

OCCLUSION HANDLING FOR OBJECT TRACKING IN CROWDED VIDEO SCENES BASED ON THE UNDECIMATED WAVELET FEATURES

M. Khansari¹, H. R. Rabiee^{1,2}, M. Asadi¹, M. Ghanbari^{1,2,3}

¹Digital Media Lab, Department of Computer Engineering, Sharif University of Technology

Emails: Khansari@mehr.sharif.edu, rabiee@sharif.edu, m_asadi@ce.sharif.edu

²Iran Telecommunication Research Center

³Department of Electronic Systems Engineering, University of Essex, England

Email: ghan@essex.ac.uk

ABSTRACT

In this paper, we propose a new algorithm for occlusion handling for object tracking in the crowded video scenes. The algorithm exploits the properties of undecimated wavelet packet transform (UWPT) coefficients and texture analysis to track arbitrary objects. The algorithm is initialized by the user through specifying a region around the object of interest at the reference frame. Then, coefficients of the UWPT of the region construct a Feature Vector (FV) for every pixel in that region. Optimal search for the best match is then performed by using the generated FVs inside an adaptive search window. Adaptation of the search window is achieved by inter-frame texture analysis to find the direction and speed of the object motion. This temporal texture analysis also assists in tracking of the object under partial or short-term full occlusion.

Experimental results show a good performance for occlusion handling for object tracking in crowded scenes, in particular crowds on stairs in airports or train stations.

Keywords: Occlusion Handling, Object Tracking, Crowded Scene, Undecimated Wavelet Packet Transform, Texture Analysis

1. INTRODUCTION

With the emergence of interactive multimedia systems, tracked objects in an object tracking process in video sequences, can be used for many applications such as video surveillance, visual navigation and monitoring, content-based indexing and retrieval, object-based coding, traffic monitoring, sports analysis for enhanced TV broadcasting and video post-production.

Video object tracking techniques vary according to user interaction, tracking features, motion-model assumption, temporal object tracking, and update procedures. The temporal object tracking methods can be classified into four groups: region-based [1], contour/mesh-based [2], model based [3], and feature based methods [4,5]. The target representation and similarity measurement – observation model – are also very important to the performance of any tracking

algorithm, and their design is closely related to the feature selection problem.

One of the challenging problems in object tracking is occlusion handling. The color histogram is typically used to model the targets to combat partial occlusion and non-rigidity [6, 8]. As color histogram only describes the global color distribution and ignores spatial or layout of the colors, tracked objects are easily confused by a background with similar colors. Moreover, it can not deal easily with illumination changes and lacks exploiting any spatial information. Therefore, feature description based on color histogram for target tracking particularly in the crowded scenes, where similar small objects exist will most likely fail.

More recently, mean-shift tracking algorithms that use color histogram has been successfully applied in object tracking and proved to be robust to partial occlusions [6, 9, 10, 11]. However, these techniques need more sophisticated motion filtering for handling occlusions in the crowded scenes; to the best of our knowledge such a motion filter for tracking and occlusion handling in the crowded scenes has not been reported yet.

Color histogram has also been integrated in probabilistic frameworks such as Bayesian and particle filters [8, 13, 4, 15]. Comparative evaluation of different tracking algorithms shows that this family of techniques is more robust to partial or temporary occlusions over a few frames than the other well known techniques [16].

The general drawback of these techniques is that similar objects (e.g. heads in a crowd) hardly change the color histogram and hence impair their reliability especially in case of occlusion.

The main contribution of this paper is evaluation and analysis of the adaptation of a feature vector generation and block matching algorithm in the Undecimated Wavelet Packet Transform (UWPT) domain [19] for tracking objects [20, 21] in crowded scenes [22] in presence of occlusion. In contrast to the conventional methods for solving the occlusion handling problem using spatial domain features with the limited functionality, it introduces a new transform domain tracking algorithm that can manage partial or short term full occlusion.

After presenting the algorithm in section 2, performance of the proposed algorithm under various

occlusion conditions is evaluated in section 3. Finally, section 4 provides the concluding remarks and the future works.

2. THE PROPOSED ALGORITHM

In our algorithm, object tracking is performed by temporal tracking of a user-defined rectangle around the object at a reference frame. A general block diagram of the algorithm is shown in Fig 1.

Initially, the user specifies a rectangle around the object's boundary at the reference frame. Then, a Feature Vector (FV) for each pixel in the rectangle is constructed by using the coefficients in the Undecimated Wavelet Packet Transform (UWPT) domain. The final step before finding the object in a new frame is the temporal tracking of the pixels in the rectangle at the reference frame. The temporal tracking algorithm uses the generated FV to find the new location of the pixels in an adaptive search window. The search window is updated at each frame based on the interframe texture analysis (Fig. 1).

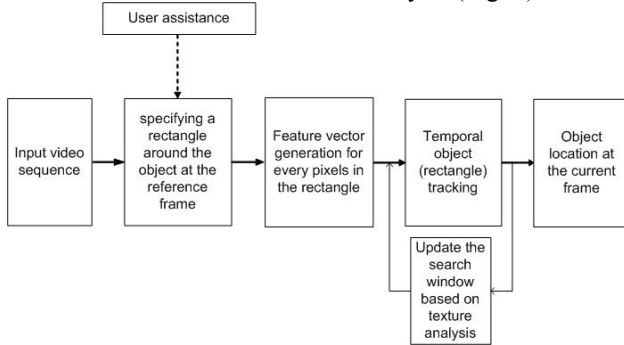


Figure 1. A block diagram of the proposed algorithm

2.1. Feature Vector Generation

In this step, the wavelet packet tree for the desired rectangular object in the reference frame is generated by the UWPT.

The key advantage of UWPT is that it is redundant (subbands have equal size to the original signal), shift invariant, and it gives a denser approximation to continuous wavelet transform than the approximation provided by the orthonormal discrete wavelet transform [17, 18, 28, 29]. Therefore, it can be used for generating invariant and robust features corresponding to each pixel in image processing applications [19-24].

Moreover, UWPT alleviates the problem of subband aliasing associated with the decimated transforms such as DWT.

The procedure for generating a FV for each pixel in the region r (which contains the target object) at frame t can be summarized in the following steps:

1. Generate UWPT for region r (Note that if needed, UWPT is constructed with zero padding when needed).

2. Perform basis selection from the generated subbands. As the approximation subband provides an average of the signal based on the number of levels at the UWPT tree, we prune the tree to have the most coefficients from the approximation subbands. For example, in Fig.2, we may let $x = (w_4^A, w_4^D)$. For our application, this type of basis selection is reasonable; because the comparison in the temporal tracking part of the algorithm is carried out between two regions that are represented by similar approximation subbands.

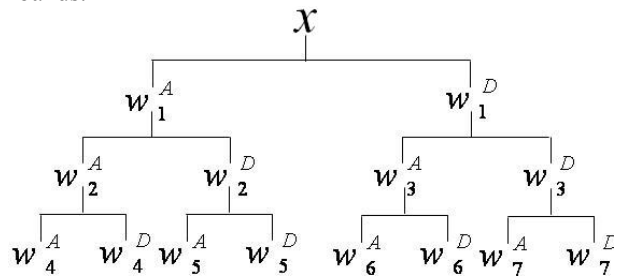


Figure 2. Undecimated Wavelet Packet Transform tree for one dimensional signal x

The output of this step is an array of node number indices of the UWPT tree that specify the selected basis for the successive frame manipulations.

The FV for each pixel in the region r , can be simply created by selecting the corresponding wavelet coefficients in the selected basis nodes of step 2. Therefore, the number of elements in the FV is the same as the number of selected basis nodes.

2.2. TEMPORAL TRACKING

The aim of temporal tracking is to locate the object of interest in the successive frames based on the information about the object at the reference and current frames. The generated FVs for the region around the object can be used to find the best matched region in successive frames; that is pixels within region r are used to find the correct location of the object in frame $t+1$. The process of matching region r in frame t to region $r+1$ in frame $t+1$ is performed through the full search of the region r in a search window in frame $t+1$; similar to the general block-matching algorithm, except it exploits the generated FV of a pixel rather than its luminance. The search window is adaptively determined by the texture analysis approach [25].

2.3. SEARCH WINDOW UPDATING MECHANISM

The change of object location requires an efficient and adaptive search window updating mechanism for three reasons:

- The proper search window location ensures that the object always lies within the search area and thus prevents the loss of the object inside the search window.
- A location-adaptive fixed size search window decreases computational complexity that is resulted by a large and variable size search window [20].
- If a moving target is occluded by another object, use of direction of motion may alleviate the occlusion problem.

Our approach to attain an efficient search window updating mechanism is to estimate the direction and the speed of motion of the object using inter-frame texture analysis technique to update the location of the search window [22, 25].

To find the direction and the speed of the object motion, we define the temporal difference histogram of two successive frames. Coarseness and directionality of the frame difference of the two successive frames can be derived from the temporal difference histogram. Finally, the direction and speed of the motion is estimated through the use of temporal difference histogram, coarseness and directionality [22, 25].

2.3.1 Temporal difference histogram

The temporal difference histogram of two successive frames is derived from absolute difference of gray level values of corresponding pixels at the two frames.

Consider the current search window ($SA_t(x, y)$) at frame t and a new search window ($SA_{t+1}(x, y)$) determined by a displacement value of $\delta = (\Delta x, \Delta y)$ to the current search window center in the next frame. It should be noted that the two search windows have the same size. We define absolute temporal difference (ATD_δ) of the two windows as follows:

$$ATD_\delta(x, y) = \sum_{y=0}^{N_y-1} \sum_{x=0}^{N_x-1} |SA_t(x, y) - SA_{t+1}(x + \Delta x, y + \Delta y)| \quad (2.3.1.1)$$

Where N_x and N_y are the width and height of the search window, respectively. Then, we calculate the histogram of the values of ATD_δ . Note that the histogram has M bins, where M is the number of gray level values in each frame (256 for an 8 bit image).

Finally, the histogram values are normalized with respect to the number of pixels in the search window

($N_x * N_y$) to obtain the probability density of each gray level value, $p_\delta(i)$ $i = 0, \dots, M - 1$.

2.3.2 Search Window direction

Assume that the search window is a rectangular block. Consider eight different blocks at the various directions with distance of δ_i from the center of search window at the current frame (Fig. 3).

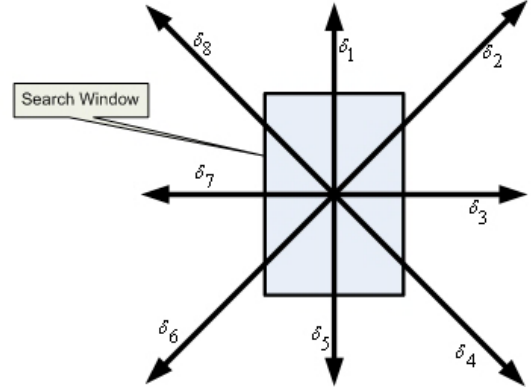


Figure 3. Fig.3. Distance assignment in the different directions to find the maximum *IDM* (Inverse Difference Moment)

Then, calculate the temporal difference histogram, p_{δ_i} , for each block with respect to the original block (search window). Now, we can easily compute the inverse difference moment, IDM_i , corresponding to each block using equation (2.3.2.1). The inverse difference moment, IDM , is the measure of homogeneity and is defined as:

$$IDM = \sum_{i=0}^{M-1} \frac{p_{\delta}(i)}{i^2 + 1} \quad (2.3.2.1)$$

In a homogeneous image, there are very few dominant gray level transitions, hence p_{δ_i} has a few entries of large magnitudes. Here IDM contains information on the distribution of the non-zero values of p_{δ_i} and can be used to identify the principle texture direction. If a texture is directional, that is coarser in one direction than the others, then the degree of the spread of the values in p_{δ_i} should vary with the direction of δ_i , assuming that its magnitude is in the proper range. Thus, texture directionality can be analyzed by comparing spread measures of p_{δ_i} for various directions of δ .

To derive the motion direction from texture direction, the direction that maximizes the IDM should be found.

$$IDM_{\max} = \max \{IDM_i\}, \quad i = 1, 2, \dots, 8 \quad (2.3.2.2)$$

The maximum value of IDM , IDM_{\max} indicates that the frame difference is more homogenous in that direction than the others, implying that the corresponding blocks in the successive frames are more correlated.

2.3.3 Search Window displacement

The quantitative measure for coarseness of texture is the temporal contrast which is defined as the moment of inertia of p_{δ} around the origin, and is given by:

$$TCON = \sum_{i=0}^{M-1} i^2 p_{\delta}(i) \quad (2.3.3.1)$$

where M is the number of gray level values in each frame as stated in section 2.3.1.

The parameter $TCON$, gives a quantitative measure for the coarseness of the texture and its value depends on the amount of local variations that are present in the region of interest. The existence of high local variations in a frame implies an object activity in the frame and this frame is called active when compared to the frames with small variations. Since active frames of an image sequence exhibit a large amount of local variations, the temporal contrast derived from the frame difference signal is related to the picture activity. The parameter $TCON$ is normalized to Local Contrast ($LCON$) in order to minimize the effect of size and texture of the search window (SW). The parameter $LCON$ which defines the pixel variance within the search window is given by:

$$LCON = \frac{1}{SW} \sum_{SW} [g(x, y) - \bar{g}]^2 \quad (2.3.3.2)$$

where $g(x, y)$ is the gray level value of the pixel located at position (x, y) and \bar{g} is the average gray value of the pixels in the search window. Based on the temporal and local contrasts, a good estimation of the average motion speed, S , within a block can be defined as:

$$S = k \frac{TCON}{LCON} \quad (2.3.2.3)$$

where k is a constant with empirically selected values. The average motion speed, S , in equation (2.3.3.3), is not only independent of the size of the moving objects but also is invariant to the orientation of their texture. The value of S approaches to zero for stationary parts of the picture such as background, independent of its texture contents [25].

The displacement value of the search window for the next frame is given by

$$\begin{aligned} R_{j-1} &= S_{j-1} - Disp_{j-1} \\ Disp_j &= \lfloor S_j + R_{j-1} \rfloor \end{aligned} \quad (2.3.3.4)$$

In some future frames the value of S might be less than 1, thus the displacement of the search window will be equal to zero. Parameter R_{j-1} denotes the displacement residue at the previous frame. Assuming low speed object movements, the parameter R_{j-1} , helps to sum up the values of displacements that are less than one pixel away, until they reach to at least one pixel displacement.

4. EXPERIMENTAL RESULTS

Throughout our experiments, we have assumed that there are no scene cuts. Clearly, In case of a scene cut, the reference frame and the target object should be updated and a new user intervention is required. Although several objective evaluation measures have been presented in the literature [31, 32], since there is no universally agreed method of evaluating the performance of object tracking for crowded scenes, we have generally analyzed our results subjectively and have introduced a method for objective evaluation, when possible.

Results of the proposed tracking method have been compared with the well-known color histogram based tracking algorithms with two different matching distance measures; i.e. Chi-square and Bhattacharyya. Therefore, most of figures are presented in three parts: a) Color Histogram based tracking with Chi-square matching distance measure (briefly CHC), b) Color Histogram based tracking with Bhattacharyya matching distance measure (briefly CHB), and c) our proposed algorithm (briefly WBMA¹).

In addition, for each tracking result, the corresponding complete set of video clips are generated that are available through Internet² for more detailed subjective comparison (21 video clips). Moreover, we have defined a method for objective evaluation of the tracking techniques; the Euclidian distance of the center of the gravity of the tracked and actual object. Here at the start of tracking, a bounding rectangle centered at the center of the gravity of the desired object is drawn. In the following frames, the bounding rectangle represents the tracked object and its distance with the center of the gravity of the actual object is measured.

We have used biorthogonal wavelet bases, which are particularly useful for object detection to generate the UWPT tree. In fact, the presence of spikes in the biorthogonal wavelet bases makes them suitable for target tracking applications [30]. In all experiments of this section, we have used 3 levels of UWPT tree

¹ Wavelet-based Block Matching Algorithm

² <http://mehr.sharif.edu/~khansari>

decomposition with wavelet family of *Bior*2.2 [28]. In *Bior* $N_r.N_d$, N_r and N_d represent the number of vanishing moments of the synthesis and analysis filters [28]. In the color histogram based algorithm implementation, the number of color bins has been set to 32.

To evaluate the algorithms in a real environment setting, we have applied them to different real video clips of Tehran Metro Stations in cooperation with the Tehran Metro authorities. These videos show the crowds at different parts of the metro such as getting on/off the train and up/down the stairs. Moreover, they include different conditions in crowded scenes such as partial and complete occlusions, high and low speed, variable occlusion duration, zooming in and out. In all the shown snap shots, solid rectangles correspond to the rectangles around the objects and the rectangles with dashed lines represent the search window. Note the difficulty in tracking of heads in a crowded scene, as there are several near distance similar objects.

Fig. 4 shows the result of tracking of a person moving up the stairs and toward the camera in the metro station. Frame #635 was the reference frame, the size of the rectangle around the object was 25x15 and the search window size was 75x65 (± 25 blocks). Empirical parameters to find the direction and speed of the motion for updating the search window was set to $d=2$ and $k=8$. The object is moving up with a constant speed and in a number of frames it was fully occluded by the head of another person. The occlusion has started in frame #656 with partial occlusion and

turned into full occlusion in frames #660 to #669. It is clear from Fig. 4 that our algorithm could successfully handle partial and full occlusion in this experiment. It has been observed that if during occlusion the object is still in the search area, immediately after appearing, the algorithm can successfully detect it.

There are two reasons for occlusion handling in this Fig. and next figures:

1- The proposed FV is robust enough to handle the partial occlusion (as shown in the figures of this section) compared to spatial space feature vectors algorithms such as color histogram based algorithms.

2- As the algorithm uses activity analysis to find the motion and direction of the search window and updates the search window location, and the object and obstacle move at the same direction, it can be predicted by the algorithm [22,25]. This type of setting for the search window location ensures that the object lies within the search window in case of occlusion. Therefore, in contrast to well-know methods loss of track is mostly prevented.

Long-duration occlusion originates from the fact that the object of interest and the occluding object both moves at the same direction. Based on item 2, the search window follows the object even in case of occlusion. The reason is that –according to item 2- the new location of the search window is updated according to the speed and direction of the motion. Therefore, it follows the occluding object. Once the occluded object appears it will be inside the search window.

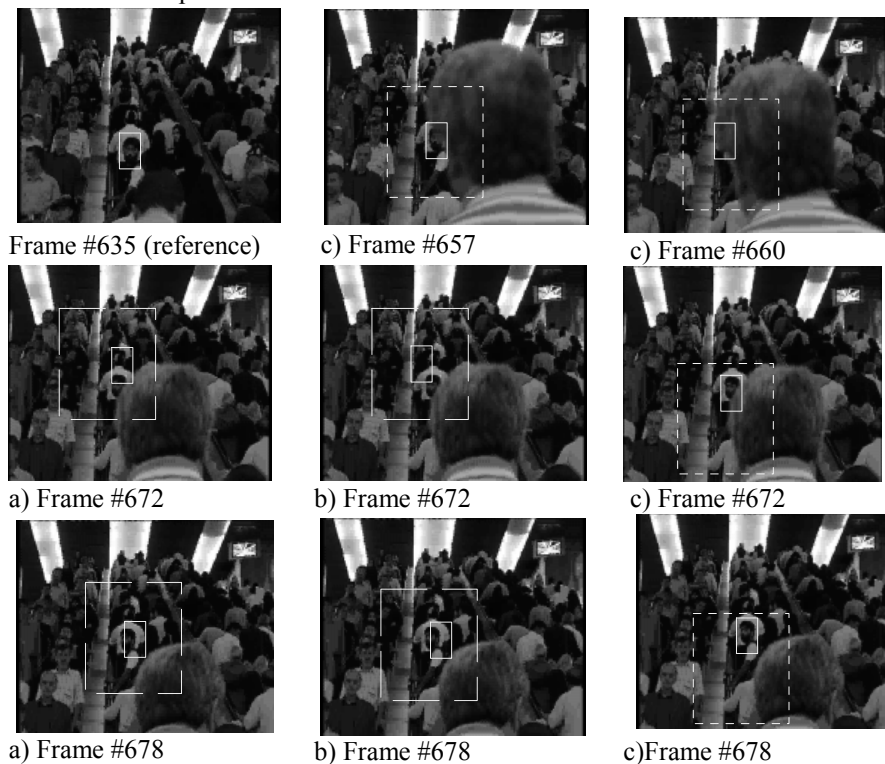


Figure 4. Tracking a man moving up the stairs, with full occlusion in some frames; a) CHC, b) CHB, c) WBMA.

Moreover, it avoids error propagation of false search window center prediction in the following frames in case of incorrect object tracking of the current frame.

In case of short-duration occlusion, motion directions of the object and the occluding object are different. Therefore, for a suitable search window size the created FV can handle the occlusion as soon as the object appears partially.

Fig. 5 shows the result of tracking where the crowd are getting off the train. Frame #35 is the reference frame with the size of bounding rectangle 42x22 and the search window of size 84x64. Empirical parameters to find the direction and speed of the motion for updating the search window was set to $d=1$ and $k=3$. The object is passing through the crowd and experiences partial occlusions and some zooming effects are also present in a number of frames. The partial occlusion begins from the frame #62 and then in frame #64 turns almost into complete occlusion that were lasted for 10 successive frames. As demonstrated in the Fig. 5, the algorithm could successfully handle partial occlusions even in the presence of zooming effects, due to the robustness of FVs and adaptability of the search window. Although in a few frames the algorithm can not find the object exactly (e.g. frames #85 to #89) due to the high movement of the object, lack of feature vector updating mechanism and object blurring compared to the reference frame, it acts much better than the color histogram based algorithms.



c) Frame #77 c) Frame #88
Figure 5. Tracking a man getting off the train in the various kinds of partial and full occlusions, long-duration occlusion and zooming in effects; a) CHC, b) CHB, c) WBMA

Fig. 6 shows the result of tracking of a man where he moves inside the crowd in presence of repeated partial and full occlusion and zooming effects. Frame #162 of the sequence is considered as the reference frame, the size of rectangle around the object and search window were 37x22 and 111x96 (± 37 pixels), respectively. Updating parameters of the search window were chosen to be $d=1$ and $k=3$. As the object (the man with the bright shirt) moves in various directions, he is partially and fully occluded by others at several successive frames. The object was occluded completely in a number of frames; for example frames #176 to #178 as shown in Fig. 6. It is important to note that when both complete occlusion and zooming effects are present, it takes a number of frames before the object could be tracked successfully (the last row of Fig. 6).



Figure 6. A man is tracked in the crowd in the metro in presence of partial and full occlusions and zooming effects (all figures are the results of the proposed algorithm)

5. CONCLUSIONS AND FUTURE WORKS

A new adaptive object tracking algorithm for crowded scenes based on pixel features in the wavelet domain and a novel adaptive search window updating mechanism based on texture analysis, has been proposed for object tracking in crowded scenes. Based on the properties of the UWPT, existence of individual robust FVs for each pixel, and the adaptive search window, this method can tolerate partial or complete occlusions in a reasonable number of successive frames. The experimental results confirmed the efficiency of our algorithm in tracking the object in crowded scenes in case of occlusion.

One of the open problems in this algorithm is the period and procedure for updating the feature vectors during the track. In the current algorithm, the FVs are kept unchanged during a track. A memory-based FV updating mechanism combined with Kalman filters for search area prediction can improve the performance of our algorithm in presence of abrupt zooming in/out or object scaling. In addition, the search window updating mechanism must be improved, because when the tracked object falls outside the search window, especially in presence of occlusion, noise, abrupt transformation and zooming, the algorithm fails to track the object successfully.

Finally, we can use color components and combination of spatial domain features such as edge and texture to further improve the performance of our algorithm in color video clips. In this case, additional information can weight the FV and improve the searching mechanism.

ACKNOWLEDGMENT

This research has been funded by advanced Information and Communication Technology Center (AICTC) of Sharif University of Technology and Iran Telecommunications Research Center (ITRC).

REFERENCES

- [1]. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Computer Vision*, vol. 1, no. 4, pp. 321-331, 1987.
- [2]. Ç. E. Erdem, A. M. Tekalp, B. Sankur, "Video Object Tracking With Feedback of Performance Measures," *IEEE Transaction for Circuit and Systems for Video Technology*, Vol. 13, No. 4, pp. 310-324, 2003.
- [3]. Y. Chen, Y. Rui, and T. S. Huang, "JPDAF based HMM for real-time contour tracking," in Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, pp. 543-550, 2001.
- [4]. C. Gu, M. Lee, "Semiautomatic Segmentation and Tracking of Semantic Video Objects," *IEEE Transaction for Circuit and Systems for Video Technology*, vol. 8, no. 5, pp. 572-584, 1998.
- [5]. D. Wang, "Unsupervised video segmentation based watersheds and temporal tracking," *IEEE Transaction for Circuit and Systems for video Technology*, vol. 8., no. 5, pp. 539-546, 1998.
- [6]. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564-577, May 2003.
- [7]. Liu, T.L., Chen, H.T, "Real-Time Tracking Using Trust-Region Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 397-402, 2004.
- [8]. P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," In Proc. European Conf. on Computer Vision, Copenhagen, Denmark, vol. I, pp. 661-675, 2002.
- [9]. D. Comaniciu and P. Meer, "Mean shift: A Robust Approach Toward Feature Space Analysis", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603-619, May 2002.
- [10]. Z. Zivkovic, B. Krose, "An EM-like algorithm for color-histogrambased object tracking," Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2004.
- [11]. Dong Xu, Yimin Wang, and Jinwen An, "Applying a New Spatial Color Histogram in Mean-Shift Based Tracking Algorithm," Image and Vision Computing New Zealand, 2005.
- [12]. D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," *Proc. IEEE Conf. on CVPR*, pp. 142-149, 2000.
- [13]. A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking," Proceedings of the International Conference on Computer Vision and Pattern Recognition, vol. I, pp. 415-422, 2001.
- [14]. A. Jacquot, P. Sturm O. Ruch," Adaptive Tracking of Non-Rigid Objects Based on Color Histograms and Automatic Parameter Selection ," IEEE Workshop on Motion and Video Computing (WACV/MOTION'05), vol. 2, pp. 103-109, 2005.
- [15]. B. Han, C. Yang, R. Duraiswami and L. Davis, "Bayesian Filtering and Integral Image for Visual Tracking. Invited to special session of Real-Time Object Tracking," Algorithms and Evaluation in Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Montreux, Switzerland, 2005.
- [16]. B. Deutsch, Ch. Gr"abl, F. Bajramovic, J. Denzler, "A Comparative Evaluation of Template and

Histogram Based 2D Tracking Algorithms,” Pattern Recognition, 27th DAGM Symposium, Springer, pp. 269–276, 2005.

[17] R. R. Coifman and M. V. Wickerhauser, “Entropy-based algorithms for best basis selection,” IEEE Transaction on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis, Vol. 38, pp. 713-718, Mar. 1992.

[18] K. Ramchandran and M. Vetterli, “Best wavelet packet bases in a rate distortion sense,” IEEE Transaction on Image Processing, Vol. 2, pp. 160-175, Apr. 1993.

[19] M. Amiri, H. R. Rabiee, F. Behazin, M. Khansari, “A new wavelet domain block matching algorithm for real-time object tracking,” IEEE ICIP, September 14-17, Barcelona, Spain, pp. 961-4, 2003.

[20] M. Khansari, H. R. Rabiee, M. Asadi, M. Ghanbari, M. Nosrati, M. Amiri, “A Semi-Automatic Video Object Extraction Algorithm Based on Joint Transform And Spatial Domain Features,” CBMI, International Workshop on Content-Based Multimedia Indexing, Riga, Latvia, June 21-23, 2005.

[21] M. Khansari, H. R. Rabiee, M. Asadi, M. Ghanbari, M. Nosrati, M. Amiri, “A Shape Tracking Algorithm Based on Generated Pixel Features by Undecimated Wavelet Packet,” CSICC, 10th Annual Computer Society of Iran Computer Conference, Tehran, Iran, 2005.

[22] M. Khansari, H. R. Rabiee, M. Asadi, P. Khadem Hamedani, M. Ghanbari, “Adaptive Search Window for Object Tracking in the Crowds using Undecimated Wavelet Packet Features,” WAC, World Automation Congress, Budapest, Hungary, July 24-26, 2006.

[23] M. Khansari, H. R. Rabiee, M. Asadi, M. Ghanbari, M. Nosrati, M. Amiri, “A Quantization Noise Robust Object's Shape Prediction Algorithm,” EUSIPCO, European Signal Processing Conference, Antalya, Turkey, September 4-8, 2005.

[24] M. Khansari, H. R. Rabiee, M. Asadi, M. Nosrati, M. Amiri, M. Ghanbari “Object Shape Prediction in Noisy Video Based on Undecimated Wavelet Packet Features,” 12th International Multimedia Modelling (MMM) Conference, Beijing, China, Jan. 2-4, 2006.

[25] V. E. Seferidis and M. Ghanbari, “Adaptive Motion Estimation Based on Texture Analysis,” IEEE Transactions on Communications, vol. 42, no. 2-4, pp. 1277-1287, 1994.

[26] M. Vetterli, J. Kovacevic, Wavelets and Subband Coding, Prentice Hall PTR, 1st, edition, 1995.

[27] I. Cohen, S. Raz, D. Malah, “Orthonormal shift-invariant wavelet packet decomposition and

representation,” Elsevier, Signal Processing, 57, pp.251-270, 1997

[28] Daubechies, Ten lectures on wavelets, CBMS, SIAM, 61, pp. 271-280, 1994.

[29] H. Guo, “Theory and applications of the shift invariant, time-varying, and undecimated wavelet transform,” Master's Thesis, Rice University, Houston, TX, 1995.

[30]. R. N. Strickland and H. I. Hahn, “Wavelet transform methods for object detection and recovery,” IEEE Transaction on Image Processing, vol. 6, no. 5, pp. 724-735, May 1997.

[31] Ç. Erdem et. al., “Performance measures for Video Object Segmentation and Tracking,” IEEE Transaction on Image Processing, Vol. 13, No. 7, July 2004.

[32] Paulo Correia and Fernando Pereira, “\,” EURASIP Journal on Applied Signal Processing, vol. 2006, Article ID 82195, 11 pages, 2006.