Video Repairing under Variable Illumination Using Cyclic Motions

Jiaya Jia, *Member*, *IEEE Computer Society*,
Yu-Wing Tai, *Student Member*,
IEEE Computer Society,
Tai-Pang Wu, *Student Member*,
IEEE Computer Society, and
Chi-Keung Tang, *Member*,
IEEE Computer Society

Abstract—This paper presents a complete system capable of synthesizing a large number of pixels that are missing due to occlusion or damage in an uncalibrated input video. These missing pixels may correspond to the static background or cyclic motions of the captured scene. Our system employs user-assisted video layer segmentation, while the main processing in video repair is fully automatic. The input video is first decomposed into the color and illumination videos. The necessary temporal consistency is maintained by tensor voting in the spatio-temporal domain. Missing colors and illumination of the background are synthesized by applying image repairing. Finally, the occluded motions are inferred by spatio-temporal alignment of collected samples at multiple scales. We experimented on our system with some difficult examples with variable illumination, where the capturing camera can be stationary or in motion.

Index Terms—Video restoration, spatio-temporal consistence, illumination consistence, tensor voting, applications.

1 Introduction

We present a system for synthesizing video completion where the missing background and foreground are too large to be repaired by image inpainting [3] or video inpainting [2]. One feature of our system is that the repaired background and foreground maintain the necessary spatio-temporal and illumination consistency. This paper is inspired by work in video repairing and space-time video completion: Video repairing [15] infers occluded background and motion from a video captured using a static or moving camera. Space-time video completion [26] is an automatic approach using nonparametric patch-based sampling to synthesize missing motion. Without using any segmentation information, the completed patches may contain errors if the background is complex (e.g., nontextures) and the result will not preserve speed irregularity and may destroy complex structure if present in the moving object. This paper contributes to video repairing by using tensor voting [18] to address the pertinent spatio-temporal issues in background and motion repair. Moreover, variable illumination and moving camera are allowed. Our alternative approach employs user-assisted video segmentation, leaving the rest of the processing fully automatic. We assume a class of camera motions where the frames can be roughly registered by planar perspective transform. Our system has the following properties:

- Large static background and cyclic motion that are missing from the input video can be synthesized in the space-time volume. The synthesized cyclic motion can vary in velocity and scale in order to integrate seamlessly with the existing video.
- Spatial and temporal consistence are maintained in the synthesized video.
- Background with variable illumination can be handled uniformly.

1.1 Related Work

We first review in this section the related work in texture synthesis, inpainting, and techniques in completion and restoration for images and videos.

Texture synthesis. The nonparametric texture synthesis technique by Efros and Leung [12] performs matching and pixelwise synthesis to infer missing colors for regular texture images. In *Video textures* [19], dynamic programming is used to derive the best permutation of existing frames to synthesize a video of longer duration. Linear dynamic systems are used in *Dynamic textures* [10] to synthesize unstructured stochastic textures, such as smoke and fire.

Image inpainting and repairing. *Image inpainting* [3] is capable of filling small holes seamlessly. *Video inpainting* [2] uses [3] to perform frame-by-frame repair and, so, only small holes can be filled. Therefore, temporal alias manifested into distortion and flickering will be observed if the missing area is large. *Image repairing* [13] uses tensor voting and explicit segmentation information to synthesize missing pixels in a large hole. Adaptive scales of analysis are used. *Image completion* [11] also fills in large image holes. Without the use of explicit segmentation, however, image completion may break salient structures present in the image.

Video completion and repairing. Our goal is similar to that of Wexler et al. in their *Space-time video completion* [26], where the missing portions are filled in by sampling spatio-temporal patches from the input video. Their method defines and uses similarity measurement in space and time domains. Global space-time coherence is achieved by using an optimization framework. One key to the success of this method, as demonstrated by their example videos taken using a static camera, is the judicious extension of [12] in the use of nonparametric sampling to handle spatial and temporal information simultaneously. With the explicit use of segmentation information, our *Video repairing* [15] infers large moving motion by sampling and aligning *movels* (structured moving objects), which can be integrated into the existing part of the video. The camera can be static or moving. Missing static background is repaired by the construction of a layered mosaic and the application of image repairing [13]. An optimal alignment in terms of a homographic transform is computed to repair moving pixels and maintain spatio-temporal consistency. The moving pixels were assumed to be the projections of cyclic motions. Cyclic motions were analyzed in [21], where the capturing camera is moving and affine invariance was used to identify periodic motion from videos. Time-frequency analysis [8] was used for cyclic motion detection.

1.2 Our Approach

Fig. 1 gives an overview of our approach, where the sections that detail the semiautomatic or fully automatic processes are indicated. The rest of this paper is organized as follows: Section 2 details our background completion. In Section 3, our technique for motion completion is detailed. In Section 4, the results generated by applying Sections 2 and 3 are presented and discussed, followed by concluding remarks in Section 5. The conference version of this paper appears in [15], where the video repairing method described cannot handle variable illumination. In this

- *J. Jia is with the Department of Computer Science and Engineering, the Chinese University of Hong Kong, Room 1018, HO Sin-Hang Engineering Bldg., Shatin N.T., Hong Kong. E-mail: leojia@cse.cuhk.edu.hk.*
- *Y.-W. Tai, T.-P. Wu, and C.-K. Tang are with the Department of Computer Science, the Hong Kong University of Science & Technology, Clear Water Bay, Hong Kong. E-mail: {yuwing, pang, cktang}@cs.ust.hk.*

Manuscript received 28 Aug. 2004; revised 21 July 2005; accepted 22 Aug. 2005; published online 13 Mar. 2006.
Recommended for acceptance by H. Sawhney.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0454-0804.

0162-8828/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

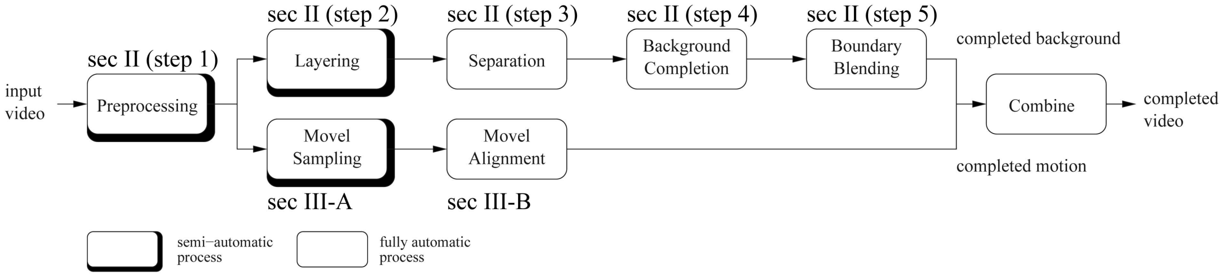

Fig. 1. Overview of our space-time video repairing/completion.

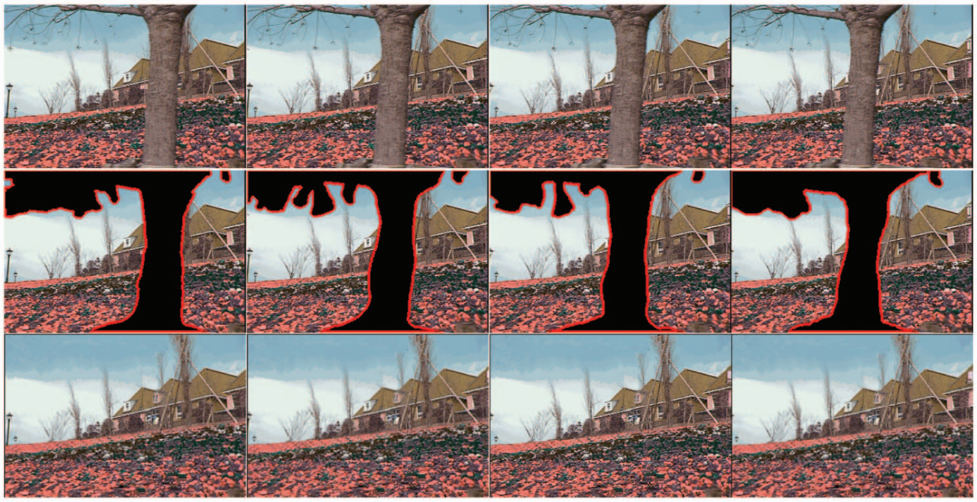

Fig. 2. Top: Sample frames from input video. Middle: Video with the occluder removed. Bottom: Space-time completion of the background video.

paper, we generalize the approach by addressing pertinent issues and presenting related new results.

2 Completion of Static Background under Variable Illumination

To simultaneously restore the missing background of a damaged video captured by a static or moving camera without visible distortion and to enforce the necessary temporal consistency when the camera motion is smooth, we adopt the layered mosaics approach. Inspired by geometric proxies used in image-based rendering systems for rendering antialiased novel views when the scene is cluttered and complex [24], [4], [1], we first segment the scene into layers. As our input is a video, the inherent challenge is an efficient method to perform multiimage segmentation. Similar to [23], a user-assisted key frame approach is adopted where a layer in video frames is regarded as a 3D image patch consisting of similar features for representing a semantically meaningful object. By segmenting the background into similar layers with depth ordering where the pixels in each layer have consistent motion, the temporal consistence inherent in a complex scene can be maintained in the completed video. We describe each step of our background completion as follows:

1. *Preprocessing*. To remove the occluder, a few key frames (e.g., every other 10 frames) are chosen to mask it off manually. The resulting holes in these key frames are tracked and located in all the remaining frames by the mean shift tracking algorithm [7]. Fig. 2 show some sample frames from the input and the resulting video after removing the foreground object. Note that only a rough specification is needed.
2. *Layer segmentation and propagation*. To maintain the temporal consistency of the restored background, we segment the video into layers with consistent motion. For example, for the middle images in Fig. 2, there are two layers with overlapping boundaries: a layer for the sky and the house, and a layer for the flower bed. The user roughly specifies the layer boundaries for the scene on the same key frames. These boundaries will be automatically propagated to the remaining frames by the mean shift tracker.
3. *Color and illumination separation*. A pixel in each segmented layer can be regarded as the composition of the corresponding intrinsic color and illumination components. For a static camera, intrinsic image separation such as [25] can be used to perform the decomposition. To handle the more complex case of a moving camera, we first perform image registration with pixel intensity normalization by tensor voting [14], which performs exposure correction with image registration. A reference mosaic for each layer is constructed. The original pixels before intensity normalization are projected onto the corresponding location on the mosaic so that [25] or other techniques for intrinsic image separation can be applied. Thus, the case of a moving camera can be reduced to that of a static mosaic when the input frames are related by homographic transformation.
4. *Completion of the background and illumination videos*. After the reference mosaic has been constructed for each layer with separated pixel colors and illumination, we apply image repairing [13] to infer, respectively, the missing pixel colors and illumination in the hole area of each layer in the reference mosaic. Fig. 3 shows one example, where the two completed components of the sample frames are depicted in the middle rows.
5. *Boundary blending*. After completing the background, we perform homography blending [15] to reduce the flickering caused by misregistration among frames and to achieve better temporal coherence:

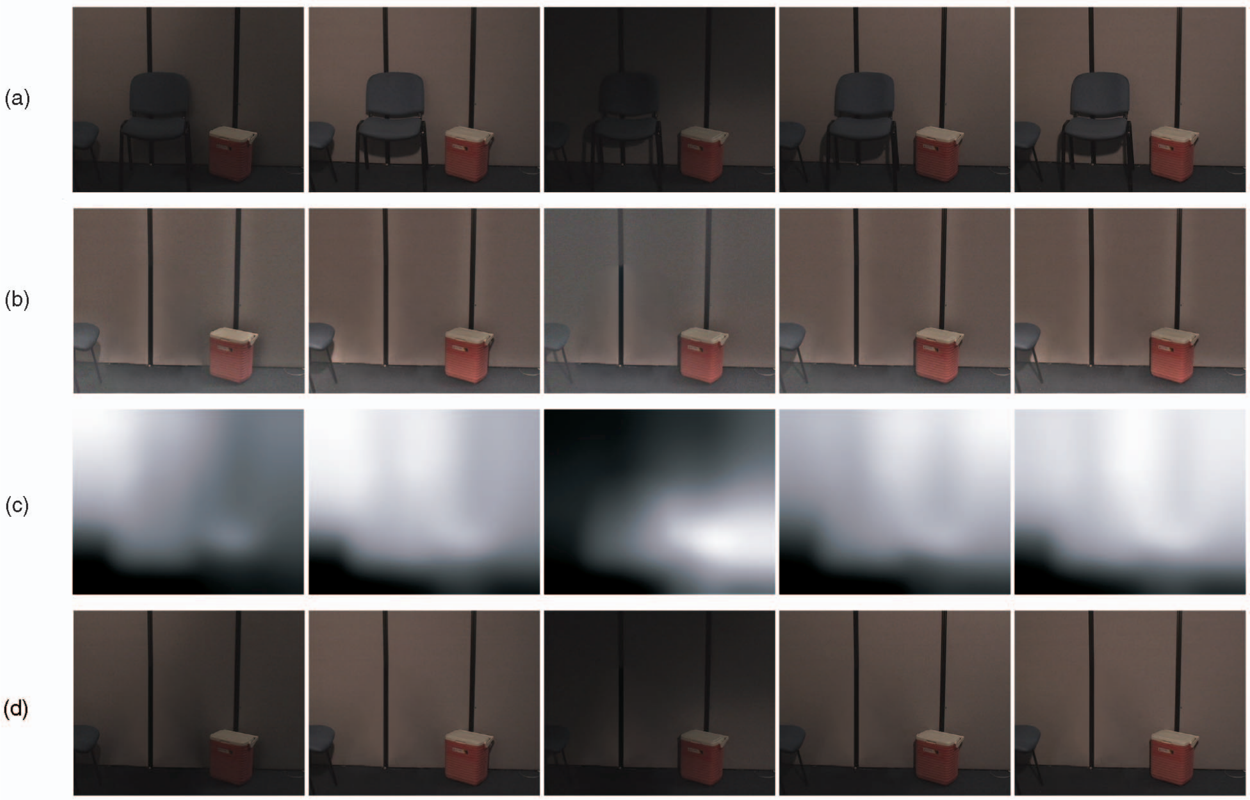

Fig. 3. *Chair and Chest*. Video completion of static background under variable illumination. Five sample frames are shown in each row. (a) Input video. (b) Completed background video (intensity is normalized). (c) Completed illumination video. (d) Final result after combining the repaired background and the illumination videos. (See the supplementary video, which can be found at http://computer.org/tpami/archives.htm.)

a. Let M_1 and M_2 be two adjacent layers and let M_3 be the overlapping area along their boundary. Let $\mathcal{H}_1^{-1}$ and $\mathcal{H}_2^{-1}$ be the homography that, respectively, warps $M_1 \cup M_3$ and $M_2 \cup M_3$ to the focal plane of the reference mosaic. We blend the homographies $\mathcal{H}_1$ and $\mathcal{H}_2$ for M_1 and M_2, respectively, and create $\mathcal{H}_3 = \alpha\mathcal{H}_1 + (1-\alpha)\mathcal{H}_2$ for M_3, where α is the blending coefficient, which is a function of the distance to the overlapping boundary on the reference mosaic.

b. A repaired frame is produced by projecting the mosaic to layer M_i by $\mathcal{H}_i$, $i = 1, 2, 3$.

By blending homographies instead of pixel colors, we reduce the abrupt change in color value along the layer boundary. If there are more than two layers, we generalize the above by processing two layers at a time, followed by merging other intermediate results. A hierarchical structure is used to store the video layers.

Fig. 2 shows the result of completing the background of the flower garden sequence. Fig. 3 shows an example with variable illumination for the background. Pixelwise multiplication is performed between the repaired illumination video and the repaired background video to produce the completed background video in Fig. 3d.

3 Motion Completion

The completed motion should maintain spatial and temporal consistency across all repaired frames if the camera motion is smooth. One reasonable constraint for large motion completion is motion periodicity. We encode this knowledge by sampling periodic motion in a video: This corresponds to the *movel sampling phase* (Section 3.1). To repair a motion, we observe that the missing pixels inside a hole can be synthesized if we know which part of the cycle is missing. Computationally, it translates into our *movel alignment phase* to perform pixel synthesis (Section 3.2). Our movel alignment is similar to [5], which also adopts an alignment scheme, processing the data in a coarse-to-fine warping framework. The technical advance we make in this paper is that moving cameras and dynamic scenes can be handled.

Let us define some terminologies here. A *movel* is a moving element, corresponding to a set of moving pixels segmented in each frame. A *sample movel* is a movel which contains at least one cycle of the periodic motion (e.g., Fig. 4a). If some frames in a movel are damaged, we call it a *damaged movel* (e.g., Fig. 4b). Our problem is thus reduced to one of aligning the sample movel with the damaged movel in order to repair the latter (e.g., Fig. 4c).

After we have collected the sample movel and located the damaged movel, we perform the following automatic preprocessing: *movel wrapping*, *movel regularization*, and *movel stabilization*. In essence, movel wrapping removes undesirable motion discontinuity in the sample movels. Movel regularization maintains temporal coherence of the repaired movel. Movel stabilization reduces the search space and alleviates the velocity mismatch problem.

3.1 Phase 1: Movel Sampling

One requirement of the synthesized motion is to maintain the necessary spatio-temporal consistency of the periodic motion. To sample a movel, the statistics of the scene background are first collected. The statistics are the mean and variance of the intensity of each pixel (x, y) in the field of view, which can be obtained directly from the input video. For a moving camera, (x, y) refers to the image coordinates of the resulting reference mosaic. After obtaining the pixel statistics, moving pixels are detected by comparing the intensity of the current frame with the collected statistics. Connected components are then constructed for the moving pixels. After

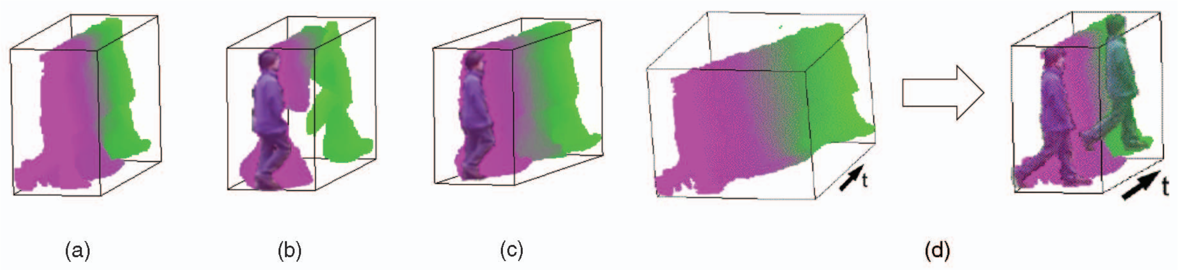

Fig. 4. (a) Sample movel, (b) damaged movel, (c) repaired movel, (d) sample movel after wrapping and stabilization.

removing isolated noise, a video mask is produced, which is used to sample a movel. The time-frequency analysis is applied to detect and characterize the periodicity from the input video after extracting moving pixels [8]. Note that both stationary (i.e., periodicity with statistics that do not change with time) and nonstationary periodicities can be handled by this method.

If there are multiple motions, multiple movels will be extracted. Interactive segmentation [17] is used in case the above simple segmentation does not work well. The video matting method proposed in [6] can also be used to produce a matte for moving the foreground so that it can be extracted from the video.

3.1.1 Movel Wrapping

We assume that the sample movels contain at least one motion cycle and that the damaged movels can be repaired by using the data embedded in the sample movels. A number of sample movels are first concatenated so that damaged movels with a large number of missing frames can be repaired. In the concatenation process, the first and the last frames of a sample movel may not be the same. They should be morphed to create a natural transition for seamless looping. We use 3D *tensor voting* to automatically infer a smooth *surface* in the spatio-temporal volume.

In our implementation, we use the last five frames and first five frames of a period of a sample movel to infer their natural connection. The video masks for these frames are used to extract the 2D boundaries of a movel. Let P_t be the corresponding set of moving pixels in a movel for frame t. Let ∂P_t be the boundary pixels of P_t. Denote the space-time volume resulted by superimposing the boundary pixels of the 10 frames along the temporal axis by

$$Boundary = \cup_{t=n-4\cdots n,1\cdots 5} \partial P_t, \qquad (1)$$

where n is the total number of frames in the sample movel. *Boundary* is used as an input to the 3D tensor voting [18] to vote for a *surface* in the spatio-temporal domain, which infers the in-between movel boundary that optimally and smoothly connects the first and the last frame in the period:

$$Surf = SurfaceExtract(TensorVoting(Boundary)), \qquad (2)$$

where $SurfaceExtract(\cdot)$ is a surface extraction procedure to produce an implicit surface representation $Surf$ by tensor voting [18]. $Surf$ is therefore used to represent the space-time boundary $Boundary$ of the movel. Finally, view morphing [20] is applied to infer the color of the pixels inside the space-time volume bounded by $Surf$.

3.1.2 Movel Regularization

In this step, we process the repaired movel to preserve the inherent temporal coherence by regularization. Again, we make use of 3D tensor voting [18]. Fig. 5a shows the centroids of all connected components in a movel. In a damaged movel, we only compute centroids for the frames where image holes are absent (Fig. 5c). Note that, even for smooth camera motion, because the centroids are found individually in each frame, the corresponding path along the temporal axes are not smooth.

Let $Centroid = \{(x, y, t)\}$ be the set of all centroids in the wrapped sample movel. Three-dimensional tensor voting [18] is used to vote for a smooth trajectory in the spatio-temporal domain, which implicitly enforces the desired spatio-temporal coherence:

$$Curve = CurveExtract(TensorVoting(Centroid)), \qquad (3)$$

where $CurveExtract(\cdot)$ is the curve extraction procedure in tensor voting which produces a smooth 3D space curve $Curve$.

For a damaged movel, a lot of centroids are missing. To regularize a damaged movel, we perform multiscale tensor voting for (3) and vote for the curve in multiple scales. This is done by prefiltering and subsampling the centroids in each pass of tensor voting. Therefore, a large gap can be filled and new centroids are inferred after tensor voting.

3.1.3 Movel Stabilization

This step translates the segmented moving pixels of all frames in a movel such that each centroid is at the image center after the translation. Movel stabilization provides the following benefits: 1) In

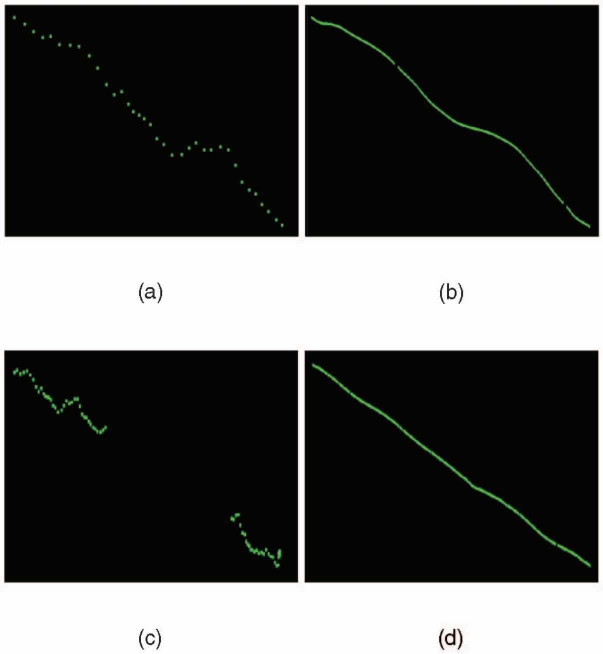

Fig. 5. Regularizing a sample movel (a) and (b) and a damaged movel (c) and (d) by 3D tensor voting. In each figure, the temporal axis is horizontal. The vertical axis is one of the spatial axes. (a) and (c) show the centroids of the connected components in all frames before regularization. (b) and (d) show the regularized centroids obtained by curve extraction using 3D tensor voting. When a large gap is present in a damaged movel (c), hierarchical 3D tensor voting is used.

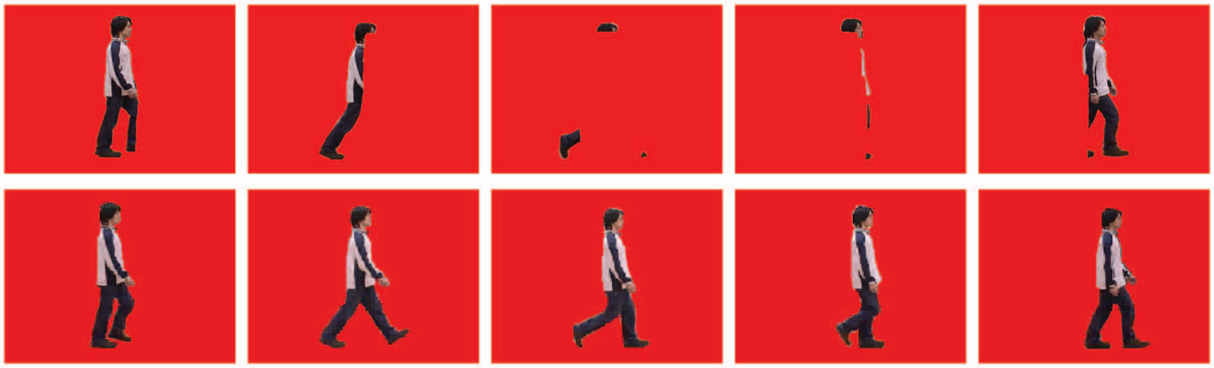

Fig. 6. An optimal 4×4 homographic transform is computed to align a sample movel and a damaged movel. Top: Damaged movel. Bottom: Repaired movel.

movel alignment (Section 3.2), Levenberg-Marquardt minimization converges much faster if we translate both the sample and damaged movels to the image center as the total number of pixels (in the spatio-temporal volume) to be processed in movel alignment can be significantly reduced. 2) The velocities of the sample movel and the damaged movel are not the same, in general, which makes the initial guess difficult to set. Movel stabilization reduces the search space in the Gaussian pyramid used in the multiscale processing.

Using the first frame at time t_0 as a reference, all other frames in a movel are translated such that their regularized centroids (existing or inferred) lie on a straight line parallel to the temporal axis at the frame center (Fig. 4d). The 2D translation for the respective frame is simply $(\delta x, \delta y) = c_t - c_{t_0}$, where c_t is the centroid of frame t. Both the sample movel and damaged movel will be stabilized before the movel alignment in the next phase.

3.2 Phase 2: Movel Alignment

Our movel alignment method is similar to the image registration method proposed in [22]. Movel alignment is performed in the 4D homogeneous coordinates. A homographic transform in the space-time domain will be estimated. In [22], the initial translation is input manually, while, in our movel alignment, we use the phase correlation to automatically estimate the initial translation from the damaged movels to sample movels in the 3D space-time coordinates by Mellin transform [9] because movels are already regularized and stabilized.

Our algorithm supports a subclass of camera and object motions. For example, we cannot recover a rotated face that has not been sampled in a sample movel. The subclass of camera motion we handle consists of transformation expressible by homography in the spatio-temporal space.

Let (x, y, t) and (x', y', t') be the respective movel coordinates before and after alignment. In our implementation, we concatenate a number of wrapped sample movels before computing the alignment with the damaged movel in order that missing motion larger than one cycle can be repaired. The concatenated sample movels and the damage movel can be aligned by a 4×4 homographic transform $\mathbf{H}$:

$$\mathbf{x}' = \mathbf{H}\mathbf{x}, \tag{4}$$

where $\mathbf{x}' = [x'\, y'\, t'\, 1]^T$ and $\mathbf{x} = [x\, y\, t\, 1]^T$.

The problem is now reduced to the estimation of $\mathbf{h} = [h_k]$, $0 \leq k \leq 15$. To speed up the estimation and to avoid local minimum, we can turn off some parameters because rotation is not allowed. The upper 3×3 submatrix is made an identity (or diagonal) matrix if the motion to be repaired only involves translation (and scaling, in addition). We minimize the squared intensity errors in the volume. Let $\mathbf{I}$ be the sample movel and $\mathbf{I}'$ be the aligned damaged movel, we define the error term in the overlapping volume of $\mathbf{I}$ and $\mathbf{I}'$ as follows:

$$E = \sum [\mathbf{I}'(x', y', t') - \mathbf{I}(x, y, t)]^2. \tag{5}$$

We perform the optimization by the Levenberg-Marquardt iterative minimization algorithm. The intensity gradient $(\frac{\partial \mathbf{I}'}{\partial x'}, \frac{\partial \mathbf{I}'}{\partial y'}, \frac{\partial \mathbf{I}'}{\partial t'})^T$ is computed at each voxel (x', y', t'). The Hessian matrix $\mathbf{A} = [a_{kl}]$ and weighted gradient vector $\mathbf{b} = [b_k]$ are computed: $a_{kl} = \sum \frac{\partial e}{\partial h_k} \frac{\partial e}{\partial h_l}$, $b_k = -\sum e \frac{\partial e}{\partial h_k}$, where the partial derivative of $e = \mathbf{I}'(x', y', t') - \mathbf{I}(x, y, t)$ with respect to $h_k, 0 \leq k \leq 15$ is computed. Then, we update $\mathbf{h}$ by $\delta \mathbf{h} = (\mathbf{A} + \beta \mathbf{I})^{-1} \mathbf{b}$ and $\mathbf{h}_{m+1} \leftarrow \mathbf{h}_m + \delta \mathbf{h}$, where β is a time-varying parameter. This method is similar to [22], except that the sampling on the temporal axis is different from that of the spatial axis in our spatio-temporal alignment: Instead of resampling along the time axis, note that our alignment method allows scale change of the movels along the x, y, and t axes. Also, because a movel has been stabilized, we can handle velocity mismatch between the sample and the damaged movels in our alignment.

Estimation efficiency can be significantly enhanced by constructing a Gaussian pyramid on the sample movel and the damaged movel, which are first converted into gray levels. The highest resolution is the sampling resolution of the movels. Lower resolution levels are obtained by prefiltering the 3D spatio-temporal data, followed by subsampling using a factor of two along the spatio-temporal axes. After automatic initialization (by phase correlation), the Levenberg-Marquardt optimization is executed to refine the warping transform. Typically, the optimal $\mathbf{H}$ converges within 20 iterations if rotation is not considered. The detailed algorithm can be found in [15]. Note that a similar hierarchical alignment on 2D images was also found in Caspi and Irani [5].

Fig. 6 shows the efficacy of our movel alignment method by using an example. Note that, in 3D space, when misalignment is present, simple blending easily generates a blurry artifact. In our method, we apply the graph cut algorithm, similar to that used in texture synthesis [16], to search for a least expensive cut in the overlapping volume of the sample and damaged movels to reduce any blurry or popping artifacts if exist.

Finally, recall that movel stabilization is performed before movel alignment. After repairing the damaged movels, we translate the repaired movels back to their original positions using the regularized centroids. Since all centroids have been regularized and the repaired frames are restored from the regularized sample movel, the resulting repaired frames exhibit the necessary spatio-temporal coherence.

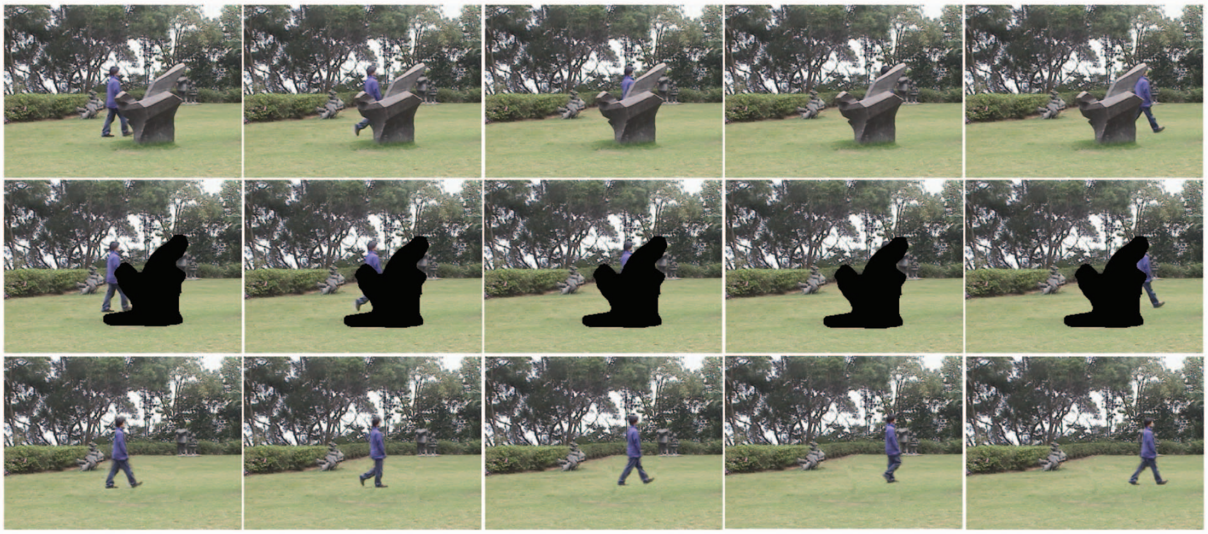

Fig. 7. *Sculpture* sequence: sample frames from the input video, damaged video, and repaired video. The camera is static. Movels are sampled automatically. (See the supplementary video, which can be found at http://computer.org/tpami/archives.htm.)

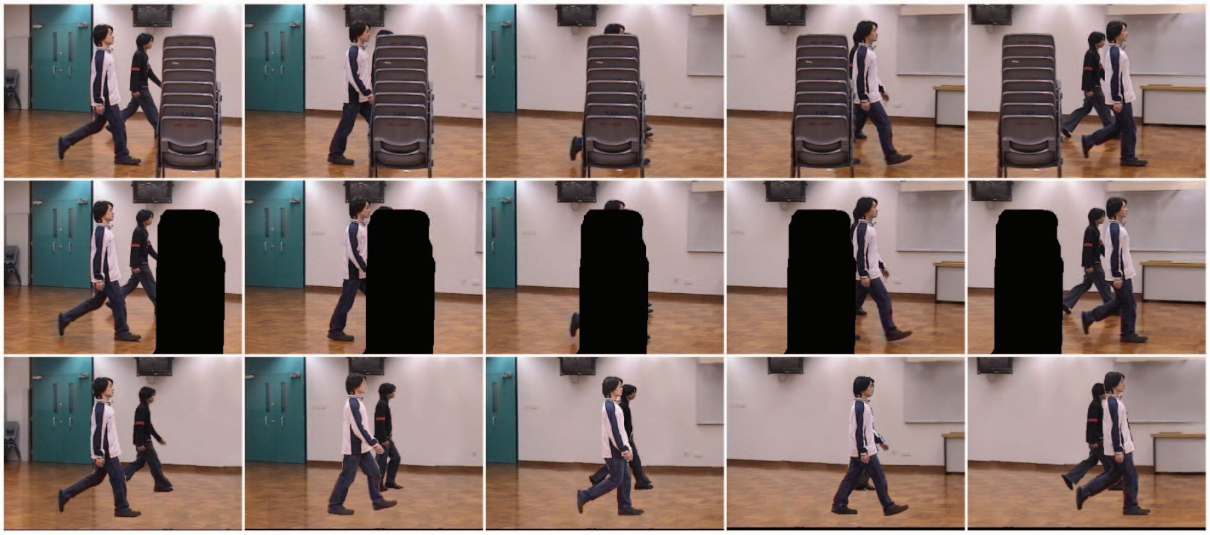

Fig. 8. *Chairs* sequence: sample frames from the input video, damaged video, and repaired video. Moving camera with multiple motions at different speed. Movels are sampled automatically. (See the supplementary video, which can be found at http://computer.org/tpami/archives.htm.)

4 Results

Fig. 7 shows the result on the *Sculpture* sequence. Five sample frames from the input video, the damaged video (with the sculpture removed), and the completed video produced by our system are shown. In particular, we are capable of repairing the motion in the fourth frame of Fig. 7, where the character is almost occluded by the sculpture in the input video. The camera is stationary in this example. All the missing motions and the previously occluded background are also repaired. Note the continuity between the synthesized motion and the existing motion when the restored video is played.

Fig. 8 shows a more challenging sequence, *Chairs*, where the camera is moving and multiple motions at different speeds are present. This result shows the strength as well as some limitations of our video repair system, which indicates areas for future research. Here, the camera is moving. There are multiple motions in the input video. The near character walks faster than the far character. The far character is occasionally occluded by the chairs and the near character. Both motions are completely occluded in the third frame. All background and motions are repaired by our method with acceptable visual quality, except that the shadows of the characters cannot be synthesized since we do not sample shadows in movels.

Fig. 3 shows the repaired result of *Chairs and Chest*. The camera is stationary. A large region was removed from every frame of the input video. No pixels in the damaged video can be used to repair the hole. Our method can synthesize the missing pixels and preserve the necessary spatial, temporal, and lighting consistencies.

Fig. 9 shows the result on *Mug and Snoopy Toy*. A moving camera was used. The moving snoopy toy was occluded by a large mug hanging from above. The object extraction is precise, and is accomplished by using Lazy Snapping [17]. Note that our system can tolerate some specular highlight here, which lies on a plane. However, the highlight on the moving snoopy toy, which is a glossy object, is not preserved. This is caused by our approach in which a movel is sampled and aligned in the separated color video, which is presumably under relatively constant illumination after separation. The repaired frame is then combined with the repaired illumination video.

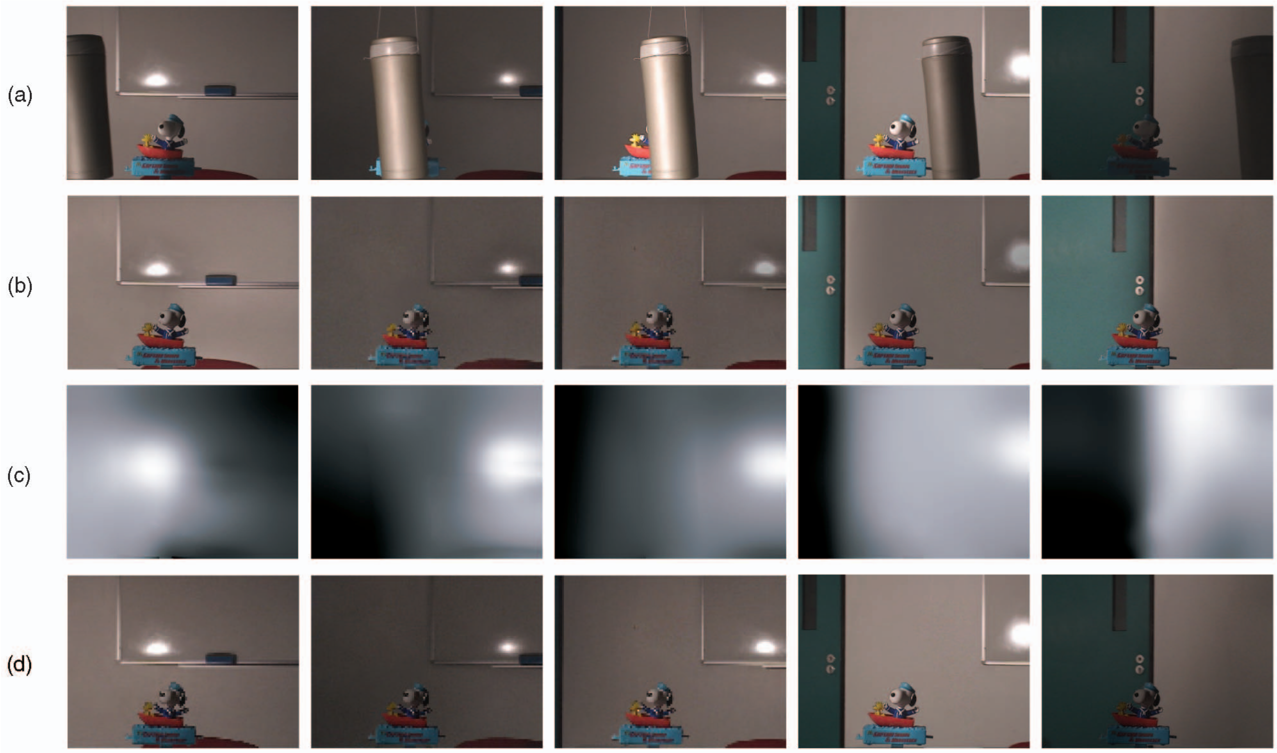

Fig. 9. *Mug and Snoopy Toy*. Video completion with a moving camera and an occluded moving object under variable scene illumination. Movels are sampled by the user-assisted method. Five samples frames are shown. (a) Input video. (b) Completed background video (intensity normalized). (c) Completed illumination video. (d) Final result after combining the repaired movel, color, and illumination video. (See the supplementary video, which can be found at http://computer.org/tpami/archives.htm.)

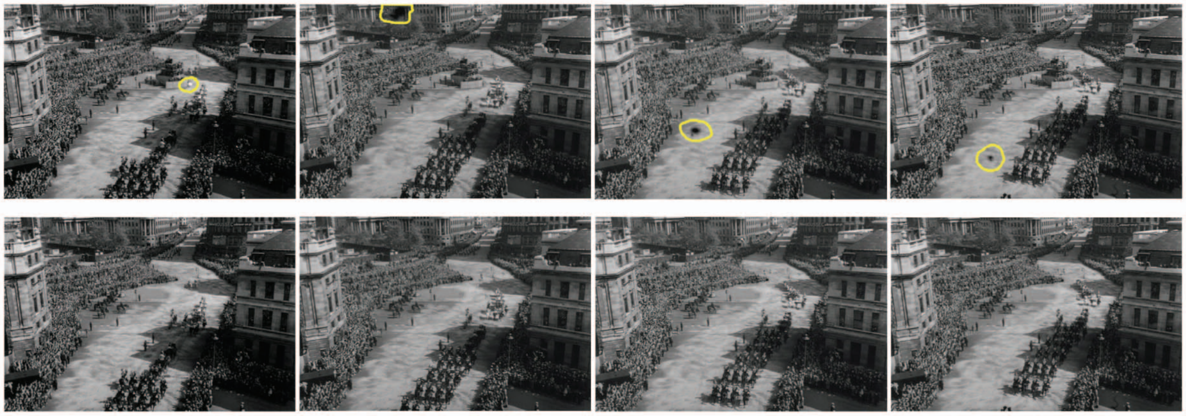

Fig. 10. *Parade*. Results for showing the potential application in film restoration. The image holes and the road texture behind the removed statue are seamlessly repaired.

Since the background illumination behind the motion is repaired, after the combination, the illumination on the repaired movel will be forced to be consistent with the repaired background illumination. For example, compare the moving snoopy toy (with respect to the environment) in the top and bottom rows of Fig. 9. Despite that, the overall completed video is a realistic and visually acceptable video restoration.

Fig. 10 shows one potential application in film restoration. In this example, we repair the highlighted damaged areas. We also remove the statue from the center of the frames.

Typically, given frame resolution 600×800 and that the number of frames in a damaged movel is 50 and the number of frames in a sample movel is 200, the total running time (excluding the easy human interaction in movel sampling) to generate the final restored video results is about 5 hours on a PIII 1GHz PC. The time complexity is linear with the number of pixels in the damaged movels.

5 Discussion and Concluding Remarks

Our video completion system is capable of synthesizing missing pixels for completing or repairing the static background and moving objects. Since a reference video mosaic is needed, our system works for a subclass of camera motions: rotation about a fixed point and panning without significant parallax. If the scene is approximately Lambertian, illumination, spatial, and temporal consistencies are maintained after background completion. Under the periodic motion assumption, our system samples the periodic motion of foreground objects as movels and aligns them with damaged movels to achieve motion completion. Temporal consistency of movel alignment is preserved by movel wrapping and movel regularization. In the presence of variable background illumination, our system separates the input video into a color component and an illumination component which are, respectively, repaired. The restored video preserves the scene structure as well as the variable illumination and maintains spatio-temporal consistency.

As shown in Fig. 8, since our movel does not capture self shadows or moving shadows, we cannot repair the shadow of a damaged movel. Another limitation is on the incorrect lighting on a repaired movel. Currently, we do not relight a repaired movel. In the future, we are interested in incorporating more knowledge into the movels so that better lighting and shadow on the repaired movels can be handled. We are also investigating ways to improve the running time of the system.

ACKNOWLEDGMENTS

The authors would like to thank the associate editor and all of the reviewers for their thorough and constructive suggestions, which were instrumental in improving the quality of the final paper. This research was supported by the Research Grant Council of the Hong Kong Special Administrative Region under grant number HKUST6200/05E and AoE/E-01/99 and the Chinese University of Hong Kong (CUHK) Direct Grant: 2050332.

REFERENCES

[1] S. Baker, R. Szeliski, and P. Anandan, "A Layered Approach to Stereo Reconstruction," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 434-441, 1998.

[2] M. Bertalmio, A.L. Bertozzi, and G. Sapiro, "Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. I355-362, 2001.

[3] M. Bertalmio, G. Sapiro, C. Ballester, and V. Caselles, "Image Inpainting," *Proc. SIGGRAPH,* pp. 417-424, 2000.

[4] M.J. Black and A.D. Jepson, "Estimating Optical-Flow in Segmented Images Using Variable-Order Parametric Models with Local Deformations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 10, pp. 972-986, Oct. 1996.

[5] Y. Caspi and M. Irani, "A Step towards Sequence-to-Sequence Alignment," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. II682-689, 2000.

[6] Y.-Y. Chuang, A. Agarwala, B. Curless, D.H. Salesin, and R. Szeliski, "Video Matting of Complex Scenes," *Proc. 29th Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 243-248, 2002.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Real-Time Tracking of Non-Rigid Objects Using Mean Shift," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. II142-149, 2000.

[8] R. Cutler and L.S. Davis, "Robust Real-Time Periodic Motion Detection, Analysis, and Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 781-796, Aug. 2000.

[9] J.E. Davis, "Mosaics of Scenes with Moving Objects," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 354-360, 1998.

[10] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto, "Dynamic Textures," *Proc. Int'l Conf. Computer Vision,* pp. II439-446, 2001.

[11] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-Based Image Completion," *Proc. SIGGRAPH,* pp. 303-312, 2003.

[12] A. Efros and T.K. Leung, "Texture Synthesis by Non-Parametric Sampling," *Proc. Int'l Conf. Computer Vision,* pp. 1033-1038, 1999.

[13] J. Jia and C.K. Tang, "Inference of Segmented Color and Texture Description by Tensor Voting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 6, pp. 771-786, June 2004.

[14] J. Jia and C.K. Tang, "Tensor Voting for Image Correction by Global and Local Intensity Alignment," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 1, pp. 36-50, Jan. 2005.

[15] J. Jia, T.P. Wu, Y.W. Tai, and C.K. Tang, "Video Repairing: Inference of Foreground and Background under Severe Occlusion," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. I364-371, 2004.

[16] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut Textures: Image and Video Synthesis Using Graph Cuts," *ACM Trans. Graphics, SIGGRAPH 2003,* vol. 22, no. 3, pp. 277-286, July 2003.

[17] Y. Li, J. Sun, C.K. Tang, and H.Y. Shum, "Lazy Snapping," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 303-308, 2004.

[18] G. Medioni, M.S. Lee, and C.K. Tang, *A Computational Framework for Feature Extraction and Segmentation.* Amsderstam: Elsevier Science, 2000.

[19] A. Schodl, R. Szeliski, D. Salesin, and I. Essa, "Video Textures," *Proc. SIGGRAPH,* pp. 489-498, 2000.

[20] S.M. Seitz and C.R. Dyer, "View Morphing," *Proc. 23rd Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 21-30, 1996.

[21] S.M. Seitz and C.R. Dyer, "View Invariant Analysis of Cyclic Motion," *Int'l J. Computer Vision,* vol. 25, no. 3, pp. 231-251, Dec. 1997.

[22] R. Szeliski, "Video Mosaics for Virtual Environments," *IEEE Computer Graphics and Applications,* pp. 22-30, Mar. 1996.

[23] J. Wang, Y. Xu, H.Y. Shum, and M.F. Cohen, "Video Tooning," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 574-583, 2004.

[24] J.Y.A. Wang and E.H. Adelson, "Layered Representation for Motion Analysis," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 361-366, 1993.

[25] Y. Weiss, "Deriving Intrinsic Images from Image Sequences," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. 68-75, 2001.

[26] Y. Wexler, E. Shechtman, and M. Irani, "Space-Time Video Completion," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. I120-127, 2004.