

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,800

Open access books available

144,000

International authors and editors

180M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Estimation for Motion in Tracking and Detection Objects with Kalman Filter

*Afef Salhi, Fahmi Ghozzi and Ahmed Fakhfakh*

## Abstract

The Kalman filter has long been regarded as the optimal solution to many applications in computer vision for example the tracking objects, prediction and correction tasks. Its use in the analysis of visual motion has been documented frequently, we can use in computer vision and open cv in different applications in reality for example robotics, military image and video, medical applications, security in public and privacy society, etc. In this paper, we investigate the implementation of a Matlab code for a Kalman Filter using three algorithm for tracking and detection objects in video sequences (block-matching (Motion Estimation) and Camshift Meanshift (localization, detection and tracking object)). The Kalman filter is presented in three steps: prediction, estimation (correction) and update. The first step is a prediction for the parameters of the tracking and detection objects. The second step is a correction and estimation of the prediction parameters. The important application in Kalman filter is the localization and tracking mono-objects and multi-objects are given in results. This works presents the extension of an integrated modeling and simulation tool for the tracking and detection objects in computer vision described at different models of algorithms in implementation systems.

**Keywords:** Kalman filter, tracking objects, detection objects, localization, video and image processing, computer vision, embedded system

## 1. Introduction

The computer vision, from the technological evolution point of view, is the most useful in our days. It is a discipline at the border of computer science, mathematics, physics, neuroscience and various other disciplines, which aims to initiate the specific issues of image and video analysis from 2D and 3D environment, and to implement a simple object tracking application. This phenomenon provokes a spectacular development of applications in various fields in many sectors of activity: imaging systems, robotics, surveillance systems, identification of interest (automatic annotation and retrieval of video from databases multimedia data), indexing and augmented reality, HMI interaction (gesture and gaze recognition for data entry on computers), etc. These systems are most used in airports, metros, prisons, banks or nuclear power plants, intelligent transport systems, the analytical

approach for medical applications, military imagery with the target weapon, applications security and computer-controlled automatic surveillance (scene surveillance, object tracking and behavior analysis, swimming systems for swimming pool surveillance, to prevent accidents and drowning victims), video conference, driving assistance (reversing radar, speed limiter or cruise control), pedestrian tracking (counting and pedestrian tracking systems using aerial cameras), biometric systems (fingerprints and recognition biometric facial), etc. Such an application uses computer vision techniques: object detection, classification of moving objects and tracking of objects, etc. The main objective is to locate a known object in the image in order to follow it up such as faces, people, hand gestures, cars, etc. The current trend is to lighten the tasks performed by humans by integrating intelligence into these systems. So in computer vision, the tracking of moving objects in a known or unknown environment is commonly studied since the year 1970. Monitoring can be a tool to give visual autonomy to robots. In this case, visual perception is a prerequisite for action and requires learning to establish links between the causes and the actions to be produced in response. Tracking objects can also automate repetitive tasks. Such a monitoring system must be robust to the following real constraints: variations in the lighting of the sequence, change in the pose of the object (front view, profile view), change in scale (change in size of the object), change of appearance, simultaneous movement of the camera and the object, partial and total occlusions, or even the kinematics (for example the space-time constraints) and low processing time (20 images/s). Our aim is to classify these methods efficiently in order to highlight the advantages and disadvantages of each method. This will allow us, later, to choose the most robust algorithm for an object tracking system. The object tracking system uses the method of tracking the region of interest of the object in a video sequence. Several points will be discussed, such as the pre-treatment methods, the change of the object and its movement, the change of appearance, the change of scale, and change of illumination. Then we will compare the tracking results for the different video sequences analyzed and show the performance of the implemented algorithm.

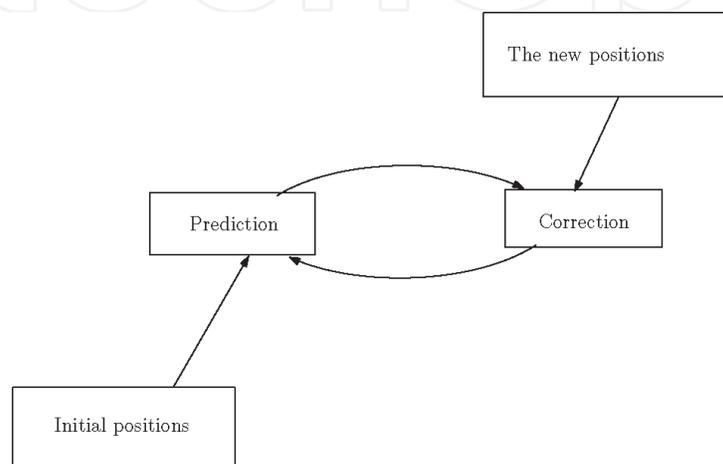
Tracking corresponds to the estimation of the location of the object in each of the images in a video sequence, the camera and/or the object (face, man, hand, animal, etc.) being able to be simultaneously in motion. The localization process is based on the recognition of the object of interest from a set of visual characteristics (color, shape, speed, etc.). Specifically, the purpose of an object tracking method is to estimate, in each image of the sequence, the functions that are used in tracking the object or objects present in the field of vision of the camera such as motion, color, corners, outline, shape, and object view. In object tracking, the class, appearance, scale, and/or location of the tracking region are predicted based on the forward images and on the underlying model for state transitions. The state of the object is generally represented by its location and its speed.

There are then three main stages in the analysis of the video sequence, the first stage consists in carrying out the detection of moving objects. Then the step of tracking these objects from one image to another and finally, we analyze the tracks of objects to recognize their behavior. Many different techniques for tracking objects have been proposed. The detection events and detection moving objects in complex scenes is difficult to analyze due to camera noise and changing lighting conditions. Each limitation must be overcome in order to avoid failure of the tracking algorithm. In an object tracking algorithm, there are generally four steps: detection, location, association, and trajectory estimation [1–3]. The algorithms are composed by three important modules: block matching and meanshift, camshift, Kalman filter. The Kalman filter is used in a wide range of technological fields. It is a major theme in automation and frame and signal processing. The Kalman filter

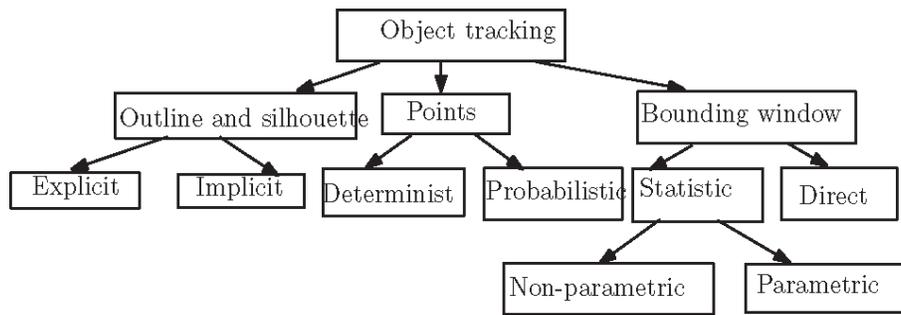
“KF” is a set of mathematical equations which provide an efficient (recursive) computation of the means for estimation the state of a process. The KF is very powerful in several aspects: it supports estimates of past, present and even future states and it can do so even when the exact nature of the modeled system tracking and detection objects. The Kalman filter is a corrective predictor filter. In the tracking system objects, this filter looks at an object as it moves, that is, it takes information on the state of the object at the precise moment. Then, it uses this information to predict where the object is in the next frame. For this, it takes as input a measurement vector (position in x, in y, width and height of the object). In the tracking process, this filter looks at an object as it moves, that is it takes information on the state of the object at the precise moment. In the case of tracking an object in motion, the Kalman filter allows us to estimate the states of motion of the object (and therefore predetermine the areas of motion in the following frames with using the combination for three algorithms (block-matching, Camshift and Kalman Filter)) and thus adds robustness tracking objects. Many authors have studied the Kalman filter in object tracking [1, 2]. In this work, we optimized many criteria in image and video processing application. For example, we can site: time execution, quality and performance in the image and video processing, artifact and noise in a frame, etc., the data flow for Kalman Filter is presented in **Figure 1**.

## 2. Different methods of modeling an object

In a follow-up scenario, an object can be defined as anything that is of interest for further analysis. For example people walking on a road, boats on the sea, fish inside an aquarium, airplanes in the air, cars on the road, a motion hand or face, motion for different objects and multi-objects, etc. It is a collection of objects that may be important to track in a specific area or environment. The implementation of an object tracking system involves designing two main parts, the object representation and the object location. The localization step is based on the representation model of the object and its location in the previous frame. The representation consists in associating with the object followed characteristics of shape and/or appearance allowing to recognize it in successive frames. In recent studies, representations by shape and appearances are classified into three families such as representation by point clouds, representation by bounding boxes (representation by geometric shapes) and representation by silhouettes. In what follows, we will describe these methods as illustrated in **Figure 2** [4–6].



**Figure 1.**  
*The data flow for a Kalman filter.*



**Figure 2.**  
Object tracking methods.

1. What representation is appropriate for tracking objects?
2. What algorithm should be used?
3. How is the movement, appearance and shape of the object modeled?

Tracking of events and detection motion objects in complex scenes is difficult to analyze due to camera noise and changing lighting conditions. Each limitation must be overcome in order to avoid failure of the tracking algorithm. In an object tracking algorithm, there are generally four steps: object detection, location, association, and trajectory estimation. We will be interested in this master's work [1, 2, 7] to study the different methods of representing objects in a video sequence.

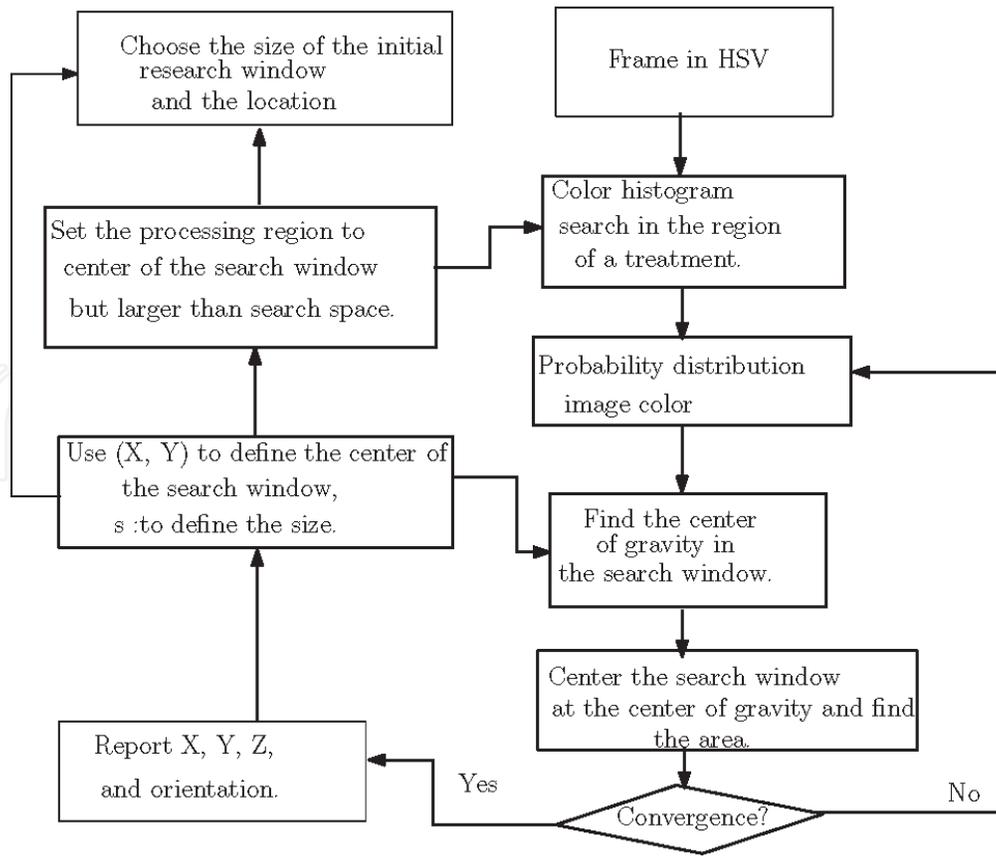
## 2.1 Camshift algorithm

Camshift is an algorithm for tracking objects in real time (people, vehicles). It is based on the colors developed in the video sequence. Camshift is based on the average displacement algorithm (Meanshift). The calculation module is based on iterations to reach convergence. Camshift take the HSV color space as a model with the color tone component (*Hue*). This component is designed to calculate the probability of the histogram of each image of the analyzed sequence. The size of the original window was just large enough to fit most of the object inside of it. The Camshift algorithm adjusts the size of the search window according to the movement of the object analyzed with constant tint. Whereas, for a quick movement, the follow-up can fall into the analysis of another object in the sequence. For this reason, we choose at the beginning of the algorithm a threshold of color hue of the object to ensure correct tracking. Once the mean displacement module converges, the center of gravity and the zero order moment are calculated. Then we calculate the new size, width and length of the search window. Then, the window is centered around center of gravity and the calculation of the next image is started. Next, we calculate the Camshift parameters such as the secondary moments, the orientation, the width and the length of the window around the object's center of gravity.

**Figure 3** shows the flowchart of the Camshift algorithm for object tracking.

### 2.1.1 Calculation of Camshift parameters

With each iteration of the Camshift algorithm, the object search window will be resized. To search for the new size, the search window obtained by the average displacement algorithm is slightly enlarged to include the object. Then, the parameters of the window must be adapted such as the width, the length and the center of gravity. The term  $M0/255$  is the normalized area since the zero order moment is calculated



**Figure 3.**  
 Flowchart of the Camshift algorithm for object tracking.

from the probability distribution of the image which can have values from 0 to 255. The search window adaptation parameters are computed in Eqs. (1), (2) and (3) [1, 7].

$$s = 2 \times \sqrt{\frac{M00}{255}} \quad (1)$$

$$W = [s \quad 1.2 \times s] \quad (2)$$

$$Lc = [(xc - (W(1)/2)) \quad (yc - (W(2)/2))] \quad (3)$$

We calculate the secondary moments using the Eqs. (4), (5) and (6), orientation is calculated by Eq. (7) and the length and width of the object search window (11) and (12).

$$M20 = \sum_x \sum_y x^2 \times I(x, y) \quad (4)$$

$$M02 = \sum_x \sum_y y^2 \times I(x, y) \quad (5)$$

$$M11 = \sum_x \sum_y x \times y \times I(x, y) \quad (6)$$

$$\theta = \frac{\arctan \frac{2 \times \left( \frac{M11}{M00} - xc \times yc \right)}{\left( \frac{M20}{M00} - xc^2 \right) - \left( \frac{M02}{M00} - yc^2 \right)}}{2} \quad (7)$$

$$\theta = \frac{\arctan \frac{2 \times b}{a - c}}{2}$$

$$a = \frac{M20}{M00} - x_c^2 \quad (8)$$

$$b = 2 \times \left( \frac{M11}{M00} - x_c \times y_c \right) \quad (9)$$

$$c = 2 \times \left( \frac{M02}{M00} - y_c^2 \right) \quad (10)$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (11)$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (12)$$

## 2.2 Kalman filter

### 2.2.1 Definition

The Kalman filter is a set of mathematical equations which provide an efficient (recursive) computation of the means for estimation the state of a process, so as to minimize the mean of the quadratic error. The filter is very powerful in several aspects: it supports estimates of past, present and even future states and it can do so even when the exact nature of the modeled system is unknown. The filter allows, thanks to its role, to correct and restrict the areas in which we seek movement in the next step. We can see the quadratic error by Eq. (13).

$$E_x = x_{Kalman} - x_{tracking} \quad (13)$$

### 2.2.2 Role of the Kalman filter in the tracking application

The Kalman filter is used in a wide range of technological. In the tracking and detection process, this filter looks at an object as it moves, that is, it takes information on the state of the object at the precise moment. Then, it uses this information to predict where the object is in the next frame. For this, it takes as input a measurement vector (position in x, in y, width and height of the object). Then it acts on so-called internal parameters (position, speed and acceleration in x and y, as well as the height, the width) to make a prediction and then an estimate of these. Finally, the result is an estimate of the following measurement. In the case of tracking an object in motion, the Kalman filter allows us to estimate the states of motion of the object. Many authors have studied the Kalman filter in object tracking [1, 8, 9], the differences of the present work and the earlier works are the type and the method of objects tracking.

### 2.2.3 Formulation and modeling of the Kalman filter

The main objective of the Kalman filter is to estimate the vector of states in a discrete time. This process is illustrated by Eq. (14) with stochastic linear differences:

$$x_k = A \times x_{k-1} + w_{k-1} \quad (14)$$

With a measurement vector which has the following form (15):

$$z_k = H \times x_k + v_k \quad (15)$$

$A$  is the transition matrix and  $H$  presents the measurement matrix. Random variables  $w_k$  and  $v_k$  present Gaussian and measurement noise (respectively). They are assumed to be independent (from each other), the covariance of  $w_k$  is a matrix  $Q$  (16), similarly the covariance of  $v_k$  is a matrix “ $R$ ” (17).

$$p(w_k) \sim N(0, Q) \quad (16)$$

$$p(v_k) \sim N(0, R) \quad (17)$$

The Matrix  $A$  models the movement of the object. The movement model used is generally at constant speed or at constant acceleration. Since the movement of objects is not uniform, this type of model is not suitable for describing all movements in general. However, we assume that the object movements that we consider in the third chapter have an adapted dynamic. In our case, we use the motion model with constant acceleration, for tracking a point in two dimensions. The state vector is written:

$$x_k = (x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}) \quad (18)$$

where  $(x, y)$  is the position,  $(\dot{x}, \dot{y})$  presents the speed and  $(\ddot{x}, \ddot{y})$  is the acceleration. Note that in the object tracking application, the noise covariance matrices ( $Q, R$ ), the transition matrix  $A$  and the measurement matrix are assumed to be constant. The variable parameters over time are the state vector and the measurement vector since the position and size of the object changes during the sequence.

### 2.3 The origins of filter formulation

We define  $\hat{x}_k^-$  the estimation of a priori states at the moment  $k$  and we give knowledge of the prior process at the moment  $k$ , and  $\hat{x}_k$  the estimate of the state a posteriori at the moment  $k$  which is given to the measurement vector  $z_k$  (15). We can then define the a priori and a posteriori estimation errors by the following Eqs. (19) and (20).

$$e_k^- = x_k - \hat{x}_k^- \quad (19)$$

$$e_k = x_k - \hat{x}_k \quad (20)$$

The covariance of the a priori estimation error is illustrated by Eq. (21):

$$P_k^- = E[e_k^- e_k^{-T}] \quad (21)$$

For computing statistics for the Kalman filter, we start with the goal of finding an equation that computes an a posteriori state estimate as a linear combination of an a priori estimate and a weighted difference between an actual measure  $z_k$  and a measurement prediction  $H\hat{x}_k^-$  as shown in the equation below (22):

$$\hat{x}_k = \hat{x}_k^- + K \times (z_k - H \times \hat{x}_k^-) \quad (22)$$

The difference  $(z_k - H \times \hat{x}_k^-)$  is called measurement innovation. This difference reflects the difference between the predicted measurement and the actual measurement. The  $K$  matrix of dimension  $n \times m$  in the previous equation (??), is chosen to be the gain or the mixing factor which minimizes the covariance of the posterior error. The gain of the Kalman filter is of the following form (23):

$$K_k = P_k^- \times H^T \times (H \times P_k^- \times H^T + R)^{-1} \quad (23)$$

The use of a Kalman filter then allows us to estimate the parameters for tracking objects. However, the Kalman filter does not allow the moving element or these parameters to be extracted in the frame. We will first propose a method for detecting the moving object in the frame or video sequence. The fact that we used a robust, reliable and precise tracking algorithm greatly helped us to extract the two measurement and state vectors for the initialization of the Kalman filter (the inputs of the Kalman filter).

### 3. Different function of the Kalman filter

The Kalman filter is an optimal recursive estimator to the linear filtering problem Data. This filter has two necessary modules, a prediction module and a correction or estimation module. The Kalman filter then makes it possible to estimate the position of the object by achieving a compromise between the position observed in the frame and the predicted position. The input parameters of the Kalman filter are respectively, the position of the object in the frame at time “k,” the size of the object and the width and length of the object search window which are variable due to the mobility of the object during the sequence. These parameters represent the state vector and the filter measurement vector from Kalman filter. From the works that are studied in the literature [1, 4, 5], we chose the Kalman filter for estimating the tracking parameters. **Figure 4** shows the Kalman filter cycle [8–10]. Generally the estimation of the tracking parameters with a Kalman filter is a process requires the following steps: itemize.

The measure which consists in taking the tracking parameters computed in the Camshift algorithm.

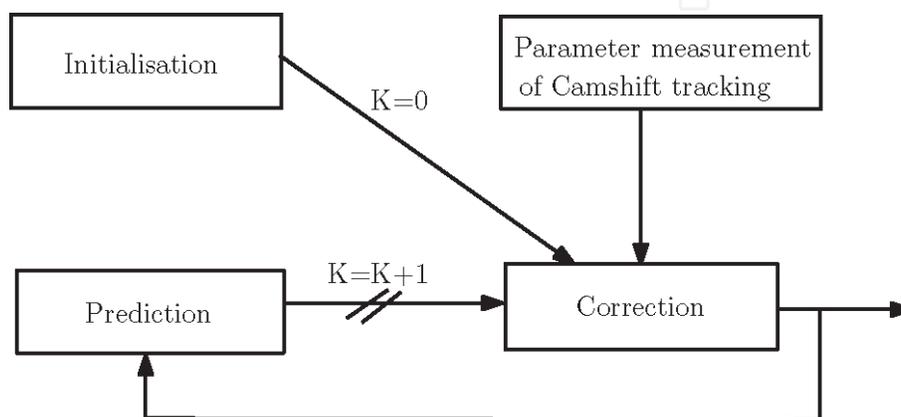
The estimate, which updates the position of the object.

The prediction, which computes the position of the object in the next frame.

The variable parameters of the Kalman filter are the state vector and the measurement vector:

The state vector is composed by the initial position, the width and the length of the search window as well as the center of gravity of the object  $(x_c; y_c)$  at time  $t_k$ . This vector is presented by the following Eq. (24):

$$s_k = (x_k; y_k; W_k; L_k; x_c; y_c) \quad (24)$$



**Figure 4.**  
Kalman filter cycle.

The measurement vector of the Kalman filter is composed of the initial position, the length and the width of the search window for the object at time  $t_k$ . This vector is given by the Eq. (25):

$$z_k = (x_k; y_k; W_k; L_k) \quad (25)$$

### 3.1 Process to estimate

The Kalman filter estimates the state “s” of a discrete process, this state is modeled by the linear Eq. (26):

$$s_k = A \times s_{k-1} + w_{k-1} \quad (26)$$

With “A” (27) is the transition matrix,  $w_k$  is the process noise and  $dt$  represents the difference between the two instants  $k$  and  $k - 1$  ( $dt = 1$ ).

$$A = \begin{pmatrix} 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (27)$$

- The measurement model is defined by the Eq. (28).

$$z_k = H \times s_k + v_k \quad (28)$$

With  $H$  (29) presents the measurement matrix:

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (29)$$

The two vectors  $s_k$  and  $z_k$  present the state and the measure at the moment  $k$ ,  $N$  is the integer vector. Process noise “ $w_{k-1}$ ” and measurement “ $v_k$ ” are assumed to be independent of the state and measurement vectors and to the normal and white distributions which are presented by Eqs. (30) and (31) [9, 10]:

$$p(w) \sim N(0, Q) \quad (30)$$

$$p(v) \sim N(0, R) \quad (31)$$

The noise process is of the following form (32):

$$w_{k-1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (32)$$

The measurement noise is presented by the dimension matrix ( $4 \times 1$ ) (33):

$$v_k = \begin{pmatrix} 0.1 \\ 0.1 \\ 0 \\ 0 \end{pmatrix} \quad (33)$$

So the noise and measurement process covariances are deduced from  $w_{k-1}$  and  $v_k$  by matrices (34) and (35):

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (34)$$

$$R = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (35)$$

### 3.2 The update equations

Finally, the output equations for the two prediction and correction blocks of the Kalman filter are:

- Prediction Eqs. (36) and (37):

$$\hat{s}_k = A \times \hat{s}_{k-1} \quad (36)$$

$$P_k^- = A \times P_{k-1} \times A^T + Q \quad (37)$$

- Correction Eqs. (38), (39) and (40):

$$K_k = P_k^- \times H^T \times (H \times P_k^- \times H^T + R)^{-1} \quad (38)$$

$$\hat{s}_k = \hat{s}_k^- + K_k \times (z_k - H \times \hat{s}_k^-) \quad (39)$$

$$P_k = P_k^- - K_k \times H \times P_k^- \quad (40)$$

With  $K_k$  presents the gain of the Kalman filter at the moment  $k$ ,  $\hat{s}_k$  the state estimated and predicted at the moment  $k$  and  $P_k$  is the prediction covariance matrix at time  $k$ . These three Eqs. (38), (39) and (40) present the output parameters of the Kalman filter. To verify the performance and results of the Kalman filter for estimation the parameters of the object tracking system. We compared the state vector values  $(x_c, y_c, W, L, x, y)$  for the sequence video "Foreman," computed for this filter, with the state vector values obtained by the tracking algorithm. These values are grouped in **Table 1**. It can be seen that the Kalman filter has good values of the state vectors, very close to those obtained by tracking algorithm. This proves the efficiency of the Kalman filter in the estimation of state vectors for tracking objects, the same for measurement vectors.

NFr	x	y	W	L	$x_c$	$y_c$	$x_{KF}$	$y_{KF}$	$W_{KF}$	$L_{KF}$	$x_{cKF}$	$y_{cKF}$
2	52	63	58	71	87.35	91.50	69	53	53	63	87.19	91.49
10	55	67	58	71	90.62	96.41	69	55	55	67	89.80	95.68
15	55	68	58	71	90.05	96.93	55	69	55	68	89.58	96.81
25	54	65	58	71	89.52	94.20	56	65	69	56	90.04	93.27
35	54	55	58	71	89.18	83.92	55	55	69	55	89.28	83.72
45	48	56	58	71	83.45	85.26	49	57	69	49	83.36	85.37
55	46	63	58	71	81.56	92.43	46	64	69	46	80.97	92.44
65	43	65	58	71	78.52	93.53	44	69	69	44	78.49	93.37
75	43	63	58	71	78.73	92.46	43	64	69	43	77.98	92.59
85	36	57	58	71	71.41	85.59	37	57	69	37	71.03	85.34
99	38	62	58	71	73.80	91.03	39	63	69	39	73.91	91.26

**Table 1.**  
 The results of the state vectors computed by the tracking algorithm (block-matching and Camshift) and the Kalman filter.

Video	Nbre	Histogram	Subtraction	Skin color
Sequence	Frames	Calculation	Background	Detection
		Execution time	Execution time	Execution time
Foreman	68	16.51 (s)	×	5.21 (s)
Redcup	68	19.21 (s)	×	×
Afef	68	17.02 (s)	×	10.58(s)
PETS2001(1)	68	43.54 (s)	36.56 (s)	×
PETS2001(2)	68	47.83 (s)	43.29 (s)	×

**Table 2.**  
 Precision values calculated for skin detection method and execution times.

The expression for the estimation error is presented by Eq. (41). We applied this Eq. (41) for all the parameters of the state vectors ( $x$ ,  $y$ ,  $W$ ,  $L$ ,  $x_c$ ,  $y_c$ ) of our implemented algorithm for tracking object in a video sequence (Camshift algorithm and Kalman filter algorithm):

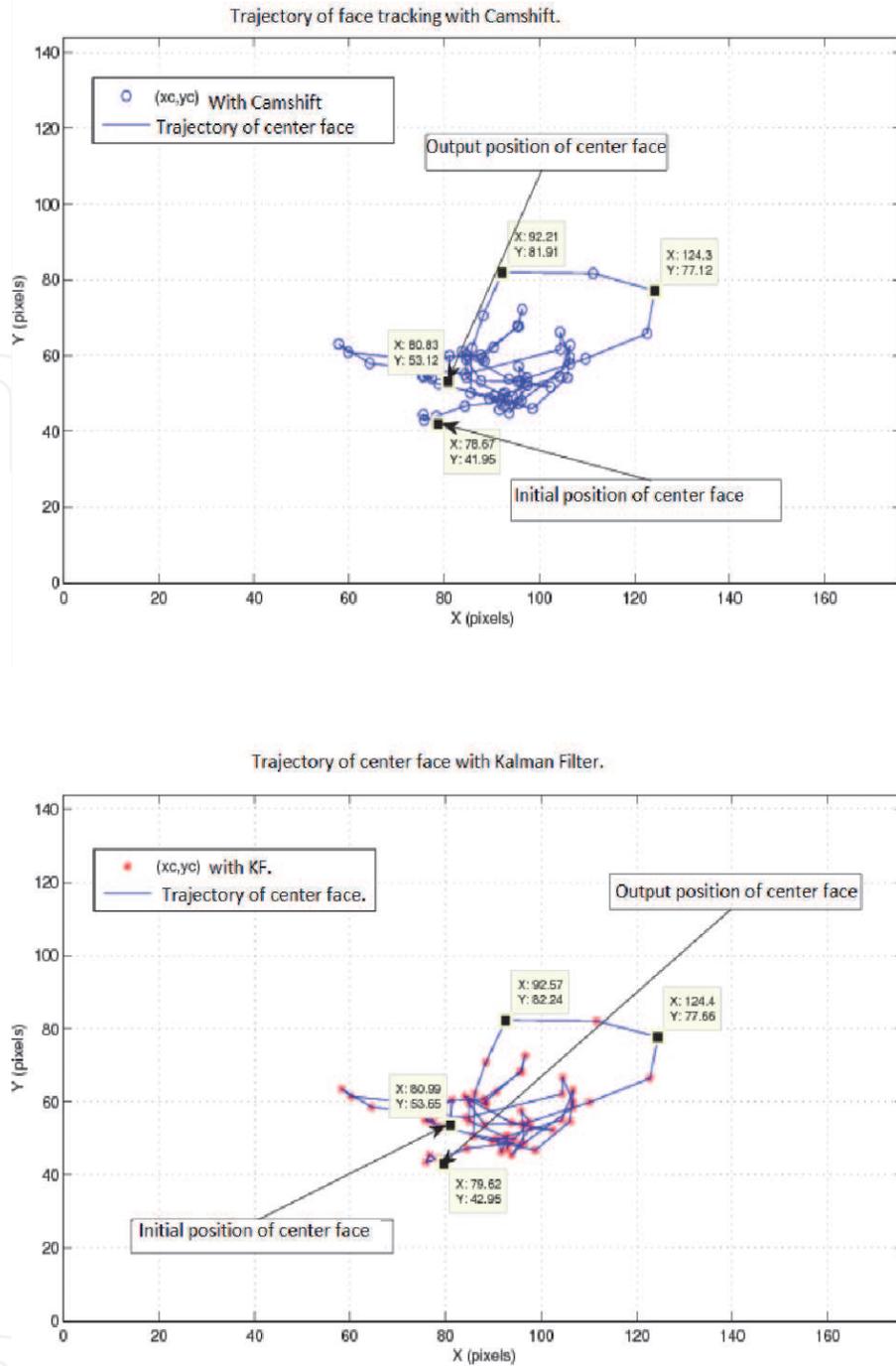
$$E_x = x_{Kalman} - x_{Camshift} \quad (41)$$

The results of estimation calculation of the tracking parameters with our algorithm (Camshift and Kalman filter) that we obtain in the different test sequence are given in **Table 2**.

We can see the two pre-treatment methods (background subtraction and skin color detection) are the fastest and the histogram calculation method is the slowest.

#### 4. The results for tracking and detection objects

The fundamental basis for estimating the parameters of tracking by the Kalman filter consists in estimating the state vector  $s_k$  and the measurement vector  $z_k$ . These vectors are calculated by the Camshift algorithm. The parameters of the vectors are the center of



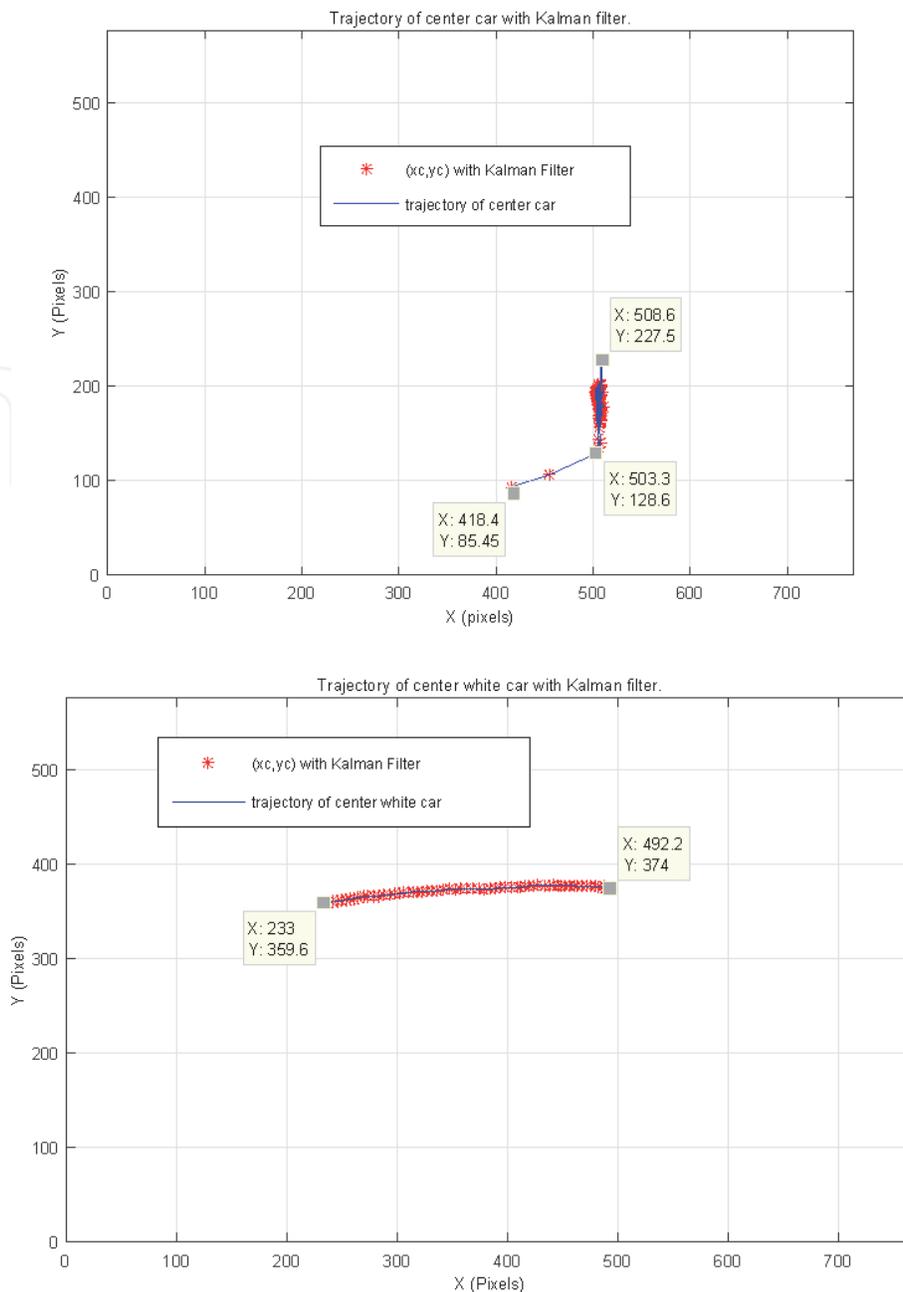
**Figure 5.** The trajectory of the gravity center of the face corrected by the Kalman filter.

gravity of the object  $(x_c, y_c)$ , the position in each image  $(x, y)$  and the width and length of the search window  $(L, W)$ . **Figure 4** shows the trajectory of the gravity center of the face estimated by the Kalman filter. We can see other results in our publications [1–3], there are presents the prediction and correction of trajectory of an different objects (human, car, glass, mono and multi-objects) in different environments (**Figure 5**).

We can see another correction of trajectory of car (in too sequences video) in **Figure 6**.

## 5. Results discussion

During the test sequences generated with the different pre-processing methods, we can conclude that object tracking differs from one object to another (a human



**Figure 6.**  
*The trajectory of the gravity center of the face corrected by the Kalman filter.*

being, a face, a hand, a glass, a car) and that several parameters can influence the monitoring result.

The experimental results obtained indicate that our algorithm (Camshift and the Kalman filter) gives superior results, in terms of precision, reliability and execution time, in comparison with the various methods presented in the literature (for example the KLT (Kanade Lucas Tomasi) algorithm and the classifier algorithm (Adaboost and SVM) [1–3, 7]). In particular, the use of several preprocessing methods to detect the object in each frame of the sequence. The results of the implemented algorithms are the meanshift displacement algorithm and block-matching, the Camshift algorithm and the Kalman filter, this combination for the this algorithms give a robust, precise, reliable and fast algorithm.

Evaluating the performance of a mobile object tracking system in a video sequence is a complex task which requires the definition of metrics bringing into play concepts specific to video analysis, such as time persistence, precision and execution time for example.

## 6. Conclusions

The detection and tracking of objects in a sequence of images or video is a topical need for several applications such as video conferencing, video indexing and especially video surveillance. Computer vision with a Human Interface Machine “HIM” is therefore an issue actively studied in many domains, especially since the prices of acquisition and processing equipment have become more attractive. This is an area that touches on everything, starting from the problems of acquisition with different linked effects and where the originality of simple ideas can still bring a lot. In this chapter, we introduced the Kalman filter algorithm for tracking and detection objects and multi-objects. Localization, target tracking, and detection objects were provided as examples for reader’s better understanding of practical usage of the Kalman filters. We proceeded to the implementation of the different modules of object tracking algorithm through the estimation of calculation parameters using a Kalman filter. The results obtained make it possible to meet the monitoring requirements of several video surveillance applications. On the one hand, the localization precision achieved by our system makes it a standard module for detection or identification or object tracking systems. On the other hand, a flow at a frequency of 20 frames per second was considered, which is reasonable for an object tracking system with a minimum execution time. The tracking algorithm with its different modules must be tested with other video sequences. Although the implementation of monitoring systems has certain weaknesses, our method has given promising results. Many avenues can be envisaged to continue this work. First of all, note that we tested the algorithm implemented for tracking two objects (a car and a pedestrian in the sequence of “PETS 2001 (1)” and two cars in the sequence of “PETS 2001 (2)”), and it can be applied for tracking multiple objects in a video sequence. Then, use the detection algorithm based on Adaboost classifiers upstream of the tracking algorithm (Camshift and Kalman filter). The association of these two modules is based on a cascade of Adaboost classifiers, improves the calculation time and improves the quality of tracking of one or more objects in a sequence of images or video. Then, validation of the detection and tracking system for faces and other objects (pedestrians, cars, hand gestures, glass, etc.) on an FPGA target platform (Saber-Lite with ARM-Cortex-A9MP). Our solution optimize the time of execution and other criteria in frame and video processing. In future, we intend to extensively evaluate the method quantitatively so that it can be well tested before trying on computer vision practice.

IntechOpen

### **Author details**

Afef Salhi<sup>1\*</sup>, Fahmi Ghozzi<sup>1,2</sup> and Ahmed Fakhfakh<sup>1,2</sup>

1 Digital Research Center of SFAX (CRNS), Laboratory of Technology for Smart Systems (LT2S), Sfax, Tunisia

2 ENET'COM, University of Sfax, Sfax, Tunisia

\*Address all correspondence to: [salhiafefge@gmail.com](mailto:salhiafefge@gmail.com); [afef.salhi.ge@enis.tn](mailto:afef.salhi.ge@enis.tn)

### **IntechOpen**

---

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Salhi A, Moresly Y, Ghozzi F, Yengui A, Fakhfakh A. Modeling from an object and multi-object tracking system. In: 2016 Global Summit on Computer Information Technology (GSCIT); 16–18 July 2016. Vol. 1. Sousse-Tunisia; 2016. pp. 80-85
- [2] Salhi A, Moresly Y, Ghozzi F, Fakhfakh A. Face detection and tracking system with block-matching, Meanshift and Camshift algorithms and Kalman filter. In: 18th International Conference on Sciences and Techniques of Automatic Control Computer Engineering; 21–23 December 2017. Monastir-Tunisia: IEEE; 2017. pp. 139-145
- [3] Salhi A, Ghozzi F, Fakhfakh A. Toward a methodology for object tracking system in computer vision. In: 14 th Tunisia-Japan Symposium on Science, Society Technology TJASSST'17; 24–26 November 2017. Gammarth-Tunisia; 2017. pp. 185-187
- [4] Nor Nadirah AA, Mostafah YM, Shafie AW, Zainuddin NA, Rashidan MA. Features-based moving objects tracking for smart video surveillances: A review. *International Journal on Artificial Intelligence Tools*. 2018;27(2):1830001
- [5] Chavda HK, Dhamecha M. Moving object tracking using PTZ camera in video surveillance system. In: International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017). IEEE; 2017. pp. 263-266
- [6] Mueller TM, Karasev P, Kolesov I, Tannenbaum A. Optical flow estimation for flame detection in videos. In: *IEEE Transactions on Image Processing*. Vol. 22. 2013. pp. 1-6
- [7] Salhi A. Study and implementation a system tracking and detection objects in video sequence. In: 2011 at National Engineering School of Sfax “ENIS”. Sfax-Tunisia; 21 July 2011. pp. 1-100
- [8] Deepak P, Krishnakumar S, Suresh S. Human recognition for surveillance systems using bounding box. In: International Conference on Contemporary Computing and Informatics Conference (IC3I). IEEE; 2014. pp. 852-856
- [9] Sakai Y, Oda T, Ikeda M, Barolli L. An object tracking system based on SIFT and SURF feature extraction methods. In: 18th International Conference on Network-Based Information Systems (ICN-BIS), 978-1-4799-9942-2/15-CPS. IEEE; 2015. pp. 561-565. DOI: 10.1109/NBiS.2015.121
- [10] Du D, Qi Y, Yu H, Yang Y, Duan K, Lu G, et al. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. Switzerland: Springer Nature; 2018. p. 375391. DOI: 10.1007/978-3-030-01249-623