Reliable and Fast Tracking of Faces under Varying Pose

Tao Yang[1] , Stan Z.Li[2], Quan Pan[1], Jing Li[1], Chunhui Zhao[1]
[1]*College of Automatic Control, Northwestern Polytechnical University, Xi'an, China , 710072*
[2]*Center for Biometrics and Security Research & National Laboratory of Pattern Recogntiion*
Institute of Automation, Chinese Academy of Sciences, Beijing, China, 100080
yangtaonwpu@163.com, szli@nlpr.ia.ac.cn , quanpan@nwpu.edu.cn, jinglinwpu@163.com

Abstract

This paper presents a system that is able to track multiple faces under varying pose (tilted and rotated) reliably in real-time. The system consists of two interactive modules. The first module performs detection of face subject to rotations. The second does online learning based face tracking. A mechanism of switching between the two modules is embedded into the system to automatically decide the best strategy for reliable tracking. The mechanism enables smooth transit between the detection and tracking module when one of them gives no results or unreliable results. Results demonstrate that the system can make reliable real-time tracking of multiple faces in complex background under out-of-plane rotation, up to 90 degree tilting, fast nonlinear motion, partial occlusion, large scale changes, and camera motion.

1. Introduction

Real-time object tracking in complex environment has many practical applications, such as visual surveillance and biometric identification, and is a challenging research topic in computer vision applications. Accurate and real-time face tracking will improve the performance of face recognition, human activity analysis and high-level event understanding.

Face detection and tracking have recently received much attention. The system of Toyama [1] made a successful face tracking that uses Incremental Focus of Attention (IFA), a state-based architecture which allows fast recovery of lost targets within a unified framework. Viola and Jones [2] use AdaBoost for face detection. This is related to an earlier work of Tieu and Viola [3] for boosting image retrieval. The systems have much advanced previous techniques in accuracy and achieve real-time performance. However, this work deals primarily with frontal faces.

The ability to detect and track faces of varying head pose (termed "multiview" faces hereafter) is important for many real applications. To solve this problem, Wang [4] propose a graphical model based method, which combines the factorial and the switching Hidden Markov Model(HMM). Feraud et al. [5] adopt the view based representation for face detection. Wiskott et al. [6] build elastic bunch graph templates for multiview face detection and tracking. Gong et al. [7] study the trajectories of faces in linear PCA feature spaces as they rotate, and use kernel support vector machines (SVMs) for multipose face detection and pose estimation [8]. Viola et al [9] train a decision tree to determine the viewpoint class. Li et al [10] propose FloatBoost algorithm and use a detector pyramid to handle rotated faces. Huang [11] develop a nested cascade detector for multiview face detection.

Many visual based tracking methods use some low level features such as color and contour to track objects including faces [12],[13],[14],[15]. Monte Carlo methods [13] adopt sampling techniques to model the posterior probability distribution of the object state and track objects through inference in the dynamical Bayesian network. A robust non-parametric technique, the mean shift algorithm, has also been proposed for visual tracking. In [14] human faces are tracked by projecting the face color distribution model onto the color frame and moving the search window to the mode (peak) of the probability distributions by climbing density gradients. In [15] tracking of non-rigid objects is done through finding the most probable target position by minimizing the metric based. Some other methods are presented to track human heads, for example, Birchfield [16] present an algorithm combining the intensity gradient and the color histogram.
Although much progress has been made in face tracking and detection, none of the existing algorithms and systems are able to handle multiple largely tilted and rotated faces reliably in real-time. The tacked faces can either be lost easily when the faces are tilted or rotated to a large degree or much occluded. Moreover, those algorithms are not fast enough to handle abrupt changes such as jumping and running, especially for non-frontal view faces. These limitations must be overcome for a wide range of real applications.

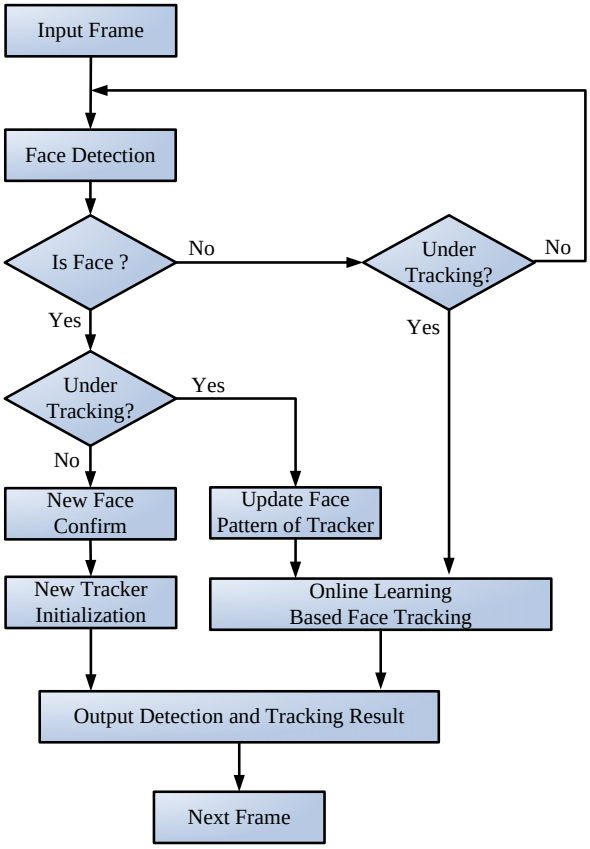

Figure 1. *Diagram of Real-time Face Tracking System.*

This paper presents a novel real-time face tracking system to solve the above problems. The system consists of two interactive modules: (1) a face detection module, and (2) an online learning based face tracking module. The advantage of the first module is the high accuracy of face detection in position and scale. However, it may fail with largely tilts and rotations. The second module could achieve real-time multiple faces tracking in various head poses, but it is sensitive with large face scale changes. To overcome the weaknesses of the two modules above, the system transition between them when one module gives no results or unreliable results.

Results demonstrate that the system can make reliable real-time tracking faces in video sequences under out-of-plane rotation, up to 90 degree tilting, partial occlusion, large scale changes, camera motion and multiple persons in complex background. The speed is 10~12 frames per second for images of size 320x240. A demo can be found at http://www.cbsr.ia.ac.cn/demos/FaceTrack.wmv .

The remainder of this paper is organized as follows: Section 2 introduces the diagram of the system. Section 3 presents online learning based face tracking module in detail. Section 4 discusses extensive results. Section 5 describes conclusion and future extension.

.

2. System Overview

The system consists of two interactive modules (Figure 1): (1) a face detection module, and (2) an online learning based face tracking module. The detection module incorporated the ideas from Viola [9] and Li [10] for detection of faces under rotations. The second module, tracking is performed by a dominant color feature selection method based on mean shift analysis. Different to other mean shift tracking methods based on minimize the distance between the kernel distribution for the object in the current frame and the model , our system learns the color distributions of the objects under track in an online mode, and computes the weight of each pixel by fusing the probabilities of the pixels in the tracked regions and the surrounding area. This way, salient color features of the faces can be selected automatically and dynamically for each frame, making the tracker robust even in complex environment.

The system is performed by interaction between the above two modules. For each input frame, the detection module is used to find all possible faces and update or initialize a new tracker of the tracking module. To reduce the influence of false detects (false alarms of the face detection module), we analyze the time-prints of each new face in consecutive frames, and use the result to remove noise. Once a new face is confirmed, it will be added to the objects list being tracked, and the color distribution of the face and its surrounding area are computed for the initialization of the tracker. If a tracked face does not be detected at a certain frame, for instance under rotation or partial occlusion, we will use the recorded face pattern to estimate the target position. Once a tracker got detection, its parameter such as scale, position and color distribution of face and surrounding will be updated by the detection result. To avoid the tracking problems, the tracker will be considered lost target if it isn't detected for several consecutive frames. The final output tracking result is the integration of the two modules.

3. Online Learning Based Face Tracking

The mean-shift algorithm is a nonparametric statistical method for seeking the nearest mode of a point sample distribution [15],[17],[18]. The algorithm has been adopted as an efficient technique for real-time object tracking. One of the key issues in mean shift algorithm is how to produce the sample weight image at time. In the sample weight image, the pixels on the object have high

weight, while pixels on the background have low weight. Any features that may separate the object form background can be used to produce the weight map. For instance, motion feature in moving object tracking with static camera, skin color model in face tracking, texture similarity and output of the correlation from a detection module such as a classifier.

The problem we address in the system is how to produce the sample weight image in which the face's weight is greatly higher than the weight of the dynamic scene in mean shift analysis. Although the pixel's motion characteristic can be used to separate the moving person and the static background, and thus improve the accuracy of the weight image, many of the existing motion segmentation algorithms are based on background subtraction technique with static camera. Considering the camera is active in many application fields, we prefer to build a system which has little constraints about the camera motion.

In the system, we take the color distribution of the object as the main feature. Typically, color feature based weight image is determined by computing the Bhattacharyya coefficient between the color histogram of the object model and the current mean shift window. Instead of using Bhattacharyya coefficient, we compute the weight image through fusion the probability of the pixel in the color distribution of the object model and the surroundings. Firstly, the color distribution of the face model is taken as the feature space. Then the color distribution of the surrounding in the current frame is used to select the dominate color in feature space for the next frame during the mean-shift tracking. There are mainly two advantages of the online learning based method: (1) It does not need compute the similarity coefficient between two color distribution and thus can achieve real-time object tracking even on common PC. (2) The online surrounding learning mechanism could make the tracking processing quite robust even in complex environment.

3.1 Candidate Initialization

Usually the face is represented by a certain region in the image, and its shape can be chosen as ellipse [16] or rectangle. In the system, the face is modeled as rectangle window and the size of the window is changed online, its state is defined as $S = \{x, y, w, h, d\}$, where (x, y) is the center of the window, (w, h) represents the width and height separately. d denotes the lost detection times of the face tracker recently. $I(x, y)$ represents the intensity value of the input image I at (x, y). During the tracking process, the state S is initialized by the face detection result. We use (1) to compute the height, and the face in the tracker is modeled as a rectangle with ratio α

$$h = \alpha \cdot w, \alpha \in [1,1.5] \quad (1)$$

Where w is the width of the detected face. α is an experimental variable of face model, and fixed as 1.2 in our system.

We use the output of face detection module to initial starting values x, y, w, h and $d0$. In the system, if a detected face doesn't overlap with any of the existing tracker, it will be assigned as a possible new face. If the detection times of the new face are higher than a threshold in the following frames, it will be confirmed and a new tracker will be initialized.

3.2. Dominant Feature Selection

The goal in feature selection is to find the dominant features in feature space, so as to produce the sample weight image in mean shift tracking. The two main components in feature selection are feature space creation and online feature selection. Considering the camera motion and dynamic background, the color cue is selected to build feature space. Without loss of generality, many color space such as RGB, HSV and YUV can be chosen as the feature space. Because we get almost the same tracking results with those color spaces above in our experiments, in this system the RGB color distribution of the object is chosen as the feature space. We build an RGB histogram with $N = Nr \cdot Ng \cdot Nb$ bins. Thus we have the object color model q_t

$$q_t = \{q_t(1), \ldots, q_t(n)\}, n = 1, \ldots, N \quad (2)$$

where $\sum_{n=1}^{N} q_t(n) = 1$

Then the observation model $M(k)$ can be defined as

$$M(k) = \sum_x \sum_y W_q^k(x, y) \quad (3)$$

Where W_q^k is the weight image with the object color model q at time k, and x, y range over the rectangle region. Given a pixel $I(x, y)$, let $P_t^k(x, y)$ denote its probability in the object color model. Instead of using Bhattacharyya coefficient , we compute the pixel weight at (x, y) from (4) without feature selection.

$$W_q^k(x, y) = P_t^k(x, y) \quad (4)$$

Because the mean shift iteration is based on observation model M , under the condition of equation (4), while the iteration stops, the candidate will move to the nearby highest modes of the observation density, where the pixels with high probabilities in the object color model.

Contrast to computing the Bhattacharyya coefficient between the color histogram of the object model and the current mean shift window, an advantage of (4) is that it's simple and fast to be implemented. However, because (4) only consider the probability of each pixel in the object color model, it may fail in complex environment which has similar color to the object model. To solve this problem, we develop an efficient feature selection method which continually using the color distribution of the surrounding area to select the dominant color feature from the object color model, the selected feature will be used to produce the weight image in mean shift tracking.

Considering the nonlinear motion of the object, we choose a large circle area around the current object position to estimate the color distribution of the background. Let q_t^k and q_b^k denote the color distribution of the object and its surrounding background separately, the distribution of dominant color feature of q_d^k is given below

$$q_d^k(n) = \begin{cases} \frac{1}{c}T & if \quad \frac{\max(q_t^k(n),\varepsilon)}{\max(q_b^k(n),\varepsilon)} > T \\ \frac{1}{c}\frac{\max(q_t^k(n),\varepsilon)}{\max(q_b^k(n),\varepsilon)} & if \quad T > \frac{\max(q_t^k(n),\varepsilon)}{\max(q_b^k(n),\varepsilon)} > 1 \\ 0 & otherwise \end{cases} \quad (5)$$

Where c is normalized coefficient, and ε is a small value to get rid of small probabilities and prevents dividing by zero. T is a threshold that prevents too high contrast between the target and the background model. Equation (4) can be modified as (6).

$$W_q^k(x,y) = P_d^k(x,y) \quad (6)$$

Where $P_d^k(x,y)$ denotes the pixel probability in the dominant color distribution q_d^k.

Figure 2 shows a sequence of dynamic weight map and face tracking result under large rotation and jumping. Here the cross in red shows the tracking result and pixel with high weight is displayed with high-luminance of green color.In Figure 2, a new face is confirmed (Figure 2, frame #11, green rectangle) and a tracker is initialized according to the face pattern. In the following frames, difficult conditions are included like out-of-plane rotation (Figure 2, frame #29, frame #61), highly nonlinear motion

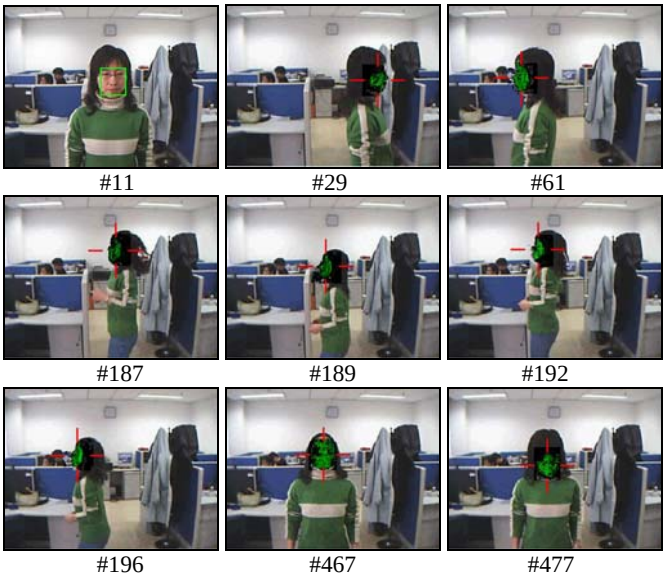

Figure 2. Dynamic weight map and face tracking result under large rotation and jumping. The green rectangle (frame #11) shows detected new face. The cross in red shows the tracking result. Pixel with high weight is displayed with high-luminance of green color. Note that after the feature selection with equation (5), the similar color of the face and the surrounding (Hair) will be punished and particular color of the face (Skin) will be encouraged.

like jumping (Figure 2, frame#61, frame #187, frame #192, frame #196), and up to 90 degree tilting (Figure 2, frame #467, frame #477). The online learning based tracker successfully handled those difficult conditions above in real-time.

Note that although the person's hair is in the searching window of mean-shift tracker (Figure 2, frame # 61), after the feature selection with equation (5), the similar color of the face and the surrounding will be punished and particular color of the face will be encouraged. As a result, pixels with discriminate color between the face and surrounding have set with high weight, and the hair inside the searching window is signed with low weight. Thus the mean shift tracker will be more robust in such weighted map.

During the tracking process, if a tracked face does not be detected at a certain frame, the size of it's rectangle window is constant for simplicity in the online learning based mean-shift analysis. Once a tracker got detection, its parameter such as scale, position and color distribution of face and surrounding will be updated by the detection result. The mean shift iteration is based on dominant feature density, while the iteration stops, the Kullback-Liebler (KL) distance D^k will be used to estimate the

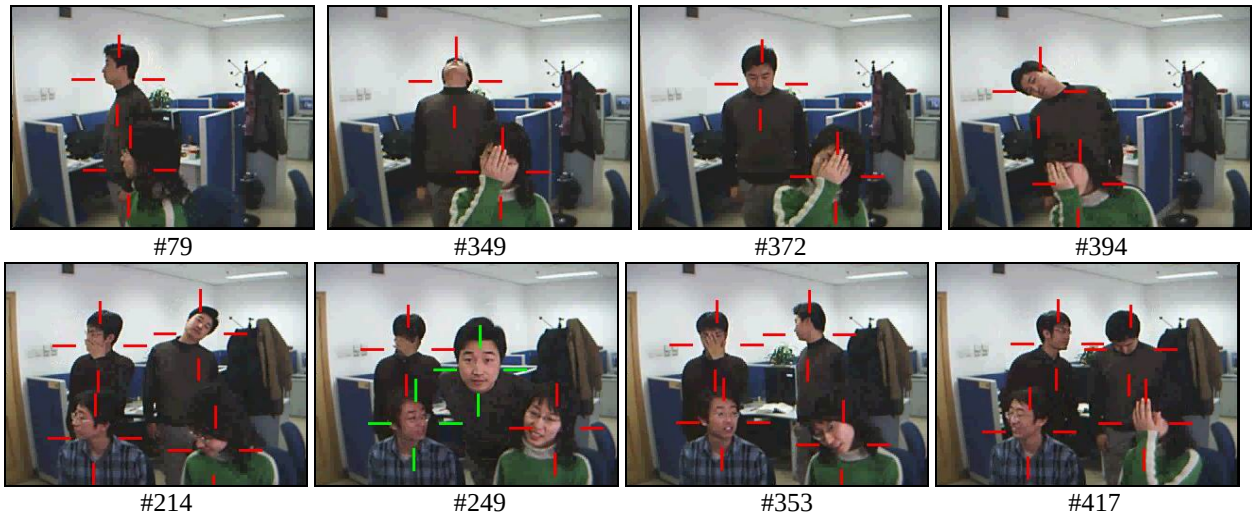

Figure 3. Real-time multiple faces tracking in indoor environment. The first and second rows contain a sequence of tracking result with two persons. The third row contains a sequence of tracking result with four persons. The cross shows the position of the person's face. The cross in green represents the face is detected by the face detector. The cross in red represents the output of online learning based tracker.

Figure 4. A sequence of two interacting persons tracking in indoor environment with an active camera. Note that serious problems as heavy rotation (Figure 4, frame #166, frame #170, frame #314, frame # 357), scale changes (Figure 4, frame # 99 and frame # 290) and changing background are correctly handled.

similarity of the color distribution between face model q_t and the current iteration result q_m^k

$$D^k = \sum_{i=1}^{N} q_m^k(i) \cdot \log(q_m^i(i) / q_t(i)) \qquad (7)$$

The iteration result will be accepted only when D^k is larger than a threshold.

4. Experimental Results

The system is implemented on a standard PC (Pentium IV at 3.0GHz). The video image size is 320x240 (24 bits per pixel) captured by Sony DCR9E at 25fps. The system is tested in typical indoor and outdoor environments, with large degree head rotations in plane and out of plane, partial occlusion, large scale changes, multiple persons, and nonlinear fast moving in complex background. It works at 10~12 fps. We use color histogram in RGB color space with 10x10x10 bins for building the color distribution of the object

and the surrounding are. We deliberately selected clips taken under difficult conditions, especially those with rotation and occlusion which well known face detection system will be failed. The following presents results.

Figure 3 shows an example of tracking multiple faces in an indoor environment. The cross shows the position of the person's face. The cross in green on a face represents the detection of the face, whereas the cross in red represents the face is under tracking. Note that our system successfully track multiple faces in real-time under various difficult conditions, such as out-of-plane rotations in the range of [-90,90] (in degrees) (first row, frame #79), up-and-down nodding rotations approximately in the range of [-90,90] (in degrees) (first row, frame #349, frame#372), partial occlusion (first row, frame #349, frame #372, frame #394) and large scale changes (third row, frame #249). Many well known detectors [2] may fail at those situations.

Figure 4 gives an example of face tracking with an active camera. This video clip includes 410 frames and only 44 frames contain frontal face that can be detected. Serious problems as heavy rotation (Figure 4, frame #166, frame #170, frame #314, frame #357), scale changes (Figure 4, frame #99 and frame #290) and changing background are correctly handled. It is hard for those systems which only detect face in still frame to achieve the same tracking result.

5. Conclusion

We have presented a reliable real-time system that is able to track multiple faces with largely tilts and rotations in fast motion with high accuracy. The main contributions of the work are the following: First, we presented a novel system architecture, which dynamically switches between face detection and tracking modules, and overcome weaknesses of the two modules.. Second, we described a online learning based faces tracking algorithm. It can improve system's performance in difficult conditions such as out-of-plane rotation, large tilting, partial occlusion, large scale changes, camera motion, multiple persons, and nonlinear fast motion in complex background.
Future work will focus on integrate more cues and features as evidences for face tracking.

Acknowledgements

The work presented in this paper was sponsored by National Natural Science Foundation of China (#60172037 and #60518002), and Foundation of National Laboratory of Pattern Recognition (#1M99G50).

REFERENCES

[1] K. Toyama, " Prolegomena for Robust Face Tracking", *MSR Technical Report, MSR-TR-98-65*, November 1998.
[2] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Dec. 2001.
[3] K. Tieu and P. Viola, "Boosting Image Retrieval", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Volume. 1, pages: 228-235, 2000.
[4] P. Wang and Q. Jin, "Multi-View Face Detection under Complex Scene based on Combined SVMs", *International Conference on Pattern Recognition*, 2004.
[5] J. Feraud, O. Bernier, and M. Collobert, "A Fast and Accurate Face Detector for Indexation of Face Images", *In Proceedings of Fourth IEEE Conference on Automatic Face and Gesture Recognition*, 2000.
[6] L. Wiskott, J. Fellous, N. Kruger, and C.V. Malsburg, "Face Recognition By Elastic Bunch Graph Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligenc*, Volume. 19, no. 7, pages: 775-779, July 1997.
[7] S. Gong, S. McKenna, and J. Collins, "An Investigation into Face Pose Distribution", *In Proceedings of the IEEE Conference on Face and Gesture Recognition*, 1996.
[8] Y.M. Li, S.G. Gong, and H. Liddell, "Support Vector Regression And Classification Based Multi-View Face Detection and Recognition", *In Proceedings of the IEEE Conference on Face and Gesture Recognition*, pages: 300-305, Mar. 2000.
[9] M. Jones and P. Viola , "Fast Multi-view Face Detection", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.
[10] S.Z. Li, L.Zhu, Z.Q. Zhang, A. Blake, H.J. Zhang, H. Shum, "Statistical Learning of Multi-View Face Detection", *In Proceedings of The 7th European Conference on Computer Vision(ECCV)*. Copenhagen, Denmark. May, 2002.
[11] C. Huang, H.Z.Ai, B. Wu, "Boosting Nested Cascade Detector for Multi-View Face Detection", *In Proceedings 17th International Conference on (ICPR'04)*, Volume 2 August 23- 26, 2004.
[12] Y. Wu, T. Yu and G. Hua, "Tracking Appearances with Occlusions", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Vol.I, pp.789-795, Madison, WI, June, 2003.
[13] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking", *International Journal on Computer Vision*, 29(1):5–28, 1998.
[14] G. R. Bradski, "Computer Vision Face Tracking as a Component of a Perceptural User Interface", *Intel Technology Journal*,1998,2, pages: 1-15.
[15] D.Comaniciu, V.Ramesh, and P.Meer, " Real-time tracking of non-rigid objects using mean shift", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2,2000. pages:142-149.
[16] Stan Birchfield, "Elliptical Head Tracking Using Intensity Gradients and Color Histograms", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, pages: 232-237, June 1998.
[17] V. R. Dorin et al, "Kernel-based object tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume:25, May 2003.
[18] Gaël JAFFRÉ and Alain CROUZIL, " Non-Rigid Object Localization from Color Model using Mean Shift", *In Proceedings of the IEEE International Conference on Image Processing*, volume 3, Barcelona, Spain, September 2003, pages: 317-320.