Transactions D:
Computer Science & Engineering and
Electrical Engineering
Vol. 16, No. 2, pp. 85–103
© Sharif University of Technology, December 2009

SCIENTIA
IRANICA

An Efficient Content-Based Video Coding Method for Distance Learning Applications

T. Lotfi[1], M. Bagheri[1], A.A. Darabi[1] and S. Kasaei[1,*]

Abstract. *This paper presents a novel method for cooperative educational dissemination systems. Taking into consideration the inherent characteristics of distance learning video streams (existence of a few slow moving objects in a classroom), we have proposed a novel content-based video coding method that is very efficient at low bitrate channels. On the encoding side, we have applied a background subtraction algorithm for motion segmentation using a novel statistical background modeling approach. At each frame, the moving objects are extrapolated with a rectangular model and tracked frame by frame (which forms the only data needed to be sent over the channel). On the decoding side, we have used a new error concealment algorithm (based on edge information of frames) to eliminate probable channel errors in the received data. Moreover, a new fuzzy scene modeling algorithm is proposed that adaptively computes the alpha blending coefficient (used in dynamic video mosaicing) and reconstructs the original video scene from partially overlapped frames. Our experiments show that the proposed coding system is very efficient in real-time video webcasting with approximately 24 fps for CIF formatted sequences (and at a minimum of 13 fps transmission for* 720×576 *frame sizes). Applying our proposed system has reduced the required bitrate of H.264 and MPEG-4 coding standards by about 2.5% to 8%, respectively, with almost the same, or even better, reconstructed video qualities.*

Keywords: *Content-based video coding; Distance learning; Motion segmentation; Mosaicing.*

INTRODUCTION

Advancements in communication technology are changing the way people around the world teach and learn, since the Internet, bulletin board systems, email and multimedia have already been incorporated into the daily lives of most college students. Applying these new technologies to instruction in technical communications introduces a great challenge for schools, lecturers and researchers in conventional classrooms, as well as in distance learning environments. One of the main aspects of distance learning systems is educational video webcasting. Because of existence of limited and low bandwidth connections to the Internet in developing countries (mostly 56 Kb dialup modem connections), introducing new video coding methods for real-time video compression forms a very important field of research.

A wide variety of methods has been reported to compress video streams, but a few have introduced context dependent methods for video coding. In this paper, we have proposed a new video encoder/decoder system adapted for real-time educational video dissemination. The system is based on the segmentation, tracking and modeling of moving objects on the encoder side, and a new method for video mosaicing and error concealment on the decoder side.

Conducted experiments show that the proposed system is very efficient in distance learning video webcasting.

The rest of this paper is organized as follows. First, a short literature review is given. Then, different steps of the proposed system are introduced. Following that experimental results are discussed and finally, the paper is concluded.

LITERATURE REVIEW

The main purpose of distance learning is to overcome the barriers of geographical separation between teachers and learners. According to the usual distance learning systems [1,2], it is well-known that a good distance learning system should provide stu-

[1] *Department of Computer Engineering, Sharif University of Technology, Tehran, P.O. Box 11155-9517, Iran.*

[*] *Corresponding author. E-mail: skasaei@sharif.edu.*

Received 14 September 2008; received in revised form 17 February 2009; accepted 28 June 2009

dents with a visual classroom environment; that is the students are able to see the teachers' motions and hear his/her voice, while they can also get the assistant data information used by the teacher during his/her teaching (i.e., used PowerPoint document). Considering these requirements, two main types of information should be provided in a distance learning system. One is the video/audio stream (recording of the whole teaching process) and the other is the data information.

One of the main aspects of distance learning systems is educational video webcasting. Because of the existence of limited and low bandwidth connections to the Internet (mostly by 56 Kb dialup modem connections), introducing new video coding methods for real-time video compression is still a crucial problem. In order to reduce the required bitrate, in this paper, on the encoder side, motion segmentation, moving object tracking and the removal of shadow cast from moving objects are applied at the preprocessing stages and on the decoder side, the error concealment and video mosaicing are applied at the postprocessing stages (see Figure 1). The related literature for each process is explained next.

Motion Segmentation and Moving Object Tracking

In recent years, motion segmentation and the tracking of moving objects have been widely used in various applications, including video compression and video surveillance. Video surveillance applications use motion segmentation and the tracking of objects of interest using multiple cameras, in order to study about movements occurred in a scene [3], whereas video compression applications use these in order to extract foreground regions from the background to compress each object via different bitrate, according to the limited network bandwidths [4].

Object segmentation methods are generally divided into feature-based and motion-based groups. The motion-based segmentation methods are divided into temporal subtraction [5], optical flow [6], background subtraction [7] and hybrid [8,9] methods. In background subtraction methods, each frame is simply subtracted from the background model to segment the moving objects. Therefore, it needs to consider a model for the background region and update it in consequent frames. The model can be obtained by temporal averaging [7], adaptive Gaussian [10] or statistic [3] methods. Obtaining the model, the moving object can be tracked, frame by frame, using a tracking method, such as Kalman filtering [11], particle filtering [12] or mean shift tracking [13].

In this paper, we propose a novel temporal motion buffering method as a background modeling technique.

Removing Shadow Cast from Moving Objects

The detection and tracking of moving objects is the core of many applications dealing with image sequences. One of the most important challenges of these applications is identifying the moving object and its cast shadow. Shadows cause serious problems when segmenting and extracting moving objects (due to misclassification of shadow regions as foreground areas). Shadows can cause object merging, object shape distortion and even object missing. Also, in object-based video compression applications, it can increase the required bitrate. The difficulties associated with shadow detection arise since shadows and objects share two important visual features. First, shadow regions are extracted as foreground areas, since they typically differ significantly from the background. Second, shadows have the same motion as the objects

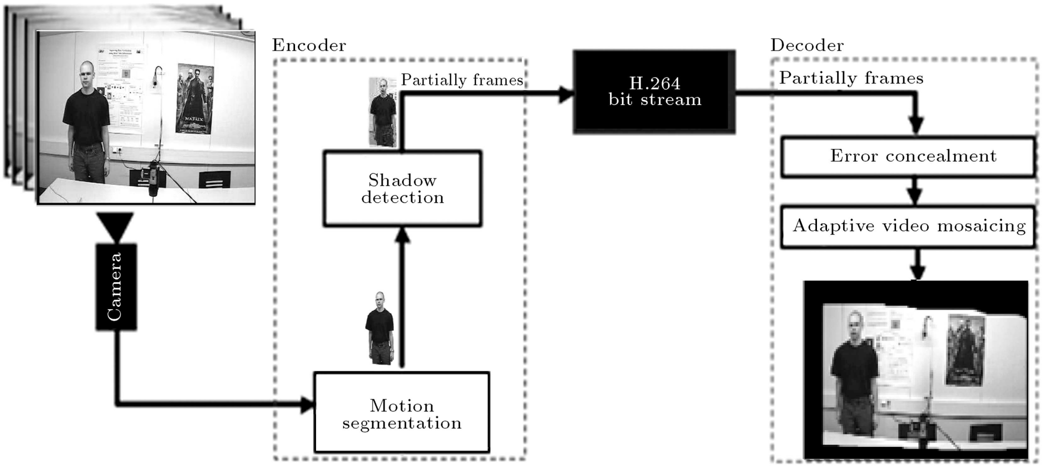


Figure 1. Block diagram of proposed coding system.

casting them. As such, shadow identification is critical both for still images and image sequences [14].

Most of the reported approaches on shadow removal have taken into account the shadow model described in [15,16]. Non-deterministic approaches consider whether the decision process introduces and exploits uncertainty. Deterministic approaches use an on/off decision process, whereas statistical approaches use probabilistic functions to describe the class membership. Introducing uncertainty to the class membership assignment can reduce noise sensitivity by relaxing ill-posed constraints. In statistical methods, the parameter selection is a critical issue. The studies reported in [17,18] are examples of parametric and non-parametric approaches, respectively.

Within the deterministic class (see [19]), another sub-classification can be based on whether the on/off decision can be supported by model-based knowledge or not. Choosing a model-based approach achieves undoubtedly the best results, but most of the time it is too complex and time consuming compared to non model-based approaches. Moreover, the number and complexity of models increase rapidly if the aim is to deal with complex and cluttered environments with different lighting conditions, object classes and perspective views.

To deal with the above mentioned problems of segmentation and tracking, in this paper, we have proposed a new non-parametric statistical shadow removal method, based on the lighting direction. Our method computes the dominant direction of lighting and segments the shadow accordingly.

Error Concealment

Modern spatial error concealment techniques can be classified into four main methods: exemplar-based, tensor voting-based, stochastic-based and interpolation-based. These are briefly explained next.

I. Exemplar-based methods assume that each block of an input image can be reconstructed using other blocks of the same image. In [20], a simple method of this group has been proposed, which conceals the whole erroneous block with another block of that image after adjusting its intensity. An advanced version of this method was proposed in [21], which divides the damaged block into small blocks and searches for each subblock, independently. The major weakness of this method is its content independent block division. Other methods of this group were proposed in [22-25].

II. Tensor voting-based methods calculate the tensors around the damaged block and approximate the tensor of lost pixels by voting [26]. Although these methods produce impressive results, their real-time implementation is impossible due to their high complexity.

III. Stochastic-based methods use statistical characteristics of pixels around the lost block to approximate its pixel values. They often assume a Markov model to simplify their calculations and employ the Bayesian formula to describe the problem [27]. These methods usually have two major weaknesses: They yield blur results because of the simplifying assumptions, and high complexity due to the required iterations.

IV. Interpolation-based methods attempt to interpolate pixel values in the lost block using border pixels of that block. In [28], a very fast method was proposed which conceals the erroneous block with the aim of directional interpolation. In [29], an advanced version of that method has been proposed, which can cope with more difficult situations. Their major limitation is weakness in texture reconstruction.

The aim of our used concealment method is to overcome the shortcomings of the exemplar-based and interpolation-based methods (namely, content independent block shape and poor texture reconstruction) and to design a fast and accurate error concealment method. The details are given in the following section.

Video Mosaicing

Mosaicing is the process of reconstructing a wide scene model by aligning and properly blending together partially overlapped frames acquired by a video sequence captured from a wide scene. According to this definition, we can reconstruct any wide scene from related mosaic frames if the transformation parameters of the video frames and the coordinates of the global frame are known. Wide scene reconstruction from a mosaic can be very useful in a wide variety of applications; such as video surveillance and object-based video compression and indexing [30].

There are two main types of video mosaicing method: static and dynamic. The static mosaicing method operates in the batch mode by aligning all images in a fixed coordinate system. However, it cannot completely depict the dynamic aspects of a video sequence. The dynamic mosaicing method can overcome this problem. This is because the content of each new mosaic scene model is updated with the most current information obtained from the most recent frame [31]. For an excellent review of various types of mosaic representation and their applications refer to [32].

In our proposed scene modeling method, the transformation parameters of each video frame are coded and sent to the decoder in the header part of each video frame. We have also used a new smooth blending method to combine registered frames, such that no obstructive boundaries exist around overlapped regions, and thus we have created a mosaic scene model that exhibits with a very low distortion from the original frames. The details will be given later.

PROPOSED METHOD

In this section, we first explain the general concepts of a distance learning video coding system and then introduce the proposed algorithms on the encoding and decoding sides of the coding system. Here, we have proposed a novel method for motion segmentation and a new algorithm for shadow removal on the encoder side. On the decoder side, we have used a robust error concealment method for error correction in I frames and an adaptive method for scene modeling based on video mosaicing. The general block diagram of our proposed video coding system for distance learning videos is illustrated in Figure 1. The main idea behind our proposed coding system is decomposing the frame-based sequences to partial frames, encoding them and then combining the received partial frames to form a scene-based presentation of the original video, in order to reduce the redundant data in the transmission of video sequences. The proposed stages on the encoder and decoder sides are explained in detail next.

A. Encoder Side

Considering the inherent characteristics of distance learning video streams (namely, a few moving objects existing in the scene and objects having slow motion), here, we have proposed a novel motion segmentation and shadow removal algorithm for distance learning video encoding that is very efficient in low bitrate channels.

On the encoder side, first, a new temporal motion buffering method which uses a hierarchical segmentation technique is introduced to extract moving objects in the scene. Then, a new shadow removal algorithm, which uses the edge information of moving objects is introduced to remove shadow casts from moving objects. These are explained next.

Motion Segmentation

A common approach for motion segmentation is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from the background model. There are many challenges in developing a proper background subtraction algorithm. The requirements include:

I. Robustness against illumination changes;

II. Avoidance of detecting non-stationary background regions and shadows cast by moving objects;

III. Fast reaction to changes in background areas and, finally;

IV. Performance in real-time [33].

Figure 2 shows the general diagram of our proposed motion segmentation model.

Regions of Interest in Foreground Model

In this process, the wavelet transform is used as a powerful tool for efficiently finding regions that contain moving objects. It is known that the averaged subimage is much more robust to noise, but misses some details, while the detailed subimages are vulnerable in the presence of noise, but they contain the required details.

The proposed algorithm fuses these two properties to obtain better results. In our approach, as shown in Figure 3, at first, the Haar wavelet is used to decompose the input frame into averaged and detailed subimages. These subimages are then subtracted

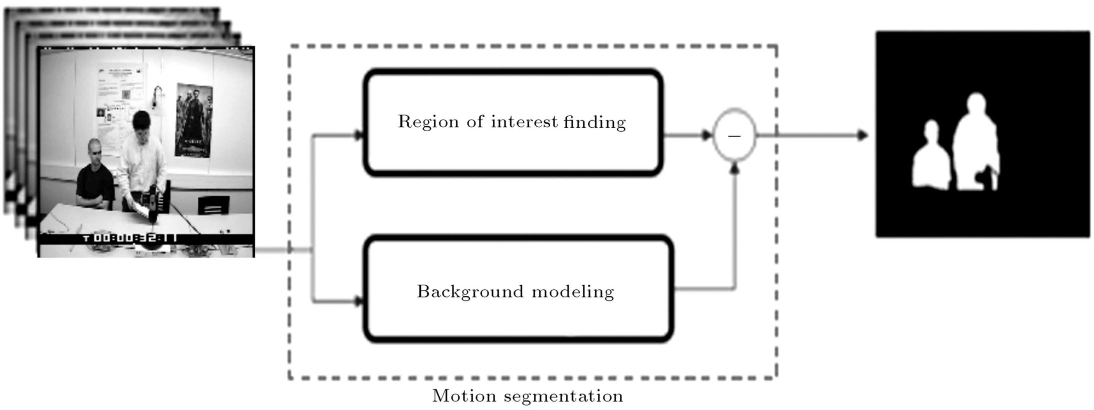


Figure 2. Proposed motion segmentation model.

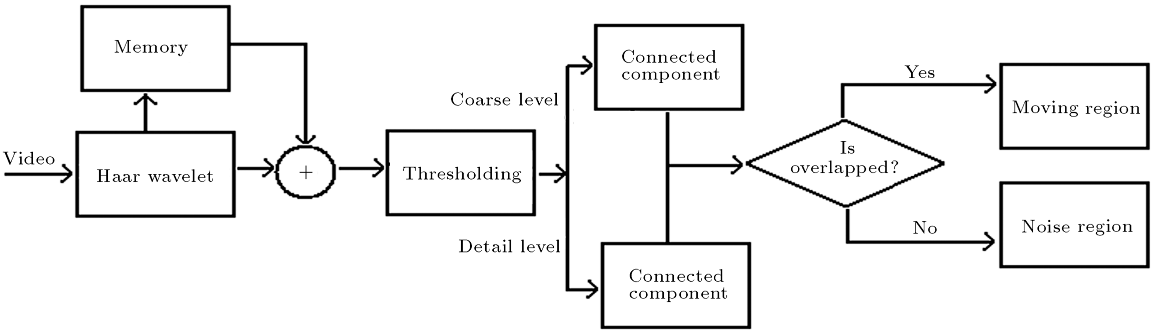


Figure 3. Flowchart of proposed region of interest finding.

from their counterparts in the prior frame. Next, by applying hard thresholding, the subimages are mapped into binary images. Having binary images, their connected components are obtained. Here, the connected components are rectangle-wise. By this assumption, a new true point is recognized if it appears near the previous component rectangles, and the rectangle is expanded to cover that point. Otherwise, a new component rectangle is established. To decrease the effect of noise, the components in detailed subimages are taken as moving regions if they have some overlaps with the components in the averaged subimage.

Figure 4 shows an example of how these subimages are fused to form the Regions Of Interest (ROI). This process improves the performance of the system, mainly because:

I The subsequent processes are only applied to the obtained ROI, and thus the complexity of the system is effectively reduced;

II Each foreground region can be processed independently using its related designed process and;

III. The powerful denoising property of the wavelet transform is employed to enhance the resulted foreground regions.

Now, we can find the ROI that are moving in the scene and extrapolate with rectangular bounding boxes. Each rectangular bounding box creates a new frame that contains moving objects. For example, in distance learning videos, the teacher is extrapolated with a rectangular frame as a moving object.

Background Modeling

The basic concept in this method is that any pixel at each frame cannot remain as the foreground for L subsequent frames or more. According to this fact, after L frames, if the value of a pixel changes more than a determined threshold, then, the model of the background in that pixel becomes equal to

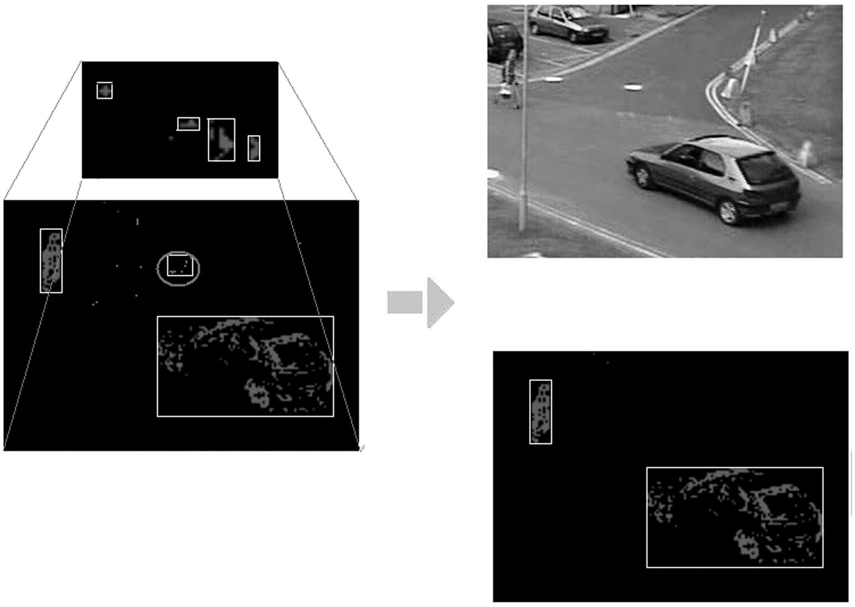


Figure 4. Finding regions of interest (noisy ellipse is removed).

the corresponding pixel value in the current frame. Otherwise, a weighted average of pixel values in the current frame and the previous model is obtained as the background model. The background model is updated using:

$$M_b(x,y,t) = \begin{cases} I(x,y,t) \text{ if } |I(x,y,t)-I(x,y,t-\Delta t)| > \text{threashold} \\ \alpha I(x,y,t) + (1-\alpha)M_b(x,y,t-1) \text{ else} \end{cases} \tag{1}$$

where $M_b(x,y,t)$ and $I(x,y,t)$ denote pixel values in the model and current frame, respectively. The value of α is chosen, based on the amount of noise and light change in the frames. In noisy environments (such as outdoor environments), using high values (near to one) for α is more appropriate. In this method, the buffer size, L, is calculated as:

$$L = \Delta t \times \text{frame rate}, \tag{2}$$

where Δt represents the maximum interval at which the largest object in the video frame moves W pixels, and W is the width of the largest object in the scene (see Figure 5). In this model, no pixel remains as the foreground more than Δt times, because this is the maximum interval at which the moving object overlaps with one pixel.

In order to improve the process, we update the model in regions with no motion. In this updating mechanism, the parameter values of the model become equal to the values of corresponding pixels at each frame.

Using the above mentioned temporal motion buffering method, we have obtained a robust and fast model of the background regions. Then, by using a simple background subtraction process, the foreground model at each frame is achieved.

In this method, using a minimum threshold value for the size of the moving object frames, and only considering the objects with a larger size for transmission can be very efficient. As such, only these frames along with some header information (such as the position of each frame in the scene and a foreground mask created from background subtraction method) are sent to the receiver.

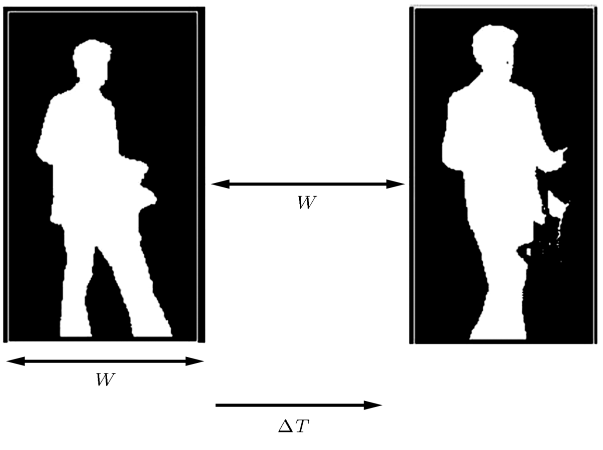


Figure 5. Calculation of Δt parameter in video frames.

Because moving objects might have different shapes and motions in time, finding a fixed size for bounding boxes is inefficient. One solution to this problem is using 16 × 16 Basic Blocks (BB). Each frame is constructed with a minimum possible number of BB. The main reason for using 16 × 16 BB is its compatibility with a H.264 standard BB size. In our implementation, the header information is coded in the header part of the transfer block of H.264 standard. We actually decompose our frames to BB with header information and then transmit it under a H.264 standard.

Shadow Cast Detection Using Moving Objects

The basic idea of our proposed shadow detection method is using the concept that, since temporal difference methodology approaches are less sensitive to shadows, they can exclusively show the borders of moving objects. On the other hand, background modeling approaches determine pixels inside the object as the moving part, but they are much more sensitive to shadows and captured noise. In this section, we have used borders of segments and a temporal difference of frames to remove shadows. To reduce the effects of shadows, it is beneficial to separate the information of chrominance from the luminance of pixels. To do so, we have used the normalized *rgb* color format by:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}. \tag{3}$$

Knowing the fact that shadows are less sensitive to chromaticity, the new coordinate is immune to shadows. For temporal differentiation $(r,\ g)$ is used as the value of each pixel at each frame.

The proposed method uses the intersections of segment boundaries (obtained in the previous section) and a temporal difference image to find sections having overlaps with the boundaries of real objects. To have segment boundaries, the Canny operator is used to retain the continuity of edges. Then, the AND operator is used to obtain the intersection of segment borders and the temporal difference image by:

$$\text{Int } (S,D) = \text{Canny } (S) \ \& \ D, \tag{4}$$

where S denotes the borders of the segmented image and D denotes the temporal subtraction of consequent frames. The aim is to specify the segments that are relevant to each pixel of Int (S, D) in Equation 4.

Obtaining the intersection of segment borders (S) and the temporal subtraction image (D), the point members of both images (Int (S, D)) must be assigned to proper segments. To fade shadows, the algorithm assigns points to segments in such a way that no points are assigned to segments related to shadows. As such, the algorithm assigns edges in the opposite direction to the main light direction. Consequently, the main light direction must be found first. Considering one light source, the algorithm is discussed below:

1. For each of the 4 main directions, named L, do the following steps;
2. Mark border pixels (obtained from temporal subtraction images) on the segmented image;
3. Move the segmented image toward the L direction and find the number of background to foreground switching according to the marked pixels, and name it N_l;
4. Move the segmented image toward the L direction and find the number of foreground to background switching according to the marked pixels, and name it N_r;
5. Compare N_l with Nr and find the difference, N_d;
6. Determine the L with the largest N_d as the main light direction.

Finding the main light direction, the following algorithm employs the main light direction (D_l), the intersection image (Int) and the segment image (I_s) to assign points properly, in order to vanish shadow segments. For each line, l, in the D_l direction, apply the following step:

1. Call the direction of lighting α and the counter of it β (i.e. if the direction of the light is from right to left, α is on the right side and β is on the left side).
2. Continue scanning Int from α to β to reach the first point at Int. Assign this point to the segment in the β neighbor direction of it (i.e., if the direction of the light is from right to left, assign that point to the segment at its left).
3. After finding the first point in the intersection, assign that point in the α neighbor direction.
4. Consider segments with no points assigned to them as shadow segments.

This procedure is illustrated in Figure 5. The reason for using the direction of light is that, when points, except the first point assigned in the opposite direction to the light, the segments of shadows have no points in their region and they can be identified. To find the main direction of light, the following process is suggested. For each D direction, take the following steps.

1. In the previously obtained foreground mask, scan lines in the D direction and count transitions from background to foreground that correspond to the temporal difference image and name it N_l.
2. In the previously obtained foreground mask, scan lines in the D direction and count transitions from foreground to background that correspond to the temporal difference image and name it N_r.
3. Name the difference of N_r and N_l as D_d.
4. The main direction of light is the direction for which D_d is maximized.

The main idea of this method is that when the direction in which the difference of pixel transitions from background to foreground placed on real borders (temporal difference of image) and foreground to background placed on real borders is maximized, it is the main direction of light. Figure 6 depicts this idea for a sample image.

B. Decoder Side

On the decoder side, two new methods are performed. The proposed error concealment method (which uses

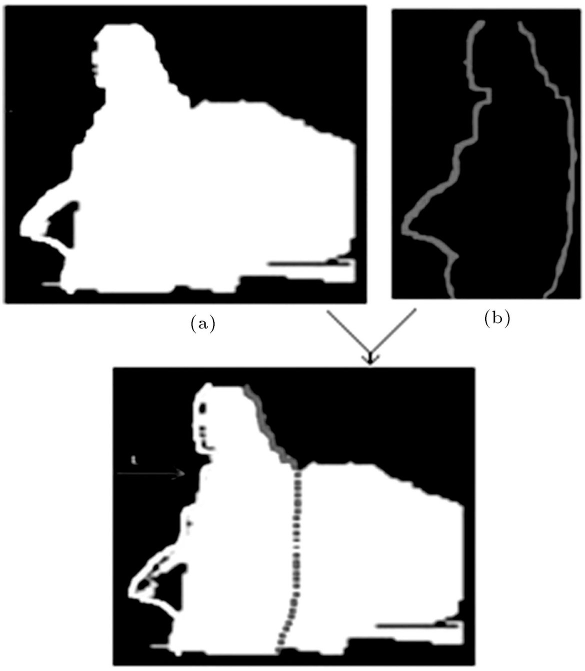


Figure 6. Finding the main light direction. (a) Segmented image; (b) Borders obtained from temporal subtraction; (c) Marked pixels on segmented image (dashed pixels do not participate in foreground-background switching, because shadow pixels after them are taken into account as foreground pixels).

the homogeneous preconception of special adjacent pixels to conceal every lost block on the channel) and the proposed fuzzy scene modeling, which uses the motion parameters of each partial frame (proportional scale and speed of moving objects) as inputs to a fuzzy approach to adaptively determine the value of alpha blending. These are explained next.

Error Concealment

In this section, the error concealment method is described. It consists of four main blocks: image cropping, image segmentation, segmentation matching and inpainting (see Figure 7). The error concealment algorithm is described as follows [34]:

1. Crop the image around the erroneous block; segment the cropped region.
2. Connect the broken segments using the borders of each segment by the following procedure (see Figure 8):
   a. Calculate the intersections of the segment borders with erroneous block sides as entry points.
   b. Select a straight line that satisfies the following properties:
      i. It connects the entry points of neighbor segments to the lost block.
      ii. The difference of their entrance angles are about 180 degrees.
   c. Consider different regions that have been created by estimating these lines as the inputs of the inpainting process.
3. Inpaint each segment individually, by the following procedure (see Figure 9):

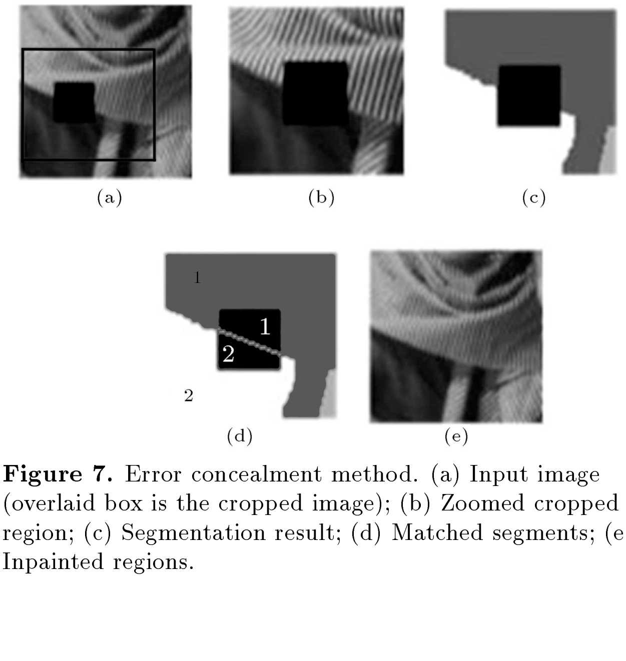


Figure 7. Error concealment method. (a) Input image (overlaid box is the cropped image); (b) Zoomed cropped region; (c) Segmentation result; (d) Matched segments; (e) Inpainted regions.

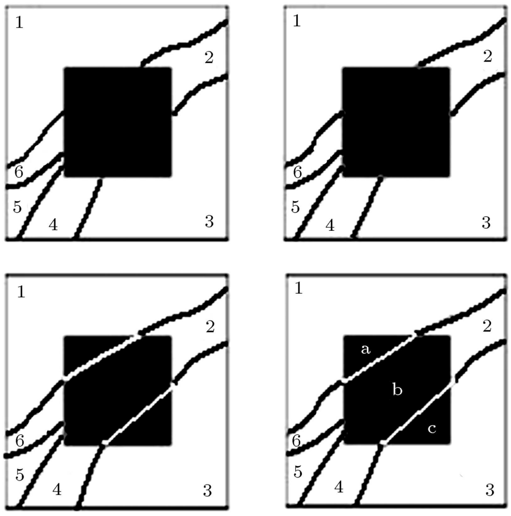


Figure 8. Segment matching process of error concealment method. (a) Segmented regions (numbered 1 to 6 according to borders); (b) Labeled intersection points of segment borders to the missed block; (c) Assigned straight lines; (d) Allocated segments of divided missed block.

a) Create a border mask around the input mask. This border mask consists of reliable information of the original image. (The width of this border mask can be adjusted by the user in the initial system setup.)

b) Define the search region. In our proposed method, since we know the segments that surround the input mask, these segments are used as the search region.

c) Obtain the best match that minimizes the absolute error between current and remote borders.

d) Adjust the mean of the remote region and replace the current region by the remote region of the remote mask.

Fuzzy Scene Modeling

In this section, the subsequent steps of the proposed fuzzy scene modeling method are explained in detail. Figure 10 shows the reconstruction of the scene on the decoder side.

Region MO and B represent the current partial frame that updates the mosaic scene, and region A represents the reconstructed region from prior partial frames. The next section introduces our proposed method, to adaptively blend region B of the current frame in the reconstructed scene.

Object Blending Criteria

There are many challenges in developing a proper blending algorithm to strictly blend partial frames and create a reconstructed mosaic from the scene. The

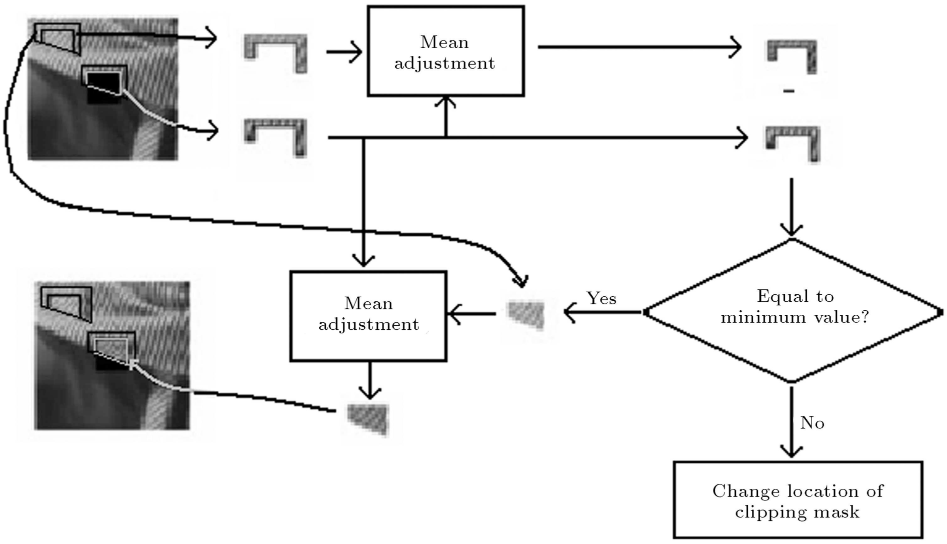


Figure 9. Inpaint process of exemplar-based error concealment method. Top: Missed part values of the segment. Bottom: Mean adjustment method (compare the area around the missed part and the selected part).

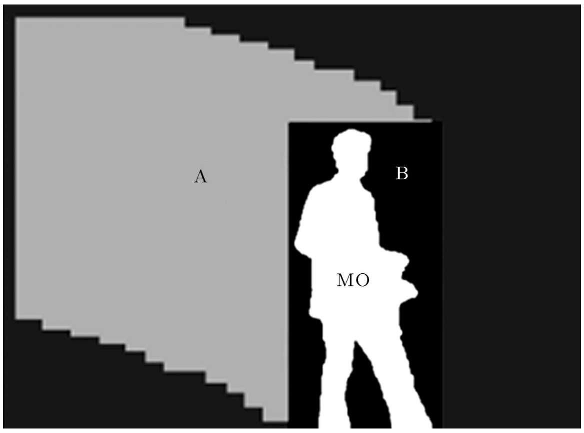


Figure 10. Scene reconstruction on decoder side.

main criteria for developing a robust and efficient blending algorithm include:

I. Minimization of temporal variations in the background area. Note that the background information contained in the current compensated image $\tilde{I}_t$ can be different from the existing mosaic image. As a consequence, the mosaic images, M_t and M_{t-1} can be significantly modified in the area updated by $\tilde{I}_t$. If these changes are too strong, they will degrade the temporal quality of the mosaic sequence by artificially introducing strong temporal variations and artificial boundaries at the limits of the updated and non-updated areas.

II. Minimization of temporal delay. Note that a mosaic image is not temporally homogeneous, since it is calculated using several images. As a consequence, a temporal delay can be associated to each pixel. For many applications it will be important to have the temporal mosaic image as close as possible to the current image.

III. Performing in real-time.

In order to meet the above mentioned criteria, here we have proposed a fuzzy approach to alpha blending. In our approach, the value of the alpha blending coefficient is obtained adaptively, with respect to the speed and scale of the moving objects.

Moving Object Blending

We have selected some related motion parameters, based on blending criteria. From the experiments, we have concluded that the speed of moving objects in a sequence and the scales of each moving object have the most effect on the blending criteria. Therefore, we have defined the speed and scale parameters of each moving object as a basis for our adaptive alpha coefficient calculation.

In order to apply warping parameters to moving objects, we have to use a rectangular model. As mentioned previously, applying the warping parameters on the rectangular model is fairly fast and suitable. In these models, some pixels which do not belong to the moving object regions and are extrapolated with model boundaries play an important role in the blending process (see Figure 9b).

In our proposed method, to create the dynamic mosaic, M_t, from the scene, we use a weighted combination of these pixel values $\tilde{I}_t$ (black region in Figure 9b) and values of the corresponding pixels in the static mosaic background, M_{t-1}, using:

$$M_t = (1 - \alpha)M_{t-1} + \alpha\tilde{I}_t. \qquad (5)$$

The weighting coefficients, α, vary as a fuzzy function of the related motion parameters of moving objects.

Moving Object Speed

Based on our experiments, there is a direct relation between the moving object speed and the temporal variation of background information in the current image $\tilde{I}_t$.

In fact, when the speed of a moving object in a scene increases, the temporal variation of background information also increases. Therefore, the alpha coefficient in blending must be decreased (the first criteria).

According to this fact, we introduce a method to calculate the speed of the moving object model in image I_t. In our method, we suppose that the maximum temporal movement of each moving object in image I_t is R, which is equal to the diameter of the rectangular model. Then, we can define the speed ratio ($0 < V_r < 1$) for each moving object model as the object model movement in proportion to the maximum model movement, R, during two consequent images in the sequence. This is illustrated in Figure 11.

Calculation of the speed ratio, V_r, for the moving object models is done by:

$$R = \sqrt{X^2 + Y^2}, \qquad V_r = \frac{v}{R},$$

$$v = \sqrt{x^2 + y^2}. \tag{6}$$

Now, we define some linguistic quantification about the speed ratio as the fuzzy sets and then assign our obtained speed ratio to these fuzzy sets with their appropriate membership values. We use three linguistic quantifications, including "low speed", "normal speed" and "high speed" motions in our implementations and design an appropriate Gaussian membership function based on our empirical results (see Figure 12.)

Moving Object Scale

The scale of moving objects in the input image sequence is one of the most effective parameters in the blending process. Based on our experiments, there is an inverse relation between the amount of background information in the object model and its alpha blending value. In fact, when we have little background information in our model, we give them the most priority to display (bigger alpha blending value). In this method, we define the scale of the moving object with an inverse relation to the amount of background information in our moving object rectangular model. Therefore, the scale parameter of a moving object is defined by:

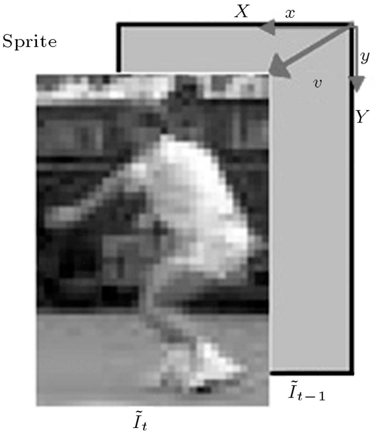


Figure 11. Moving object speed calculation.

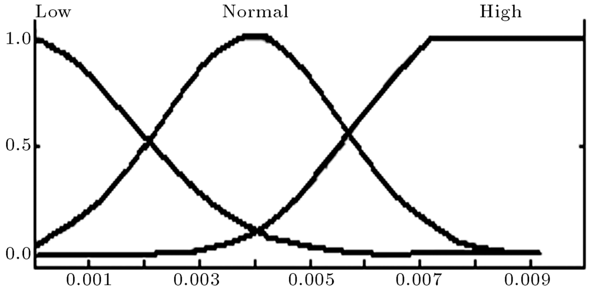


Figure 12. Speed membership function.

$$\text{Scale} = \frac{\text{Object Area}}{\text{Model Area}}. \tag{7}$$

The scale parameter is in range [0-1]. Now, we can define some other fuzzy sets and fuzzificate our scale parameters. We introduce our fuzzy sets with a Gaussian membership function and assign a linguistic quantification to each fuzzy set, such as "big scales", "medium scales" and "small scales". Figure 13 illustrates the used membership function for scale parameters.

Adaptive Alpha Blending Coefficients

In order to adaptively obtain the most appropriate value for the alpha blending coefficient, we complete our fuzzy inference system with fuzzification in our outputs (alpha blending coefficients) and design an adequate rule based on our knowledge.

There are many alternatives in setting up our output membership function. Our experimental results show that using four fuzzy sets with a Gaussian membership function is flexible enough to adapt to the

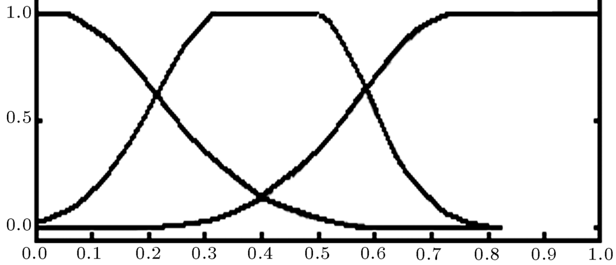


Figure 13. Scale membership function.

variations of motion parameters. We use four linguistic quantifications, including "low", "normal", "high" and "very high" to show each fuzzy set. These are shown in Figure 14.

One of the most important steps in our method is the configuration of the rule base. Our experimental results showed that the output values (alpha blending coefficients) are more sensitive to the scale of moving objects. Thus, we implemented this property in rule 6, 7 in our system (see Figure 15). Also, based on our empirical results, we concluded that the best behavior for the alpha blending coefficient generator is shown in Figure 16.

EXPERIMENTAL RESULTS

In order to evaluate the performance of the different steps of the proposed method, we have used three different video sequences captured from a classroom and also the PETS 2002 indoor and outdoor standard videos [35].

In all experiments, Videos 1-3, Video 4 and Video 5 denote the average results of the algorithm applied to our captured videos; outdoor PETS and indoor PETS standard videos, respectively.

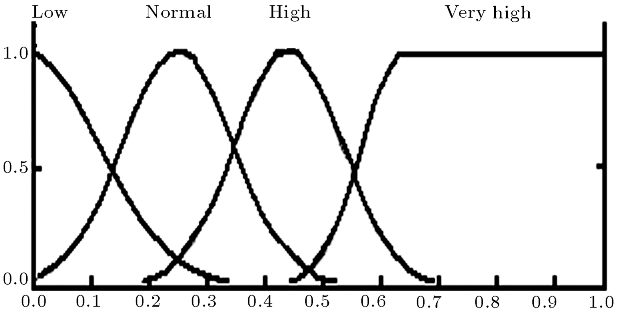


Figure 14. Alpha coefficient membership function.

If (Speed=Low) & (Scale = Small) ⇒ (Alpha = High)

If (Speed=Low) & (Scale = Medium) ⇒ (Alpha = High)

If (Speed=Low) & (Scale = Big) ⇒ (Alpha = Very high)

If (Speed=Normal) & (Scale = Small) ⇒ (Alpha = Medium)

If (Speed=Normal) & (Scale = Medium) ⇒ (Alpha = Medium)

If (Speed=Normal) & (Scale = Big) ⇒ (Alpha = High)

If (Speed=High) & (Scale = Small) ⇒ (Alpha = Low)

If (Speed=High) & (Scale = Medium) ⇒ (Alpha = Medium)

If (Speed=High) & (Scale = Big) ⇒ (Alpha = Medium)

Figure 15. Assigned fuzzy rule bases.

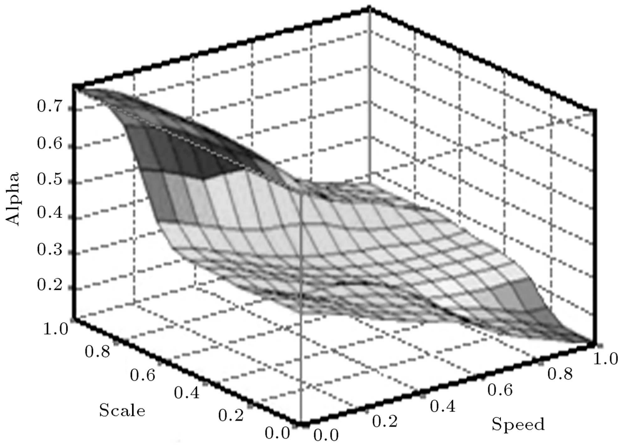


Figure 16. Fuzzy surface of the system.

Results of Background Modeling with Proposed Temporal Motion Buffering

In order to evaluate the performance of our proposed method in background modeling and temporal motion buffering, we used three alternative methods that are mostly related to the proposed methods. These include temporal averaging, gradient Gaussian and a mixture of the Gaussian model. The obtained results are shown in Figures 17 and 18.

To compare the segmentation quality of different methods, we have manually labeled the ground-truth data on the video sequences. The ground-truth data refer to a number of 2-D polygons in each video frame, which approximate the contour of motion regions. The labeled polygons do not include the shadow regions, since our method removes object shadows as well. For each video sequence, more than 60 frames (25% of the whole sequence) are labeled. Based on the ground-truth data, the quality of each method is calculated

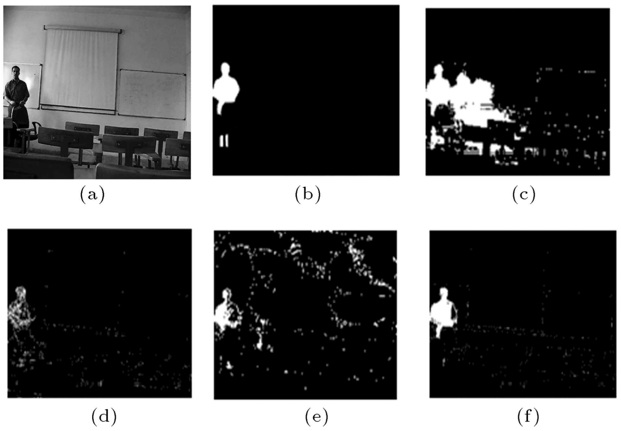


Figure 17. Results of different background models. (a) Original frame; (b) Ground truth frame; (c) Temporal averaging; (d) Gradient Gaussian; (e) Mixture of Gaussian; (f) Proposed temporal motion buffering.

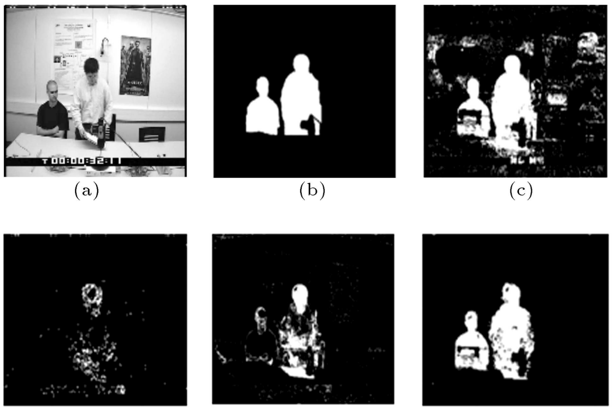


Figure 18. Results of different background models. (a) Original frame; (b) Ground truth frame; (c) Temporal averaging; (d) Gradient Gaussian; (e) Mixture of Gaussian; (f) Proposed temporal motion buffering.

using:

$$\text{Error} = \frac{\sum\limits_{\forall x,y} XOR(I_{\text{ideal}}(x,y).I_{\text{motion}}(x,y))}{\sum\limits_{\forall x,y} 1}. \qquad (8)$$

Table 1 lists the percentage of correct pixels found for each method and Table 2 shows the elapsed time of each method.

Results of Segmentation Algorithm

In order to assess our segmentation method, we compared its performance with that of the best known graph-based segmentation method called a graph cut. Tables 3 and 4 list the percentage of correct pixels found for each method and their related complexity cost, respectively.

Results of Shadow Cast Detection

In order to evaluate our shadow removal method, we compared the performance of our algorithm with that of the LBP, which is one of the fast statistical shadow removal algorithms. The LBP method is based on the texture similarity of the neighborhood of each pixel. We compared our method in percentage of correctly obtained pixels and time complexity. According to the obtained results, although LBP is about two times faster that our method, according to Equation 8, its quality is about 30% lower than ours. Figure 19 shows the preprocessing step to extract edges for the proposed

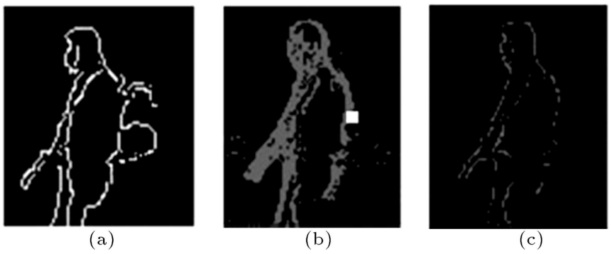


Figure 19. Sample images used for proposed shadow removal algorithm. (a) Canny edge detected result; (b) Two consequent frames difference; (c) Logical "AND" of images shown in (a) and (b).

Table 1. Error comparison of different background modeling methods using Equation 8.

Video Type	Frame Size	Temporal Averaging Model	Gradient Gaussian Model	Mixture of Gaussian Model	Proposed Method
Video 1	160 × 120	0.112	0.087	0.071	0.052
Video 2	320 × 240	0.982	0.088	0.075	0.056
Video 3	640 × 480	0.104	0.091	0.0762	0.051
Video 4	768 × 520	0.121	0.080	0.0721	0.056
Video 5	720 × 576	0.982	0.082	0.0691	0.057

Table 2. Elapsed time of different background modeling methods (in milliseconds).

Video Type	Frame Size	Temporal Averaging Model	Gradient Gaussian Model	Mixture of Gaussian Model	Proposed Method
Video 1	128 × 96	1.21	162	14	8.23
Video 2	360 × 288	1.51	231	31	20.2
Video 3	640 × 480	2.53	308	64	34.3
Video 4	768 × 520	2.92	432	87	42.8
Video 5	720 × 576	2.81	354	76	60.2

Table 3. Percentage of correct pixel by two different segmentation methods.

Video Type	Video 1	Video 2	Video 3	Video 4	Video 5
Graph-cut	87.8%	95.7%	83.8%	88.8%	87.75%
Proposed	91.96%	96.95%	91.47%	91.33%	89.91%

Table 4. Elapsed time of segmentation methods (in milliseconds).

Video Type	Video 1	Video 2	Video 3	Video 4	Video 5
Graph-cut	62	142	258	358	410
Proposed	3.85	13.37	17.18	16.19	20.25

shadow removal algorithm. Comparing the results shown in Figure 20e and 20f shows that this method is naive in removing shadow cast. Also, Table 5 lists the elapsed time of these methods and Table 6 shows the percentage of correctly obtained pixels from each algorithm.

Results of Error Concealment

In our system, as we concealed probable errors in I frames caused by transmission failures, the algorithm was tested on some standard still images and different QCIF formatted video sequences using an OpenCv package [36] in [34].

Other results obtained from our videos have been tested subjectively, using the opinions of 15 persons. These results are shown in Figure 21. Table 7 shows the subjective measurement using:

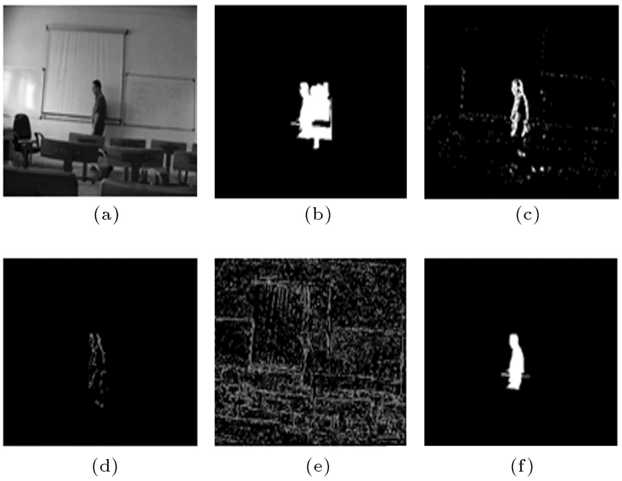

(a) (b) (c) (d) (e) (f)

Figure 20. Results of different shadow cast removal algorithms. (a) Main frame; (b) Without shadow removal; (c) Difference frame; (d) Intersection of difference frame and object boundaries; (e) LBP algorithm; (f) Proposed method.

$$R = \frac{\sum_{k=1}^{n} s_k n_k}{\sum_{k=1}^{n} n_k}. \tag{9}$$

Result of Fuzzy Scene Modeling

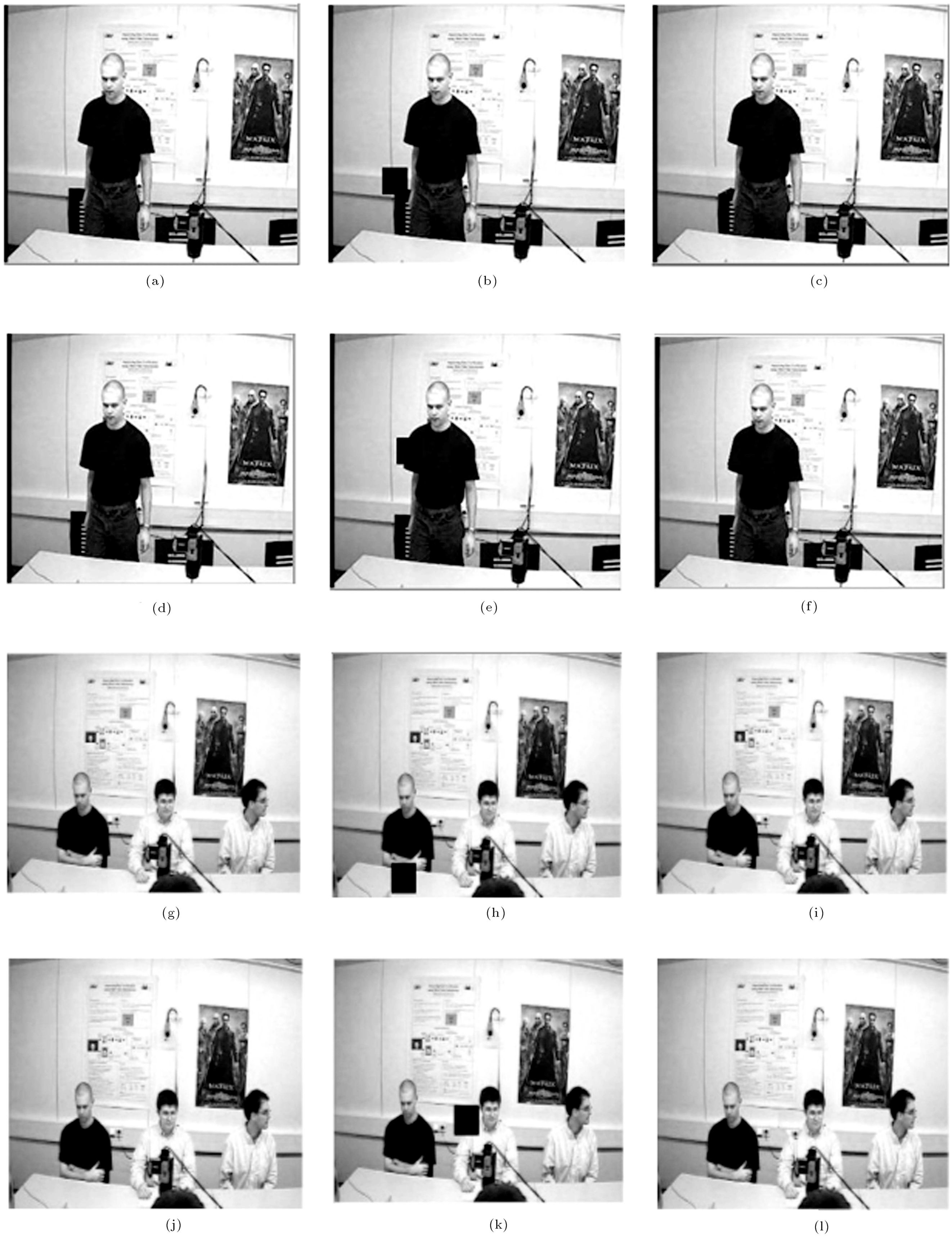
The experiments were carried out using the CIF formatted test sequences *"Stefan"*, *"Indoor PETs 2002"* and *"Bus"*. Figure 22 shows the mosaic image obtained from 60 images of the well-known *"Stefan"* sequence. Figures 22d and 22f show the obtained wide scene of the *"Stefan"* sequence constructed by the dynamic

Table 5. Elapsed time of two shadow removal methods at each frame (in milliseconds).

Video Type	Video Size	LBP Method	Proposed Method
Video 1	160 × 120	0.31	0.61
Video 2	320 × 240	0.42	0.82
Video 3	640 × 480	0.53	1.34
Video 4	768 × 520	0.62	1.52
Video 5	720 × 576	0.67	1.35

Table 6. Percentage of correctly obtained pixels of different shadow removal methods using Equation 8.

Video Type	Video Size	LBP Method	Proposed Method
Video 1	160 × 120	68.11%	86.54%
Video 2	320 × 240	68.03%	81.32%
Video 3	640 × 480	67.68%	84.87%
Video 4	768 × 520	67.36%	86.31%
Video 5	720 × 576	67.20%	83.9%



(a) (b) (c)

(d) (e) (f)

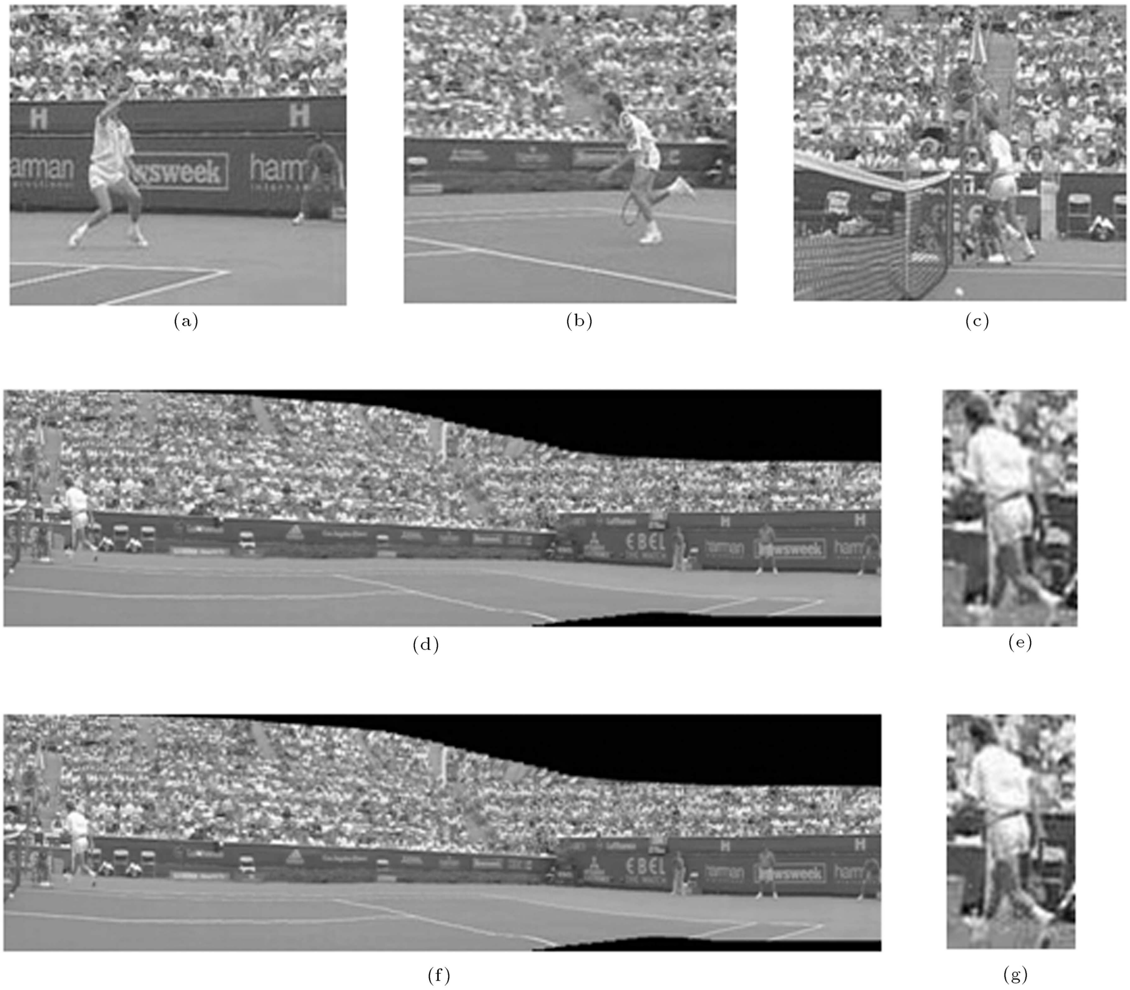
(g) (h) (i)

(j) (k) (l)

Figure 21. Results of error concealment method. (a), (d), (g) and (j): Original frames; (b), (e), (h) and (k): Erroneous frames, (black blocks are missed); (c), (f), (i) and (l): Concealed frames (see [37] for more results).

Table 7. Subjective measures of error concealment method.

	Excellent (5)	**Good (4)**	**Fail (3)**	**Poor (2)**	**Unsatisfactory (1)**	**Total Measure**	**Gold Measure**
Video 3	9	4	2	0	0	4.47	5
Video 4	10	3	1	1	0	4.47	5
Video 5	10	4	1	0	0	4.6	5



(a) (b) (c)

(d) (e)

(f) (g)

Figure 22. (a)-(c) Original images of input *"Stefan"* sequence; (d) Blended moving object on the background mosaic with constant alpha blending coefficient (Alpha = 0.8); (e) Zoomed on moving object shown in (d); (f) Blended moving object using proposed method; (g) Zoomed on moving object shown in (f); Borders of the chair (overlaid zone) in background of (g) are sharper than in (e).

mosaicing method. Then, moving objects are blended on them by applying two specific coefficient generation methods: the constant alpha blending and the adaptively alpha blending. In Figures 22e and 22f, for better observation of the quality of these results, we have shown the zoomed on parts of the blended moving object in the obtained dynamic mosaic images.

In Figure 22e, one can observe some critical misalignment artifacts around the moving object (note the chair that overlaps with the moving object). But, in the same region in Figure 22g, one can find that the misalignments are smoothed in the distance between the moving object and the rectangular model boundary. Figures 22 and 23 present the constructed dynamic mosaic image of the standard sequences *"Stefan"* and *"Indoor PETs 2002"*, respectively.

One type of artifact that often occurs in the dynamic mosaic construction process is the ghosting effect; it occasionally occurs in dynamic mosaics because of some misalignments in moving object registration or

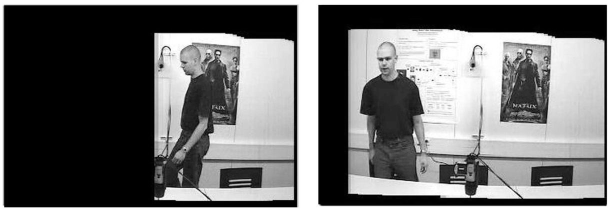
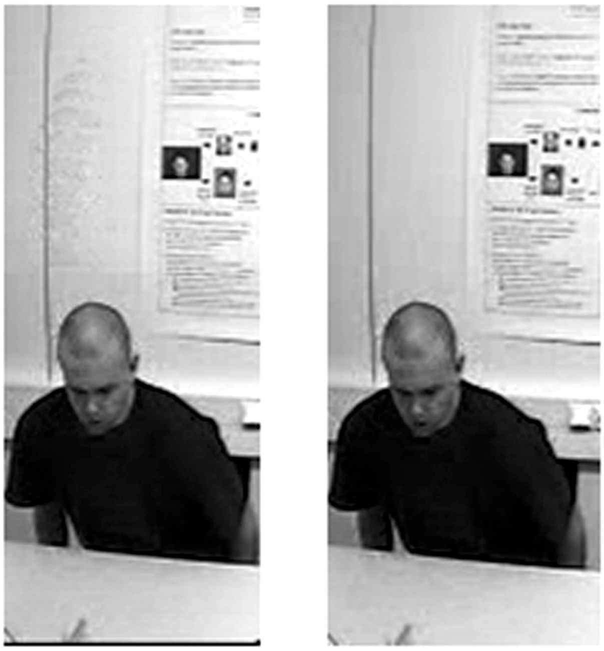
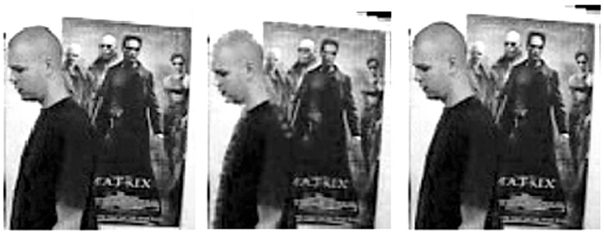


Figure 23. Constructed scene by proposed dynamic mosaicing method for *"Indoor PETs 2002"* sequence.



(a) (b)

Figure 24. (a) Ghosting effect in constant Alpha value (image 112); (b) Removal of ghost effect by proposed method.



(a) (b) (c)

Figure 25. Scene modeling on decoder side. (a) Original frame; (b) Reconstructed scene from feathering method; (c) Reconstructed scene from proposed method.

some difficulties in the motion segmentation process. Figure 24 shows that the ghosting effects are removed when using the proposed adaptive generating alpha blending coefficient values, based on object speeds.

Figure 25 shows the results of the proposed adaptive alpha blending method when applied to the *"PETs 2002"* sequence, compared to the feathering blending method. The feathering algorithm caused a boundary blurring effect on moving objects in the scene. For a computational complexity comparison, we have executed some of the well-known and conventional blending methods, such as the constant alpha blending coefficient, pyramid blending [38] and GIST [39] on three well-known sequences: *"Stefan"*, *"Indoor PETs"* and *"Bus"*. Table 8 lists the elapsed time of these methods.

As you can see in this table, the time complexity of the *"GIST"* and pyramid blending methods is too high for real-time applications. On average, the resulting mosaic qualities of these methods are visually better than the alpha blending methods. But, due to their high computational cost, they are more appropriate for offline image blending applications.

Result of Video Compression

The time complexity of the whole system at the encoder is listed in Table 9. It clearly shows that our proposed method is appropriate for real-time video transmission over the Internet, while the object's motion is smooth and slow.

The compression rate (in MB), comparing *"H.264 CODEC"*, *"MPEG4 CODEC Normal Mode"* and *"Our Proposed Method"*, is shown in Table 10. Tables 11 and 12 compare the performance of the proposed method with that of two H.264 and MPEG-4 standards by bitrate saving and PSNR quality measures. In Figure 26, a complete PSNR comparison between the proposed method and the H.264 baseline is presented. At a glance, one can note that the achieved PSNR measure is marginal, with the H.264 over 65 frames of indoor "$PETs$2002" sequences.

Figure 27 shows the reconstructed scene using our proposed method over classroom video sequences.

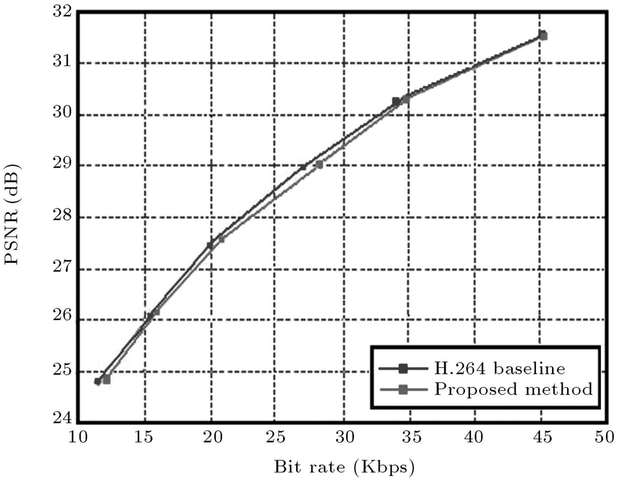


Figure 26. PSNR comparison between proposed method and H.264 baseline on 65 frame of PETs 2002 indoor sequence.

Table 8. Elapsed time of different methods (in msec/f).

Video Type	Video Format	Pyramid Blending	GIST	Constant Alpha Blending	Proposed Method
"Stefan"	CIF	1894	2099	28	120
"Indoor PETs"	CIF	2254	2584	29	208
"Bus"	CIF	2018	3602	21	156

Table 9. Required elapsed time of different parts of proposed coding system.

Function/Video Type	160 × 120 Video 1	320 × 240 Video 2	640 × 480 Video 3	768 × 520 Video 4	720 × 576 Video 5
Moving object extraction	5.11	7.161	20.14	30.93	35.76
Background subtraction	0.64	1.53	2.74	3.31	2.80
Foreground modeling	0.21	1.31	11.13	13.43	12.21
Segmentation	3.85	13.37	17.18	16.19	20.25
Shadow removal	0.61	0.84	1.34	1.52	1.42
Total	10.42	34.211	52.19	65.38	72.44
Frames/sec	95.97	29.23	19.16	15.30	13.80

Table 10. Compressed video size by different video coders (in MB).

Video Type	Video Size	Original Video size	H.264 Encoded	MPEG4 "Normal Mode"	Proposed Method
Video 1	160 × 120	3.847	0.938	1.015	0.912
Video 2	320 × 240	5.371	1.239	1.357	1.206
Video 3	640 × 480	4.865	1.034	1.149	1.005
Video 4	768 × 520	2.894	0.494	0.575	0.487
Video 5	720 × 576	3.348	0.917	0.967	0.893

Table 11. Bitrate saving comparison of different coders.

Video Type	Video Size	Compared to H.264	Compared to MPEG4 "Normal Mode"
Video 1	160 × 120	2.89%	11.33%
Video 2	320 × 240	2.81%	12.59%
Video 3	640 × 480	2.94%	14.39%
Video 4	768 × 520	1.52%	18.21%
Video 5	720 × 576	2.72%	8.36%

Table 12. PSNR quality measure.

Video Type	Video Size	H.264 Encoded	MPEG4 "Normal Mode"	Proposed Method
Video 1	160 × 120	30.21	27.23	30.00
Video 2	320 × 240	27.54	23.45	27.34
Video 3	640 × 480	31.33	28.65	31.25
Video 4	768 × 520	35.71	33.36	34.59
Video 5	720 × 576	33.65	31.78	33.51

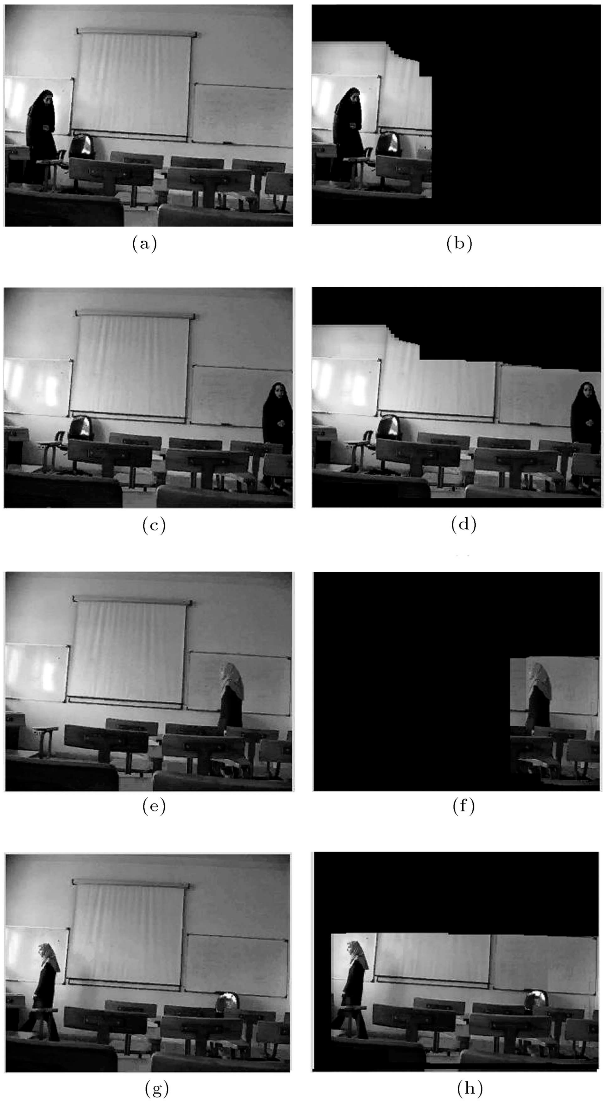


Figure 27. Reconstructed scene using proposed method on classroom sequences. (a-d) Original video scene; (e-h) Reconstructed scene by proposed encoding method.

The basic background has been removed from the reconstructed results so that only the reconstructed information from our scene modeling method has been shown in Figure 26.

CONCLUSION

In this paper, we introduced a new content-based video coding method for distance learning video dissemination. We have proposed a novel method to implement a sprite coding functionality in frame-based video encoders, such as the H.264 standard. Our proposed method consists of a novel motion segmentation method and a new algorithm for shadow removal on the encoder side. Also, on the decoder side of the proposed system, we used a robust error concealment method for error correction of lost data, and an adaptive method for scene modeling based on video mosaicing. The main idea behind our proposed coding system was decomposing the frame-based sequences to partial frames, encoding them, and then combining the received partial frames to form a scene-based (sprite) presentation of the original video. As such, the proposed method effectively reduced the redundant data needed for transmitting the video sequences. Our conducted experiments showed that the proposed encoding /decoding methods are very efficient for real-time distance learning video transmission, with approximately 24 fps for CIF formatted sequences and a minimum of 13 fps transmission for 720 × 576 frame sizes on the encoder side. It results in about 2.5% to 8% bitrate saving, with almost the same reconstructed video quality when compared to H.264 and MPEG-4 standards, respectively.

REFERENCES

1. Castro, M. et al. "Examples of distance learning projects in the European community", *IEEE Transactions on Education*, **44**(4), pp. 406-411 (2001).
2. Microsoft NetMeeting Web Site: http://www. microsoft.com/netmeeting/.
3. Haritaoglu, I., Harwood, D. and David, L.S. "W4: real-time surveillance of people and their activities", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **22**(8), pp. 809-830 (2000).
4. Wiegand, T., Sullivan, G. and Luthraet, A. "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)", JVT-G050rl, Geneva (May 2003).
5. Lipton, A.J., Fujiyoshi, H. and Patil, R.S. "Moving target classification and tracking from real-time video", *Fourth IEEE Workshop on Applications of Computer Vision*, Princeton, NJ, USA, pp. 8-14 (1998).
6. Meyer, D., Denzler, J. and Niemann, H. "Model-based extraction of articulated objects in image sequences for gait analysis", *Int. Conf. on Image Processing (IEEE)*, **3**, Washington, DC, USA, pp. 78-81 (1997).
7. Koller, D. et al. "Toward robust automatic traffic scene analysis in real-time", *Int. Conf. on Pattern Recognition (IEEE)*, Israel, pp. 126-131 (1994).
8. Felzenswal, P.F. and Hutencheler, D.T. "Efficient graph-based image segmentation", *International Journal of Computer Vision*, **59**(2), pp. 167-181 (2004).
9. Abadpour, A. and Kasaei, S. "A new FPCA-based fast segmentation method for color images", *4th IEEE Int. Symposium on Signal Processing and Information Technology (Isspit)*, Rome, Italy, pp. 72-75 (2004).
10. Stauffer, C. and Grimson, W. "Adaptive background mixture models for real-time tracking", *IEEE Conf. on Computer Vision and Pattern Recognition*, **2**, Fort Collins, CO, USA, pp. 246-252 (1999).

11. Ridder, C., Munkelt, O. and Kirchner, H. "Adaptive background estimation and foreground detection using kalman-filtering", *Int. Conf. on Recent Advances in Macaronis (ICRAM)*, pp. 193-199 (1995).

12. Satoh, Y., Okatani, T. and Deguchi, K. "A color-based tracking by kalman particle filter", *17th Int. Conf. on Pattern Recognition (ICPR)*, **3**, pp. 23-26 (2004).

13. Yilmaz, A., Javed, O. and Shah, M. "Object tracking: A survey", *ACM Computing Surveys*, **38**(4), pp. 1-45 (2006).

14. Prati, A. et al. "Detecting moving shadows: Algorithms and evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(7), pp. 918-923 (2003).

15. Prati, A. et al. "Shadow detection algorithms for traffic flow analysis: A comparative study", *Int. Conf. on Intelligent Transportation Systems (IEEE)*, pp. 340-345 (2001).

16. Stander, J., Mech, R. and Ostermann, J. "Detection of moving cast shadows for object segmentation", *IEEE Transactions on Multimedia*, **1**(1), pp. 65-76 (1999).

17. Mikic, I. et al. "Moving shadow and object detection in traffic scenes", *15th Int. Conf. on Pattern Recognition (IEEE)*, **1**, Barcelona, Spain, pp. 321-324 (2000).

18. Horprasert, T., Harwood, D. and Davis, L.S. "A statistical approach for real-time robust background subtraction and shadow detection", *7th IEEE Int. Conf. on Computer Vision, Frame Rate Workshop (ICCV)*, **99**, Kerkyra, Greece, pp. 1-19 (1999).

19. Tang, Z., Miao, Z. and Wan, Y. "Background subtraction using running Gaussian average and frame difference", *6th Int. Conf. on Entertainment Computing (ICEC)*, **4740**, Shanghai, China, pp. 411-414 (2007).

20. Zhang, D. and Wang, Zh. "Image information restoration based on long-range correlation", *IEEE Trans. on Circuits and System*, **12**(5), pp. 331-341 (2002).

21. Criminisi, A., Perez, P. and Toyama, K. "Region filling and object removal by exemplar-based image inpainting", *IEEE Trans. on Image Processing*, **13**(9), pp. 1200-1212 (2004).

22. Ashikhmin, M. "Synthesizing natural textures", *ACM Symposium on Interactive 3D Graphics (SIGGRAPH)*, New York, NY, USA, pp. 217-226 (2001).

23. Efros, A. and Freeman, W.T. "Image quilting for texture synthesis and transfer", *28th ACM Conf. on Computer Graphics (SIGGRAPH)*, New York, NY, USA, pp. 341-346 (2001).

24. Efros, A. and Leung, T. "Texture synthesis by non-parametric sampling", *Int. Conf. on Computer Vision*, **2**, Kerkyra, Greece, pp. 1033-1038 (1999).

25. Hertzmann, A. et al. "Image analogies", *28th ACM Conf. on Computer Graphics (SIGGRAPH)*, New York, NY, USA, pp. 327-340 (2001).

26. Jia, J. and Tang, Ch. "Inference of segmented color and texture description by tensor voting", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **26**(6), pp. 771-786 (2004).

27. Salama, P., Shroff, N. and Delp, E.J. "A Bayesian approach to error concealment in encoded video streams", *Int. Conf. on Image Processing (IEEE)*, **1**, Lausanne, Switzerland, pp. 49-52 (1996).

28. Kung, W., Kim, C. and Kuo, C. "Spatial and temporal error concealment techniques for video transmission over noisy channels", *IEEE Trans. on Circuits and Systems for Video Technology*, **16**(7), pp. 789-803 (2006).

29. Zeng, W. and liu, B. "Geometric-structure-based error concealment with novel applications in block-based low bit rate coding", *IEEE Trans. on Circuits and Systems for Video Technology*, **9**(4), pp. 648-665 (1999).

30. Richardson, I. *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*, Wiley; 1st Ed. (August 12, 2003).

31. Nicolas, H. "New methods for dynamic mosaicing", *IEEE Trans. on Image Processing*, **10**(8), pp. 1239-1251 (2001).

32. Irani, M. and Anandan, P. "Video indexing based on mosaic representation", *Proceedings of the IEEE*, **86**(5), pp. 805-921 (1998).

33. Bagheri, M. et al. "Content-base video coding for distance learning", *Int. Symposium on Signal Processing and Information Technology (ISSPIT)*, Giza, pp. 1005-1010 (2007).

34. Ranjbar M. and Kasaei, S. "Fast and accurate image inpainting for advanced video coders", *4th IEEE GCC Conference*, Manama, Kingdom of Bahrain (2007).

35. PETS database site: http://www.cvg.rdg.ac.uk/slides/pets.html

36. OpenCv library site: http://sourceforge.net/projects/opencvlibrary/

37. Distance learning database site: http://ipl.ce.sharif.edu/htmls/research.html

38. Adelson, E.H. et al. "Pyramid method in image processing", *RCA Engineer*, **29**(6), pp. 33-41 (1984).

39. Zomet, A. et al. "Seamless image stitching by minimizing false edges", *IEEE Trans. on Image Processing*, **15**(4), pp. 969-977 (2006).