VIDEO OBJECT INPAINTING USING POSTURE MAPPING

Chih-Hung Ling[1], Chia-Wen Lin[2], Chih-Wen Su[3], Hong-Yuan Mark Liao[4], and Yong-Sheng Chen[5]

[1,5]Department of Computer Science, National Chiao Tung University
[2]Department of Electrical Engineering, National Tsing Hua University
[3,4]Institute of Information Science, Academia Sinica
cwlin@ee.nthu.edu.tw

ABSTRACT

This paper presents a novel framework for object-based video inpainting. To complete an occluded object, our method first samples a 3-D volume of the video into directional spatio-temporal slices, and then performs patch-based image inpainting to repair the partially damaged object trajectories in the 2-D slices. The completed slices are subsequently combined to obtain a sequence of virtual contours of the damaged object. The virtual contours and a posture sequence retrieval technique are then used to retrieve the most similar sequence of object postures in the available non-occluded postures. Key-posture selection and indexing are performed to reduce the complexity of posture sequence retrieval. We also propose a synthetic posture generation scheme that enriches the collection of key-postures so as to reduce the effect of insufficient key-postures. Our experimental results demonstrate that the proposed method can maintain the spatial consistency and temporal motion continuity of an object simultaneously.

***Index Terms*—**video inpainitng, object completion, posture mapping, synthetic posture.

1. INTRODUCTION

Video inpainting [1-6] has attracted great attention in recent years due to its powerful capability in fixing/restoring damaged videos. In recent years, a number of methods have been proposed. These methods can be classified into two types: patch-based [1][2][4], and object-based [3][4]. In [1], Patwardhan *et al*. proposed a video inpainting technique which combines motion vector and image inpainting together to perform video inpainting. Three mosaics including the background, the foreground and the optical-flow, are constructed based on motion vector to provide information for video inpainting. This patch-based approach produces good visual effect for each frame, but it cannot maintain continuity along the temporal axis. Wexler *et al*. [2] used a fixed-sized cube as the unit of similarity measure. For each missing pixel, a set of constituent cubes are used to calculate the value of a missing pixel. To save computation time, a multi-scale approach is adopted. The process starts at the coarsest pyramid level and the solution is propagated to finer levels for further refinement. Although the result reported in [2] is good, only low resolution videos are shown and the multi-scale nature may cause over-smoothing artifacts and high computation complexity. Shen *et al*. [5] proposed to construct motion manifolds of the space-time volume. The constructed motion manifolds contain the entire trajectory of pixels. The motion manifold inpainting process proposed in [4] is rather computationally intensive as it adopts a complex image inpainting scheme, it would consume more computation time. Besides, Shen *et al.'s* approach would result in the problem of incomplete structure.

In addition to patch-based approach, object-based approach [4] [3] is also a video inpainting mechanism that needs to be mentioned. In [3], Cheung *et al.* proposed an efficient object-based video inpainting technique to deal with videos captured by a stationary camera. To inpaint foreground, they make use of all available object templates. For each missing object, a sliding window covering the missing object and its neighbor object templates is used to find a most similar sliding window. They then use this corresponding object template to replace the missing object. However, if the number of postures in database is not sufficient, it would influence the inpainting result. On the other hand, this method does not provide a systematic way to identify a good filling position of object template. In [4], Jia *et al.* propose a user-assisted video layer segmentation technique. Their method decomposes a target video into color and illumination videos. Then, a tensor voting technique is used to maintain consistency in both the spatio-temporal domain and the illumination domain. For an occluded object their method is able to reconstruct it through synthesizing other available objects. However, the synthesized object does not have a real trajectory. Besides, only textures are allowed in their background.

From the survey conducted above, we know that some approaches (e.g., [2]) can deal with spatial and temporal information simultaneously. However, they suffer from the over-smoothing artifact problem. Some patch-based approaches [1][4], on the other hand, do not generate over-smoothing artifact. But, it can only maintain either spatial consistency or temporal continuity. Besides, a patch-based approach very often generates inpainting error at foreground. Therefore, many researchers put their hope on object-based approach. Although an object-based approach has better chance to generate high-quality visual effect, it still has some difficult issues to be tackled with, for example: the unrealistic trajectory problem, and the inaccurate representation problem due to insufficient number of postures in the database.

2. OVERVIEW OF PROPOSED SCHEME

In this paper, we propose a new object-based video inpainting technique that can tackle simultaneously the problems of spatial consistency, temporal continuity, over-smoothing artifact, and insufficiency of available postures. Fig. 1 shows the flowchart of the proposed object completion scheme, which involves three steps: virtual contour construction, key posture-based posture sequence matching, and synthetic key posture generation. The first step of

object inpainting involves sampling a 3-D volume of video into directional spatio-temporal slices. Then a patch-based image inpainting scheme [8] is performed to complete the partially damaged object trajectories in the spatio-temporal slices. The completed spatio-temporal slices are then combined to form a sequence of virtual contours of the target object. Next, the derived virtual contours and a posture sequence matching technique are used to retrieve the most similar sequence of object postures from among the available non-occluded postures. The available postures are collected from the non-occlusion part of the input video. We perform key posture selection, indexing, and coding to convert the posture sequence retrieval problem into a substring search problem If a virtual contour cannot find a good match in the database of available postures, we construct synthetic postures by combining the constituent components of key postures to enrich the posture database so as to mitigate the problem of insufficient available postures. After retrieving the most similar posture sequence, the occluded objects are completed by replacing the damaged objects with the retrieved ones.

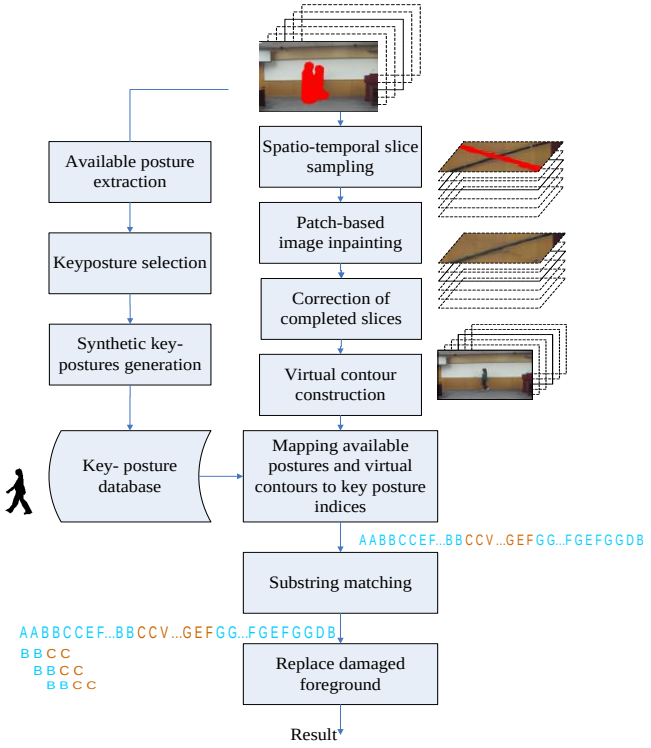


Fig. 1. Flowchart of the proposed object completion scheme.

3. Occluded Object Completion Using Posture Mapping

3.1. Virtual Contour Construction

To make an object completion process visually pleasing, it is important to extract, from a damaged object in consecutive frames, a set of features that not only represents the object's characteristics (e.g., shape, appearance, and posture) but also takes into account the temporal motion continuity. Here we propose the use of spatio-temporal slices to compose virtual object contours and then use them as the features to guide the object completion process.

After object extraction and removal, we sample a 3-D video volume which is composed of a number of contiguous frames. We then obtain a set of directional 2-D spatio-temporal slices which can fully capture an object's motion if the object only has horizontal motions. We assume during an occlusion period which is usually not long, the object's motion trajectory can be approximated by a line. Based on the assumption, those directionally sampled slices can capture the object's motion trajectory well. For simplicity, we only discuss the horizontal case. Under these circumstances, we need to deal with XT slices. As shown in Fig. 1, after removing the foreground object, object occlusions result in incomplete trajectories of an object in the XT slices. These missing regions need to be completed before composing a virtual contour. As long as the missing regions are completed properly, the reconstructed trajectories will be continuous, thereby preserving the temporal continuity of an object.

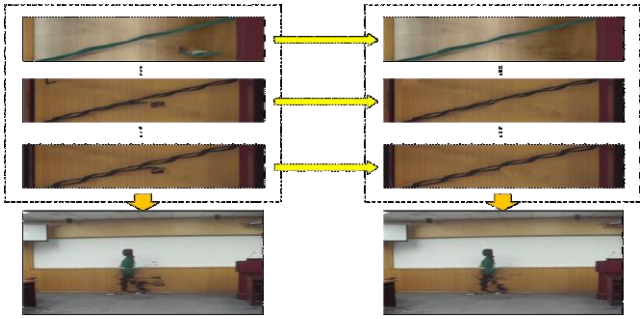
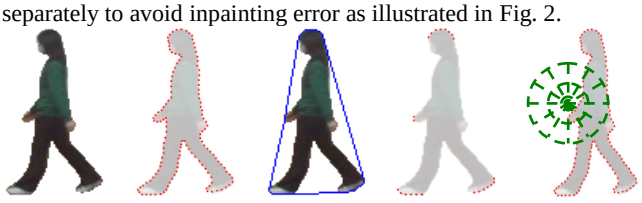


Fig. 2. Illustration of virtual contour construction by combining 2-D slices completed by image inpainting. The left-hand-side shows the virtual contour combined from completed slices without correction, and the righ-hand-side shows the corrected contour.

We use the patch-based image inpainting scheme proposed in [8] to complete the missing region and to obtain continuous object trajectories. This inpainting method first determines the filling order of those missing regions based on some confidence and data terms. According to the filling order, a missing region is filled with its most similar neighboring patches. After completing an XT slice, we use the Sobel edge detector to detect the boundary of object trajectory in the slice. These completed 2-D slices of a video frame are then combined back together to construct a virtual contour. This virtual contour is then used to guide the subsequent posture mapping process. Sometimes, image inpainting error leads to imprecise virtual contours, making it difficult to retrieve correct postures for object inpainting. We use an object tracking scheme as a post-processing step to correct the image inpainting error. Object tracking process is first performed to obtain rough object positions. Then, each spatio-temporal slice is divided into two regions: background and foreground trajectory. As a result, we apply image inpainting to the background and foreground trajectory regions separately to avoid inpainting error as illustrated in Fig. 2.



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

Fig. 3. Extracting the local context of a posture: (a) the original object posture; (b) the object's silhouette described by a set of feature points; (c) extracting significant feature points of the object silhouette using a convex hull surrounding the silhouette; (d) the resultant significant feature points of the object silhouette; and (e) the local histogram of a significant feature point.

3.2. Key-Posture Selection and Posture Mapping

After obtaining virtual contours, the contours are subsequently used for matching the most similar postures from the set of available postures to complete the occluded objects.

Our method first uses the key-posture selection method proposed in [6] to select the most representative postures out of the available postures. This method uses a set of feature points to describe an object's silhouette. A convex hull bounding the silhouette (see Fig. 3(c)) is then used to select a subset of feature points which are more important than the others as key feature points to describe the shape context of the object.

As illustrated in Fig. 3(b), we use the local histogram of a feature point to describe the object's local shape context. The similarity between the local contexts of two feature points is measured by the distance of their corresponding histograms by

$$H(\pi) = \sum_{j \in A} C\left(p_j, q_{\pi(j)}\right) \tag{1}$$

$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{\left[h_i(k) - h_j(k)\right]^2}{h_i(k) + h_j(k)} \tag{2}$$

where $H(\pi)$ represents the similarity between two posture silhouettes. $C(p_i,q_j)$ represents the similarity between two feature points p_i and q_j, and $h_i(k)$ and $h_j(k)$ denote the corresponding histograms of p_i and q_j, respectively.

A posture is selected as a key posture, if its degree of similarity with all key postures exceeds a predefined threshold, TH_p. After the key posture selection process, each key posture is labeled with a unique number. Each available posture and virtual contour are then matched with the key posture that has the most similar context, as defined in (1). If a virtual contour cannot be matched in this way, it is given a special label. As a result, a sequence of contiguous available postures and virtual contours can be converted into a string of key-posture labels based on the temporal order, as shown in Fig. 4.

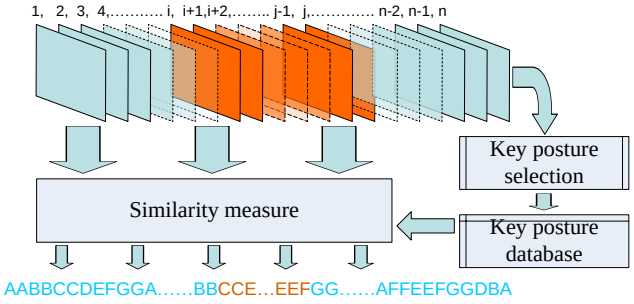


Fig. 4. Process for converting available postures and virtual contours into a sequence of key posture labels. The blue frames and numbers indicate the frames with available postures and their corresponding key-posture labels. The orange frames and numbers indicate the frames with constructed virtual contours and their corresponding key-posture labels.

After the encoding process, the problem of retrieving the most similar sequence of postures for a sequence of virtual contours becomes a substring matching problem that, given an input segment of codes, searches for the most similar substring in a long string of codes. The occluded objects are then replaced with the retrieved sequence of available postures. In Fig. 5, we use two examples to show how substring matching is applied to solve the posture mapping problem. In our example, it is clear that a no-match situation is solved using the proposed synthetic key-postures.

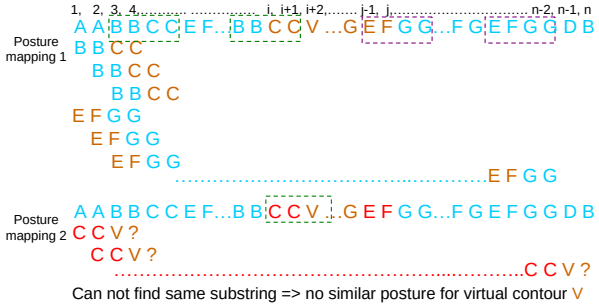


Fig. 5. Using substring matching to solve posture mapping problem. The substring length is four, blue number indicate posture number of available posture, brown number indicate posture number of virtual contour and red number indicate posture number of replaced posture for virtual contour.

3.3. Synthetic Postures Generation

The occlusion problem occurs in real-world applications all the time; hence, a virtual contour generated from an occlusion event may not find a good match among the selected key postures due to the lack of available non-occluded object postures. The problem of insufficient postures usually arises when the occlusion period for a to-be-completed object is long, resulting in many reconstructed virtual contours, or when the object's non-occlusion period is too short to collect a sufficiently rich set of non-occluded postures. Using a poorly matched posture to complete an occluded object can result in visually annoying artifacts. To resolve the problem where a virtual contour cannot find a good-match in the available key-posture database, we synthesize more postures by combining the constituent components of the available postures to enrich the content of the database. Fig. 6 shows how a new posture is synthesized by using three constituent components (the head, body, and legs) from different available postures selected by a skeleton matching process.

The above mentioned constituent components that can be used to synthesize a new database posture all come from the components of existing database postures. To use these components, we need to perform segmentation on those database postures in advance. Fig. 6(b) shows how this idea works. After performing the alignment postures, we compute the difference between every consecutive key posture pairs. From the distribution of the variances, one is able to identify what parts are moving most frequently. We then label these "frequently moving" components as the constituent components of a key posture synthesis process.

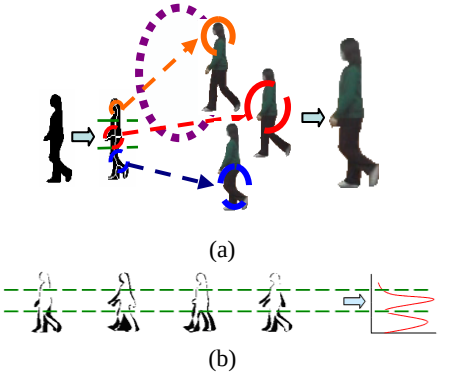
We use object skeletons to retrieve similar posture parts and then use the posture parts to synthesize new postures. We adopt the method proposed in [7] to calculate the object skeletons. In this method, a Euclidean distance map is used to derive object skeleton. The following similarity function, K, is used to measure the contribution of an arc to the shape.

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)} \tag{3}$$

where s_1 and s_2 stand for two line segments of object contour, respectively. $\beta(s_1,s_2)$ represents the turn angle at the common vertex of segments s_1 and s_2, and l represents the length function.

The similarity measure can be used to select arcs that make low contribution to an object shape. Shape reduction can be

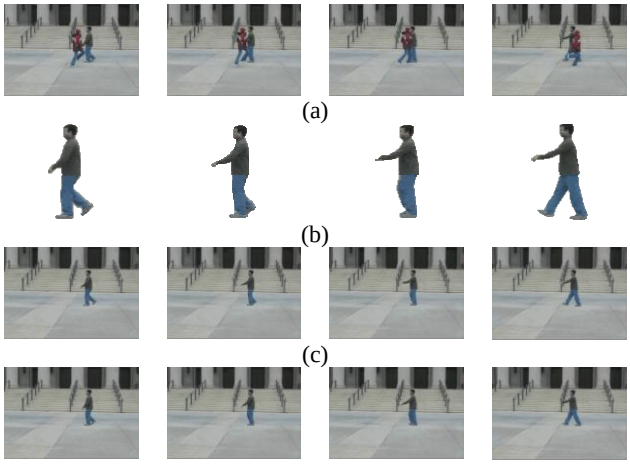
accomplished by removing those low contribution arcs. The reduced shape contour is then used to remove unimportant skeleton points. We use the thresholds used in the posture classification step to separate the skeletons of virtual contours and those of available postures. The skeleton parts of virtual contours are then used to find the corresponding skeleton parts of database postures. As a result, new postures can be synthesized by combining constituent parts of available posture according to the virtual contour skeletons.



(a)

(b)

Fig. 6. Illustration of synthesizing new postures using available postures. (a) A new posture is composed of constituent parts (e.g., head, body, and legs in this example) from different postures. (b) The parts of posture are partitioned by local variance extraction.



(a)

(b)

(c)

(d)

Fig. 7. Test sequence 4 taken from [1]: (a) some snapshots of the original video; (b) synthetic postures used to replace damaged foreground regions; (c) the completion result using the proposed method; and (d) the completed result reported in [1].

4. EXPERIMENTAL RESULTS

In our experiments, we used five test sequences to test the effectiveness of our method. Among these sequences, the first three were captured by a commercial digital camcorder with a frame rate of 30 fps, and a resolution of 352×240 (SIF). The remaining two test sets were taken, respectively, from [1] and [3]. Due to the page limit, we only show the results of one sequence. One can find the complete test results, including the original test videos, the videos after object removal, and the completed videos in our project website [9].

As shown in Fig. 7(a), we used this sequence to compare the performance of the proposed method with that of the method proposed in [1]. The inpainting result for test sequence 4 provided in [1] shows flickering artifacts at the boundaries of both the original and the inpainted objects due to transitions that are not smooth, even though the inpainted object looks good in individual frames. The result is reproduced on our project website [9]. In this test case, the number of available postures is insufficient. Because the method in [1] also uses available postures to complete damaged objects, insufficient available postures reduces the accuracy of inpainting. As mentioned earlier, the proposed synthetic posture generation scheme can maintain the motion continuity of the object effectively. Fig. 7(c) shows some of the additional synthetic postures compiled for sequence 4; and Fig. 7(d) and Fig. 7(e) show, respectively, some snapshots of the frames inpainted by the proposed method and the method in [1].

5. CONCLUSION

In this paper, we have proposed a novel method for completing an occluded object. The method involves three major steps: virtual contour construction, key posture-based sequence retrieval, and synthetic posture generation. We have proposed an efficient posture mapping method that uses key posture selection, indexing, and coding to convert the posture sequence retrieval problem into a substring matching problem. We have also developed a synthetic posture generation scheme that enhances the variety of postures available in the database. Our experimental results show that the proposed method generates completed objects with good subjective quality in terms of the objects' spatial consistency and temporal motion continuity.

ACKNOWLEDGEMENT

This work was supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under NSC Grants: NSC98-2631-001-013

REFERENCES

[1] K. A. Patwardhan, G. Sapiro, and M. Bertalmío, "Video inpainting under constrained camera motion," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 545–553, Feb. 2007

[2] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. PAMI*, vol. 29, no. 3, pp. 1–14, Mar. 2007

[3] S.-C. S. Cheung, J. Zhao and M. V. Venkatesh, "Efficient object-based video inpainting," *IEEE Conf. Image Process.*, pp. 705–708, 2006.

[4] J. Jia, Y.-W. Tai, T.-P. Wu, and C.-K. Tang, "Video repairing under variable illumination using cyclic motions," *IEEE Trans. PAMI.*, vol. 28, no. 5, pp. 832–839, May 2006.

[5] Y. Shen, F. Lu, X. Cao, and H. Foroosh, "Video completion for perspective camera under constrained motion," in *Proc. IEEE Conf. Pattern Recognit.*, pp. 63–66, 2006.

[6] Y.-M. Liang, S.-W. Shih, C.-C. A. Shih, H.-Y. M. Liao, and C.-C. Lin, "Learning atomic human actions using variable-length Markov models," *IEEE Trans. Syst., Man, Cybern. B, Cybern*. vol. 39, no. 1, pp. 268–280, 2009.

[7] X. Bai, L. J. Latecki, and W.-Y. Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," *IEEE Trans. PAMI*, vol. 29, no. 3, pp. 449–462, Mar. 2007.

[8] A. Criminisi, P Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol.13, no.9, pp. 1200–1212, Sept. 2004.

[9] *NTHU Video Inpainting project. [Online]. Available: http://www.ee.nthu.edu.tw/cwlin/inpainting/inpainting.htm*