Video Completion by Motion Field Transfer

Takaaki Shiratori[†*] Yasuyuki Matsushita[‡] Sing Bing Kang[§] Xiaoou Tang[‡]

The University of Tokyo[†]
Tokyo, Japan
siratori@cvl.iis.u-tokyo.ac.jp

Microsoft Research Asia[‡]
Beijing , P.R.China
{yasumat,xitang}@microsoft.com

Microsoft Research[§]
WA, U.S.A.
sbkang@microsoft.com

Abstract

Existing methods for video completion typically rely on periodic color transitions, layer extraction, or temporally local motion. However, periodicity may be imperceptible or absent, layer extraction is difficult, and temporally local motion cannot handle large holes. This paper presents a new approach for video completion using motion field transfer *to avoid such problems. Unlike prior methods, we fill in missing video parts by sampling spatio-temporal patches of local motion instead of directly sampling color. Once the local motion field has been computed within the missing parts of the video, color can then be propagated to produce a seamless hole-free video. We have validated our method on many videos spanning a variety of scenes. We can also use the same approach to perform frame interpolation using motion fields from* different *videos.*

1. Introduction

Video completion refers to the process of filling in missing pixels or replacing undesirable pixels in a video. One useful application of video completion is restoration of damaged or vintage videos. This technology can also be applied in other areas: post-production in the movie-making industry (e.g., to remove unwanted objects), and restoration of corrupted internet video streams due to packet drops.

In this paper, we propose a new approach for video completion. Instead of transferring color/intensity information directly, our approach transfers motion field into missing areas or areas targeted for removal. The key idea is to use the local motion field of a video as its intrinsic representation. While previous video completion methods typically rely on color, our method relies on local motion information. We call our approach *motion field transfer*: it warps known local motion vectors to predict motion in the holes. This approach is particularly effective in cases where periodic motion (such as a person walking) is imperceptible or absent.

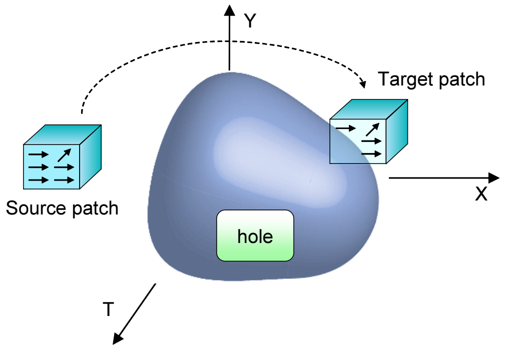

Figure 1. Illustration of motion field transfer. The spatio-temporal video hole is filled in by motion fields sampled from other portions of the video.

1.1. Related Work

Video completion can be considered as an extension of 2D image completion to 3D. The problem of filling in 2D holes has been well studied: image completion [10, 9, 15, 7, 22] and inpainting [2, 16]. (Inpainting approaches typically handle smaller or thinner holes compared to image completion approaches.) The 3D version of the problem, video completion [20, 14, 19, 23], has been getting increasing attention. Unfortunately, video completion is less effective if the video is treated as a set of independent 2D images [20, 18]. While the temporal independence assumption simplifies the problem, temporal consistency in the filled areas cannot be guaranteed.

One approach to ensure temporal consistency is to first separate the input video into foreground and background layers. Jia *et al.* [14] and Patwardhen *et al.* [19] separated the video into a static background and dynamic foreground, and filled the holes separately. Zhang *et al.* [23] used motion segmentation to achieve layer extraction. These methods

[*] This work was done while the first author was visiting Microsoft Research Asia.

work well if the layers are correctly estimated; however, it is difficult to obtain accurate layers in general, especially for scenes with complex motion.

Our method is inspired by Wexler *et al.*'s technique [20]. Their technique fills holes by sampling spatio-temporal patches from other portions of the video. In addition, global optimization is performed in order to enforce consistency in the holes. The method worked well in cases where the video contains periodic motion. The approach proposed by Matsushita *et al.* [18] also bears some similarity with our approach in that their method propagates the local motion field towards the missing image areas. However, their method is limited to temporally thin holes because their hole-filling process is done on a per-frame basis. Additionally, their propagation is based on simple diffusion, which cannot handle holes with complex motion fields well.

We borrow the fundamental idea of *motion warping* [21], in which motion parameter curves are warped for human body animation. Gleicher [12] and Cheung [6] adapted this idea to transfer articulated motion parameters to different objects. While these methods have been successful, motion warping has been limited to 3D targets which have the similar degrees of freedom. Instead of using high-level motion parameters, our method instead transfers low-level local motion field which can be estimated from videos. Haro *et al.* [13]'s work on transferring video effects has shown interesting results, including transfer of motion blur. Unlike their technique, however, we do not require input/output training pairs.

1.2. Overview of Our Approach

Our method is based on the assumption that motion information is sufficient to fill holes in videos. This assumption is valid if object motion is continuous in the video, which is usually the case. We regard the sequence of local motion field as an intrinsic video representation that is independent of color. By using motion field, we increase the chances of good matches for hole-filling. We need only match based on similarity in motion (regardless of color), which is surprisingly effective, as shown in our results.

Our approach consists of the following steps:

Local motion estimation. The pixelwise local motion vector field in the video is computed except at the holes.

Motion field transfer. The motion vectors in the holes are progressively estimated by sampling spatio-temporal patches of motion field from different portions of the video.

Color propagation. The motion field computed in holes is used to propagate color information from surrounding video pixels to finally complete the video.

The primary contribution of this paper is the motion field transfer step, which warps the local motion field for video completion. Why motion field instead of color? Fig. 2 illustrates the scenario where our motion-based method is effective while a color-based sampling method may fail. Here we have a common scenario where the foreground object is moving in front of a stationary background that has a unique distribution of colors. The color-based sampling technique has a matching problem similar to that of window-based stereo within the vicinity of object boundaries. The motion-based technique mimics motion observed elsewhere, and it works well because it does not rely on the distribution of color (which can be temporally discontinuous).

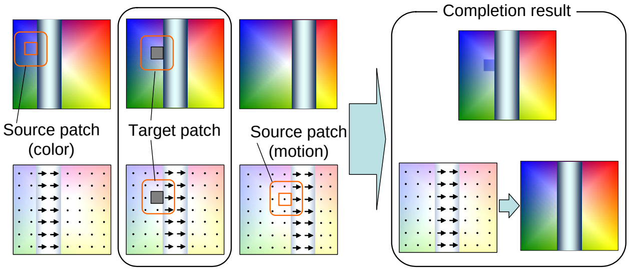

Figure 2. A simple example where motion based method performs better.

2. Video Completion Algorithm

This section describes how our proposed algorithm works.

2.1. Local motion estimation

The first step of our method is to compute the local motion field. To do this, we use the hierarchical Lucas-Kanade optical flow computation method [17, 3]. In a coarse-to-fine manner, the method estimates the motion vector $(u, v)^T$ that minimizes the error function

$$\arg\min_{(u,v)} \sum \left(u\frac{\partial I}{\partial x} + v\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \right), \tag{1}$$

where $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are image derivatives along spatial and temporal axes. We represent the estimated motion vector at point $\mathbf{p} = (x, y, t)^T$ in the video sequence by $(u(\mathbf{p}), v(\mathbf{p}))^T$.

2.2. Dissimilarity measure of motion vectors

Before we describe the next step (motion field transfer), we first define our patch dissimilarity measure for the motion data. Our video completion is based on non-parametric sampling of the motion field. Since the 2D optical flow can be viewed as a 3D vector in spatio-temporal domain with the constant temporal element being t, the 3D vector $\mathbf{m}$ is defined as $\mathbf{m} \equiv (ut, vt, t)^T$. We measure the distance between two motion vectors using the angular difference (in

3D space) as was done in Barron *et al*. [1]:

$$d_{\mathrm{m}}(\mathbf{m}_0, \mathbf{m}_1) = 1 - \frac{\mathbf{m}_0 \cdot \mathbf{m}_1}{|\mathbf{m}_0||\mathbf{m}_1|} = 1 - \cos\theta, \tag{2}$$

where θ is the angle between two motion vectors $\mathbf{m}_0$ and $\mathbf{m}_1$. This angular error measure accounts for differences in both direction and magnitude, since measurements are in homogeneous coordinates.

2.3. Motion Field Transfer

Using the dissimilarity measure defined in Eq. (2), the algorithm next seeks the most similar source patch given the target patch in order to assign the motion vectors to the missing pixels in the target patch. The dissimilarity between the source patch P_s and the target patch P_t is calculated by aggregating the dissimilarity measure over the patch (ignoring missing pixels in the target patch). Suppose the set of valid pixels in the target patch is $\mathcal{D}$; the aggregate distance between the source and target patches is then defined as

$$d\big(P_s(\mathbf{x}_s), P_t(\mathbf{x}_t)\big) = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{p}\in\mathcal{D}} d_m\big(\mathbf{m}(\mathbf{p}+\mathbf{x}_s), \mathbf{m}(\mathbf{p}+\mathbf{x}_t)\big), \tag{3}$$

where $|\mathcal{D}|$ is the number of defined pixels, $\mathbf{x}_s$ and $\mathbf{x}_t$ represent the position of the source and target patches, and $\mathbf{p}$ is the relative position from the center of each patch. Given the target patch P_t with its location $\mathbf{x}_t$, the optimal source patch $\hat{P}_s$ is obtained by finding the appropriate $\mathbf{x}_s$ which minimizes Eq. (3) as

$$\hat{P}_s(\hat{\mathbf{x}}_s) = \arg\min_{P_s(\mathbf{x}_s)} d(P_s(\mathbf{x}_s), P_t(\mathbf{x}_t)). \tag{4}$$

Once the optimal source patch $\hat{P}_s$ is found, the missing pixels in the target patch are filled by copying the motion vectors from the corresponding positions of the source patch.

The computation of the motion field transfer starts from the boundary of the holes, progressively advancing towards the inner holes. The holes are gradually filled with the new motion vectors which are copied from the source patches. Once the missing pixel is assigned a motion vector, the pixel is treated as a defined video pixel in the following computation. The order of fill-in, i.e., the order selection of the target patch, is determined by the number of non-hole pixels in the target patch, and the target patch with the highest number of non-hole pixels is first used for completion.

For efficiency, matching is done hierarchically through a Gaussian pyramid [5] of the video volume. Let l_m be the number of levels in the pyramid. Starting from the finest level $l = 1$, the coarser levels of the video volume are successively generated by convolving with a Gaussian kernel and sub-sampling. The patch size for matching in pyramid level l is set to $2^\beta \times 2^\beta \times 2^\beta$, where $\beta = l_m - l + 1$. The Gaussian kernel sigma used to blur one level to the next coarser level is set to one-fourth the patch size for matching [8], i.e., in our case, $2^{\beta-2}$.

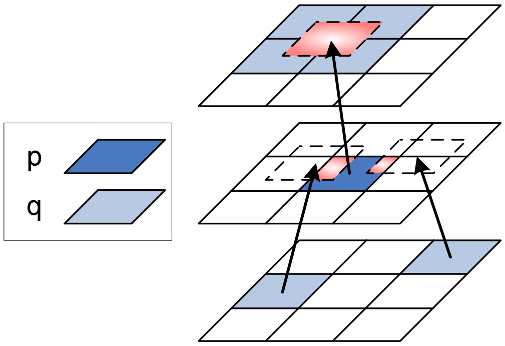

Figure 3. Illustration of color propagation. Color values of missing video pixels are computed by propagating the color information from the defined image pixels using the transferred motion vector field.

2.4. Color Propagation

Once the motion vectors have been found in the holes, color information is then propagated to generate the output video. The transferred motion vectors in the holes indicate the missing pixels' relationship with their neighbors; the motion vectors form a graph as shown in Fig. 3. Our color propagation method uses this graph to assign color values in the holes. We treat the motion vectors as undirected edges which represent pixel correspondences among frames.

Suppose we are to assign a color value to a pixel $\mathbf{p}$ using the connected pixels $\mathbf{q} \in \mathcal{N}$. Note that the edge originating from $\mathbf{p}$ may point to a fractional location in the neighboring frame as shown in Fig. 3. Similarly, a point $\mathbf{q}$ in the previous frame may be connected to a fractional location of pixel $\mathbf{p}$. We use the sizes of overlapped areas $s(\mathbf{p}, \mathbf{q})$ as weight factors to determine the contribution of neighboring pixels $\mathbf{q}$ to pixel $\mathbf{p}$. We also use the reliability of the edge $r(\mathbf{p}, \mathbf{q})$, which is measured by the inverse of dissimilarity measure defined in Eq. (3). The contribution from the neighboring pixel $\mathbf{q}$ to the pixel $\mathbf{p}$ is given by the product of r and s as

$$w(\mathbf{p}, \mathbf{q}) = r(\mathbf{p}, \mathbf{q}) s(\mathbf{p}, \mathbf{q}). \tag{5}$$

Thus, the color $\mathbf{c}(\mathbf{p})$ at pixel $\mathbf{p}$ is a weighted average of colors at the neighboring pixels $\mathbf{q}$:

$$\mathbf{c}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\mathcal{N}} w(\mathbf{p}, \mathbf{q})\mathbf{c}(\mathbf{q})}{\sum_{\mathbf{q}\in\mathcal{N}} w(\mathbf{p}, \mathbf{q})}. \tag{6}$$

Given n hole pixels, for each pixel $\{\mathbf{p}_i; i = 1, \ldots, n\}$ an equation is obtained from Eq. (6). Assuming there are m boundary pixels $\{\mathbf{p}^b_j; j = 1, \ldots, m\}$ with known colors, the n equations form the following linear system of equations:

$$\mathbf{C} = [\mathbf{W}|\mathbf{W}_b] \begin{bmatrix} \mathbf{C} \\ \mathbf{C}_b \end{bmatrix}, \tag{7}$$

where $\mathbf{C}$ is a $3 \times n$ matrix $\mathbf{C} = [\mathbf{c}(\mathbf{p}_1), \ldots, \mathbf{c}(\mathbf{p}_n)]^T$, $\mathbf{C}_b$ is a $3 \times m$ matrix $\mathbf{C}_b = [\mathbf{c}(\mathbf{p}_{b_1}), \ldots, \mathbf{c}(\mathbf{p}_{b_m})]^T$, and the $n \times n$

matrix $\mathbf{W}$ and $m \times n$ matrix $\mathbf{W}_b$ are given by

$$\mathbf{W} = \begin{bmatrix} 0 & w_{12} & \cdots & w_{1n} \\ w_{21} & 0 & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & 0 \end{bmatrix}, \mathbf{W}_b = \begin{bmatrix} w^b_{11} & \cdots & w^b_{1m} \\ \vdots & \ddots & \vdots \\ w^b_{n1} & \cdots & w^b_{nm} \end{bmatrix}.$$

Here w_{ij} represents the weight factor $w(\mathbf{p}_i, \mathbf{p}_j)$ after normalization, such that each row of $[\mathbf{W}|\mathbf{W_b}]$ sums to one. Therefore, w_{ij} falls in the range $[0, 1]$. The diagonal elements of $\mathbf{W}$ are all zero, since the motion vector never points the source pixel to itself. In order to obtain $\mathbf{C}$, Eq. (7) can be written as

$$\mathbf{C} = (\mathbf{I} - \mathbf{W})^{-1}\mathbf{W}_b\mathbf{C}_b, \tag{8}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. The matrix $(\mathbf{I} - \mathbf{W})$ is usually invertible, and the solution can be efficiently obtained by LU decomposition since the matrix is structurally symmetric and sparse. If the determinant of $(\mathbf{I} - \mathbf{W})$ is very small (indicating closeness to singularity), we compute its pseudo-inverse through singular value decomposition to obtain the least-squares solution.

3. Experimental Results

To validate our proposed method, we used it on a variety of videos. For all our experiments, a three-level Gaussian pyramid was used ($l_m = 3$).

3.1. Results of Video Completion

We have tested our method on 15 different videos, each containing a variety of motion and color.

Walking scene We first show the result for a video involving a walking person.The top row of Fig. 4 shows five frames of the original 80-frame input video of resolution 240×180. In the second row, the spatio-temporal hole created in the middle of the scene can be seen together with computed local motion field (overlaid). The volume of the spatio-temporal hole is about $12,000$ pixels, spanning five frames. Using motion field transfer, the hole is filled with the estimated local motion field as shown in the third row. Video completion is achieved by propagating color as shown in the bottom row. Our method was able to produce results that are visually identical to the ground truth.

Performer scene Fig. 5 shows the video completion result for the video '*performer*' captured with a moving camera.The resolution of the 60-frame input video is 352×240. The top row shows the original image sequence, and the corresponding image sequence with a spatio-temporal hole is shown in the middle row. The hole size is 72×120, spanning three frames. With our video completion method, the spatio-temporal hole is seamlessly filled as shown in the bottom row in Fig. 5. Fig. 6 is a close-up view of the result in the vicinity of the hole. From left to right, the figure shows (a) the ground truth, (b) the result of our method, (c) the intensity difference of the two images (b) and (a) in l^2 norm, (d) the result of the color-based method and (e) the intensity difference of (d) and (a) in l^2 norm. Although the result (b) looks slightly blurry due to the bilinear interpolation in the color propagation step, the structure of the moving object and background were well preserved. On the other hand, the color based method produces a significant amount of artifacts, especially around the leg, because the motion of the leg is not well observed in the image sequence. As we can see in the difference images (c) and (e) of Fig. 6, our method did significantly better than the color-based sampling method.

Object Removal Fig. 7 shows a useful application of video completion: object removal. Here, we manually removed the foreground person and automatically filled the hole using our motion-based technique. The size of the spatio-temporal hole is about $700,000$ pixels spanning about 60 frames. Even though there is complex motion in the background, the fill-in was accomplished well. There were noticeable blur effects, which was not surprising, since the hole is large. We briefly discuss the blur effect in Sec. 4.

Quantitative evaluation To further validate our proposed method, we performed a quantitative comparison between our method and the color-based non-parametric sampling method. For this evaluation, we used short video clips whose lengths range from 35 to 100 frames. We believe these videos are reasonably good representatives of videos captured by the average consumer. Fig. 8 shows the deviation from the ground truth for the two methods. The deviation is represented by the root-mean-squared (RMS) error in intensity. For all these 10 video samples, our method performed better than the color-based sampling method.

3.2. Another Application: Frame Interpolation

Fig. 9 shows another result of video completion for a *breakdance* scene. Unlike previous examples, here we recover *entire* frames (within dashed boxes). The video contains rather complex motions that are not periodic. Despite this, our method was able to interpolate frames by treating the new intermediate frames as holes.

Since the motion field is low-level information that is independent of color, we can extend our video completion method to fill holes in one video using motion fields from *another video*. This should work in principle if the two video sequences have similar motions (such as a person walking). This property suggests it is plausible to accumulate a video library of all types of motions for the purpose of general video completion.

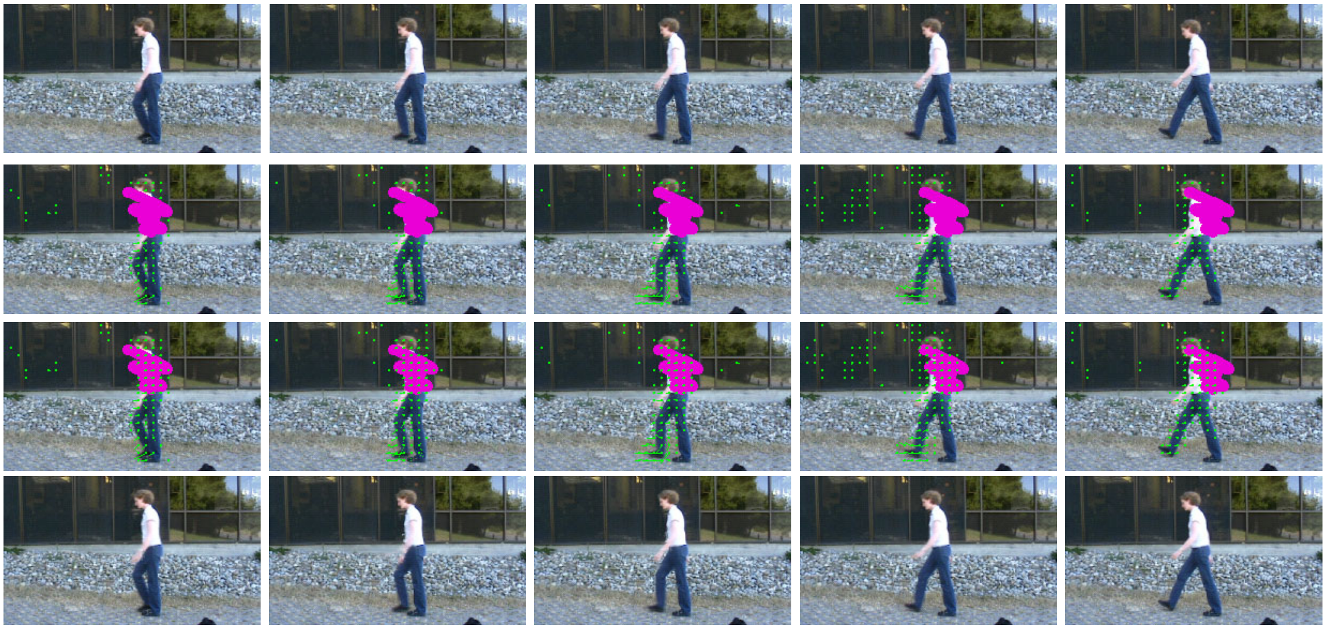

Figure 4. *Walking* scene result (five frames shown). Top row: original frames, second row: spatio-temporal hole and computed local motion field, third row: result of motion field transfer. bottom row: result of video completion. Notice the frames in the first and last rows are virtually identical.

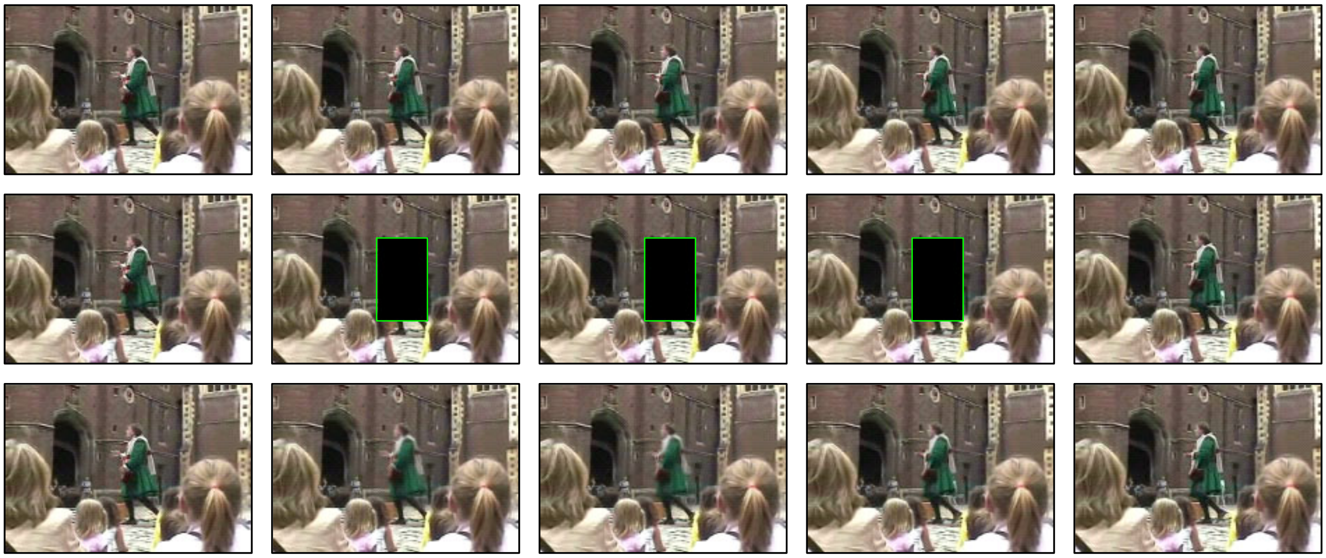

Figure 5. *Performer* video result (five frames shown). Top row: original image sequence, middle row: corresponding image sequence with a hole, and bottom row: result of video completion using our method.

The idea of using a precomputed set of motion fields as priors for video completion is particularly germane to internet video applications, where frames do get dropped occasionally due to heavy traffic. To prevent the video on the client side from appearing choppy, one can fill in the dropped frames using our video completion technique.

More specifically, we concentrated on the video chat scenario. This seems tractable because the motions in such a video are those of the human head, and they are rather limited. In our experiment, we pre-store the full-rate video as shown in the top row of Fig. 10. We use this video to help complete other videos.

To use the prior video, the local motion field of the prior video is pre-computed. Given a low-frame rate input video, we recompute the local motion in the prior video with the same frame rate as the input video. Motion field transfer is then performed to find out the most similar motion patch from the prior video. Once the patch is found, the

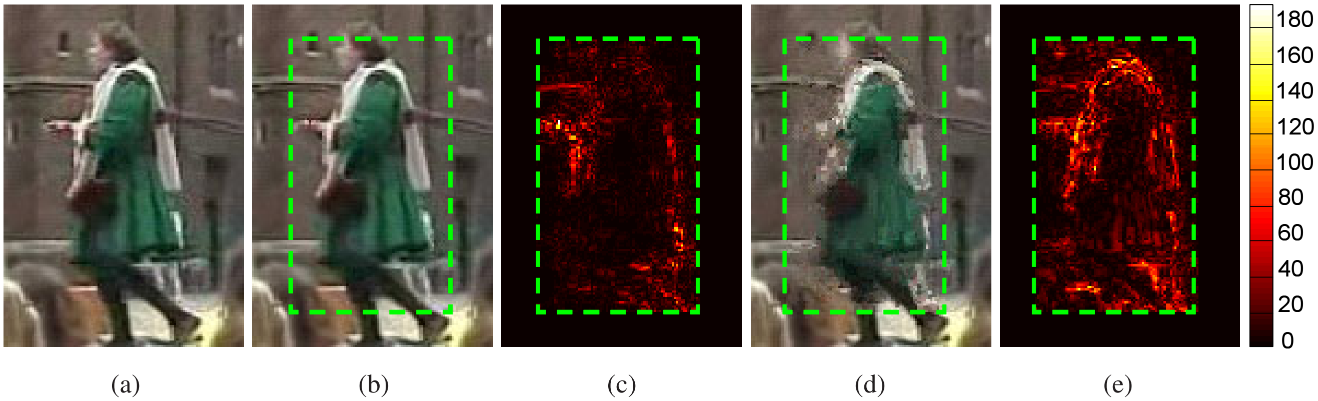

Figure 6. Close-up views. (a) Ground truth, (b) result of video completion using our method, (c) intensity difference between (b) and (a), (d) result of video completion using color-based sampling, and (e) intensity difference between (d) and (a). The dashed boxes indicate the location of the spatio-temporal hole.

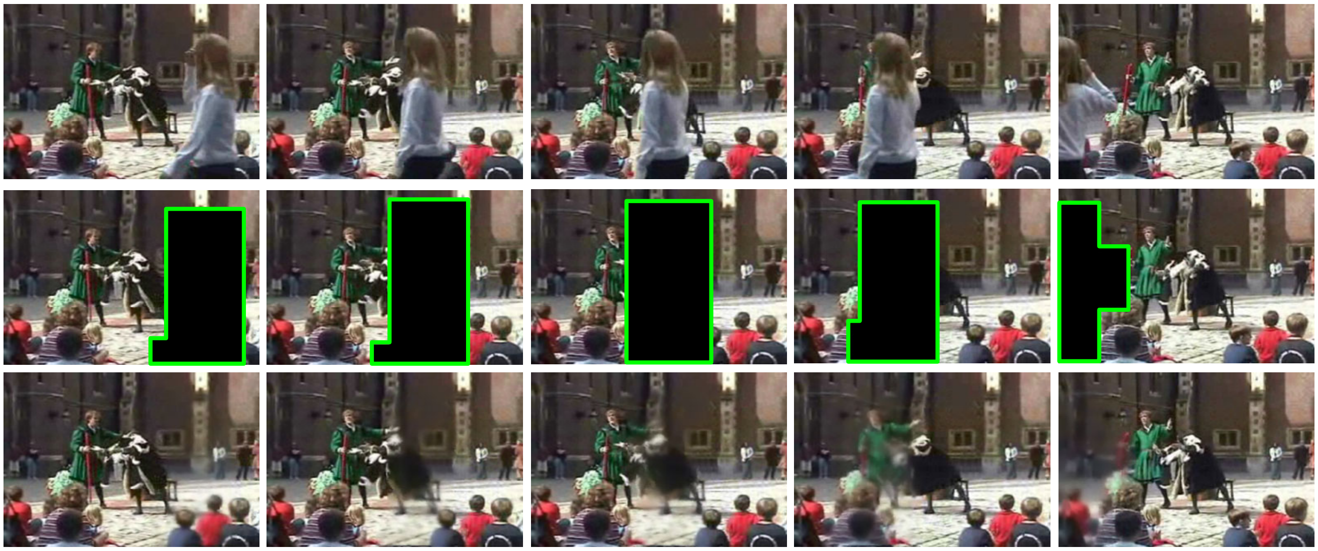

Figure 7. Object removal example (five representative frames shown). Top row: input video, where the two performers are occluded by a foreground person walking across the view. Middle row: after manually removing the foreground person. Bottom row: result of fill-in using our method.

full-resolution motion field is warped to the low-frame rate video to achieve a higher frame rate.

Fig. 10 shows the result of frame rate recovery. The prior video is captured at 30 frames per second (fps), and the frame rate of the input video is 15 fps. Although the color distributions in two videos are different, our method was able to seamlessly generate intermediate frames using motion field transfer. In this example, we only show the video chat scenario; however, the same technique can be used in other scenarios where the types of possible motion are predictable.

4. Discussion

While our motion field transfer approach to video completion has been shown to be effective for a variety of videos, it has limitations. For one, we are computing and comparing first-order derivatives of video data, and such measures tend to be more sensitive to noise than directly using color. However, we can alleviate the problem by prefiltering (if the extent of noise is somewhat known) and/or using more robust motion estimation algorithms (such as [11, 4]). As with any motion estimation techniques, large motions cannot be reliably extracted. Such an example is the running animal video in the rightmost column of Fig. 8; notice the large prediction error. Color-based methods would likely work better for videos with large motions that are periodic (the running animal video is not periodic).

Our method also tends to produce slightly blurry results due to bilinear resampling in the color propagation step. While better resampling techniques may be used (e.g., cubic interpolation), blur artifacts would still persist. To overcome this problem, one can apply a color-based sam-

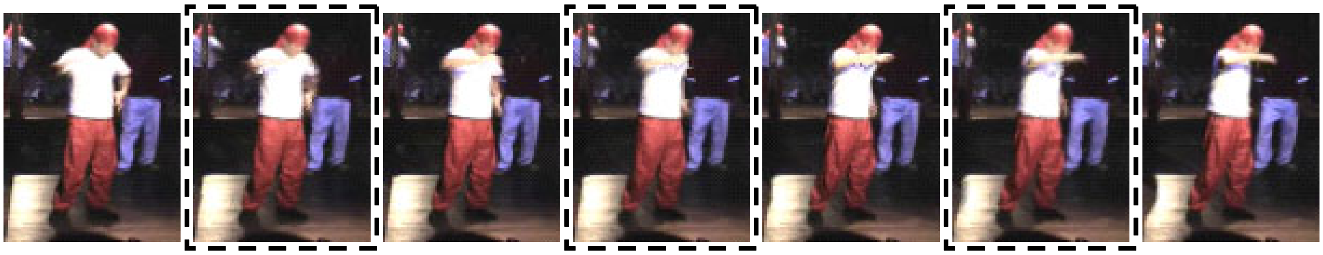

Figure 9. *Breakdance* scene result. The frames within dashed boxes were recovered by applying our video completion method to the entire set of missing frames.

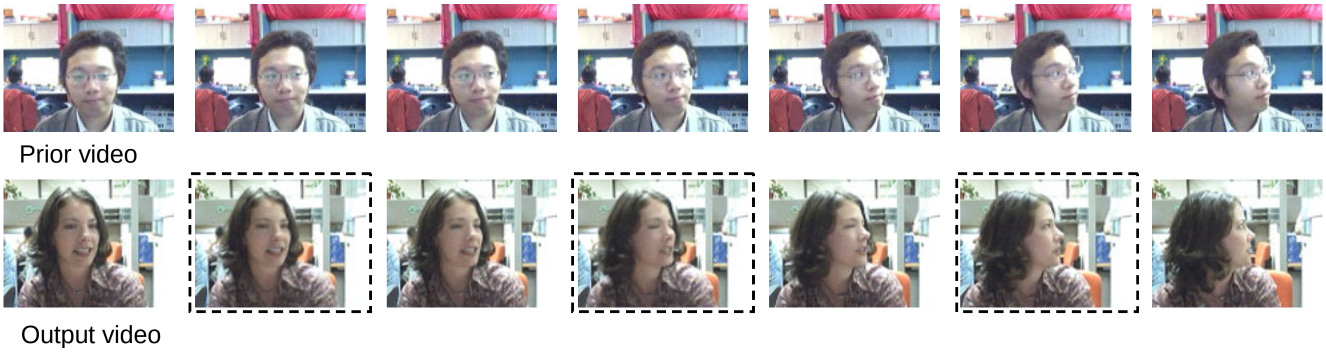

Figure 10. Result of frame rate recovery using the motion field prior. By transferring the motion field from the prior video, intermediate frames (within dashed boxes) that appear seamless can be synthesized. Note that here we are *doubling* the frame rate.

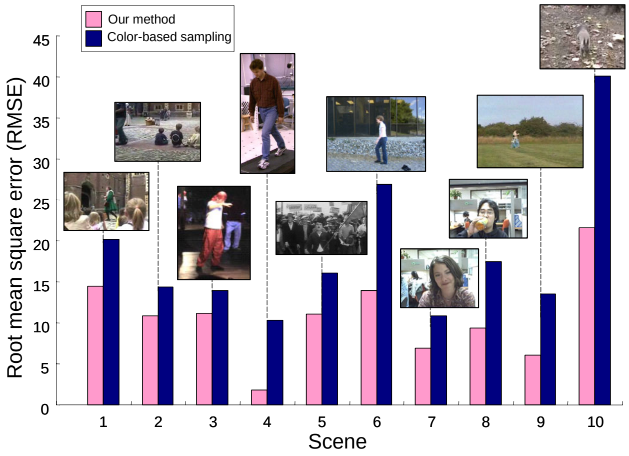

Figure 8. Reconstruction errors for our method and the color-based sampling method for 10 different videos.

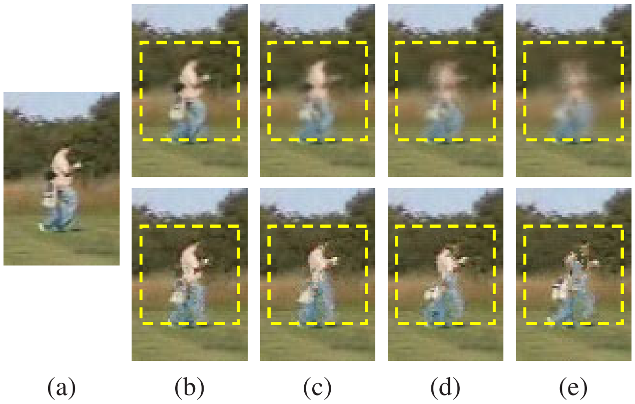

Figure 11. Effect of temporal size of hole on resulting blur. (a) Ground truth, (b-e), top row: results of our motion-based sampling with the hole set to 50 × 50 (within dashed boxes) spanning 2, 5, 7, and 10 frames respectively, (b-e), bottom row: results of color-based method under the same conditions.

pling pass to replace the estimated colors from the motion-based approach, since the color-based method produces directly copies the patches and thus avoids the blurring problem. This is the main idea behind the deblurring method of Matsushita *et al*. [18], which transfers sharper pixels from neighboring frames.

Directly copying pixels from one location to another ignores the effect of mixed pixels. The proper approach would be to separate colors (together with the blending factor or matting information) and move the colors independently, very much in the spirit of Zitnick *et al.*'s approach [24]. They simultaneously estimate motion vectors and matting information for frame interpolation specifically to handle the mixed pixel problem.

The color-based sampling method is expected to work better for videos with clear-cut periodic color variations and large motions. Both color-based and motion-based techniques have complementary strengths and weaknesses. This is illustrated in Fig. 11, which compares results for the video of walking people. As described earlier, our method produced blurry effects (depending on the temporal extent of the hole). On the other hand, the color-based method produced some missing parts, especially around the head, but parts that do show up look sharp. It seems logical to combine these two techniques. As future work, it would be interesting to investigate the effectiveness of this hybrid approach.

Computational Cost The most time-consuming part of our method is searching the source patches. Given a video of resolution $w \times h$ and length L frames, the number of matches performed for each target patch is roughly $w/4 \times h/4 \times L/4$ when $l_m = 3$. It took about 40 minutes to process the *performer* video ($w = 352$, $h = 240$, $L = 60$) using a P4 3GHz PC.

5. Conclusion

In this paper, we proposed the idea of *motion field transfer* for video completion, which involves sampling similar motion patches from different portions of the video. Motion field is an intrinsic information embedded in a video that is independent of color. Motion field transfer works better than conventional color-based non-parametric sampling techniques in cases where periodic motion is either imperceptible or absent. Our method has been tested on a number of videos spanning a variety of motion and color distributions, and has been shown to have significantly better performance over the color-based non-parametric sampling method.

In addition to hole-filling, we have shown results of seamless frame interpolation, using motion information within the same video and from another video.

References

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *Int'l Journal of Computer Vision*, 12(1):43–77, 1994. 3

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH 2000*, pages 417–424, 2000. 1

[3] J.-Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker: Description of the algorithm. In *Intel Research Laboratory, Technical Report.*, 1999. 2

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. Europ. Conf. on Computer Vision (4)*, pages 25–36, 2004. 6

[5] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, COM-31,4:532–540, 1983. 3

[6] G. K. M. Cheung, S. Baker, J. K. Hodgins, and T. Kanade. Markerless human motion transfer. In *Int'l Symp. on 3D Data Processing, Visualization and Transmission*, pages 373–378, 2004. 2

[7] A. Criminisi, P. Pérez, and K. Toyama. Object removal by exemplar–based inpainting. In *Proc. Computer Vision and Pattern Recognition*, volume 2, 2003. 1

[8] R. Deriche. Fast algorithms for low-level vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):78–87, 1990. 3

[9] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH 2001*, pages 341–346, 2001. 1

[10] A. A. Efros and T. Leung. Texture synthesis by non–parametric sampling. In *Proc. Int'l Conf. on Computer Vision*, pages 1033–1038, 1999. 1

[11] G. Farnebäck. Very high accuracy velocity estimation using orientation tensors parametric motion and simultaneous segmentation of the motion field. In *Proc. Int'l Conf. on Computer Vision*, pages 171–177, 2001. 6

[12] M. Gleicher. Retargetting motion to new characters. In *SIGGRAPH 98*, pages 33–42, 1998. 2

[13] A. Haro and I. Essa. Learning video processing by example. In *Proc. Int'l Conf. on Pattern Recognition (1)*, pages 487–491, 2002. 2

[14] J. Jia, T.-P. Wu, Y.-W. Tai, and C.-K. Tang. Video repairing: Inference of foreground and background under severe occlusion. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 364–371, 2004. 1

[15] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. on Graphics (SIGGRAPH)*, 22(3), 2003. 1

[16] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. In *Proc. Int'l Conf. on Computer Vision*, pages 305–312, 2003. 1

[17] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th Int'l Joint Conf. on Artificial Intelligence*, pages 674–679, 1981. 2

[18] Y. Matsushita, E. Ofek, X. Tang, and H.-Y. Shum. Full-frame video stabilization. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 50–57, 2005. 1, 2, 7

[19] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *Proc. Int'l Conf. on Image Processing*, 2005. 1

[20] Y. Wexler, E. Shechtman, and M. Irani. Space–time video completion. In *Proc. Computer Vision and Pattern Recognition*, volume 1, pages 120–127, 2004. 1, 2

[21] A. Witkin and Z. Popović. Motion warping. In *SIGGRAPH 95*, pages 105–108, 1995. 2

[22] Q. Wu and Y. Yu. Feature matching and deformation for texture synthesis. *ACM Trans. on Graphics (SIGGRAPH)*, 23(3):364–367, 2004. 1

[23] Y. Zhang, J. Xiao, and M. Shah. Motion layer based object removal in videos. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 516–521, 2005. 1

[24] L. Zitnick, N. Jojic, and S. Kang. Consistent segmentation for optical flow estimation. In *Proc. Int'l Conf. on Computer Vision*, pages 1308–1315, 2005. 7