

AN ADAPTIVE SEMI-AUTOMATIC VIDEO OBJECT EXTRACTION ALGORITHM BASED ON JOINT TRANSFORM AND SPATIAL DOMAINS FEATURES

M. Khansari¹, H. R. Rabiee¹, M. Asadi¹, M. Nosrati¹, M. Amiri¹, M. Ghanbari^{1,2}

¹Digital Media Lab, Computer Engineering Department, Sharif University of Technology
Azadi Ave., Tehran, Iran

Emails: Khansari@mehr.sharif.edu,

rabiee@sharif.edu, asadi@dml.aictc.com, mohadese_nosrati@yahoo.com, amiri@mehr.sharif.edu,

Web: <http://www.aictc.com/dml>

Department of Electronic Systems Engineering, University of Essex, England

Email: ghan@essex.ac.uk

Web: <http://privatewww.essex.ac.uk/~ghan/>

ABSTRACT

We propose a new adaptive algorithm for semi-automatic video object segmentation based on joint pixel features using the undecimated wavelet packet transform (UWPT) and luminance value. The method starts with the object's boundary specification at the reference frame assisted by the user. After selecting a set of feature points which approximate the object's boundary, the amplitude of coefficients in the best basis tree expansion of the UWPT is used to create a Feature Vector (FV) corresponding to each pixel. Weighting the FV with the magnitude of the pixel's luminance results in a pixel-wise feature that can be tracked temporally in the video sequence. Full search for the best match has been performed through the use of generated FVs and an adaptive search window updating.

Experimental results show a good performance in case of object translation, small rotation and scaling. The method can be used to track both rigid and non-rigid shapes in video and image sequences.

1. INTRODUCTION

During the last two decades object segmentation in image and video processing has found several applications and has been a universal challenge to researchers both in academia and industry. Object segmentation in the context of video applications is the process of partitioning the video frames into semantically meaningful objects and background. Segmentation of video is performed in both intra-frame (spatial) and inter-frame (temporal) fashions.

Segmented video objects can be employed in many applications including automatic target tracking, visual navigation and monitoring, surveillance, content-based indexing and retrieval, object-based coding in MPEG-4, sport scene analysis, and video post-production.

In the context of the content-based functionalities of the emerging video standards such as MPEG-4 and MPEG-7, the importance of video object extraction would become more apparent. For example, MPEG-4 uses a Video Object Layer (VOL) to support content-based functionalities. This layer is composed of Video Object Planes (VOPs). Each VOP is a semantically meaningful partition of a video frame resulted from spatial and temporal segmentation. VOP allows content-aware compression, manipulation and indexing of MPEG-4 video and flexible and interactive video representation at the decoder. The success of MPEG-4 depends on how good VOPs are fed into an MPEG-4 encoder. Therefore, generating accurate VOPs for a video sequence is of great importance.

In general, automatic recognition of semantic objects is hard to realize, especially when there is no priori information available. Designing a fully automatic VOP generation algorithm still remains a difficult problem for typical video sequences. A simple homogeneity criterion (color, motion, etc) does not lead to a complete semantic object because some parts of a semantic object maybe static [1].

Recent research results have shown that semi-automatic object tracking can solve the problem to some extent. Semi-automatic VOP generation algorithms make use of the user's knowledge in selecting the objects of interest and initializing the algorithm [1]. These methods employ user's assistance in order to specify the object's shape at some reference frame.

In this paper, we have developed a new algorithm for extracting user-defined objects in a video sequence to be used in semi-automatic VOP generation systems. The algorithm makes use of generated features for pixels within the small blocks approximating the object's shape. The features are generated through the use of Undecimated Wavelet Packet Transform (UWPT) and pixel's luminance.

After surveying the literature of video object segmentation in section 2, we present our new algorithm in section 3. Section 4 illustrates the experimental results and in section 5 conclusions and the future works are presented.

2. RELATED WORKS

Various techniques for video object segmentation have been proposed in the literature [2,3]. Based on the user interaction, these techniques can be classified into automatic (unsupervised) and semi-automatic (supervised) categories. Automatic video segmentation requires no user intervention in guiding or improving the segmentation. Generally, this class of techniques segments the video into moving objects. As it is hard to find specific, invariant, and good homogeneity criteria to describe the semantics, automatic methods do not usually result in complete, semantically meaningful objects. On the contrary, semi-automatic or supervised methods use the intelligence and recognition capabilities of the human recognition system to initialize and perform the segmentation.

Semi-automatic video segmentation algorithms differ with respect to user interaction, tracking features, motion-model assumption, temporal object tracking, and update procedure.

The temporal object tracking methods can be classified into four groups: region-based, contour/mesh-based, model based, and feature based methods [3].

In the region-based methods information such as motion, color, and texture of the regions are used to track the regions. By using a combination of these regions, one can construct the intended object's shape. The proposed method by Wang [4] has four steps. A linear motion model is used to project the moving objects into the next frame. In the marker extraction step, reliable object markers are obtained using morphological erosion and pixel difference thresholding. A modified watershed transform, followed by region merging, yields a complete segmentation of the next frame.

Contour-based methods try to track an object by following the pixels on the object boundary by building the contour of the object. These methods make use of the motion information to project the contour and then adapt it to the detected object in the next frame. Deformable object motion models lead to active contour models (snakes) [5] or meshes [6]. These methods allow tracking of both rigid and non-rigid objects and present solutions to solve the problem of tracking for partially occluded objects. Erdem et al. [7] presented a contour tracking framework for both rigid and non-rigid objects in the presence of occlusion. At first, they divide the object's contour into sub contours and estimate the mapping parameters for each subcontour. They, then apply a boundary correction based on active contour model with weighted low level features of color and motion on each subcontour. A feedback measure is also used to adjust the weights.

In model-based methods the parameterized object model makes the priori information. Chen et al. [8] utilize hidden Markov model (HMM) to model an object's boundary. The method takes advantage of a joint probability data associated filter to compute HMM parameters.

Haenselmann et al. [9] have proposed a wavelet-based algorithm to extract an object from an image in a semi-automatic manner. The method selects a piece of object's boundary that separates the object from the background and traces it based on a multi-scale wavelet analysis of the sample area. The sample boundary does not have to be a sharp edge.

Marcotegui et al. [10] describe an interactive video segmentation tool called VOGUE. The system combines three different sets of automatic and semi-automatic tools (spatial segmentation, object tracking, and temporal segmentation) to obtain a complete user-assisted segmentation. For spatial segmentation, a morphological multi-scale segmentation has been proposed. The system uses different types of user interactions such as refining and coarsening a region, marker drawing, and contour adjustment. The result of temporal tracking is displayed to the user for refinement of the object mask.

Gu and Lee [11] have proposed a combination of mathematical morphology and perspective motion model to extract objects in an image sequence. First, a human assisted system is used to specify the object at a key frame using morphological segmentation tools. Then, shape tracking is achieved through the use of global perspective motion estimation and compensation plus a boundary refinement algorithm. They assume that the shape of the object does not change dramatically from frame to frame so the previously declared object boundary is predicted using motion compensation and correct boundary is determined by morphological operations. Since the framework of tracking uses a global rigid motion model along with the local non-rigid refinement, it has difficulty in dealing with a large non-rigid object movement.

Kim and Hwang [12] have used a robust double-edge map derived from the difference of two successive frames to generate a moving edge. After removing previous frame edge points, the resulted moving edge is used to extract the object shape which in the context of MPEG-4 is called video object plane (VOP).

Tsaig and Averbuch [13] formulate the problem as graph labeling over a region adjacency graph (RAG), based on motion information. The label field is modeled as a Markovian Random Field (MRF). An initial spatial partition of each frame is obtained by a watershed algorithm. The motion of each region is estimated by hierarchical region matching. A dynamic memory, based on object tracking, is incorporated into the segmentation process to maintain temporal coherence of the segmentation. Finally, the labeling is obtained by maximizing a posteriori probability of the MRF using motion information, spatial information and the memory.

3. THE PROPOSED ALGORITHM

In our proposed algorithm, object extraction is performed by temporal tracking of some selected feature points located at the neighborhood of the object's boundary in a reference frame. The feature points should give a good approximation of the object's boundary. The algorithm is semi-automatic in the sense that the user assistance to specify the object's boundary in the reference frame is necessary. A general block diagram of the algorithm is shown in Fig. 1.

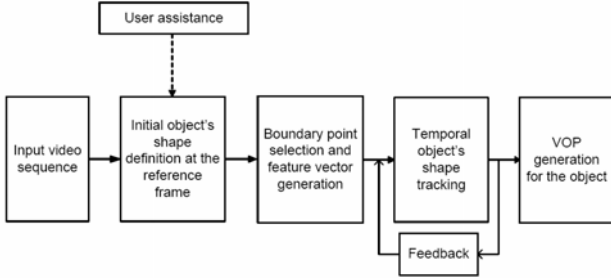


Fig. 1. A general block diagram of the proposed algorithm

After the object's boundary has been completely specified in the reference frame by the aid of the user, an algorithm selects a set of feature points which approximate the object's border at the reference frame. The algorithm makes use of the specified boundary to choose the feature points.

Then, a joint feature vector is constructed by combining the generated feature in the Undecimated Wavelet Packet Transform (UWPT) domain and luminance value of the pixel. The final step before extracting the object in a new frame is temporal tracking of the selected feature point at the reference frame. Temporal tracking uses the generated feature vector to find the new location of the feature points and reconstruction of the new object's boundary in the new frame.

A selected feature point is "near" the object's boundary if the distance between this point and the boundary is less than a predefined threshold T . To achieve the best prediction of the object's shape, the number (M) and the position of these points are computed based on the threshold (T) and the size of frame. Let S_r be the set of M feature points which approximate the object's shape at the reference frame r .

$$S_r = [s_r^1, s_r^2, \dots, s_r^M], \quad s_r^i = [x_r^i, y_r^i]$$

where $[x_r^i, y_r^i]$ is the coordinates of point s_r^i at the reference frame r .

We would like to find the object's boundary in one of the next frames, n by finding the corresponding pixel in frame n to each point in S_r . If S_n is known, the

object's shape at frame n can be fully reconstructed. Assume that the corresponding pixel to s_r^i in frame n is represented by s_n^i . s_r^i may undergo a complex transformation during passing through different frames between frames r and n . Generally, it is hard to find s_n^i using variable and noise sensitive spatial domain features such as luminance, texture, etc.

Our approach to track s_r^i in frame n makes use of shift-invariant Feature Vector (FV) for s_r^i . This FV is derived from the Undecimated Wavelet Transform (UWT) domain by exploiting the invariance property of the UWT. Moreover, UWPT removes the problem of subband aliasing associated with the decimated transform such as (DWT, FT, etc).

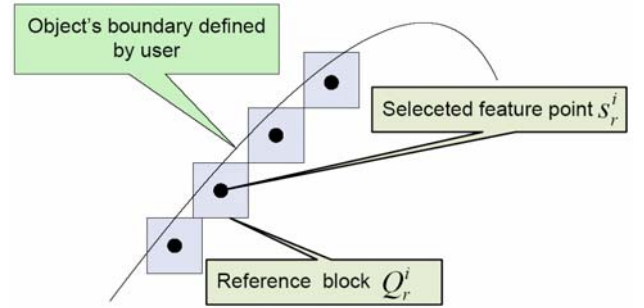


Fig.2. Sample set of feature points and associated squares in the reference frame

For the sake of robustness of pixel tracking, for each s_r^i we define a square Q_r^i centered at s_r^i (Fig. 2). The pixels within the Q_r^i are used to find the correct location of s_r^i in frame n (s_n^i).

The algorithm tracks square Q_r^i in the next frame and finds the best matched Q_n^i and hence s_n^i . Having found S_n , the new object's shape in frame n can simply be reconstructed by interpolating between the points.

The problem that needs to be specified is that how the algorithm generates the joint feature vector (JFV) for each pixel in Q_r^i using the wavelet packet tree and pixel's luminance and tracks the generated JFV in frame n .

The wavelet packet tree is generated by the Undecimated Wavelet Packet Transform (UWPT). UWPT has two properties, which make it suitable for generating invariant and noise robust features corresponding to each pixel. These are:

1. It has the shift invariant property. Due to this property, when pixels of Q_r^i move from frame r to new positions in frame n (translation), there would be little changes in the value of the wavelet coefficients. Therefore, feature vectors, which are based on wavelet coefficients in frame r , can be found again in frame n .

- The subbands in the decomposition tree are all of the same size, which are also equal to the size of the input frame. This feature simplifies the feature extraction procedure.

In general, biorthogonal wavelet bases, which are particularly useful for object detection, could be used to generate the UWPT tree. The presence of spikes in biorthogonal wavelet bases makes it suitable for object detection applications [14, 17].

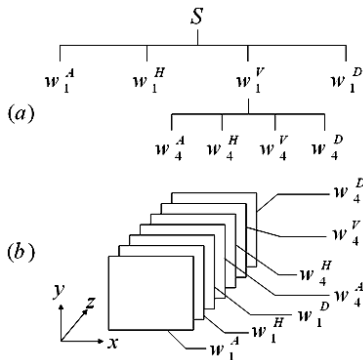
The procedure for generating a feature vector for each pixel in frame r can be summarized as follows:

Stage 1:

1. Generate UWPT for frame r .
2. Perform entropy-based algorithms for the best basis selection [1] and prune the wavelet packet tree. The output of this step is an array of node indices of the UWPT tree, which specify the best basis.
3. Having considered the second property of the UWPT, feature vector (FV) for each pixel in frame r (therefore in Q_r^i) can be simply created by selecting the corresponding wavelet coefficients in the best basis nodes of step 2. Therefore, the number of elements in FV is the same as the number of best basis nodes.

Step two is only performed for the reference frame and then the determined nodes indices of the best basis in this step are used to prune the UWPT tree of the successive frames. Therefore, the process of creating FVs for the pixels in the successive frames is simplified.

Consider a pruned UWPT tree as illustrated in Fig. 3a. To simplify the understanding of feature vector generation procedure, the best basis subbands are put into the 3D coordinate system as shown in Fig. 3b. In this case, FV for the pixel located at position (x,y) can simply be generated (Fig. 3c).



$$(c) FV(x, y) = \{w_1^A(x, y), w_1^H(x, y), w_1^V(x, y), w_4^A(x, y), w_4^H(x, y), w_4^V(x, y), w_4^D(x, y)\}$$

Fig. 3. Feature vector selection a) A sample of best basis tree b) ordering subbands coefficients to extract the feature vector c) FV generation formula for pixel (x,y)

For instance, for the pixel located at position $(4,5)$, the corresponding feature vector, $FV(4,5)$, contains

$$w_1^A(4,5), w_1^H(4,5), w_1^V(4,5), w_4^A(4,5), w_4^H(4,5), w_4^V(4,5), w_4^D(4,5) \text{ respectively.}$$

Since s_r^i is located at the neighborhood of object's boundary, pixels in block Q_r^i can be classified into two different groups i) inside the object's boundary (foreground pixels) and ii) outside the object (background pixels). These two groups usually have different spatial domain characteristics (e.g. color, texture, etc). Therefore, we can use the luminance of each pixel to weight the generated FVs in order to achieve better pixel discrimination. To weight the FV of each pixel, we multiply the extracted FV from UWPT by the normalized luminance of the pixel. As the pixels' luminance values are in the range of 0 to 255, we divide each pixel luminance value to 255 to normalize them between 0 and 1. The result will be a joint FV (JFV) of the pixel.

The procedure to match Q_r^i in frame r to Q_n^i in frame n is as follows (Fig. 4).

Stage 2:

1. Assume a search window in frame n centered at $[x_r^i, y_r^i]$.
2. By performing the procedure in stage 1, we have the JFV for pixels in both Q_r^i and the search window
3. Choose a search block with the same size as Q_r^i to sweep the search window
4. Find the best match of Q_r^i in the search window by calculating the minimum sum of Euclidean distances between the search block and Q_n^i pixels' JFVs (e.g. full search algorithm in the search window).

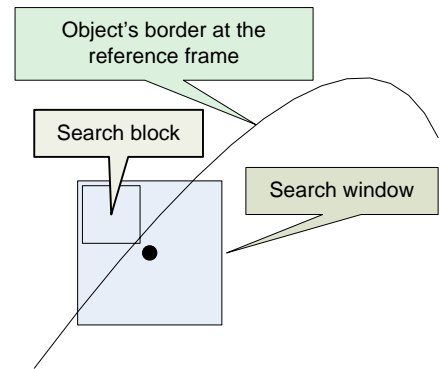


Fig. 4. Different elements involved in the search procedure in frame n with regards to Fig. 2

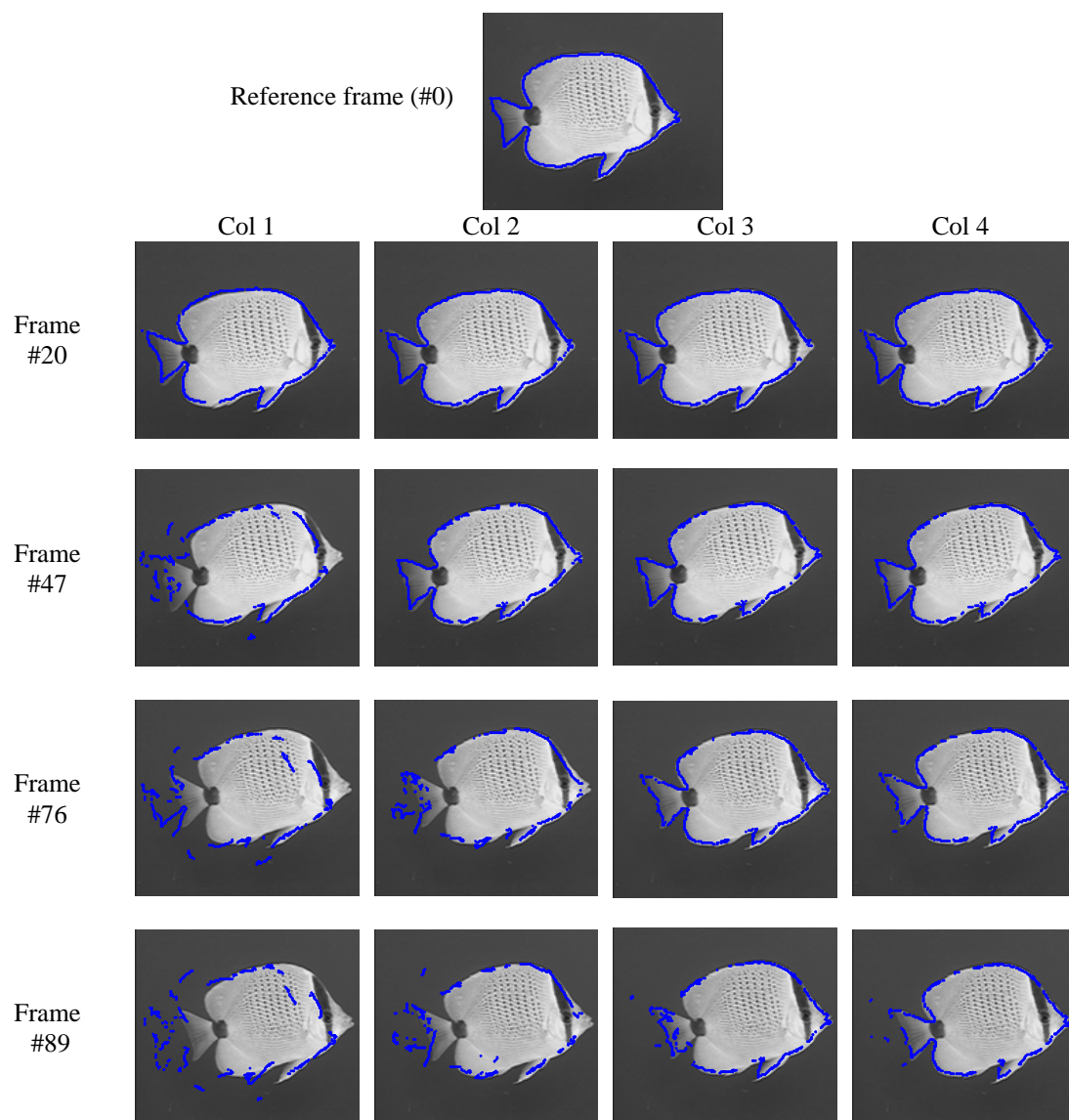
The procedure to search for the best-matched block is similar to the general block-matching algorithm except it exploits the generated JFV of the aforementioned procedure rather than the luminance of the pixel. In case the boundary block does not appear in the next frame (or has changed), the algorithm finds the best-matched block based on the search method.

It should be noted that in many cases the object movement in the successive frames makes the desired pixel out of the search area. There are two solutions for this problem. First, the search area should be set wide enough to encompass the pixel movement at new frames. This option requires extra computation and it is not clear how the size of search window should be increased. Second, the object's boundary at successive frames should be followed with an adaptive fixed size search area. Adaptation means to use the found pixel at the previous frame as the center of search window in the new frame. In this manner, the center of search window is not fixed and is updated at each frame to the matched pixel in the previous frame.

generation of the feature vector in the undecimated wavelet packet domain for each pixel and tracking this feature vector in a new frame, then success of this algorithm depends on how well this feature acts [15,16]. The performance of this part has been evaluated through some experiments over a number of video sequences under different conditions. The generated JFV has been tested in various mixed types of object's transformation such as translation, rotation, and scaling. Different values for block sizes, search areas, and levels of UWPT decomposition have been tested. Throughout the experiments we have assumed that there is no scene cut. In case of a scene cut detection, the reference frame should be updated and a new user intervention is required.

4. EXPERIMENTAL RESULTS

Since the core of the proposed algorithm is the



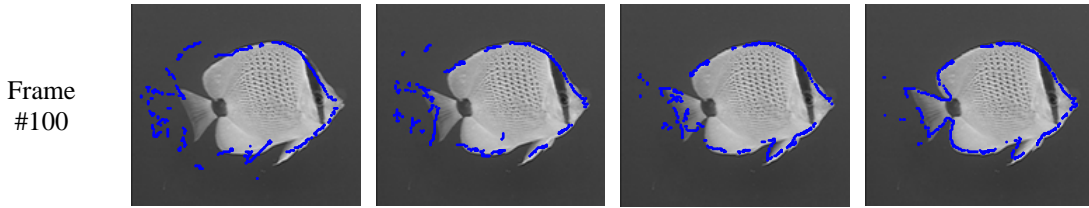


Fig. 5. The effect of search window size on the Bream sequence.
 Col 1: Fixed 15x15 search window, Col 2: Fixed 27x27 search window, Col 3: Fixed 47x47 search window, Col 4: Adaptive 15x15 search window

In our experimental studies, the user generates the object's shape in the reference frame by defining the border of the object and then a function automatically generates the selected feature points as the input of the algorithm. Since there is no universally agreed method of evaluating the performance of object temporal tracking, we compare them subjectively. The outcome of tracking in various conditions is shown in bold dots at the object border without interpolation.

Translation is the simplest form of transformation that normally occurs in object tracking [7]. The Bream sequence is a good example of translation with a small amount of zooming. We have tested the algorithm from frames #0 to #100 with a block size of 5x5 pixels, 4 levels of UWPT tree decomposition, wavelet family of biort2.2 [14], Shannon entropy criteria and different search window sizes (15x15, 27x27, and 47x47). In BiorNr,Nd, Nr and Nd represent the number of vanishing moments for the synthesis and analysis wavelets [17].

The first frame (frame #0) of the sequence was considered as the reference frame. As the object (the Fish) moves to the right in successive frames, the search area has to be set wider to encompass the desired pixels in the search area. Otherwise, the actual pixel position will be out of search area and so cannot be found. For example loss of tracking in Fig. 5, Col 1 frame #47 and Col 2 frame #76 are due such mismatch. Enlarging the

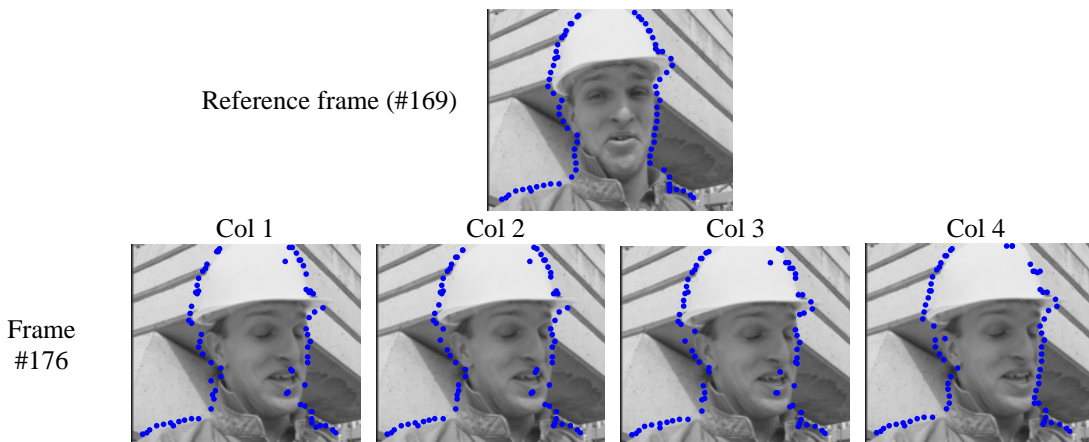
search window size gives better results but at the expense of an increase in the computational complexity (see Fig. 5, Cols 2 and 3). As shown in Fig. 5 Col. 3, in spite of wide search area (47x47 pixels) the JFV is good enough to be used to find the matched pixel well.

When the proposed adaptive search window center method is applied the results of 15x15 block size are as good as the non-adaptive wide search window of size 47x47 (compare Fig. 5 Col 3 and Col 4).

As shown in Fig. 5 the sequence has small deformation (frame #47) and zooming out (frame #100) which the proposed method can handle them successfully. Therefore, the algorithm is robust against translation and scaling.

Figs. 6 and 7 show the simulation results for different levels of the UWPT tree decomposition and block size respectively. The simulation has been conducted on frames #169 to #200 of the Foreman sequence with the wavelet family of biort2.2, Shannon entropy criteria and a search window size of 15x15 pixels. In both figures frame #169 of the sequence was considered as the starting and the reference frame. The chosen video clip includes different object's shape deformation such as translation, rotation, and a small scaling.

Fig. 6 illustrates the reference frame (frame 169 of the foreman sequence) which the object's boundary is



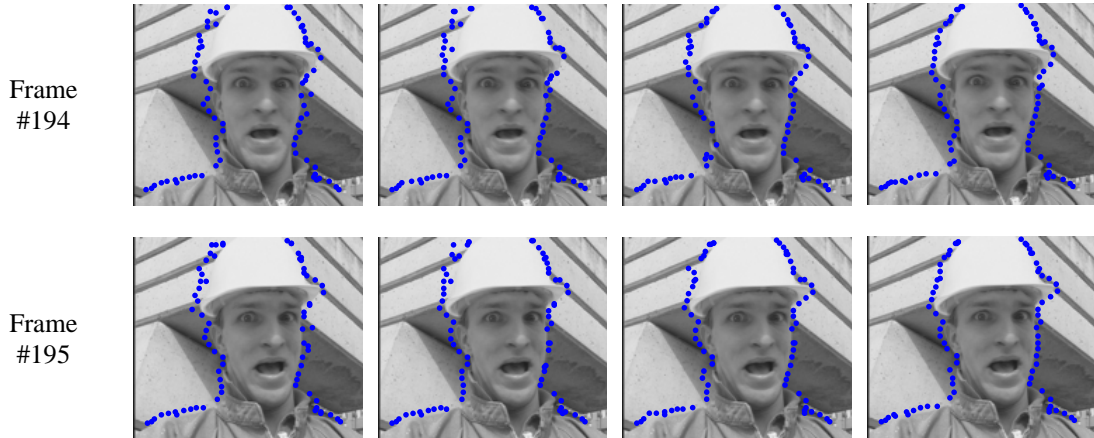


Fig. 6. The effect of different levels of UWPT tree decomposition on Foreman sequence, Col 1: 1 level, Col 2: 2 levels, Col 3: 3 levels, Col 4: 4 levels

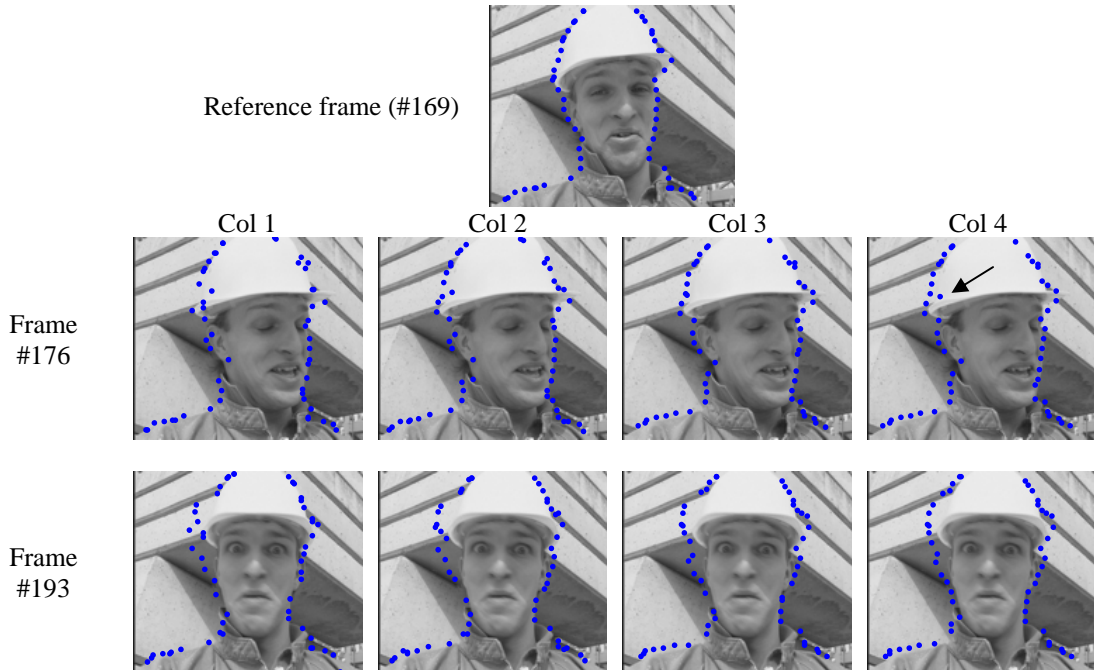


Fig. 7. Effect of different block sizes, Col 1: Single pixel, Col 2: 3x3 block size, Col 3: 5x5 block size, Col 4: 7x7 block size.

determined manually and the results of object's boundary temporal tracking at frame numbers 176, 194 and 195 are shown for different levels of the UWPT tree from 1 to 4. Frame #176 has a rotation in respect to the reference frame. The results for depths 1, 2 and 3 of the UWPT tree on these frames are similar and are not satisfactory but the result for the tree depth of 4 is acceptable. The simulation results for frames #194 and #195 show that decomposition levels of 3 and 4 give similar results but level 4 acts better in some point which leads to more accurate object mask. An UWPT tree with depth 3 has 64 more nodes than depth 2 and depth 4 has 256 more nodes than depth 3. Therefore, the results with depth 4 are better because of more exact FVs generation.

Fig. 7 shows the snapshots for temporal object boundary tracking at frames #176 and #193 for different values of the block size (single pixel, 3x3, 5x5, 7x7). The number of feature points at different columns of Fig. 7 is equal. For frame #176, Cols 3 and 4 are very similar and are better than others. In this frame, 7x7 block size (Col 4) has found a point on the left side of the Forman's hat by mistake (indicated by an arrow). The reason is that larger search areas are more prone to errors. Besides, a 7x7 block size requires more computations than a 5x5 one. The results of block sizes 5x5 and 7x7 in frame #193 are again similar. Therefore, a block size of 7x7 can't give much performance in comparison to a 5x5 block size and so the 5x5 block size is the best case for feature point tracking.

5. CONCLUSIONS

A new adaptive semi-automatic object extraction method based on pixel features in the wavelet domain has been proposed. Temporal object tracking has been achieved by tracking small squares centered at the selected pixels near the object's boundaries. For this purpose, a special feature vector corresponding to each pixel in the frame has been generated. These features are extracted from the Undecimated Wavelet Packet Transform (UWPT) of the frame. The generated FVs combined with the pixels' luminance values result in the weighted feature vectors. Considering the shift invariant property of the UWPT, the full search of the reference block in the search window at the next frame could simply be carried out. An adaptive search window center updating procedure makes the search method efficient.

A simple search method along with the aforementioned properties of the UWPT makes the pixel tracking successful even in the presence of translation, rotation and zooming. Suitable features for selected pixels have been formed due to the best basis selection algorithm.

The experimental results indicate a 5x5 block size, 4 levels of the UWPT tree decomposition, and a 15x15 pixels search window are the best object tracking parameters.

Based on the properties of the UWPT, existence of individual robust JFV for each pixel, and an adaptive search window center, this method can tolerate complex object's boundary transformation including translation, rotation, scaling and even small deformation well.

The proposed method can be integrated into a semi-automatic video object plane (VOP) generation system that can be used in the MPEG-4 encoders and Content-based multimedia indexing systems.

ACKNOWLEDGMENTS

This research has been funded by the Advanced Information and Communication Technology Center (AICTC) of Sharif University of Technology.

6. REFERENCES

[1] J. Guo, C.-C.J Kuo, *Semantic Video Object Segmentation for Content-based Multimedia Applications*, Kluwer Academic Publisher, 2001.

[2] D. Zhang, G. Lu, "Segmentation of Moving Objects in Image Sequence: A Review," *Circuits, Systems and Signal Processing (Special issue on multimedia communication services)*, vol. 20, no. 2, pp. 143-183, 2001.

[3] A. Cavallaro, "From visual information to knowledge: semantic video object segmentation, tracking and description," PhD thesis, EPFL Switzerland, 2002.

[4] 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.

[5] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models, ," *Int. J. Computer Vision*, vol. 1, no. 4, pp.

321--331, 1987

[6] B. Günsel, A. M. Tekalp, and P. J.L. van Beek, "Content-based access to video objects: Temporal segmentation, feature extraction and visual summarization," *Signal Processing (special issue)*, vol. 66, no. 2, pp. 261-280, April 1998.

[7] 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

[8] 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*, 2001, pp. 543-550.

[9] T. Haenselmann, C. Schremmer, W. Effelsberg, "Wavelet-Based Semi-Automatic Segmentation of Image Objects," *Signal and Image Processing (SIP)*, pp. 387-392, Las Vegas, Nevada, USA, 2000.

[10] B. Marcotegui, et al, " A Video Object Generation Tool Allowing Friendly User Interaction," *IEEE Conference on Image Processing*, Kobe, Japan, 1999.

[11] 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.

[12] C. Kim, J. N. Hwang, "Fast and Automatic Video Object Segmentation and Tracking for Content-Based Applications," *IEEE Transaction for circuit and systems for video Technology*, Vol. 12, No. 2, pp. 122-129, 2002.

[13] Y. Tsaig, A. Averbuch, "Automatic Segmentation of Moving Objects in Video Sequences: A Region Labeling Approach," *IEEE Transaction for circuit and systems for video Technology*, Vol. 12, No. 7, pp. 597-612, 2002.

[14] 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.

[15] 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, 2003.

[16] 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, Feb. 2005.

[17] I. Daubechies, *Ten lectures on wavelets*, CBMS, SIAM, 61, 1994, 271-280.