Virtual Contour Guided Video Object Inpainting Using Posture Mapping and Retrieval

Chih-Hung Ling, Chia-Wen Lin, *Senior Member, IEEE*, Chih-Wen Su, Yong-Sheng Chen, *Member, IEEE*, and Hong-Yuan Mark Liao, *Senior Member, IEEE*

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

***Index Terms*—Object completion, posture mapping, posture sequence retrieval, synthetic posture, video inpainting.**

I. Introduction

VIDEO inpainting [1]–[11] has attracted a great deal of attention in recent years because of its powerful ability to fix/restore damaged videos and the flexibility it provides for editing home videos. It also ensures visual privacy in security [12] applications. More specifically, inpainting techniques have been used extensively for fixing/restoring damaged digital images [13]–[18]. Depending on how they restore damaged images, the techniques can be categorized into three groups: texture synthesis-based methods [13], [14], partial difference equation-based (PDE-based) methods [15], and patch-based methods [16]. The concept of texture synthesis is borrowed from computer graphics. Its main purpose is to insert a chosen input texture into a damaged/missing region. In contrast, PDE-based approaches propagate information from the boundary of a missing region toward the center of that region. They are suitable for completing a damaged image in which thin regions are missing. Texture synthesis and PDE-based propagation cannot handle cases of general image inpainting because the former does not consider structural information and the latter frequently introduces blurring artifacts. A patch-based approach [16], on the other hand, is much more suitable for image inpainting because it can produce high-quality visual effects and maintain the consistency of local structures. Because of the success of patch-based image inpainting, researchers have applied a similar concept in video inpainting [3]; however, the issues that need to be addressed in video inpainting are much more challenging.

Although video inpainting is a relatively new research area, a number of methods have been proposed in recent years. Generally, the methods can be classified into two types: patch-based methods [1]–[6], and object-based methods [7], [8]. As the patch-based approach has been successfully applied in image inpainting [16], researchers have extended a similar concept to video inpainting. For example, video inpainting under constrained camera motion [1] and under space-time completion [3] can be regarded as extensions of the nonparametric sampling technique developed by Efros and Leung [13]. In [1], Patwardhan *et al.* propose a video inpainting technique that combines motion information and image inpainting. Like most existing methods, it is assumed that the camera's movements are constrained in some directions. In the preprocessing step, three mosaics, i.e., the background, the foreground and the optical-flow, are constructed to provide information for video inpainting. Each missing region in a frame has a corresponding missing region in the foreground or background mosaic. The candidate patch in the foreground mosaic that is most similar to the missing region in the frame is used to fill the missing region. For background inpainting, the image inpainting method proposed in [16] is adopted to fill the missing regions of the background mosaic. Although the approach in [1] produces a good visual effect for each frame, it cannot maintain continuity along the temporal axis. The lack of temporal continuity leads to flickering artifacts. Wexler *et al.* [3] use a fixed-size cube with three dimensions as the unit of the similarity measure function. A set of constituent cubes are used to calculate the value of a missing pixel. Based on the similarity measure function, which is the sum of squared differences (SSD), each cube finds the most similar candidate cube. Although the results reported in [3] are good, only low resolution videos

Manuscript received May 28, 2010; revised October 02, 2010; accepted November 09, 2010. Date of publication November 29, 2010; date of current version March 18, 2011. This work was supported in part by the Research of Techniques and Tools for Digital Archives program sponsored by the National Science Council (NSC), Taiwan, R.O.C., under grant NSC99-2631-H-001-020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Nadia Magnenat-Thalmann.

C.-H. Ling and Y.-S. Chen are with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.

C.-W. Lin is with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan.

C.-W. Su is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan.

H.-Y. M. Liao is with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, and also with the Institute of Information Science, Academia Sinica, Taipei, Taiwan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2010.2095000

are considered and the multi-scale nature of the solution may cause blurred results due to sampling and smoothing and high computational complexity. In [4], Cheung *et al.* propose a patch-based probability model for video inpainting. It is suitable for synthesizing data that does not contain structured information, but the reported inpainting results are of low resolution and contain over-smoothing artifacts. Shen *et al.* [6] propose constructing motion manifolds of the space-time volume. The manifolds contain the entire trajectory of each pixel, and the strategy proposed by Sun *et al.* [17] is used to inpaint missing regions. In [17], a user only needs to draw one structure line to perform inpainting; however, in [6], two lines must be drawn because both the foreground and the background must be considered. The inpainting process proposed in [17] is computationally expensive, but the process proposed in [6] is even more complicated and thus requires more time. Shen *et al.*'s approach uses two-dimensional patches (one for the spatial dimension and one for the temporal dimension). As a result, if a patch contains both spatial and temporal dimensions, they cannot be handled smoothly at the same time, which results in either motion discontinuity or an incomplete structure.

Object-based approaches, such as [7] and [8], also employ a video inpainting mechanism. In [8], Cheung *et al.* propose an efficient object-based video inpainting technique for dealing with videos recorded by a stationary camera. To inpaint the background, they use the background pixels that are most compatible with the current frame to fill a missing region; and to inpaint the foreground, they utilize all available object templates. A fixed-size sliding window is defined to include a set of continuous object templates. The authors also propose a similarity function that measures the similarity between two sets of continuous object templates. For each missing object, a sliding window that covers the missing object and its neighboring objects' templates is used to find the most similar object template. The corresponding object template is then used to replace the missing object. However, 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. This may cause visually annoying artifacts if the chosen position is inappropriate. In [7], Jia *et al.* propose a user-assisted video layer segmentation technique that 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. 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.

Patch-based methods often have difficulty handling spatial consistency and temporal continuity problems. For example, the approaches proposed in [1] and [6] can only maintain spatial consistency or temporal continuity; they cannot solve both problems simultaneously. On the other hand, the approaches proposed in [3] and [4] can deal with spatial and temporal information simultaneously, but they suffer from the over-smoothing artifacts problem. In addition, patch-based approaches often generate inpainting errors in the foreground. As a result, many researchers have focused on object-based approaches, which usually generate high-quality visual results. Even so, some difficult issues still need to be addressed; for example, the unrealistic trajectory problem and the inaccurate representation problem caused by an insufficient number of postures in the database.

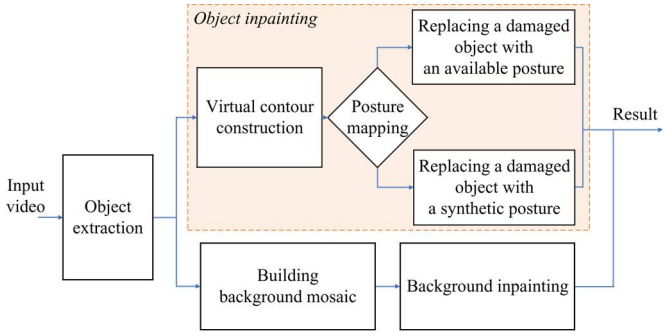


Fig. 1. Simplified flowchart of the proposed video inpainting scheme.

In this paper, we propose an object-based video inpainting scheme that can solve the spatial inconsistency problem and the temporal continuity problem simultaneously. The scheme is comprised of three steps: virtual contour construction, key-posture selection and mapping, and synthetic posture generation. The contribution of this work is three-fold. First, we propose a scheme that is able to derive the virtual contour of an occluded object. The contour provides a fairly precise initial estimate of the posture and filling location of the occluded object, even if the object is completely occluded. Therefore, the virtual contour is suitable for finding a good replacement for the occluded object from the available postures in the input video. Second, we propose a key posture-based mapping scheme that converts the posture sequence retrieval problem into a substring matching problem, thereby reducing the computational complexity significantly, while maintaining the matching accuracy. Since the occluded objects are completed for a whole subsequence rather than for individual frames, the temporal continuity of object motion is maintained as well. Third, for a sequence in which we cannot find a sufficiently rich set of available postures for completing occluded postures, our proposed synthetic posture generation scheme can effectively enrich the database of postures by combining the constituent parts of different available postures. As a result, improved inpainting performance is achieved.

The remainder of this paper is organized as follows. Section II provides an overview of the proposed video completion scheme. In Section III, we present the proposed posture-based, inpainting scheme for occluded objects; and in Section IV, we evaluate the scheme's performance. Section V contains some concluding remarks.

II. Overview of the Proposed Scheme

We propose an object-based video inpainting scheme that can maintain the spatial consistency and temporal motion continuity of an object simultaneously. The scheme can also handle the problem of insufficiency of available postures. Fig. 1 shows a block diagram of the proposed scheme. Initially, we assume that the objects to be removed and the occluded objects to be restored have been extracted by an automatic object segmentation scheme [19], or by an interactive extraction scheme [20]–[22].

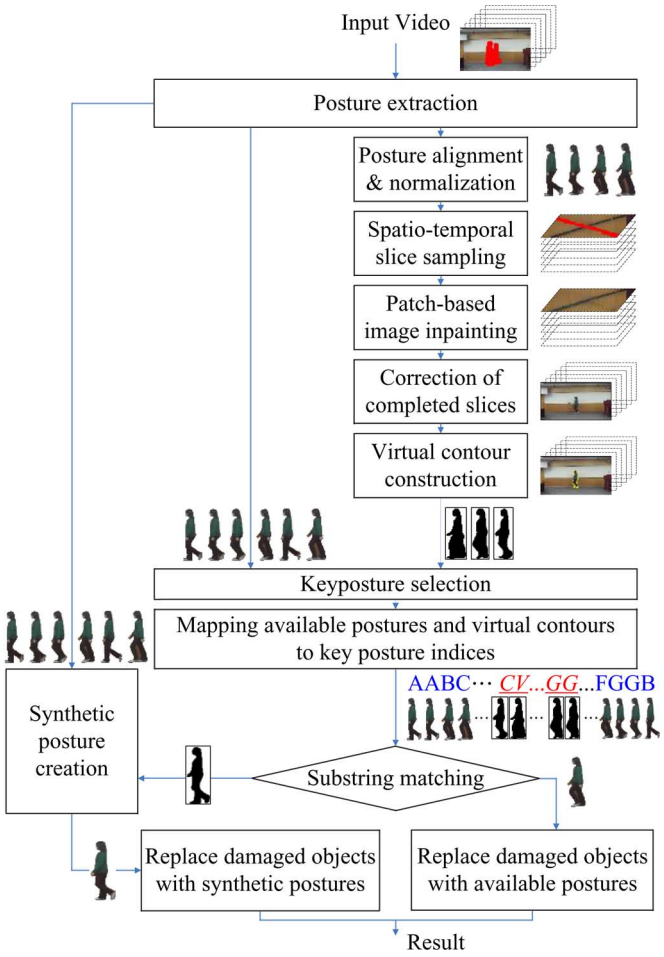


Fig. 2. Flowchart of the proposed object completion scheme.

After object extraction, the occluded objects and the background are completed separately. We also assume that the trajectory of each occluded object can be approximated by a linear line segment during the period of occlusion. This assumption is reasonable for many practical applications because the duration of an occlusion is typically short, and an object does not usually perform complex motions during such a short period.

Our primary goal is to solve the problem of completing partially or totally occluded objects in a video. Fig. 2 shows the flowchart of the proposed object completion scheme which is comprised of 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 (exemplar-based) image inpainting [16] operation is performed to complete the partially damaged object trajectories in the 2-D spatio-temporal slices. The objective is to maintain the trajectories' temporal continuity. The completed spatio-temporal slices are then combined to form a sequence of virtual contours of the target object to infer the missing part of the object's posture [29]. 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-occluded part of the input

Fig. 3. 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; (c) the local histogram of a significant feature point where the numbers of feature points contained in some bins are shown; (d) extracting significant feature points of the object's silhouette using a convex hull surrounding the silhouette; and (e) the resultant significant feature points of the object's silhouette.

video. We perform key posture selection, indexing, and coding operations to convert the posture sequence retrieval problem into a substring search problem, which can be solved efficiently by existing substring-matching algorithms [23]. 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. This process mitigates 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.

For background inpainting, we follow the background mosaics method proposed in [1]. The method first constructs a background mosaic for each video shot based on global motion estimation (GME), and then finds the corresponding available data in the background mosaic for each pixel in a missing region. The data are used to fill the missing regions and thereby achieve spatio-temporal consistency in the completed background. Since background inpainting is not the focus of this paper, we do not consider its implementation in detail.

III. Occluded Object Completion Using Posture Sequence Matching

A. Shape Context Descriptor

Before discussing the proposed method in detail, we describe the shape context descriptor in [23] and [24], which we use for posture alignment/normalization and key posture selection. The descriptor is invariant to translation, scaling, and rotation; and it is even robust against small amounts of geometrical distortion, occlusion, and outliers. As shown in Fig. 3, given an object image [Fig. 3(a)], the descriptor selects a set of feature points to describe the object's silhouette [Fig. 3(b)]. The object's local shape context is described by the local histograms of the regions centered at the feature points. Under this method, for each feature point, a circle with radius r [Fig. 3(c)] is used to find the local histogram. The circle is then divided into N_{bin} partitions and the number of feature points in each partition is calculated, resulting in a histogram with N_{bin} bins. The value of N_{bin} is empirically set to be 60 for all sequences. The cost of matching two different sampled points which belong to two different postures can be defined as follows:

$$F(a_i, c_j) = \frac{1}{2}\sum_{k=1}^{N_{\text{bin}}} \frac{\left[h_{a_i}(k) - h_{c_j}(k)\right]^2}{h_{a_i}(k) + h_{c_j}(k)} \quad (1)$$

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

$$H(\pi) = \sum_j F\left(a_j, c_{\pi(j)}\right) \tag{2}$$

where π is a permutation of $1, 2, \ldots, N_{\text{bin}}$. 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 A and C can be computed as follows:

$$F_{sc}(A, C) = \frac{1}{N_A}\sum_i F\left(a_i, c_{\pi(i)}\right) + \frac{1}{N_C}\sum_j F\left(a_j, c_{\pi(j)}\right) \tag{3}$$

where N_A and N_C are the number of sample points on the shape A and C, respectively.

B. Virtual Contour Construction Using Spatio-Temporal Slices

The main difficulty in completing a damaged video object is that the information left in a badly damaged object is usually insufficient to reconstruct the object properly by using spatio-temporal clues. Furthermore, completing an object frame-by-frame often causes temporal discontinuity in the object's appearance and motion, since a frame-wise completion process does not consider an object's temporal dependency in consecutive frames. Such temporal discontinuity results in visually annoying artifacts like flickering and jerkiness. To ensure that a completed object is visually pleasing, it is important to extract a set of features from a damaged object in a number of consecutive frames. As a result, the features not only represent the object's characteristics (e.g., motion, appearance, and posture), but also take its temporal continuity into account.

Manifold learning-based methods [10], [25] have been proposed to recover the damaged/missing poses of an occluded object. Although the consecutive poses of an object with regular and cyclic motion can be well represented by a low-dimensional manifold embedded in a high-dimensional visual space, poses with nonregular motions (e.g., transitions in two types of motions) are usually not the case. As a result, mapping reconstructing a high-dimensional video object with irregular or noncyclic motion from the object's low-dimensional manifold approximation usually leads to annoying artifacts (e.g., ghost images).

As mentioned earlier, we use spatio-temporal slices of a video to derive virtual object contours, which are then used as features to infer the occluded object poses. More specifically, after object extraction and removal, we sample a 3-D video volume comprised of several consecutive frames to obtain a set of directional 2-D spatio-temporal slices, as shown in Fig. 4. For example, if a 3-D video volume [Fig. 4(a)] is sampled at different Y values [Fig. 4(b)], each resulting XT slice represents the horizontal trajectory of an object over time. The trajectory can fully capture an object's motion if it only has horizontal motions. Other directional sampling schemes can be used to deal with objects that have different motion directions. Note that a nonpure horizontal motion will cause an object's size to vary over time due to the zoom-in/zoom-out effect, as shown in Fig. 4(c). In this case, posture alignment and normalization can be used to avoid the inference of different posture scales. Without loss of generality, we use the largest posture of an object as a reference for aligning and normalizing the other postures. First, we establish the correspondence between the contour points of every two adjacent postures by shape matching [23] and [24]. The affine transformation parameters between the largest posture and the others can then be estimated from the corresponding points using the least squares optimization method. As a result, all postures are aligned and normalized with the largest posture via the affine transformations. As shown in Fig. 4(d), after removing the foreground object and posture alignment, object occlusion results in incomplete trajectories of the object in the spatio-temporal slices. The missing regions of object trajectories in the 2-D spatio-temporal slices must be completed using an image inpainting method before composing a virtual contour. Because an object's occlusion period is usually short, we assume that the occluded part of a motion trajectory in a 2-D slice can be approximated by a line. Based on this assumption, the occluded part in each directionally sampled slices can be inpainted well. Since the trajectory of an object on each 2-D slice records the locations of the same part of object over time, as long as the missing regions of trajectories are completed properly, the reconstructed trajectories will be continuous, thereby preserving the temporal continuity of an object.

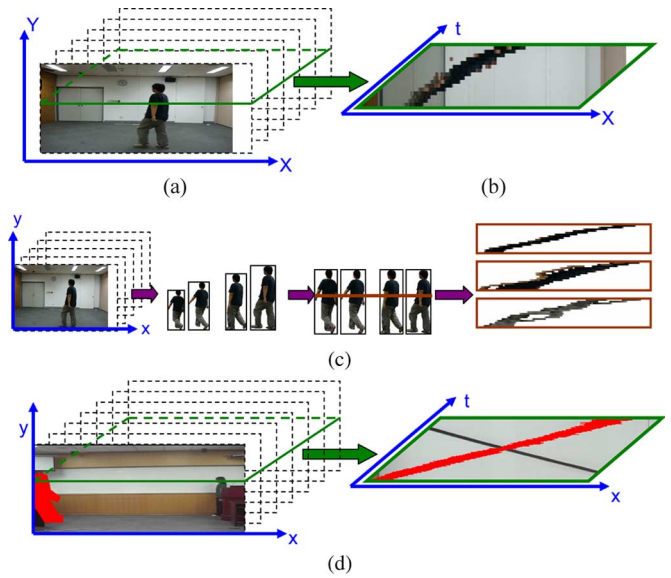


Fig. 4. Sampling a 3-D video volume comprised of several consecutive frames. (a) Original frame. (b) Object trajectory on a sampled XT plane s, indicated by the green lines in (a). (c) 2-D spatio-temporal slices sampled on a video shot, where the object's size varies due to nonpure horizontal motion. (d) Removed occluded object trajectories on the XT plane sampled on the 2-D plane.

To obtain continuous object trajectories, we use the patch-based image inpainting scheme proposed in [16] to complete missing regions in the spatio-temporal slices. The method first determines the filling order of the missing regions based on the confidence term and data term as follows:

$$P(p) = C(p) \cdot D(p) \tag{4}$$

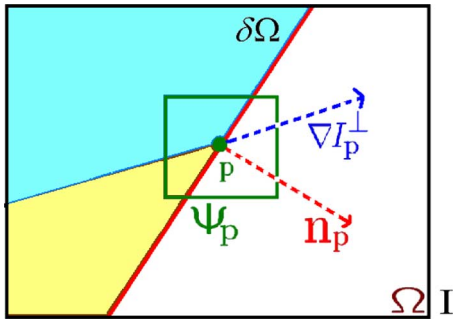


Fig. 5. Notations used for the data and confidence terms in patch-based image inpainting [14].

where $P(p)$ represents the priority of a missing region p; and $C(p)$ and $D(p)$ denote the confidence term and the data term expressed in (5) and (6), respectively:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (\mathrm{I}-\Omega)} C(q)}{|\Psi_p|} \tag{5}$$

$$D(p) = \frac{\left|\nabla I_p^{\perp} \cdot n_p\right|}{\alpha} \tag{6}$$

where $|\Psi_p|$ represents the area of region Ψ_p, α is a normalization factor, n_p denotes the unit vector orthogonal to the front $\delta\Omega$ at point p, and $\perp$ stands for the orthogonal operator, as illustrated in Fig. 5.

Based on the filling order, a missing region is filled with the most similar neighboring patches (measured by the sum of squared differences). After completing each spatio-temporal slice of a video frame, we use the Sobel edge detector to find the boundary of the object's trajectory in the slice. Then, the completed spatio-temporal slices are combined to construct a virtual contour, which is used to guide the subsequent posture mapping and retrieval process.

Sometimes, image inpainting errors lead to imprecise virtual contours, making it difficult to retrieve correct postures for object inpainting. To resolve this problem, we use the object tracking scheme proposed in [27] to correct image inpainting errors. To inpaint an occluded object, our method tracks the object in the non-occlusion period to obtain their positions. Accordingly, each spatio-temporal slice is then divided into two regions, the background region and the foreground trajectory, which allows us to apply image inpainting to the regions separately and thereby avoid inpainting errors. That is, available foreground information will only be used to infill the missing region of foreground region, and vice versa. Fig. 6 shows that the tracking-based correction technique significantly reduces the distortion of a virtual contour caused by inpainting errors.

The rationale behind the proposed virtual contour construction method is that if the continuity of object trajectories can be maintained in individually completed spatio-temporal slices, then the motion continuity of an object reconstructed by combining all the inpainted slices will also be maintained. Thus, so long as the linear line motion assumption holds during the occlusion period, a virtual contour can provide fairly precise information about the posture and filling location of an occluded object, even if the object is badly damaged.

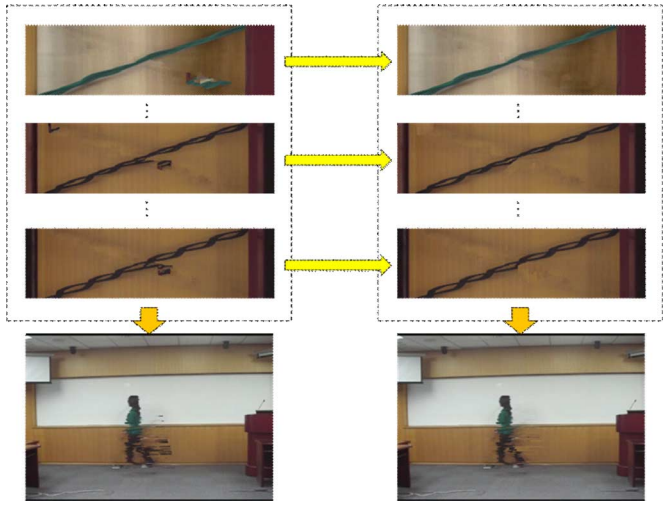


Fig. 6. Virtual contours constructed by combining 2-D spatio-temporal slices derived via the patch-based inpainting method proposed in [14]. The left-hand side shows the virtual contours obtained by combining completed spatio-temproal slices without corrections, and the right-hand side shows the virtual contours with corrections.

C. Key Posture-Based Posture Sequence Matching

After composing a sequence of consecutive virtual contours, we use them to match the most similar posture sequence in the set of available postures to complete the occluded objects. To simplify the posture sequence matching process, we use the key posture selection method proposed in [24] to select the most representative postures from among the available postures. The method also uses the shape context descriptor in [24] to measure the distance (dissimilarity) between two postures. As illustrated in Fig. 3, given an object's posture [Fig. 3(a)], a set of feature points are selected to describe the object's silhouette [Fig. 3(b)]. To reduce the complexity of posture matching without sacrificing the matching accuracy significantly, a convex hull bounding the silhouette [Fig. 3(d)] is used to select a subset of key feature points [Fig. 3(e)] to describe the shape context of the object. The distance (dissimilarity) between two postures is evaluated by matching the two corresponding posture silhouettes by (3). A posture is deemed a key posture if its degree of dissimilarity to all key postures exceeds a predefined threshold, TH_{posture}, that is empirically set to be 0.08. The key-posture selection algorithm is summarized below.

Algorithm: Key Posture Selection

The set of key-postures $Q = \{q_1, q_2, \ldots, q_n\}$

The available posture database $B = \{b_1, b_2, \ldots, b_n\}$

For $i = 1$ to n

{

If $(Q = \phi)$

$Q = Q \cup b_t$

else if $(H(b_i, q_j) > TH_{\mathrm{posture}}, \forall q_j)$

$\mathrm{Q} = \mathrm{Q} \cup b_t$

}

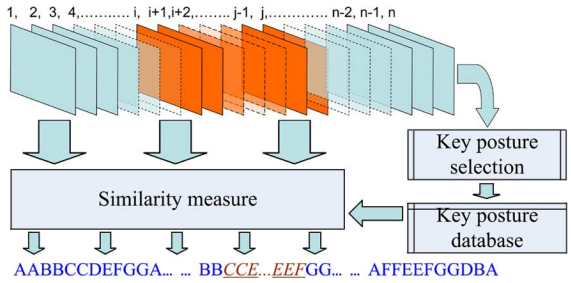


Fig. 7. Process for converting available postures and virtual contours into a sequence of key posture labels. The blue frames and numbers indicate the frames with available postures and their corresponding key-posture labels. The orange frames and numbers indicate the frames with constructed virtual contours and their corresponding key-posture labels.

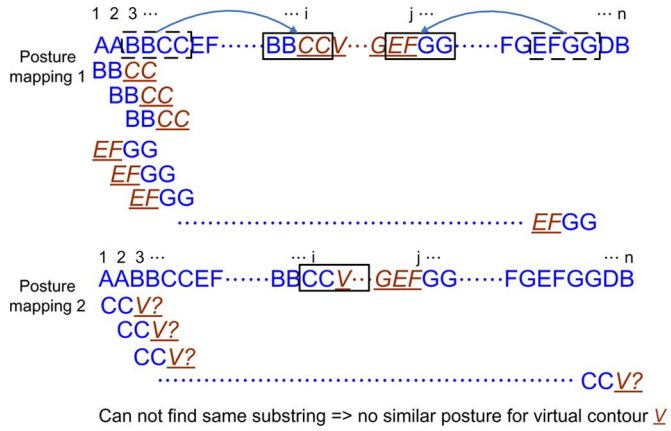


Fig. 8. Examples of using substring matching to solve the posture mapping problem. The length of the substring is 4. The blue numbers indicate the key-posture labels of available postures; the brown numbers indicate the labels of virtual contours; and the red numbers indicate the labels of available postures used to replace the occluded objects. In the first posture mapping, the available postures in frames 5, 6, $n-5$, and $n-4$ are deemed the best matches to replace the damaged objects in frames i, $i+1$, $j-1$, and j, respectively. In the second mapping, however, a good match cannot be found for the damaged object in frame $i+2$ (with the virtual contour labeled "V").

After the key posture selection process, each key posture is labeled with a unique number. The virtual contour of each occluded posture is then matched with the key posture that has the most similar context, as defined in (3). 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. 7. 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 [26] 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. Fig. 8 shows two examples of using substring matching to solve the posture mapping problem. During the occlusion period, a string of labels in a fixed-size sliding window (the size is 4 in the example) is matched to the substring of labels in the normal periods. We use two sliding windows that start from each end of the occlusion period and move toward the center of the period. Each sliding window overlaps with the neighboring normal period by half a window. As a result, half of the labels in the initial string are derived from available postures and the remaining labels are obtained from the virtual contours. As illustrated in the first example of Fig. 8, the left sliding window initially consists of four postures encoded as "BBCC" including two available postures (the "BB" part) in frames $i-2$, $i-1$ and two virtual contours (the "CC" part) in frames i, $i+1$. The right sliding window initially contains four postures encoded as "EFGG" where "EF" represents the two virtual contours in frame $j-1$ and j and "GG" represents the two available postures in frames $j+1$, and $j+2$, respectively. In this example, the available postures in frames 5, 6, $n-5$, and $n-4$ of the two initial sliding windows are deemed the best-match sequence to replace the damaged objects in frames i, $i+1$, $j-1$, and j. In the second matching, however, a good match cannot be found for the damaged object in frame $i+2$ (with virtual contour label "V") after substring matching. Our method handles such situations by constructing synthetic key-postures, as will be discussed later.

Using the proposed key-posture selection and mapping method to encode a sequence of virtual contours and available postures with a compact representation of key-posture labels has two advantages. First, since there are many efficient substring matching algorithms, converting the posture sequence retrieval problem into a substring matching problem reduces the computational complexity substantially. Second, as the occluded objects are completed for a whole subsequence rather than for individual frames, the temporal continuity of object motion is maintained.

D. Synthetic Posture 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. 9 shows how a new posture is synthesized by using three constituent components (the head, torso, and legs) from different available postures selected by a skeleton matching process.

The flowchart of the proposed synthetic posture generation process is shown in Fig. 10. First, the skeleton of a virtual contour that cannot find a good match in the posture database is extracted using the scheme proposed in [28], which is also used to extract the skeletons of all available postures. Then, the constituent components of each selected key-posture are decomposed based on the distribution of the variance in alignment errors between every two aligned key-postures. The component decomposition result of key postures is used to help segment the extracted skeletons into their constituent components. We use the segmented skeleton components of a virtual contour to

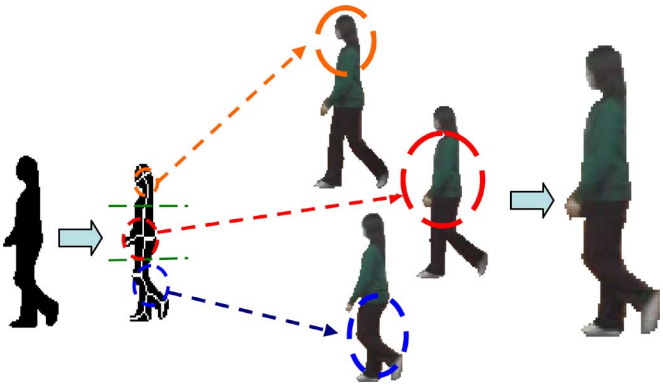


Fig. 9. Synthesizing a new posture using available postures. The new posture is comprised of three components (the head, body, and legs) taken from different postures.

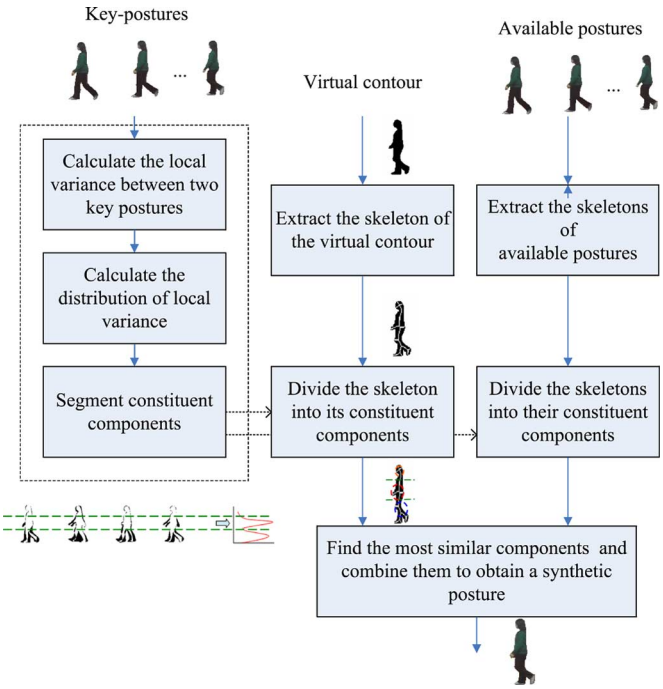


Fig. 10. Flowchart of the proposed synthetic posture generation process.

retrieve similar posture components, which are then used to synthesize new postures.

All of the above-mentioned constituent components are derived from the components of existing database postures. To use these components, we need to perform segmentation on the key-postures in advance, as shown in Fig. 11. After aligning the postures, we compute the difference between every two consecutive key postures. From the distribution of the variance, it is possible to identify the components that move more frequently. Then, we label the "frequently moving" components as the constituent components of the key-posture synthesis process.

We use the skeletons of objects to retrieve similar posture components, which are then used to synthesize new postures. To extract object skeletons, we employ the method proposed in [28]. It defines candidate skeleton points as the centers of the maximal disks located inside the planar shape. Then, a

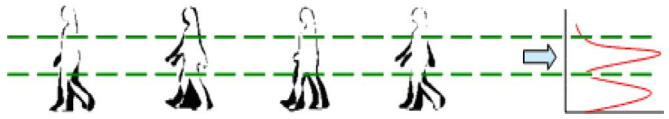


Fig. 11. Constituent components of a posture are partitioned based on local variance extraction. The dashed lines which separate postures into constituent components are determined based on the distribution of local variance shown on the right-hand side.

Euclidean distance map is used to determine whether or not a candidate skeleton point is a genuine skeleton point. A candidate skeleton point is deemed a real skeleton point if any one of its eight neighbors satisfies the following connectivity criterion:

$$\frac{r_2^2 - r_1^2}{\max(x,y)} \leq 1 \quad \text{and} \quad D^2 \geq \rho \tag{7}$$

where $x = |x_2 - x_1|$ and $y = |y_2 - y_1|$, in which (x_1, y_1) and (x_2, y_2) denote, respectively, the coordinates of the two nearest contour points e_1 and e_2; r_1 and r_2 represent, respectively, the shortest and longest distances between the contour point and the neighbors of the skeleton point; D is the distance between the two nearest contour points; and ρ is a pre-determined threshold.

We use the following relevance metric, K, to measure the contribution of an arc to the shape of a contour in order to determine whether the arc is a redundant branch of the skeleton:

$$K(l_1, l_2) = \frac{\beta(l_1, l_2) l(l_1) l(l_2)}{l(l_1) + l(l_2)} \tag{8}$$

where l_1 and l_2 represent, respectively, two line segments of the object's contour; $\beta(l_1, l_2)$ is the turn angle at the common vertex of segments ls_1 and ls_2; and $l(\cdot)$ denotes the length function.

The relevance metric allows us to select and remove arcs that only make a small contribution to an object's shape. This operation reduces the shape's contour, which is then used to remove unimportant skeleton points. We use the thresholds derived in the posture classification step to separate the skeletons of virtual contours and those of the available postures. After aligning the parts of a skeleton in the virtual contours with the corresponding parts in the available postures, the best-matched skeleton components of the available postures can be identified based on the following similarity metric:

$$S(T, S) = \sum_{(t_{x,y} \in T) \cap (s_{x,y} \in S)} w(t_{x,y}, s_{x,y}) \tag{9}$$

where T and S denote, respectively, the skeleton component of a virtual contour and the corresponding part in an available posture; and $w(t_{x,y}, s_{x,y})$ represents the matching score of the corresponding skeleton points, $t_{x,y}$ and $s_{x,y}$, of the virtual contour and the available posture, defined as follows:

$$w(t_{x,y}, s_{x,y}) = \begin{cases} score_1, & \text{if } t_{x,y} \text{ and } s_{x,y} \text{ belong to the skeleton region} \\ score_2, & \text{if } t_{x,y} \text{ and } s_{x,y} \text{ belong to the foreground region} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

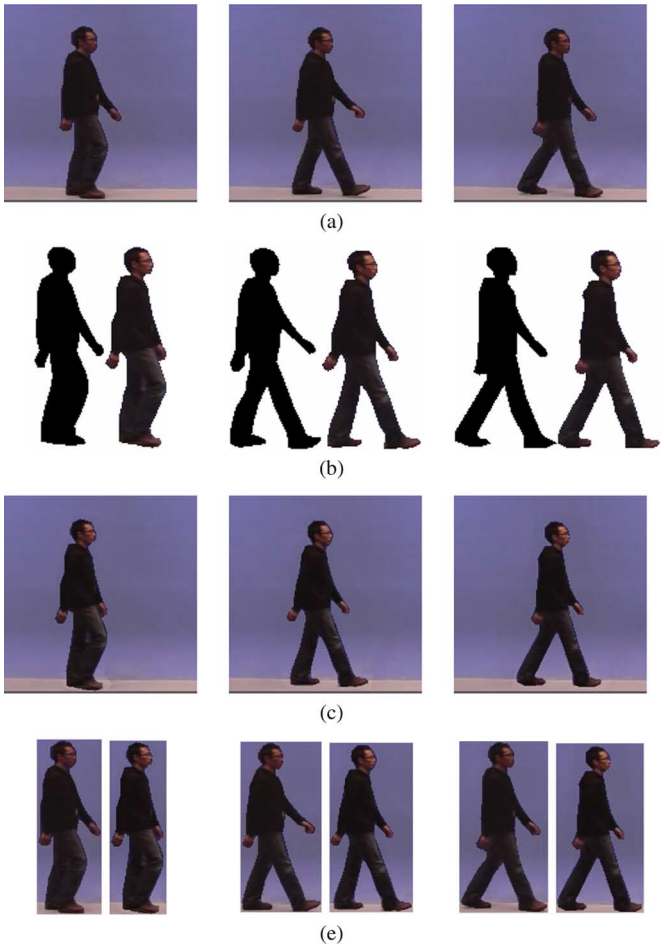


Fig. 12. Test sequence #1 containing a single pedestrian. (a) Some snapshots of the original video (ground-truths). (b) Virtual contours (on the left), which are constructed by combining the completed spatio-temporal slices and their corresponding best-match postures (on the right). (c) Corresponding completed frames. (d) Comparison of the completed objects (on the left) and the ground-truths (on the right).

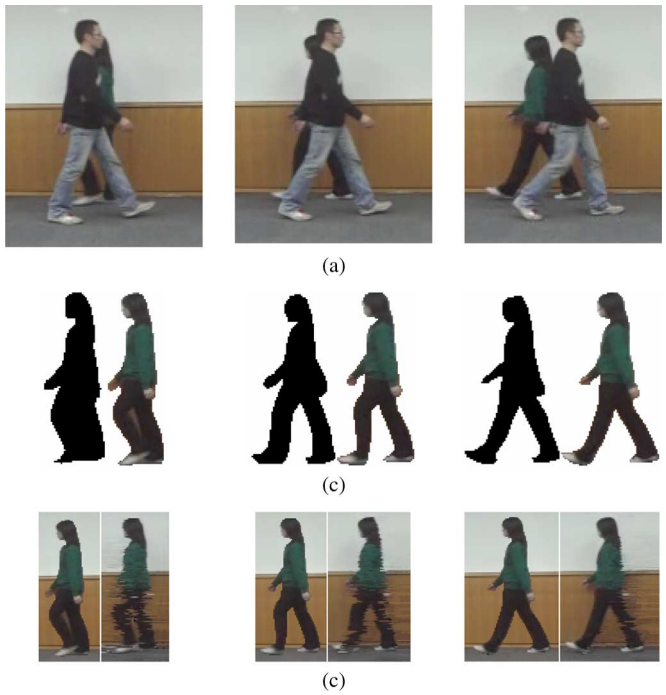


Fig. 13. Test sequence #2 with two people walking toward each other. (a) Original video frames. (b) Virtual contours (on the left), which are constructed by combining the completed spatio-temporal slices and the corresponding best-match postures (on the right). (c) Completed frames (on the left) using the original key-postures and the additional synthetic postures and the corresponding frames composed from the completed 2-D slices (on the right).

Here, the two score constants, $score_1$ and $score_2$, are set empirically as 3 and 1, respectively.

Finally, a new posture can be synthesized by combining all the best-matched constituent components of the available postures selected by the component-wise skeleton retrieval process.

IV. Experimental Results

We used six test sequences to evaluate the efficacy of our method. Five sequences were captured by a commercial digital camcorder with a frame rate of 30 fps, and a resolution of 352 $\times$ 240 (SIF). The remaining one was taken from [1]. In the experiments, we first removed unwanted objects and occluded objects completely, and then used the proposed inpainting method to reconstruct the occluded objects. For subjective performance comparison, readers can obtain the complete set of test results, including the original test videos, the videos after object removal, and the completed videos, from our project website [30].

Fig. 12(a) shows some snapshots of test sequence #1, which contains a pedestrian. In this experiment, we intentionally removed the person from 20 consecutive frames, and then used the proposed method to restore the missing person. This test case simulates a real-world situation in which objects in a number of consecutive frames are damaged due to packet loss during transmission of the video (e.g., the loss of several video-object-planes of an MPEG-4 stream), or due to a damaged hardware component (e.g., a hard disk or an optical disk). Since we have the ground-truth of the missing object in this case, we can evaluate the performance of our object completion method based on the ground-truth. First, we observe that the virtual contours of the missing objects, constructed by combining the completed spatio-temporal slices [shown in Fig. 12(b)], retain most of the objects' posture information. This verifies that the virtual contour of a missing object provides a fairly good initial estimate for finding the best-matched available posture to complete the missing object. Fig. 12(c) show that the objects completed frame-by-frame by the proposed posture mapping scheme conform to the ground-truths very well. Moreover, the scheme maintains the temporal continuity of object motion even if the object is lost completely in several consecutive frames.

Test sequence #2, shown in Fig. 13(a), simulates a common real-life situation that occurs in home videos, i.e., two people walking toward each other. In this scenario, one person is occluded by the other, which is not desirable. This case is similar to the situation where a moving object is occluded by a stationary object. After removing the unwanted object, we use the proposed method to restore the partially/completely occluded object. Fig. 13(b) shows, once again, that the virtual contours of damaged objects provide reasonably good estimates of the objects' postures. We do not have a ground-truth for this test sequence. However, Fig. 13(c) shows that the restored person

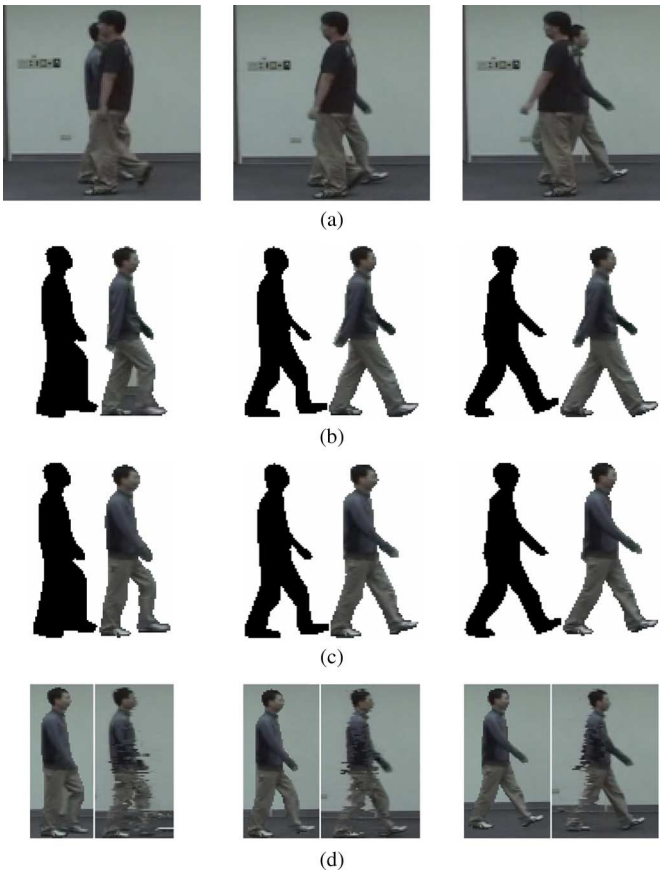


Fig. 14. Test sequence #3 containing two people walking toward each other (with a long occlusion period). (a) Original video frames. (b) Virtual contours (on the left) and the corresponding best-match postures (on the right) without including synthetic postures. (c) Virtual contours (on the left) and the corresponding best-match postures (on the right) with the additional synthetic key-postures. (d) Completed frames (on the left) using the original key-postures and the additional synthetic postures and the corresponding frames composed from the completed 2-D slices (on the right).

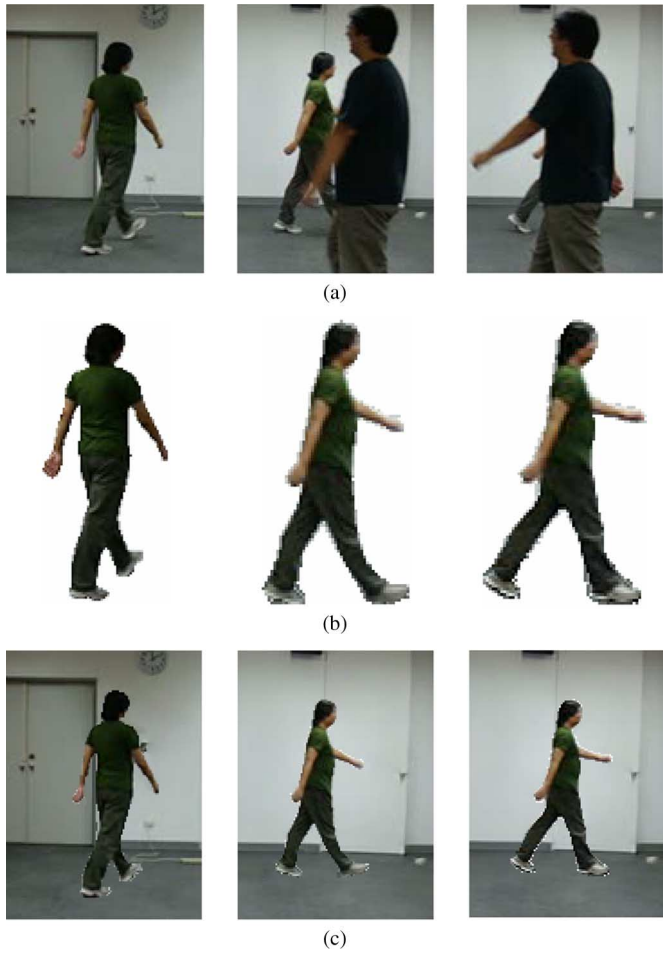


Fig. 15. Test sequence #4. (a) Some snapshots of the original video. (b) Corresponding best-match postures. (c) Result derived by the proposed method.

TABLE I
RUN-TIME ANALYSIS OF KEY OPERATIONS IN THE PROPOSED METHOD

	Virtual contour generation	Posture mapping	Synthetic posture generation
Sequence #1	838.53 s	9.82 s	not used
Sequence #2	181.56 s	9.36 s	not used
Sequence #3	195.64 s	9.06 s	82.89 s
Sequence #4	608.16 s	9.21 s	not used

moves with rather natural and continuous postures. Besides, our method maintains the temporal motion continuity of the object well. Note, the occluded girl turns her body a bit (i.e., the pose angle is changed) during the occlusion period. Since the pose angles of available postures are slightly different from the actual ones, the occluded objects are replaced with the available postures with similar silhouette information but different pose angles, leading to some artifact during the transition of pose angle (see the video in [30]). Such pose angle change problem has not yet been addressed in this work.

Test sequence #3 [Fig. 14(a)] is similar to test sequence #2, except that the person is occluded for a significantly longer period than in sequence #2. The longer occlusion period made it difficult to complete the occluded object because only a small number of available non-occluded postures were available in the sequence. In other words, the key-postures selected from the available postures were not sufficiently comprehensive, so we could not find a good match among the key-postures for the occluded object. Fig. 14(b) shows the virtual contours of the occluded object and its corresponding matched postures. The postures matched with the set of insufficient available postures appear to be incorrect in the hands and legs, leading to visually unpleasant artifacts in the completed video. Recall that our scheme minimizes the effect of insufficient available postures by adding synthetic postures to the available posture database to enrich the choice of postures, as shown in Fig. 14(c) and (d).

Test sequence #4, shown in Fig. 15(a), also shows two people walking toward each other, where the subject moves both horizontally and vertically. Moreover, the subject changes direction leading to nonlinear motion and change of object size. In this scenario, we perform posture alignment/normalization prior to sampling the 2-D spatio-temporal slices. After removing the unwanted object, we use the proposed method to restore the occluded object. Fig. 15(c) shows that, even with nonpure horizontal motion and nonlinear motion, the proposed method is still effective in maintaining the spatial consistency and temporal continuity.

The proposed system was implemented on a PC equipped with Intel Core2 Duo CPU 2.83 GHz and 3.5 GB system memory. The codes (implemented in MATLAB) for patch-based image inpainting and skeleton generation are obtained from [16] and [28], respectively. The remaining codes are all implemented in C++. The run time of each step for each test sequence is listed in Table I. In the four test sequences, the number of available postures in sequence #3 is not rich enough to achieve satisfactory object inpainting performance. Therefore, the synthetic posture generation process is used to improve the performance.

V. Conclusion and Discussion

To resolve a number of problems related to video completion, we have proposed a novel method that treats the completion of objects and completion of the background separately. The method is comprised of three steps: virtual contour construction, key posture-based sequence retrieval, and synthetic posture generation. We have also proposed an efficient posture mapping method that uses key posture selection, indexing, and coding operations to convert the posture sequence retrieval problem into a substring matching problem. In addition, we have developed a synthetic posture generation scheme that enhances the variety of postures available in the database. For background inpainting, we use a background mosaic-based scheme and correspondence maps to complete missing background segments. Our experiment results show that the proposed method generates completed objects with good subjective quality in terms of the objects' spatial consistency and temporal motion continuity. It also avoids over-smoothing artifacts and compensates for insufficient available postures.

The proposed method still has a few constraints. First, if an object moves nonlinearly during an occlusion period, the virtual contour construction may not compose sufficiently accurate postures. But should there be enough non-occluded portion of the object, the linear motion constraint may be relaxed. Second, currently the proposed method does not deal with the illumination change problem that occurs if lighting is not uniform across the scene. Third, the synthetic posture generation method can only deal with objects that can be explicitly decomposed into constituent components (e.g., a walking person), but may not synthesize complex postures.

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. T. Jia, S. M. Hu, and R. R. Martin, "Video completion using tracking and fragment merging," *Visual Comput.*, vol. 21, no. 8–10, pp. 601–610, Aug. 2005.

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

[4] V. Cheung, B. J. Frey, and N. Jojic, "Video epitomes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, Jun. 2005, pp. 42–49.

[5] T. K. Shih, N. C. Tang, and J.-N. Hwang, "Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 347–360, Mar. 2009.

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

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

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

[9] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Kauai,, HI, Dec. 2001, pp. 355–362.

[10] 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, Oct. 2007, pp. 1–8.

[11] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full frame video stabilization with motion inpainting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1150–1163, Jul. 2006.

[12] M. V. Venkatesh, S.-C. Cheung, J. Paruchuri, J. Zhao, and T. Nguyen, "Protecting and managing privacy information in video surveillance system," in *Protecting Privacy in Video Surveillance*, A. Senior, Ed. New York: Springer, 2009.

[13] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Conf. Comput. Vis.*, 1999, vol. 2, pp. 1033–1038.

[14] L. Wei and M. Levoy, "Fast texture synthesis using tree structured vector quantization," in *Proc. ACM SIGGRAPH*, 2000, pp. 479–488.

[15] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM SIGGRAPH*, 2000, pp. 417–424.

[16] 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, Sep. 2004.

[17] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, Jul. 2005.

[18] T. Huang, S. Chen, J. Liu, and X. Tang, "Image inpainting by global structure and texture propagation," in *Proc. ACM Conf. Multimedia*, Augsburg, Germany, Sep. 2007, pp. 517–520.

[19] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Who? When? Where? What? A real-time system for detecting and tracking people," in *Proc. IEEE Int. Conf. Automatic Face Gesture Recognit.*, Los Alamitos, CA, 1998, pp. 222–227.

[20] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, Aug. 2004.

[21] Y. Li, J. Sun, and H.-Y. Shum, "Video object cut and paste," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 595–600, Aug. 2005.

[22] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1120–1129, May 2006.

[23] 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.

[24] 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, Jan. 2009.

[25] A. Elgammal and C.-S. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, Jun. 2004, pp. 681–688.

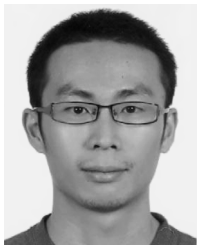
[26] T. H. Corman, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.

[27] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

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

[29] D. Cremers, S. J. Osher, and S. Soatto, "Kernel density estimation and intrinsic alignment for shape priors in level set segmentation," *Int. J. Comput. Vis.*, vol. 69, no. 3, pp. 335–351, Sep. 2006.

[30] NTHU Video Inpainting Project. [Online]. Available: http://www.ee.nthu.edu.tw/cwlin/inpainting/inpainting.htm.



Chih-Hung Ling received the B.S. and M.S. degrees in computer science and information engineering from National Chung-Cheng University, Chiayi, Taiwan, in 2003 and 2005, respectively. He has been pursuing the Ph.D. degree in the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, since 2005. His research interests include computer vision, pattern recognition, and multimedia signal processing.

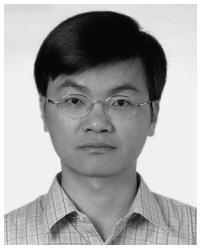
Chia-Wen Lin (S'94–M'00–SM'04) received the Ph.D. degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2000.

He is currently an Associate Professor with the Department of Electrical Engineering, NTHU. He was with the Department of Computer Science and Information Engineering, National Chung Cheng University (CCU), Chiayi, Taiwan, during 2000–2007. Prior to joining academia, he worked for the Information and Communications Research Laboratories, Industrial Technology Research Institute (ICL/ITRI), Hsinchu, Taiwan, during 1992–2000, where his final post was Section Manager. From April 2000 to August 2000, he was a Visiting Scholar with the Information Processing Laboratory, Department of Electrical Engineering, University of Washington, Seattle. He has authored or coauthored over 90 technical papers. He holds more than 20 patents. His research interests include video content analysis and video networking.

Dr. Lin is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the *Journal of Visual Communication and Image Representation*. He has served as a Guest Co-Editor of three special issues for the IEEE TRANSACTIONS ON MULTIMEDIA, the *EURASIP Journal on Advances in Signal Processing*, and the *Journal of Visual Communication and Image Representation*, respectively. He served as Technical Program Co-Chair of the IEEE International Conference on Multimedia & Expo (ICME) in 2010, and Special Session Co-Chair of the IEEE ICME in 2009. He was a recipient of the 2001 Ph.D. Thesis Awards presented by the Ministry of Education, Taiwan. His paper won the Young Investigator Award presented by SPIE VCIP 2005. He received the Young Faculty Awards presented by CCU in 2005 and the Young Investigator Awards presented by National Science Council, Taiwan, in 2006.

Chih-Wen Su received the B.S. degree in mathematics and the M.S. degree in computer science, both from Fu-Jen University, Hsinchuang, Taiwan, in 1999 and 2001 respectively, and the Ph.D. degree in computer science and information engineering from National Central University, Chung-Li, Taiwan, in 2006.

He is currently a postdoctoral research fellow with the Institute of Information Science, Academia Sinica, Taipei, Taiwan. His research interests are in image and video analysis, and content-based indexing and retrieval.



Yong-Sheng Chen (M'03) received the B.S. degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 1993 and the M.S. degree and the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1995 and 2001, respectively.

He is currently an Assistant Professor in the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. His research interests include biomedical signal processing, medical image processing, and computer vision.

Dr. Chen received the Best Paper Award in the 2008 Robot Vision Workshop and the Best Annual Paper Award of the *Journal of Medical and Biological Engineering*, 2008.

Hong-Yuan Mark Liao (SM'01) received the B.S. degree in physics from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981 and the M.S. degree and Ph.D. degree in electrical engineering from Northwestern University, Evanston, IL, in 1985 and 1990, respectively.

In July 1991, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, as an Assistant Research Fellow. He was promoted to Associate Research Fellow and then Research Fellow in 1995 and 1998, respectively. He is now the division chair of the computer science and information engineering division II, National Science Council of Taiwan. He is also jointly appointed as a Professor of the Computer Science and Information Engineering Department of National Chiao-Tung University. Since February 2009, he has been jointly appointed as the Multimedia Information Chair Professor of National Chung Hsing University. Since August 2010, he has been appointed as an Adjunct Chair Professor of Chung Yuan Christian University. His current research interests include multimedia signal processing, video-based surveillance systems, content-based multimedia retrieval, and multimedia protection.

Dr. Liao is on the editorial boards of the IEEE SIGNAL PROCESSING MAGAZINE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. He served as a Guest Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Special Issue on Video Surveillance (September 2008). He was an associate editor of the IEEE TRANSACTIONS ON MULTIMEDIA during 1998–2001. During 2004–2007, he served as a member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society. Since January 2010, he has served as a member of the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society. During 2006–2008, he served as the president of the Image Processing and Pattern Recognition Society of Taiwan. In June 2004, he served as the conference co-chair of the 5th International Conference on Multimedia and Exposition (ICME) and technical co-chair of the 8th ICME held in Beijing. In 2011, he will serve as General co-chair of the 17th International Conference on Multimedia Modeling. He was a recipient of the Young Investigators' award from Academia Sinica in 1998. He received the Distinguished research award from the National Science Council of Taiwan in 2003 and the National Invention Award of Taiwan in 2004. In 2008, he received a Distinguished Scholar Research Project Award from the National Science Council of Taiwan. In 2010, he received the Academia Sinica Investigator Award.