MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Video Mosaicing for Non-Chronological Time Editing*

Shmuel Peleg
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91903 Jerusalem, Israel

Abstract

New concepts in the manipulation of time in video editing are presented, enabling some unique visual effects. These effects include (i) slowing down (or postponing) some dynamic events while speeding up (or advancing) others. (ii) Creating dynamic panoramic images of dynamic scenes. (iii) Generating new views of a scene without a 3D model. (iv) Creating panoramic stereo images.

*Time manipulations are obtained by first constructing an aligned space-time volume from the input video, and then sweeping a 2D slice (*time front*) through that volume, generating a new sequence of images.*

1. Introduction

Imagine a person standing in the middle of a crowded square looking around. When requested to describe his dynamic surroundings, he will usually describe ongoing actions. For example: "some people are talking in the southern corner, others are eating in the north", etc. This kind of description ignores the chronological order in which each activity was observed, focusing on the activities themselves instead.

The same principle of manipulating the progression of time while relaxing the chronological constrains may be used to obtain a flexible scene representation. It allows us not only to postpone or advance some activities, but also to manipulate their speed. Dynamic panoramas are indeed the most natural extension of panoramic mosaicing. But dynamic mosaicing can be used also with a video taken from a static camera where we present a scheme to control the time progress for individual objects. We will start the description of temporal video manipulations in the case of a static camera, before we will continue to the case of dynamic panorama.

In our framework, the input video is represented as an aligned space-time volume. The time manipulations we explore are those that can be obtained by sweeping a 2D slice (*time front*) through the space-time volume, generating a new sequence of images.

Even when a camera is translating in a static scene, objects in the video have dynamics. This is the dynamics caused by motion parallax, where closer objects move faster than objects further away. Motion parallax is an indication to the 3D structure of the scene, and slicing the space time volume generated from such a video can create 3D effects. We describe virtual walkthrough that can be generated by rotating planar slices in the space-time volume, or generation of stereo pairs from parallel planar slices.

In order to analyze and manipulate videos of dynamic scenes, several challenging problems must be addressed: The first one is the stabilization of the input video sequence. The second problem is that time slices in the space-time volume may pass through moving objects. As a result, visual seams and other visual artifacts may occur in the resulting movie. These topics are not addresses in this paper, but are addressed in [20, 21].

1.1. Related work

The most popular approach for the mosaicing of dynamic scenes is to compress all the scene information into a single static mosaic image. There are numerous methods for dealing with scene dynamics in the static mosaic. Some approaches eliminate all dynamic information from the scene, as dynamic changes between images are undesired [26]. Other methods encapsulate the dynamics of the scene by overlaying several snapshots of the moving objects into the static mosaic, resulting in a "stroboscopic" effect [10, 7, 1]. In contrast to these methods that generate a single still mosaic image, we use mosaicing to generate a dynamic video sequence having a desired time manipulation.

The mechanism of slicing through a stack of images (which is essentially the space-time volume) is similar to video-cubes [12], which produces composite still images, and to panoramic stereo [17, 28]. Unlike these methods, dynamosaics are generated by coupling the scene dynamics, the motion of the camera, and the shape and the motion of the time front.

In [13, 6], two videos of dynamic textures (or the same video with two different temporal shifts) are being stitched seamlessly side by side, yielding a movie with a larger field of view. An approach towards seamless stitching in the case of dynamic textures (with the ability to produce infinite loops) was suggested in [2]. In this work we are interested in more general time manipulations, in which the edited movies combine information from many frames of the input sequence.

The basic idea of dynamosaicing was presented in several earlier papers [17, 28, 19, 20], and this paper attempts to present a global view on the family of these methods.

*This paper is a survey of earlier work done together with M. Ben-Ezra, D. Feldman,D. Lischinski, Y. Pritch, A. Rav-Acha, D. Weinshall, and A. Zomet. This research has been partially supported by grants from the Israel Science Foundation.

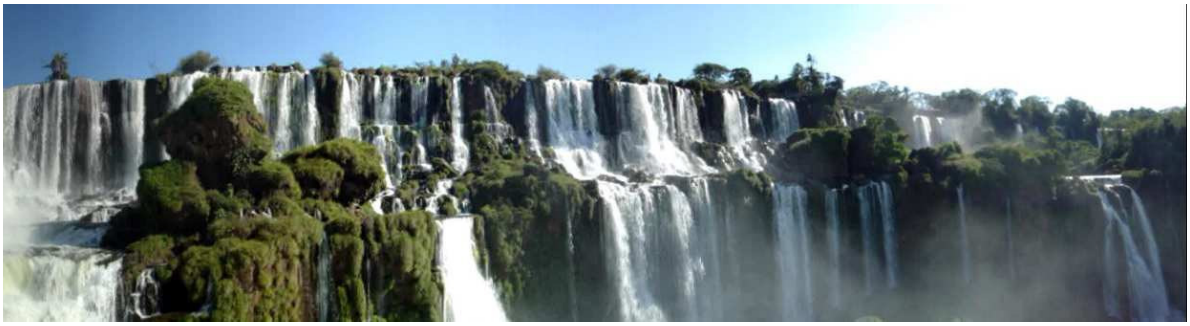


Figure 1. Dynamosaicing can create dynamic panoramic movies of a scene. This figure shows a single frame in a panoramic movie, generated from a video taken by a panning camera (420 frames). When the movie is played (see *www.vision.huji.ac.il/dynmos*), the entire scene comes to life, and all water flows down simultaneously.

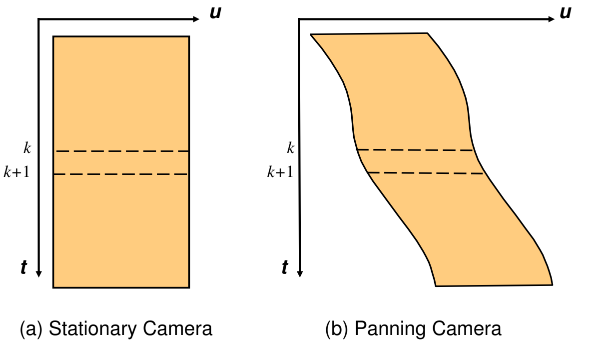


Figure 2. Aligned space-time volumes: Each frame is represented by a 1D row, and the frames are aligned along the global x axis. A static camera defines a rectangular space-time region (a), while a moving camera defines a more general swept volume (b).

1.2. The Aligned Space-Time Volume

Given a sequence of input video frames $I_1, \ldots, I_N$, they are first registered and aligned to a global spatial coordinate system. A specialized alignment scheme for sequences of dynamic scenes is described in [20], but other stabilization methods can sometimes be used (e.g., [4, 25]).

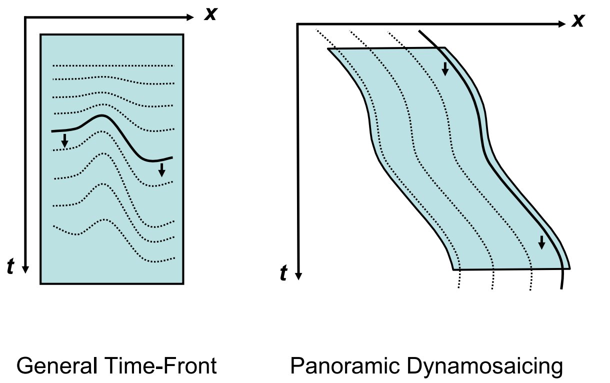
Stacking the aligned video frames along the time axis results in a 3D space-time volume $V(x, y, t)$. Fig. 2 shows two examples of 2D space-time volumes. For a static camera the volume is a rectangular box, while a moving camera defines a more general swept volume. In either case, planar slices perpendicular 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.

A point $V(x, y, t)$ in the aligned space-time volume therefore represents the point (x, y) in frame t **after** it has been aligned to the a global coordinate system, and not the (x, y) location in the original frame.

2. Dynamosaicing

2.1. Evolving Time Fronts

Image mosaicing is the process of creating new images by selecting patches from the frames of the input sequence and combining them to form a new image ([18, 9, 1] are just a few examples of the wide literature on mosaicing). It can be described by a function $M(x, y)$ that maps each



(a) (b)

Figure 3. Slicing the space-time volume: (a) Snapshots of an evolving time front surface produce a sequence of time slices; each time slice is mapped to produce a single output video frame. (b) The particular time flow for generating dynamic panoramas from a panning camera

pixel (x, y) in the output mosaic image S to the input frame from which this pixel is taken and its location in that frame. In this work we focus only on temporal warping, that is $S(x, y) = V(x, y, M(x, y))$, where $V(x, y, t)$ is the aligned space-time volume. This function can be represented by a slice (*time front*) in the space-time volume as illustrated in Fig. 3. A time front determines the mosaic patches by its intersection with the frames of the original sequence at the original discrete time values (shown as dashed lines in Fig. 3).

To get a desired time manipulation we specify 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 that are represented by temporal-shift functions $S_k(x, y) = V(x, y, M_k(x, y))$.

2.2. What Time Manipulations Can be Obtained?

In this section we describe the manipulation of chronological time vs. local time using dynamosaicing. We first describe the dynamic panoramas, where the chronological time is eliminated. We then show other applications where a video should be edited in a way that changes the chronological order of events in the scene. The realistic ap-

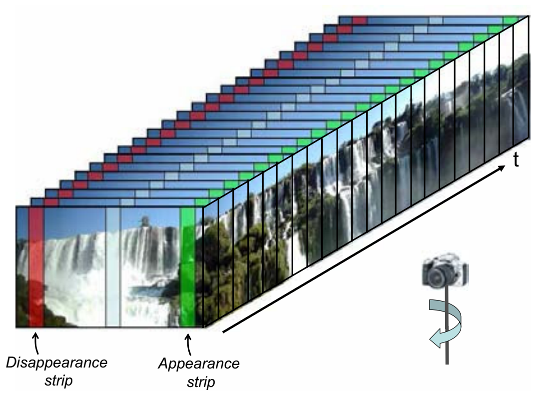
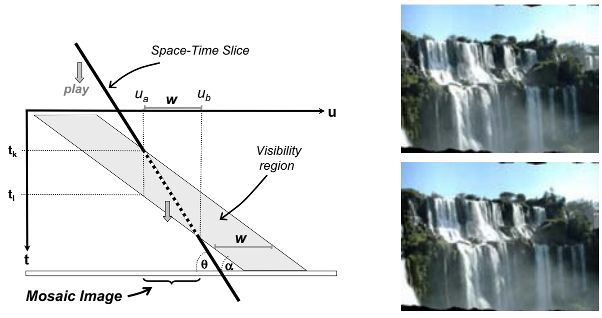


Figure 4. Input frames are stacked along the time axis to form a space-time volume. Given frames captured with a video camera panning clockwise, panoramic mosaics can be obtained by pasting together vertical strips taken from each image. Pasting together strips from the right side of the images will generate a panoramic image where all regions appear as they first enter the sequence, regardless of their chronological time.



(a) (b)

Figure 5. (a) A slicing scheme that reverses the scanning direction using a time front whose slope is twice the slope of the occupied space-time region ($\tan\theta = 2\tan\alpha$). The width of the generated mosaic image is w, the same as that of the original image. Sweeping this time front in the positive time direction (down) moves the mosaic image to the left, in the opposite direction to the original scan. However, each region appears in the same relative order as in the original sequence: u_a first appears in time t_k, and ends in time t_l.
(b) Two frames from an edited movie. The scanning direction of the camera was reversed, but the water continues to flow down. The entire video appears at *www.vision.huji.ac.il/dynmos*.

pearance of the movie is kept by preserving the time flow locally, even when the global chronological time is being changed.

2.2.1 Panoramic Dynamosaicing

Panoramic dynamosaics may be generated using the approach described above with the time slices shown in Fig. 3b. Assuming that the camera is scanning the scene from left to right, the first mosaic in the sequence will be constructed from strips taken from the right side of each input frame, showing regions as they first appear in the field of view (see Fig. 4). The last mosaic in the resulting sequence will be the mosaic image generated from the strips on the left, just before a region disappears from the field of view. Between these two marginal slices of the space-time volume we take intermediate slices, smoothly showing regions from their appearance to their disappearance. Each of the mosaic images is a panoramic image, and the resulting movie is a dynamic panorama in which local time is preserved. Fig. 1 shows a single panorama from such a movie.

Panoramic dynamosaics represent the elimination of the chronological time of the scanning camera. Instead, all regions appear simultaneously according to the local time of their visibility period: from their first appearance to their disappearance. But there is more to time manipulation than eliminating the chronological time.

Figure 1 shows an example of panoramic dynamosaic. To generate the panoramic movie simple slices, as those demonstrated in Fig. 3.b), were used. Since it is impossible to visualize the dynamics effects in these static images, this example and more video clips can be examined at *www.vision.huji.ac.il/dynmos*.

2.2.2 Advancing Backwards in Time

This effect is best demonstrated with the waterfalls sequence (Figure 1), which was scanned from left to right by a video camera. If we want to reverse the scanning direction, we can simply play the movie backwards. However, playing the movie backwards will result in the water flowing upwards.

At first glance, it seems impossible to play a movie backwards without reversing its dynamics. Yet, this can also be achieved by manipulating the chronological time, while preserving the local dynamics. Looking at panoramic dynamosaics, one can claim that all objects are moving simultaneously, and the scanning direction does not have any role. Thus, there must be some kind of symmetry, which enables to convert the panoramic movie into a scanning sequence in which the scanning is at any desired direction and speed.

Indeed, the simple slicing scheme shown in Fig. 5 reverses the scanning direction while keeping the dynamics of the objects in the scene. In the water falls example, the scanning direction is reversed, but the water continues to flow down!

2.2.3 Time Manipulations with Planar Time Fronts

The different types of time manipulations that can be obtained with planar time fronts are described in Fig. 6. The time fronts always sweep "downwards" in the direction of positive time at the original speed to preserve the original local time.

The different time fronts, as shown in Fig. 6, can vary both in their angles relative to the x axis and in their lengths. Different angles result in different scanning speeds of the scene. For example, maximum scanning speed is achieved with the panoramic slices. Indeed, in this case the resulting movie is very short, as all regions are played simultaneously. (The scanning speed should not be confused with the dynamics of each object, which preserve the original speed and direction).

The field of view of the resulting dynamosaic frames may be controlled by cropping each time slice as necessary. This can be useful, for example, when increasing the

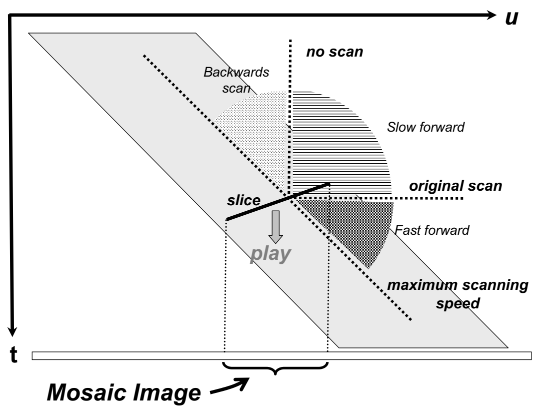


Figure 6. The effects of various planar time fronts. While the time front always sweeps in a constant speed in the positive time direction, various time front angles will have different effects on the resulting video.

scanning speed of the scene while preserving the original field of view.

2.3. Temporal Video Editing

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. 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.

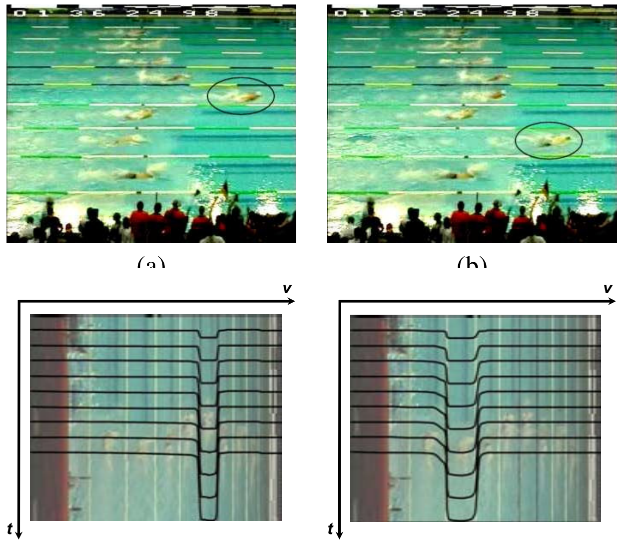
An example is shown in Figure 7. Here the input is a video clip of a swimming competition, taken by a stationary camera. By offsetting the time front at regions of the space-time volume corresponding to a particular lane one can speed up or slow down the corresponding swimmer, thus altering the outcome of the competition at will. The shape of the time slices used to produce this effect is shown as well.

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.

3. Mosaicing X-Slits Images

Slicing the space time volume of a translating camera can give X-Slits images. The X-Slits projection is a generalization of the perspective projection, and is defined by two slits. The projection ray of every 3D point is defined by the line that passes through the point and intersects both slits. The image of a point will be the intersection of the projection ray with the image surface. In this section we only address the case in which the slits are orthogonal to each other and are parallel to the image plane, as shown schematically in Fig. 8. The X-Slits model is presented in details in [28]

One of the important features of the X-Slits projection is that new X-Slits images can be easily generated by planar time fronts intersecting the space-time volume. The space-time volume is constructed from images captured by a translating pinhole camera, and the newly generated X-Slits images look compelling and realistic. Different planar time fronts can generate images which correspond to different X-Slits projections that correspond to different translations and rotations of the X-Slits camera. X-Slits images generated by slicing the space-time volume can be used for rendering movies of virtual walkthroughs, while providing a strong 3D sensation of parallax, reflections and occlusions.



(c) (d)

Figure 7. Who is the winner of this swimming competition? Temporal editing enables time to flow differently at different locations in the video, creating new videos with any desired winner, as shown in (a) and (b). (c) and (d) show several time slices superimposed over a y-t slice passing through the center of the space-time volume. In each case the time front is offset forward over a different lane, resulting in two different "winners". The full video clips are available at *www.vision.huji.ac.il/dynmos*.

Fig. 8 shows the basic model of the X-Slits projection. Specifically, the X-Slits "camera" has two slits which should be two different lines in $\mathcal{R}^3$, together with an image plane that does not contain any of the slits. For every 3D point not on either of the slits there is a single ray which connects the point with both slits simultaneously. The intersection of this ray with the image plane defines the projected image of the 3D point (which is a point, unless the ray lies on the image plane). The X-Slits model is thus a valid 3D to 2D projection, defining a many-to-one mapping from the 3D world to the 2D image plane.

The X-Slits projection equations can be derived simply. Assume that the vertical slit at $X = 0$, $Z = Z_2$ and the horizontal slit is at $Y = 0$, $Z = Z_1$. The image plane is the $X - Y$ plane at $Z = 0$. The X-Slits projection equation is

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -Z_2 \frac{X}{Z-Z_2} \\ -Z_1 \frac{Y}{Z-Z_1} \end{pmatrix} \quad (1)$$

In Section 3.3 we discuss how to generate new X-Slits images by slicing a space-time volume generated from images taken by a regular pinhole camera translating along a straight line.

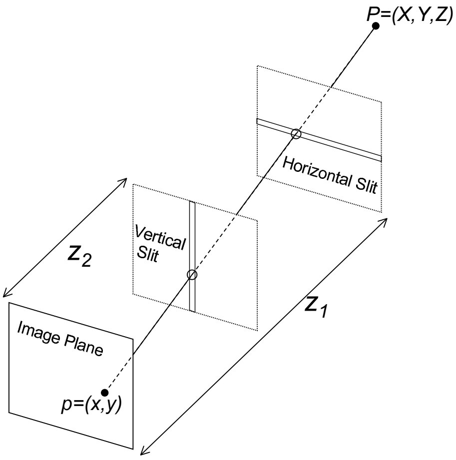


Figure 8. A design of a X-Slits camera where the slits are orthogonal to each other and parallel to the image plane. Z_1 denotes the horizontal focal length and Z_2 denotes the vertical focal length. The projection ray of a 3D point $\mathbf{p} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ is shown, with circles showing its intersection points with the 2 slits.

Applications are discussed in Section 3.4. The first application includes the generation of an arbitrary virtual walkthrough from a single sequence of images. The second application is 3D object visualization. Specifically, the X-Slits model allows us to virtually place one slit behind an object, keeping the other slit in front of the object. As a result nearby objects appear narrower than distant objects, which yields unusual pictures. When the vertical slit is at infinity we get a pushbroom projection. With different angles of the pushbroom projection, as can be obtained by different parallel slices of the space-time volume as described in Sec. 3.5, panoramic stereo pair images can be generated.

3.1. Relation to Previous Work

There has been much work on non-perspective projection models [16] and the generation of non-perspective images from video sequences [18, 22, 27, 11]. In most cases these images are used as a visual summary of the video, or for 3D visualization. The present work uses a similar mosaicing technique with one important difference: the mosaiced strips are sampled from varying positions in the input images. This makes the generation of virtual walkthroughs possible.

In many image-based rendering (IBR) techniques rays from a set of input images are collected and a new image is rendered by resampling the stored rays [15, 8, 14]. In order to create new perspective images of reasonable quality, the requirements become prohibitive: the number of stored rays becomes larger than available memory, and those rays are derived from a very large collection of carefully taken pictures. There are attempts to make IBR more efficient and more general [5, 3], or to use such approximations as moving the camera in a lower dimensional space [23, 24].

The present approach to X-slits is mostly related to [23, 24, 5] with several differences: First, rather than trying to approximate the perspective projection, we accurately define the projection geometry of the resulting images, and analyze the model limitations. Second, the rendering tool

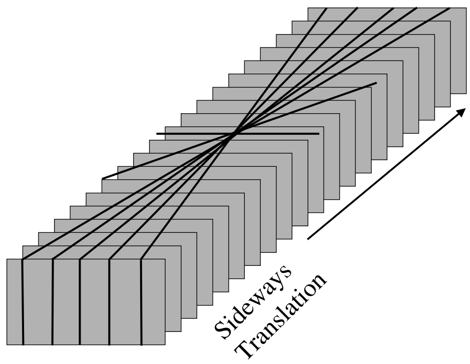


Figure 9. A schematic description of images generated as planar slices of the rectified space-time Volume. Changing the angle of the planar slice moves the vertical slit inside and outside the scene.

is very simple - slicing of the space-time volume obtained from a translating perspective camera. Consequently the most important feature of our technique is the fact that ray-sampling for the generation of new views does not require detailed accounting of the parameters of the generating images. As we show below, if the camera's motion is sideways every vertical planar slice of the space-time volume gives a valid X-Slits image.

3.2. The Rectified Space-Time Volume

In dynamosaicing, when time was of interest, we created the *Aligned Space-Time Volume* described in Sec. 1.2. Frame separation remained the original time between frames, and the translation of the camera has been compensated by shifting the the frames in space and aligning them to a global coordinate system.

In the X-Slits case the dynamics is motion parallax, which represents the scene geometry. In this case it is necessary that the t direction in the space-time volume be proportional to the camera translation. When the cameras velocity and frame rate are constant, the time of frame capture is proportional to the location of the camera along the camera path, and no manipulation of the space-time volume is needed. When the velocity of the camera varies, the time of frame capture is no longer proportional to the location of the camera. In this case we shift the frames in time to get the Rectified Space-Time Volume.

In the rectified space-time volume, the image location along the time axis is replaced with the cameras location along the x axis. Thus the temporal spacing between frames in the t axis is not constant any longer, and depends on the magnitude of the camera translation. The x location of the camera can be computed in various methods that compute camera translation, and we used the method presented in [21].

3.3. Slicing the Rectified Space-Time Volume

Our goal is to synthesize new X-Slits views from "regular" perspective images. The input sequence is assumed to be captured by a pinhole camera translating dsideways along a straight line in 3D space, and without changing its orientation or internal calibration. As we will show below, we can generate a new X-Slits image where the directions

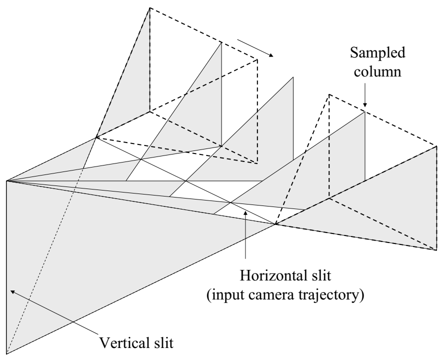


Figure 10. The non-stationary column sampling routine which is used to synthesize new images.

of the two slits of the underlying virtual X-Slits camera are as follows:

1. One slit lies on the path of the optical center of the moving pinhole camera.
2. The second slit is parallel to image's vertical axis, and its location is determined by the parameters of the new view synthesis process.

The X-Slits rendering is done by slicing the rectified space-time volume with a vertical planar time front which cuts image t at location $s(t)$.

The parameters of the function $s(t)$ determine the exact location of the second slit of the virtual camera. Using this observation, a virtual walkthrough is the result of generating a sequence of X-Slits images by moving the planer time front, which corresponds to moving the second slit. We develop bellow the relation between the location of the vertical slit and the planar time front.

Let our input image sequence be captured by a pinhole camera translating in constant speed along the X axis from left to right. In case the camera velocity varies, we create the rectified space-time volume as described in Sec. 3.2. We claim that each planar slice of the rectified space-time volume, as shown in Fig. 9, corresponds to an X-Slits image. Let's examine in Fig. 9 the diagonal slice cutting from the left side of the leftmost (first) image to the right side of the rightmost (last) image. This planar slice is generated by pasting columns from the input images, as illustrated in Fig. 10. We start by sampling the left column of the first (leftmost) image, and conclude by sampling the right column of the last (rightmost) image. In between, intermediate columns are sampled from successive images using a linear sampling function.

A schematic illustration of this setup is given in Fig. 11a, in a top-down view. A sequence of positions of the real pinhole camera is shown, together with the corresponding field of view. The moving input camera, whose optical centers are located at positions $\mathbf{c}(t) = (X_t, 0, 0)$, generates images according to the following mapping:

$$\mathbf{p} = (X, Y, Z) \implies p = (x, y) = (f\frac{X - X_t}{Z}, f\frac{Y}{Z}) \quad (2)$$

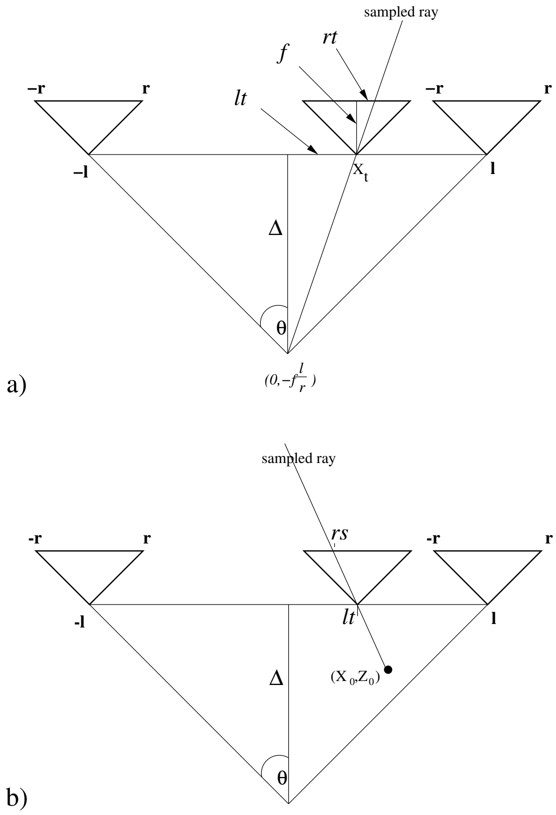


Figure 11. New image formation with two possible positions of the vertical slit (see text).

We denote the range of columns (x) in each pinhole image as $[-r, r]$, and the range of camera pinhole positions (X_t) as $[-l, l]$ (see Fig. 11a). The new synthesized image is constructed by pasting columns from the input images. The range of columns in the synthesized image is $[-(r + l), r + l]$. For each $t \in [-1, 1]$, we assign to the $(l + r)t$ column of the new image the image values at the rt column of the pinhole camera positioned at $(lt, 0, 0)$ (i.e., $X_t = lt$, see Fig. 11a). It now follows from Eq. (2) that $rt = f\frac{X-lt}{Z}$. In addition, for each column $x \in [-(r + l), (r + l)]$ in the new image, $t = \frac{x}{l+r}$ and therefore

$$X = \frac{rt}{f}Z + lt = x\left(\frac{r}{l+r} \cdot \frac{Z}{f} + \frac{l}{l+r}\right)$$

or

$$x = \frac{l+r}{r}f \cdot \frac{X}{Z + f\frac{l}{r}}$$

Observe that this defines a vertical slit at $Z = -f\frac{l}{r}$ (see Fig. 11a). The horizontal slit is at $Z = 0$, same as for each pinhole camera. Eq. (2) therefore becomes the following projection

$$\mathbf{p} = (X, Y, Z) \implies p = (x, y) = (f_x\frac{X}{Z + \Delta}, f_y\frac{Y}{Z}) \quad (3)$$

where $f_x = \frac{l+r}{r}f$ is the horizontal focal length, $f_y = f$ is the vertical focal length, and $\Delta = f\frac{l}{r}$ is the distance between the two slits.

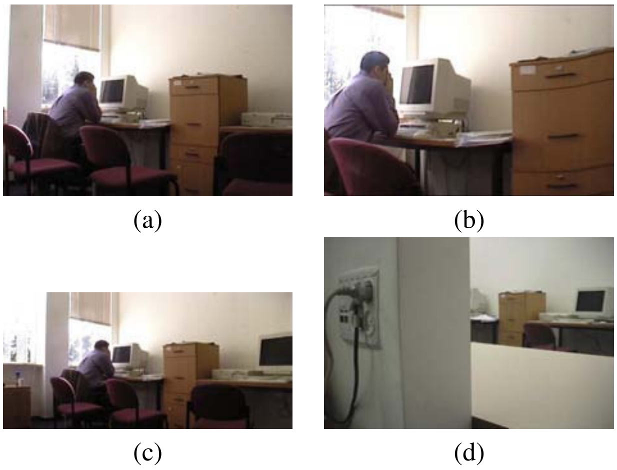


Figure 12. This scene is located in a narrow room where moving backward to capture the entire room is impossible. The scene was filmed by a sideways moving camera, total of 591 frames; one of the original frames is shown in (a). We show two synthetic images: one where the vertical slit is located in front of the original track (b), and one where the vertical slit is located behind the original track (c). For comparison, we took a normal (pinhole) picture from the same location as (c), where part of the scene is obscured by the wall; this picture is shown in (d), and it demonstrates our ability to make images from impossible camera positions.

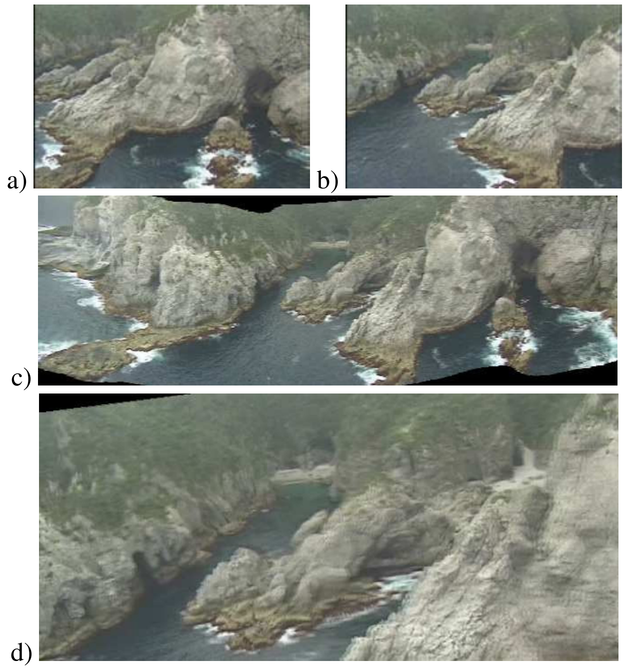


Figure 13. Virtual walkthrough generated from a sequence taken by a flying helicopter. a, b) Two frames from the input sequence. c, d) Two images rendered in forward motion (diagonal slices). Note the change of visible areas as the camera moves (synthetically) forward.

Suppose next that instead of taking the rt column from the camera at $(lt, 0, 0)$, we choose an arbitrary linear column sampling function. More specifically, for $t = \alpha s + \beta$, we take the rs column of the lt camera, see Fig. 11b. (Recall that r, l are fixed, while t, s are free parameters which determine the rate of column sampling). Let the field of view of the original pinhole camera be 2θ. It can be shown that such a choice of columns defines the mapping

$$(x, y) = ((f + \frac{\alpha l}{\tan\theta})\frac{X - \beta l}{\frac{\alpha l}{\tan\theta} + Z}, f\frac{Y}{Z}) \qquad (4)$$

This can be written simply as

$$(x, y) = (f_x \frac{X - X_0}{\Delta + Z}, f_y \frac{Y}{Z}) \qquad (5)$$

where $X_0 = \beta l$, $\Delta = \frac{\alpha l}{\tan\theta}$, $f_y = f$ and $f_x = f + \Delta$.

The method described so far produces images which do not follow the perspective projection model, since the focal lengths are not the same vertically and horizontally. They do, however, follow the X-Slits projection model defined above. To see this, we observe that all rays producing the image must intersect the following two lines:

1. The line of camera motion; this is because each projection ray must be collected by some camera whose optical center is on this line.

2. The vertical line located at (X_0, Z_0) (as in Eq. (5), where $Z_0 = Z + \Delta$); this is shown above to be the vertical slit.

The projection model is therefore defined by a family of rays intersecting a pair of lines (*"slits"*), projecting 3D points onto a plane. Moreover, the model is X-Slits (compare Eq. (5) with Eq. (1)).

In the derivation leading to Eq. (5) we effectively showed that any linear sampling function yields a valid new X-Slits image. Furthermore, we can set the location of the vertical slit to (X_0, Z_0) by fixing $\alpha = -\frac{Z_0}{l}\tan\theta$ and $\beta = \frac{X_0}{l}$. This result enables us to synthesize new views of the scene with any vertical slit of our choice, by sampling the columns of the original input sequence according to $t = \alpha s + \beta$, with α and β assigned the appropriate values.

3.3.1 Aspect-Ratio Normalization

The most apparent aspect of the distortion in X-Slits images is the variation of aspect-ratio. To reduce this distortion, we vertically scale the new images. This normalization is essential to achieve compelling results.

Specifically, the distortion on the *image plane* of objects at depth Z can be written as $\frac{Z}{Z+\Delta} \cdot \frac{f_y}{f_x}$ in the notation of Eq. (5). In order to keep the horizontal field-of-view angle constant in the walk-through animation, we sample all the columns from left to right (from the appropriate frames, according to the column sampling function). Without any scaling, this process generates an image in which only the plane at infinity ($Z = \infty$) appears undistorted. Therefore, in order to cancel the distortion at depth Z_0, we must scale the image vertically by the factor:

$$1 + \frac{\Delta}{Z_0} \qquad (6)$$

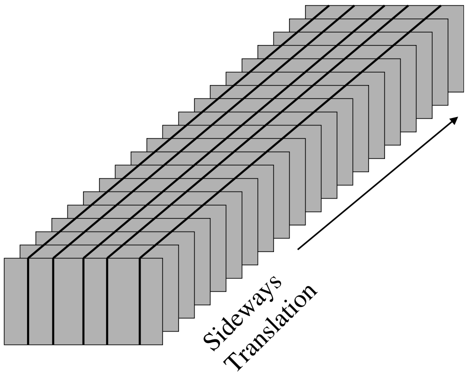


Figure 14. Parallel slices of the rectified space-time volume. Every pais of images generates from a pair of parallel planar slices represents a panoramic stereo pair.

3.4. X-Slits Experimental Results

As discussed above, new view generation is done by slicing the rectified space-time volume with a planar time front. The parameters of the time front determine the location of the vertical slit. In the synthesized movies we manipulated the time front so that the location of the vertical slit moved according to the desired ego-motion. Although the horizontal slit of the remained fixed, our results demonstrate a very compelling impression of synthetic camera motion. In Fig. 12 we show an example indoors, and in Fig. 13 we show an outdoors example.

3.5. Panoramic Stereo Imaging

When we slice the rectified space-time volume with parallel slices, every pair of images is a stereo pair. This slicing is shown on Fig. 14.

Fig 15 shows that a parallel slice on the left collects parallel rays directed from left to right, while a parallel slice on the right collects rays directed from left to right. As each generated panoramic image has a different viewing direction, every two such images are a stereo pair. Generation of panoramic stereo images is discussed in detail in [**?**].

Panoramic stereo images generated by parallel slicing of the rectified space time volume are shown in Fig. 16.

3.6. X-Slits Discussion

X-Slits is a new non-perspective projection model, which is defined by two slits and a projection surface. The main application used in this paper is new view generation, or image based rendering. New view generation with the X-Slits camera is greatly simplified as compared with perspective new view generation, as it is performed by slicing the space-time volume by a planar time front. The X-Slits theory helps the user to "drive" the slicing process in order to get the desired effect. When compared to traditional mosaicing, X-Slits images can be shown to be closer to perspective images than linear pushbroom images.

Using our method we can also generate new images taken from "impossible" positions, like behind the back wall of a room or in front of a glass barrier. Movies with new ego motion can also be generated, such as forward-moving movies from a side-moving input sequence. Although not perspective, the movies generated in this way appear compelling and realistic.

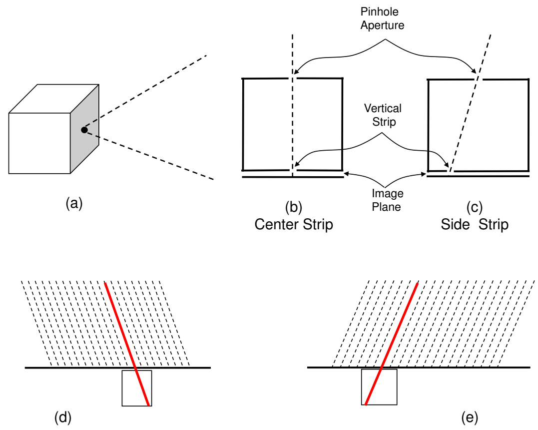


Figure 15. Mosaicing together rays from the center of the images (b), as done in a parallel slice at the center of the rectified space-time volume, generated an image of parallel rays perpendicular to the image plane. Mosaicing together rays from the side of the input images (c), as done in a parallel slice at the side of the space-time volume, will generate an image of rays slanted relative to the image plane (e).

Figure 16. A pair of panoramic stereo views generated from a video camera mounted on a helicopter, scanning a derailed Shinkansen train. The two images were generated from two parallel slices of the rectified space-time volume.

4. Concluding Remarks

It was shown that by relaxing the chronological constraints of time, a flexible representation of dynamic videos can be obtained. Specifically, when the chronological order of events is no longer considered a hard restriction, a wide range of time manipulations can be applied. An interesting example is creating dynamic panoramas where all events occur simultaneously, and the same principles hold even for videos taken by a static camera.

Manipulating the time in movies is performed by sweeping an evolving time front through the aligned space-time volume. The strength of this approach is that accurate segmentation and recognition of objects are not needed. This fact significantly simplifies and increases the robustness of the method. This robustness comes at a cost of limiting the time manipulations that can be applied on a given video. Assume that one moving object occludes another moving object. With our method, the concurrency of the occlusion must be preserved for both objects.

We have also shown that when a camera translated in a rigid scene we can get 3D effects by slicing the rectified space-time volume with planar slices. This included both walkthrough effects, and the generation of panoramic stereo images.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *SIGGRAPH 2004*, pages 294–302, August 2004.

[2] A. Agarwala, C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski. Panoramic video textures. In *SIGGRAPH 2005*, pages 821–827, July 2005.

[3] D. Aliaga and I. Carlbom. Plenoptic stitching: A scalable method for reconstructing interactive walkthroughs. In *SIGGRAPH'01*, pages 443–450, 2001.

[4] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV'92*, pages 237–252, Italy, May 1992.

[5] C. Buehler, M. Bosse, S. Gortler, M. Cohen, and L. McMillan. Unstructured lumigraph rendering. In *SIGGRAPH'01*, pages 425–432, 2001.

[6] G. Doretto and S. Soatto. Towards plenoptic dynamic textures. In *Texture'03*, pages 25–30, Nice, France, October 2003.

[7] W. Freeman and H. Zhang. Shape-time photography. In *CVPR'03*, volume II, pages 151–157, 2003.

[8] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *SIGGRAPH'96*, pages 43–54, 1996.

[9] S. Hsu, H. S. Sawhney, and R. Kumar. Automated mosaics via topology inference. *IEEE Trans. on Computer Graphics and Applications*, 22(2):44–54, 2002.

[10] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu. Mosaic representations of video sequences and their applications. *Signal Processing: Image Communication*, 8(4):327–351, May 1996.

[11] H. Ishiguro, M. Yamamoto, and S. Tsuji. Omnidirectional stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):257–262, 1992.

[12] A. Klein, P. Sloan, A. Colburn, A. Finkelstein, and M. Cohen. Video cubism. Technical Report MSR-TR-2001-45, Microsoft Research, 2001.

[13] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, 22(3):277–286, July 2003.

[14] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH'96*, pages 31–42, 1996.

[15] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH'95*, pages 39–46, Los Angeles, California, August 1995. ACM.

[16] T. Pajdla. Stereo with oblique cameras. In *IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 85–91, Hawaii, December 2001.

[17] S. Peleg, M. Ben-Ezra, and Y. Pritch. Omnistereo: Panoramic stereo imaging. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(3):279–290, March 2001.

[18] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(10):1144–1154, October 2000.

[19] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg. Dynamosaicing: Video mosaics with non-chronological time. In *CVPR '05*, San Diego, June 2005.

[20] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg. Dynamosaicing: Mosaic of dynamic scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, page in press, 2007.

[21] A. Rav-Acha, Y. Shor, and S. Peleg. Mosaicing with parallax using time warping. In *Second IEEE Workshop on Image and Video Registration*, Washington, DC, July 2004.

[22] S. Seitz. The space of all stereo images. In *ICCV'01*, volume I, pages 26–33, Vancouver, Canada, July 2001.

[23] H. Shum and L. He. Rendering with concentric mosaics. In *SIGGRAPH'99*, pages 299–306, Los Angeles, California, August 1999. ACM.

[24] T. Takahashi, H. Kawasaki, K. Ikeuchi, and M. Sakauchi. Arbitrary view position and direction rendering for large-scale scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 296–303, Hilton Head, SC, June 2000.

[25] P. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Journal of Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[26] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating ghosting and exposure artifacts in image mosaics. In *CVPR'01*, volume II, pages 509–516, Hawaii, December 2001.

[27] J. Zheng and S. Tsuji. Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9:55–76, 1992.

[28] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(6):741–754, June 2003.