International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:6, No:4, 2012

Object tracking system using Camshift, Meanshift and Kalman filter

Afef SALHI and Ameni YENGUI JAMMOUSSI

Abstract—This paper presents a implementation of an object tracking system in a video sequence. This object tracking is an important task in many vision applications. The main steps in video analysis are two: detection of interesting moving objects and tracking of such objects from frame to frame. In a similar vein, most tracking algorithms use pre-specified methods for preprocessing. In our work, we have implemented several object tracking algorithms (Meanshift, Camshift, Kalman filter) with different preprocessing methods. Then, we have evaluated the performance of these algorithms for different video sequences. The obtained results have shown good performances according to the degree of applicability and evaluation criteria.

Keywords—Tracking, Meanshift, Camshift, Kalman filter, Evaluation.

I. Introduction

Object tracking is an important research topic in computer vision. The rapid increase in numbers of great powered computers, the avaibility of high quality and high precision and cheap video cameras. Consequently, the use of object tracking (person, face, hand, glass, car) is in the tasks of automobile driver assistance, vehicle navigation, robotics, human-computer interaction, video surveillance, biometric, video games and industrial automation and security. Most of these applications require reliable object tracking techniques that meet in real-time constraints. The object tracking in a video sequence can be defined the dynamic entities that constantly change under the influence of several factors. In the physical world, there are five parameters to be considered: changes of the appearance, illumination, scale, object, and movement of the object (fast or slow). The variation of one of these parameters can influence the results of our tracking algorithm (the change of illumination, the change of scale, the change in appearance, the change of motion of the object and the change of the object). A huge number of tracking methods which have been proposed in earlier researches. In a tracking video sequence, an object can defined as anything that is interesting for analysis. For example, people walking on a road, planes in the air, cars in the road, and hand or a face in motion, etc. In recent researches, the form and appearance representations are classified into three families as the representation by points, representation by bounding boxes and the object silhouettes and contour [4], [16]. These methods for object tracking in video sequence and the feature selection for tracking are applied for many tracking algorithms in the earlier researches. Selecting the right feature plays a critical role in the tracking object in video sequence. Feature selection is closely related to the object representation. The object motion analysis usually focuses on simple characteristics such as texture, color, shape, geometry, etc. In general, many tracking algorithms use a combination of these features. In general, these features are selected manually by the user, depending on the application domain. However, the problem of automatic feature selection has received considerable attention in pattern recognition, namely the detection of objects to achieve tracking in video sequences. Tracking is often the first step in an analysis of activities, interactions and relationships between objects of interest. Many algorithms of object tracking have been proposed and developed. Tracking is the estimation and analysis of trajectories of an object in the plane of the image by moving through a sequence of images or video. These include for example when the main existing algorithms in the literature as the block-matching, KLT, the Kalman filter [6], [5] , Meanshift [2], [7] and Camshift [11], [9], [10].

The rest of paper is organized as follow: section 2 presents the object tracking system. In section 3, we describe the principle methods of preprocessing, to detect the exact position of the object in the images of the sequence analyzed and the experimental results and shows the performance of our object tracking in a video sequence. In section 4 summarizes our work and outlines potential further improvements.

Computers Imaging Electronics and Systems Group(CIELS)
Research unit on Intelligent Control, design and Optimisation of complex Systems (ICOS)
University of Sfax, Sfax Engineering School, BP W, 3038 Sfax, Tunisia
Phone: (216-74) 274.088, Fax. (216-74) 275.595.
Emails:salhiafefge@gmail.com.

II. Object tracking system

To ensure good organization the progress of work, we used the benefits of modular design in our algorithm implemented using MATLAB. The goal of an object tracking is to generate the trajectory of an object over time by discovering its exact position in every frame of the video sequence. we have implemented several object tracking algorithms (Meanshift, Camshift, Kalman filter) with different preprocessing methods. The algorithm for object tracking is composed of three modules: selection object module in the first frame of the sequence, the module of Meanshift algorithm and the module of Camshift algorithm. The selection module selects the position of the object in the first frame. It consists of extracting the module initialization parameters that are moving through the position, size, width, length of the search window of the object in the first frame of the sequence. The step of object tracking system are shown in figure 1.

A. Meanshift

The meanshift algorithm is a non-parametric method [13], [2], [7]. It provides accurate localization and efficient matching

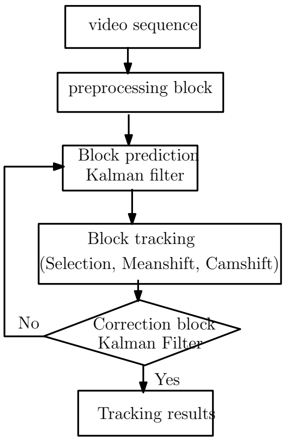

Fig. 1. flow chart of the object tracking system.

without expensive exhaustive search. The size of the window of search is fixed. It is an iterative process, that is to say, first compute the meanshift value for the current point position, then move the point to its meanshift value as the new position, then compute the meanshift until it fulfill certain condition. For an frame, we use the distribution of the levels of grey which gives the description of the shape and we are going to converge on the centre of mass of the object calculated by means of moments. The flow chart of meanshift in figure 2 described the steps of the algorithm. The number of iterations of the convergence of the algorithm is obtained when the subject is followed within the image sequence [9].

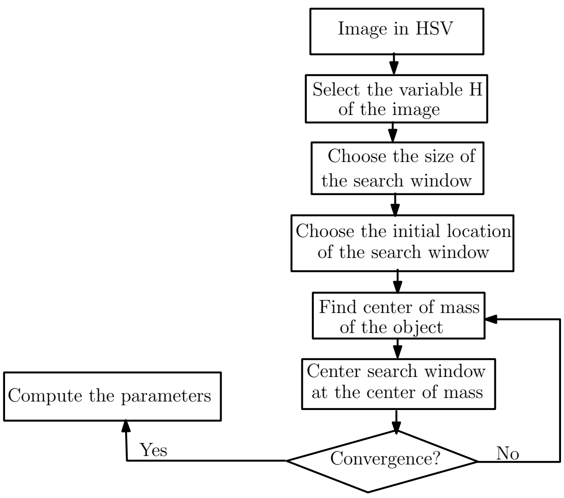

Fig. 2. flow chart of Meanshift algorithm.

1) Calculation of parameters:: The inputs of this module are the initial position, width and length of the initial search window of the object. Let $I(x, y)$ the image of "$2D$", (x, y) is the position of the object in the image. The zero-order moment is given by equation (1), an order moments spatial and the center of mass are given by equations (2) and (3). These parameters are the output of Meanshift algorithm module and the input of Camshift algorithm module. The moment of order zero (1) represents the area occupied by the shape of the frame.

$$M00 = \sum_x \sum_y I(x, y) \quad (1)$$

The moments of order one ($M10$,$M01$) are calculated by equations (2).

$$M10 = \sum_x \sum_y x \times I(x, y); M01 = \sum_x \sum_y y \times I(x, y) \quad (2)$$

We find the centre of mass of the object by means of the moments of zero order and one, (xc, yc) is expressed by the equation (3).

$$xc = \frac{M10}{M00}; yc = \frac{M01}{M00} \quad (3)$$

B. Camshift

The principle of the CamShift algorithm is given in [12], [9], [10], [11], is based on the principles of the algorithm Meanshift. Camshift is able to handle the dynamic distribution by adjusting the size of the search window for the next frame based on the moment zero of the current distribution of images. In contrast to the algorithm Meanshift who is conceived for the static distributions, Camshift is conceived for a dynamic evolution of the distributions. It adjusts the size of searching window by invariant moments. This allows the algorithm to anticipate the movement of objects to quickly track the object in the next frame. Even during the fast movements of an object, Camshift is still capable of tracking well. It occurs when objects in video sequences are tracked and the object moves such that the size and location of the change in probability distribution changes in time. The initial search window was determined by a detection algorithm or software dedicated to video processing. The CamShift algorithm calls upon the MeanShift one to calculate the target centre in the probability distribution image, but also the orientation of the principal axis and dimensions of the probability distribution. Defining the first and second moments for x and y. These parameters are given from the first and second moments, are defined by equations (4, 5 et 6).

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

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

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

The orientation of the major axis and the scale of the distribution are determined by finding an equivalent rectangle that has the same moments as those measured from the 2D probability distribution image. The orientation is defined by the equation (7).

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

The first two eigenvalues (the length and width of the probability distribution) are calculated in closed form as follows in the equations (8), (9) and (10).

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

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

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

The length and width of the distribution around the centroid is then given by equations (11) and (12).

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

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

C. Block Kalman filter

The Kalman filter estimates the position of the object in each frame of the sequence [6], [5]. The input parameters of the Kalman filter, respectively, the position of the object in the image at time k, the size of the object and the width and length of the search window of the object which vary due to the mobility of the object during the sequence. These parameters represent the state vector and measurement vector of the Kalman filter. In general, the estimation of parameters followed with a Kalman filter is a process that requires the following steps:

- The measure is to take the tracking parameters calculated by the algorithm Camshift.
- The estimate, which updates the position of the object.
- The prediction, which calculates the position of the object in the next frame.

The variable parameters of the Kalman filter are the state vector and measurement vector: The state vector is composed of the initial position, width and length of the search window and the center of mass of the object (xc, yc) at time tk . This vector is presented by equation (13).

$$s_k = (x_k, y_k, W_k, L_k, x_c, y_c) \quad (13)$$

The measurement vector of the Kalman filter is composed of the initial position, length and width of the search window of the object at time tk. This vector is given by equation (14).

$$z_k = (x_k, y_k, W_k, L_k) \quad (14)$$

1) Process to estimate : The Kalman filter estimates the state s a discrete process, this state is modeled by the linear equation (15).

$$s_k = A \times s_{k-1} + w_{k-1} \quad (15)$$

With A (16) is the transition matrix, wk is the noise process and dt is the difference between the two moments k and $k - 1 (dt = 1)$.

$$A = \begin{pmatrix} 1 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (16)$$

The measurement model is defined by equation (17).

$$z_k = H \times s_k + v_k \quad (17)$$

With H (18) is the measurement matrix:

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (18)$$

Noise process "w_{k-1}" and as "v_k" are assumed independent of state vectors and measurement and normal distributions and white that are presented by equations (19) et (20):

$$p(w) \sim N(0, Q) \quad (19)$$

$$p(v) \sim N(0, R) \quad (20)$$

The process noise is of the form (21):

$$w_{k-1} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (21)$$

The measurement noise is presented by the matrix of dimension (4×1) (22):

$$v_k = \begin{pmatrix} 0.1 \\ 0.1 \\ 0 \\ 0 \end{pmatrix} \quad (22)$$

So the covariance of process noise and measurement are deducted from w_{k-1} and v_k by the matrices (23) et (24):

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (23)$$

$$R = \begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (24)$$

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

- Equations for predicting (25), (26):

$$\hat{s}_k = A \times \hat{s}_{k-1} \quad (25)$$

$$P_k^- = A \times P_{k-1} \times A^T + Q \quad (26)$$

- Correction Equations (27), (28) and (29):

$$K_k = P_k^- \times H^T \times (H \times P_k^- \times H^T + R)^{-1} \quad (27)$$

$$\hat{s}_k = \hat{s}_k^- + K_k \times (z_k - H \times \hat{s}_k^-) \quad (28)$$

$$P_k = P_k^- - K_k \times H \times P_k^- \quad (29)$$

The expression of the estimation error is given by equation 30. We applied this equation 30 for all parameters of state vectors (x, y, W, L, x_c, y_c) algorithm implemented in our tracking of an object in a video sequence (Camshift algorithm and the algorithm of the Kalman filter):

$$E_x = x_{Kalman} - x_{Camshift} \quad (30)$$

The results of error calculation for estimating the parameters of our monitoring algorithm (Camshift and Kalman filter) we obtain the test sequence "Foreman" are given in table I.

TABLE I
THE RESULTS OF ERROR CALCULATION OF THE PARAMETERS OF STATE VECTORS CALCULATED.

N d'images	E(x)	E(y)	E(W)	E(L)	$E(x_c)$	$E(y_c)$
2	17	-10	-5	-8	-0.157	-0.0162
10	14	-12	-3	-4	-0.8106	-0.7257
15	0	1	-3	-3	-0.4723	-0.1218
25	2	0	11	-15	0.5163	-0.9389
35	1	0	11	-16	0.1011	-0.2007
45	1	1	11	-22	-0.0832	0.1103
55	0	-17	11	-25	-0.5864	0.0067
65	1	4	11	-27	-0.0258	-0.1601
75	0	6	11	-28	-0.7492	0.1237
85	1	-1	11	-34	-0.389	-0.2474
99	1	1	11	-32	0.1073	0.2374

III. EXPERIMENTAL RESULTS

We implemented the proposed algorithm in MATLAB, and tested it on five sequence video. The tests video sequences was a public in Internet, which can be downloaded from [22], [21], [20], [19]. This video has been adopted by many researchers to test their algorithms, because of its capacity in simulating various tracking conditions, including illumination changes, pose variations, occlusions, and distraction. Since ensemble tracking is a general framework for tracking objects, several object tracking algorithms (Meanshift, Camshift, Kalman filter), and a pre-specified methods for preprocessing (method of histogram color [9], method of background subtraction [1], [9], and method of detection skin color [18], [17]) are evaluated to achieve a good tracking performance. An adaptive first pre-processing the histogram color was tested to the video sequences. The algorithm tracks the object as it moves from one frame to the next one in approximately 0.15 second on a 2.2GHz PC in Matlab. The first expriment is on a video sequence of glass in hand man with 100 frames of spatial resolution 320×240. The tracking target is the moving is the glass. The second experiment is on a sequence with 100 frames of spatial resolution 144×176. In this video, we will track a motion of face. To save space, we only show the experimental results preprocessing and tracking object in three sequence an by the proposed method in Figure 3, 4. As shown in Figure 5, the trajectories of the glass for the three algorithms tracking by the Camshift algorithm and the estimated trajet location of the Kalman filter in the frame plane.

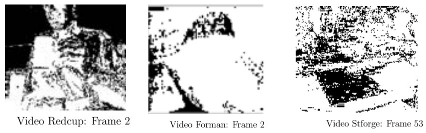

Fig. 3. Preprocessing results of 3 sequences video by the histogram method.

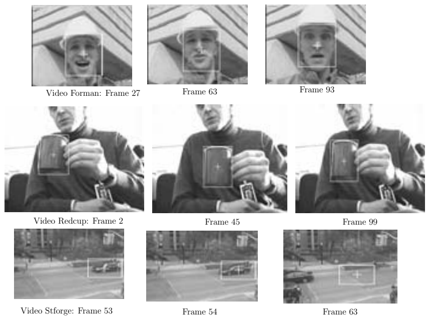

Fig. 4. Tracking results of 3 sequences video by the Histogram methods.

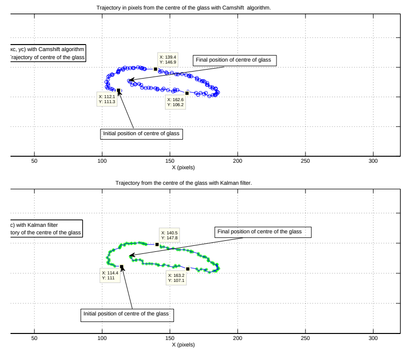

Fig. 5. Trajectory in pixels from the centre of the glass.

Table II shows the experimental results obtained with the tests sequences by the proposed method preprocessing of histogram. Thus precision and execution time ratio were obtained comparing to those for the tests video sequence. The experimental results show that the proposed method tracks the face and the glass more reliably and more accurately but the car information is lost thereby leading to failure tracking. I remedied the problems tracking a moving car using the second method of background subtraction. TableIII shows measurements of precision calculated and execution time for test sequences of our tracking algorithm using as preprocessing method of background subtraction. The Figure 6 show tracking

TABLE II
THE PRECISION VALUES CALCULATED FOR THE HISTOGRAM METHOD.

video	frames	Precision(%)	execution time(s)
Redcup	100	99	24.408675
Foreman	100	96	15.938718
PETS 20011	100	44	118.523104
PETS 20012	100	4	145.519240
Stforge	10	3	10.104903

process of the moving car obtained in this method. The third

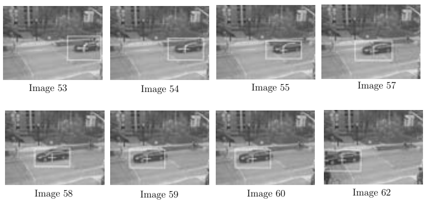

Fig. 6. Tracking results of the moving car by the background substraction.

TABLE III
PRECISION VALUES CALCULATED FOR THE METHOD OF BACKGROUND SUBTRACTION AND EXECUTION TIME.

video	frames	Precision (%)	execution time(s)
PETS 20011	100	99	79.931015
PETS 20012	133	88,72	93.119679
Stforge	10	10	7.476155

proposed method preprocessing is the detection of human skin color by RGB space. This method is what determines that an image contains faces or not and if any is to return their size and position in the image. The detection is based on the RGB color space. Detection in this case requires no skin model and no conversion of color space, it checks if just the triple (R, G, B) is a skin color or no-skin color. The proposed method successfully tracks the face over the whole sequence of Forman. The precision value and the time execution are respectively 97,05%, 17.021 secondes. Table IV lists the execution time corresponding to the different preprocessing used with our algorithm object tracking for all video sequences analyzed. We note that the two pre-processing

TABLE IV
THE EXECUTION TIME FOR DIFFERENT METHOD PRE-PROCESSING.

Sequence Video	frames	Histogram execution time	Background time	skin color time
Foreman	100	18.558961 (s)	×	17.021(s)
Redcup	100	21.113581 (s)	×	×
PETS 20011	100	118.523104(s)	79.931015(s)	×
PETS 20012	133	145.519240 (s)	93.119679(s)	×

methods (background subtraction and detection of skin color) are the fastest and the calculation of a histogram is the slowest.

IV. CONCLUSION

This paper focuses on simultaneous tracking of multiple object (a human,a face, a hand, a glass, a car). During the test sequences generated with different methods of pre-processing, we can conclude that the tracking of objects differs from one object to another and several parameters can affect the results of tracking. Experimental results show that our algorithm (Camshift and the Kalman filter) is superior in terms of precision, reliability and execution time, compared the various methods presented in the literature. In particular, the use of several methods preprocessing to detect the object in each frame of the sequence, provides satisfactory results in the case of the complex video sequence (illumination change, change in appearance, scale, change of subject and object movement).

REFERENCES

[1] A. Alper and J. Omar and S. Mubarak, " *Object Tracking: A Survey*". ACM Computing Surveys, vol. 38, no. 45, pp. 1–45. 2006.
[2] D. Chen and F. Bai and B. Li and T. Wang, " *BEST: A Real-time Tracking Method for Scout Robot*". IEEE/RSJ International Conference on Intelligent Robots and Systems, no. 6, pp. 1001–1006. 2009.
[3] J. Wang and F. He and X. Zhang and G. Yun, " *Tracking Object throuth Occuusions Using Improved Kalman Filter*". IEEE 2010, no. 6, pp. 223-228. 2010.
[4] N.A Gmez, " *A Probabilistic Integrated Object Recognition and Tracking Framework for Video Sequences*". Universitat Rovira I Virgili, PHD thesis, Espagne, 2009.
[5] X. Sun and H. Yao, and S. Zhang, " *A Refined Particle Filter Method for Contour Tracking*". Visual Communications and Image Processing, no. 8, pp. 1–8. 2010.
[6] D. Comaniciu, and V. Ramesh, and P. Meer , " *Kernel-Based Object Tracking*". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25 no. 14, pp. 564–577. 2003.
[7] L. Ido, and L. Michael, and R. Ehud, " *Mean Shift tracking with multiple reference color histograms*". Computer Vision and Image Understanding, no. 9, pp. 400–408. 2010.
[8] G. bradski, and R. Gary, and L.M. Olga, " *Computer Vision Face Tracking For Use in a Perceptual User Interface*". Intel Technology Journal Q2'98, vol. 10, no. 15, pp. 1–15. 1998.
[9] G. Bradski, and T. Ogiuchi, and M. Higashikubo, " *Visual Tracking Algorithm using Pixel-Pair Feature*". International Conference on Pattern Recognition, no. 4, pp. 1808–1811. 2010.
[10] Y. Ruiguo, and Z. Xinrong, " *The Design and Implementation of Face Tracking in Real Time Multimedia Recording System*". IEEE Transaction, no. 3, pp. 1–3. 2009.
[11] E. David, and B. Erich, and K. Daniel, and S. Anselm, " *Fast and Robust Camshift Tracking*". IEEE Transaction, no. 8, pp. 1–8. 2010.
[12] S. Avidan, " *Support vector tracking*". In IEEE Conference on Computer Vision and Pattern Recognition, no. 8, pp.184–191. 2001.
[13] S. Franois, B. and R.J. Alexandre, " *Camshift Tracker Design Experiments*". IMSC, no. 11, pp. 1–11. 2004.
[14] L. Zhang, and D. Zhang, " *Robust Object Tracking Using Joint Color Texture Histogram*". International Journal of Pattern Recognition and Artificial Intelligence, vol. 23 no. 19, pp. 1245–1263. 2009.
[15] X. Song, and R. Nevatia, " *Camshift Tracker Design Experiments with Intel OpenCV and Sai*". International Conference on Pattern Recognition,IEEE Computer Society, no. 4, pp. 1–4. 2004.
[16] S. Afzal, and N. Vaswani, and A. Tannenbaum ,A. and Yezzi, " *Object Tracking: A Survey*". IEEE, no. 6, pp. 4244–4249. 2009.
[17] A. Iyad, and H. Mahmoud, " *Smart Human Face Detection System*". International Journal of Computers, vol. 5, no. 8, pp. 210–221. 2011.
[18] C. Abbas, and C. Joan, and C. Kevin, and M.K. Paul " *A Skin Tone Detection Algorithm for an Adaptive Approach to Steganography*". , no. 23, pp. 1–23. 2011.
[19] http://developer.com/technology/itj.
[20] http://ftp.pets.rdg.ac.uk/PETS2001/DATASET1/TESTING/.
[21] http://developer.com/technology/itj.
[22] http://ftp.pets.rdg.ac.uk/PETS2001/DATASET1/TESTING/.

SALHI Afef was born in Sfax, Tunisia in 1985. He received a degree in electrical engineering and Master degree in Control and Computer Engineer- ing from National School of Engineers of Sfax in 2009 and 2011. He is currently pursuing a thesis at the same university in Laboratory of Electronics and Information Technologies (L.E.T.I). His main research interests include Codesign software Hard- ware high level of MPSoCs (FPGA, DSP, GPP) for real time application, image and video processing,multimedia application, medical application, system tracking.was born in Sfax, Tunisia in 1985. He received a degree in electrical engineering and Master degree in Control and Computer Engineering from National School of Engineers of Sfax in 2009 and 2011. He is currently pursuing a thesis at the same university in Laboratory of Electronics and Information Technologies (L.E.T.I). His main research interests include Code- sign software Hardware high level of MPSoCs (FPGA, DSP, GPP) for real time application, image and video processing, multimedia application, medical application, system tracking.