Space-Time Video Completion *

Y. Wexler E. Shechtman M. Irani
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science
Rehovot, 76100 Israel

Abstract

We present a method for space-time completion of large space-time "holes" in video sequences of complex dynamic scenes. The missing portions are filled-in by sampling spatio-temporal patches from the available parts of the video, while enforcing global spatio-temporal consistency between *all* patches in and around the hole. This is obtained by posing the task of video completion and synthesis as a global optimization problem with a well-defined objective function. The consistent completion of static scene parts simultaneously with dynamic behaviors leads to realistic looking video sequences.

Space-time video completion is useful for a variety of tasks, including, but not limited to: (i) Sophisticated video removal (of undesired static or dynamic objects) by completing the appropriate static or dynamic background information, (ii) Correction of missing/corrupted video frames in old movies, and (iii) Synthesis of new video frames to add a visual story, modify it, or generate a new one. Some examples of these are shown in the paper.

1. Introduction

We present a method for *space-time completion* of large space-time "holes" in video sequences of complex dynamic scenes. We follow the spirit of [10] and use non-parametric sampling, while extending it to handle static and dynamic information simultaneously. The missing video portions are filled-in by sampling spatio-temporal patches from other video portions, while enforcing global spatio-temporal consistency between all patches in and around the hole. Global consistency is obtained by posing the problem of video completion/synthesis as a global optimization problem with a *well-defined objective function* and solving it appropriately. The objective function states that the resulting completion should satisfy the two following constraints: (i) Every *local* space-time patch of the video sequence should be similar to some local space-time patch in the remaining parts of the video sequence (the "input data-set"), while (ii) *globally* all these patches must be consistent with each other, both spatially and temporally.

Solving the above optimization problem is not a simple task, especially due to the large dimensionality of video data. However, we exploit the spatio-temporal relations and redundancies to speed-up and constrain the optimization process in order to obtain realistic looking video sequences with complex scene dynamics at reasonable computation times.

Figure 1 shows an example of the task at hand. Given the input video (Fig. 1.a), a space-time hole is specified in the sequence (Figs. 1.b and 1.c). The algorithm is requested to complete the hole using information from the remainder of the sequence. The resulting completion and the output sequence are shown in Figs. 1.d and 1.e, respectively.

The goal of this work is close to few well studied domains. *Texture Synthesis* (e.g. [10, 19]) extends and fills regular fronto-parallel image textures. It is close to *Image Completion* (e.g., [8, 6]) which aims at filling in large missing image portions. Although impressive results have been achieved recently in some very challenging cases (e.g., see [8]), the goal and the proposed algorithms have so far been defined only in a heuristic way. Global inconsistencies often result from independent local decisions taken at independent image positions. Due to that reason, these algorithms heuristically use large image patches in order increase the chance for correct output. The two drawbacks of this approach is that elaborate methods for combining the large patches are needed for hiding inconsistencies [9, 8, 14] and the dataset needs to be artificially enlarged by including various skewed and scaled replicas that might be needed for completion.

Image Inpainting (e.g., [4, 18, 15]) was defined in a principled way (as an edge continuation process), but is restricted to small (narrow) missing image portions in highly structured image data. These approaches have been restricted to completion of *spatial information* alone in images. Even when applied to video sequences (as in [3]), the completion was still performed spatially. The temporal

*This work was supported in part by the Israeli Science Foundation(Grant No. 267/02) and by the Moross Laboratory at the Weizmann Institute of Science. The work of Y. Wexler was supported by E. & L.Kaufmann Postdoctoral Fellowship.

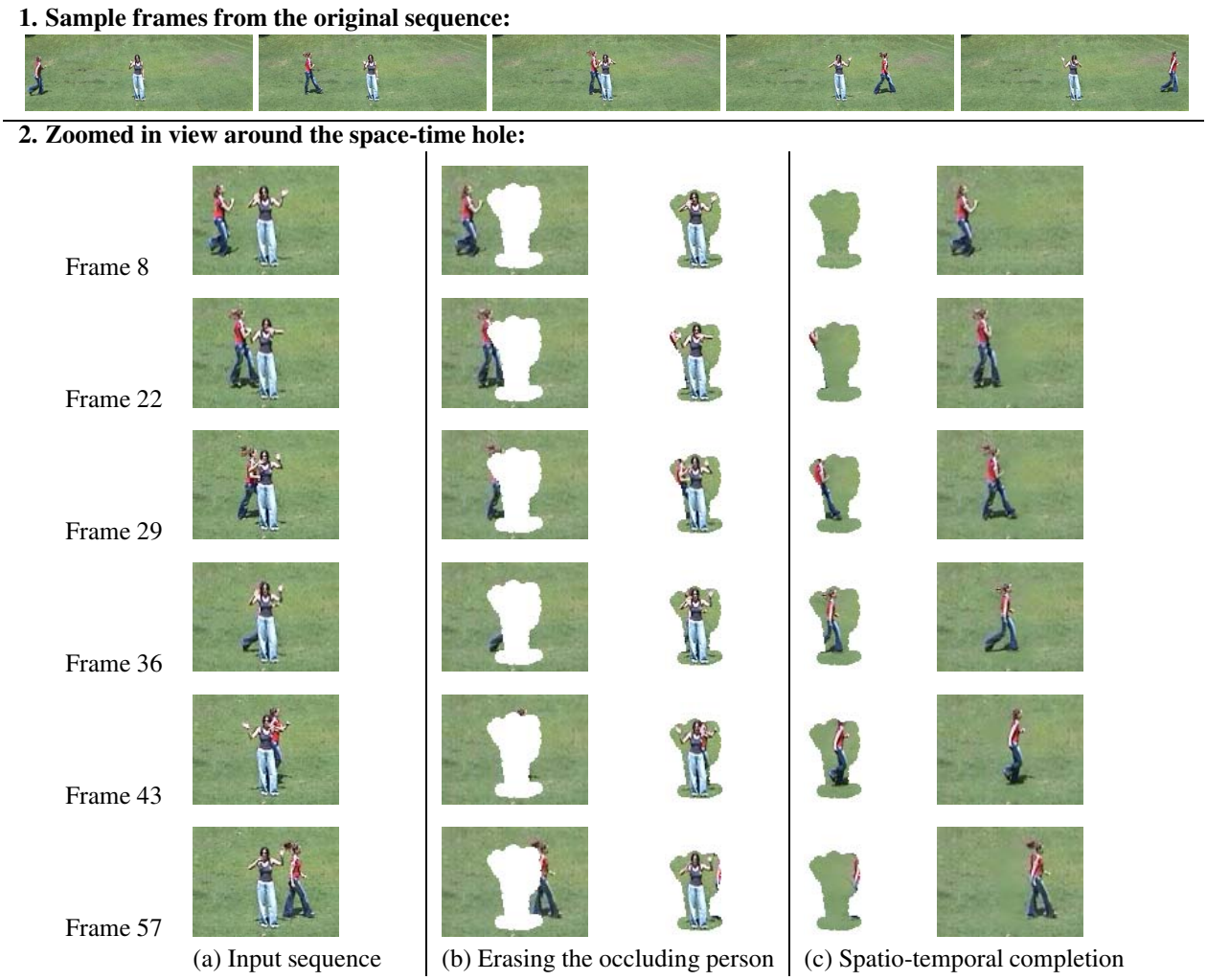
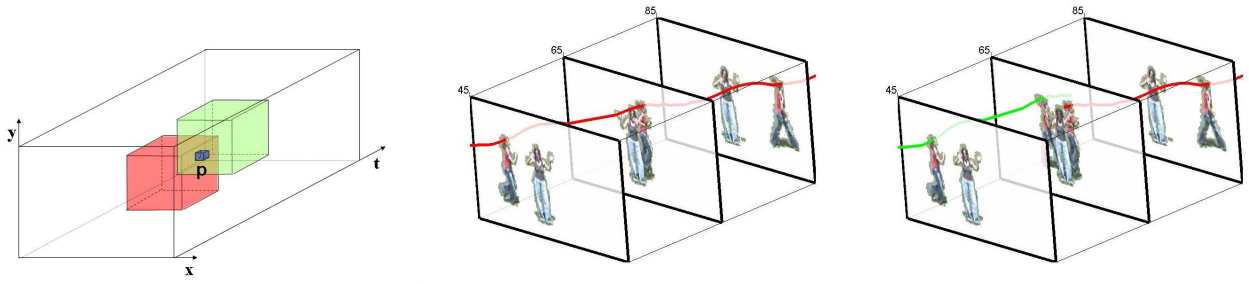


Figure 1: **(1)** Top part of the figure shows a few frames from a 240 frame sequence showing one person standing and waving her hands while the other person is hopping behind her. **(2)** A zoomed-in view on a portion of the video around the space-time hole before and after the completion. Note that the recovered arms of the hopping person are at slightly different orientations than the removed ones. As this particular instance does not exist anywhere else in the sequence, a similar one from a different time instance was chosen to provide an equally likely completion. See video in `www.wisdom.weizmann.ac.il/~vision/VideoCompletion.html`

Figure 2: Sources of information. Output frame 36 (shown in Figs. 1 and 10) is a combination of the patches marked here in red over the input sequence. It is noticeable that large continuous regions have been automatically picked whenever possible. Note that the hands were taken from a completely different frame than the legs.



(a) (b) (c)

Figure 3: Local and global space-time consistency. (a) Enforcement of the global objective function of Eq. (1) requires coherence of all space-time patches containing the point p. Such coherence leads to a globally correct solution (b). True trajectory of the moving object is marked in red and is correctly recovered. When only local constraints are used and no global consistency enforced, the resulting completion leads to inconsistencies (c). Background figure is wrongly recovered twice with wrong motion trajectories.

component of video has mostly been ignored.

Moreover, the basic assumption of Image Inpainting, that edges should be interpolated in some smooth way, does not naturally extend to time. Temporal aliasing is typically much stronger than spatial aliasing in video sequences of dynamic scenes. Often a pixel may contain background information in one frame and foreground information in the next frame, resulting in very non-smooth temporal changes. These violate the underlying assumptions of Inpaining.

In [13] a method has been proposed for employing spatio-temporal information to correct scratches and noise in poor-quality video sequences. This approach relies on optical-flow estimation, and is limited to small missing regions that undergo small motions.

A closely related area of research which regards temporal information explicitly is that of *dynamic texture synthesis* in videos (e.g. [7, 2]). Dynamic textures are usually characterized by an unstructured stochastic process. They model and synthesize smoke, water, fire, etc. but cannot model nor synthesize *structured dynamic objects*, such as behaving people. While [17] has been able to synthesize/complete video frames of structured dynamic scenes, it assumes that the "missing" frames already appear in their *entirety* elsewhere in the input video, and therefore needed only to identify the correct permutation of frames. An extension of that paper [16] manually composed smaller "video sprites" to a new sequence.

The approach presented here can automatically handle the completion and synthesis of both structured dynamic objects as well as unstructured dynamic objects under a single framework. It can complete frames (or portions of them) that never existed in the dataset. Such frames are constructed from various space-time patches, which are automatically selected from different parts of the video sequence, all put together consistently. The use of global objective function removes the pitfalls of local inconsistencies and the heuristics of using large patches. As can be seen in the figures and in the attached video, the method is capable of completing large space-time areas of missing information containing complex structured dynamic scenes, just as it can work on complex images. Moreover, this method provides a unified framework for various types of image and video completion and synthesis tasks, with the appropriate choice of the spatial and temporal extents of the space-time "hole" and of the space-time patches.

2. Completion as a global optimization

To allow for a uniform treatment of dynamic and static information, we treat video sequences as space-time volumes. A pixel (x, y) in a frame t will be regarded as a space-time point $p = (x, y, t)$ in the volume. We say that a video sequence $\mathcal{S}$ has *global visual coherence* with some other sequence $\mathcal{T}$ if every local space-time patch in $\mathcal{S}$ can be found somewhere within the sequence $\mathcal{T}$.

Let $\mathcal{S}$ be an input sequence. Let the "hole" $H \subseteq \mathcal{S}$ be all the missing space-time points within $\mathcal{S}$. For example, H can be an undesired object to be erased, a scratch or noise in old corrupt footage, or entire missing frames, etc.

We wish to complete the missing space-time region H with some new data H^* such that the resulting video sequence $\mathcal{S}^*$ will have as much global visual coherence with some reference sequence $\mathcal{T}$ (the dataset). Typically, $\mathcal{T} = \mathcal{S} \setminus H$, namely - the remaining video portions outside the hole are used to fill in the hole. Therefore, we seek a sequence $\mathcal{S}^*$ which maximizes the following objective function:

$$\text{Coherence}(\mathcal{S}^*|\mathcal{T}) = \sum_{p \in \mathcal{S}^*} \max_{q \in \mathcal{T}} s\left(W_p, W_q\right) \qquad (1)$$

where p, q run over all space-time points in their respective sequences, and W_p, W_q denote small space-time patches around p, q, respectively. The patches need not necessarily be isotropic and can have different size in the spatial and

Figure 4: Video removal. See video in `www.wisdom.weizmann.ac.il/~vision/VideoCompletion.html`

temporal dimensions. We typically use $5 \times 5 \times 5$ patches. $d(\cdot,\cdot)$ is a local distance measure between two space-time patches defined below (section 2.1).

Figure 3 explains why Eq.(1) induces global coherence. Each space-time point p belongs to other space-time patches of other space-time points in its vicinity. For example, for a 5×5×5 window, 125 different patches involve p. The red and green boxes in Fig. 3.a are examples of two such patches. Eq. (1) requires that all 125 patches agree on the value of p. Eq. (1) therefore leads to globally coherent completions such as the one in Fig.3.b.

If, on the other hand, the global coherence of Eq. (1) is not enforced, and the value of p is determined locally by a single best matching patch (i.e. using a sequential greedy algorithm as in [10, 6]), then global inconsistencies will occur in a later stage of the recovery. An example of temporal incoherence is shown in Fig. 3.c.

2.1. The local space-time similarity measure

At the heart of the algorithm is a well-suited similarity measure between space-time patches. A good measure needs to agree *perceptually* with a human observer. The Sum of Squared Differences (SSD) of color information, that is so widely used for image completion, does not suffice to provide the desired results in video (regardless of the choice of color space). The main reason for this is that the human eye is very sensitive to motion. Maintaining motion continuity is more important than finding the exact spatial pattern match within an image of the video. Figure 5 illustrates (in

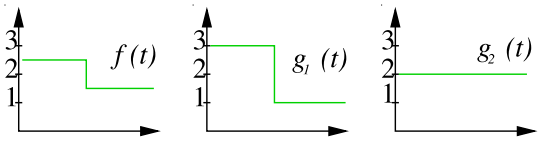


Figure 5: Importance of matching derivatives (see text).

1D) that very different temporal behaviors can lead to the same SSD score. The function $f(t)$ has a noticeable temporal change. Yet, its SSD score relative to a similar-looking function $g_1(t)$ is the same as the SSD score of $f(t)$ with a flat function $g_2(t)$: $\int (f-g_1)^2 dt = \int (f-g_2)^2 dt$. However, perceptually, $f(t)$ and $g_1(t)$ are more similar as they both encode a temporal change.

We would like to incorporate this into our algorithm and therefore use a measure that is similar to that of normal-flow to obtain a quick and rough approximation of the motion information as follows: Let Y be the sequence containing the grayscale (intensity) information obtained from the color sequence. At each space-time point we compute the spatial and temporal derivatives (Y_x, Y_y, Y_t). If the motion were only horizontal, then $u = \frac{Y_t}{Y_x}$ would capture the instantaneous motion in the x direction. If the motion were only vertical, then $v = \frac{Y_t}{Y_y}$ would capture the instantaneous motion in the y direction. We add these two measures to the RGB measurements, to obtain a 5-dimensional representation for each space-time point: (R, G, B, u, v). We apply an SSD measure to this 5-D representation in order to capture space-time similarities for static and dynamic parts simultaneously. Namely, for two space-time windows W_p and W_q we have $d(W_p, W_q) = \sum_{(x,y,t)} ||W_p(x,y,t) - W_q(x,y,t)||^2$ where for each (x,y,t) within the patch W_p, $W_p(x,y,t)$ is its 5D measurement vector (R, G, B, u, v). The distance is translated to a similarity measure

$$s(W_p, W_q) = e^{\frac{-d(W_p, W_q)}{2*\sigma^2}} \qquad (2)$$

Where σ was chosen empirically to reflect image noise of $5/255$ graylevels.

3. The optimization

The inputs to the optimization are a sequence $\mathcal{S}$ and a "hole" $H \subset \mathcal{S}$, marking the missing space-time points to be corrected or filled-in. We associate two quantities with each $p \in \mathcal{S}$: its 5D measurement vector (R, G, B, u, v) and a confidence value. Known points $p \in \mathcal{S} \setminus H$ will have high confidence, whereas missing points $p \in H$ will have low

Figure 6: Video removal. See video in `www.wisdom.weizmann.ac.il/~vision/VideoCompletion.html`

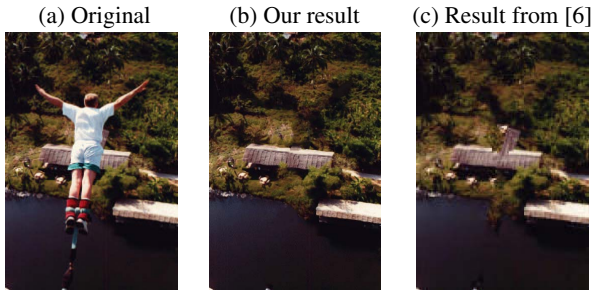


Figure 7: Image completion example.

Figure 8: Texture synthesis example. The original texture (a) was rotated 90^o and its central region (marked by white square) was filled using the original input image. (b) Our result. (c) Best results of (our implementation of) [10] were achieved with 9×9 windows (the only user defined parameter). Although the window size is large, the global structure is not maintained.

confidence. As Eq. (1) is a non-linear objective function, we solve for it iteratively, where in each step the current guess is updated locally.

Eq.(1) is satisfied *iff* the following two local conditions are satisfied:
(i) For every space-time point p all the space-time patches $W_{p_1} \dots W_{p_k}$ which contain it agree on the color value at p.
(ii) All those $W_{p_1} \dots W_{p_k}$ appear in the dataset $\mathcal{S} \setminus H$.
Therefore, the iterative step will aim at satisfying these two conditions at every point p. Let $p \in H$ be a missing point. Let $W_{p_1} \dots W_{p_k}$ be all these space-time patches containing p. Let $W_{q_1} \dots W_{q_k}$ denote the patches in $\mathcal{S} \setminus H$ that are most similar to $W_{p_1} \dots W_{p_k}$ according to Eq. (2). According to condition (ii) above, if W_{p_i} is reliable then $d_i = d(W_{p_i}, W_{q_i}) \approx 0$. Therefore $s_i = s(W_{p_i}, W_{q_i})$ measures the degree of reliability of the patch W_{p_i}. According to condition (i) above, the most likely color c at p should minimize the variance of the colors $c_1 \dots c_k$ proposed by $W_{q_1} \dots W_{q_k}$ at location p. However, as not all patches are equally reliable, the most likely color at p will minimize $\sum_i s_i (c - c_i)^2$. Therefore:

$$c = \frac{\sum_i s_i c_i}{\sum_i s_i} \tag{3}$$

This c is assigned to be the color of p at the end of the current iteration. The added "confidence" measure speeds up the convergence, as areas that have reached consensus will have more influence than areas that either did not converge yet or contain two conflicting solutions. This update rule minimizes the local error around each space-time point $p \in H$ while maintaining global consistency.

To further enforce global consistency and to speed up convergence, we perform the iterative process in multiple scales using spatio-temporal pyramids. Each pyramid level contains half the resolution in the spatial and in the temporal dimensions. The optimization starts at the coarsest pyramid level and the solution is propagated to finer levels for further refinement. This propagation is performed by finding the best matching patches in the current resolution level, and merging their higher resolution versions (from the dataset) at the finer resolution. This way the initial guess for the next level is not blurred unnecessarily, and high spatio-temporal frequencies are preserved.

One drawback of the update rule (3) is its sensitivity to outliers. As the mean is sensitive to outliers, it is enough that few neighbors will suggest wrong color to bias c and thus prevent or delay the convergence of the algorithm. In order to avoid such effects, we use density estimation of the feature set and choose the mean of the main cluster. This is done with the Mean-Shift algorithm [5] with a large window size equivalent to $50/255$ graylevels in RGB space.

A formalism similar to Eq. (1) was already used in [11] for successfully resolving ambiguities which are otherwise

inherent to the geometric problem of new-view synthesis from multiple camera views. The objective function of [11] was defined on 2D images. Their local distance between 2D patches was based on SSD of color information and included geometric constraints. The algorithm there did not take into account the dependencies between neighboring pixels as only the central point was updated in each step.

The above procedure bears some similarity to Belief Propagation (BP) approaches to completion [15]. However, BP is limited to no more than three neighboring connections, whereas this method takes into account all the neighbors (e.g. 125) at once. This allows us to deal with more complex situations while still converging quickly.

3.1. Handling large amounts of data

Even a small video sequence as in Fig. 1 contains $20,000,000$ space-time points. For comparison, a typical image used in [8] contains about $25,000$ pixels. An algorithm that is quadratic in the size of the dataset is impractical. Minimizing Eq.(1) requires searching the dataset for the most similar patch. A Naive search for a patch of diameter D, in a dataset with N space-time windows has complexity of $O(D^3 \cdot N)$. We avoid this cost in two ways. First, as described in Sec. 2.1 we do not intend to fill dynamic areas with static data. The motion components (u, v) provide an easy way for pruning the search. When searching for a patch W_p, we first compute its motion measure. The addition of the elements (u, v) makes p unique enough to compare on its own. The first step compares the $1 \times 1 \times 1$ vector against the dataset. We collect the locations of small portion of the best matching windows (typically 1000) and then compare them fully. This reduces the complexity of the brute-force search to $O(N)$. As this space has much lower dimensionality, nearest-neighbor algorithms such as [1] can provide a much better speedup with actual logarithmic time search thus reducing the computations in each iteration.

4. Space-time visual tradeoffs

The spatial and temporal dimensions are very different in nature, yet are inter-related. This introduces visual tradeoffs between space and time, that are beneficial to our space-time completion process. On one hand, these relations are exploited to narrow down the search space and to speed up the computation process. On the other hand, they often entail different treatments of the spatial and temporal dimensions in the completion process. Some of these issues have been mentioned in previous sections in different contexts, and are therefore only briefly mentioned here. Other issues are discussed here in more length.

Temporal vs. spatial aliasing: Typically, there is more temporal aliasing than spatial aliasing in video sequences of dynamic scenes. This is mainly due to the different nature of blur functions that precede the sampling process (digitization) in the spatial and in the temporal dimensions: The spatial blur induced by the video camera (a Gaussian whose extent is several pixels) is a much better low-pass filter than the temporal blur induced by the exposure time of the camera (a Rectangular blur function whose extent is less than a single frame-gap in time). This leads to a number of observations:

1. Extending the family of Inpainting methods to include the temporal dimension may be able to handle completion of (narrow) missing video portions that undergo slow motions, but it will unlikely be able to handle fast motions or even simple everyday human motions (such as walking, running, etc). This is because Inpainting relies on edge continuity, which will be hampered by strong temporal aliasing.

   Space-time completion, on the other hand, does not rely on smoothness of information within patches, and can therefore handle aliased data as well.

2. Because temporal aliasing is shorter than spatial aliasing, our multi-scale treatment is not identical in space and in time. In particular, after applying the video completion algorithm of Sec. 3, residual spatial (appearance) errors may still appear in fast recovered moving objects. To correct for those effects, an additional refinement step of space-time completion is added, but this time only the spatial scales vary (using a spatial pyramid), while the temporal scale is kept at the original temporal resolution. The completion process, however, is still space-time. This allows for completion using patches which have a large spatial extent, to correct the spatial information, while maintaining a minimal temporal extent so that temporal coherence is preserved without being affected too much by the temporal aliasing.

The local patch size: In our space-time completion process we typically use $5\times5\times5$ patches. Such a patch size provides $5^3 = 125$ measurements per patch. This usually provides sufficient statistical information to make reliable inferences based on this patch. To obtain a similar number of measurements for reliable inference in the case of 2D image completion, we would need to use patches of approximately 10×10. Such patches, however, are not small, and are therefore more sensitive to geometric distortions (effects of 3D parallax, change in scale and orientation) due to different viewing directions between the camera and the imaged objects. This restricts the applicability of image-based completion, or else requires the use of patches at different sizes and orientations [8], which increases the complexity of the search space combinatorially.

One may claim that due to the new added dimension (time) there is a need to select patches with a larger number of samples, to reflect the increase in data complexity. This,

Figure 9: Completion of missing frames. See video in `www.wisdom.weizmann.ac.il/~vision/VideoCompletion.html`

however, is not the case, due to the large degree of spatio-temporal redundancy in video data. The data complexity indeed increases a bit, but this increase is in no way is it proportional to the increase in the amounts of data.

The added temporal dimension therefore provides greater flexibility. The locality of the $5 \times 5 \times 5$ patches both in space and in time makes them relatively insensitive to small variations in scaling, orientation, or viewing direction, and therefore applicable to a richer set of scenes (richer both in the spatial sense and in the temporal sense).

Interplay between time and space: Often the lack of spatial information can be compensated by the existence of temporal information, and vice versa. To show the importance of combining the two cues of information, we compare the results of spatial completion alone to those of space-time completion. The top row of Fig. 10 displays the resulting completed frames of Fig. 1 using space-time completion. The bottom row of Fig. 10 shows the results obtained by filling-in the same missing regions, but this time using only image (spatial) completion. In order to provide the 2D image completion with the best possible conditions, the image completion process was allowed to choose the spatial image patches from *any* of the frames in the input sequence. It is clear from the comparison that image completion failed to recover the dynamic information. Moreover, it failed to complete the hopping woman in any reasonable way, regardless of the temporal coherence.

Furthermore, due to the large spatio-temporal redundancy in video data, the added temporal component provides additional flexibility in the completion process. When the missing space-time region (the "hole") is spatially large and temporally small, then the temporal information will provide most of the constraints in the completion process. In such cases image completion will not work, especially if the missing information is dynamic. Similarly, if the hole is temporally large, but spatially small, then spatial information will provide most of the constraints in the completion, whereas pure temporal completion/synthesis will fail. Our approach provides a unified treatment of all these cases, without the need to commit to a spatial or a temporal treatment in advance.

5. Unified approach to completion

The approach presented in this paper provides a unified framework for various types of image and video completion/synthesis tasks. With the appropriate choice of the spatial and temporal extents of the space-time "hole" H and of the space-time patches W_p, our method reduces to any of the following special cases:

1. When the space-time patches W_p of Eq. (1) have only a spatial extent (i.e., their temporal extent is set to 1), then our method becomes the classical *spatial* image completion and synthesis. However, because our completion process employs a global objective function (Eq. 1), global consistency is obtained that is otherwise lacking when not enforced. A comparison of our method to other image completion/synthesis methods is shown in Figs 7 and 8. (We could not check the performance of [8] on these examples. We have, however, applied our method to the examples shown in [8], and obtained comparably good results.)
2. When the spatial extent of the space-time patches W_p of Eq. (1) is set to be the entire image, then our method reduces to *temporal* completion of missing frames or synthesis of new frames using existing frames to fill in temporal gaps (similar to the problem posed by [17]).
3. If, on the other hand, the spatial extent of the space-time "hole" H is set to be the entire frame (but the patches W_p of Eq. (1) remain small), then our method still reduces to *temporal* completion of missing video frames (or synthesis of new frames), but this time, unlike [17], the completed frames may have never appeared in their entirety anywhere in the input sequence. Such an example is shown in Fig. 9, where three frames were dropped from the video sequence of a man walking on the beach. The completed frames were synthesized from bits of information gathered from different portions of the remaining video sequence. Waves, body, legs, arms, etc., were automatically selected from different space-time locations in the sequence so that they all match coherently (both in space and in time) to each other as well as to the surrounding frames.

6. Applications

Space-time video completion is useful for a variety of tasks in video post production and video restoration. A few ex-

Figure 10: Image Completion versus Video Completion

Figure 11: Restoration of a corrupted old Charlie Chaplin movie.

ample applications are listed below.
1. Sophisticated video removal: Video sequences often contain undesired objects (static or dynamic), which were either not noticed or else were unavoidable in the time of recording. When a moving object reveals all portions of the background at different time instances, then it can be easily removed from the video data, while the background information can be correctly recovered using simple techniques (e.g., [12]). Our approach can handle the more complicated case, when portions of the background scene are never revealed, and these occluded portions may further change dynamically. Such examples are shown in Figs. 1, 4, and 6).
2. Restoration of old movies: Old video footage is often very noisy and visually corrupted. Entire frames or portions of frames may be missing, and severe noise may appear in other video portions. These kinds of problems can be handled by our method. Such an example is shown in Fig 11.
3. Modify a visual story: Our method can be used to make people in a movie change their behavior. For example, if an actor has absent-mindedly picked his nose during a film recording, then the video parts containing the obscene behavior can be removed, to be coherently replaced by information from data containing a range of "acceptable" behaviors.

References

[1] S. Arya and D. M. Mount. Approximate nearest neighbor queries in fixed dimensions. In *ACM SODA*, 1993.

[2] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, , and M. Werman. Texture mixing and texture movie synthesis using statistical learning. In *IEEE, Visualization and Computer Graphics*, 2001.

[3] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *CVPR*, 2001.

[4] M. Bertalmio, G. Sapiroand V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH*, 2000.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE PAMI*, 2002.

[6] A. Criminisi and P. Perez K. Toyama. Object removal by exemplar-based inpainting. In *CVPR*, 2003.

[7] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *IJCV*, 2003.

[8] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. In *ACM TOG*. SIGGRAPH, 2003.

[9] A. Efros and W.T. Freeman. Image quilting for texture synthesis and transfer. *SIGGRAPH*, 2001.

[10] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999.

[11] A.W. Fitzgibbon, Y. Wexler, and A. Zisserman. Image-based rendering using image-based priors. In *ICCV*, 2003.

[12] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *VCIR*, 1993.

[13] A. Kokaram. Practical mcmc for missing data treatment in degraded video. In *ECCV*, 2002.

[14] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. In *ACM TOG*. SIGGRAPH, 2003.

[15] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. In *ICCV*, 2003.

[16] A. Schödl and I. Essa. Controlled animation of video sprites. In *SIGGRAPH*, 2002.

[17] A. Schödl, R. Szeliski, D.H. Salesin, and I. Essa. Video textures. In *Siggraph 2000*. ACM SIGGRAPH, 2000.

[18] T.Chan, S.H.Kang, and J.Shen. Euler's elastica and curvature based inpainting. *SIAM J. of Apllied Math*, 2001.

[19] L. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, 2000.