Video Motion Interpolation for Special Effect Applications

Timothy K. Shih, *Senior Member, IEEE*, Nick C. Tang, Joseph C. Tsai, and Jenq-Neng Hwang, *Fellow, IEEE*

***Abstract*—Video forgery, also referred as video falsifying, is a technique for generating fake videos by altering, combining, or creating new video contents. For instance, the outcome of a 100 m race in the olympic game is forged as an example in this paper. We track objects and segment motions using a modified mean shift mechanism. The resulting video layers can be played in different speeds and from different reference points with respect to the original video. In order to obtain a smooth movement of target objects, a motion interpolation mechanism is proposed based on reference stick figures (i.e., a structure of human skeleton) and a video inpainting mechanism. The video inpainting mechanism is performed in a quasi-3-D space via guided 3-D patch matching. Interpolated target objects and background layers are then fused. The objective is to create a forged video, which is almost indistinguishable from the original video. We demonstrate the original and the forged videos in our Web site at http://member.mine.tku.edu.tw/www/TSMC09/. Although video forgery may create moral or legal issues, which is beyond the scope of this paper, our intension is to create special effects in video editing applications.**

***Index Terms*—Image completion, mean shift, motion interpolation, object tracking, video falsifying, video forgery, video inpainting, video special effect.**

I. Introduction

ALTHOUGH the research of altering contents in a video may create a potential sociological problem, it is interesting and challenging to investigate video forgery, also referred as video falsifying, technologies if they are used with good intension (e.g., special effects of a movie). For instance, in [20], a spatiotemporal warping mechanism applied to video was proposed. By changing the time fronts obtained from a swimming competition video, the winner can be changed. The author further extended the technique to enable creating dynamic panoramic image sequences, which also allow new views to be generated from the panoramic image sequences, without using a 3-D model [18]. To change the content of video, techniques such as object tracking, motion interpolation, video inpainting, and video layer fusing are commonly used. The spatiotemporal warping mechanism [20] does not track objects, and there is no motion interpolation applied to the target objects; therefore, it prevents the mechanism from creating fake videos using different video sources. Video object tracking has been studied for a while. For instance, adaptive block matching [6] could be used to track objects efficiently. Contributions of motion interpolation techniques are mostly found in computer graphics. However, those techniques are difficult to interpolate video motions. One approach to solving the problem can be found in the video inpainting techniques [21], [22], which rely on an image completion mechanism. Image inpainting/image completion [3] is a technique to restore/complete the area of a removed object, which is manually selected by the users. Image inpainting techniques can complete holes based on both spatial and frequency features. Structural properties, such as edges of a house, are extracted and extended from spatial domain and used to complete an object [3]. In addition to [3], the work presented in [4] further refines image completion results in multiple runs from a coarse level to a fine level. The coarse level is used as an estimation to approximate a fine level. Another image completion approach [7] uses automatic semantic scene matching to search for potential scenes in a very large image database. The mechanism fills the missing regions (i.e., scenes) using information usually not in the same picture and provides a diverse set of results for a given input. On the other hand, textural information can be propagated from the surrounding areas toward the center of the hole such that a seamlessly natural scene can be recovered [11].

Although object selection is the only step that the user has to intervene in the completion procedure, many mechanisms suggest that human intelligence can be incorporated to produce a better result [15], [23]. The work discussed in [15] uses an interface to identify a source area, where texture information is used to inpaint another selected target area. The work discussed in [23] further suggests that most natural or artificial objects can be defined by a few main curves. By asking the user to draw a few curve lines, the algorithm proposed in [23] can produce excellent image inpainting results. However, for video inpainting, object selection is not possible due to the amount of frames.

In general, the problem of image completion can be defined as the following. Assuming that the original image I is decomposed into two parts, $I = \Phi \cup \Omega$, where Ω is a target area/hole manually identified by the user, and Φ is a source area with information to be used to complete Ω. And there is no overlap between the target area and the source area. These terms

Manuscript received August 17, 2009; revised December 19, 2009, March 21, 2010, and June 21, 2010; accepted August 23, 2010. Date of publication November 29, 2010; date of current version August 19, 2011. This work was supported by the National Science Council of Taiwan under Contract NSC98-2631-H-008-002 and Contract NSC99-2631-H-008-007. This paper was recommended by Associate Editor L. Zhang.

T. K. Shih is with the Department of Computer Science and Information Engineering, National Central University, Jhongli City 32001, Taiwan (e-mail: timothykshih@gmail.com).

N. C. Tang is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: nickctang@iis.sinica.edu.tw).

J. C. Tsai is with the Department of Computer Science and Information Engineering, Tamkang University, Taipei 251, Taiwan (e-mail: kkiceman@gmail.com).

J.-N. Hwang is with Department of Electrical Engineering, University of Washington Seattle, WA 98195, USA (e-mail: hwang@u.washington.edu).

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/TSMCC.2010.2077674

1094-6977/$26.00 © 2010 IEEE

(i.e., I, Φ, and Ω) are commonly used in most papers discussing inpainting algorithms.

Given the target object, one approach to completing the hole in video is to directly apply the techniques used in image completion, i.e., treating each video frame as an independent image. Most image completion techniques are based on one assumption, i.e., the target object Ω has a similar texture and continuous structure from the source area Φ. Therefore, the source and target areas are divided into equal-size patches, with the size of a patch being small (e.g., 3-by-3 or 5-by-5 pixels). Patches from the source area, using a sophisticated matching mechanism, are selected to fill in holes in the target area. Two important measurements are used: priority and confidence values [3]. Priorities of patches, which lay on the boundary of source and target areas, are computed according to spatial properties of the patches. Confidence values indicate the degree of reliable information of a patch inpainted onto the target area. The fill-in process is repeated from the outer boundary of the target area toward the inner boundary, until the target area is completely filled.

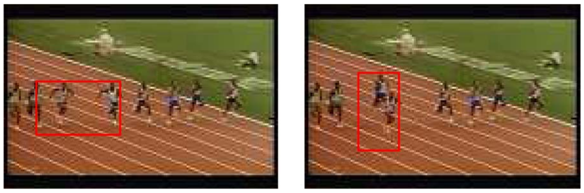
The video inpainting algorithm discussed in [16] extends the image completion approach proposed in [3] based on a static camera assumption. A moving target in a stationary video can be removed by filling-in exemplar patches in the same frame onto the missing background. Since the moving target can be partially occluded by another moving object, an interframe search is needed to find the best candidate patch. Only the portion of moving foreground in the selected patch is copied. The priority of the rest of pixels in the background is adjusted to zero such that the background is not changed. The video inpainting algorithm for still camera [16] is further extended to cope with nonstationary videos under restricted camera motions [17]. Background and foreground of video are separated, using the generated optical-flow mosaics. A similar priority concept used in [16] is also used in [17], to find the highest priority filling-in patches in the foreground. After foregrounds of all frames are inpainted, the remaining background holes are filled using patches directly extracted from adjacent frames and texture synthesis. A few assumptions are made in [17], e.g., camera motion is parallel to the plane of frames (i.e., no intrinsic camera rotations). And a moving object may not change its size (i.e., no zooming). Another interesting paper [13], which is less similar to our work, proposed detection and reconstruction of missing data in video based on a 3-D autoregressive model. Interpolation instead of patch duplication is used. Usually, only small missing regions can be repaired by interpolation techniques.

The paper [26] uses a motion layer segmentation algorithm to separate a video sequences into several layers according to the amount of motion. Each separated layer is completed by applying motion compensation and image completion algorithms. Except the layer with objects to be removed, all the rest of layers are combined in order to restore the final video. However, temporal consistency among inpainted areas between adjacent frames was not taken into account in [26]. The work discussed in [11] also removes objects from video frame by a priority scheme. Fragments (i.e., patches) on the boundary of the target area are selected with a higher priority. However, a fragment is completed using texture synthesis instead of copying from a similar source. A graph cut algorithm is used to maintain the smooth boundary between two fragments. To maintain a smooth temporal continuity between two target fragments, two continuous source fragments are selected with a preference. However, complex camera motion is not considered in [11].

The paper [25] looks at the problem from a 3-D perspective, which includes pixel positions (2-D) and frame numbers (time). The algorithm optimizes searching of patches at different resolution level. The inpainting results are visually pleasant. The only drawback is that the work [25] assumes the missing hole of every video frame is provided. Therefore, there is no tracking mechanism used to identify the object to be removed. The inpainting algorithm discussed in [9] repairs static background as well as moving foreground. The algorithm takes a two-phase approach, sampling phase and alignment phase, to predict motion of moving foreground and to align the repaired foreground with the damaged background. Since the algorithm can be extended to use a reference video mosaic, with proper alignment, the algorithm discussed in [9] can also work for different intrinsic camera motions. However, the background video generated based on the reference video mosaic is important. Mistreatment of a generated background video will result in ghost shadows of the repaired video. In [10], the same group of authors [9] further extends their algorithm to deal with variable illumination. Although part of the process (i.e., layer separation and moving foreground sampling) is semiautomatic, the completion process is automatic.

In one earlier investigation [21], the authors analyze temporal continuities based on stationary and nonstationary video with slow or fast foreground. The authors use a modified exemplar-based image completion mechanism. More specifically, for stationary videos, patches can be searched from different frames and copied to fill in a target area, which is tracked by a simple computation of optical flow. For nonstationary video, the tracking mechanism can be extended to deal with fast or slow moving objects. However, complicated video motions degrade the inpainting performance, due to the problem of "ghost shadows." That is, due to the temporal discontinuity of inpainted area, flickers can be produced and lead to a visual annoyance. In [22], the authors partially solve the "ghost shadows" problem by using a motion segmentation mechanism and a modified video inpainting procedure. The inpainted video is improved with less ghost shadows. However, it is still hard to deal with more complicated camera motions.

Video inpainting mechanism can be used to produce special effects (e.g., hollow man) in movie industry. The proposed video forgery technique further uses video inpainting and a newly proposed *motion interpolation* technique to create special effects. For instance, the behavior (e.g., speed of walking, direction of running, height of jumping) of a person in a video can be altered. An example of video forgery is shown in Fig. 1. We changed the winner of a 100-m running race. Several challenging issues in video technology need to be solved, such as object tracking, video layer separation, motion interpolation repositioning, and video layer fusion. In order to precisely model the spatiotemporal behavior of each video layer (and object), we consider a video as a spatiotemporal fluctuated domain. That is, different



(a) (b)

Fig. 1. Example of video forgery (In the same video, speed of the two runners in the red boxes are changed. The rest are intact.). (a) Original video frame; (b) corresponding falsifying result.

video space (layer) can be manipulated in different speed and screen position.

The procedure of video forgery is illustrated in Fig. 2. More specially, given a specification for video forgery, the original video is subdivided into several motion layers. We use a modified mean shift algorithm [2] to separate foregrounds from the video. The cycle of repeated motion is predicted to be used in motion interpolation. Motions of stick figures are computed such that precise patches on relative stick positions can be used as guidance for motion interpolation. Motion of target objects can be interpolated by using a quasi-3-D video inpainting mechanism. Depending on the relative time of a viewer (i.e., the speed of seeing a video), background motion (i.e., camera motion) can also be interpolated. Finally, we can optionally adjust the relative position of each object and combine video layers.

The critical step the user has to get involve in the forgery process is to select the object and starting position for slow or fast motion. However, it is also necessary for the user to select foreground video and background video. In addition, we also use fade-in/fade-out and object duplication steps in some of our examples. Thus the process is semiautomatic.

The procedure of video forgery is challenging, which requires a series of mechanisms. In the literature, there are papers addressing how to find forgery in image or video [5], [8], [12]. However, to create forgery video is quite different, as evidenced by many related technologies used in the procedure illustrated in Fig. 2. Our contribution in this paper can be summarized as the following:

1) Image/photo forgery can be achieved using commercial image processing tools. However, it is not possible to use such tools for video forgery. It is the first time that video forgery is attempted based on video inpainting techniques.
2) Fake motions are hard to produce. A new motion interpolation mechanism is proposed. Usually, motion interpolation algorithms are for 3-D avatars in virtual reality systems. We propose a new concept called guided inpainting for motion interpolation of video objects.
3) Although video consists of images, which can be inpainted, visual continuity in inpainting is essential. A guided quasi-3-D (i.e., *X*, *Y*, and time) video inpainting mechanism is proposed. The exemplar-based inpainting mechanism is extended to cope with temporal continuity.

The rest of this paper is organized as follows. Section II starts with a quick description of the adopted object tracking algorithm and our motion layer segmentation and tracking schemes. Section III discusses our video inpainting-based motion interpolation procedure. After the motion of target objects are interpolated and the background camera motions are compensated, we also address a simple but efficient layer fusion mechanism in this section. Experimental results and evaluations are given in Section IV, followed by the conclusion in Section V.

II. Motion Layer Segmentation and Tracking

In order to change the behavior of human objects in a video, the video forgery procedure needs to separate target objects from the background. In fact, a mechanism for video layer separation is necessary. We adopt the mean shift feature space analysis algorithm [2] for color region segmentation in each frame. This algorithm allows an initial segmentation of objects from their background. Color region segmentation is then combined with motion segmentation for object parts separation and video layer separation. Although object tracking is not a main contribution of our paper, it is necessary to present our modification of the mean shift algorithm to be applicable in our tracking program. Even though the tracking results may not be perfect, our inpainting technique can cope with some limited amount of missing details.

Algorithm: Reference Stick Figure Searching

Input: approximate contour region C_0 of target object on the 1st frame
Output: tracked object C_1 in all frames

1) Let C_0 = approximate contour region (i.e., the contour plus its inner pixels) of object be manually identified (see Fig. 3(a)); and C_0' = result of applying Mean Shift to C_0 for a refined segmentation (figure of C_0' is omitted).
2) We use a revised fast tracking mechanism [6] to identify the bounding box, B_1, of target object in the second frame (see Fig. 3(b)).
3) The Mean Shift algorithm is applied again to the bounding box B_1 to obtain B_1' (see Fig. 3(c)). The target object C_1' in the second frame is then roughly tracked (see Fig. 3(d)), by comparing color segments of C_0' and B_1'.
4) We then refine the tracking result by using morphological operations and comparison of pixels in the LUV color space (only L is used):
   4.1 The rough target object C_1' is enlarged to C_1^* by applying dilation three times (empirically set in our experiments).
   4.2 For each pixel, p, in C_1^*
      If the difference between L of p and L of p's corresponding pixel in $C_0 > L_2$
         exclude p in the tracked object C_1
      else
         keep p in the tracked object C_1

The threshold used for color value L in pixel comparison is empirically set to $L_2 = 2$ (small value) in our experiments, with the tracked object C_1, as shown in Fig. 3(e). Although the fast tracking algorithm [10] or another similar block searching approach [19] works well in general, however, their results do not provide a very accurate boundary. With the aforementioned four-step tracking algorithm produces a good tracking result. In addition, we use a larger area as compared to the tracked object. This strategy is used through our inpainting algorithm. However, the tracking procedure is not performed in real time. Therefore, the overall procedure of video forgery is not in real time either. The detailed computation time of several examples are given in Table I in IV-A.

Although a target is tracked, however, different parts of the target may move in different directions. To deal with motion

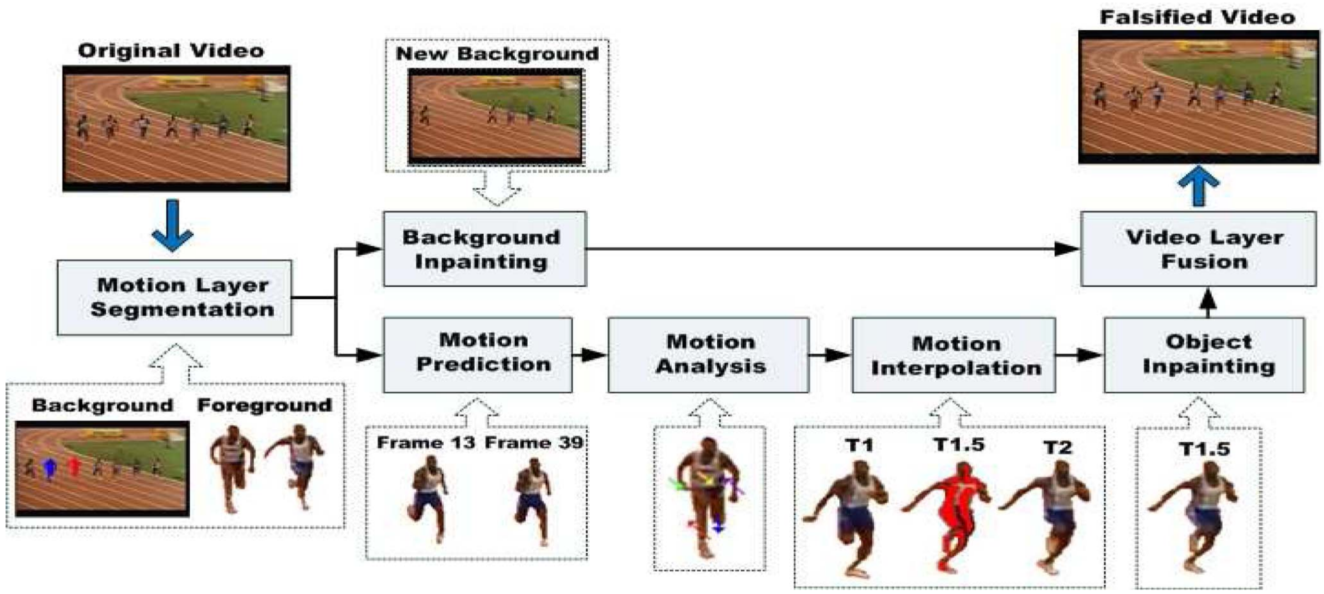


Fig. 2. Proposed framework

Fig. 3. Illustration of the proposed object tracking process. (a) User selects the target object in frame 0; (b) we apply [6] to locate the object in frame 1; (c) then the mean shift color segmentation algorithm [2] is applied to the located bounding box B_1 to obtain B_1'; (d) regions of object is roughly tracked by comparing color segments of C_0' and B_1'; (c) corresponding result of object tracking.

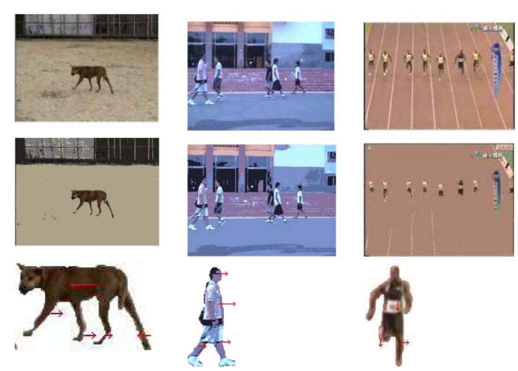


Fig. 4. Examples of motion segmentation. Top: the original video frame. Middle: corresponding result of color segmentation by using [2]. Bottom: the example of tracked object and estimated vectors.

segmentation, we use a revised block searching algorithm [10] to compute motion map (i.e., a matrix of direction of block movements in the next frame) based on the HSI color space. The selection of a different color space is to double check the consistency of tracking and segmentation procedure. To estimate the similarities among blocks of size 5-by-5 pixels (fixed in our algorithm), we use sum of absolute differences (SADs), based on the HSI color components separately, and the differences are accumulated. The equation is also used in [22]. The SAD equation is used to find a minimal distance between two similar blocks of adjacent frames. However, it is not possible to use the resulting motion map to identify an object (or layer) since the search results of similar blocks in the next frame may not be spatially close to each other. But a color region due to mean shift segmentation has adjacent pixels. Thus, we calculate the overlap areas between motion map and color region segments resulted from mean shift segmentation. We use color segments to identify object parts while use the average motion vectors of the overlapped areas to identify the motion of object parts. However, although one object should belong to a video layer in general, it is possible for an object to have two or more segments with different motions (e.g., legs and hands of a walking person are moving in different vectors). Therefore, we have to decompose an object into various regions. The mean shift color segmentation [2] can also be revised to deal with motion segmentation of various moving object regions by using the following five-tuple feature space (M_x, M_y, L', x, y), where M_x, M_y denotes the motion vector of a 5-by-5 block with its average pixel luminance L'. Therefore, mean shift is used to segment the motion based on blocks, not pixels in this case. Thus, the resulting object is segmented into different regions based on motions. Fig. 4 illustrates the results of object motion segmentation of three examples. The target objects and their motion vectors are displayed in row (c), which shows different regions of a target object with different motion vectors. Motion segmentation is important for video inpainting [22]. With a proper searching range for patches, ghost shadows can be eliminated. The solution will be discussed in the new video inpainting procedure in the following section.

III. Motion Interpolation Using Video Inpainting

Motions of the target object need to be interpolated. In some situations, motion interpolation may create background holes due to different shape of objects that are removed. Thus, video

TABLE I.
COMPUTATION TIME OF DIFFERENT STEPS IN VIDEO FORGERY FOR VIDEOS IN FIG. 16

Video Sequences			1. Motion Layer Segmentation	2.Motion Prediction	3.Motion Interpolation	4.Background Inpainting	5.Layer Fusion	Total
			(seconds per frame in average and percentages of computation time)					
(a), (b) the winner is changed	**Image Size:** 640*480 **Object Size:** 75*130	**Frame No.:** 150 **Cycle:** 13 frames	0.038 (3%)	0.097 (8%)	0.155 (12%)	0.861 (69 %)	0.094 (8%)	1.245
(c), (d) the ghost shadow jumps	Image Size: 640*480 **Object Size:** 102*181	**Frame No.:** 153 **Cycle:** 42 frames	0.070 (5%)	0.126 (10%)	0.188 (14%)	0.873 (66%)	0.072 (5%)	1.329
(e), (f) replaced background and somersault slow motion	Image Size: 640*480 **Object Size:** 153*140	**Frame No.:** 145 **Cycle:** 64 frames	0.081 (6%)	0.178 (13%)	0.198 (14%)	0.855 (61%)	0.091 (6%)	1.403
(g), (h) the reflected twin	Image Size: 320*240 **Object Size:** 40*68	**Frame No.:** 62 **Cycle:** 31 frames	0.009 (1%)	0.011 (2%)	0.065 (9%)	0.575 (81%)	0.047 (7%)	0.707
(i), (j) who chases who	Image Size: 640*480 **Object Size:** 344*304	Frame No.: 90 Cycle: 8 frames	0.401 (13%)	0.316 (10%)	0.372 (12%)	1.743 (58%)	0.196 (7%)	3.028
(k), (l) 4-people	Image Size: 320*240 **Object Size:** 30*80	**Frame No.:** 139 **Cycle:** 31 frames	0.012 (2%)	0.013 (2%)	0.057 (8%)	0.601 (82%)	0.047 (6%)	0.730
(m), (n) same person, different distances	Image Size: 320*240 **Object Size:** 21*46, 16*36	**Frame No.:** 135 **Cycle:** 34 frames	0.011 (2%)	0.010 (2%)	0.041 (7%)	0.423 (73%)	0.097 (16%)	0.582
(o), (p) 7-people	**Image Size:** 320*240 **Object Size:** 69*122	**Frame No.:** 66 **Cycle:** 40 frames	0.039 (13%)	0.015 (5%)	0.067 (22%)	0 (0 %)	0.174 (60%)	0.295
(q), (r) 2 people in complicated background	**Image Size:** 640*480 **Object Size:** 88*162, 86*152	**Frame No.:** 349 **Cycle:** 35 frames 40 frames	0.07 (9%)	0.02 (2%)	0.08 (11%)	0 (0 %)	0.573 (78%)	0.743

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

Fig. 5. Example of Motion Interpolation. (a) and (d) are the original objects at time t_n and t_{n+3}. (b) and (c) are the interpolated target objects at time t_{n+1} and t_{n+2} generated by the proposed interpolation technique.

inpainting to these holes is required. We discuss how to use inpainting procedure for motion interpolation in this section.

A. Motion Interpolation of Target Objects

A target object can be segmented into a layer. To produce a slow motion of the target layer, motion interpolation is required. In Fig. 5(a) and (d), two target objects at t_n and t_{n+3} are given. Fig. 5(b) and (c) are interpolated target objects. With this interpolation, the resulting video can be played in three times slower than the original video. The interpolation technique can be applied multiple times to create additional middle objects up to a limit. If the differences between frames are too close, interpolation stops.

B. General Inpainting Strategy

In order to obtain the interpolated figures, we use a new video inpainting technique. This new mechanism uses a rule-based thinning algorithm [1] to obtain the stick figures of target objects. The stick figures are used to guide the selection of patches, which are copied from the original video to fill the interpolated objects. The stick figures have been effectively used in analyzing golfing and walking video postures [24]. The mechanism is useful even with a small training set.

In an ordinary video inpainting algorithm [13], [14], the target area Ω receives copied patches of 3-by-3 or 5-by-5 pixels from the source area Φ. The information is searched and pasted in a

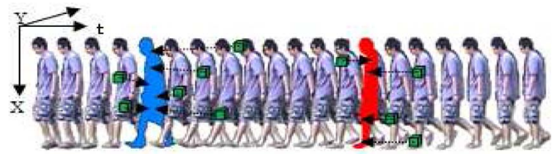
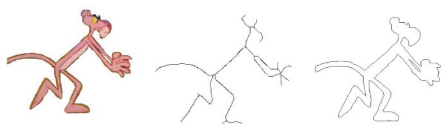


Fig. 6. Patch Selection in Quasi-3-D Video Inpainting. The undamaged patches will be selected by the proposed quasi-3-D video inpainting process to fill the damaged areas (marked by blue and red).



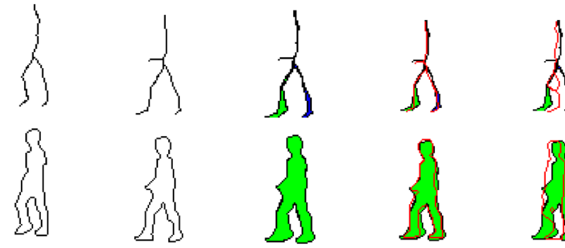
(a) (b) (c)

Fig. 7. Example of stick figure and contour of an object. (a) target object. (b) Stick figure of (a) generated by [1]. (c) Contour of (a).

2-D space. The concept can be extended to a quasi-3-D video space (2-D plus time). As illustrated in Fig. 6, there is a concatenation of target objects, with missing regions denoted by blue and red polygons. If we consider spatial continuities in *X* and *Y* axes, as well as temporal continuity in time (i.e., between frames), the advantage of using 3-D patches in quasi-3-D video inpainting can allow us to produce a smooth movement (or continuity) of the target object. However, for quasi-3-D motion interpolation, additional criteria should be considered, which will be discussed later.

1) Prediction and Interpolation of Cyclic Motion: We consider the following scenario. First, it is common for target objects to perform actions in a repeated cycle (e.g., walking man, flying bird). Second, a stick figure can be used to estimate the relative positions of patches (e.g., head, body, and legs). Thus patch selection needs to consider reproducing cyclic motions and maintaining the relationship of object parts. The computation of stick figures [1] does not need to include all body parts. Since the target objects have missing portions in most cases, stick figures are used only to estimate relative positions of object parts if they are visible. Stick figures and the contours of target objects can be used to predict repeated cycles. Fig. 7 illustrates a target object (a), its stick figure (b), and the corresponding contour (c) of the target object. Objects in the video are normalized by the maximum size of bounding boxes, as well as the centroid positions such that prediction of motion cycle can be precise.

The prediction of cyclic motion can be computed by comparing the stick figure and contour of the normalized target object in a selected frame and a group of subsequent frames. The comparison is carried out by finding the most similar overlapped stick figures and contours. A majority count of the estimated cycle number along the time is used to decide the cycle of repeated motion. For instance, in the example of 100 m race, the cycle is found to be 13 frames. The number of frames in a repeated cycle is denoted by *r* in the procedure of finding reference stick figures.



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

Fig. 8. Example of reference stick figure searching. (a) and (b) show the stick figure and contour of an object at time t_n and t_{n+1} respectively. (c) Union of stick figure and contour in (a) and (b). (d) and (e) show the example of whether a referenced stick figure is fit or not.

A missing stick figure can be reproduced either by searching for similar reference stick figures in a repeated motion or by interpolation of two known stick figures. The algorithm below finds reference stick figure in motion cycles. Before the algorithm starts, we use a best effort approach to compute *r* as the number of frames in a motion cycle of the target object, by comparing and finding similar stick figures and contours of the target object in the video. Also, the target object is normalized according to the size of its bounding box and centroid position. We define an index range function of a given frame number *x* as $idx(x) = [x+r-2, x+r-1, x+r, x+r+1, x+r+2] \cup [x-r-2, x-r-1, x-r, x-r+1, x-r+2]$.

ALGORITHM: REFERENCE STICK FIGURE SEARCHING

Input: A target object *L*
Output: Compute missing reference stick figure of *L* at a given time t_x

1) Let the stick figure of L in time t_x = *SFx* (see Figure 8)
2) For frame indices n in *idx(x)*, where *L* at a given time t_n is known (i.e., *L* is not missing at frame n)
   2.1 Compute the closing boundary $SF_{n+0.5}$ of stick figures SF_n in t_n and SF_{n+1} in t_{n+1}, apply the thinning algorithm to $SF_{n+0.5}$ to obtain a new stick figure *NSF*
   2.2 If all pixels of *NSF* are in $SF_{n+0.5}$
         return *NSF*
   else
         return stick figure not found

If the reference stick figure cannot be obtained from the aforementioned algorithm, an interpolated stick figure from two known target stick figures is used. Fig. 9(a) and (b) are known sticks, with the union stick, as shown in Fig. 9(c). The thinning algorithm is then applied to the union sticks to obtain the interpolated sticks in Fig. 9(d). The corresponding objects and the generated objects are shown in Fig. 9(e)–(g), respectively.

2) Motion Interpolation Algorithm: After the reference or interpolated stick figures are appropriately determined, it is time for us to fill in the necessary texture information on the stick figures. This leads us to the motion interpolation step. Our motion interpolation algorithm is based on the following two assumptions:

a) The missing region Ω has a similar texture and color representation to the source region Φ; and
b) The missing target region has a continuous motion with respect to the source region.

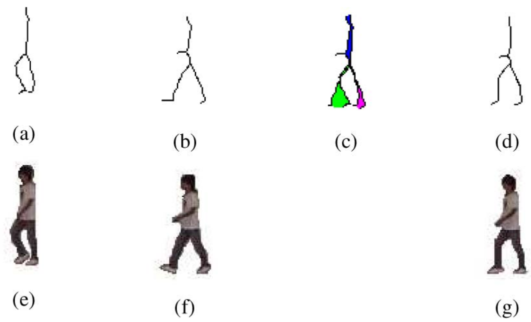


Fig. 9. Example of stick figure interpolation. (a) and (b) show the stick figure of object O_a and O_b; (c) union of stick figure shown in (a) and (b); (d) corresponding thinning result of (c); (e) and (f) show the object image of object O_a and O_b; (g) reference object image can found according to (d).

The first assumption motivates us to extend our image inpainting algorithm for motion interpolation. The second assumption allows us to consider a video as a 2-D plus time domain, as we have discussed in the general inpainting strategy. Thus, our motion interpolation algorithm can be implemented. Assume that, $I^3 = \Phi^3 \cup \Omega^3$, where Φ^3 is a source space and Ω^3 is a target space, and $\Phi^3 \cap \Omega^3 = \emptyset$ (i.e., an empty set). The notation of our quasi-3-D video inpainting algorithm follows one discussed in [3]. However, new concepts are proposed in our mechanism.

- Patches on the reference stick figures will be copied as guidance before the inpainting algorithm starts. This is an important step to ensure that motion is properly interpolated.
- Patches copied should be obtained from a corresponding position relative to the whole target object from the source (i.e., copy patches from head to head, from legs to legs). Information of relative position is obtained while we track objects via mean shift. The separation of patch search not only guarantees the visual quality but also improves the speed of inpainting procedure, since search space is reduced for each individual portion.

To implement the aforementioned two new concepts, we use the following algorithm. The Patch Assertion algorithm first selects the objects in frame to be inpainted. The algorithm then computes the missing portion of each object and finds the corresponding sticks and patches on these sticks [see Fig. 10(a)]. These patches (e.g., block of 3-by-3 pixels) are inserted into Ω^3 (the space of missing objects). A contour defined as missing boundary ω, [shown in red line in Fig. 10(b)] is searched in the nearby positions in the motion cycle. Patches on ω are asserted. Thus Ω^3 is reduced to a smaller nonfilled region [shown in red color in 2-D in Fig. 10(c)].

ALGORITHM: PATCH ASSERTION

Input: A sequence of L objects in frames
Output: A sequence of L objects in frames with patches asserted

For each objects in frames to be inpainted
1) Find stick figure of L object and copy patches
2) Define missing boundary of L = ω
3) Find and assert patches on ω

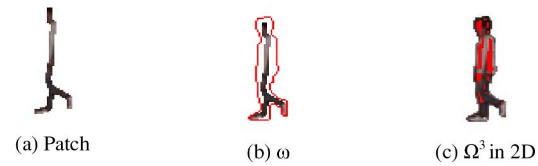


Fig. 10. Example for patch assertion. (a) Patches on stick figure. (b) Contour ω of (a), which can be searched in the nearby motion cycle. (c) Result of patch assertion.

Fig. 11. Mask for searching motion vectors. (a) and (b) show the mask used in the first and second round.

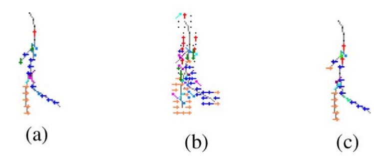


Fig. 12. Example of searched motion vectors. (a) and (b) show the searched motion vectors in first round and second round, respectively. (c) is the combined result of (a) and (b).

However, to precisely copy patches on sticks (the first step in algorithm Patch Assertion), additional effort is necessary to ensure the robustness of motion interpolation. Assuming that a target person includes several portions of body parts, which are not in the same color (in most cases this happens), we need to maintain the same body part color on interpolated stick figures, thus different portions of body can keep the colors intact after the target person is inpainted. We use a fast block searching mechanism to search for each patch located on the reference frame, with a patch size of 10-by-10 pixels. This first search attempt with patch location [shown as a red square in Fig. 11(a)] results in motion vectors, as shown in Fig. 12(a). However, to increase the accuracy of search, the second search attempt use a patch size of 7-by-7 pixels from eight surrounding patches [shown as eight red squares in Fig. 11(b)], with a motion vector map, as shown in Fig. 12(b). The two steps of motion vector maps are then combined based on a majority count strategy, which checks if the vector from the first search (10-by-10 pixel block) is consistent with more than four vectors from the second search (7-by-7 pixel blocks). If the check is consistent, the identified vector is used. Otherwise, the two types of blocks are enlarged by 1 pixel wide in X and in Y directions separately (i.e., 11-by-11 pixels and 8-by-8 pixels). Then, the majority count strategy is used again. If there is no match after blocks are enlarged, the vector of the 11-by-11 pixel block is used. With stick patches precisely selected from the reference frame, relative positions of body colors and details are maintained in the inpainting procedure.

Now, we explain our main inpainting algorithm. Let $P(p) = C(p) \times D(p)$, where $C(p)$ is a confidence term and $D(p)$ is a data term. Let $\partial\Omega^3$ be a front surface (similar to the front curve line discussed in [3]) on Ω^3 and adjacent to Φ^3. Thus, the front surface consists of front curve lines obtained from consecutive frames. Each front curve line $\delta\Omega$ [3] contains the pixels of the missing area Ω, which is adjacent to the source area Φ. Basic concepts and meaning of image inpainting notations can be found in [3].

$$C(p) = 1.0 \text{ iff} p \in \Phi^3 \text{and } C(p) = 0.0 \text{ iff} p \in \Omega^3 \tag{1}$$

$$\frac{\forall\ p \in \delta\Omega^3 \qquad C(p) = \left(\sum_{q\in(\Psi p^3 \cap \Phi^3)} C(q)\right)}{|\Psi_p^3|} \tag{2}$$

where Ψ_p^3 is a 3-D patch centered at point *p*. And the size of 3-D patch is denoted as $|\Psi_p^3| = 27$ pixels. Alternatively, *a* $5^3 = 125$ pixels patch can be used with a less efficient computation. Basically, the confidence term represents the percentage of useful information inside a patch centered at *p*.

The data term *D*(*p*), however, is different from *D*(*p*) defined in [3]. We use a 3×3 Sobel convolution kernel to obtain the edge map $\Phi^3\varepsilon$ of Φ^3. Instead of computing the isophote [3], we compute the percentage of edge pixels in the patch, by obtaining information from the edge map of the source region. Therefore, our data term is defined as

$$\frac{\forall\ p \in \delta\Omega^3\ D(p) = \min\left(1, \left(\sum_{q\in(\Psi p3 \cap \Phi 3\varepsilon)} \mathrm{C}\right)\right) \times \mathrm{var}\left(\Psi_p^3\right)}{\left|\Psi_p^3\right|} \tag{3}$$

where the constant *c*, which represents the weight, is set to 1 (for simplicity). And var(Ψ_p^3) is the color variation of the patch [21]. Finally, the motion interpolation algorithm is summarized in the following.

ALGORITHM: MOTION INTERPOLATION (THE MAIN ALGORITHM)

Input: Ω^3
Output: Completed Ω^3 using information from Φ^3

Repeat until region Ω^3 is empty
1) Compute boundary $\delta\Omega^3$ and P(p), $\forall$ p $\in \delta\Omega^3$
2) Propagate texture and structure information
 2.1 Find $\Psi_{p^\wedge}^3 = \max(\forall \Psi_p^3, p \in \delta\Omega^3)$
 2.2 Find $\Psi_{q^\wedge}^3 = \min_{\Psi q3 \in \Phi 3} SSD(\Psi_{p^\wedge}^3, \Psi_q^3)$
 2.3 Copy $\Psi_{q^\wedge}^3 \cap \Omega^3$ to $\Psi_{p^\wedge}^3 \cap \Omega^3$
3) Set $C(p) = C(p^\wedge) * (SAD(\Psi_{p^\wedge}^3, \Psi_{q^\wedge}^3) / \alpha)$, $\forall p \in \Psi_{p^\wedge}^3 \cap \Omega^3$

In this algorithm, $C(p^\wedge)$ represents an approximation of *C*(*p*), in a previous iteration before the confidence value is updated. The algorithm uses SAD to estimate the color difference of two 3-D patches and use a normalization factor α to make *C*(*p*) between 0.0 and 1.0. In addition, the searching area of 3-D patches is limited to the bounding box of tracked objects within the nearby five frames. As an example, parts of a source video are removed in Fig. 13(a). Stick figures are obtained in Fig. 13(b), with the generated reference sticks illustrated in Fig. 13(c). The results shown in Fig. 13(d) are computed by our quasi-3-D video inpainting algorithm for motion interpolation. Thus, the foreground is completed. However, we still need to deal with the background, which is discussed in the following.

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

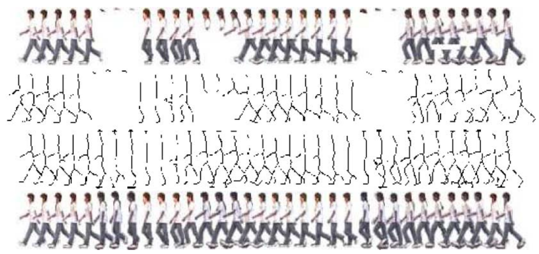


Fig. 13. Example of motion interpolation via inpainting. (a) Concatenation of damaged object motions; (b) Corresponding stick figure of (a); (c) shows the reference stick figures targeted by our motion interpolation process. (d) Correspond result of quasi-3-D video inpainting.

C. Inpainting Camera Motions

Video inpainting tasks can be more difficult due to different camera motions (e.g., zooming, tilting). To inpaint background video with constrained motions, we do not use the mosaic approach as discussed in [17]. Instead, we use a mechanism [22] to segment motions into different regions and based on the occluded objects, the background can be inpainted properly. In addition, depending on the speed of viewing the video playback, interpolation of background may be needed. To create fast video (i.e., similar to fast forwarding a video tape), we down sample the number of video frames. However, to create slow video, we interpolate video frames. The mechanism to segment motion regions discussed in [22] is further extended to estimate interpolated frames. These interpolated frames are produced after target objects are removed in each frame. Thus, our inpainting mechanism uses a multiple pass approach to process all video frames.

The segmentation of motion vector map is critical to inpaint camera motions. Ideally, each object is separated as one single segment. However, in real situation, an object usually has different portions moving in different directions (e. g., the four legs of a brown dog are moving in different directions). To inpaint an area in a frame, the inpainted area in the previous frame needs to be incorporated. In order to maintain a smooth temporal continuity of video flow, different portions of background due to camera motions need to be inpainted differently. Our mechanism proposed in [22] ensures that there is no "ghost shadows" created in the background.

D. Layer Fusion

After the motion of target objects are interpolated and the background camera motions are compensated, we need to merge video layers to produce the forged video. A simple but efficient layer fusion mechanism is presented here.

Our previous attempt in layer merging uses graph cut [14] to decide how to merge two layers and uses dilation to produce an overlapped area for pixel selection in the merging process. The synthesis mechanism discussed in [14] may produce nice results, especially with textures. However, to fuse two video layers, we found the use of image intensity is more effective than the use of image texture. In addition, structural properties of background can be miscalculated due to improper selection in

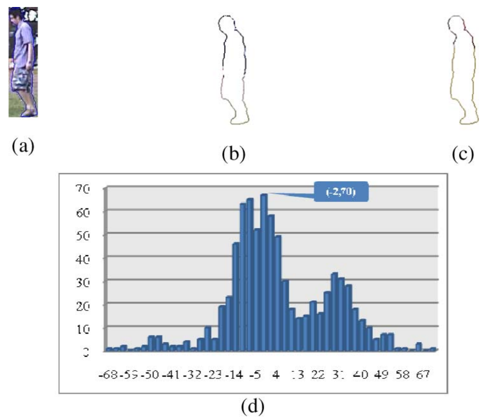


Fig. 14. Layer fusion elements. (a) Object layer obj. (b) Dilation area $\delta_{\rm obj}$ of obj. (c) Corresponding area on background layer $\delta_{\rm bkg}$. (d) Histogram of the intensity difference in $\delta_{\rm obj}$ and $\delta_{\rm bkg}$.

the merging process. Thus, we use a simple but acceptable layer fusion mechanism in our forgery procedure. In Fig. 14, elements used in our algorithm are illustrated. Assuming that the fusion process merges an object layer and a background layer, with contour of object layer computed based on the object tracking. The layer fusion algorithm can be summarized as follows.

ALGORITHM: LAYER FUSION

Input: the object layer

Output: the intensity adjusted object layer pasted on the background layer

Repeat until region Ω^3 is empty

1) Given the object layer (Fig. 14(a)), use dilation to enlarge the layer three times.
2) Let the enlarged area on object layer be δ_{obj} (Fig. 14(b)). And, let the corresponding area (where object to be placed) on background layer be δ_{bkg} (Fig. 14(c)).
3) Compute the difference of relative pixel intensities in δ_{obj} and δ_{bkg}. Calculate the histogram of the intensity differences (Fig. 14(d)), and find out the maximum count (e.g.,, 70 in this example) of intensity difference (e.g., -2 in this example). Let the intensity corresponding to the maximum count be υ_L.
4) For each pixel in the original object layer, adjust only pixel intensity according to υ_L and paste the pixel into the background layer.

In the algorithm, intensities between foreground and background are calculated and the foreground is adjusted. Since we only paste pixels in the object layer to the background, there is no affect to the structural properties of the background.

IV. EXPERIMENTAL RESULTS AND EVALUATION

We use the proposed video forgery procedures with various special effects (such as reflecting, dithering, or changing object positions) to create several sets of forgery videos at http://member.mine.tku.edu.tw/www/TSMC09/.

In our experiments, the block size used in the proposed patch assertion is 10-by-10 pixels. The patch size used in the proposed motion interpolation procedure is 3-by-3-by-3 pixels. Some examples are illustrated in Fig. 15. In videos (a) and (b), two runners are selected. One is applied with a fast motion mechanism, while the other is slowing down using motion interpolation. The outcome of 100 m race is changed. Videos (c) and (d) use a dithering operator and object removal technique. The effect is similar to the movie "Jumper" with a person suddenly shown in a place. Note that these are all nonstationary videos. Thus, to produce a similar result using frame-by-frame object pasting is difficult. Videos (e) and (f) use foreground and background from different videos. The somersault of human is motion interpolated to show slow motion. Videos (g) and (h) have a person and his reflected person walking in different directions. Videos (i) and (j) demonstrates slow motion of the pink panther in one video and slow motion of the white rabbit in another. The positions of panther and rabbit can be swapped as well. In videos (k) and (l), we add a stranger with red coat into the original video. In videos (m) and (n), we add two people at different distances. Video (o) and (p) demonstrate that we add a person in a crowed scene. In video (q) and (r), we add two people in a complicated background. Also, we show slow motion and fast motion on the demo Web site. These special effects are common in the movie industry, which commonly requires additional equipments, such as blue background and devices to merge analog video. Although a similar effect can be produced using image/video editing tools; however, most of them are labor intensive. With our tool, the creation of these special effects can be fairly systematic with limited users' interaction (i.e., the user has to provide the target object contour in initialization).

A. Evaluation

Table I summarizes the computation time in different steps, with respect to the examples shown in Fig. 15. The hardware used is CPU 2.1G with 2G RAM. We subdivide the video processing procedure into the following steps, as shown in Fig. 2.

1) **Motion Layer Segmentation**: The algorithm discussed in Section II to separate background and tracked objects is evaluated.
2) **Motion Prediction**: The algorithm Find Reference Stick-Figure to predict cycle of motion discussed in Section III-A.2 is evaluated.
3) **Motion Interpolation**: This step includes several portions, including motion analysis, patch assertion, and motion completion via inpainting. Mainly, the algorithms discussed in Section III-A.3 are evaluated.
4) **Background Inpainting**: Inpainting background of different camera motions discussed in Section III-B and [22] is evaluated.
5) **Layer Fusion**: The layer fusion procedure discussed in Section IV is evaluated.

Table I gives the average computation time by seconds per frame and the computation time in percentage to the entire video forgery process. According to these numbers, it is hard to produce forgery video in real time. The resolutions of video and the size of object bounding boxes for tracking are also provided. The tracked object in Fig. 15(a) is the runner on the third track, and the tracked object in Fig. 15(i) is the pink panther. In addition, in Table I, we provide the number of frames used in

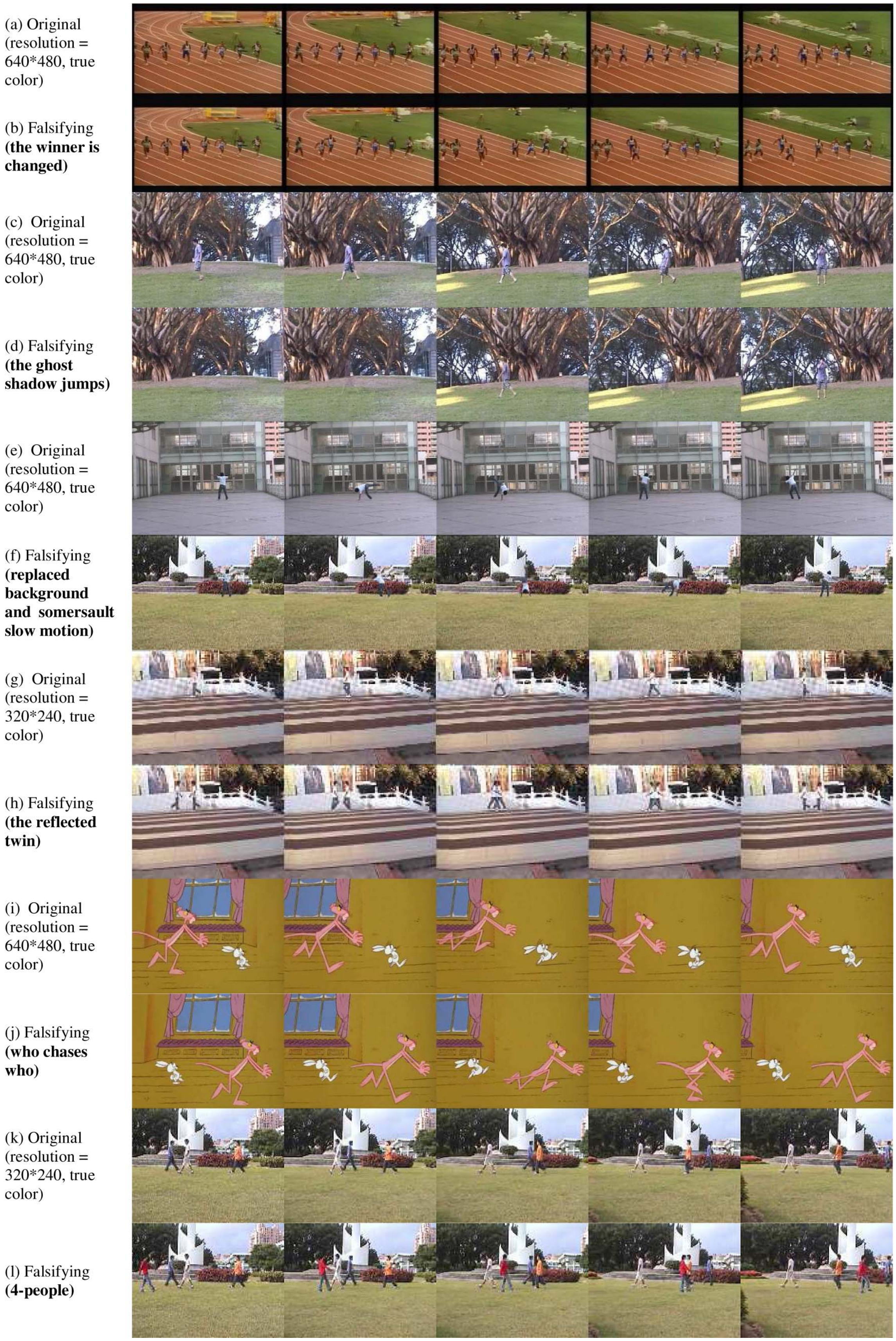


Fig. 15. Experiment results of video falsifying (see videos at http://member.mine.tku.edu.tw/www/TSMC09/). *(Continued.)*

(m) Original (resolution = 640*480, true color)

(n) Falsifying **(same person, different distances)**

(o) Original (resolution = 320*240, true color)

(p) Falsifying **(7-people)**

(q) Original (resolution = 640*480, true color)

(q) Falsifying **(2 people in complicated background)**

Fig. 15. *(Continued.)*

(a) (b)

Fig. 16. Improper estimation of ν_L. (a) Source video frame. (b) Corresponding falsified result of (a).

the video, as well as the number of frames per cycle computed using algorithm in Section III.

B. Limitations of the Proposed Mechanism

Although the results in our demonstration Web site are visually pleasant, however, a few limitations remain to be solved. We consider these as our future work and discuss possible direction for solutions.

1. For extremely fast motion, it is possible to have blurred video sources, which make precise tracking difficult. Blurred objects are difficult to be merged with any background video to produce a visually pleasant result.
2. Shadows cannot be tracked precisely in our examples. We are working on problem related to tracking shadows by using available algorithms in the literature.
3. The layer fusion algorithm only estimates pixel intensities in δ_{obj} and δ_{bkg}. If the object layer is large, the dilation enlarged area may fail to estimate the intensity difference U_L. Fig. 16 illustrates an example to show the limitation. It is relatively hard to merge two videos as compared to altering the behavior of actors in a single video. To solve the problem of misestimated U_L, the entire object layer can be used to estimate the intensity. In addition, the intensity differences can be classified as majority and minority. Majority can be averaged to obtain a new U_L.
4. Since we only use intensity to merge layers without considering the chrominance information. If the chrominance of object layer is quite different from the background, abnormal results occur. The layer fusion algorithm should consider the chrominance information if necessary.
5. In most cases, 3-D information is hard to be forged. Improper forgery video makes the fake video unnatural. A sophisticated 3-D reconstruction mechanism needs to be investigated.
6. For slow motion, our motion interpolation only produces actions two times slower. We do not allow a nonlinear stretching of time.

V. Conclusion

This paper proposes an interesting technology to alter the behavior of moving objects in a video. This paper effectively extends the inpainting technique [21] to a quasi-3-D space, with patch guidance. The first contribution of this paper is the motion interpolation technique for slow motions. In addition, the video forgery process allows a video to be separated into several layers, played in different speeds, and then merged. A series of difficult problems are solved and solutions are successfully integrated. In summary, this paper points out an interesting issue for video processing research. In contrast with video forgery, forgery detection is necessary. We provide our results on the demonstration Web site for the public to work for research purposes.

In addition to the limitations that we consider as our future work, it is necessary to develop an authoring tool to allow the users to specify the spatiotemporal fluctuation property for creating forgery videos. We are working on a video editing tool for users to create forgery video from existing resources, such as multiple video and image files. The tool uses a panorama associated with a reference timeline for all layers. Motion interpolation/extrapolation and action tracking provided by the user allow the positions and timing of objects to be defined.

Finally, we must say that, a quantitative evaluation metric for video forgery is difficult. Mostly, it is a subjective feeling of how a fake video looks real. However, it will be an interesting direction to see forgery detection as an evaluation mechanism to video forgery techniques.

REFERENCES

[1] M. Ahmed and R. Ward, "A rotation invariant rule-based thinning algorithm for character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1672–1678, Dec. 2002.
[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[3] 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.
[4] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," in *Proc. ACM Trans. Graphics (SIGGRAPH)*, vol. 22, San Diego, CA, 2003, pp. 303–312.
[5] H. Farid, "Image forgery detection," *Signal Process. Mag., IEEE*, vol. 26, no. 2, pp. 16–25, Mar. 2009.
[6] K. Hariharakrishnan and D. Schonfeld, "Fast object tracking using adaptive block matching," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 853–859, Oct. 2005.
[7] J. Hays and A. A. Efros, "Scene completion using millions of photographs," in *Proc. ACM Trans. Graphics (SIGGRAPH 2007)*, 2008, vol. 51, no. 10, pp. 87–94.
[8] C.-C. Hsu, T.-Y. Hung, C.-W. Lin, and C.-T. Hsu, "Video forgery detection using correlation of noise residue," in *Proc. 2008 IEEE 10th Workshop Multimedia Signal Process.*, Oct. 8–10, pp. 170–174.
[9] J. Jia, T.-P. Wu, Y.-W. Tai, and C.-K. Tang, "Video repairing: Inference of foreground and background under severe occlusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun./Jul. 2004, pp. 364–371.
[10] 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.
[11] 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, 2005.
[12] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 450–461, Sep. 2007.
[13] A. C. Kokaram and S. J. Godsill, "Joint detection, interpolation, motion and parameter estimation for image sequences with missing data," in *Proc. Int. Conf. Image Proc.* Oct. 26–29, 1997, vol. 2, pp. 191–194.
[14] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," in *ACM SIGGRAPH 2003*, vol. 22, pp. 277–286.
[15] F. Nielsen and R. Nock, "ClickRemoval: Interactive pinpoint image object removal," in *Proc. 13th Annu. ACM Int. Conf. Multimedia 2005*, pp. 315–318.
[16] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," in *Proc. 2005 IEEE Int. Conf. Image Process.*, Genova, pp. II, pp. 69–72.
[17] K. A. Patwardhan, G. Sapiro, and M. Bertalmío, "Video inpainting under constrained camera motion," *IEEE Trans. Image Proc.*, pp. 545–553, Feb 2007.
[18] S. Peleg, "Video mosaicing for non-chronological time editing," in *presented at the 2007 IAPR Conf. Mach. Vision Applications*, Tokyo, Japan, May 16–18, 2007.
[19] L.-M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Video Tech.*, vol. 6, no. 3, pp. 313–317, Jun. 1996.
[20] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg, "Evolving time fronts: Spatio-Temporal video warping," Tech. Rep. 2005-10, The Hebrew University of Jerusalem.
[21] T. K. Shih, N. C. Tang, W.-S. Yeh, T.-J. Chen, and W. Lee, "Video inpainting and implant via diversified temporal continuations," in *Proc. 2006 ACM Multimedia Conf.*, Santa Barbara, CA, Oct. 23–27, 2006, pp. 133–136.
[22] T. K. Shih, N. C. Tang, and J.-N. Hwang, "Ghost shadow removal in multilayered video inpainting," in *proc. IEEE 2007 Int. Conf. Multimedia Expo*, Beijing, China, Jul. 2–5, pp. 1471–1474.
[23] J. Sun, L. Yuan, J. Jia, and H. Y. Shum, "Image completion with structure propagation," in *Proc. ACM SIGGRAPH 2005*, vol. 24, no. 3, pp. 861–868.
[24] R. Urtasun, D. J. Fleet, A. Hertzmann, and P. Fua, "Priors for people tracking from small training sets," in *in Proc. 10th IEEE Int. Conf. Comput. Vis.*, Oct. 17–21, 2005, vol. 1, pp. 403–410.
[25] Y. Wexler, E. Shechtman, and M. Irani, "Space-Time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.
[26] Y. Zhang, J. Xiao, and M. Shah, "Motion layer based object removal in videos," in *Proc. Seventh IEEE Workshops Appl. Comput. Vis.*, 2005, pp. 516–521 (in the generated falsified video story).

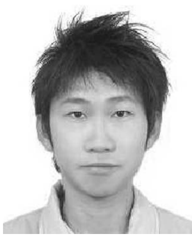
Timothy K. Shih (M'98–SM'01) received the B.S. degree from Tamkang University, Taipei, Taiwan, in 1983, and the M.S. degree from California State University, Chicaco, in 1985, and the Ph.D. degree from Santa Clara University, Santa Clara, CA, in 1993.

He is currently a Professor at the Department of Computer Science and Information Engineering, National Central University, Jhongli City, Taiwan. He was a Department Chair at the Department of Computer Science and Information Engineering, Tamkang University, Taiwan. He joined the Educational Activities Board of the Computer Society. He has edited many books and published more than 440 papers and book chapters, as well as participated in many international academic activities, including the organization of more than 60 international conferences. He was the Founder and Co-Editor-in-Chief of the *International Journal of Distance Education Technologies*. He has been invited to give more than 30 keynote speeches and plenary talks in international conferences, as well as tutorials in the IEEE International Conference on Multimedia and Expo 2001 and 2006 and the Association for Computing Machinery (ACM) Multimedia 2002 and 2007. His research interests include multimedia computing and distance learning.

Prof. Shih is a Fellow of the Institution of Engineering and Technology, a Senior Member of the ACM, and an Associate Editor of the ACM Transactions on Internet Technology and the IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES. He was also an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA. He has received many research awards, including research awards from the National Science Council of Taiwan, the International Institute for Advanced Studies Research Award from Germany, the Hellenic Society for Systemic Studies Award from Greece, the Brandon Hall Award from USA, and several best paper awards from international conferences.

Nick C. Tang received the B.S., M.S., and Ph.D. degrees from the Institute of Information Science, Academia Sinica, Taipei, Taiwan, in 2003, 2005, and 2008, respectively.

He is currently a Postdoctoral Fellow at the Institute of Information Science, Academia Sinica. His research interests include computer vision, image and video processing, and their applications.



Joseph C. Tsai received the B.S. and M.S. degrees from the Tamkang University, Taipei, Taiwan, in 2006 and 2008, respectively, where he is currently working toward the Ph.D. degree at the Department of Computer Science and Information Engineering.

His current research interests include computer vision, image and video processing, and their applications.

Jenq-Neng Hwang (F'01) received the B.S. and M.S. degrees from the National Taiwan University, Taipei, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree from the Signal and Image Processing Institute, University of Southern California, Los Angeles, in 1988, all in electrical engineering.

After two years of obligatory military services, he enrolled as a Research Assistant in 1985 at the Signal and Image Processing Institute, Department of Electrical Engineering, University of Southern California. From 1987 to 1989, he was also a Visiting Student at Princeton University, Princeton, NJ. In the summer of 1989, he joined the Department of Electrical Engineering, University of Washington, Seattle, where he was promoted to a Full Professor in 1999. From 2003 to 2005, he also served as the Associate Chair for the Research and Development, Department of Electrical Engineering. He is the author or coauthor of more than 200 published journal and conference papers and book chapters in the areas of image/video signal processing, computational neural networks, multimedia system integration, and networking. He is an Editor for the *Journal of Information Science and Engineering* and member the editorial board of the *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology.*

Prof. Hwang is an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He has also served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1992 to 1994 and the IEEE TRANSACTIONS ON NEURAL NETWORKS from 1992 to 2000; the Secretary of the Neural Systems and Applications Committee of the IEEE Circuits and Systems Society from 1989 to 1991; the Chairman of the Neural Networks Signal Processing Technical Committee of the IEEE Signal Processing Society from 1996 to 1998; the Society's representative to the IEEE Neural Network Council from 1996 to 2000; and was a member of Design and Implementation of Signal Processing Systems Technical Committee of the IEEE Signal Processing Society. He is also a founding member of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society. He was the Conference Program Chair of the 1994 IEEE Workshop on Neural Networks for Signal Processing held in Ermioni, Greece, September 1994; the General Co-Chair of the International Symposium on Artificial Neural Networks held in Hsinchu, Taiwan, in December 1995; chaired the tutorial committee for the IEEE International Conference on Neural Networks, Washington, DC, June 1996; the Program Co-Chair of the International Conference on Acoustics, Speech, and Signal Processing, Seattle, 1998; and served as the conference chairs for the International Association of Science and Technology for Development (IASTED) Signal and Image Processing Conference and IASTED Internet Multimedia Systems and Applications in 2006. He was the Special Session Co-Chair of the International Symposium on Circuits and Systems (ISCAS) 2008 and the Program Co-Chair of ISCAS 2009. He was the corecipient of the 1995 IEEE Signal Processing Society's Annual Best Paper Award in the area of neural networks for signal processing.