VIDEO OBJECT INPAINTING USING MANIFOLD-BASED ACTION PREDICTION

Chih-Hung Ling[1], Yu-Ming Liang[2], Chia-Wen Lin[3], Yong-Sheng Chen[1], and Hong-Yuan Mark Liao[14]

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

ABSTRACT

This paper presents a novel scheme for object completion in a video. The framework includes three steps: posture synthesis, graphical model construction, and action prediction. In the very beginning, a posture synthesis method is adopted to enrich the number of postures. Then, all postures are used to build a graphical model of object action which can provide possible motion tendency. We define two constraints to confine the motion continuity property. With the two constraints, possible candidates between every two consecutive postures are significantly reduced. Finally, we apply the Markov Random Field model to perform global matching. The proposed approach can effectively maintain the temporal continuity of the reconstructed motion. The advantage of this action prediction strategy is that it can handle the cases such as non-periodic motion or complete occlusion.

Index Terms—video inpainting, object completion, action prediction, synthetic posture, motion animation.

1. INTRODUCTION

Automatic video inpainting is an important research area which has attracted great attention in recent years due to its powerful ability to fix/restore damaged videos and the flexibility it offers for editing home videos. A number of algorithms for automatic video inpainting have been proposed in the past few years [1–6]. In video painting, an important problem is to complete a partially or even totally occluded object in a video. Several schemes have been proposed to address the object inpainting problem based on available object templates [3–5] or on recovering the missing manifold trajectory via nonlinear dimension reduction [6].

As to the category of template-based video inpainting, Cheung *et al*. [7] proposed an efficient template-based video inpainting technique for dealing with videos recorded by a stationary camera. To inpaint the foreground, they utilize all available object templates. For each missing object, a fix-sized sliding window that covers a missing object and its neighboring templates is used to find the most similar object template. The drawback of this approach is that if the number of postures in the database is not sufficient, the inpainting result could be unsatisfactory. Moreover, the method does not provide a systematic way to identify a good filling position for an object template. An inappropriately chosen position may cause visually annoying artifacts. In [4], Jia *et al*. proposed a user-assisted video layer segmentation technique that decomposes a target video into color and illumination videos. A tensor voting technique is used to maintain consistency in both the spatio-temporal domain and the illumination domain. The method reconstructs an occluded object by synthesizing other available objects, but the synthesized object does not have a real trajectory and only textures are allowed in the background.

Recently, a manifold learning based approach was proposed by Ding *et al*. [6] to perform video inpainting. They made use of Local Linear Embeddings (LLE) to transform observed data in frames to the embedded features in low dimension manifold. Then, the embedded features were reordered to obtain a Hankel matrix and the embedded features of missing data can be obtained by minimizing the rank of the Hankel matrix. Finally, the Radial Basis Function (RBF) is used for inverse mapping. Although the consecutive poses of an object with regular and cyclic motions can be well represented by a low-dimensional manifold embedded in a high-dimensional visual space, poses with non-regular motions (e.g., transitions in two different types of motions) are usually not the case. As a result, mapping reconstructing a high-dimensional video object with irregular or non-cyclic motions from the object's low-dimensional manifold approximation usually leads to annoying artifacts (e.g., ghost images).

As mentioned above, most of the existing object inpainting algorithms to some extent generate artifacts if an object is completely occluded or its corresponding motion is not periodic. To avoid the difficulties, we propose an action prediction method for object inpainting in this paper. The framework is composed of three steps: posture synthesis, graphical model construction, and action prediction. In the very beginning, a posture synthesis method is adopted to enrich the number of postures. Then, the generated postures are used to build a graphical model of object action which can provide possible motion tendency. We define two constraints to confine the motion continuity property. One is to set a threshold for providing the maximum search distance if a trajectory in the constructed graphical model is discontinuous. The other constraint is to constrain the motion tendency. With the above two constraints, possible candidates between every two consecutive postures are significantly reduced. Finally, we apply the Markov Random Field model to perform global matching. A potential trajectory that receives the maximum total probability will be identified as the final result. The proposed action prediction model can help identify a set of suitable postures from posture database to restore those damaged/missing postures. The proposed approach can effectively maintain the temporal continuity of the reconstructed motion. The advantage of this action prediction strategy is that it can handle the cases such as non-periodic motion or complete occlusion. These capabilities are powerful because

conventional model-based action prediction methods [7] need a training process to achieve the same goal.

2. OBJECT INPAINTING USING ACTION PREDICTION METHOD

A. Posture synthesis

The problem of insufficient posture number would affect the visual quality of any video sequence generated by an action prediction-based approach. To solve the short-of-postures problem, we use our previous posture synthesis method [5] to enrich the number of postures. The main concept of a posture creation process is to combine the constituent parts of different available postures to enrich the contents of the posture database. Therefore, the first process is to perform appropriate segmentation on the postures in the database. To do a better posture segmentation job, we need to know the amount and speed that each component of a posture moves. For a component that moves significantly and faster, we need to take more intermediate postures to interpolate the gap generated by missing frames. Taking any two postures from the posture database, we use a bounding rectangle to bound each posture first. Then, we align these two bounding rectangles (including orientation and scale) as indicated in the middle part of Fig. 1. Then, we take the difference between these two postures and project these differences onto the y-axis as indicated at the right side of Fig. 1. To detect which parts of a human body move significantly and speed, one has to calculate the differences between a posture and all other database postures. These posture differences are all projected onto the y-axis and the accumulated y-axis component will be like the distribution shown at the right hand side of Fig. 2. From the peaks and valleys of the projected distribution, one can segment properly a posture as indicated by the posture sequence shown in Fig. 3. From the segmented components of a posture, new postures can be synthesized by combining constituent components as shown in Fig. 4.

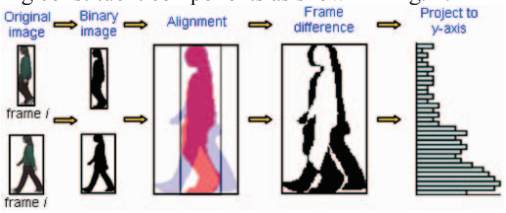


Fig. 1. Project posture differences onto the y-axis.

Fig. 2. Project all the differences between any two postures onto the y-axis.

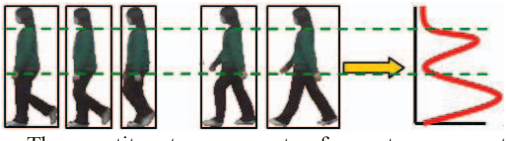


Fig. 3. The constituent components of a posture are partitioned based on local variance extraction.

B. Graphical model construction

After synthetic posture creation, the posture database will have much more number of postures. These postures can be used to build the graphical model (as shown in Fig. 5) of an object action. A graphical model provides a simple representation of an object action. To obtain the graphical model of an object action, we project all postures (including synthetic and existing postures) onto a feature space. Then, we link those postures that appear in adjacent frames in the constructed feature space. After applying the above procedure, we can obtain a graphical representation of an object action. To model the distribution of postures in the feature space, we need to know the distances between distinct postures. We use the shape context descriptor [9] to make a detailed description of a posture. We calculate the value of shape context along the silhouette of a posture. Later these shape contexts will be used to compare the degree of similarity between two distinct postures.

Fig. 4. A new posture is composed of three components (head, body, and legs).

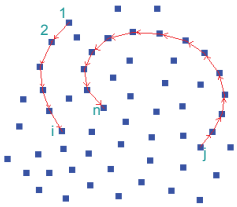


Fig. 5. The graphical model of an object action in low dimensional manifold.

(a) (b) (c)

Fig. 6. Extracting the local context of a posture: (a) the object's original posture; (b) the object's silhouette described by a set of feature points; and (c) a shape context mask on a feature point.

To calculate the shape context, the silhouette of a posture needs to be represented as a set of sampled points $P=\{p_1,p_2,...p_n\}$ (as indicated in Fig. 6(b)). For each sampled point $p_i \in P$, a corresponding local histogram is computed in a log-polar space (as indicated in Fig. 6(c)) to represent the local shape context of p_i. The cost of matching two different sampled points which belong to two different postures can be defined as follows

$$D(p_i,q_j)=\frac{1}{2}\sum_{k=1}^{K}\frac{\left[h_{p_i}(k)-h_{q_j}(k)\right]^2}{h_{p_i}(k)+h_{q_j}(k)}, \quad (1)$$

where $h_{p_i}(k)$ and $h_{q_j}(k)$ denote the k-th bin of the two sampled points p_i and q_j, respectively. The best match between two different postures can be accomplished by minimizing the following total matching cost:

$$H(\pi) = \sum_j D\left(p_j, q_{\pi(j)}\right) \quad (2)$$

where π is a permutation of 1, 2, …, n. Due to the constraint of one-to-one matching, shape matching can be considered as an assignment problem that can be solved by a bipartite graph matching method. Therefore, the shape context distance between two shapes P and Q can be computed as follows

$$D_{sc}(P,Q) = \frac{1}{n}\sum_i C(p_i, q_{\pi(i)}) + \frac{1}{m}\sum_j C(p_j, q_{\pi(j)}), \quad (3)$$

where n and m are the number of sample points on the shape P and Q, respectively.

Using the shape context descriptor, we can calculate the degree of similarity between two distinct postures. Based on these similarity measures between postures, we can cluster the database postures. We make use of a nonlinear dimension reduction method, ISOMAP, to perform clustering on postures. In our application, existing/synthetic postures are regarded as input data points of ISOMAP and the distances between data points are the same as the similarity values between postures.

C. Action prediction

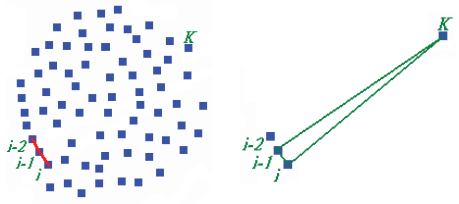
Based on the graphical model of an object action, we can find suitable postures to replace damaged/missing posture by finding an approximate path that can link data points x_i and x_j in the low dimension manifold. Intuitively, the reconstruction of a motion path can be accomplished by taking the shortest path between two nodes. We define two constraints to regulate the manner of the search process. The first constraint is to limit the search range to stay within a reasonable neighborhood. For all the data points on a trajectory, we compute all the distances between any two consecutive data points. The distance between any two consecutive data points on a trajectory can be determined by calculating the shape context difference between the two corresponding posture. Among the computed distances mentioned above, the maximum distance will be chosen as the search range for executing the first constraint. Therefore, the radius which defines the circular search range can be determined as follows:

$$r = \max_{\forall e_{ij} \text{ on a complete trajectory}} e_{ij}, \quad (4)$$

where e_{ij} represents the distance between two consecutive points x_i and x_j on the trajectory of an object action.

The second constraint can be applied to maintain the tendency of object motion in each local region. It can be realized by checking the motion trajectory tendency in a graphical model. In a low dimensional manifold, a motion trajectory does not change direction significantly in a neighborhood region. Based on this observation, we define a variance constraint of motion tendency to limit the variance of motion tendency in each neighborhood region. Fig. 7 illustrates an example of motion tendency constraint. Fig. 7(a) shows three consecutive data points x_{i-2}, x_{i-1}, and x_i forming a motion trajectory. x_k is a point which is far away from the above three points. x_{i-1}, x_i, and x_k can be connected to form a triangle (Fig. 7(b)).

From the basic knowledge of triangulation, if the data point x_k is very far away from both x_{i-1} and x_i, the distances to these two points, $\overline{x_i x_k}$ and $\overline{x_{i-1} x_k}$, will be close to each other. As we have mentioned, the motion tendency cannot change abruptly between two consecutive postures. This constraint can be defined as follows. For a random starting point x_s, we select G data points which are far away from x_s. Among these G data points, if any $x_k \in G$ satisfies $d_{s,k}/r > 15$, then it is chosen because it passes the motion tendency test. Here, r is the maximum distance between adjacent postures defined in (4) and $d_{s,k}$ is the distance between x_s and x_k. With the above criterion, we calculate the distance between x_s and each of the G chosen points. Therefore, we can obtain in total G distances. These G distances form a histogram to associate with point x_s. For a candidate point x_c which is nearby x_s, we can also form a similar histogram to associate with it using the same process. A candidate point x_c can maintain the motion tendency only if the value of each bin in its associated histogram is close to the value of each bin in the associated histogram of x_s.



(a) (b)

Fig. 7. An example of motion tendency constraint (a) three consecutive data points x_{i-2}, x_{i-1}, and x_i form a motion trajectory, and x_k is a point far away from the above three points; (b) data points x_{i-1}, x_i, and x_k form a triangle.

The above process is able to keep local motion continuity. For maintaining global motion continuity of an object action, we propose a two-way prediction mechanism based on the theory of Markov random field. We use three time instants t–1, t, and t+1 to explain how the proposed mechanism operates. The forward direction operation proceeds as follows. At time *t-1*, we make forward prediction on each data point. The motion tendency constraint and the search range constraint are applied to determine m probable data points at next time state t. These m selected data points will be used further to predict the candidate data points at time t+1. Following the same strategy, we do similar processing in reverse direction and collect related information from t+1 to t, and then from t to t–1. With the results collected from the bi-directional processing, we combine them and form final ranking for the time t. A probability value associated with each candidate data point is obtained by the bi-directional voting process.

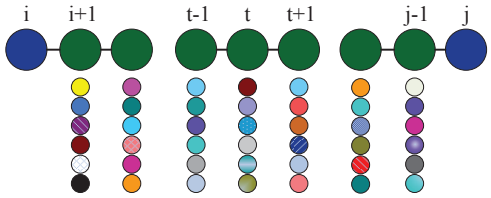


Fig. 8. The Markov network is used to build the relation between each local region.

Since the above mentioned motion continuity constraint only works on local region, we use the Markov Random Field approach to achieve global motion continuity. To predict an object action, we make the following Markov assumption: assign one node of a Markov network to every time state as shown in Fig. 8. A constructed Markov network can reflect statistical dependencies. Given a set of data points located at intervening nodes, two nodes of a Markov network are statistically independent. Since our Markov network contains no loops, the above defined Markov assumption results in simple "message-passing" rules for

computing the probability during inference. The data point estimated at node *j* is

$$\hat{c}_j = \arg \max_{c_j} p(c_j) M_j^{j-1} M_j^{j+1}, \quad (5)$$

where c_j is the candidate point associated with node *j*, $p(c_j)$ is the self probability of candidate point c_j, and M_j^{j+1} is the message from node *j-1* to node *j*. M_j^{j+1} can be calculated as follows:

$$M_j^{j+1} = \max_{[c_k]} \Psi(c_j, c_{j+1}, c_{j+2}) p(c_{j+1}) \tilde{M}_{j+1}^{j} \tilde{M}_{j+1}^{j+2}, \quad (6)$$

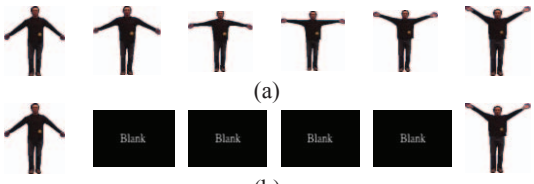
where $\tilde{M}_{j+1}^{j}$ is the previous message that can be used to generate M_j^{j+1} through executing Eg.(7). M_j^{j+1} includes the probability information of all candidate data points of node *k*. The initial $\tilde{M}_{j+1}^{j}$ is set as a column vector with all 1s. The function $\Psi(c_j, c_{j+1}, c_{j+2})$ is defined as follows:

$$\Psi(c_j, c_{j+1}, c_{j+2}) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(\theta - u)^2}{2\sigma^2}) \quad (8)$$

where θ is the angle between line $\overline{c_j c_{j+1}}$ and $\overline{c_{j+1} c_{j+2}}$, u and σ are the mean and variance of all angles in a complete trajectory of an object action

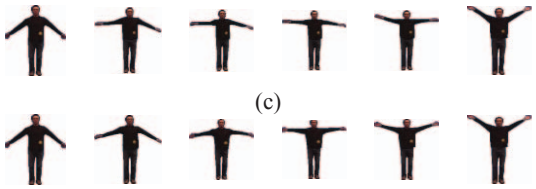
3. EXPERIMENT RESULT

To test the effectiveness of the proposed action prediction method, we used several test sequences to evaluate the efficacy of the proposed method. However, we only use one sequence to demonstrate the power of our approach. The sequence was captured by a commercial digital camcorder with a frame rate of 30 fps, and a resolution of 352×240 (SIF). In the experiments, we first removed several consecutive frames to simulate a real-world situation in which objects in a number of consecutive frames are damaged due to packet loss during transmission of the video or due to a damaged hardware component. We applied the proposed action prediction method to reconstruct object actions. Besides, we also made a comparison between Xu *et al.*'s approach [8] and ours. For test sequence, the proposed method could keep the motion continuity of a reconstructed action and provided better result than Xu *et al.*'s approach. Fig. 9(a) shows some snapshots of the test sequence #1 and the experiment results of Xu *et al.*'s approach [8] and ours are shown in Fig. 9(c) and Fig. 9(d), respectively. According to the experiment result, it could be observed that the proposed method can maintain continuity on an action and provided better result than the result generated by applying Xu *et al.*'s approach. Compared with original the video, the reconstructed object action using our method is close to each other. Therefore, the proposed action prediction method is suitable for object inpainting which can better recover an object action and maintain motion continuity simultaneously.



(a)

(b)



(c)

(d)

Fig. 9. The experiments on test sequence #1; (a) original video frames; (b) remove several consecutive frames (c) the result of [8]; and (d) the result obtained by applying the proposed method

4. CONCLUSION

In this paper, we proposed a novel framework for object inpainting. The proposed method consists of three steps: posture synthesis, graphical model construction, and action prediction. The advantage of this action prediction strategy is that it can handle the cases such as non-periodic motion or complete occlusion. Our experimental results also show that the proposed method can keep the reconstructed motion look continuous.

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: NSC99-2631-H-001-020.

5. 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. Pattern Anal. Match. Intell*., 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," in *Proc. IEEE Conf. Image Process*., Atlanta, GA, pp. 705–708, Oct. 2006.

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

[5] C.-H. Ling, C.-W. Lin, C.-W. Su, H.-Y. Mark Liao, and Y.-S. Chen, "Video object inpainting using posture mapping," *IEEE Conf. Image Process.*, Cairo, Egypt, Nov. 2009.

[6] T. Ding, M. Sznaier, and O. I. Camps, "A rank minimization approach to video inpainting," in *Proc. IEEE Conf. Comput. Vis*., Rio de Janeiro, Brazil, pp. 1–8, Oct. 2007.

[7] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognit.*, vol. 36, no. 3, pp. 585–601, Mar. 2003.

[8] X. Xu, L. Wan, X. Liu, T.-T. Wong, L. S. Wang, C.-S. Leung, "Animating animal motion from still," *ACM Trans. Graphics*, vol. 27, no. 5, Dec. 2008.

[9] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.