Mosaicing New Views: The Crossed-Slits Projection

Assaf Zomet, Doron Feldman, Shmuel Peleg, *Member*, *IEEE*, and
Daphna Weinshall, *Member*, *IEEE Computer society*

Abstract—We introduce a new kind of mosaicing, where the position of the sampling strip varies as a function of the input camera location. The new images that are generated this way correspond to a new projection model defined by two slits, termed here the Crossed-Slits (X-Slits) projection. In this projection model, every 3D point is projected by a ray defined as the line that passes through that point and intersects the two slits. The intersection of the projection rays with the imaging surface defines the image. X-Slits mosaicing provides two benefits. First, the generated mosaics are closer to perspective images than traditional pushbroom mosaics. Second, by simple manipulations of the strip sampling function, we can change the location of one of the virtual slits, providing a virtual walkthrough of a X-slits camera; all this can be done without recovering any 3D geometry and without calibration. A number of examples where we translate the virtual camera and change its orientation are given; the examples demonstrate realistic changes in parallax, reflections, and occlusions.

Index Terms—Nonstationary mosaicing, crossed-slits projection, pushbroom camera, virtual walkthrough, image-based rendering.

1 Introduction

Perspective projection forms the foundation of imaging. Since our eyes, as well as most of our cameras, observe the world through a pinhole (via a lens whose effects we shall ignore), we are used to viewing images that are generated by perspective projections. Therefore, techniques for the generation of new views are designed to achieve the effects of perspective projection. What happens if we relax this requirement? Can we do better computationally when not limited by the perspective projection, but are free to use other projection models? In this paper, we introduce and study an alternative projection model, defined by two slits—the Crossed-Slits (X-Slits) projection. In the X-Slits model, 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.

Curiously, a physical X-Slits camera was designed in the 19th century by one of the pioneers of color photography, Ducos du Hauron [10], under the title "transformisme en photographie." Ducos du Hauron thought that X-Slits images would be used in the following (20th) century to "create visions of another world" [14]. A century later, Rudolf Kingslake reviewed this device in his book [10]; in his analysis, Kingslake concludes that "the pair of slits working together thus constitutes a pinhole camera in which the image is stretched or compressed in one direction more than in the other." This should make this exotic device rather useful for the new emerging technology of wide-screen cinematography. However, as we show below, the X-Slits projection does much more to images than horizontal stretching.

Independent of the physical device, we argue that the X-Slits projection model is useful and worthy of our attention. This is because new X-Slits images can be easily generated by mosaicing a sequence of images captured by a translating pinhole camera, and because those images look compelling and realistic.

The mosaicing process is done by sampling vertical strips from the sequence of frames and pasting them together.[1] Unlike traditional mosaicing techniques [16], [18], [25], [8], the strip sampling location may vary as a function of the location of the input translating camera. We thus call it "nonstationary" mosaicing. Mosaic images obtained in this way are more similar to perspective images than traditional mosaic images.

1.1 The X-Slits Camera: Outline

Fig. 1a shows the basic design of the X-Slits camera as built by Ducos du Hauron in 1888 [10]. A more general design is shown in Fig. 1b. A X-Slits camera has two slits $\mathbf{l}_1, \mathbf{l}_2$ which should be two different lines in 3D space, and an image plane Π that does not contain any of the slits. For every 3D point not lying on either of the slits, there is a single ray that connects the point with both slits simultaneously. The intersection of this ray with the image plane defines the projected image of the 3D point. The camera in Fig. 1a is a special case of the X-Slits camera, where the two slits are orthogonal to each other and parallel to the image plane. We call this special arrangement the Parallel-Orthogonal X-Slits camera (POX-Slits camera).

The X-Slits model is a valid 3D to 2D projection, defining a many-to-one mapping from the 3D world to the 2D image plane. In Section 2, we develop the specific equations of the camera mapping as a function of the slits $\mathbf{l}_1, \mathbf{l}_2$ and the image plane Π.

In Section 3, we discuss how to generate new X-Slits images from images taken by a regular pinhole camera translating along a line in 3D space. We show that virtual

- *The authors are with the School of Computer Science and Engineering, The Hebrew University of Jerusalem, 91904 Jerusalem, Israel. E-mail: {zomet, doronf, peleg, daphna}@cs.huji.ac.il.*

Manuscript received 25 June 2002; revised 7 Nov. 2002; accepted 2 Dec. 2002. Recommended for acceptance by S. Seitz.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 116845.

[1] We often refer to the vertical strips as "columns," with the understanding that they can be wider than a single pixel if necessary.

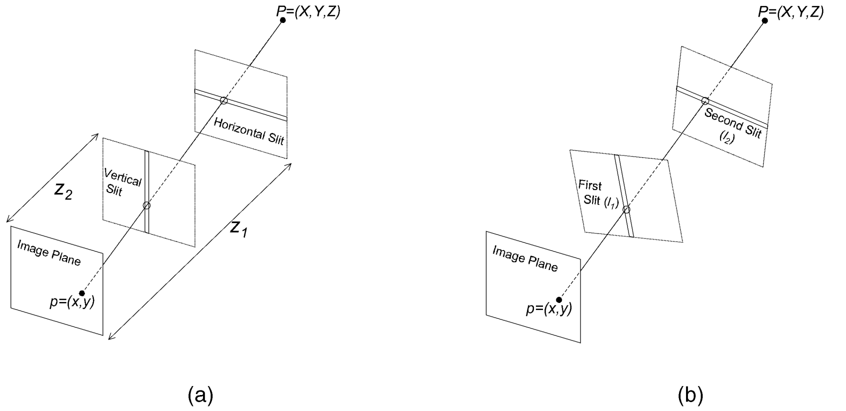


Fig. 1. (a) A design of a X-Slits camera where the slits are orthogonal to each other and parallel to the image plane (POX-Slits camera). The projection ray of a 3D point $\mathbf{p} = (\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ is shown, with circles showing its intersection points with the two slits. (b) A general X-Slits design, with two arbitrary slits $\mathbf{l}_1, \mathbf{l}_2$. Note that the camera is defined by the specific configuration of the two slits and the image plane; any of these three factors can change independently, giving rise to a different X-Slits camera.

X-Slits images can be generated by mosaicing of perspective images, or equivalently, by slicing the space-time volume[2] of images. A readily available tool for slicing this volume is the "video cube" [11]. X-Slits projection models with nonlinear slits or nonplanar image planes are discussed in Section 4.

Applications are discussed in Section 5. The first application includes the generation of an arbitrary virtual walkthrough from a single sequence of images without using any 3D model. The results show that, in many practical cases, the discrepancies between X-Slits images and perspective images are hardly noticeable. The second application is 3D object visualization. The X-Slits model allows us to virtually place one slit behind an object, keeping the other slit in front of the object. Such visualization can be useful when one wants to see simultaneously different sides of an object. We note that both applications are done by plain mosaicing, without recovering the scene or the object model.

1.2 Relation to Previous Work

There has been much work on nonperspective projection models [15] and the generation of nonperspective images from video sequences by mosaicing [16], [18], [25], [8]. 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 [13], [5], [12]. 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 [4], [1], [21], or to use such approximations as moving the camera in a lower dimensional space [20], [22].

The present work is mostly related to [20], [22], [4], [21] 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, our rendering tool is very simple—slicing of the space-time volume obtained by a simple motion of a 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 and constant, every vertical planar slice of the space-time volume gives *some* valid X-Slits image.

The idea of using linear nonstationary mosaicing for IBR and the generation of a virtual walkthrough was originally reported in [23]. But, while the bicentric camera model analyzed in [23] is mathematically equivalent to the POX-Slits model analyzed below, the general X-Slits model and the realization with two slits are new. This realization allows for a wider range of applications, as discussed in Section 5.

2 The X-Slits Projection

In Section 2.1, we derive the projection model of the X-Slits camera. Properties of this model are discussed in Section 2.2. Special configurations of slits and planes are discussed in Section 2.3. All derivations are done in the projective spaces $\mathcal{P}^2$ and $\mathcal{P}^3$, where points in 2D and 3D, respectively, are represented by homogeneous coordinates.

2.1 Projection Geometry

Consider the camera configuration as shown in Fig. 1b. The *projection ray* of a point $\mathbf{p} \in \mathcal{P}^3$ intersects the two camera slits $\mathbf{l}_1, \mathbf{l}_2$. It is thus the intersection of two planes, defined by joining the point $\mathbf{p}$ with each of the slits. The image of $\mathbf{p}$ is the intersection of this projection ray with the image plane. Furthermore, all points on a projection ray will project to the same image point (unless the ray lies on the image plane).

More formally, let $\mathbf{u}_i, \mathbf{v}_i$ denote two planes in $\mathcal{P}^3$ which contain slit $\mathbf{l}_i$, for $i = 1, 2$, respectively. For any α,[3] $\mathbf{u}_i + \alpha \mathbf{v}_i$ is also a plane which contains the slit $\mathbf{l}_i$ and vice versa, every plane that contains the slit can be described by $\mathbf{u}_i + \alpha \mathbf{v}_i$ for some α. Thus, for each scene point $\mathbf{p} \in \mathcal{P}^3$ and

[2] The space-time volume is obtained by stacking the input images, a.k.a. the epipolar volume.

[3] α is a projective parameter as in [19 p. 43].

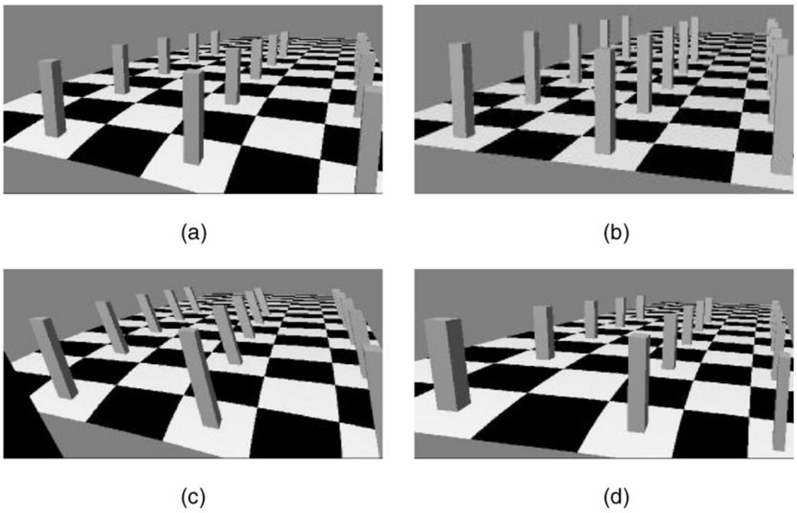


Fig. 2. Simulated X-Slits images of an artificial scene. (a) POX-Slits image using the projection equation from (5). (b) A regular pinhole image. (c) Same as (a), but with the vertical slit rotated about the Z axis. (d) Same as (a), but with the vertical slit rotated about the X axis.

$i = 1, 2$, there exist $\alpha \in \mathcal{R}$ such that $(\mathbf{u}_i + \alpha\mathbf{v}_i)\mathbf{p} = 0$. Eliminating α, we get the following expression for the plane:

$$\mathbf{u}_i - \frac{(\mathbf{u}_i^T\mathbf{p})}{(\mathbf{v}_i^T\mathbf{p})}\mathbf{v}_i \quad \propto \quad (\mathbf{v}_i^T\mathbf{p})\mathbf{u}_i - (\mathbf{u}_i^T\mathbf{p})\mathbf{v}_i \quad = \quad (\mathbf{u}_i\mathbf{v}_i^T - \mathbf{v}_i\mathbf{u}_i^T)\mathbf{p}. \tag{1}$$

Let us define the skew-symmetric matrix

$$\mathbf{S}_i = \mathbf{u}_i\mathbf{v}_i^T - \mathbf{v}_i\mathbf{u}_i^T. \tag{2}$$

Matrix $\mathbf{S}_i$ is actually the dual Plucker matrix representation of the line $\mathbf{l}_i$ [7]. It can be shown to be independent of the choice of $\mathbf{u}_i$, $\mathbf{v}_i$. The elements of such matrices must satisfy one nonlinear constraint (since a line has only 4 degrees of freedom up to scale).

From (1)-(2), it follows that the plane which contains point $\mathbf{p}$ and slit $\mathbf{l}_i$ is $\mathbf{S}_i\mathbf{p}$. The projection ray of the scene point $\mathbf{p}$ is therefore defined by the intersection of the two planes $\mathbf{S}_i\mathbf{p}, i = 1, 2$ defined in (2).

Next, we observe that the image of scene point $\mathbf{p}$ is the intersection of the projection ray with the image plane Π. Let $\mathbf{m} \in \mathcal{P}^3$ denote a point on plane Π, and let $\mathbf{j}, \mathbf{k} \in \mathcal{P}^3$ denote two distinct points at infinity which also lie on plane Π. Every point on Π can be expressed as $x\mathbf{j} + y\mathbf{k} + w\mathbf{m}$. The projection ray of $\mathbf{p}$ intersects plane Π at a certain point $\mathbf{q} \propto x\mathbf{j} + y\mathbf{k} + w\mathbf{m}$ such that $(\mathbf{S}_1\mathbf{p})^T\mathbf{q} = 0$ and $(\mathbf{S}_2\mathbf{p})^T\mathbf{q} = 0$. This gives us a set of linear equations in x, y, and w (the homogeneous image coordinates of point $\mathbf{q}$), namely,

$$\begin{bmatrix} \mathbf{p}^T\mathbf{S}_1\mathbf{j} & \mathbf{p}^T\mathbf{S}_1\mathbf{k} & \mathbf{p}^T\mathbf{S}_1\mathbf{m} \\ \mathbf{p}^T\mathbf{S}_2\mathbf{j} & \mathbf{p}^T\mathbf{S}_2\mathbf{k} & \mathbf{p}^T\mathbf{S}_2\mathbf{m} \end{bmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} = 0$$

and the solution of this linear system is[4]

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \propto \begin{pmatrix} \mathbf{p}^T\mathbf{S}_1(\mathbf{k}\mathbf{m}^T - \mathbf{m}\mathbf{k}^T)\mathbf{S}_2\mathbf{p} \\ \mathbf{p}^T\mathbf{S}_1(\mathbf{m}\mathbf{j}^T - \mathbf{j}\mathbf{m}^T)\mathbf{S}_2\mathbf{p} \\ \mathbf{p}^T\mathbf{S}_1(\mathbf{j}\mathbf{k}^T - \mathbf{k}\mathbf{j}^T)\mathbf{S}_2\mathbf{p}. \end{pmatrix}.$$

Let us define the following skew-symmetric matrices

$$\mathbf{Q}_1 = \mathbf{k}\mathbf{m}^T - \mathbf{m}\mathbf{k}^T, \quad \mathbf{Q}_2 = \mathbf{m}\mathbf{j}^T - \mathbf{j}\mathbf{m}^T, \quad \mathbf{Q}_3 = \mathbf{j}\mathbf{k}^T - \mathbf{k}\mathbf{j}^T.$$

It follows that

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \propto \begin{pmatrix} \mathbf{p}^T\mathbf{S}_1\mathbf{Q}_1\mathbf{S}_2\mathbf{p} \\ \mathbf{p}^T\mathbf{S}_1\mathbf{Q}_2\mathbf{S}_2\mathbf{p} \\ \mathbf{p}^T\mathbf{S}_1\mathbf{Q}_3\mathbf{S}_2\mathbf{p} \end{pmatrix} \tag{3}$$

and $\mathbf{p}$ is projected to $(\frac{x}{w}, \frac{y}{w})$. Note that $\mathbf{S}_i$ depends only on the optical slit l_i, while $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3$ depend only on the plane of projection Π. $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$ are the Plucker matrix representations of the image X axis, the image Y axis, and the image line at infinity, respectively (all in 3D). Finally, note that the choice of $\mathbf{j}$ and $\mathbf{k}$ to be points at infinity limits the internal calibration to an affine transformation of the image.

The camera projection model in (3) is quadratic and, therefore, a X-Slits camera cannot be described by a 3×4 projection matrix in $\mathcal{P}^3$. However, it can be described by a $3 \times 4 \times 4$ tensor $\mathcal{M}^i_{jk}$ in $\mathcal{P}^3$ as follows

$$p^i \propto \mathcal{M}^i_{jk}\mathbf{p}^j\mathbf{p}^k,$$

where

$$\mathcal{M}^i_{jk} = (\mathbf{S}_1)_{kl}\mathbf{Q}^{ilm}(\mathbf{S}_2)_{mj}.$$

2.2 Properties of the X-Slits Camera

To get some intuition for X-Slits images, Fig. 2 shows examples of X-Slits images as compared with pinhole images of the same scene. We also show the effects of varying the relative geometry of the slits and image plane. Next, we demonstrate algebraically a few interesting properties of X-Slits images.

2.2.1 Projection of Lines

In X-Slits images, the image of a 3D line is a conic. Let $\mathbf{q}_1 + \lambda\mathbf{q}_2$ denote a line $\mathbf{l}$ which goes through the two points $\mathbf{q}_1, \mathbf{q}_2 \in \mathcal{P}^3$. For each point on $\mathbf{l}$, $\mathbf{q}_1 + \lambda\mathbf{q}_2$ for some $\lambda \in \mathcal{R}$, define the planes $\mathbf{S}_1(\mathbf{q}_1 + \lambda\mathbf{q}_2)$ and $\mathbf{S}_2(\mathbf{q}_1 + \lambda\mathbf{q}_2)$ which

[4] The solution is a unique point unless $\mathbf{p}$ resides on the line joining the intersections of the two slits with the image plane.

contain the point and each of the two slits. The intersection of these planes gives the projection ray of that point.

Consider the surface which is the union of all such projection rays for all $\lambda \in \mathcal{R}$. Let $\mathbf{p}$ denote a point on this surface. It follows that $\mathbf{p}^T\mathbf{S}_1(\mathbf{q}_1 + \lambda\mathbf{q}_2) = 0$ and $\mathbf{p}^T\mathbf{S}_2(\mathbf{q}_1 + \lambda\mathbf{q}_2) = 0$ for some $\lambda \in \mathcal{R}$. Therefore,

$$\frac{\mathbf{p}^T\mathbf{S}_1\mathbf{q}_1}{\mathbf{p}^T\mathbf{S}_1\mathbf{q}_2} = -\lambda = \frac{\mathbf{p}^T\mathbf{S}_2\mathbf{q}_1}{\mathbf{p}^T\mathbf{S}_2\mathbf{q}_2}.$$

Thus, $\mathbf{p}^T\mathbf{S}_1(\mathbf{q}_1\mathbf{q}_2^T - \mathbf{q}_2\mathbf{q}_1^T)\mathbf{S}_2\mathbf{p} = 0$ which means that the surface is a quadric. Since the intersection of a quadric with a plane is a conic, the X-Slits image of a line $\mathbf{l}$ is always a conic.

The distortion of straight lines is illustrated in Fig. 2. In practice, this distortion is not very disturbing, as can be seen in the examples in Section 5. In the various scenes we have experimented with, this distortion was rather minor. Many scenes, particularly natural scenes, do not have very dominant straight lines, in which case this distortion is hardly noticed. Furthermore, people are accustomed to similar effects caused by lens distortions.

2.2.2 Transforming the Image Plane

What is the relation between two X-Slits images which are generated by the same slits, but different projection planes Π and Π'? Note that the set of rays intersecting the image plane are determined by the two slits. Thus, there exists a mapping between the two image planes that is invariant to the 3D structure of the viewed scene, similarly to the case of rotation in pinhole cameras.

Specifically, we derive the relation between a point (x', y', w') on Π' and a point (x, y, w) on Π, both of which are the projection of the same scene point $\mathbf{p}$. Recall that the 3D point which corresponds to (x, y, w) on Π is $\mathbf{q} \propto x\mathbf{j} + y\mathbf{k} + w\mathbf{m}$. Similarly, (x', y', w') on Π' corresponds to $\mathbf{q}' \propto x'\mathbf{j}' + y'\mathbf{k}' + w'\mathbf{m}'$. By definition, $\mathbf{q}$ lies on the planes $\mathbf{S}_1\mathbf{p}$ and $\mathbf{S}_2\mathbf{p}$ and, therefore, $\mathbf{q}$ is also projected to (x', y', w') on Π'. Denoting

$$\mathbf{M} = (\mathbf{j} \quad \mathbf{k} \quad \mathbf{m}), \quad p = \begin{pmatrix} x \\ y \\ w \end{pmatrix},$$

it follows that $\mathbf{q} = \mathbf{M}p$. We now project $\mathbf{M}p$ on Π' to obtain

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} \propto \begin{pmatrix} p^T\mathbf{M}^T\mathbf{S}_1\mathbf{Q}'_1\mathbf{S}_2\mathbf{M}p \\ p^T\mathbf{M}^T\mathbf{S}_1\mathbf{Q}'_2\mathbf{S}_2\mathbf{M}p \\ p^T\mathbf{M}^T\mathbf{S}_1\mathbf{Q}'_3\mathbf{S}_2\mathbf{M}p \end{pmatrix}. \tag{4}$$

Note that $\mathbf{M}$ depends only on plane Π, $\mathbf{S}_1$ and $\mathbf{S}_2$ depend only on slits $\mathbf{l}_1$ and $\mathbf{l}_2$, respectively, and $\mathbf{Q}'_1$, $\mathbf{Q}'_2$, $\mathbf{Q}'_3$ depend only on plane Π'.

The relation between the two image points $p = (x, y, w)^T$ and $p' = (x', y', w')^T$ is therefore not a homography as in the pinhole case. But, similar to the above, we can write this relations using a $3 \times 3 \times 3$ tensor $\mathcal{H}^i_{jk}$ in $\mathcal{P}^2$

$$p'^i \propto \mathcal{H}^i_{jk}p^jp^k,$$

where

$$\mathcal{H}^i_{jk} = (\mathbf{M}^T)^n_k(\mathbf{S}_1)_{ln}(\mathbf{Q}')^{iml}(\mathbf{S}_2)_{pm}\mathbf{M}^p_j.$$

2.3 Special Case: The POX-Slits Projection

A configuration of special interest is when the slits are orthogonal to each other and parallel to the image plane. We will refer to this configuration as POX-Slits (see Fig. 1a).

Without loss of generality, we fix the slits by assigning $\mathbf{u}_1 = (1, 0, 0, 0)^T$, $\mathbf{v}_1 = (0, 0, 1, -Z_1)^T$, $\mathbf{u}_2 = (0, 1, 0, 0)^T$, and $\mathbf{v}_2 = (0, 0, 1, -Z_2)$. This defines a vertical slit at $X = 0$, $Z = Z_1$ and a horizontal slit at $Y = 0$, $Z = Z_2$. We also denote $\Delta = Z_2 - Z_1$. Plane Π is the $X - Y$ plane at $Z = 0$ and, therefore, we assign $\mathbf{m} = (0, 0, 0, 1)^T$, $\mathbf{j} = (1, 0, 0, 0)^T$, $\mathbf{k} = (0, 1, 0, 0)^T$. From (3), it follows that

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -Z_1 \frac{X}{Z-Z_1} \\ -Z_2 \frac{Y}{Z-Z_2} \end{pmatrix}. \tag{5}$$

These projection equations are identical to the model analyzed in [23], where it was called bicentric projection.

2.3.1 Projection of Lines

Recall that, under the perspective projection, straight lines are projected to straight lines and, under the X-Slits projection, straight lines are projected to conic sections. Let us look more closely at the images of lines in the POX-Slits case. If the 3D line is perpendicular to the Z axis, i.e., of the form $(a + c\lambda, b + d\lambda, e)$, then its image is the curve $(x, y) = \left(-Z_1 \frac{a+c\lambda}{e-Z_1}, -Z_2 \frac{b+d\lambda}{e-Z_2}\right)$, which is a line. If the 3D line is *not* perpendicular to the Z axis, i.e., of the form $(a + c\lambda, b + d\lambda, \lambda)$, then its image is the curve

$$(x, y) = \left(-Z_1 \frac{a + c\lambda}{\lambda - Z_1}, -Z_2 \frac{b + d\lambda}{\lambda - Z_2}\right). \tag{6}$$

Solving for λ, we find that the line is projected to the hyperbola given by

$$\begin{aligned} &(Z_1 - Z_2)xy + Z_2(b + Z_1d)x \\ &\qquad - Z_1(a + Z_2c)y + Z_1Z_2(bc - ad) = 0. \end{aligned} \tag{7}$$

For $Z_1 \neq Z_2$, this hyperbola degenerates to a line if and only if $a + Z_1c = 0$ or $b + Z_2D = 0$, that is, only if the line intersects one of the slits.

2.3.2 Aspect Ratio Distortions

The most apparent aspect of the distortion in POX-Slits images is the variation of aspect-ratio (especially in pushbroom images [6]). The apparent aspect-ratio of objects in the image depends on their depth. This is unlike the perspective model, in which the distortion in aspect-ratio is constant for all objects.

From (5), it follows that an object at depth Z with aspect-ratio of 1 would appear on the image plane to have an aspect-ratio of

$$\frac{\Delta y}{\Delta x} = \frac{Z_2}{Z_1} \cdot \frac{Z - Z_1}{Z - Z_2}. \tag{8}$$

In practice, we found this distortion to be typically rather insignificant. If the range of depth values of scene objects is not too large, we can normalize the image to compensate for this distortion by scaling, as discussed in Section 3.2.2. Specifically, if we cancel the aspect-ratio distortion for some intermediate depth value Z_0, the distortion at depth Z would be

$$\frac{Z - Z_1}{Z - Z_2} \cdot \frac{Z_0 - Z_2}{Z_0 - Z_1}. \tag{9}$$

To demonstrate the magnitude of the distortion, consider the following example: Suppose the depth range of objects in the scene is $3 - 5$ meters (measured from the horizontal slit at Z_1, i.e., $3m < Z - Z_1 < 5m$), and assume that the images are normalized so that objects at the depth of $3.84m$ appear undistorted (i.e., $Z_0 - Z_1 = 3.84m$). If the vertical slit is behind the horizontal slit at $\Delta = -2.5m$, the aspect-ratio distortion would not exceed 10 percent.

2.3.3 Rotated Image Plane

In order to see what happens when the image plane is rotated, we study the correspondence between POX-Slits image points and image points on a rotated plane. Here, for convenience and without loss of generality, we fix a different configuration of perpendicular slits by assigning $\mathbf{u}_1 = (1, 0, 0, 0)^T$, $\mathbf{v}_1 = (0, 0, 1, 0)^T$, and $\mathbf{u}_2 = (0, 1, 0, 0)^T$, $\mathbf{v}_2 = (0, 0, 1, Z_1 - Z_2)$. We fix the parallel plane Π at $\mathbf{m} = (0, 0, -Z_1, 1)^T$, $\mathbf{j} = (1, 0, 0, 0)^T$, $\mathbf{k} = (0, 1, 0, 0)^T$, and the oriented plane Π' at $\mathbf{m}' = (m_1, m_2, m_3, 1)^T$, $\mathbf{j}' = (j_1, j_2, j_3, 0)^T$, $\mathbf{k}' = (k_1, k_2, k_3, 0)^T$. This configuration consists of a vertical slit at $Z = 0$, a horizontal slit at $Z = Z_2 - Z_1$, and the parallel plane at $Z = -Z1$, which is exactly the POX-Slits configuration, translated to align the vertical slit with the Y axis.

We now rotate the image plane about the vertical slit. In order to find the transformation from Π to Π', we substitute the above into (4) to obtain:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -Z_1 \frac{j_1 x' + k_1 y' + m_1}{j_3 x' + k_3 y' + m_3} \\ -Z_2 \frac{j_2 x' + k_2 y' + m_2}{j_3 x' + k_3 y' + m_3 + Z_1 - Z_2} \end{pmatrix}. \tag{10}$$

This transformation allows us to readily produce the images corresponding to oriented image planes, given the corresponding POX-Slits image (with parallel image plane). This will come in handy in the experiments described in Section 5. Note that the transformation is "almost" a homography, degenerating to a homography when $Z_1 = Z_2$ (the pinhole case).

2.3.4 Tilted Slit

Suppose we tilt the vertical slit sideways. Assigning all substitutions as with the POX-Slits configuration, except that $\mathbf{u}_1 = (1, a, 0, 0)^T$ for some $a \in \mathcal{R}$, we obtain

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -Z_1 \frac{X + aY}{Z - Z_1} + a\left(Z_2 \frac{Y}{Z - Z_2}\right) \\ -Z_2 \frac{Y}{Z - Z_2} \end{pmatrix}. \tag{11}$$

By substituting $x' = x + ay$, $y' = y$ and $X' = X + aY$, $Y' = Y$, $Z' = Z$, we get the simple model of (5) for the projection of (X', Y', Z') to (x', y'). In other words, we get a skewed image of a skewed scene. Such a projection is demonstrated in Fig. 2c (note that the skew is three-dimensional).

2.3.5 X-Slits and Other Projection Models

Both perspective projection and linear pushbroom projection [6] are special cases of the X-Slits projection. Perspective projection is obtained when the two slits intersect and the intersection point is the optical center of the perspective projection (the projection of points on the plane containing the two slits should be defined accordingly). Linear Pushbroom projection is a special case of the POX-Slits projection, when the vertical slit resides on the plane at infinity. It was shown in [6] that a line in 3D is projected by this projection model to a hyperbola in the image. This is a special case of the result shown in Section 2.3.1.

3 Generating X-Slits Images by Nonstationary Mosaicing

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 along a horizontal line in 3D space in a roughly constant speed, and without changing its orientation or internal calibration. As we show below, in the simplest case, we can generate a new X-Slits image where the two slits of the underlying virtual X-Slits camera are defined as follows:

1. A horizontal slit that lies on the path of the optical center of the moving pinhole camera.
2. A vertical slit that is parallel to the image's vertical axis and whose location is determined by the parameters of the mosaicing process.

In practice, new view synthesis is performed by nonstationary mosaicing. Basic nonstationary mosaicing is defined as follows:

- From each frame t, sample the vertical column (strip) centered on the horizontal coordinate $s(t)$.
- Paste the strips into a mosaic image, as in [16].

In the general case, we may sample slanted strips rather than vertical columns (strips), and the orientation may also change as a function of t. In this case, the "vertical" slit of the underlying virtual camera may not be parallel to the image's vertical axis. However, for clarity of presentation and without loss of generality, we will continue calling one slit of the new virtual camera "horizontal" and the other slit "vertical." Typically, the "horizontal" slit is aligned with the path of the camera, while the second "vertical" slit is not constrained a priori and need not be orthogonal to the first slit.

The parameters of the strip sampling function $s(t)$ determine the location of the vertical slit of the virtual camera. A virtual walkthrough is obtained by generating a sequence of X-Slits images via nonstationary mosaicing, while moving the vertical slit along a planar path. Adjusting the image plane orientation is done by warping the mosaiced image, using (10).

In Section 3.1 and in the Appendix, we show how to sample vertical strips from the input images in the sequence in order to generate a valid X-Slits image. We also discuss the relation between the sampling function $s(t)$ and the parameters of the virtual X-Slits camera. In Section 3.2, we discuss implementation issues, including the treatment of deviation from constant speed and aspect-ratio normalization.

3.1 Nonstationary Strip Sampling

We start our analysis with the simplest case where the input image sequence is generated by a camera moving sideways in a direction parallel to the X-axis of the image. The camera is also assumed to be internally calibrated. In this simple case, the new synthesized image is a POX-Slits image (see Section 2.3) and the nonstationary strip sampling is a linear function. Below, we show the exact relation between the parameters of the linear sampling function and the parameters of the virtual POX-Slits camera. In the Appendix, it is shown that, even when the camera is not internally calibrated, any linear strip sampling function results in a X-Slits image (but not necessarily POX-Slits). When the motion of the camera is not parallel to the image plane, the sampling function is not linear anymore.

When the basic assumptions of the analysis are violated, namely, the camera changes its orientation and internal calibration arbitrarily along the input sequence, we need to preprocess the sequence. One solution involves registering all the images with each other using the homography of the plane at infinity. This computation requires, however, either (partial) internal camera calibration or some domain knowledge (such as parallel lines in the scene) [7].

3.1.1 Mosaicing by Linear Strip Sampling

Let our input be a sequence of images captured by a pinhole camera translating in constant speed along the X axis from left to right. We generate a new panoramic image by pasting columns from the input images, as illustrated in Fig. 3. 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. 4a, 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:

$$\mathbf{p} = (X, Y, Z) \implies p = (x, y) = \left(f\frac{X - X_t}{Z}, f\frac{Y}{Z} \right). \qquad (12)$$

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. 4a). 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. 4a). It now follows from (12) 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,

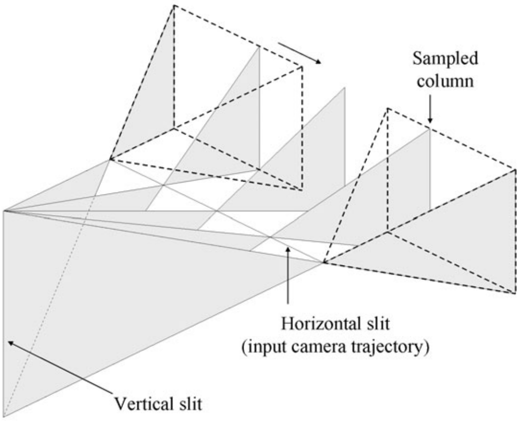


Fig. 3. The nonstationary column sampling routine that is used to synthesize new images.

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

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. 4a). The horizontal slit is at $Z = 0$ (all pinhole camera centers are at $Z = 0$). Equation (12) can therefore be rewritten as

$$\mathbf{p} = (X, Y, Z) \implies p = (x, y) = \left(f_x \frac{X}{Z + \Delta}, f_y \frac{Y}{Z} \right), \qquad (13)$$

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.

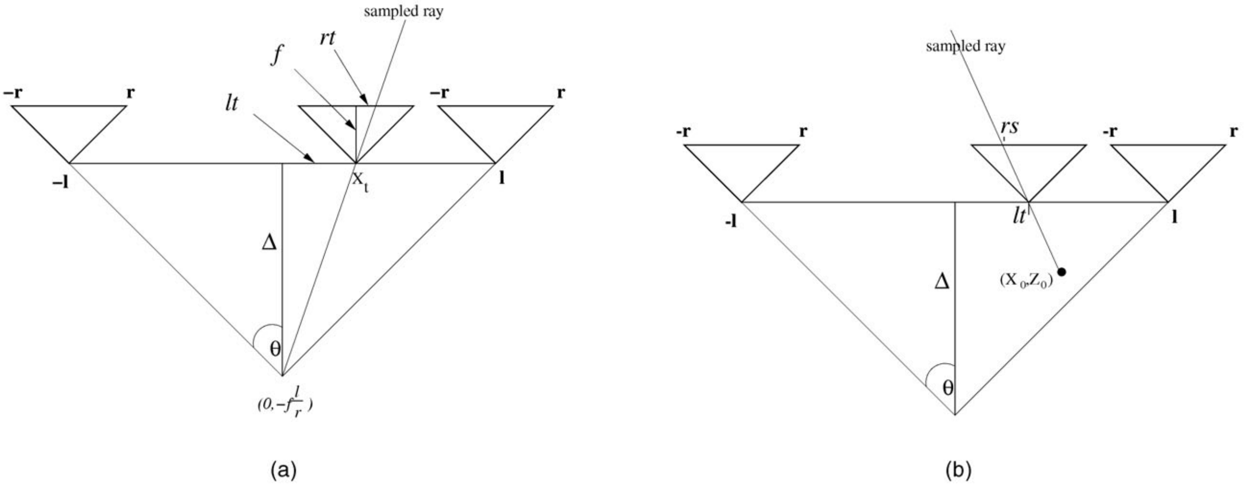


Fig. 4. New image formation with two possible positions of the vertical slit (see text).

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. 4b. (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 as above that such a choice of columns defines the mapping

$$(x, y) = \left(\left(f + \frac{\alpha l}{\tan\theta}\right)\frac{X - \beta l}{Z + \frac{\alpha l}{\tan\theta}}, f\frac{Y}{Z}\right). \qquad (14)$$

This can be written simply as

$$(x, y) = \left(f_x\frac{X - X_0}{Z + \Delta}, f_y\frac{Y}{Z}\right), \qquad (15)$$

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 that do not follow the perspective projection model. They do, however, follow the X-Slits projection model. To see this, we observe that all the rays, which participate in the generation of each new 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 (15), where $Z_0 = Z + \Delta$).

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 POX-Slits (compare (15) with (5)).

In the derivation leading to (15), we effectively showed that any linear sampling function yields a valid new POX-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.

3.2 Implementation Issues

In this section, we address the case when the motion of the camera deviates from constant speed (Section 3.2.1) and how the aspect ratio of the resulting mosaic is determined (Section 3.2.2). We also present an alternative implementation of mosaicing, namely, the slicing of space-time volume (Section 3.2.3).

3.2.1 Variable Camera Speed

When the camera moves in a linear trajectory but varying orientation and speed, we compensate for this variability by estimating the camera motion (see [7]) and by derotating the image planes. We found that, when the changes in camera orientation are small, a simple approximation is sufficient. Specifically, we compute the 2D rotation and translation between consecutive input frames using the method described in [3], and warp the images to cancel 2D rotation and vertical translation. The residual 2D translation is used as a rough approximation to the 3D velocity of the translating camera and determines the thickness of the vertical strip. This approach is similar to the pushbroom mosaicing technique described in [16].

3.2.2 Aspect-Ratio Normalization

The most apparent aspect of the distortion in X-Slits images is the variation of aspect-ratio, as analyzed in Section 2.3.2. To reduce this distortion, we vertically scale the new images. This normalization is essential for achieving compelling results.

Specifically, the distortion on the *image plane* of objects at depth Z given in (8) can be written as $\frac{Z}{Z+\Delta} \cdot \frac{f_y}{f_x}$ in the notation of (15). 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 scale the image vertically by the factor:

$$1 + \frac{\Delta}{Z_0}. \qquad (16)$$

3.2.3 The Space-Time Volume

In Section 3.1, we described how to synthesize an X-Slits image by sampling columns from the input images using the following linear sampling formula:

$$t = \alpha s + \beta, \qquad (17)$$

where t denotes the camera translation. Recall that α, β are free parameters which control the location of the vertical slit.

A useful representation for the visualization of this process is the Space-Time Volume (or the *epipolar volume*), which is constructed by stacking all input images into a single volume. In case of constant sideways camera motion, any vertical planar slice in the volume according to (17) is an X-Slits image. This process is illustrated in Fig. 5; it assumes that the input sequence has high frame-rate and negligible spatial aliasing, so that simple interpolation (such as bilinear or bicubic) of the volume is sufficient. Thus, rendering new X-Slits images is as simple as slicing a plane in the space-time volume.

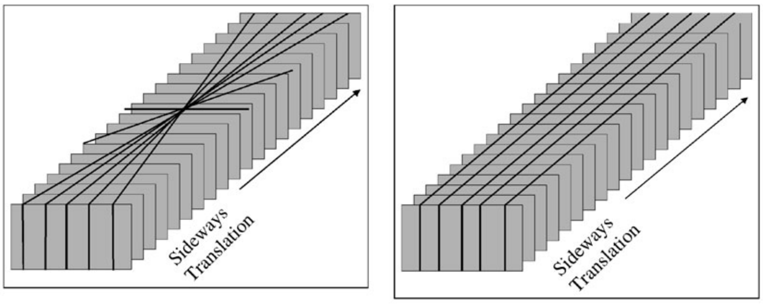
4 Extended X-Slits Models

The X-Slits projection presented so far was characterized by two properties: The set of projection rays was defined by two linear slits and the image surface was assumed to be a plane. This model can be generalized by modifying the form of the slits, or the shape of the image surface, or both. In this section, we introduce a few useful generalizations of the basic X-Slits model.

In Section 4.1, we discuss nonlinear slits and, specifically, the case of a circular-slit. Linear slits with nonplanar imaging surfaces are discussed in Section 4.2. This case is useful, for example, for synthesizing virtual panoramic views from a panoramic camera input.

4.1 Nonlinear Slits

One of the slits in the synthetic X-Slits images is the trajectory of the camera. When the camera's trajectory is not linear, an X-Slits view can be generated for which one of the slits is curved. An interesting family of such views has one circular slit and one linear slit. One way to generate such images is to use a camera rotating off-axis on a circle, as in concentric mosaics [20]. Concentric mosaics allow the generation of images in which the viewer can move continuously in a



(a) (b)

Fig. 5. A schematic description of images generated as slices in the space-time volume. (a) Changing the orientation of the slice moves the vertical slit inside and outside the scene. (b) The central slice gives a pushbroom image (the "traditional" mosaic). Sliding parallel slices in the space-time volume results in different viewing directions of oblique pushbroom images.

circular region. Each image generated from concentric mosaics is consistent with a circular X-Slits projection: One slit is the horizontal circular path of the camera center and the second slit is a vertical linear slit placed at the location of the viewer. To generate an image from a different viewing position, the vertical slit is placed in the new location.

While images generated from the concentric mosaics point outward, and the viewer location is inside the circle, it is also interesting to generate inward looking images from locations outside the circle. This can be realized by moving the camera in a circle around an object or by having a stationary camera viewing an object rotating on a turntable. Now, the location of the viewer in the synthesized images can be as far or as close to the object as we wish, inside or outside the circular slit.

The column sampling function which generates circular X-Slits images is not linear. It is easier to express it in angular terms, so we denote columns by their angle from the optical axis, and frames by the angular position of the camera. Assume that the images were taken by a pinhole camera rotating off-axis at radius R_0. In order to synthesize a X-Slits image with vertical slit at radius R_1, we take column β from the pinhole camera at γ and paste it as column α in the new image, as shown in Fig. 6. It can readily be seen that the following relation should hold: $\beta = \arcsin(\frac{R_1}{R_0}\sin\alpha)$ and $\gamma = \beta - \alpha$.

The distortions of such images are more complicated to analyze. However, in practice, since the field-of-view angles of cameras tend to be rather small, the circular slit in the relevant region is nearly linear and, therefore, the distortions are approximately the same as with linear slits. Using the same aspect-ratio normalization method as with linear slits, the results we achieve are quite convincing.

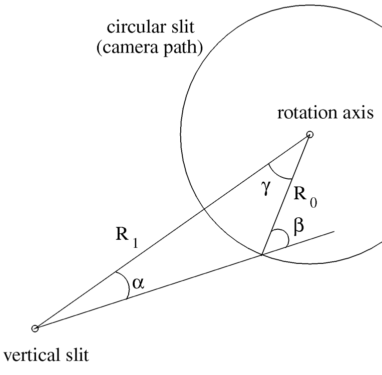


Fig. 6. Synthesizing new views using the circular slit model.

4.2 Panoramic Images

Synthesizing novel panoramic views turns out to be simple under the X-Slits model. In this case, a cylinder can be used as the image surface for both the input and output images, and the novel panoramic views can be generated in a method very similar to the usual X-Slits images.

We assume that the cylindrical input images are captured by a panoramic camera moving on a straight line. Without loss of generality, we assign the origin of the coordinate system to the first location of the camera and the point $(\, l \quad 0 \quad 0 \,)$ to the last location of the camera. One slit of the synthesized panoramic view is the path of the input camera, as illustrated in Fig. 7. The other slit can have an arbitrary location $(X_0 \;\; 0 \;\; Z_0) + \gamma(0 \;\; 1 \;\; 0)$.

From Fig. 7, one can see that the virtual image can be generated by pasting and scaling columns from the input images. Note that columns are indexed by horizontal angle. It follows that column α should be sampled from a corresponding column α from the image captured at time t, where $t = Z_0 \tan\alpha + X_0$, as illustrated in Fig. 7. Thus, new view synthesis can be done simply by stacking the input images in a space-time volume and slicing the volume by the function $t = Z_0 \tan\alpha + X_0$. After slicing, every column α should be scaled by $\frac{R}{\frac{Z_0}{\cos\alpha}+R}$, where R is the radius of the cylinder.

5 Experimental Results

In all our experiments, we used a camera moving in the horizontal plane. As discussed above, new view generation,

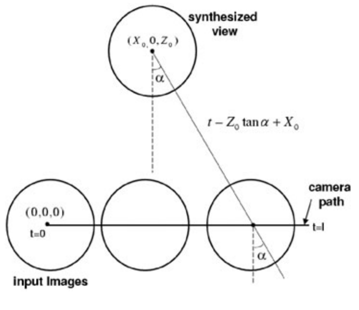


Fig. 7. Synthesizing new panoramic images using the X-Slits projection model.

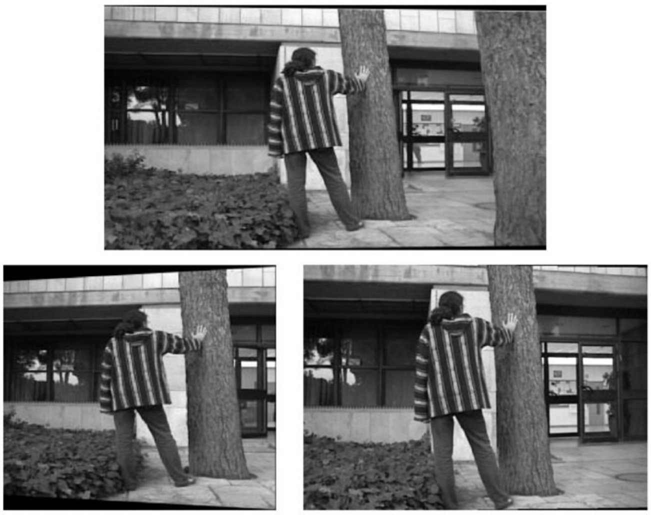


Fig. 8. This scene was filmed by a sideways moving camera, a total of 585 frames. We generated a movie that included both rotation about the person and forward motion. Three frames of this movie are shown above; note the changes in the window reflections, which appear vertical.

in this case, is done by sampling vertical strips from successive images and pasting them together into individual X-Slits images. The parameters of the strip sampling function determine the location of the vertical slit of the X-slit camera. In our experiments below, we manipulated the parameters of the sampling function so that the location of the vertical slit moves according to the desired ego-motion. A very compelling impression of camera motion is obtained, even though the horizontal slit of the X-slits camera, which is the trajectory of the input camera, remains fixed.

In addition, we also simulated a change of camera orientation using the equations derived in Section 2.3.3. Note that, we leave the slits as they are, changing only the orientation of the image plane. This is because the full rotation of the slits would change the set of visible projection rays and, therefore, cannot be performed by a 2D transformation of the image.

Next, we discuss two applications: the generation of a virtual walkthrough from a sequence of perspective images (Sections 5.1 and 5.2) and 3D object visualization (Section 5.3). The input video sequences used in our examples, as well as the synthesized walkthrough movies, are currently available on the Web at http://www.cs.huji.ac.il/~daphna/demos.html\#xslits.

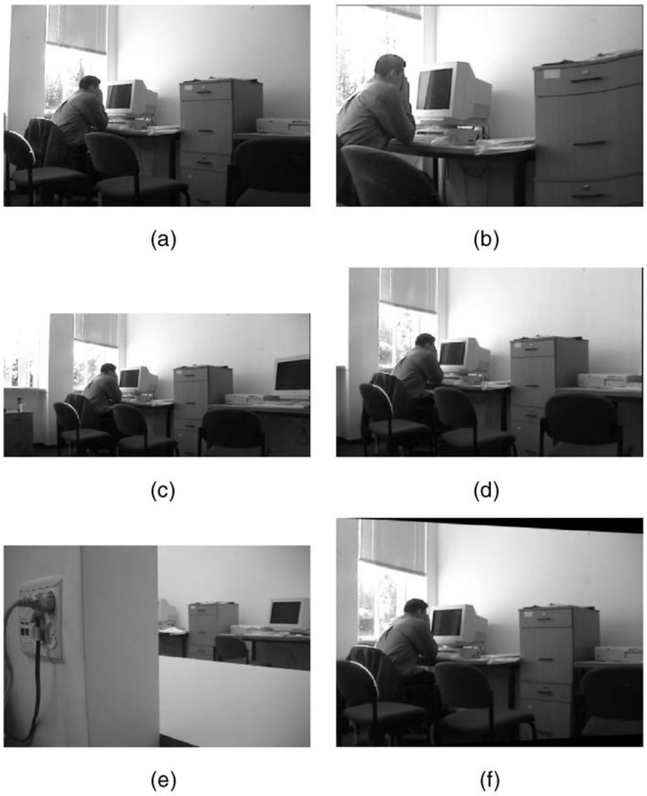


Fig. 9. This scene is located in a small room where moving backward to capture the whole room is impossible. The scene was filmed by a sideways moving camera, a total of 591 frames; one of the original frames is shown in (a). We show three new images: (b) one where the vertical slit is located in front of the original track and two (c) and (d) where the vertical slit is located behind the original track. 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 (e), and it demonstrates our ability to make images from impossible camera positions. Finally, (f) shows a simulated image where the camera was translated and rotated.

5.1 Virtual Walkthrough

In the first experiments (Figs. 8, 9, 10, and 11), we synthesized new sequences that correspond to a camera motion that has forward motion component, with visible parallax and lighting effects. In addition, the direction of the image plane was changed.

Another example used a sequence taken by a helicopter flying along a rocky coast in an unknown path and viewing direction (Fig. 12). Here, we synthesized a new sequence that corresponds to a forward moving camera. This sequence was more challenging since the input sequence

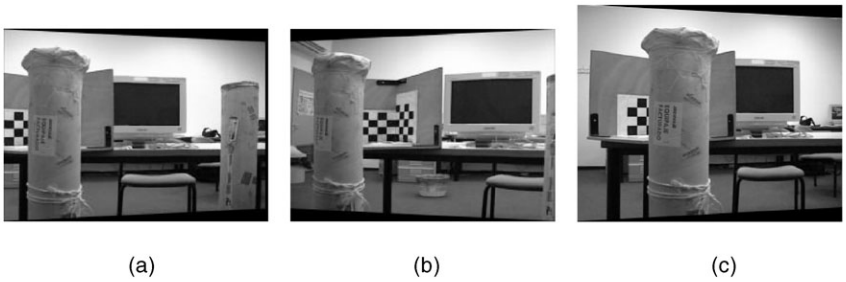
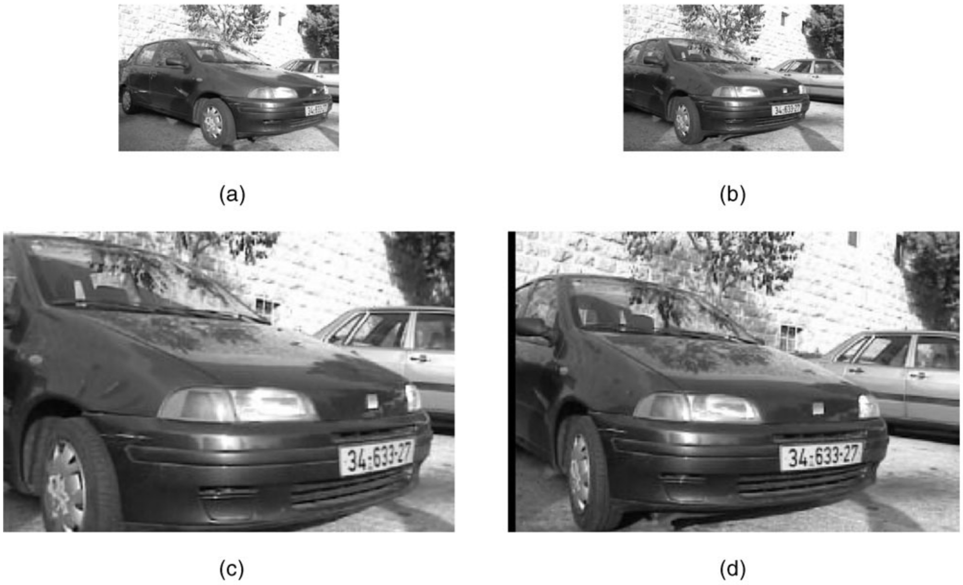


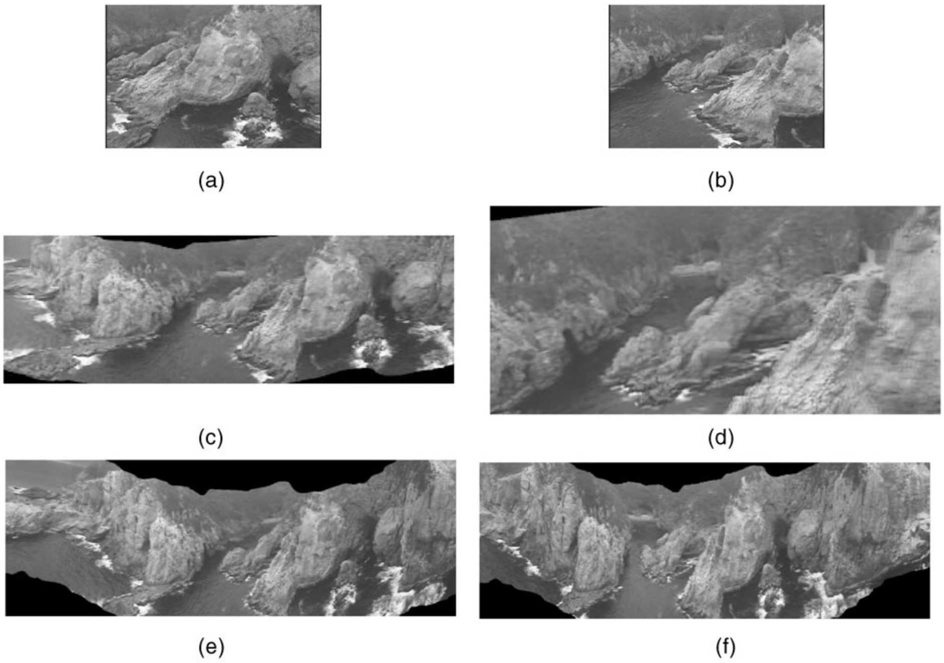
Fig. 10. This scene was filmed by a sideways moving camera in our lab, a total of 567 frames. (a) and (b) We generated an X-Slits movie where the virtual camera rotated about an object in the scene and (c) then translated ahead in a diagonal.



(a) (b)

(c) (d)

Fig. 11. Virtual walkthrough from a translating camera. (a) and (b) Two frames from the input sequence. (c) and (d) Two images rendered in forward motion. Note the apparently realistic changes in parallax and reflection.



(a) (b)

(c) (d)

(e) (f)

Fig. 12. Virtual walkthrough generated from a sequence taken by a freely flying helicopter. (a) and (b) Two frames from the input sequence. (c) and (d) Two images rendered in forward motion (diagonal slices). (e) and (f) Two images rendered in different viewing angles (parallel slices).

was taken in free motion with random disturbances (e.g., the effect of wind) and, thus, motion compensation was required (see discussion in Section 3.2.1).

5.2 New Views of Extended X-Slits Images

In this example, we show how to generate new views from a sequence of a rotating object, where the new sequence demonstrates forward motion with parallax (Fig. 13). The projection model of the new images correspond to nonlinear slits, as discussed in Section 4.

5.3 Object Visualization

Here, we demonstrate the use of the X-Slits projection for object visualization. Specifically, we show how an object can be "flattened," revealing several of its sides simultaneously. This is done by positioning the vertical slit behind the object (Fig. 14). Since the image is a valid X-Slits image that can be characterized and analyzed, we need not worry about such issues as duplicate images, which usually require handcrafted stitching.

6 Discussion

Related rendering methods to the one proposed here were discussed in the IBR and computer vision literature, e.g., [20], [22], [4]. There are two important differences between the method presented here and these earlier methods:

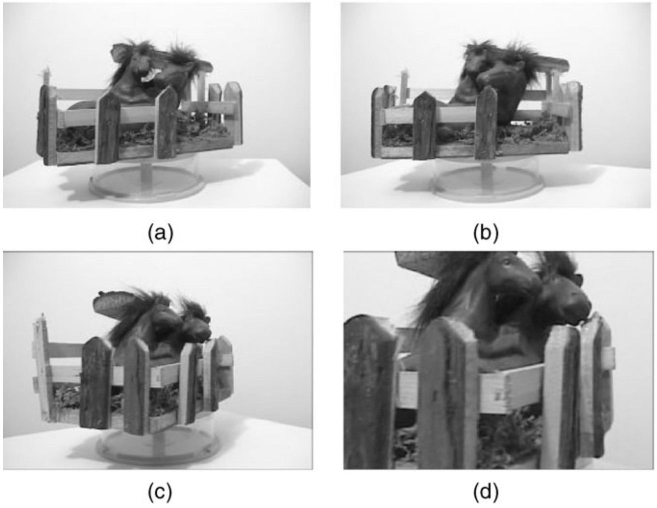


Fig. 13. A rotating object: (a) and (b) Two images from the original sequence of a rotating object. (c) and (d) Two synthesized images from a forward moving viewpoint.

1. In order to generate a virtual camera translation, our method does not need the calibration parameters of the input sequence. All we need to know is that the camera moved in constant velocity along a line in space; we don't need to know the camera's internal parameters. Instead of exact calibration, we can use qualitative guidelines to achieve a certain effect. For example, "rotate the orientation of the epipolar-volume slice counterclockwise to obtain a sense of forward motion." Or, "slide the slice forward to obtain the effect of sideways motion."
2. The theoretical analysis of the projection model is unique to our paper; it also allows us to do a few things that were not easy to do previously. One example is the generation of a synthetic movie. If we just put together a sequence of images generated along a path in space using one method or another, a lot of manual fiddling will be needed to achieve a sense of realism because each frame has a different distortion as compared to the pinhole ideal model. Our understanding of the distortion as a matter of depth-dependent scaling allows us to scale the images individually and automatically as described in Section 3.2.2; as a result, we get the compelling movies shown above with very little human interference.

Our method generates X-Slits images that are inherently different from pinhole images, as described above. The distortion is depth dependent and, therefore, (due to occlusions) there is no practical way to fix it even when given the exact 3D coordinates of the scene. If the distortion is not acceptable, the current alternatives are to use computer vision techniques or IBR techniques with 2D sampling, as described in Section 1.2. Our examples in Section 5.1 are intended to demonstrate that X-Slits images can appear realistic and compelling and, therefore, can be used effectively for visualization or even recognition when 3D reconstruction or full IBR are not possible or not practical. Finally, we note that our method scales easily to long input sequences due to its simple nature.

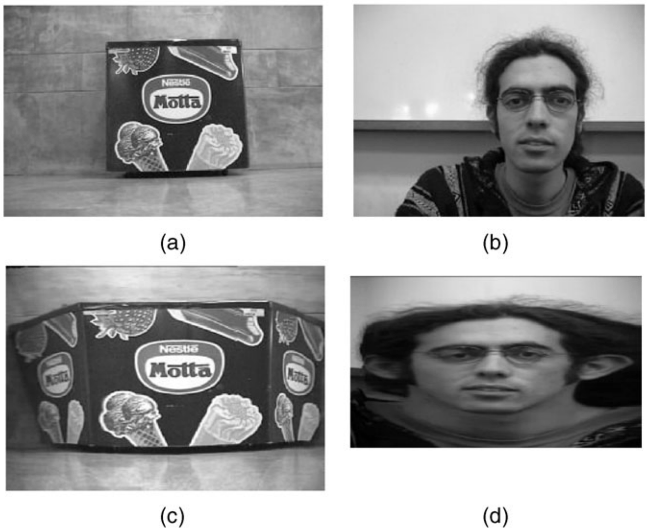


Fig. 14. Object representation. (a) and (b) The original input images. (c) and (d) Visualization with the vertical slit located behind the object. The object is seen as if "opened" inside out, giving a cubist effect: multiple sides are seen in a single picture.

7 Concluding Remarks

We presented a new nonperspective projection model, which is defined by two slits and a projection surface. This model can be physically realized and has been built in the late 19th century. Algebraically, we showed that this model corresponds to a second order transformation from three-dimensional space to two-dimensional space (while perspective projection is a linear, or first order, transformation).

The main application we pursued 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 since it is performed by nonstationary mosaicing or by slicing the space-time volume. 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.

Appendix A

Column Sampling for Linear Camera Trajectory

A.1 Linear Column Sampling: The Uncalibrated Case

Assume that the motion direction of the input camera is only parallel to the image plane, with the internal parameters of the camera being fixed but unknown. We now show that any linear sampling of the columns results in a valid X-Slits image.

Let J denote the inverse of the calibration matrix ([7, p. 141]) and assume that column $x = s(t)$ is sampled from the image captured at time t. The 3D physical location of this column on

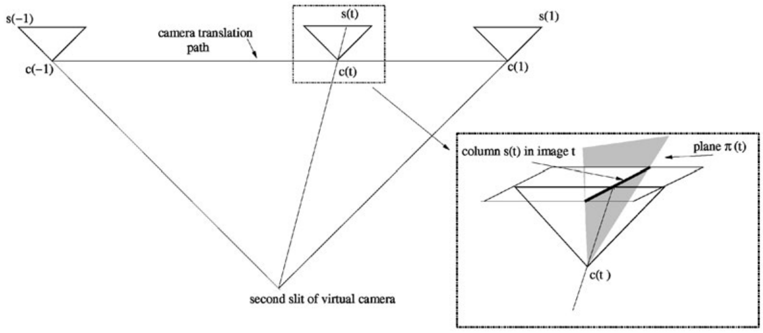


Fig. 15. Illustration of the relevant geometry, including the camera's path, the center of projection $\mathbf{c}(t)$, the column sampling function $s(t)$, and the plane $\pi(t)$.

the camera's projection plane, in standard coordinates denoted p, q, is the line defined by:

$$\begin{pmatrix} p \\ q \\ 1 \end{pmatrix} = J \begin{pmatrix} s(t) \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} J_{11}s(t) + J_{12}y + J_{13} \\ J_{22}y + J_{23} \\ 1 \end{pmatrix}.$$

(J is upper triangular.) Eliminating the free variable y, the line is parameterized by:

$$p = K_1 q + K_2 + K_3 s(t), \tag{18}$$

where

$$K_1 = \frac{J_{12}}{J_{22}}, K_2 = J_{13} - \frac{J_{12}}{J_{22}} J_{23}, K_3 = J_{11}.$$

Assume as before that the camera center at time t is $\mathbf{c}(t) = (lt, 0, 0)$. Assume also that columns are sampled linearly, i.e., the column sampled from the image taken at time t is defined by$(s(t), y)\, \forall y$, and $s(t) = \alpha t + \beta$ for some α, β. We will show that the result of pasting these columns together into a mosaic image is an X-Slits image.

First, from the definition, it follows that the image taken at time t contributes a set of rays that lie on a 3D plane. This plane is defined by the translating camera center $\mathbf{c}(t)$ and the line on the image plane which corresponds to column $s(t)$ (see Fig. 15, right inset). Denote this plane by $\pi(t)$. We first show that all the planes $\pi(t)$ intersect in a line; this line defines the vertical slit of our X-Slits camera. The horizontal slit is defined by the trajectory of the camera.

Specifically, the plane $\pi(t)$ is defined by the camera center $\mathbf{c}(t) = (lt, 0, 0)$ and two points on the line given in (18); we choose two such points by setting $q = 0$ and $q = 1$:

$$\begin{pmatrix} lt \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} K_2 + K_3(\alpha t + \beta) \\ 0 \\ 1 \end{pmatrix},$$
$$\begin{pmatrix} lt \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} K_1 + K_2 + K_3(\alpha t + \beta) \\ 1 \\ 1 \end{pmatrix}.$$

As can be readily verified, the plane incident with all three points is defined by

$$\begin{aligned} \pi(t) &= (-1, K_1, K_2 + K_3(\alpha t + \beta), lt) \\ &= (-1, K_1, K_2 + K_3\beta, 0) + t(0, 0, K_3\alpha, l). \end{aligned} \tag{19}$$

It follows from (19) that the family of planes $\pi(t)$ is a pencil of planes which intersect in a line and we define this line to be the vertical slit of the camera. The slit direction is determined by the cross product:

$$(-1, K_1, K_2 + K_3\beta) \times (0, 0, K_3\alpha) \propto (K_1, 1, 0).$$

From the requirement that the slit coincides with $\pi(t)$ from (19) for every t, we arrive at the following slit equation:

$$\begin{pmatrix} \frac{-l}{K_3\alpha}(K_2 + K_3\beta) \\ 0 \\ \frac{-l}{K_3\alpha} \end{pmatrix} + \lambda \begin{pmatrix} K_1 \\ 1 \\ 0 \end{pmatrix} \quad \forall \lambda \in \mathcal{R}.$$

Thus, by collecting vertical strips linearly and placing them in a mosaic, the resulting image is equivalent to an X-Slits image whose slits are parallel to the image plane. In case of zero skew, $J_{12} = 0$ and, so, $K_1 = 0$; now the two slits are orthogonal as in the POX-Slits camera discussed in Section 2.3.

A.2 Input Camera Moving on a General Straight Line

We shall now analyze the sampling function when the motion of the input camera is not parallel to the image plane. Let the motion direction be an arbitrary line in 3D. Let $\mathbf{c}(t) = \mathbf{c}_0 + \Delta\mathbf{c} \cdot t$ denote the camera position at time t (in $\mathcal{R}^3$) and let the pinhole camera's projection matrix be $[\mathbf{A}; -\mathbf{A}\mathbf{c}(t)]$, where $\mathbf{A} = [A_1, A_2, A_3]^T$ denotes the 3×3 matrix determined by the camera's internal calibration and its orientation. It follows that, when the camera is located at $\mathbf{c}(t)$, a scene point $\mathbf{p} \in \mathcal{R}^3$ is projected to $(\frac{x}{w}, \frac{y}{w})$ such that

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} A_1^T(\mathbf{p} - \mathbf{c}(t)) \\ A_2^T(\mathbf{p} - \mathbf{c}(t)) \\ A_3^T(\mathbf{p} - \mathbf{c}(t)) \end{pmatrix}. \tag{20}$$

Suppose we sample strips from these images, at varying positions and varying orientations. Let $s(t)$ and $\alpha(t)$ denote the sampling functions, giving the position of the strip on the image x-axis and the strip's orientation, respectively, so that we sample, in image t, the points on the oriented line $\frac{x}{w} = s(t) + \alpha(t)\frac{y}{w}$. Using (20), we obtain

$$\begin{aligned} &x - \alpha(t)y - s(t)w = \\ &A_1^T(\mathbf{p} - \mathbf{c}(t)) - \alpha(t)A_2^T(\mathbf{p} - \mathbf{c}(t)) - s(t)A_3^T(\mathbf{p} - \mathbf{c}(t)) = 0. \end{aligned}$$

This defines a plane (on which $\mathbf{p}$ lies)—the plane coincident with the sampled oriented line in image t and the center of projection of the camera $\mathbf{c}(t)$. We denote this plane by $\pi(t)$ (as in Fig. 15). This plane in homogeneous coordinates is given by

$$\pi(t) = \begin{pmatrix} A_1 - \alpha(t)A_2 - s(t)A_3 \\ -(A_1^T - \alpha(t)A_2^T - s(t)A_3^T) \cdot \mathbf{c}(t) \end{pmatrix}. \qquad (21)$$

For our sample to give an X-Slits image, it is necessary and sufficient that all these planes $\forall t$ intersect in a line. There are three cases to note in (21):

1. If $s(t) = constant$ and $\alpha(t) = constant$, then $\pi(t)$ is linear in t and, therefore, describes a line. When $\alpha(t) = 0$, we get the linear (oblique) pushbroom camera model [6], [17], see Fig. 5.
2. If $A_3^T \Delta\mathbf{c} = 0, s(t)$ is linear in t and $\alpha(t) = constant$, then once again $\pi(t)$ is linear in t. This is the case when the camera motion is parallel to the projection plane—the POX-Slits projection discussed in Section 2.3 (when $\alpha(t) \neq 0$, we get the tilted slit variant discussed in Section 2.3.4).
3. In the general case, if $\alpha(t)$ and $s(t)$ are of the form

$$\alpha(t) = -\frac{a_2 + b_2 t}{a_1 + b_1 t}, \quad s(t) = -\frac{a_3 + b_3 t}{a_1 + b_1 t}. \qquad (22)$$

Substituting these expressions into (21) gives

$$\pi(t) \propto \pi(t) \cdot (a_1 + b_1 t) = \begin{pmatrix} (a_1 + b_1 t)A_1 + (a_2 + b_2 t)A_2 + (a_3 + b_3 t)A_3 \\ -((a_1 + b_1 t)A_1 + (a_2 + b_2 t)A_2 + (a_3 + b_3 t)A_3)^T \\ (\mathbf{c}_0 + \Delta\mathbf{c} \cdot t) \end{pmatrix}$$

which is linear in t if and only if

$$(b_1 A_1 + b_2 A_2 + b_3 A_3)^T \Delta\mathbf{c} = 0. \qquad (23)$$

It is easy to see that the first two cases are special cases of (22).

There are four degrees of freedom (up to scale) in choosing a pair $\langle \alpha(t), s(t) \rangle$ of sampling functions which satisfy constraint (23). Each pair of sampling functions of the form (22) for which (23) holds implies that all $\pi(t)$ intersect in a single line (the virtual slit). And vice versa, it can also be shown that every family of planes $\pi(t)$ which intersect in a single line uniquely defines such a pair of sampling functions.

We can therefore conclude the following: Using the sampling method described above with sampling functions as in (22) and given a linear camera path, we can produce the image of any X-Slits camera with one slit overlapping the camera path.

Acknowledgments

The authors would like to thank Moshe Ben-Ezra for reinventing the X-Slits camera [2] (being unaware of Ducos du Hauron's work), Yoav Schechner for directing us to the relevant optics literature, Scott Kirkpatrick for directing us to the relevant events in the history of photography, and Tomas Pajdla for many useful comments. This research was supported (in part) by the EU under the Presence Initiative through contract IST-2001-39184 BENOGO and by a grant from the Israel Science Foundation.

References

[1] D. Aliaga and I. Carlbom, "Plenoptic Stitching: A Scalable Method for Reconstructing Interactive Walkthroughs," *Proc. ACM SIGGRAPH '01*, pp. 443-450, 2001.

[2] M. Ben-Ezra and S. Peleg, "Multi-View Omni-Directional Imaging," Technical Report-00-21, Hebrew Univ., Dec. 2000.

[3] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *Proc. European Conf. Computer Vision*, pp. 237-252, 1992.

[4] C. Buehler, M. Bosse, S. Gortler, M. Cohen, and L. McMillan, "Unstructured Lumigraph Rendering," *Proc. ACM SIGGRAPH '01*, pp. 425-432, 2001.

[5] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The Lumigraph," *Proc. ACM SIGGRAPH '96*, pp. 43-54, 1996.

[6] R. Gupta and R.I. Hartley, "Linear Pushbroom Cameras," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 963-975, Sept. 1997.

[7] R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge Univ. Press, 2000.

[8] H. Ishiguro, M. Yamamoto, and S. Tsuji, "Omni-Directional Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 257-262, 1992.

[9] S. Kang, "A Survey of Image-Based Rendering Techniques," *Videometric VI*, vol. 3641, pp. 2-16, Jan. 1999.

[10] R. Kingslake, *Optics in Photography.* SPIE Optical Eng., Press, 1992.

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

[12] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proc. ACM SIGGRAPH '96*, pp. 31-42, 1996.

[13] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System," *Proc. ACM SIGGRAPH '95*, pp. 39-46, Aug. 1995.

[14] B. Newhall, *The History of Photography, from 1839 to the Present Day.* The Museum of Modern Art, p. 162, 1964.

[15] T. Pajdla, "Stereo with Oblique Cameras," *Int'l J. Computer Vision*, vol. 47, nos. 1/2/3, pp. 161-170, 2002.

[16] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet, "Mosaicing on Adaptive Manifolds," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1144-1154, Oct. 2000.

[17] S. Peleg, M. Ben-Ezra, and Y. Pritch, "Omnistereo: Panoramic Stereo Imaging," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 279-290, Mar. 2001.

[18] S.M. Seitz, "The Space of All Stereo Images," *Proc. Int'l Conf. Computer Vision '01*, vol. I, pp. 26-33, July 2001.

[19] J. Semple and G. Kneebone, *Algebraic Projective Geometry.* Oxford: Clarendon Press, 1998.

[20] H. Shum and L. He, "Rendering with Concentric Mosaics," *Proc. ACM SIGGRAPH '99*, pp. 299-306, Aug. 1999.

[21] P.J. Sloan, M.F. Cohen, and S.J. Gortler, "Time Critical Lumigraph Rendering," *Proc. Symp. Interactive 3D Graphics*, pp. 17-24, Apr. 1997.

[22] T. Takahashi, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Arbitrary View Position and Direction Rendering for Large-Scale Scenes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 296-303, June 2000.

[23] D. Weinshall, M. Lee, T. Brodsky, M. Trajkovic, and D. Feldman, "New View Generation with a Bi-Centric Camera," *Proc. Seventh European Conf. Computer Vision*, pp. 614-628, May 2002.

[24] D. Wood, A. Finkelstein, J. Hughes, C. Thayer, and D. Salesin, "Multiperspective Panoramas for Cel Animation," *Proc. ACM SIGGRAPH '97*, pp. 243-250, Aug. 1997.

[25] J.Y. Zheng and S. Tsuji, "Panoramic Representation for Route Recognition by a Mobile Robot," *Int'l J. Computer Vision*, vol. 9, no. 1, pp. 55-76, Oct. 1992.

Assaf Zomet received the BA and MSc degrees in computer science from the Hebrew University of Jerusalem in 1997 and 1999, repectively. Currently, He is a PhD student in computer science at the Hebrew university of Jerusalem. His main research interests are motion analysis, geometrical aspects of computer vision, and low-level image processing. For the last several years, he has been working as a consultant/engineer for several computer vision companies on diverse topics.

Doron Feldman received the BA degree in math and natural science from the Open University of Israel in 1995. Currently, he is a PhD student in computer science at the Hebrew University of Jerusalem. His main research interests are geometry and motion in computer vision.

Shmuel Peleg received the BSc degree in mathematics from The Hebrew University of Jerusalem, Israel, in 1976 and the MSc and PhD degrees in computer science from the University of Maryland, College Park, in 1978 and 1979, respectively. He has been a faculty member at the Hebrew University of Jerusalem since 1980 and has held visiting positions at the University of Maryland, New York University, and the Sarnoff Corporation. He is a member of the IEEE.

Daphna Weinshall received the BSc degree in mathematics and computer science from Tel-Aviv University, Tel-Aviv, Israel, in 1982. She received the MSc and PhD degrees in mathematics and statistics from Tel-Aviv University in 1985 and 1986, respectively, working on models of evolution and population genetics. Between 1987 and 1992, she visited the center for biological information processing at MIT and the IBM T.J. Watson Research Center. In 1993, she joined the Institute of Computer Science at the Hebrew University of Jerusalem, where she is now an associate professor. Her research interests include computer and biological vision, as well as machine and human learning. She has published papers on learning in machine and human vision, qualitative vision, visual psychophysics, Bayesian vision, invariants, multipoint and multiframe geometry, image and model point-based metrics, motion and structure from motion. Selected publications can be obtained from http://www.cs.huji.ac.il/~daphna. She is member of the IEEE Computer Society.