Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 243534, 18 pages
doi:10.1155/2008/243534

Research Article

Object Tracking in Crowded Video Scenes Based on the Undecimated Wavelet Features and Texture Analysis

M. Khansari,[1] H. R. Rabiee,[1] M. Asadi,[1] and M. Ghanbari[1,2]

[1] *Digital Media Lab, AICTC Research Center, Department of Computer Engineering, Sharif University of Technology, Azadi Avenue, Tehran 14599-83161, Iran*
[2] *Department of Electronic Systems Engineering, University of Essex, Colchester CO4 3SQ, UK*

Correspondence should be addressed to H. R. Rabiee, rabiee@sharif.edu

Received 9 October 2006; Revised 21 May 2007; Accepted 8 October 2007

Recommended by Jacques G. Verly

We propose a new algorithm for object tracking in crowded video scenes by exploiting the properties of undecimated wavelet packet transform (UWPT) and interframe texture analysis. 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 are used to 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 interframe 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. Moreover, the tracking algorithm is robust to Gaussian and quantization noise processes. Experimental results show that the proposed algorithm has good performance for object tracking in crowded scenes on stairs, in airports, or at train stations in the presence of object translation, rotation, small scaling, and occlusion.

1. INTRODUCTION

Object tracking is one of the challenging problems in image and video processing applications. With the emergence of interactive multimedia systems, tracked objects 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 postproduction.

Video object tracking techniques vary according to user interaction, tracking features, motion-model assumption, temporal object tracking, and update procedures. The target representation and observation models are also very important for the performance of any tracking algorithm. In general, the temporal object tracking methods can be classified into four groups: region-based [1], contour/mesh-based [2], model-based [3, 4], and feature-based methods [5, 6]. Two major components can be distinguished in all of the tracking approaches; target representation/localization and filtering/data association. The former is a bottom-up process dealing with the changes in the appearance of the object, while the latter is a top-down process dealing with the dynamics of the tracking [7]. Feature-based algorithms, along with Kalman or particle filters, are widely used in many object tracking systems [4, 7].

Color histogram is an example of a simple and good feature-based method for object tracking in the spatial domain [7–12]. The color histogram techniques are robust to noise and they are typically used to model the targets to combat partial occlusion and nonrigidity of objects. However, color histogram only describes the global color distribution and ignores spatiality or layout of the colors, and the tracked objects are easily confused with a background having similar colors. Moreover, it cannot deal easily with illumination changes and full occlusion. Therefore, feature description based on color histogram for target tracking, particularly in the crowded scenes where similar small objects exist (e.g., heads of the crowd), will most likely fail.

Mean-shift tracking algorithms that use color histogram have been successfully applied in object tracking and proved to be robust to appearance changes [7, 10, 13, 14]. However, these techniques need more sophisticated motion filtering to handle 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. More recently, color histogram with spatial information has been used by some researchers [15, 16]. Color histogram has also been integrated into probabilistic frameworks such as Bayesian and particle filters [9, 11, 17, 18] or kernel-based models along with Kalman filters [7]. Comparative evaluation of different tracking algorithms shows that among histogram-based techniques, the mean-shift approach [13] leads to the best results in absence of occlusions, and probabilistic color histogram trackers are more robust to partial or temporary occlusions over a few frames than the other well-known techniques [12]. In addition, the kernel-based histogram tracker performs better in longer sequences [7]. A good discussion on the state-of-the-art object tracking under occlusion can also be found in [19].

In recent years, feature-based techniques in the wavelet domain have gained more attention in object tracking [20–23]. In [20], an object in the current frame is modeled by using the highest energy coefficients of Gabor wavelet transform as local features, and the global placement of the feature point is achieved by a 2D mesh structure around the feature points. In order to find the objects in the next frame, the 2D golden section algorithm is employed.

In [21], a wavelet subspace method for face tracking is presented. At the initial stage, a Gabor wavelet representation for the face template is created. The video frames are then projected into this subspace by wavelet filtering techniques. Finally, the face tracking is achieved in the wavelet subspace by exploiting the affine deformation property of Gabor wavelet networks and minimization of Euclidean distance measure.

In [22], a particle filter algorithm for object tracking using multiple color and texture cues has been presented. The texture features are determined using the coefficients of a three-level conventional discrete wavelet transform expansion of the region of interest. In addition, a Gaussian sum particle filter based on a nonlinear model of color and texture cues is also presented.

In [23], a real-time multiple object tracking algorithm is introduced. In their algorithm, instead of using the wavelet coefficients as object features, the original frame is only preprocessed using a two-level discrete wavelet transform to suppress the fake background motions. The approximation band of the wavelet transform is then used to compute the difference image of successive frames. Then, the concept of connected components is applied to the difference image to identify the objects. The classified objects are then marked by a bounding box in the original approximation image, and some color and spatial features are extracted from the bounding box. These features are then used to track the objects in successive frames.

Most of the previous work based on wavelet transform has been evaluated on simple scenarios: either a talking head with various movements or face expressions [20, 21] or walking people who might have been occluded by another person in the reverse direction in a short period of time [22, 23] and not for more complex scenes such as dense crowds of very close and similar objects with short- or long-term occlusions. The general drawback of these techniques is that similar nearby objects (e.g., heads in the crowd) with short- and long-term occlusions may impair their reliability. Other challenging issues of the aforementioned methods are robustness against noise and stability of the selected features in presence of various object transformations and occlusions.

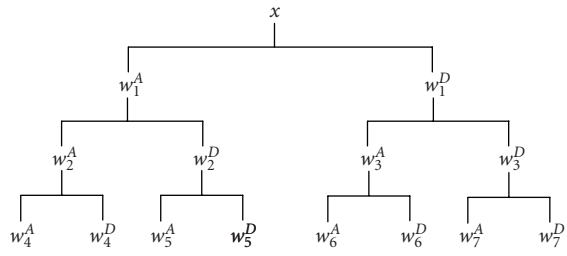
In this paper, we present a new algorithm for tracking arbitrary user-defined regions that encompass the object of interest in the crowded video scenes. It is based on feature vectors generated via the coefficients of the undecimated wavelet packet transform (UWPT) for target representation/localization and filtering/data association are achieved through an adaptive search window by using an interframe texture analysis scheme. The key advantage of UWPT is that it is redundant and shift-invariant, and it gives a denser approximation to continuous wavelet transform than that provided by the orthonormal discrete wavelet transform [24, 25].

The main contribution of this paper is the adaptation of a feature vector generation and block matching algorithm in the UWPT domain [26] for tracking objects [27, 28] in crowded scenes in presence of occlusion [29] and noise [30, 31]. In addition, it uses an interframe texture analysis scheme [32] to update the search window location for the successive frames. In contrast to the conventional methods for solving the tracking problem that use spatial domain features, it introduces a new transform domain feature-based tracking algorithm that can handle object movements, limited zooming effects, and, to a good extent, occlusion. Moreover, we have shown that the feature vectors are robust to various types of noise [30, 31].

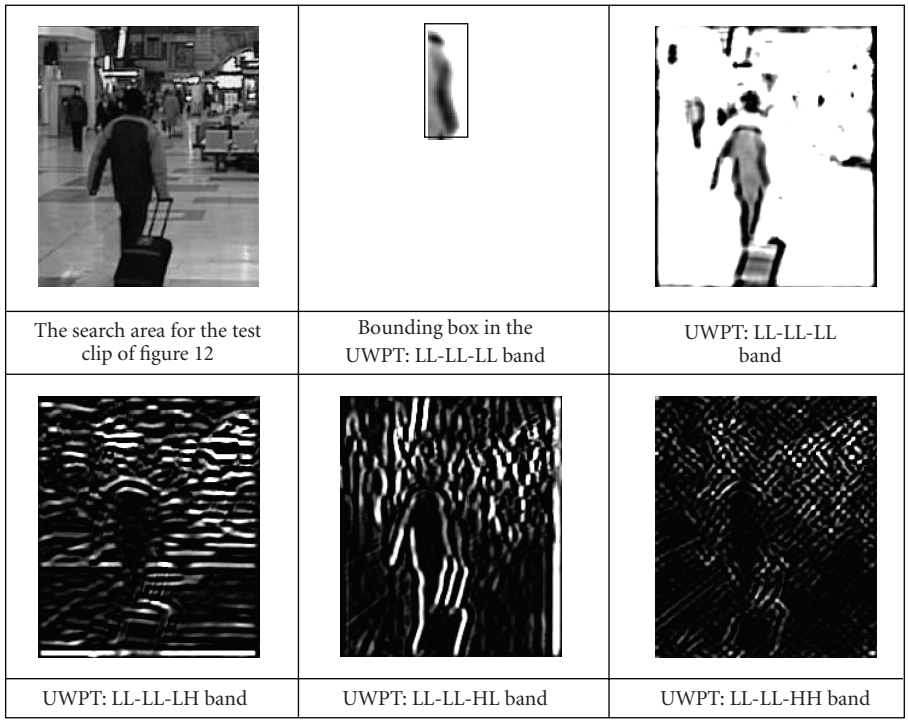
Organization of the rest of this paper is as follows. After presenting an overview of the UWPT in Section 2, the elements of the proposed algorithm are described in Section 3. These elements include feature generation, temporal tracking, and search window updating mechanism. Performance of the proposed algorithm under various test conditions is evaluated in Section 4. Finally, Section 5 provides the concluding remarks and the future work.

2. OVERVIEW OF THE UWPT

The process of feature selection in the proposed algorithm relies on the multiresolution expansion of images. The idea is to represent an image by a linear combination of elementary building blocks or atoms that exhibit some desirable properties. Recently, there has been a growing interest in the representation and processing of images by using dictionaries of basis functions other than the traditional dictionary of sinusoids such as discrete cosine transform (DCT). These new sets of dictionaries include Gabor functions, chirplets, warplets, wavelets, and wavelet packets [25, 33–35]. In contrast to DCT, the discrete wavelet transform (DWT) gives good frequency selectivity at lower frequencies and good time selectivity at higher frequencies. This tradeoff in the time-frequency (TF) plane is well suited to the representation of many natural signals and images that exhibit short-duration high-frequency and long-duration low-frequency events. One well-known disadvantage of the DWT is the lack



(a)



(b)

Figure 1: (a) Undecimated wavelet packet transform tree for one-dimensional signal x, where A stands for the approximation (lowpass) signal and D for the detailed signal (highpass). (b) Sample bands of UWPT for the search area for the test clip of Figure 12 (L stands for lowpass and H for highpass filtered images).

of shift invariance. The reason is that there are many legitimate DWTs for different shifted versions of the same signal [25].

Wavelet packets were introduced by Coifman and Meyer as a library of orthogonal bases for $L^2(R)$ [24]. Implementation of a "best-basis" selection procedure for a signal (or family of signals) requires introduction of an acceptable "cost function," which translates "best" into a minimization process. The cost function can be simplified in an additive nature when entropy [24] or rate distortion [36] is used. The cost function selection is related to the specific nature of the application at hand. Entropy, for example, may constitute a reasonable choice if signal classification, identification, and compression are the applications of interest. A major deficiency of decimated wavelet packet is sensitivity to the signal location with respect to the chosen time origin, that is, lack of shift-invariance property.

The desired transform for object tracking application should be linear and shift-invariant. The wavelet transform, which is both linear and shift-invariant, is the undecimated wavelet packet transform (UWPT) [25, 35]. Moreover, the UWPT expansion is redundant and provides a denser approximation compared to the approximation provided by the orthonormal discrete wavelet transform [24, 25].

From the implementation point of view in the context of filter banks, in addition to the lowpass band, we repeat the filtering on the highpass band without any downsampling (decimation). The result is a complete undecimated wavelet packet transform. A tree representation and sample bands of UWPT are depicted in Figure 1.

The computational complexity of the UWPT is as follows [25]:

$$\begin{aligned} \mathrm{NM}_{\mathrm{UWPT}}(N,L,M) &= M(2^{L+1}-1)N, \\ \mathrm{NA}_{\mathrm{UWPT}}(N,L,M) &= M(2^{L+1}-1)N. \end{aligned} \tag{1}$$

In the above formulas, the length of the input signal is N, the length of the quadrature mirror filter (QMF) for creating the subbands is M, and the number of decomposition levels is L such that $L \leq \log_2 N$. NM and NA represent "number of multiplications" and "number of additions" that are needed to convolve the signal with both highpass and lowpass QMFs, respectively. It is important to note that there are a number of fast and real-time algorithms to compute DWT and UWPT of natural signals and images [25].

3. THE PROPOSED ALGORITHM

3.1. Overview of the proposed algorithm

In our algorithm, object tracking is performed by temporal tracking of a rectangle around the object at a reference frame. The algorithm is semi-automatic in the sense that the user draws a rectangle around the target object or specifies the area around pixels along the boundary of the object in the reference frame. A general block diagram of the algorithm is shown in Figure 2.

Initially, the user specifies a rectangle around the boundary of the object 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 FVs 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.

The main advantages of this algorithm are as follows.

(1) It can track both rigid and nonrigid objects without any preassumption, training, or object shape model.
(2) It can efficiently track the objects in the crowded video sequences such as crowds on stairs, in airports, or at train stations.
(3) It is robust to different object transformations such as translation and rotation.
(4) It is robust to different types of noise processes such as additive Gaussian noise and quantization noise.
(5) The algorithm can handle object deformation due to perspective transform.
(6) Partial or short-term full occlusion of the object can be successfully handled due to the robust transform domain FVs and temporal texture analysis.

3.2. The feature vector generation

In the first step, the wavelet packet tree for the desired object in the reference frame is generated by the UWPT. As mentioned in the previous section, the UWPT has two properties that make it suitable for generating invariant and robust features in image processing applications [26–31].

(1) It has the shift-invariant property. Consequently, feature vectors that are based on the wavelet coefficients in frame t can be found again in frame $t + 1$, even in the presence of partial occlusion.
(2) All the subbands in the decomposition tree have the same size equal to that of the input frame (no downsampling), which simplifies the feature extraction process (see Figure 3).

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

As shown in Figure 1, there are many redundant representations of a signal x, by using different combinations of subbands. For example, $x = (w_1^A, w_1^D)$, $x = (w_2^A, w_2^D, w_1^D)$, and $x = (w_4^A, w_4^D, w_2^D, w_1^D)$ are all representations of the same signal.

The procedure for generating an 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 UWPT is constructed with zero padding when needed).
(2) Perform basis selection from the approximation and detail subbands. Different pruning strategies can be applied on the tree to generate the FV as follows.
   (a) Apply entropy-based algorithms for the best basis selection [24, 36] and prune the wavelet packet tree. The goal of this type of basis selection is removing the inherent redundancy of UWPT and providing a denser approximation of the original signal. Entropy-based basis selection algorithms have been mostly used in compression applications [36].
   (b) Select leaves of the expansion tree for representing the signal. This signal representation includes the greatest number of subbands which imposes an unwanted computational complexity to solve our problem. For example, in Figure 1, $x = (w_4^A, w_4^D, w_5^A, w_5^D, w_6^A, w_6^D, w_7^A, w_7^D)$. We should note that, in the presence of noise, this set of redundant features may be used to enhance the performance of the tracking algorithm.
   (c) 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. This type of basis selection gives more weight to the approximations which are useful for our intended application. For example, in Figure 1, we may let $x = (w_4^A, w_4^D)$ or $x = (w_4^A)$. For our application, this type of basis selection is more reasonable, because the comparison in the temporal tracking part of the algorithm is carried out between two regions that are represented by similar approximation and detail subbands.

   The output of this step is an array of node index numbers of the UWPT tree that specifies the selected basis for the successive frame manipulations.
(3) The FV for each pixel in region r can be simply created by selecting the corresponding wavelet coefficients in the selected basis nodes of step (2). Therefore, the

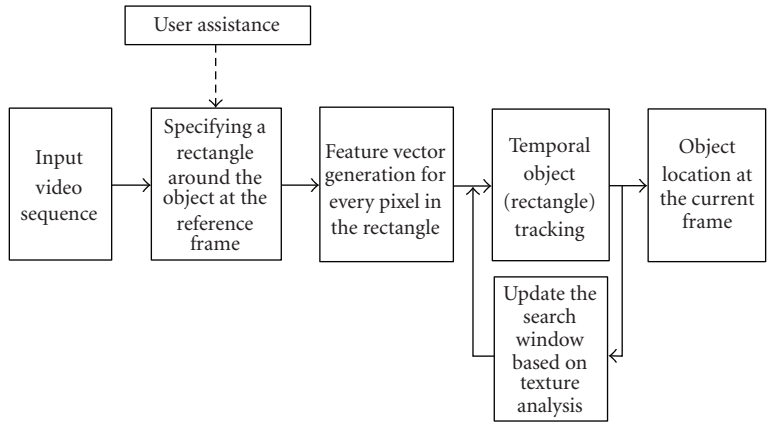


Figure 2: A block diagram of the proposed algorithm.

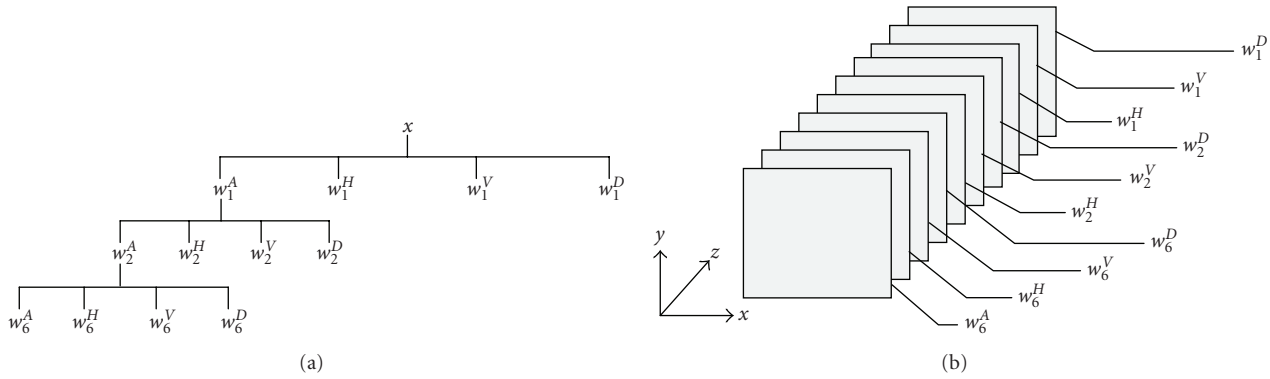


Figure 3: Feature vector selection: (a) a selected basis tree, (b) ordering of the subband coefficients to extract the feature vector, (c) FV generation formula for pixel (x, y).

number of elements in the FV is the same as the number of selected basis nodes.

Consider a pruned UWPT tree and the 3D representation of the selected basis subbands in Figures 3(a) and 3(b), respectively. In this case, FV for the pixel located at position (x, y) can simply be generated as shown in Figure 3(c).

3.3. The 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. As stated in the previous section, we can construct a feature vector that corresponds to each pixel in the region around the object. These FVs 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 the corresponding region in frame $t + 1$ is performed through the full search of the region in a search window in frame $t + 1$, which is adaptively determined by the texture analysis approach that will be discussed in Section 3.4 [32].

More specifically, every pixel in region r may undergo a complex transformation within successive frames. In general, it is hard to find each pixel using variable and sensitive spatial domain features such as luminance, texture, and so forth. Our approach to track r in frame t makes use of the aforementioned FV of each pixel and Euclidean distances to find the best matched regions as described below.

The procedure to match r in frame t to $r + 1$ in frame $t + 1$ is as follows.

(1) Generate an FV for pixels in both region r and the search window by using the procedure presented in Section 3.2.
(2) Sweep the search window with a search region that has the same dimension as r.
(3) Find the best match for r in the search window by calculating the minimum sum of the Euclidean distances between the FVs of the pixels of search regions and FVs of the pixels within region r (e.g., full search algorithm in the search window).

The procedure to search for the best matched region is similar to the general block-matching algorithm, except that it exploits the generated FV of a pixel rather than its luminance. Therefore, when some pixels of r do not appear in the next frame (due to partial occlusion or some other changes), our algorithm is still capable of finding the best matched region based on the above search procedure.

3.4. The search window updating mechanism

The change of object location requires an efficient and adaptive search window updating mechanism for the following reasons.

(1) The proper search window location ensures that the object always lies within the search area and thus prevents loss of the object inside the search window.
(2) A location-adaptive fixed size search window decreases computational complexity that results due to a large and variable size search window [27].
(3) If a moving target is occluded by another object, use of direction of motion may alleviate the occlusion problem.

To attain an efficient search window updating mechanism, different approaches can be employed. Most of these techniques use spatial and/or temporal features to guide the search window and to find the best match for it with the least amount of computation [32].

We have considered two different mechanisms for updating the location of the search window as follows.

(1) Updating the center of the search window based on the center of the rectangle around the object at the current frame. In this case, the center of search window is not fixed and it is updated at each new frame to the center of the matched rectangle at the previous frame. This approach is simple, but loss of tracking propagates through the frames [28]. In addition, when occlusion occurs at the current frame, the object may not be found correctly in the following frames.
(2) Another approach is to estimate the direction and the speed of motion of the object to update the location of the search window.

In this paper, we have selected the latter approach as our updating strategy by using the interframe texture analysis technique [32]. To find the direction and 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 [32]. Finally, the direction and speed of the motion are estimated through the use of temporal difference histogram of coarseness and directionality.

3.4.1. 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.

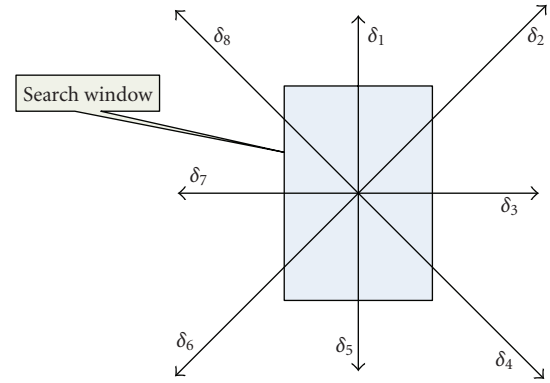


Figure 4: Distance assignment in the different directions to find the maximum inverse difference moment (IDM).

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 $\delta = (\Delta x, \Delta y)$ of the current search window center in the next frame. We assume N_x and N_y are the width and height of the search window, respectively. 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:

$$\mathrm{ATD}_\delta(x, y) = \left| SA_t(x, y) - SA_{t+1}(x + \Delta x, y + \Delta y) \right|, \quad (2)$$

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

3.4.2. The search window direction

Assume that the search window is a rectangular block. Consider eight different blocks at the various directions with distance δ_i from the center of search window at the current frame (see Figure 4).

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 (3). The inverse difference moment, IDM, is the measure of homogeneity and it is defined as

$$\mathrm{IDM} = \sum_{i=0}^{M-1} \frac{p_\delta(i)}{i^2 + 1}. \quad (3)$$

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 nonzero values of p_{δ_i}, and it can be used to identify the main texture direction. If a texture is directional, it is coarser in one direction than in 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 IDM should be found:

$$\mathrm{IDM}_{\max} = \max\{\mathrm{IDM}_i\}, \quad i = 1, 2, \ldots, 8. \tag{4}$$

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

3.4.3. The 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 it is given by

$$\mathrm{TCON} = \sum_{i=0}^{M-1} i^2 p_\delta(i), \tag{5}$$

where M is the number of gray-level values in each frame as stated in Section 3.4.1.

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 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

$$\mathrm{LCON} = \frac{1}{\mathrm{SW}} \sum_{\mathrm{SW}} \left[g(x, y) - \overline{g}\right]^2, \tag{6}$$

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

$$S = k \frac{\mathrm{TCON}}{\mathrm{LCON}}, \tag{7}$$

where k is a constant with empirically selected values. The average motion speed, S, in (7) is not only independent of the size of the moving objects but also invariant to the orientation of their texture. The value of S approaches zero for stationary parts of the picture such as background, independent of their texture contents [32].

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

$$\begin{aligned} R_{j-1} &= S_{j-1} - \mathrm{Disp}_{j-1}, \\ \mathrm{Disp}_j &= \lfloor S_j + R_{j-1} \rfloor. \end{aligned} \tag{8}$$

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 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.

Several objective evaluation measures have been suggested in the literature [37, 38]. In this section, we have used the ground truth information to objectively evaluate the performance of our algorithm.

The experimental results of the proposed tracking algorithm have been compared with the conventional wavelet transform (WT) as well as the well-known color histogram-based tracking algorithms with two different matching distance measures, that is, chi-squared and Bhattacharyya. In the figures, color histogram-based tracking with the chi-squared distance measure is denoted by CHC, the color histogram-based tracking with Bhattacharya distance measure by CHB, wavelet transform by WT, and the proposed algorithm by UWPT. We have used biorthogonal wavelet bases, which are particularly useful for object detection and generation of the UWPT tree. In fact, the presence of spikes in the biorthogonal wavelet bases makes them suitable for target tracking applications [39]. In all experiments, we have used 3 levels of UWPT tree decomposition with the *Bior*2.2 wavelet [35]. In the color histogram-based algorithm implementation, the number of color bins was set to 32.

To evaluate the algorithms in a real-environment setting, we have applied them to different real-time video clips of Tehran Metro Stations in cooperation with the Tehran Metro authorities as well as to a longer sequence extracted from the dataset S7 of IEEE PETS 2006[1] workshop. These video clips 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, object deformation, and object rotation. In all the snapshots, solid rectangles correspond to the rectangles around the objects, and the rectangles with dashed lines represent the search window. Note the difficulty in tracking heads in a crowded scene, as there are several nearby similar objects.

In addition, for each tracking result, the corresponding set of video clips is available through Internet[2] for more detailed subjective evaluation. Moreover, we have defined

[1] Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance.

[2] http://ce.sharif.edu/~khansari/JASP/videoclips.html.

Reference: frame no. 245

(a)

UWPT: frame no. 252

UWPT: frame no. 309

(b)

CHC: frame no. 252

CHC: frame no. 309

(c)

CHB: frame no. 252

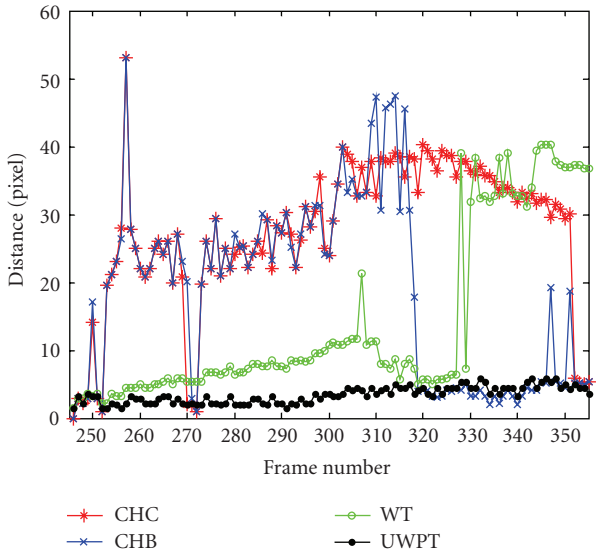
CHB: frame no. 309

(d)

WT: frame no. 252

WT: frame no. 309

(e)

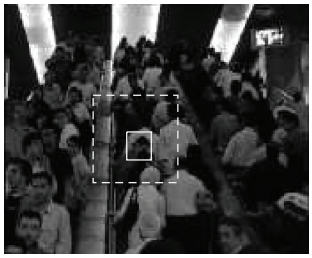


(f)

Figure 5: Tracking the head of a man coming down the stairs in a crowded metro station. (a) Reference frame, (b) UWPT, (c) CHC, (d) CHB, (e) WT, (f) objective evaluation: distance between the center of tracked bounding box and the expected center, for all methods.

Reference: frame no. 97

(a)



UWPT: frame no. 139

UWPT: frame no. 160

(b)

CHC: frame no. 139

CHC: frame no. 160

(c)

CHB: Frame no. 139

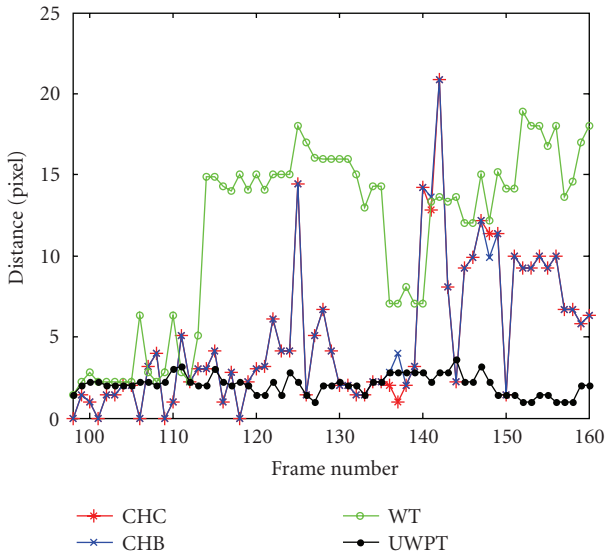
CHB: Frame no. 160

(d)

WT: frame no. 139

WT: frame no. 160

(e)



(f)

Figure 6: Tracking a man going up the stairs, in presence of partial occlusion and zooming out effects. (a) Reference frame, (b) UWPT, (c) CHC, (d) CHB, (e) WT, (f) objective evaluation: distance between the center of tracked bounding box and the expected center, of all four methods.

Reference: frame no. 635

(a)

UWPT: frame no. 657 UWPT: frame no. 660

UWPT: frame no. 672 UWPT: frame no. 678

(b)

CHC: frame no. 672 CHC: frame no. 678 CHB: frame no. 672 CHB: frame no. 678

(c) (d)

WT: frame no. 672 WT: frame no. 678

(e)

Figure 7: Tracking a man moving up the stairs, with full occlusion in some frames: (a) UWPT, (b) CHC, (c) CHB, (d) WT.

a measure for objective evaluation of tracking techniques based on the Euclidian distance of the center of gravity of the tracked and actual objects. Here, at the start of tracking, a bounding rectangle located at the center of the gravity of the desired object is selected. 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.

Figure 5 shows the snapshots of tracked head of a man, shown in frame 245, coming down the stairs in a crowded metro station. The size of the rectangle around the object was set to 19×13 pixels, and the size of the search window was 57×51 pixels. Empirical parameters to find the direction and speed of the motion for updating the search window were set to $d = 1$ and $k = 6$. The object is stepping down the stairs with a constant speed, small amount of zooming, and

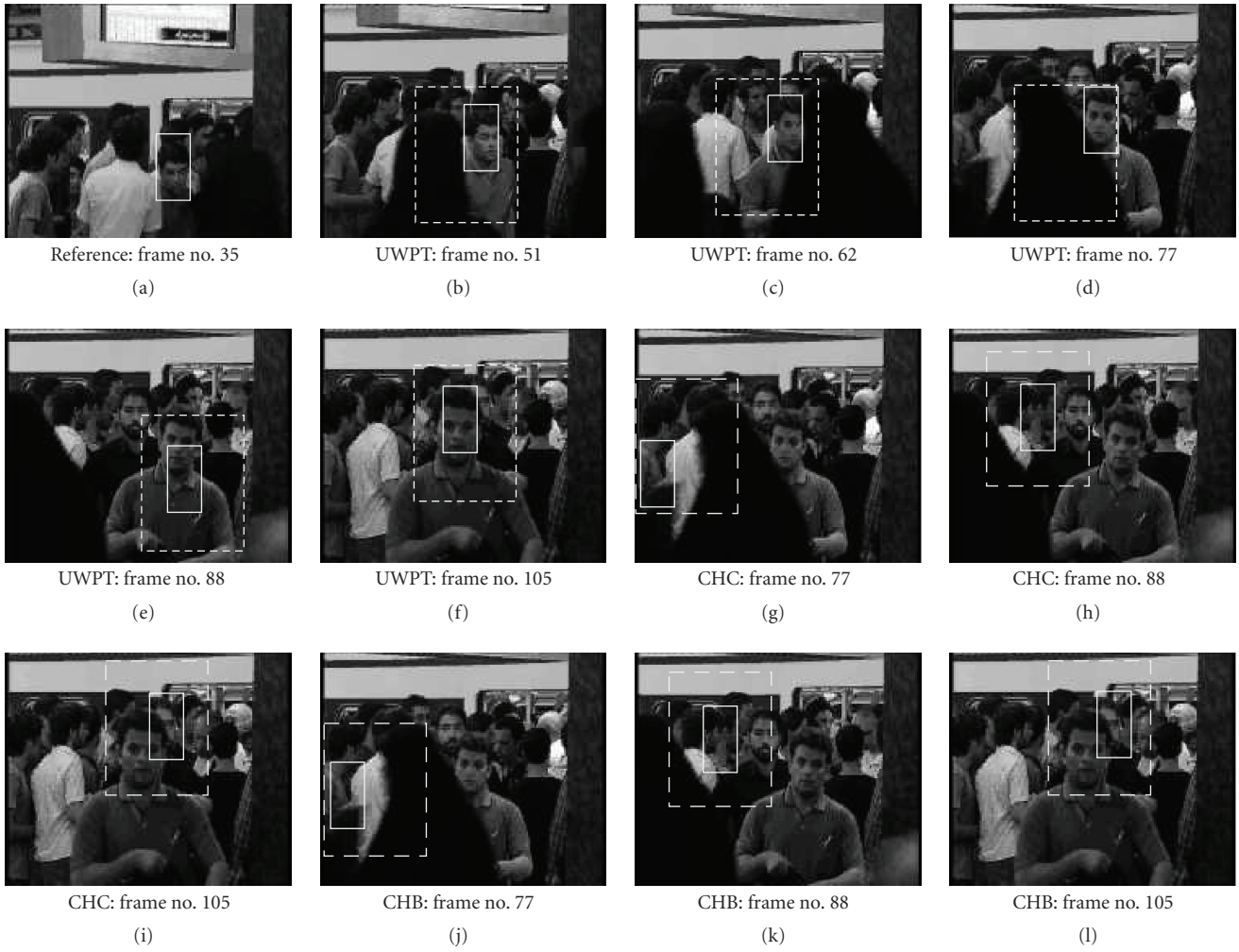


Figure 8: Tracking a man getting off the train in various kinds of partial and long-duration full occlusions and zooming in effects: UWPT, CHC, and CHB.

some cross-movements. There is no partial or full occlusion of the object in this case, but there are similar faces within the search window that complicate the tracking process.

As the results show, the object of interest has been successfully tracked by UWPT despite the presence of several similar objects inside the search window (see Figure 5(b)). The WT and color histogram-based algorithms (CHC and CHB) have mistakenly tracked a wrong object in frame no. 309 (see Figures 5(c), 5(d), and 5(e)). For detailed analysis, we present the result of our objective measure on a frame-by-frame basis in Figure 5(f). For both CHC and CHB, from frame no. 252, their Euclidian distance of the center of gravity is well above the bounding rectangle size, implying that the objects are totally miss-tracked. The color histogram-based algorithms are able to find the objects in a random manner, however in some cases their behavior is unpredictable (for instance, see the CHB tracking results before and after frame no. 320 in Figure 5(f)). The WT-based tracking has also miss-tracked the object after frame 330, and its performance is worth than UWPT for the previous frames. On the contrary, our proposed algorithm is tracking the object in a consistent and stable manner.

Figure 6 shows the result of tracking a person moving up the stairs and away from the camera in a metro station. Frame no. 97 was the reference frame (see Figure 6(a)), the size of the rectangle around the object was 17×15 pixels, and the size of the search window was 51×49 pixels. Empirical parameters to find the direction and the speed of the motion for updating the search window were set to $d = 1$ and $k = 6$.

The target object is stepping up the stairs with a constant speed and its movement exhibits a small amount of zooming out, some degree of rotation of the head, and partial occlusion. The WT and color histogram-based algorithms (CHC and CHB) have mistakenly tracked a wrong object in frame nos. 139 and 160 (see Figures 6(c), 6(d), and 6(e)). As shown in Figure 6(f), after frame no. 124, both CHC and CHB miss the target off and on randomly because of the partial occlusion and zooming out effects. The WT algorithm loses the track after frame no. 110 and never finds the target correctly. In all the sequences, our proposed algorithm can successfully track the target object even in the presence

Figure 9: A man is tracked in the crowd in the metro in the presence of partial and full occlusions and zooming effects (all figures are the results of the proposed algorithm).

of object rotation and partial occlusion. The tracking results are available in separate video clips through the Internet (http://ce.sharif.edu/~khansari).

Figure 7 shows the result of tracking a person moving up the stairs and away from the camera in a metro station. Frame no. 635 was the reference frame (see Figure 7(a)), the size of the rectangle around the object was 25×15, and the search window size was 75×65 (± 25 pixels). Empirical parameters to find the direction and speed of the motion for updating the search window were set to $d = 2$ and $k = 8$. The object is moving up with a constant speed, and in a number of frames, it is fully occluded by the head of another person. The partial occlusion has started in frame no. 656 and turned into full occlusion in frame nos. 660–669. As shown in Figure 7(b), our algorithm successfully handles partial and full occlusions in this experiment. It has been observed that if during occlusion the object is still in the search area, immediately after reappearing, our algorithm can successfully detect it. The CHC, CHB, and WT fail to track the object because of occlusion, as shown in Figures 7(c), 7(d), and 7(e).

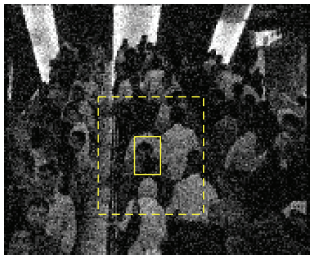
There are two reasons for occlusion handling of UWPT in Figure 7 and the following figures.

(1) The proposed FV is robust against the partial occlusion compared to the spatial space feature vectors such as those used in color histogram-based algorithms.

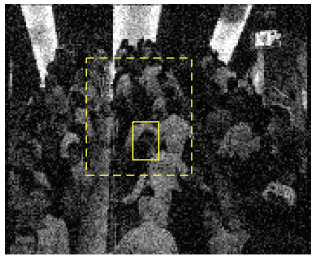
(2) Long-duration occlusion originates from the fact that the object of interest and the occluding object both move at the same direction. Since the algorithm uses activity analysis to find the motion and direction of the search window and hence updates the search window location, it can predict the location of the object after occlusion [29, 32]. Therefore, our updating mechanism ensures that the object lies within the search window in case of occlusion, and our robust FV allows for successful tracking afterward. 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 partially appears.

Reference: frame no. 97

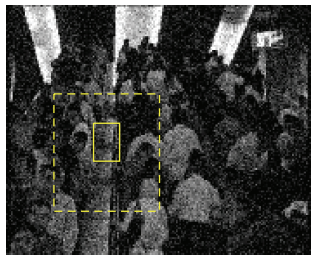
(a)



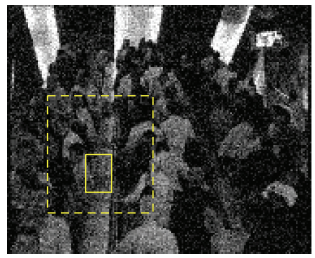
UWPT: frame no. 130



UWPT: frame no. 152

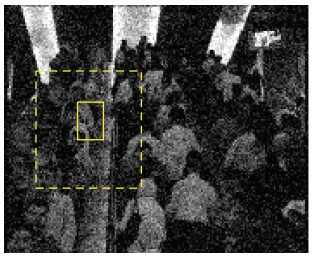


CHC: frame no. 130

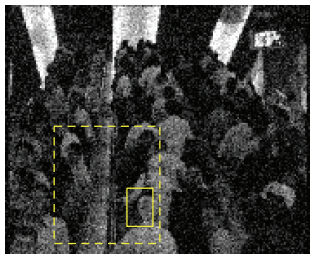


CHC: frame no. 152

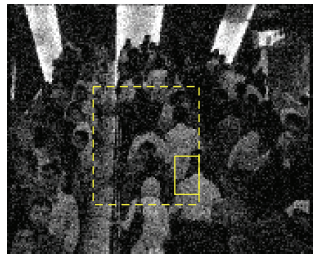
(b) (c)



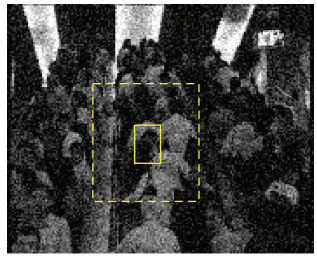
CHB: Frame no. 130



CHB: Frame no. 152

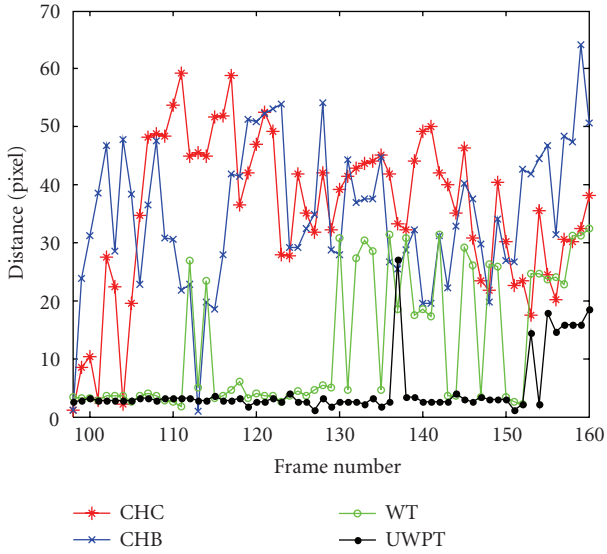


WT: frame no. 130



WT: frame no. 152

(d) (e)

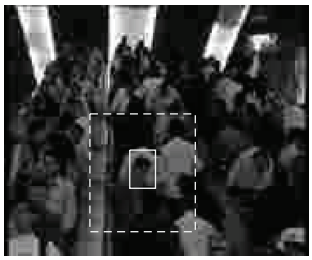


(f)

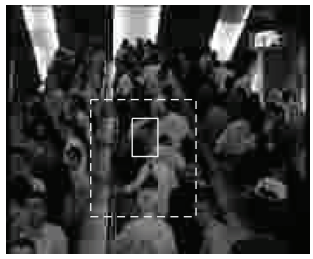
Figure 10: Tracking a man stepping up the stairs in presence of partial occlusions, zooming out, and additive Gaussian white noise (PSNR = 20 dB). (a) Reference frame, (b) UWPT, (c) CHC, (d) CHB, (e) WT, (f) objective evaluation: distance between the center of tracked bounding box and the expected center.

Reference: frame no. 97

(a)



UWPT: frame no. 108



UWPT: frame no. 156

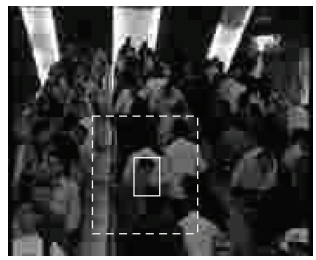
CHC: frame no. 108

CHC: frame no. 156

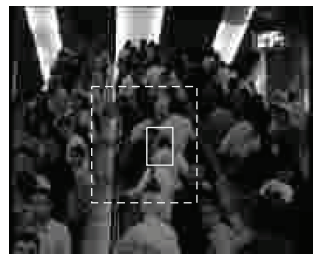
(b) (c)

CHB: Frame no. 108

CHB: Frame no. 156

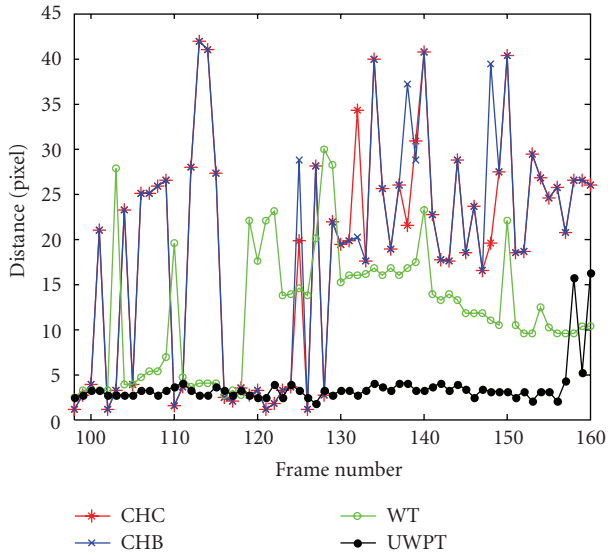


WT: frame no. 108



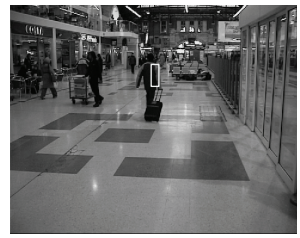
WT: frame no. 156

(d) (e)



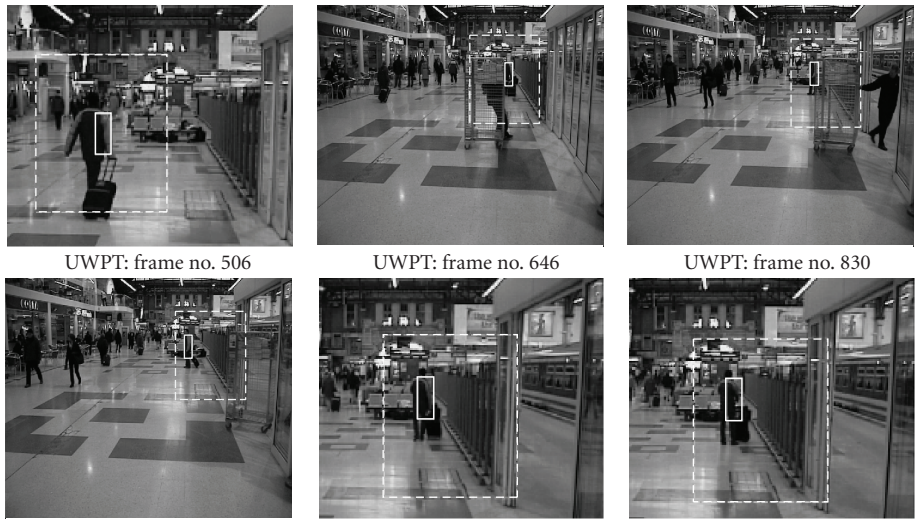
(f)

Figure 11: Tracking a man going up the stairs, in presence of partial occlusions and zooming out effects in presence of quantization distortion (bit rate of 0.22 bpp and PSNR of 31.95 dB). (a) Reference frame, (b) UWPT, (c) CHC, (d) CHB, (e) WT, (f) objective evaluation: distance between the center of tracked bounding box and the expected center.



Reference: frame no. 505

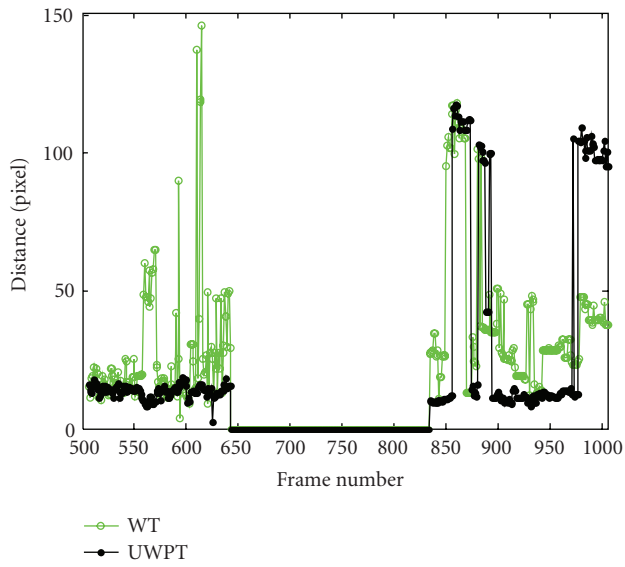
(a)



UWPT: frame no. 506 UWPT: frame no. 646 UWPT: frame no. 830

UWPT: frame no. 855 UWPT: frame no. 894 UWPT: frame no. 976

(b)



(c)

Figure 12: Tracking a man with a long-term occlusion in a larger number of frames (dataset S7 (camera 1) from IEEE PETS 2006 workshop (http://www.cvg.rdg.ac.uk/PETS2006)). (a) Reference frame, (b) UWPT, (c) objective evaluation: distance between the center of the tracked bounding box and the expected center for WT and UWPT.

Figure 8 shows the result of tracking where the crowd is getting off the train. Frame no. 35 is the reference frame with the bounding rectangle having the size of 42×22 and the search window having the size of 84×64. Empirical parameters to find the direction and speed of the motion for updating the search window were set to $d = 1$ and $k = 3$. The object is passing through the crowd and experiencing partial occlusions, and some zooming is also present in a number of frames. The partial occlusion begins from frame no. 62, and then in frame no. 64 it turns almost into complete occlusion that lasts for 10 frames. As demonstrated in Figure 8, our algorithm can successfully handle partial occlusions even in presence of zooming, due to the robustness of FVs and the adaptability of the search window. Although in few frames the algorithm cannot exactly find the object (e.g., frame nos. 85–89) due to the high movement of the object, lack of feature vector updating mechanism, and object blurring compared to the reference frame, in contrast to the histogram-based techniques it never loses object in the process of tracking.

Figure 9 shows the result of tracking a man where he moves inside the crowd in the presence of repeated partial and full occlusions and zooming. Frame no. 162 of the sequence was considered as the reference frame; the size of the rectangle around the object and the search window size were 37×22 and 111×96 (± 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, frame nos. 176–178. It is important to note that when both complete occlusion and zooming are present, it takes a number of frames before the object can be tracked successfully (the last row of Figure 9).

Figure 10 shows the result of tracking the sequence in Figure 6 in the presence of additive white Gaussian noise with a peak signal-to-noise ratio (PSNR) of 20 dB. This type of noise is very common with low-light video, especially in undergrounds. Again, the noisy reference frame is frame no. 97 (see Figure 10(a)). To highlight the resilience of the proposed algorithm against noise [30, 31], we have used a larger bounding box (23×16) and therefore a larger search area (62×69). Empirical parameters to find the direction and speed of the motion for updating the search window were set to $d = 1$ and $k = 3$ in Figure 10.

The presence of noise has degraded the performance of the color histogram-based algorithms tremendously without having any effect on the wavelet-based methods (see Figure 10(f)). However, the performance of WT is not consistent and is worse than the UWPT algorithm.

Figure 11 shows the impact of quantization noise on tracking objects, where the quantization distortion was set for a 0.22 bit per pixel compression with a PSNR of 31.95 dB. The directional motion search parameters were set to $d = 1$ and $k = 6$.

Again, the proposed algorithm outperforms the CHC, CHB, and WT algorithms because of the robustness of the proposed feature vector to object tracking in the noisy crowded environments. It should be noted that in the last few frames of this test clip, because of the larger bounding box and search area, almost full occlusion of the target object, and longer distance between the feature vectors of those frames and the feature vectors of the reference frame, the tracking results are not precise.

The proposed algorithm was also applied to longer-duration sequences with complex contents. The test sequence shown in Figure 12 is the dataset S7 (camera 1) from IEEE PETS 2006 workshop, which is one of the most difficult datasets with long duration. It contains a single person with a suitcase who loiters before leaving his luggage unattended. There is a long-term occlusion in the captured video from camera 1 (the scene has been captured using 4 different cameras from different locations).

The search window size should be large enough to handle the occlusion and encompass the target object after the occlusion. Therefore, for a bounding box of 56×21 pixels, the search window size was 212×177 (± 78 pixels). The directional motion search parameters were set to $d = 4$ and $k = 16$. Since the previous results clearly demonstrate that UWPT outperforms the other methods, we only objectively compare the performances of WT and UWPT for this test clip.

For the sake of clarity of the images, all of them (except for the reference frame) are zoomed in to better show the bounding box and search window status. Note that the size of the bounding box is kept unchanged during the tracking, and the object has a fast progressive zoom in effect. After 140 frames at the beginning of the clip, a full occlusion starts for a rather long period of 200 frames. However, our search window updating mechanism acts very well in this occlusion period and keeps the object in search area during and after the occlusion. Moreover, the robustness of our FV enables the proposed algorithm to find the object of interest after occlusion as depicted in Figure 12(b).

As shown in Figure 12(c), there is an initial 18-pixel offset between the gravity center of the bounding box and the person for both WT and UWPT methods. This error is mainly due to the large size of the bounding box. The WT tracker behaves inconsistently even at the starting frames, while UWPT can track the bounding box consistently, with the exception of few frames after the occlusion.

Because of the unusual movement of the target object, its exact position at the start of tracking and after occlusion will be too much different. In other words, due to the perspective transformation as well as deformation, some pixels may disappear or appear within the bounding box, and hence it is not expected that in this situation any kind of object tracking algorithm will work properly. In addition, after frame no. 976, the object reappears with a completely different view with respect to the original reference frame, and hence the tracking is inconsistent.

In [26], it has been shown that the core of the proposed algorithm can be implemented in real time for object tracking. However, the experimental results of this section have been implemented using MATLAB simulation environment. On a typical PC with 3 GHz Xeon processor, 1 GB of RAM, and Linux operating system, the processing rate of the

proposed algorithm is about 10–15 frames per second (FPS) depending on the size of the bounding box, search area, and specific I/O of the algorithm (on the screen, in JPEG or MATLAB file). As an example, the processing rate for the setting in Figure 5 is equal to 15 FPS. Therefore, by using an optimized C code, real-time performance can be achieved.

5. CONCLUSIONS AND FUTURE WORK

A new 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 have been proposed for object tracking in crowded scenes. Based on the properties of UWPT, existence of individual robust FVs for each pixel, and the adaptive search window, this method can tolerate complex object transformations including translation, small rotation, scaling, and partial or complete occlusions in a reasonable number of successive frames. Moreover, the algorithm is robust to different types of noise processes such as additive Gaussian and quantization noises, and it can be implemented in real time. We have also shown that the performance of UWPT is significantly better than that of the usual decimated wavelet transform, and color histogram-based methods for object tracking.

In the current algorithm, the FVs are kept unchanged during a tracking session. A memory-based FV updating mechanism combined with Kalman filtering for search area prediction can improve the performance of our algorithm in presence of abrupt zooming in/out or object scaling. Particle filters can also be integrated with the proposed FV generation to cope with the complex object movement and search window updating.

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.

ACKNOWLEDGMENTS

This research has been funded by the Advanced Information and Communication Technology Center (AICTC) of Sharif University of Technology. The authors also would like to acknowledge the helpful comments of the anonymous reviewers and Tehran Metro authorities for providing crowded scene video clips.

REFERENCES

[1] D. Xu, J.-N. Hwang, and J. Yu, "An accurate region based object tracking for video sequences," in *Proceedings of the 3rd IEEE Workshop on Multimedia Signal Processing (MMSP '99)*, pp. 271–276, Copenhagen, Denmark, September 1999.

[2] Ç. E. Erdem, A. M. Tekalp, and B. Sankur, "Video object tracking with feedback of performance measures," *IEEE Transactions on Circuits 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 *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 543–550, Kauai, Hawaii, USA, December 2001.

[4] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.

[5] C. Gu and M.-C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 572–584, 1998.

[6] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 539–546, 1998.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[8] T.-L. Liu and H.-T. Chen, "Real-time tracking using trust-region methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 397–402, 2004.

[9] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proceedings of the 7th European Conference on Computer Vision-Part I (ECCV '02)*, pp. 661–675, London, UK, May 2002.

[10] D. Xu, Y. Wang, and J. An, "Applying a new spatial color histogram in mean-shift based tracking algorithm," in *Proceedings of the Image and Vision Computing New Zealand (IVCNZ '05)*, University of Otago, Dunedin, New Zealand, November 2005.

[11] A. Jacquot, P. Sturm, and O. Ruch, "Adaptive tracking of non-rigid objects based on color histograms and automatic parameter selection," in *Proceedings of the IEEE Workshop on Motion and Video Computing (WACV/MOTION '05)*, vol. 2, pp. 103–109, Breckenridge, Colo, USA, January 2005.

[12] B. Deutsch, Ch. Gräßl, F. Bajramovic, and J. Denzler, "A comparative evaluation of template and histogram based 2D tracking algorithms," in *Proceedings of the 27th DAGM Symposium on Pattern Recognition*, vol. 3663 of *Lecture Notes in Computer Science*, pp. 269–276, Springer, Vienna, Austria, August-September 2005.

[13] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[14] Z. Zivkovic and B. Krose, "An EM-like algorithm for color-histogram-based object tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. 798–803, Washington, DC, USA, June-July 2004.

[15] H. Zhang, W. Gao, X. Chen, and D. Zhao, "Learning informative features for spatial histogram-based object detection," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '05)*, vol. 3, pp. 1806–1811, Montreal, Quebec, Canada, July-August 2005.

[16] S. T. Birchfield and S. Rangarajan, "Spatiograms versus histograms for region-based tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 2, pp. 1158–1163, San Diego, Calif, USA, June 2005.

[17] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 415–422, Kauai, Hawaii, USA, December 2001.

[18] B. Han, C. Yang, R. Duraiswami, and L. Davis, "Bayesian filtering and integral image for visual tracking," in *Proceedings of the Worshop on Image Analysis for Multimedia Interactive Services (WIAMIS '05)*, Montreux, Switzerland, April 2005.

[19] P. F. Gabriel, J. G. Verly, J. H. Piater, and A. Genon, "The state of the art in multiple object tracking under occlusion in video sequences," in *Proceedings of the Advanced Concepts for Intelligent Vision Systems (ACIVS '03)*, pp. 166–173, Ghent, Belgium, September 2003.

[20] C. He, Y. F. Zheng, and S. C. Ahalt, "Object tracking using the Gabor wavelet transform and the golden section algorithm," *IEEE Transactions on Multimedia*, vol. 4, no. 4, pp. 528–538, 2002.

[21] R. S. Feris, V. Krueger, and R. M. Cesar Jr., "A wavelet subspace method for real-time face tracking," *Real-Time Imaging*, vol. 10, no. 6, pp. 339–350, 2004.

[22] P. A. Brasnett, L. Mihaylova, N. Canagarajah, and D. Bull, "Particle filtering with multiple cues for object tracking in video sequences," in *Image and Video Communications and Processing*, vol. 5685 of *Proceedings of SPIE*, pp. 430–441, San Jose, Calif, USA, January 2005.

[23] F.-H. Cheng and Y.-L. Chen, "Real time multiple objects tracking and identification based on discrete wavelet transform," *Pattern Recognition*, vol. 39, no. 6, pp. 1126–1139, 2006.

[24] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 713–718, 1992.

[25] H. Guo, "Theory and applications of shift invariant, time-varying and undecimated wavelet transform," M.S. thesis, Rice University, Houston, Tex, USA, 1995.

[26] M. Amiri, H. R. Rabiee, F. Behazin, and M. Khansari, "A new wavelet domain block matching algorithm for real-time object tracking," in *Proceedings International Conference on Image Processing (ICIP '03)*, vol. 3, pp. 961–964, Barcelona, Spain, September 2003.

[27] M. Khansari, H. R. Rabiee, M. Asadi, M. Ghanbari, M. Nosrati, and M. Amiri, "A semi-automatic video object extraction algorithm based on joint transform and spatial domain features," in *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI '05)*, Riga, Latvia, June 2005.

[28] M. Khansari, H. R. Rabiee, M. Asadi, M. Ghanbari, M. Nosrati, and M. Amiri, "A shape tracking algorithm based on generated pixel features by undecimated wavelet packet," in *Proceeding of the 10th Annual Computer Society of Iran Computer Conference (CSICC '05)*, Tehran, Iran, February 2005.

[29] M. Khansari, H. R. Rabiee, M. Asadi, P. Khadem Hamedani, and M. Ghanbari, "Adaptive search window for object tracking in the crowds using undecimated wavelet packet features," in *Proceedings of the World Automation Congress (WAC '06)*, pp. 1–6, Budapest, Hungary, July 2006.

[30] M. Khansari, H. R. Rabiee, M. Asadi, M. Ghanbari, M. Nosrati, and M. Amiri, "A quantization noise robust object's shape prediction algorithm," in *Proceedings of the 13th European Signal Processing Conference (EUSIPCO '05)*, Antalya, Turkey, September 2005.

[31] M. Khansari, H. R. Rabiee, M. Asadi, M. Nosrati, M. Amiri, and M. Ghanbari, "Object shape prediction in noisy video based on undecimated wavelet packet features," in *Proceedings of the 12th International Multimedia Modelling Conference (MMM '06)*, Beijing, China, January 2006.

[32] 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.

[33] M. Vetterli and J. Kovačevic, *Wavelets and Subband Coding*, Prentice-Hall, Upper Saddle River, NJ, USA, 1st edition, 1995.

[34] I. Cohen, S. Raz, and D. Malah, "Orthonormal shift-invariant wavelet packet decomposition and representation," *Signal Processing*, vol. 57, no. 3, pp. 251–270, 1997.

[35] I. Daubechies, *Ten Lectures on Wavelets*, vol. 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*, SIAM, Philadelphia, Pa, USA, 1994.

[36] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 160–175, 1993.

[37] Ç. E. Erdem, B. Sankur, and A. M. Tekalp, "Performance measures for video object segmentation and tracking," *IEEE Transactions on Image Processing*, vol. 13, no. 7, pp. 937–951, 2004.

[38] P. Correia and F. Pereira, "Video object relevance metrics for overall segmentation quality evaluation," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 82195, 11 pages, 2006.

[39] R. N. Strickland and H. I. Hahn, "Wavelet transform methods for object detection and recovery," *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 724–735, 1997.