Video object inpainting: a scale-robust method

A Koochari* and **M Soryani**
School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

Abstract: Video inpainting is the process of reconstructing damaged regions of corrupted frames. In this research, we raise a few issues in existing video inpainting systems. They are usually not robust to the change in the object scale and cannot handle large missing regions behind the moving object. In this attempt, we will address the above issues as following: first, we extract moving objects from the background and construct two mosaic images for each object, a small mosaic and a large mosaic image. The small mosaic is used to detect the amount of scale changes in the moving objects and the large one is used to inpaint partially or completely corrupted objects. We next place the inpainted moving foreground in its location and rescale the objects to their original scale. Finally, we combine the inpainted moving foreground and the background to obtain the corrected video. To speed up the process, we have utilised a multi-resolution approach so that the patch are initially matched in a coarse resolution and later are refined in a fine resolution. The results confirm the robustness of our method in handling the scale change of moving objects and large missing regions.

Keywords: video inpainting, rectification, mosaic, scale change, multi-resolution

1 INTRODUCTION

Inpainting is the process of reconstructing damaged regions in corrupted images. Video inpainting and completion are subsets of video restoration which have been investigated in the past decade. Several works have been performed in this field of image processing and machine vision.[1–8] In the early works for video inpainting, image-based methods were applied on video sequences, but since in these methods, spatial and temporal coherence of video were not considered, the output quality was not acceptable and therefore, video approaches were considered. Video inpainting methods try to use frames continuity and objects motion.

A work by Bertalmio *et al.*[9] might be the first effort in video inpainting in which the partial difference equation method was applied to all frames of a video sequence. Wexler *et al.* proposed a space–time video completion by modelling the problem to a local and global optimisation problem.[10,11] This approach performs an exhaustive search to find a cubic patch which is very time-consuming and also scale changing of the objects has not been considered in it. A video completion method has been proposed by Shiratori *et al.*, which uses motion field transfer.[12] This approach estimates motion vectors of the holes by sampling spatio-temporal patches of local motion. It then propagates colour in the missing region using the estimated motion vectors. Shih *et al.* proposed an exemplar-based inpainting to remove objects from video without ghost shadow artefacts.[13] They extended an image-based method and improved the patch matching strategy to maintain temporal continuity of video.

Zhang *et al.* isolated different layers of a video using motion segmentation, and then performed motion compensation for filling in the missing region.[14] However, obtaining accurate different video motion layers is so complex. Patwardhan *et al.* separated the moving foreground from background and then reconstructed the missing data by finding similar frames using mosaic of motion.[15] The approach proposed by Patwardhan does not consider

The MS was accepted for publication on 3 October 2011.
* *Corresponding author: Abbas Koochari, School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran; email: koochari@iust.ac.ir*

 IMAG 172 © RPS 2012 DOI: 10.1179/1743131X11Y.0000000047

scale changing in objects and does not work well when the moving object is faced with a large missing region. A matrix completion method using rank minimisation has been proposed in manifold space for reconstruction of missing regions by Ding *et al.*[16] This approach has only been examined on small missing regions and object scale changing has not been considered in it.

By Wang *et al.*,[17] a feature-based video inpainting for largely occluded moving human has been proposed. It models human behaviour with predefined features. Shen *et al.*[18] presented a video completion method for perspective camera. It constructs a space–time manifold to detect direction of object motion and then by rectification and using trajectory of object's motion repairs the moving objects. An object-based video inpainting has been proposed by Cheung *et al.*[19] and Venkatesh *et al.*,[20] where the authors extract a set of object templates and perform object-based approach video interpolation by considering motion continuity. The drawback of the template object-based approach is that pose changes occurs in some frames and also change of object structure is a serious problem. Ling *et al.*[21] inpainted a moving object using action prediction in manifold space. In their approach, an object is partitioned into three parts in order to synthesise new objects by merging these parts together. In our previous work,[22] we segmented the moving object from the background and constructed a large mosaic from object frames. Then the mosaic was inpainted by applying a large exemplar-based method. This method is suitable for periodic motions without scale changing.

Exemplar-based methods, such as space–time video completion[11] and video inpainting with large patches,[22] cannot find appropriate patches to complete video sequences when object scale changing occurs and therefore, these methods face difficulty in such situations.

In this paper, we present a framework for filling in missing parts in video frames and we raise some issues in existing video inpainting systems such as facing with a large missing region and scale changing in objects. Also, a multi-resolution method has been utilised to speed up the process of the proposed method. The paper has been organised as follows. the algorithm is explained in Section 2. Section 3 presents experimental results and Section 4 concludes the paper with a discussion of future works.

2 THE PROPOSED ALGORITHM

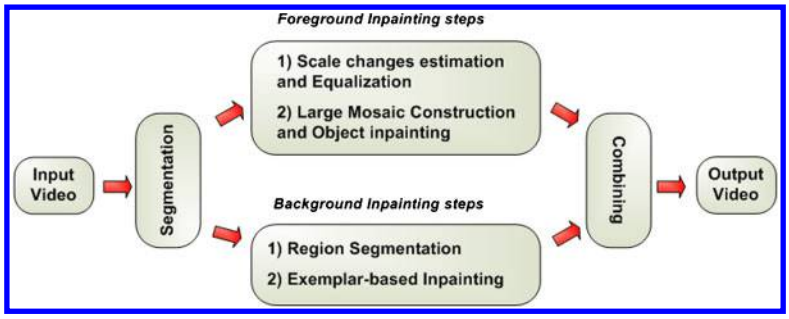
In this algorithm, some issues in existing inpainting systems such as negligence of scale changing in objects and not considering large corrupted regions are noticed. In the beginning, we consider several assumptions for the video sequences which our algorithm can inpaint. These assumptions which have been considered by most state of the art works are as follows:

- the analysed video sequences are obtained by a static camera and under suitable lighting conditions (without objects shadows)
- the scene is assumed to be static and consists of a stationary background without dynamic texture
- motion of the moving objects is periodic or semi-periodic and scale changing occurs in some video sequences
- objects move horizontally (from left to right or vice versa) and contrast of each object with respect to the background is high for better background subtraction.

In the presented algorithm, the three-dimensional problem is transformed to a two-dimensional problem and spatio-temporal consistency is maintained by constructing a large mosaic. The two-dimensional problem is then solved using an image-based method. Schematic diagram of the proposed method is shown in Fig. 1. At first, the moving foreground is separated from the background, and then the objects are rectified in all frames and a large mosaic image is constructed. Afterwards, the corrupted frames are inpainted using an exemplar-based method in two separate steps. Static background is repaired separately and the resulting video is obtained by combining the inpainted moving foreground and the background.

2.1 Foreground and background segmentation

One of the basic phases in video pre-processing is performing accurate background estimation and segmenting the moving object. Moving object detection usually is done by background subtraction. In this method, an estimation of the background is obtained and the moving foreground is detected by subtracting the current frame from it. Several methods have been proposed for background modelling.[23–26] Mixture of Gaussian[23] is a traditional method for background modelling that has some disadvantages such as weakness to model fast changing backgrounds and



1 Schematic overview of the proposed algorithm

its dependency to learning rate. Kernel density-based[24] method requires high memory to model a background. In this paper, a codebook-based method, which has been proposed by Sigari and Fathy,[26] is used. The codebook model is very fast and works better than mixture of Gaussian and kernel density-based method. Also, it uses less memory relative to other methods.

The background modelling method is a pixel-based approach that has two layers. The main layer contains the basic background model, and the second layer is used to model a new background when large changes in input frames are detected. Since the background is static, the main layer is used more than second layer. Each pixel is modelled by a codebook which contains one or more codewords. So, each pixel models a section of the background. In the segmentation step, if a pixel is in the bound of available codewords, it is known as background; otherwise, it is a foreground pixel.

Suppose that $I=\{I_1, I_2, \ldots, I_N\}$ is the set of illumination values of a given pixel at sequential frames (N is number of frames). A codeword is considered by a six-tuple vector as follows: $cw_i=(\breve{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i)$, where $\breve{I}_i$ and $\hat{I}_i$ are minimum and maximum values of the illumination respectively. f_i, is the frequency of access to codeword in the training step, λ_i, is the maximum negative run length, and p and q are numbers of the first and last frames for which access to this codeword has been accomplished. Codeword constructing for each pixel is performed as follows: first, all codewords of the input pixel, I_t, are searched to find a codeword in range $\left[\breve{I}_i, \hat{I}_i\right]$. If no codeword is found to match with the input pixel, a new codeword is generated as follows:

$$cw_L \leftarrow [\max(0, I_t-\alpha), \min(255, I_t+\alpha), 1, t-1, t, t] \quad (1)$$

Otherwise, the matching codeword, cw_m, is updated as follows:

$$cw_m \leftarrow \left[\frac{I_t-\alpha+f_m\breve{I}}{f_m+1}, \frac{I_t+\alpha+f_m\hat{I}}{f_m+1}, f_m+1, \max(\lambda_m, t-q_m), p_m, t\right] \quad (2)$$

After codeword construction, values of all λ_i are changed to $\lambda_i \leftarrow \max(\lambda_i, N-q_i+p_i-1)$.

To separate the foreground from the background, if a matched codeword from pixel's codebook is not found with input pixel, the pixel belongs to foreground; otherwise, it belongs to background and the found codeword, cw_m, is updated as follows:

$$cw_m \leftarrow \left[(1-\beta)(I_t-\alpha)+\beta\breve{I}_t, (1-\beta)(I_t+\alpha)+\beta\breve{I}, f_m+1, \max(\lambda_m, t-q_m), p_m, t\right] \quad (3)$$

$$BGS(I)=\begin{cases}\text{background}, & \text{if } \breve{I}_t \le I_t \le \hat{I}_t \\ \text{foreground}, & \text{otherwise}\end{cases} \quad (4)$$

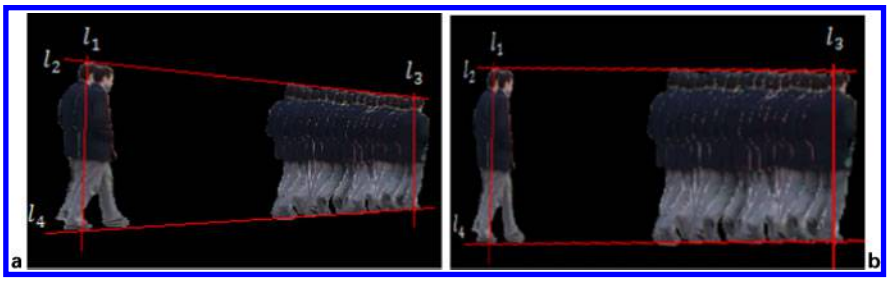
The parameters α and β are considered less than one and equal to 10, respectively.

2.2 Foreground inpainting

Foreground and background are inpainted separately. At first, the damaged frames are automatically detected to be used for foreground inpainting step. Then, to inpaint the foreground, two mosaic images (a small mosaic and a large mosaic) are created. The small mosaic is used for rectification and scale equalisation of the moving objects and the large mosaic is used to inpaint the corrupted objects.

2.2.1 Detection of damaged key-frames

To inpaint a moving object in the corrupted region, the first and the last damaged frames in which the object has been corrupted must be specified firstly. To ease the process, these key-frames (i.e. the first and last damaged frames) are automatically detected instead of manual specification. Automatic detection of damaged moving foreground has not been mentioned in other works.



2 The small mosaics before (a) and after (b) removing the projective distortion

To select the damaged frames, the corrupted regions are copied to foreground frames. Then, if boundary of each moving object in a foreground frame intersects with the hole boundary, that frame is considered as a damaged frame. If the sequence of damages frames, which have overlapped with the hole, are denoted as f_i, f_{i+1}, …, f_a, f_b, …, f_{j-1}, f_j; $i<a<b<j$, we consider the first (f_i) and last (f_j) damaged frames as key-frames in which the moving foreground enters to and exits from the occluding area. The frames between f_a and f_b are those for which the object has been completely corrupted. It is clear that for other holes, this procedure should be repeated.

2.2.2 *Scale change estimation and equalisation*

Object scales should be equal in all frames for the inpainting step. For this purpose, since object motion is assumed cyclic, duration of periodic motion can be calculated to obtain the amount of projective distortion. Another way to calculate the amount of projective distortion, as depicted in Fig. 2, is line fitting in which the highest and lowest points of objects in small mosaic, when feet of the object are closed (l_2 and l_4), are connected together. Using fitted lines l_2 and l_4 and lines between head and feet in the first and last frames (l_1 and l_3), the amount of projection is obtained. Also, if camera is not orthogonal to the image plane, location of head and feet of the object can be obtained by Eigen Analysis.[27] After calculating the amount of projective distortion, object scale is equalised in all frames by affine and metric rectification. Let l_1, l_2, l_3 and l_4 be lines of projective distortion; therefore, $p_1=l_1\times l_3$ and $p_2=l_2\times l_4$ are junction points of these lines. To remove projective distortion and recover the affine properties from a frame, the vanishing line, $l_v=p_1\times p_2$, is mapped into a line at infinity.[28] This work is done by a homographic matrix as follows:

$$H_p=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ l_1 & l_2 & l_3\end{bmatrix} \tag{5}$$

where $l_v=[l_1\ l_2\ l_3]^T$. Then, the H_a Matrix (equation (1)) is obtained for affine to metric rectification using the two sets of orthogonal lines. Here, Cholesky factorisation is used to obtain the K matrix [28] (K is a 2×2 matrix). Fig. 2 shows the small mosaic before and after applying the H_r matrix, respectively. The H_r matrix is obtained by composition of the affine and the metric transformations.

$$H_a=\begin{bmatrix}K & 0\\ 0 & 1\end{bmatrix} \tag{6}$$

After calculating the H_r matrix, it is applied to all frames separately for equalising the object scale.

2.2.3 *Large mosaic construction and object inpainting*

After equalising the object's scales, the maximum width and height of the segmented object when feet and hands of the object are open in all frames are obtained and considered as vertices of a reference window. Then the segmented objects are aligned to construct a large mosaic without any possible overlaps. For this purpose, the centre of each object in the large mosaic frame is defined as follows:

$$cm_{O_i}=cm_{O_{i-1}}+w+v_i,\ v_i=cm_{O_i}-cm_{O_{i-1}} \tag{7}$$

where cm_{O_i} is the centre of object of frame i in the large mosaic, w is the width of the reference window and v_i is defined as distance traversed by the object between two consecutive frames. To simplify the reconstruction of the large mosaic, it is possible to copy each segmented object into the mosaic image (similar to Fig. 2) and then provide a distance w between centre of mass of each object i and object $i-1$ (O_i and O_{i-1}). Also, the first and the last damaged frames are automatically specified, and all missing regions are copied to the large mosaic for foreground inpainting step. In Fig. 3, section of large

3 Section of the large mosaic image including the missing region

mosaic image, when the object enters and exits to/from the missing region, has been shown.

Determination of the amount object's movement between consecutive frames is very important and effective in selection of best objects for inpainting. Otherwise, single-object template matching cannot preserve temporal continuity of object's motion by its own.

As object motion was periodic in previous work:[22] only one step was used to inpaint the large mosaic. But here, since motion is considered semi-periodic and scale changing occurs, after constructing the large mosaic, inpainting is done in two steps. In the first step, objects which are partially corrupted are inpainted, and the remaining objects, which are completely corrupted, are inpainted using a large patch in the next step.

The partially corrupted objects, whose sizes are more than a threshold, are determined as follows:

$$O_{\mathrm{i}} = \begin{cases} \text{partially corrupted} & S_{\mathrm{O_i}} \geq \alpha \\ \text{completely corrupted} & S_{\mathrm{O_i}} < \alpha \end{cases} \tag{8}$$

$$\alpha = \frac{1}{2n} \sum_{\mathrm{i}=1}^{\mathrm{n}} S_{\mathrm{O_i}} \tag{9}$$

where O_{i} is object in frame i, $S_{\mathrm{O_i}}$ is the number of uncorrupted pixels of the object O_{i}, n is the number of complete objects and α is a threshold which is considered half of the mean number of the complete objects' pixels. Figure 4 shows the mosaic image where complete, partially corrupted and completely corrupted objects have been specified.

2.2.3.1 Partially corrupted objects inpainting

Inpainting of partially corrupted objects is an important step of this method and relies on uncorrupted pixels of each object. The exemplar-based method is used to complete these objects in the large mosaic. To simplify the explanation, only one moving object has been considered. A patch from the sequence of complete objects is considered for this step. The patch should be sufficiently large to enclose a whole object, but because the spatio-temporal consistency is an important goal in this problem, the height and width of the patch are considered equal to the height and three folds of the width of the reference window, respectively. Since the large patch covers three objects entirely, the structure of the moving object and temporal continuity of objects in three consecutive frames are preserved. This assumption helps to maintain periodicity of each object with regard to its left or right objects. To complete each corrupted object, the following steps are performed:

Step 1. Consider two target-patches, one from front and one from rear of the large mosaic. The front target-patch includes the first corrupted object and its two previous complete objects, and the rear target-patch includes the first corrupted object in the right side of the mosaic with its two next complete objects.

Step 2. Find the target-patch, Ψ_{T}, which has higher priority between these two target-patches, based on their confidence terms as follows:

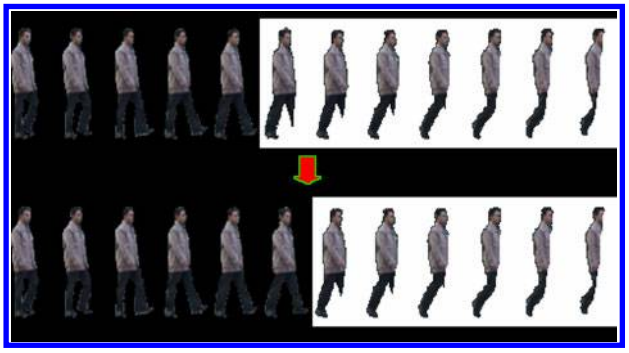
$$C(\Psi_{\mathrm{T}}) = \frac{\sum_{q \in \Psi_{\mathrm{T}} \cap \bar{\Omega}} C(q)}{|\Psi_{\mathrm{T}}|} \tag{10}$$

where Ψ_{T} is the target-patch, $|\Psi_{\mathrm{T}}|$ is the area of the target-patch, q is an uncorrupted pixel in the target-patch and Ω is the target region (corrupted region).

Step 3. The target-patch, Ψ_{T}, is compared with all patches of complete object's region (source patches), and the best matching patch is selected.

Step 4. Only the corrupted part of the partially corrupted object is replaced by corresponding

4 The mosaic image which illustrates complete, partially corrupted and completely corrupted objects



5 Partially corrupted object inpainting. The first and second rows show the mosaic before and after one step of partially corrupted object inpainting

pixels of matching patch. At the end, the confidence terms for all pixels of target region are updated based on $C(\Psi_T)$.

Sum squared difference metric is used to compute similarity of the two patches to find best match. It must be noticed that to initialise the algorithm, confidence terms of all uncorrupted and corrupted pixels are set to 1 and 0, respectively.

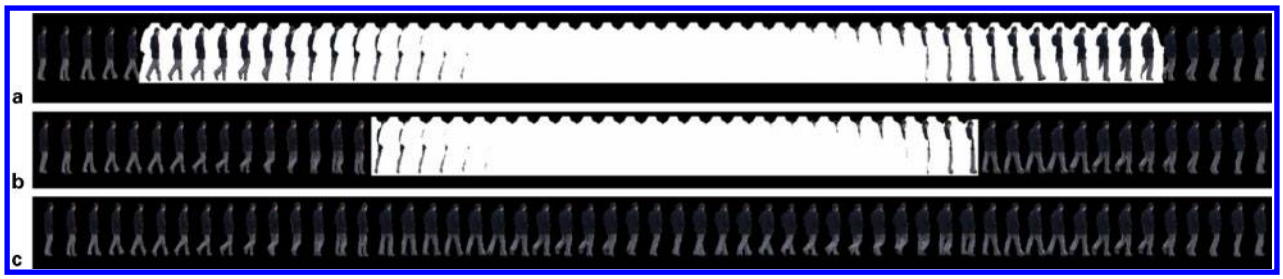
This algorithm is run until all partially corrupted objects are repaired. Since object appearance is changed in different frames, repairing more than one object in each iteration creates abnormal results. However, when the corrupted region is very small, considering the patch size equal to maximum height and width can be acceptable and can maintain texture and structure of the object, but it cannot preserve temporal consistency of the object's motion. Figure 5 shows one step of the partially corrupted object inpainting.

Since utilisation of this algorithm creates a visible pose change in the connection point between left and right of the missing region, a different approach has been considered for completely corrupted objects.

2.2.3.2 Completely corrupted object inpainting

For inpainting of completely corrupted objects, a large target-patch, which encloses the remaining missing region, is considered in the large mosaic. Inpainting is done by finding the best matching patch from the complete object's area. Since a large patch is used for this step, only the confidence term (per cent of uncorrupted pixels; see Appendix) is calculated to obtain the best matching patch. In fact the remaining parts of objects help to obtain the best matching patch. After selecting the best matching patch, all the remaining objects in the target-patch are totally replaced with new objects. Since the objects are deformable, only substituting corrupted parts from the matching patch cannot reconstruct objects well and creates abnormal objects (even if the object motion is periodic). The second and third rows in Figs. 6 and 8 show the steps of partially and completely object inpainting.

Also, a multi-resolution method is used to speed up the patch searching. Pyramid resolution is constructed by down-sampling of the large mosaic by a factor of two at each dimension. Coarse-to-fine levels are denoted by the first and kth levels in the pyramid. Therefore, to find the best patch, first, the coarser level (the first level, i.e. lower scale) is searched to obtain approximate location of the best patch. Then searching is bounded in the neighbouring patch in finest level (the kth level) to find accurate location of the best match. Since the patch size is large, searching the low level of the pyramid is an effective approach for speeding up the proposed algorithm. In our experiments, number of pyramid levels is defined based on the size of input video sequence. Since the moving objects in image mosaic must be distinguished in lower levels, we cannot increase the pyramid level very much. Therefore, k was set to 4 in all of our test videos.



6 (a) A section of the mosaic image with the determined missing region; (b) the mosaic image after inpainting of partially corrupted objects; (c) the final result after total inpainting

7 Results of the proposed method for the first video. The first row shows consecutive frames when the man enters to and exits from the occluding area. The second row shows the inpainted frames using the proposed method. In the second row, the objects in the third and fourth frames have been considered completely occluded

After completing the mosaic, the inpainted mosaic is decomposed into frames by placement of each object in its location. At the end, the video frames are returned to original scale using the inverse of the H_r matrix.

2.3 Background inpainting

To inpaint the background, the exemplar-based method proposed by Criminisi *et al.*[29] is used on a frame (see Appendix for more detail). The chosen frame, for background inpainting, is selected from frames in which the object is completely occluded. Since the background is assumed to be static, the hole (the damaged region) is filled in by the source region (the safe parts of the background) and the reconstructed hole is copied into all other frames. This approach can maintain the texture and structure of static backgrounds. Dynamic backgrounds are more challenging than static or stationary backgrounds since temporal continuation in them should be carefully observed.

In our method, background is segmented into some regions, and then different areas of the hole are inpainted separately using neighbouring regions of the segmented background. At the end, the boundary of regions is filled in. The reason of doing this is to prevent the propagation of wrong data to other regions.

Table 1 Details of the test video sequences (DF: damaged frames)

Video	Frame size	Length	No. of DF
Statue	320×180	110	46
Tree	320×180	185	54
Girl	300×100	240	52
Three person	320×240	75	15

3 EXPERIMENT RESULTS

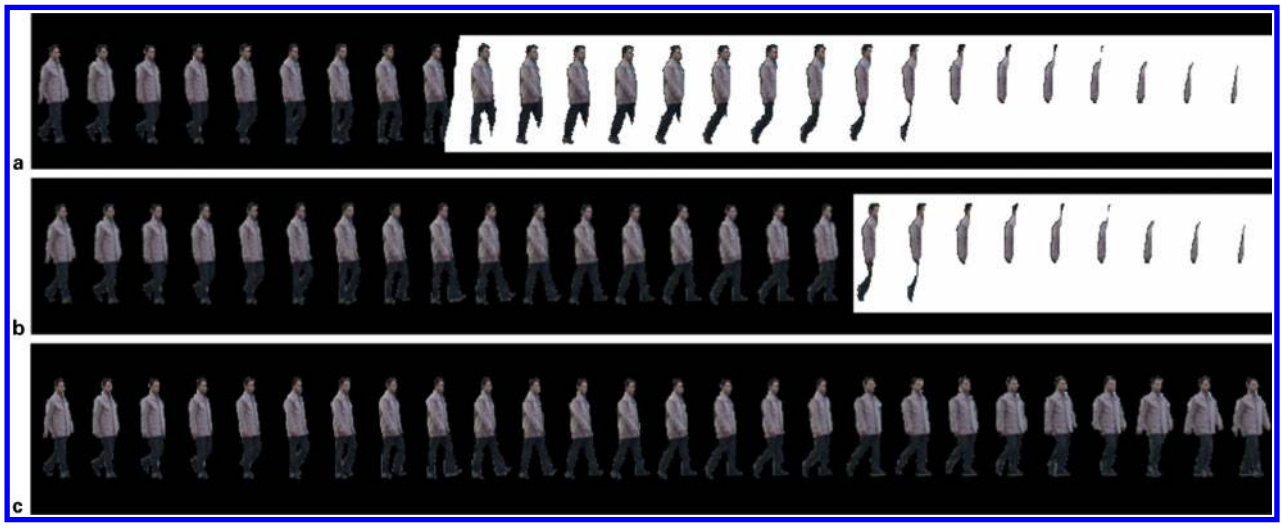
To test the proposed algorithm, a number of video sequences with a static background have been used. A hand-held camera was used to make two video sequences in which object scale is changed and the object is faced with a large occluding region. Two other test videos do not contain object scale changing and are used for comparison of our method with methods of other papers. Details of the video sequences, which have been used in this section, are shown in Table 1.

In the first video, a person moves from left to right while object scale is decreased. The occluding object in this video is a statue (Ferdowsi statue) that has been created manually. Figure 6 shows the results of the proposed algorithm on the first video. In Fig. 6a, the original mosaic image is shown, in Fig. 6b, the results after inpaiting of the partially occluded objects is illustrated, and in Fig. 6c, the final result after applying the second step of inpainting algorithm on the mosaic image is shown.

As shown in Fig. 6, the remaining parts of object are used to obtain the best matching patch, and then these parts are removed and replaced with the new complete objects in the selected large patch. Also Fig. 7 shows the results of proposed method on some selected frames.

In the second video, a person moves from left to right and crosses behind an occluding object (a tree), while object scale is increased. Figure 8 shows a section of mosaic image, the mosaic image after inpainting of the partially corrupted objects and the final result of inpainting after second step, respectively.

Some selected frames from the original video sequence and results of the proposed method for the second video can be seen in Fig. 9.



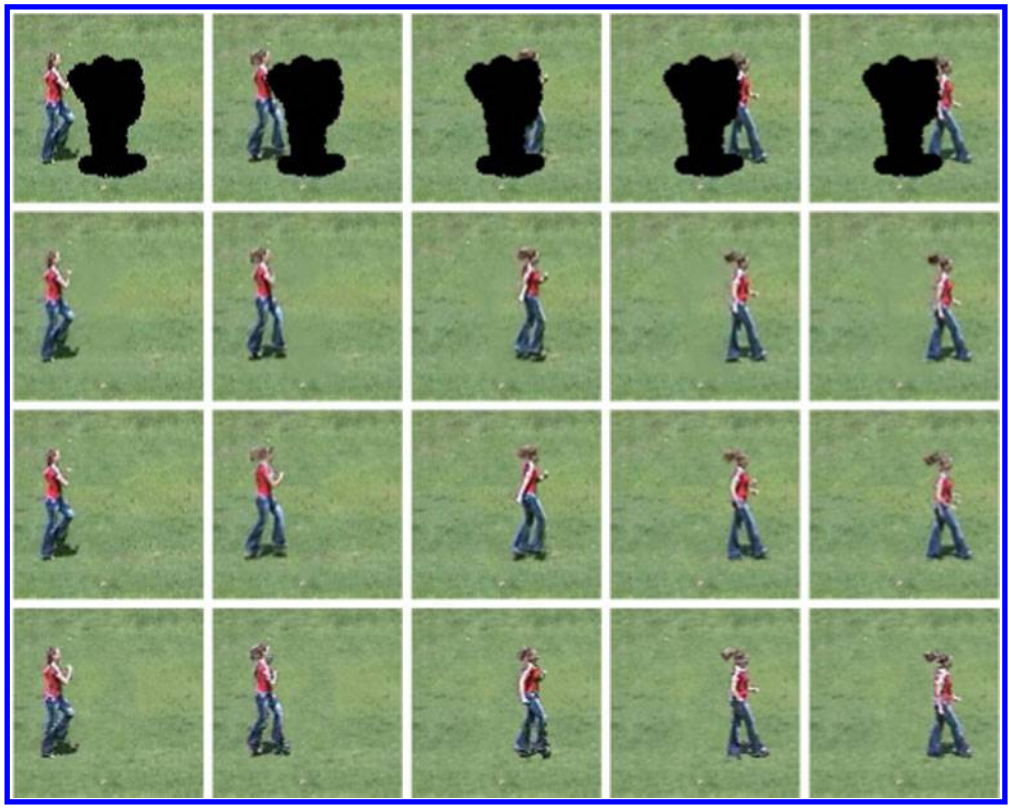
8 (a) Section of the mosaic image with determined missing region; (b) the mosaic image after inpainting of partially corrupted objects; (c) final result after total inpainting

To compare the proposed method with other methods, the 'jumping girl' video, which was captured and used by Wexler,[11] has been used. In this video, a girl moves from left to right and passes behind an occluding object (a person). Figure 10 shows results of the proposed method compared to the results of Wexler *et al.*'s (space–time video completion)[11] and Venkatesh *et al.*'s [20] methods, respectively. In the first row of Fig. 10, the occluding object has been shown with a black mask.

As Fig. 10 shows in the second row (from left to right) results of space–time video completion method,[11] object structure has been changed in some of the frames such as the third and fourth images, and hand of the object has not been inpainted in the second image. Also in all frames of the second row, over-smoothing is seen in the background. Complex and time consuming computations are other disadvantages of the space–time video completion method. As shown in the third row, results of Venkatesh *et al.*,[20] a redundant hand is created in some of the frames such as the fourth and the fifth frames. The fourth row shows the results of our propose algorithm. Since our proposed method separates the foreground from the background, over-smoothing is not seen in the background and object structure has been preserved in all of the frames.

The other test video, which has been captured by Venkatesh *et al.*,[20] was used for comparison between our method and their method, when three persons walk in the scene. Since Venkatesh *et al.*'s method is an object-based method, this comparison was done to show advantages of our method relative to their

9 Results of the proposed method for the second video. The first row shows some frames when the man enters to and exits from behind the tree. The second row shows inpainted frames using the proposed method. In the third and fourth frames, the object has been replaced entirely



10 Comparison of the results of our proposed method with the results of algorithms proposed by Wexler *et al.*[11] and Venkatesh *et al.*[20] The first row shows the girl passing behind the occluding mask; the second row shows the inpainted sequence using space–time completion method;[11] the third row shows the inpainted sequence using Venkatesh *et al.*'s method[20] and finally, the fourth row shows the results of our algorithm

method. In this video, scale change has not occurred. We want to inpaint the person who moves from left to right, while the occluding objects are two persons that move from right to left. Figure 11 shows the results of this comparison. From left to right: the first, second and third columns show a number of original frames, results of Venkatesh *et al.*'s method and our method, respectively.

As can be seen in the first row of the second column of Fig. 11, structure of the moving object is damaged due to abnormality in the hinder foot (by Venkatesh algorithm), while the hinder foot has not been occluded. Pose of the moving object has been changed in some of the frames (for example, see the second and the third rows of the second column; the left leg of the object must be front, but direction of the moving object has been changed and the right leg is in front after inpainting by Venkatesh *et al.*'s method). Also, a redundant hand has been created in some of the frames such as in the second to sixth frames of the second column; in these figures, the inpainted object has two right hands, while this is not the case for our results (the third column). Since in our method, multiple-object patches are used in which distance between objects of consecutive frames is considered, and the pose changing and redundant hand has not occurred.

As mentioned above, background inpainting is done separately. Figure 12 show the original frame and the inpainted background after removing the statue, respectively. Also, results of background inpainting for second and third videos are shown in Figs. 13 and 14. In these figures, the tree and the person have been removed from background and the resulting holes have been inpainted using the exemplar-based algorithm. Since the fourth video does not contain a static occluding area (Fig. 11), median of a number of frames is used as the background.

Finally, the inpainted background and foregrounds are composed to obtain the output video. Output videos can be viewed at: http://webpages.iust.ac.ir/koochari/inpainting/objectinpainting.html.

11 Comparison of the results of our algorithm with results of Venkatesh *et al.*'s algorithm[20] when two occluding persons, moving from right to left, are removed. The first column shows a number of original frames of the test video. The second column shows the inpainted frames using algorithm by Venkatesh *et al.*[20] The third column shows the results of the proposed algorithm

4 CONCLUSION AND FUTURE WORKS

In this paper, a video inpainting method has been presented which deals with situations where an object is faced with a large occluding region and object scale is changed. At first, the moving object is segmented from the background and then the segmented sections are repaired separately. Next step, the damaged frames are automatically detected to be used for foreground inpainting step. Since object scale is changed, a small mosaic image is created for determining the amount of projective distortion. After affine and metric

12 Background inpainting of the first video: (a) original frame; (b) inpainted background after removing the statue

13 Background inpainting of the second video: (a) original frame; (b) inpainted background after removing the tree

14 Background inpainting of the third video: (a) original frame; (b) inpainted background after filling in the mask

rectification, all objects in the frames become ready for construction of a large mosaic image without any overlap between two objects in the frame sequence. The large mosaic is used to inpaint foreground objects in two steps. In the first step, the partially occluded objects are inpainted, and in the second step, a large patch is used to inpaint the completely occluded objects in order to maintain the continuity of the objects. In both steps, a multi-resolution method has been used to increase the speed of the process. After mosaic inpainting, objects are put in their locations and foreground frames are built. Also, the background is inpainted separately using an exemplar-based method. Finally, the objects return to the original scale and the output video is created by superimposing the inpainted object on the inpainted background. It has been shown that the proposed method works better than other approaches and produces more visually pleasant results.

The main problem in the proposed algorithm is the lack of smooth transition when objects enter the occluding region or exit from it. This problem could be solved using object synthesis to create new object motions in a future work. Other works can be conducted to study sever conditions such as camera motion and non-periodic object motion. Object representation and analysis of its motion can also be considered for future works.

REFERENCES

1 Bertalmio, M., Sapiro, G., Caselles, V. and Ballester, C. Image inpainting, Proc. ACM SIGGRAPH Conf. on *Computer graphics: SIGGRAPH 2000*, New Orleans, LA, USA, July 2000, ACM, pp. 417–424.

2 Oliveira, M., Bowen, B., McKenna, R. and Chang, Y.-S. Fast digital image inpainting, Proc. Int. Conf. on *Visualization, imaging and image processing: VIIP 2001*, Marbella, Spain, September 2001, IASTED, pp. 261–266.

3 Bertalmio, M., Vese, L., Sapiro, G. and Osher, S. Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.*, 2003, **12**, 882–889.

4 Sun, J., Yuan, L., Jia, J. and Shum, H. Y. Image completion with structure propagation. *ACM Trans. Graph.*, 2005, **24**, 861–868.

5 Ho, H. T. and Goecke, R. Automatic parametrisation for an image completion method based on Markov random fields, Proc. IEEE Int. Conf. on *Image processing: ICIP 2007*, San Antonio, TX, USA, September 2007, IEEE Computer Society, Vol. 3, pp. 541–544.

6 Liu, D., Sun, X., Wu, F., Li, S. and Zhang, Y. Q. Image compression with edge-based inpainting. *IEEE Trans. Circuits Syst. Video Technol.*, 2007, **17**, 1273–1287.

7 Matsushita, Y., Ofek, E., Ge, W., Tang, X. and Shum, H. Y. Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, 1150–1163.

8 Cai, J.-F., Chan, R. H. and Shen, Z. A framelet-based image inpainting algorithm. *Appl. Comput. Harmon. Anal.*, 2008, **24**, 131–149.

9 Bertalmio, M., Bertozzi, A. L. and Sapiro, G. Navier-stokes, fluid dynamics, and image and video inpainting. *Comput. Vis. Pattern Recogn.*, 2000, **1**, 355–362.

10 Wexler, Y., Shechtman, E. and Irani, M. Space–time video completion. *Comput. Vis. Pattern Recogn.*, 2004, **1**, 120–127.

11 Wexler, Y., Shechtman, E. and Irani, M. Space–time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007, **29**, 463–476.

12 Shiratori, T., Matsushita, Y., Kang, S. B. and Tang, X. Video completion by motion field transfer, Proc. IEEE Computer Society Conf. on *Computer vision and pattern recognition: CVPR'06*, New York, USA, 2006, IEEE Computer Society, Vol. 1, pp. 411–418.

13 Shih, T. K., Tang, N. C. and Hwang, J. N. Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity. *IEEE Trans. Circuits Syst. Video Technol.*, 2009, **19**, 347–360.

14 Zhang, Y., Xiao, J. and Shah, M. Motion layer based object removal in videos, Proc. 7th IEEE Workshop on *Applications of computer vision: WACV 2005*, Breckenridge, CO, USA, January 2005, IEEE Computer Society, pp. 516–521.

15 Patwardhan, K. A., Sapiro, G. and Bertalmio, M. Video inpainting under constrained camera motion. *IEEE Trans. Image Process.*, 2007, **16**, 545–553.

16 Ding, D., Sznaier, M. and Camps, O. A rank minimization approach to video inpainting, Proc. IEEE 11th Int. Conf. on *Computer vision: ICCV 2007*, Rio de Janeiro, Brazil, October 2007, IEEE Computer Society, pp. 1–8.

17 Wang, H., Li, H. and Li, B. Video inpainting for largely occluded moving human, Proc. IEEE Int. Conf. on *Multimedia and expo: ICME 2007*, Beijing, China, July 2007, IEEE Computer Society, pp. 1719–1722.

18 Shen, Y., Lu, F., Cao, X. and Foroosh, H. Video completion for perspective camera under constrained motion, Proc. 18th Int. Conf. on *Pattern recognition: ICPR 2006*, Hong Kong, China, August 2006, IEEE Computer Society, Vol. 3, pp. 63–66.

19 Cheung, S., Zhao, J. and Venkatesh, M. V. Efficient object-based video inpainting, Proc. IEEE Int. Conf. on *Image processing: ICIP 2006*, Atlanta, GA, USA, October 2006, IEEE Signal Processing Society, pp. 705–708.

20 Venkatesh, M. V., Cheung, S. and Zhao, J. Efficient object-based video inpainting. *Pattern Recogn. Lett.*, 2009, **30**, 168–179.

21 Ling, C.-H., Liang, Y.-M., Lin, C.-W., Chen, Y.-S. and Mark Liao, H.-Y. Video object inpainting using manifold-based action prediction, Proc. IEEE 17th Int. Conf. on *Image processing: ICIP10*, Hong Kong, China, September 2010, IEEE Signal Processing Society, pp. 425–428.

22 Koochari, A. and Soryani, M. Exemplar-based video inpainting with large patches. *J. Zhejiang Univ. — Sci. C*, 2010, **11**, 270–277.

23 Stauffer, C. and Grimson, W. E. L. Adaptive background mixture models for real-time tracking, Proc. IEEE Conf. on *Computer vision and pattern recognition: CVPR'99*, Ft Collins, CO, USA, June 1999, IEEE Computer Society, pp. 246–252.

24 Elgammal, A., Harwood, D. and Davis, L. S. Non-parametric model for background subtraction. *Lect. Notes Comput. Sci.*, 2000, **1843**, 751–767.

25 Piccardi, M. Background subtraction techniques: a review, Proc. IEEE SMC 2004 Int. Conf. on *Systems, man and cybernetics: SMC 2004*, The Hague, The Netherlands, October 2004, IEEE Computer Society, Vol. 4, pp. 3099–3104.

26 Sigari, M. H. and Fathy, M. Real-time background modeling/subtraction using two-layer codebook model, Proc. Int. MultiConf. of *Engineers and computer scientists: IMECS 2008*, Hong Kong, China, March 2008, Vol. 1, IAENG, pp. 717–720.

27 Lv, F., Zhao, T. and Nevatia, R. Self-calibration of a camera from video of a walking human, Proc. 16th IEEE Int. Conf. on *Pattern recognition: ICPR 2002*, Quebec City, Que., Canada, August 2002, IEEE Computer Society, pp. 562–567.

28 Hartley, R. I. and Zisserman, A. Multiple View Geometry in Computer Vision, 2004 (Cambridge University Press, Cambridge).

29 Criminisi, A., Perez, P. and Toyama, K. Region filling and object removal by exemplar-based inpainting. *IEEE Trans. Image Process.*, 2004, **13**, 1200–1212.

APPENDIX

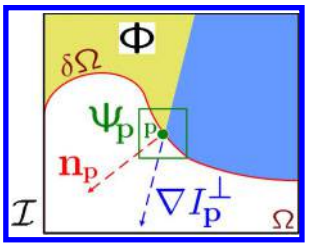
Review of the exemplar-based method

The missing region (the target region) is filled in by the source region (the safe region). To fill the holes of the image, a square patch is considered for each pixel on the boundary of the target region, and then this square patch is filled in by the patch which has the most similarity to the source region. To keep the structure of the image, the priority value is calculated for each pixel on the boundary of the target region, and the pixel with maximum priority is selected to fill in at first. The priority value of the pixel is the product of confidence and data terms as follows:[29]

$$P(p)=C(p)D(p) \tag{11}$$

The confidence term $C(p)$, is the number of undamaged pixels divided into all pixels of surrounding patch of the pixel P. The data term $D(p)$, is high if gradient of the pixel and orthogonal to boundary of the pixel are unidirectional which are explained as follows:

$$C(p)=\frac{\sum_{q\in\Psi_p\cap\bar{\Omega}} C(q)}{|\Psi_p|} \tag{12}$$



15 Exemplar-based inpainting (borrowed from Criminisi *et al.*[29])

$$D(p)=\frac{\left|\nabla I_{\mathrm{p}}^{\perp} n_{\mathrm{p}}\right|}{\alpha} \tag{13}$$

where Ψ_{p} is a patch which its centre is location p, $|\Psi_{\mathrm{p}}|$ is the area of the patch, α is the normalisation factor (e.g. $\alpha=255$ for greyscale images), n_{p} is normal to the boundary of the target region and $\nabla I_{\mathrm{p}}^{\perp}$ is an isophote in location p (Fig. 15). The target region, boundary of the target region and source region are denoted as Ω, $\partial\Omega$ and Φ, respectively.

The patch $\Psi_{\hat{\mathrm{p}}}$ with maximum priority is found, i.e. $\Psi_{\hat{\mathrm{p}}}|\hat{p}=\arg\max_{\mathrm{p}\in\partial\Omega} P(p)$, and the best matching patch in source region with $\Psi_{\hat{\mathrm{p}}}$ is selected and copied into $\Psi_{\hat{\mathrm{p}}}$. At the end, the confidence terms for all pixels intersecting with the target region are updated. This algorithm is run iteratively until the missing region fades away.