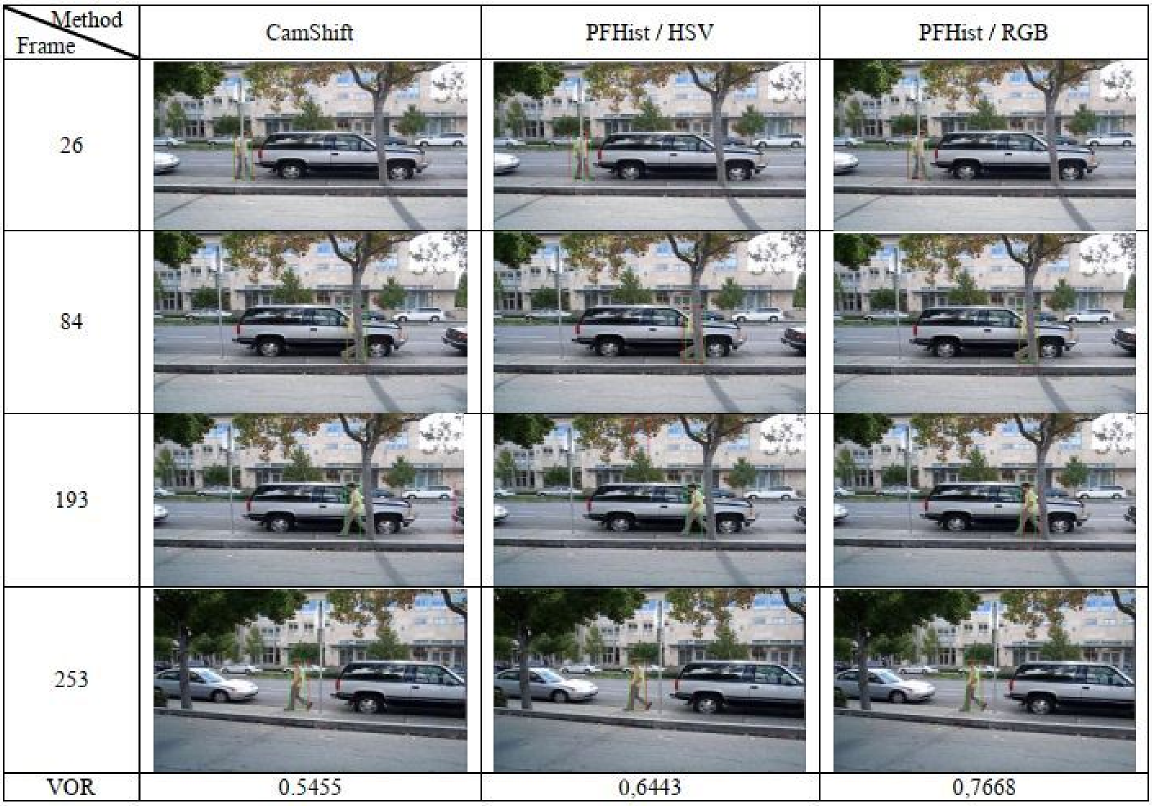

Figure 1: Comparison between CamShift, PFHist/HSV and PFHist/RGB trackers. The green rectangle represents the bounding box of the target (the ground truth) and the red rectangle represents the predicted box of the target.

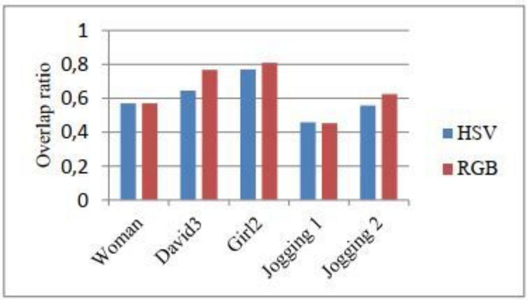

Figure 2: Comparison of the overlap ratio obtained with the two color spaces (HSV and RGB) tested on some sequences from the OTB database

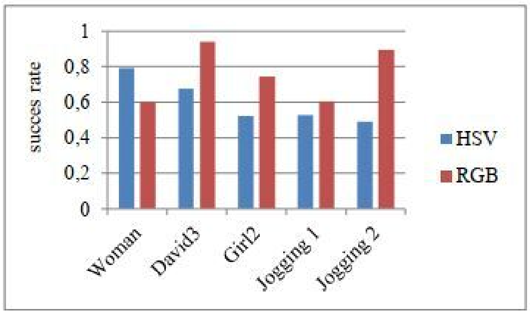

Figure 3: Comparison of the success rate obtained with the two color spaces (HSV and RGB) tested on some sequences from the OTB database

– Performance varies from sequence to sequence because challenges are different.

– The RGB color space gave us more satisfactory results; the best result was with the Girl2 sequence with an overlap ratio equal to 0.8079 (Figure 2 and table 3) and the David3 sequence with a success rate equal to 0.9368 (Figure 3 and table 3).

– Increasing the number of particles did not affect the results too much. The best results are obtained, in most cases, with 100 particles.

In another experiment, we wanted to test our tracker with other trackers implemented in the OTB database. The results presented in the precision and success plots (Figure 4), together with the measured values for the three sequences David3, Girl2, and Jogging, provide a clear evaluation of the robustness of the considered trackers: CT [51], CSK [52], TM [53] and PFHist.

The CT (Compressive Tracking) tracker, although computationally efficient, exhibits high sensitivity to appearance variations and occlusions. Although it achieves 20.52% of precision on Jogging, the poor results on the David3 sequence (precision close to 0%) and on the Girl2 sequence

Table 3: Overlap ratio and success rate in some sequences with different conditions (the best result is indicated in **red**, and the second-best result in **green**)

Sequence with difficulties	Number of frame	Resolution	Space color	Number of particles	Average VOR	Success rate
Woman (IV, SV, OCC, DEF, MB, FM, OPR)	597	352*288	HSV	100	0,5700	0,7906
				200	0,6834	0,7990
			RGB	100	0,4372	0,6013
				200	0,5276	0,8208
David3 (OCC, DEF, OPR, BC)	253	640*480	HSV	100	0,6443	0,6759
				200	0,5375	0,5257
			RGB	100	0,7668	**0,9368**
				200	0,5336	0,6166
Girl2 (SV, OCC, DEF, MB, OPR)	630	640*480	HSV	100	**0,7698**	0,5206
				200	0,7492	0,5000
			RGB	100	**0,8079**	0,7444
				200	**0,7698**	0,7460
Jogging (1st person) (OCC, DEF, OPR)	307	352*288	HSV	100	0,4593	0,5244
				200	0,6124	0,6352
			RGB	100	0,4528	0,6026
				200	0,4886	0,6026
Jogging (2nd person)(OCC, DEF, OPR)			HSV	100	0,5570	0,4886
				200	0,6059	0,5472
			RGB	100	0,6254	0,8502
				200	0,6808	**0,8925**

(precision of 2.54%) show that it quickly loses the target after a few frames. Its compressed feature model is not sufficiently discriminating in complex contexts. The CSK (Circulant Structure Kernel) tracker, relying on correlations in the Fourier domain, achieves moderate performance with good stability on David3 (65.87% precision, 62.70% success), but fails in Girl2(precision 16.98%) and Jogging (precision 22.80%) where rapid appearance changes disturb the correlation model. The TM (Template Matching) tracker shows limited performance. Although it reaches 22.62% precision on David3, it is ineffective in sequences with significant deformation or occlusion (7.78% accuracy on Girl2 and 18.24% on Jogging). Due to its reliance on a fixed template. Our Tracker PFHist, demonstrates remarkable stability across all three sequences. For David3, it achieves a precision of 69.44% and a success rate of 62.70%, showing its ability to maintain accurate tracking despite partial occlusions and illumination changes. For Girl2, PFHist achieves 60.79% precision and 70.95% success, still maintaining good performance despite rapid motion and shadow effects. Similar results are observed for Jogging (74.59% precision and 65.15% success), confirming its robustness against body movements and appearance variations. Across all sequences, the PFHist is clearly the best-performing tracker, combining stability, precision, and adaptability. The CT, CSK, and TM trackers remain competitive for rigid and slightly deformable targets, but their low resilience to variations in scale, lighting, and motion limits their use in dynamic contexts.

3.2 Qualitative result

The figure 5 illustrates the variation in the overlap ratio during the sequence of David3 images.

The curve shows that in most frames the overlap ratio is higher than the set threshold (0.5). Frames with an overlap ratio equal to zero or almost zero are those whose object is partially or totally occluded. The figure 6 presents a set of frames during the tracking process with our PFHist tracker, the results obtained show that the tracker was able to follow the target in the majority of frames and was able to detect it quickly after occlusion, for example from the occlusion in frames 86-87 he recovered the target in frame 89 and from the occlusion in frame 188 he recovered it in frame 193.

To ensure that the PFHist tracker can detect and track the target after occlusion, we have tested it with other video sequences. Figure 7 shows the results obtained. We notice that in the majority of cases the target object is detected after a few frames of occlusion.

By applying the PFHist tracker to other video sequences, we have noticed that the tracker does not always give satisfactory results. However, by choosing the intersection distance as the metric for comparison histograms, we were able to obtain acceptable results. Figure 8 columns 1 and 2 illustrate the result obtained with the Woman video sequence having difficulties (IV, SV, OCC, DEF, MB, FM, OPR).

The tracker is also tested with sequences containing blurry images BlurCar1 (MB, FM) or objects having the same colors as the target object Basketball (IV, OCC, DEF, OPR,

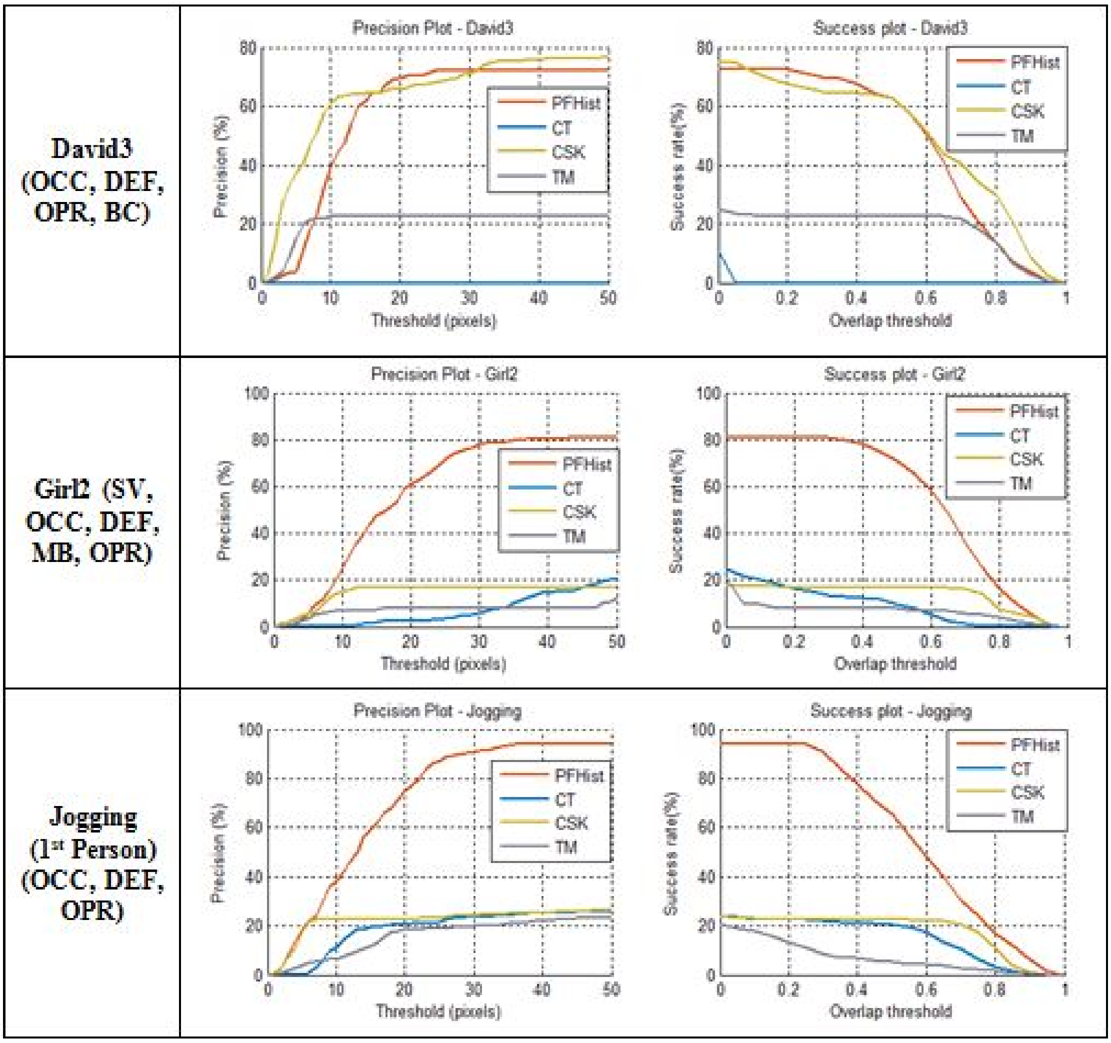

Figure 4: Comparative evaluation of tracking performance on David3, Girl2, and Jogging sequences

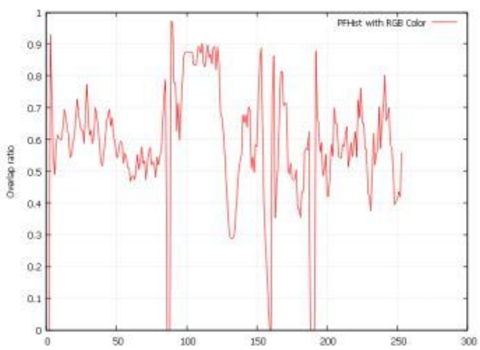

Figure 5: The overlap ratio during the sequence David3

BC). The results obtained (Figure 8 columns 3 and 4) showed that the object can be followed in these cases, since the particles distributed in each frame make it possible to detect the object even if it is lost in the previous frames.

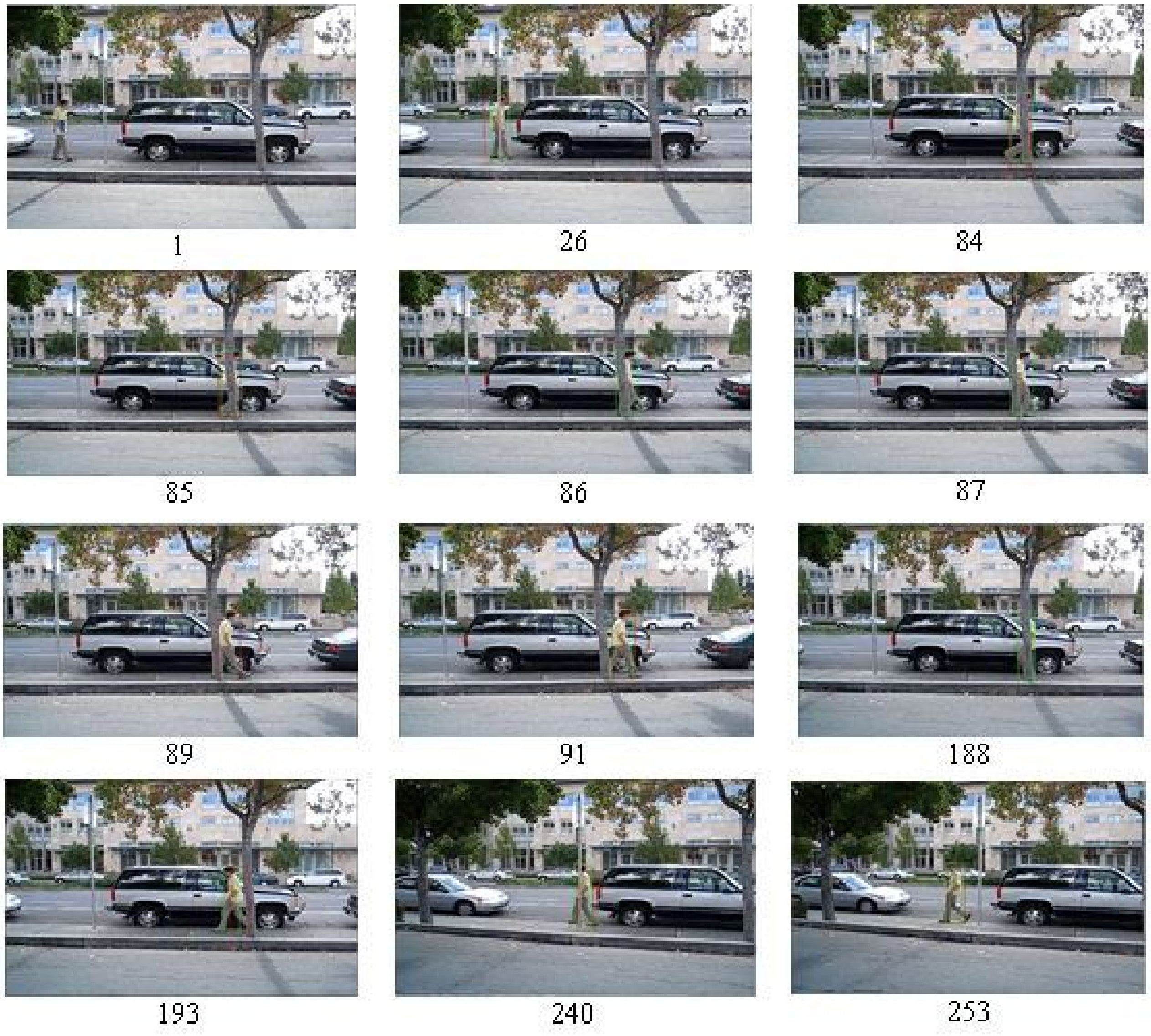

Figure 6: Object detection and tracking during the sequence David3

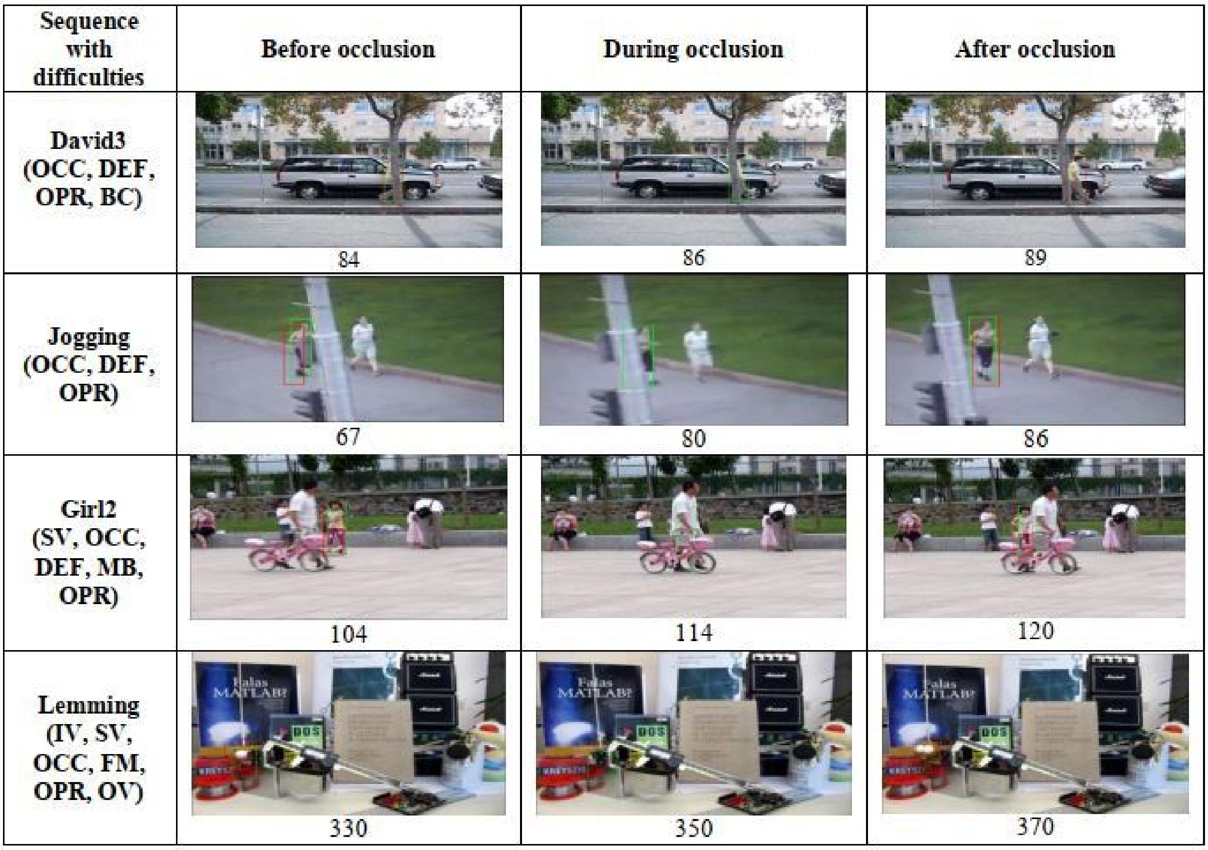

Figure 7: Object detection and tracking after an occlusion in some sequence of the OTB database

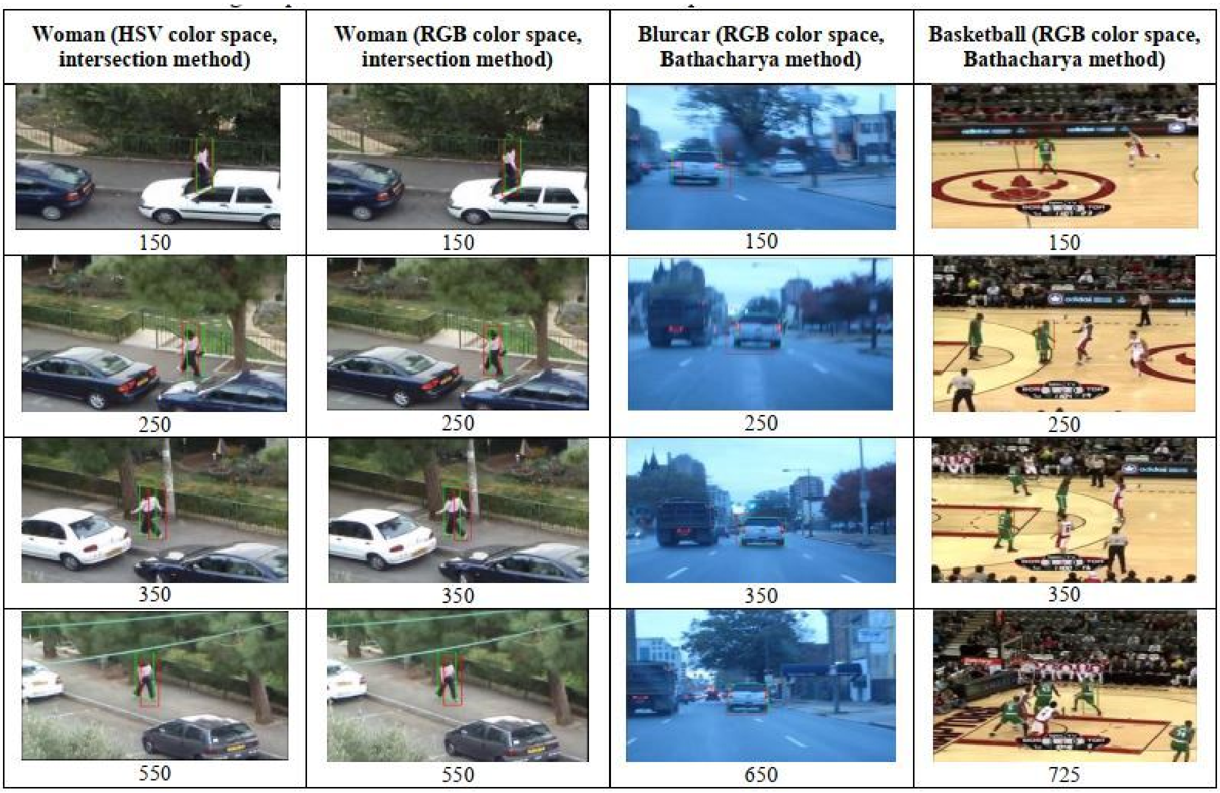

Figure 8: Object tracking in some sequence of the OTB database with different conditions

4 Conclusion

In this paper, we present a hybrid method for detecting and tracking objects; the method is based on the principle of particle filters and color histograms. The hybridization of these two methods allowed us to benefit from their advantages. The proposed method allowed us to obtain results with a reduced number of particles. In addition, the fact of considering each particle as an area or window allowed us to use the color histograms, which makes the calculation of the likelihood function that measures the compatibility between the state of the particle and the real observation easier and more precise. Moreover, the proposal of a prediction region in which a large part of particles will be regenerated in the next frame allowed us to detect the target easily even after an occlusion.
In a future study, the proposed PFHist tracker can be improved by the addition of components such as the development of methods allowing automatic selection of the most appropriate color space for each sequence of images with the aim of improving the tracking quality and the choice of other characteristics of the target objects in order to exploit them in the detection of the object in each frame. The method can also be improved so that it can track multiple objects.

References

[1] P. A. Brasnett, L. Mihaylova, N. Canagarajah, and D. Bull, "Particle filtering with multiple cues for object tracking in video sequences," in *Image and Video Communications and Processing 2005*, vol. 5685, pp. 430–441, International Society for Optics and Photonics, SPIE, 2005, https://doi.org/10.1117/12.585882.

[2] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, pp. 197–208, 2000, https://doi.org/10.1023/A:1008935410038.

[3] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, pp. 13–es, 2006, https://doi.org/10.1145/1177352.1177355.

[4] S. Balaji and S. Karthikeyan, "A survey on moving object tracking using image processing," in *2017 11th international conference on intelligent systems and control (ISCO)*, pp. 469–474, IEEE, 2017, https://doi.org/10.1109/ISCO.2017.7856037.

[5] I. Pathan and C. Chauhan, "A survey on moving object detection and tracking methods," in *Computer Science*, 2015, https://api.semanticscholar.org/CorpusID:6112133.

[6] R. Sharma and S. Gupta, "A survey on moving object detection and tracking based on background subtraction," *The Oxford Journal of Intelligent Decision and Data Science*, vol. 01, pp. 55–62, 2018, https://doi.org/10.5899/2018/OJIDS-00041.

[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transaction of the ASME Journal of Basic Engineering*, p. 35–45, 1960, https://doi.org/10.1115/1.3662552.

[8] S.-Y. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on industrial electronics*, vol. 59, no. 11, pp. 4409–4420, 2011, https://doi.org/10.1109/tie.2011.2162714.

[9] D. Ponsa, A. López, J. Serrat, F. Lumbreras, and T. Graf, "Multiple vehicle 3d tracking using an unscented kalman," in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pp. 1108–1113, IEEE, 2005, https://doi.org/10.1109/itsc.2005.1520206.

[10] K. Robert, "Night-time traffic surveillance: A robust framework for multi-vehicle detection, classification and tracking," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 1–6, IEEE, 2009, https://doi.org/10.1109/avss.2009.98.

[11] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE proceedings F (radar and signal processing)*, vol. 140, 2, pp. 107–113, IET, 1993, https://doi.org/10.1049/ip-f-2.1993.0015.

[12] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 1–25, 1996, https://doi.org/10.1080/10618600.1996.10474692.

[13] E. Arnaud, E. Memin, and B. Cernuschi-Frias, "Conditional filters for image sequence-based tracking-application to point tracking," *IEEE Transactions on image processing*, vol. 14, no. 1, pp. 63–79, 2004, https://doi.org/10.1109/tip.2004.838707.

[14] I. J. Cox and S. L. Hingorani, "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 2, pp. 138–150, 1996, https://doi.org/10.1109/34.481539.

[15] D. Reid, "An algorithm for tracking multiple targets," *IEEE transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 2003, https://doi.org/10.1109/tac.1979.1102177.

[16] S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *Proceedings. 1998 IEEE Computer Society conference on computer vision and pattern recognition*

(Cat. No. 98CB36231), pp. 232–237, IEEE, 1998, https://doi.org/10.1109/cvpr.1998.698614.

[17] H. Schweitzer, J. W. Bell, and F. Wu, "Very fast template matching," in *European Conference on Computer Vision*, pp. 358–372, Springer, 2002, https://doi.org/10.1007/3-540-47979-1_24.

[18] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," in *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 723–730, IEEE, 2010, https://doi.org/10.1109/cvpr.2010.5540145.

[19] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011, https://doi.org/10.1109/tpami.2011.239.

[20] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking vot2015 challenge results," in *Proceedings of the IEEE international conference on computer vision workshops*, pp. 1–23, 2015, https://doi.org/10.1109/iccvw.2015.79.

[21] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 2, pp. 142–149, IEEE, 2000, https://doi.org/10.1109/cvpr.2000.854761.

[22] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002, https://doi.org/10.1109/34.1000236.

[23] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 5, pp. 564–577, 2003, https://doi.org/10.1109/tpami.2003.1195991.

[24] A. S. Jalal and V. Singh, "The state-of-the-art in visual object tracking," *Informatica*, vol. 36, no. 3, 2012, https://informatica.si/index.php/informatica/article/download/405/408.

[25] S. Avidan, "Support vector tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 8, pp. 1064–1072, 2004, https://doi.org/10.1109/tpami.2004.53.

[26] R. Mishra, M. K. Chouhan, and D. D. Nitnawwre, "Multiple object tracking by kernel based centroid method for improve localization," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 7, pp. 137–140, 2012.

[27] M. J. Black and A. D. Jepson, "Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 26, pp. 63–84, 1998, https://doi.org/10.1023/a:1007939232436.

[28] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, "Object tracking via partial least squares analysis," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4454–4465, 2012, https://doi.org/10.1109/tip.2012.2205700.

[29] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, "Robust visual tracking based on incremental tensor subspace learning," in *2007 IEEE 11th international conference on computer vision*, pp. 1–8, IEEE, 2007, https://doi.org/10.1109/iccv.2007.4408950.

[30] J. Wen, X. Li, X. Gao, and D. Tao, "Incremental learning of weighted tensor subspace for visual tracking," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 3688–3693, IEEE, 2009, https://doi.org/10.1109/icsmc.2009.5346874.

[31] S. Boltz, *A statistical framework in variational methods of image and video processing problems with high dimensions*. PhD thesis, Université Nice Sophia Antipolis, 2008, https://theses.hal.science/tel-00507488/.

[32] A. Elgammal, R. Duraiswami, and L. S. Davis, "Probabilistic tracking in joint feature-spatial spaces," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, pp. I–I, IEEE, 2003, https://doi.org/10.1109/cvpr.2003.1211432.

[33] V. Garcia, *Tracking objects of interest in an image sequence: from salient points to statistical measurements (Suivi d'objets d'intérêt dans une séquence d'images : des points saillants aux mesures statistiques)*. PhD thesis, University of Nice – Sophia Antipolis, 2008, https://hal.science/tel-00374657/.

[34] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge, "Tracking non-rigid objects in complex scenes," in *1993 (4th) International Conference on Computer Vision*, pp. 93–101, IEEE, 1993, https://doi.org/10.1109/iccv.1993.378231.

[35] L. Ma, J. Liu, J. Wang, J. Cheng, and H. Lu, "An improved silhouette tracking approach integrating particle filter with graph cuts," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1142–1145, IEEE, 2010, https://doi.org/10.1109/icassp.2010.5495366.

[36] B. Li, R. Chellappa, Q. Zheng, and S. Z. Der, "Model-based temporal object verification us-

ing video," *IEEE Transactions on Image Processing*, vol. 10, no. 6, pp. 897–908, 2001, https://doi.org/10.1109/83.923286.

[37] D. Terzopoulos and R. Szeliski, "Tracking with kalman snakes," *Active vision*, vol. 20, pp. 3–20, 1992, https://www.academia.edu/download/30742188/Snakes.pdf.

[38] M. Bertalmio, G. Sapiro, and G. Randall, "Morphing active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 733–737, 2000, https://doi.org/10.1109/34.865191.

[39] Y. Chen, Y. Rui, and T. S. Huang, "Jpdaf based hmm for real-time contour tracking," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I–I, IEEE, 2001, https://doi.org/10.1109/cvpr.2001.990521.

[40] A.-R. Mansouri, "Region tracking via level set pdes without motion computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 947–961, 2002, https://doi.org/10.1109/tpami.2002.1017621.

[41] K. Safjan, "Metrics used to compare histograms," *Krystian's Safjan Blog*, 2020, https://safjan.com/metrics-to-compare-histograms/.

[42] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2411–2418, 2013, https://doi.org/10.1109/cvpr.2013.312.

[43] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013 https://doi.org/10.1109/tpami.2013.230.

[44] L. Čehovin, M. Kristan, and A. Leonardis, "Is my new tracker really better than yours?," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 540–547, IEEE, 2014, https://doi.org/10.1109/wacv.2014.6836055.

[45] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *2006 IEEE Computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 1, pp. 798–805, IEEE, 2006, https://doi.org/10.1109/cvpr.2006.256.

[46] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International journal of computer vision*, vol. 77, pp. 125–141, 2008, https://doi.org/10.1007/s11263-007-0075-7.

[47] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *2009 IEEE Conference on computer vision and Pattern Recognition*, pp. 983–990, IEEE, 2009, https://doi.org/10.1109/cvprw.2009.5206737.

[48] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276, 2010, https://doi.org/10.1109/cvpr.2010.5539821, organization=IEEE.

[49] I. Leang, *On-line fusion of visual object tracking algorithms (Fusion en ligne d'algorithmes de suivi visuel d'objet)*. PhD thesis, Pierre et Marie Curie-Paris VI University, 2016, https://hal.science/tel-01485292/.

[50] S. MEDOUAKH, *Object detection and tracking (Détection et suivi d'objets)*. PhD thesis, Mohamed Khider University, Biskra, Algeria, 2019, http://thesis.univ-biskra.dz/4499/1/THESE

[51] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Computer Vision – ECCV 2012* (A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), pp. 864–877, Springer Berlin Heidelberg, 2012, https://doi.org/10.1007/978-3-642-33712-3_62.

[52] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*, pp. 702–715, Springer, 2012, https://doi.org/10.1007/978-3-642-33765-9_50.

[53] F. Liu, C. Gong, X. Huang, T. Zhou, J. Yang, and D. Tao, "Robust visual tracking revisited: From correlation filter to template matching," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2777–2790, 2018, https://doi.org/10.1109/tip.2018.2813161.