Evolving Time Fronts: Spatio-Temporal Video Warping

Alex Rav-Acha Yael Pritch Dani Lischinski Shmuel Peleg
The Hebrew University of Jerusalem

January 26, 2005

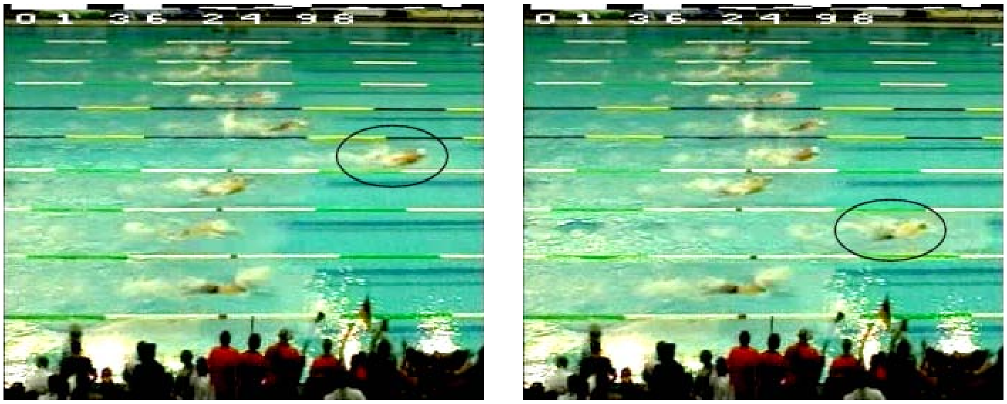


Figure 1: Who is the winner of this swimming competition? Spatio-temporal warping enables time to flow differently at different locations in the video, creating new videos with any desired winner.

Abstract

We present *evolving time fronts*, a new framework for spatio-temporal warping of video. The proposed framework is simple yet general, allowing a large variety of spatio-temporal warps to be specified in an intuitive manner. Specifically, we manipulate the time flow of a video sequence by sweeping an evolving *time front* surface through the video's aligned space-time volume. In this paper we first introduce the general framework, and then describe and discuss several specific strategies for time front evolution that we have experimented with so far. These strategies are demonstrated to produce a variety of interesting and useful operations on video, ranging from subtle timing changes to eye-catching special effects, creation of dynamic panoramic mosaics, and parallax effects.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—display algorithms; I.3.6 [Computer Graphics]: Methodology and Techniques—interaction techniques; I.4.9 [Image Processing and Computer Vision]: Applications

Keywords: video processing; video editing; video-based rendering; warping

1 Introduction

While spatial image warping is extensively used in image and video editing applications for creating a wide variety of interesting special effects, there are only very primitive tools for manipulating the temporal flow in a video. For example, we can compare temporal speeding up (slowing down) of the video to image zoom, or the "in-out" video selection to image crop and shift. But there are no tools that implement the spatio-temporal analogues of more general image warps, such as the various image distortion effects found in common image editing applications.

In this paper we present *evolving time fronts*, a new framework for spatio-temporal warping of video, which provides much more general *time flow* manipulation capabilities. The proposed framework is simple yet general, allowing a large variety of warps to be specified in an intuitive manner, resulting in many interesting and useful operations on video, ranging from subtle timing changes to eye-catching special effects. For example, within our framework it is easy to slow down the time flow in a particular spatio-temporal region of the video while speeding it up in another region. As demonstrated in Figure 1, this makes it possible to modify a competition video to produce a number of new videos, each having a different winner.

Other effects demonstrated in this paper include the spatio-temporal magnifying glass, which is clearly useful for instant replays in sports broadcasts; creation of patterns on dynamic texture videos, which offer an impressive way to animate logos or titles; dynamic panoramic video mosaics; and motion parallax effects. Finally, we suggest that our framework also offers a general mechanism for video splicing.

Our framework consists of three conceptual stages:

1. Given an input video sequence, an aligned space-time volume is constructed using computer vision techniques for video motion analysis.
2. A sequence of *time slices* is generated by sweeping an evolving *time front* surface through the space-time volume.
3. A region on each time slice is transformed to yield a frame of the warped output video sequence.

In this paper we focus primarily on the second stage of this process. In particular, we describe and analyze the spatio-temporal meaning of several different strategies for slicing the space-time volume, corresponding to the different effects mentioned earlier.

The concept of extracting various slices from a space-time volume is not new. This idea surfaced in several different contexts, which are briefly surveyed in the next section. However, these previous works have mostly looked at planar slices and/or focused on producing still images. Our contribution, therefore, lies in generalizing from planar and fixed time fronts to free-form and deforming ones; synthesizing entire videos, rather than still images; and exploring some of the video editing effects that may be achieved in this manner. While some of these effects are not new, we demonstrate that they all fit nicely within the powerful and flexible evolving time fronts paradigm.

Obviously, some of the effects described in this paper have certain limitations, which we shall point out as we discuss each effect. We believe that some of these limitations may be overcome, but addressing the technical challenges they present is outside the scope of this paper; our main goal here is to introduce the new evolving time fronts paradigm and to unveil (some of) its potential.

Following a review of related work, we introduce the concepts of time front and time flow in more detail (Section 3). Subsequent sections describe some of the different types of effects that we have been able to generate using our framework. We conclude the paper and offer directions for further research in Section 10.

2 Related Work

The space-time volume, where the 2D frames of a video sequence are stacked along the time axis, is not new to computer vision and graphics. It was introduced as the *epipolar volume* by Bolles *et al.* [1987; 1989], who analyzed slices perpendicular to the image plane (epipolar plane images) to track features in image sequences.

Light fields are also related to the space-time volume: they correspond to 4D subsets of the general 7D plenoptic function [Adelson and Bergen 1991], which describes the intensity of light rays at any location, direction, wavelength, and time. Light field rendering algorithms [Levoy and Hanrahan 1996; Gortler et al. 1996] operate on 4D subsets of the plenoptic function, extracting 2D slices corresponding to desired views. The space-time volume is a 3D subset of the plenoptic function, where two dimensions correspond to ray directions, while the third dimension defines the time or the camera position. In this work we use 2D slices through the space-time volume to define new video frames. Thus, similarly to light field rendering, different pixels in a frame may correspond to different time and/or camera position. Concentric mosaics [Shum and He 1999] are another example of a 3D plenoptic function subset, where the motion of the camera is constrained to coplanar concentric circles.

Multiple center of projection images [Rademacher and Bishop 1998] and multiperspective panoramas [Wood et al. 1997; Peleg et al. 2001] could also be considered as two-dimensional slices through a space-time volume spanned by a moving camera.

Klein *et al.* [2001; 2002] also utilize the space-time volume representation of a video sequence, and explore the use of arbitrary-shaped slices through this volume. This was done in the context of developing new non-photorealistic rendering tools for video, inspired by the Cubist and Futurist art movements. They define the concept of a *rendering solid*, which is a sub-volume carved out from the space-time volume, and generate a non-photorealistic video by compositing planar slices which advance through these solids.

Cohen *et al.* [2003] describe how a non-planar slice through a stack of images (which is essentially a space-time volume) could be used to combine different parts from images captured at different times to form a single still image. This idea was further explored by Agarwala *et al.* [2004]. Their "digital photomontage" system presents the user with a stack of images as a single, three-dimensional entity. However, the goal of their system is to produce a single composite still image, and they have not discussed the possibilities of generating dynamic movies from such 3D image stacks. For example, they discuss the creation of a stroboscopic visualization of a moving subject from a video sequence, but not the manipulation of the video segment to produce a novel video.

Video textures [Schödl et al. 2000] and graphcut textures [Kwatra et al. 2003] are also related to this work, as they describe techniques for video-based rendering. Schödl *et al.* generate new videos from existing ones by finding good transition points in the video sequence, while Kwatra *et al.* show how the quality of such transitions may be improved by using more general cuts through the space-time volume.

In contrast to these previous works, in this paper we are concerned with exploring the various meaningful ways in which the user may specify and control various spatio-temporal warps of dynamic video sequences, resulting in a variety of interesting and useful effects.

In an earlier work [Rav-Acha et al. 2005] we introduced the creation and playback of dynamic mosaics: video mosaics of dynamic scenes constructed by slicing through the space-time volume with a non-planar time front. As will become apparent in Sections 4 and 7, dynamic mosaics are instances of the more general video warping framework introduced in this paper. The new framework is applicable to videos captured by stationary, panning, and translating cameras, utilizes free-form evolving slices through the space-time volume, and incorporates simultaneous warping of both space and time.

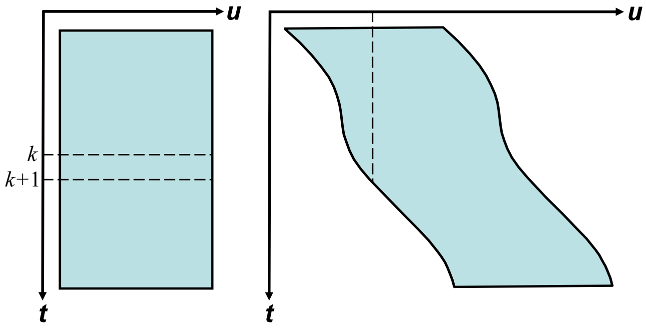
3 The Evolving Time Fronts Framework

In this section we describe the three conceptual stages of our spatio-temporal video warping framework: constructing a space-time volume, sweeping the volume with an evolving time front surface, and mapping the resulting time slices to produce the warped output video frames. Before proceeding with a more detailed description of the process, we introduce the notation for the different coordinates systems involved:

1. Original video coordinates (x,y,t) denote the (x,y) location in input video frame t, where (x,y) are given in the local coordinate system of each frame.
2. Registered space-time coordinates (u,v,t) denote locations in the space-time volume. Here (u,v) refer to some global coordinate system available after video registration.
3. Warped video coordinates (x',y',t') denote the (x',y') location in the output video frame t', again, in the local coordinate system of each frame.

3.1 The Space-Time Volume

Given a sequence of input video frames, they are first registered and aligned to a global spatial coordinate system (u,v). This defines a mapping $R(x,y,t) \to (u,v,t)$, typically leaving t unchanged, and only warping the spatial coordinates of each frame to their place on the global manifold. The necessary registration may be



(a) (b)

Figure 2: 2D space-time volumes: Each frame is represented by a 1D row, and the frames are aligned along the global u axis. A static camera defines a rectangular space-time region (a), while a moving camera defines a more general swept volume (b).

performed using previously described computer vision techniques [Bergen et al. 1992; Fitzgibbon 2001; Irani et al. 2002] and we will not discuss this aspect further in this paper.

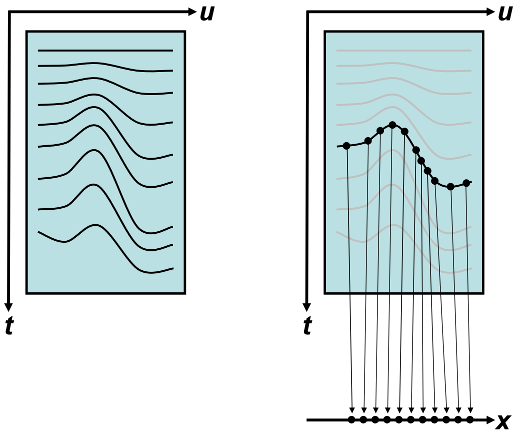
Stacking the aligned video frames along the time axis results in a 3D space-time volume. Figure 2 shows two examples of 2D space-time volumes. For a static camera the volume is shaped as a rectangular box, while a moving camera defines a more general swept volume. In either case, planar slices perpendicular to the t axis correspond to the original video frames. A static scene point traces a line parallel to the t axis (for a static or panning camera), while a moving point traces a more general trajectory.

3.2 The Time Front

One could conceive many different ways of transforming one space-time volume into another, yielding a novel video sequence. In the most general case, each pixel (x', y', t') in the novel video may be generated by an arbitrary function of the entire original space-time volume. In practice, however, such general transformations could be unintuitive and difficult to specify. Thus, we'd like to focus on a more restrictive class of transformations that correspond to *meaningful* spatio-temporal manipulations of the video. Furthermore, such transformations should be easy to specify, either programmatically or interactively by the user.

Spatial image warping geometrically transforms images, typically by applying a bijective mapping to transform the spatial coordinates of an input image to yield the warped image. Informally, this allows a user to change the position and size of various features in the image, but without breaking continuity. By the same token, we would like to allow a user to specify new spatio-temporal locations and sizes for various regions in the original space-time volume. For example, shrinking (stretching) a region along the temporal dimension would cause time to flow faster (slower) in the warped video. We would also like our mappings to be bijective in order to maintain a continuous spatio-temporal flow.

The approach that we explore in this paper, and one that has the desired characteristics outlined above, is to define the warping by specifying an *evolving time front* — a free-form surface that deforms as it sweeps through the space-time volume. Taking snapshots of this surface at different times results in a sequence of *time slices* (Figure 3a). These time slices are then spatially warped to yield the final output video frames (Figure 3b).



(a) (b)

Figure 3: Slicing the space-time volume: (a) snapshots of an evolving time front surface produce a sequence of time slices; (b) each time slice is mapped to produce a single output video frame.

Specifying the spatio-temporal warping in this manner separates between the manipulation of the temporal and the spatial components of the video and provides an intuitive interface for controlling such warps. For example, we can slow down or speed up the time flow in various regions at will by varying the speed at which the time front advances in the corresponding regions of the space-time volume.

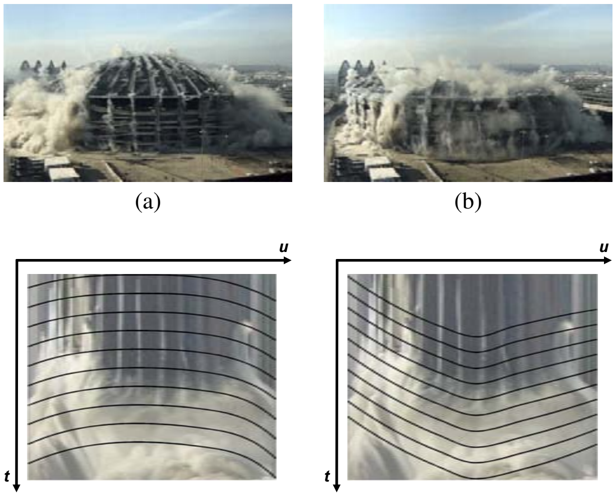
3.3 User Interface

Some of the effects described in this paper are generated with very specific and well-defined time front geometries. A video editing tool may present such effects to the user as a black box with a few input parameters that control the outcome. In other cases, a more elaborate user interface is required.

The temporal evolution of general time fronts and the speed at which they sweeps through the space-time volume may be specified via a keyframing user interface, similar to the interfaces used in computer animation. The user is required to specify a number of key time slices and indicate which output frames these slices correspond to. By interpolating between these key slices a continuously evolving time front is defined, which is then sampled at the appropriate time intervals to compute a time slice for each output frame.

We experimented with two different user interfaces for shaping the key time slices: (i) defining a free-form surface by manipulating a control mesh and (ii) a painting interface. In the latter interface the user starts with a gray image corresponding to a planar time slice perpendicular to the time axis and paints on it with a soft-edged brush. Darker colors are used to displace the time slice backwards in time, while brighter colors advance it forward. Both interfaces provide feedback to the user by displaying the image defined by the manipulated time slice.

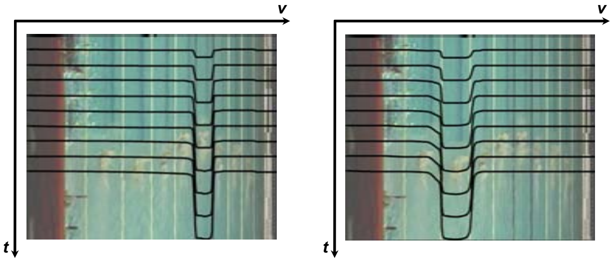
As for defining the spatial warp between the resulting time slices and output frames, we found that simple parallel projection of the slice onto a plane perpendicular to the t axis is sufficient for many useful video manipulations. However, if one wishes to define a more general spatio-temporal mapping (as in the spatio-temporal magnifying glass described in Section 5) it is possible to use any spatial image warping interface, such as [Beier and Neely 1992].



(a) (b)

(c) (d)

Figure 4: (a) and (b) are frames from two video clips, generated from the same original video sequence with different time flow patterns. (c) and (d) show several time slices superimposed over a *u-t* slice passing through the center of the space-time volume. The full video clips for all of the examples in this paper are available at `http://www.vision.huji.ac.il/videowarping/`.



(a) (b)

Figure 5: (a) and (b) show several time slices superimposed over a *v-t* slice passing through the center of the space-time volume of a swimming competition video. In each case the time front is offset forward over a different lane, resulting in two different "winners". (see Figure 1 and the corresponding video clips).

In the next sections we discuss several different strategies for time front evolution that we have experimented with, and demonstrate the corresponding video warping effects.

4 Spatially Varying Time Flow

Consider a space-time volume generated from a video of a dynamic scene captured by a static camera (as in Figure 2.a). The original video may be reconstructed from this volume by sweeping forward in time with a planar time front perpendicular to the time axis. As explained in Section 3.2, we can manipulate dynamic events in the video by varying the shape and speed of the time front as it sweeps through the space-time volume.

Figure 4 demonstrates two different manipulations of a video clip capturing the demolition of a stadium. In the original clip the entire stadium collapses almost uniformly. By sweeping the space-time volume as shown in Figure 4.c the output frames use points ahead in time towards the sides of the frame, causing the sides of the stadium to collapse before the center (Figure 4.a). Sweeping with the time fronts in Figure 4.d produces a clip where the collapse begins at the dome and spreads outward, as points in the center of the frame are taken ahead in time. It should be noted that Agarwala *et al.* [2004] used the very same input clip to produce still *time-lapse mosaic* images where time appears to flow in different directions (e.g., left-to-right or top-to-bottom). This was done using graph-cut optimization in conjunction with a suitable image objective function. In contrast, our framework generates entire new dynamic video clips.

Because of the unstructured nature of the expanding dust clouds in this example, we were able to obtain satisfactory results without graph-cuts optimization. In more structured cases, we believe that graph-cuts could be used to make time slices appear seamless by introducing local temporal displacements into each time slice.

Another example is shown in Figure 1. Here the input is a video clip of a swimming competition, taken by a stationary camera. By offsetting the time front at regions of the space-time volume corresponding to a particular lane one can speed up or slow down the corresponding swimmer, thus altering the outcome of the competition at will. Figure 5 shows the shape of the time slices used to produce this effect.

In this example we took advantage of the fact that the trajectories of the swimmers are parallel. In general, it is not necessary for the trajectories to be parallel, or even linear, but it is important that the tube-like swept volumes that correspond to the moving objects in space-time do not intersect. If they do, various anomalies, such as duplication of objects, may arise.

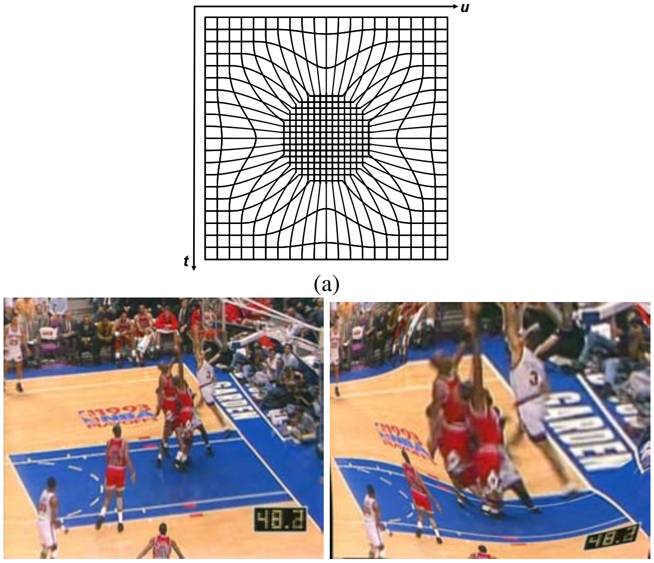
Another interesting application that we have yet to explore is dubbing a video with another soundtrack. The new soundtrack rarely matches the lip motion of the original video, and particularly disturbing are cases when the mouth moves but no sound is heard, or when sound is heard but the mouth does not move. This problem can be partially overcome by using the approach described in this section. The mouth motion can be accelerated or slowed down using an appropriate time flow, such that only the spoken moments correspond to mouth motions, while during silent moments the mouth does not move. If the head is moving, head tracking as known in the art can be performed, so that the different times will be taken from the same mouth area even though the head may be in different locations.

5 Spatio-Temporal Magnifying Glass

While our previous examples have demonstrated only time manipulations, in a general spatio-temporal mapping the spatial coordinates may be manipulated simultaneously with the temporal ones. In this case, all three output video coordinates (x', y', t') are functions of the space-time coordinates (u, v, t). That is,

$$\left(x', y', t'\right) = \left(f_x(u,v,t), f_y(u,v,t), f_t(u,v,t)\right).$$

This more general spatio-temporal warp provides a tool for creating additional interesting and useful effects. For example, we can apply a spatio-temporal magnifying glass to videos of sport events. Such a device enables us to magnify a spatial region in which some particularly interesting action takes place, while simultaneously slowing down the action. Unlike in ordinary instant replay, in our case the spatial and temporal magnification occur in the original context of the action, with a continuous transition between the magnified and the surrounding regions. Thus, when a basketball player dunks the ball into the basket, the viewer is able to see the dunk in greater



(a)

(b) (c)

Figure 6: A spatio-temporal magnifying glass. (a) A *u-t* slice of the space-time volume. Horizontal curves show the evolving time fronts, while the vertical curves specify the spatial warping of these time fronts. (b) A frame from the original sequence. (c) A frame in which the action under the basket is inside the focus volume of the magnifying glass.

detail, and at the same time keep track of the other players. This effect is demonstrated in several video clips.

The magnifying glass effect is achieved by deforming and warping the time fronts as illustrated in Figure 6a. This figure shows a slice of the space-time volume with horizontal curves describing the evolution of the time front. The vertical curves define the warping on the time slices to the frames of the output video. The dense grid in the center of the diagram is the *focal volume* of the lens. Action taking place inside this volume is both magnified and slowed down, while action outside this volume (but still inside the lens) is compressed and accelerated. Everything entirely outside the lens remains unaffected. In other words, time flow is accelerated when entering the lens, slows down in the central focal region, and accelerates again when exiting, to "catch up" with the time flow outside the lens. The spatial dimensions are affected in an analogous way (shrinking when entering/exiting the lens and expanding inside the focal volume). Specifically, we use a slightly modified version of the clamped focal radius lens with the Gaussian drop-off function, as proposed by Carpendale *et al.* [2004].

While the spatial and temporal mappings are inter-related, a different magnification factor may be applied in each domain. In this effect we did not perform registration of the input video frames; the space-time volume was formed by simply stacking the frames on top of each other. The user may control the effect by keyframing the center of the magnifying glass, specifying the magnification factors, and the drop-off function parameters. Instead of keyframing, automatic tracking may also be used to position the magnifying glass over a moving object.

Figure 7: The letters "SIGGRAPH" are formed over a dynamic texture of fire and smoke. See the accompanying video clip.

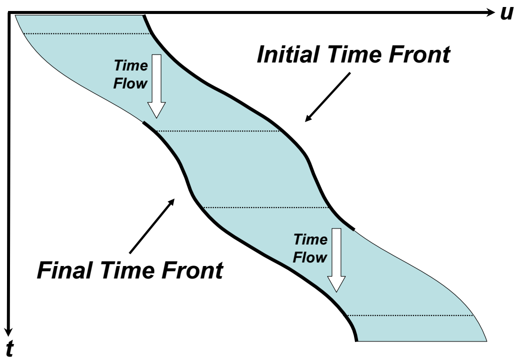


Figure 8: Time flow for generating dynamic mosaics from a panning camera.

Obviously, the amount of useful spatial and temporal magnification depends on the spatial and temporal resolution of the source video. The duration of the effect should also be limited to a short period of time, if temporal continuity is to be maintained: if we let a subject spend too much time inside the lens, the temporal disparity between the time flow inside and outside the lens becomes too great.

6 Patterns from Temporal Shifts

Mildly displacing the time front preserves the characteristics of the original video, but more abrupt displacements may introduce visible distortions in dynamic potions of the video. We can take advantage of such distortions to form various patterns over dynamic textures. For example, we have experimented with creation of text and logos over dynamic textures of fire and water. We start by rasterizing the desired shape to a binary image, and then displace the points in the interior of the shape forward or backward in time based on their distance from the shape's boundary. More specifically, for points closer to the boundary than some user specified value w, the displacement is linear in the distance, and constant for the remaining interior points. The resulting time front surface is then used to sweep through the space-time volume to produce the resulting video. A frame from one such video is shown in Figure 7. By starting and ending with the original planar time fronts we can make the pattern gradually emerge and disappear.

Note that the resulting effect is only visible in dynamic potions of the original video, and works best when there is sufficient fluctuation in brightness or in color. An interesting alternative which we have yet to explore is to animate the pattern used to define the displacement.

7 Dynamic Mosaics

Traditional mosaicing from a panning camera creates static panoramic images even when the scene is dynamic. In this section we show that using appropriate time flow patterns one can pro-

Figure 9: Dynamic panoramic mosaics. This figure is only a single frame in a panoramic movie, generated from a video taken by a panning camera (420 frames). When the panoramic movie is played (see video), the entire scene comes to life, with all water flowing down simultaneously.

duce dynamic panoramic movies from a panning camera scanning a dynamic scene [Rav-Acha et al. 2005]. The time-flow pattern is shown in Figure 8, and assumes a camera panning from left to right. In this time flow pattern, the initial time front is passing through the right side of each input frame, where regions are captured as they first enter the camera's field of view. Thus, the first time slice is a panoramic image capturing the earliest appearance of each point in the scene. The final time front is passing through the left side of each frame, where regions are captured just before leaving the field of view. This time slice shows the last appearance of each point in the scene. Similarly, each intermediate time slice (generated by linear interpolation) corresponds to an image where each region is captured at some time between entering and exiting the field of view. Although each panorama consists of regions taken from different points in time, the local dynamics inside each region is preserved. For example, in a waterfall scene, water in each region will be seen flowing down. Figure 9 shows a single panorama from such a movie, and the entire dynamic panoramic mosaic is available in the accompanying video clips. In this example, the effect is enabled by the fact that the motion inside each region (water flow) is roughly perpendicular to the panning direction. Additional examples and analysis are available in [Rav-Acha et al. 2005].

7.1 Advancing backwards in time

The original waterfalls video was captured by a video camera panning from left to right. If we wanted to reverse the scanning direction, we could simply reverse the order of frames in the video, but this would result in the water flowing upward. It turns out, however, that there is another time flow pattern that could be used to reverse the scanning direction, while preserving the original water flow. This time flow pattern is shown in Figure 10. Again, we refer the reader to the accompanying video clips.

8 Parallax Effects

In previous sections we have discussed the effects of time flow manipulation on a scene with moving objects. In this section we consider a different type of image motion: motion parallax. While general video sequences may have both motion parallax and moving objects, for the sake of clarity we discuss the parallax issue separately from moving objects. In this section we assume that the input video sequences are captured by a camera translating sideways.

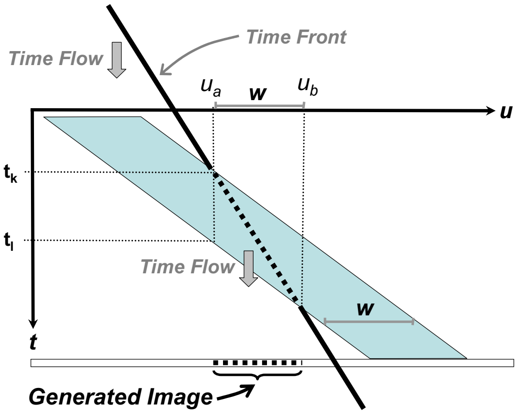


Figure 10: A time flow pattern that reverses the scanning direction of the camera. The width of the generated images remains unchanged. Time flow in the positive time direction (down) moves the generated images to the left, reversing the original panning direction. However, each local region exhibits the same temporal evolution as it did in the original sequence. Local point u_a, for example, will first appear as it was in time t_k, and will evolve in time until it disappears at time t_l.

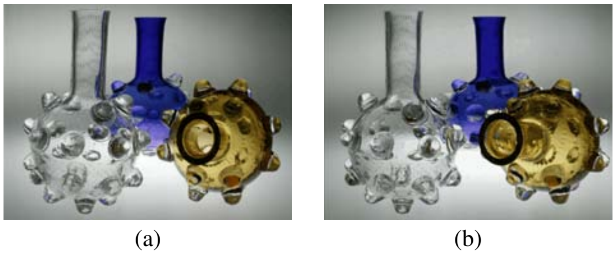
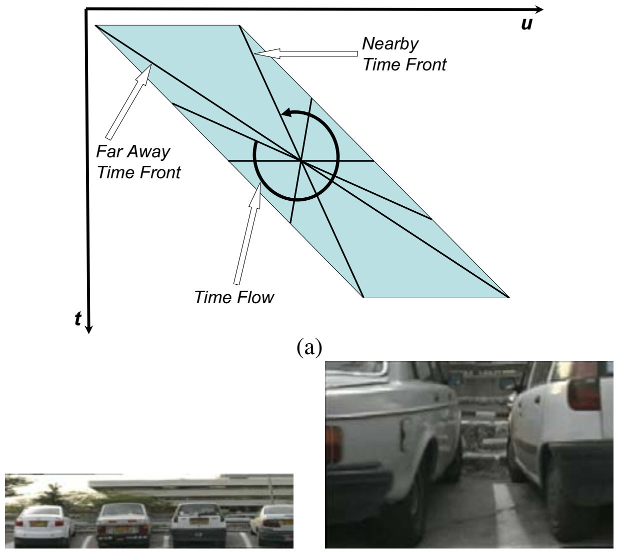

(a) (b)

Figure 11: Parallax stereo effects. The scene was scanned by a translating video camera. In (a) the time front includes the right side of all input frames, and in (b) the time front includes the left side of all frames. Time flow from (a) to (b) results in a video where all objects rotate as if on a turntable.



(a)

(b) (c)

Figure 12: Forward parallax effect created by rotating the time front. The scene was scanned by a translating video camera. (a) Shows the progression of time flow with a rotating time front. Image (b) was created by the “Far Away” time front, and the resulting image seems to be taken from a camera further away from the scene. Image (c) was created by the “Nearby” time front, and the resulting image seems to be taken by a camera closer to the scene than the one used to capture the original sequence. Note the two sides of the central cars visible in (c) but not in (b). The effect of moving into or out of the scene, generated by rotating the time front, is shown in the enclosed video.

It turns out that when a scene is scanned by a translating camera, the time flow pattern shown in Figure 8, which we used to generate dynamic panoramic mosaics, may also be used to produce a stereo parallax effect. For example, Figure 11 shows two “stereo” views generated from a space time volume captured by a translating camera. The time fronts corresponding to these two views are similar to those marked as “initial” and “final” in Fig 8, except that they are planar in order to avoid geometric distortions. The initial time front consists of the right sides of all input frames, and therefore each point in the resulting image appears to be viewed from a viewpoint to its left. Similarly, each point in the image corresponding to the final time front appears to be viewed from the right. Sweeping the time front from initial to final results in a sequence where the bottles appear to rotate on a turntable in front of a static camera. A detailed geometric interpretation of this case, as well as the effects of different camera trajectories, can be found in [Shum and He 1999; Peleg et al. 2001]

Another interesting case, shown in Fig. 12, is the case of the XSlits camera [Zomet et al. 2003; Yu and McMillan 2004]. In this case the time front does not sweep forward in time, as was the case with all of the examples discussed so far; instead, the time front rotates inside the space-time volume. As demonstrated in this figure, and explained in [Zomet et al. 2003; Yu and McMillan 2004], the rotation of the time front results in an apparent forward or backward motion of the generated views. We feel that this special kind of video warping, which can transform a sideways moving video into a forward moving video simply by defining an appropriate time front motion, is an important testament to the power and elegance of the evolving time fronts paradigm.

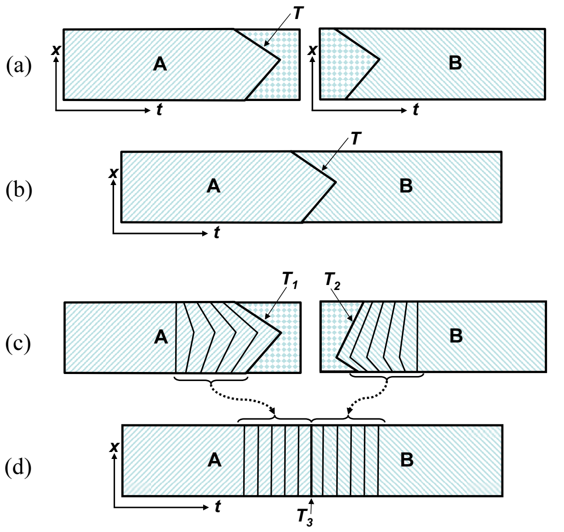


Figure 13: Splicing video clips. (a) represents two time-lines of video clips to be spliced together. An optimal space-time slice T is selected by [Kwatra et al. 2003] for a smooth spliced video in (b). The same space-time slice T is used for both clips A and B and the resulting spliced video. (c) represents our approach of evolving time fronts: a space-time slice T_1 is selected in Clip A, and a possibly different space time T_2 is selected in Clip B. In (d) the clips are spliced together by mapping gradually evolving time fronts in clips A and B to a common space-time slice T_3 in the spliced video.

9 Video Splicing

Kwatra *et al.* [2003] describe a method for splicing together video clips using graph cuts. Specifically, they search for an optimal spatio-temporal surface T that will make the seam between the two video clips as invisible as possible. This splicing scheme is illustrated in Figure 13.a-b and can be summarized as follows: Given two video clips $A(x,y,t)$ and $B(x,y,t)$ (A and B can be the same clip), and given a time shift d between them, a new video clip $C(x,y,t)$ is generated by splicing A and B together using the following rule:

$$C(x,y,t) = \begin{cases} A(x,y,t) & \text{if } t < T(x,y) \\ B(x,y,t-d) & \text{if } t > T(x,y) \end{cases},$$

where the space-time surface $T(x,y)$ corresponds to a graph cut that minimizes the cost of transition between A and B, in order to make the clip C seamless.

In many cases, however, seamless splicing is impossible, since no single spatio-temporal cut $T(x,y)$ achieves a sufficiently small transition cost. In such cases, we believe that evolving time fronts offer a more flexible solution by allowing the transition to occur between two different spatio-temporal surfaces, T_1 in A and T_2 in B. This is illustrated in Figure 13.c-d. A spliced video clip C may then be generated by warping both T_1 and T_2 to a common time front T_3 in the spliced clip C. We expect that using evolving time fronts for video splicing should be most significant when different regions of the scene have different temporal behavior (e.g., different periodicity). In such cases, the video can be better synchronized by slowing down or accelerating different parts of the scene.

We are presently implementing a video splicing tool that uses graph-cuts in conjunction with spatio-temporal warping as outlined above, and hope to present the results in the near future.

10 Conclusion

Given an input video sequence, new video sequences with a variety of interesting, and sometimes even surprising, effects may be generated by sweeping various evolving time fronts through its space-time volume. The space-time volume is "aligned" with respect to the camera motion, and this alignment is important for all cases involving a moving camera.

While the generation of new images by slicing through the space time volume is not new, this paper presents a new methodology to design the time flow for a specific desired effect. The time flow, which is the progression of time fronts through the space-time volume, can be manipulated to generate effects which include: (i) Shifting in time or changing the speed of selected spatial regions. (ii) Simultaneous spatial and temporal manipulations. (iii) Creating patterns in dynamic textures. (iv) Generation of dynamic panoramas. (v) Producing parallax in new directional views or even in forward motion.

While (i), (iv), and (v) were introduced before as unrelated cases, they are shown in this paper to be just special cases of the more general and powerful evolving time fronts framework.

This paper concentrated on the introduction of the evolving time fronts framework and some of the effects it can generate. Many important aspects were not covered, including: (i) Tracking of moving objects. This tracking is necessary to avoid the distortion of moving objects when they are reconstructed from their appearances at different times. In this case care should be taken to always select a moving object from a single frame, or a small number of adjacent frames. (ii) Interpolation. In the presence of image motion, more sophisticated interpolation should take into account this motion to prevent blurring and ghosting.

References

ADELSON, E. H., AND BERGEN, J. R. 1991. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, M. Landy and J. A. Movshon, Eds. MIT Press, Cambridge, Mass., 3–20.

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D. H., AND COHEN, M. F. 2004. Interactive digital photomontage. *ACM Transactions on Graphics 23*, 3 (Aug.), 294–302.

BAKER, H. H., AND BOLLES, R. C. 1989. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision 3*, 1 (May), 33–49.

BEIER, T., AND NEELY, S. 1992. Feature-based image metamorphosis. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, E. E. Catmull, Ed., vol. 26, 35–42.

BERGEN, J. R., ANANDAN, P., HANNA, K. J., AND HINGORANI, R. 1992. Hierarchical model-based motion estimation. In *European Conference on Computer Vision (ECCV '92)*, 237–252.

BOLLES, R. C., BAKER, H. H., AND MARIMONT, D. H. 1987. Epipolar-plane image anlysis: an approach to determining structure from motion. *International Journal of Computer Vision 1*, 1, 7–56.

CARPENDALE, M. S. T., LIGHT, J., AND PATTISON, E. 2004. Achieving higher magnification in context. In *Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology*, ACM, vol. 6, 71–80.

COHEN, M. F., COLBURN, A., AND DRUCKER, S. 2003. Image stacks. Tech. Rep. MSR-TR-2003-40, Microsoft Research.

FITZGIBBON, A. W. 2001. Stochastic rigidity: Image registration for nowhere-static scenes. In *Proc. International Conference on Computer Vision (ICCV 2001)*, vol. 1, 662–670.

GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. 1996. The Lumigraph. In *Proceedings of SIGGRAPH 96*, Addison-Wesley, H. Rushmeier, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 43–54.

IRANI, M., ANANDAN, P., AND COHEN, M. 2002. Direct recovery of planar-parallax from multiple frames. *IEEE Transactions on PAMI 24*, 11 (Nov.), 1528–1534.

KLEIN, A. W., SLOAN, P.-P. J., COLBURN, R. A., FINKELSTEIN, A., AND COHEN, M. F. 2001. Video cubism. Tech. Rep. MSR-TR-2001-45, Microsoft Research, Apr.

KLEIN, A. W., SLOAN, P.-P. J., FINKELSTEIN, A., AND COHEN, M. F. 2002. Stylized video cubes. In *Proc. Symp Comp Animation (SCA 2002)*.

KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics 22*, 3 (July), 277–286.

LEVOY, M., AND HANRAHAN, P. 1996. Light field rendering. In *Proceedings of SIGGRAPH 96*, Addison-Wesley, H. Rushmeier, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 31–42.

PELEG, S., BEN-EZRA, M., AND PRITCH, Y. 2001. Omnistereo: Panoramic stereo imaging. *IEEE Transactions on PAMI 23*, 3 (Mar.), 279–290.

RADEMACHER, P., AND BISHOP, G. 1998. Multiple-center-of-projection images. In *Proceedings of SIGGRAPH 98*, ACM Press / Addison-Wesley, M. Cohen, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 199–206.

RAV-ACHA, A., PRITCH, Y., LISCHINSKI, D., AND PELEG, S. 2005. Dynamosaics: Video mosaics with non-chronological time. In *Proceedings of CVPR 2005*. to appear.

SCHÖDL, A., SZELISKI, R., SALESIN, D. H., AND ESSA, I. 2000. Video textures. In *Proceedings of SIGGRAPH 2000*, ACM Press / Addison Wesley, K. Akeley, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 489–498.

SHUM, H.-Y., AND HE, L.-W. 1999. Rendering with concentric mosaics. In *Proceedings of SIGGRAPH 99*, ACM Press / Addison-Wesley, A. Rockwood, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 299–306.

WOOD, D. N., FINKELSTEIN, A., HUGHES, J. F., THAYER, C. E., AND SALESIN, D. H. 1997. Multiperspective panoramas for cel animation. In *Proceedings of SIGGRAPH 97*, Addison-Wesley, T. Whitted, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 243–250.

YU, J., AND MCMILLAN, L. 2004. General linear cameras. In *Proceedings of ECCV 2004*, 14–27.

ZOMET, A., FELDMAN, D., PELEG, S., AND WEINSHALL, D. 2003. Mosaicing new views: The crossed-slits projection. *IEEE Transactions on PAMI* (June), 741–754.