Mosaicing with Parallax using Time Warping *

Alex Rav-Acha Yael Shor[†] Shmuel Peleg
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
E-Mail: {alexis,yaelshor,peleg}@cs.huji.ac.il

Abstract

2D image alignment methods are applied successfully for mosaicing aerial images, but fail when the camera moves in a 3D scene. Such methods can not handle 3D parallax, resulting in distorted mosaicing. General ego-motion methods are slow, and do not have the robustness of 2D alignment methods.

We propose to use the *x*-*y*-*t* space-time volume as a tool for depth invariant mosaicing. When the camera moves on a straight line in the x direction, a *y*-*t* slice of the space-time volume is a panoramic mosaic, while a *x*-*t* slice is an EPI plane. *Time warping*, which is a resampling of the t axis, is used to form straight feature lines in the EPI planes. This process will simultaneously give best panoramic mosaic in the *y*-*t* slices.

This approach is as robust as 2D alignment methods, while giving depth invariant motion ("ego motion"). Extensions for two dimensional camera motion on a plane are also described, with applications for 2D mosaicing, and for image based rendering such as "light field".

1 Introduction

Mosaicing from translating cameras can be used for many purposes, such as panoramic views from driving cars in and out of the city, inspection of long structures with a moving camera, etc. However, mosaicing is commonly used only for aerial photography, and is rarely used when the camera is translating in a 3D scene. In aerial photography the camera is far from the scene, which could therefore be considered flat. In most other applications the camera moves inside a 3D scene, and 2D image alignment is inadequate, resulting in substantial distortions due to depth parallax.

A common solution for the difficulty to compute motion with depth variations is to mechanically control the camera motion, or to measure it with external devices such as a GPS [9, 12]. This solution simplifies the mosaicing process, but restricts the use of mosaicing to fewer applications. A system based on image analysis without external devices will have by far more applications and at a reduced cost.

We propose to use the space-time *x*-*y*-*t* volume for both motion computation and for mosaicing. A mosaic is a *y*-*t* slice in this volume, and the EPI plane is a *x*-*t* slice. When a camera translates in a constant velocity, image features are located on straight lines in the EPI planes. When the camera motion varies between frames, the EPI lines are not straight any longer. Time warping that straightens the EPI lines gives a depth invariant motion computation.

Compared to other more general approaches for computing camera motion [3, 5, 7], the main strength of our approach is its simplicity and its elegant geometrical intuition. The space-time approach is also extended to handle camera rotations, and for cameras moving in a plane with two degrees of freedom. Setups describing camera motions which are applicable to this work are shown in Fig. 1. Two dimensional camera motion can also be used for image based rendering [10, 4].

* This research was supported (in part) by the EU under the Presence Initiative through contract IST-2001-39184 BENOGO, and by the Israel Science Foundation grant No. 191/02.

[†] Currently with Humaneyes Technologies Ltd.

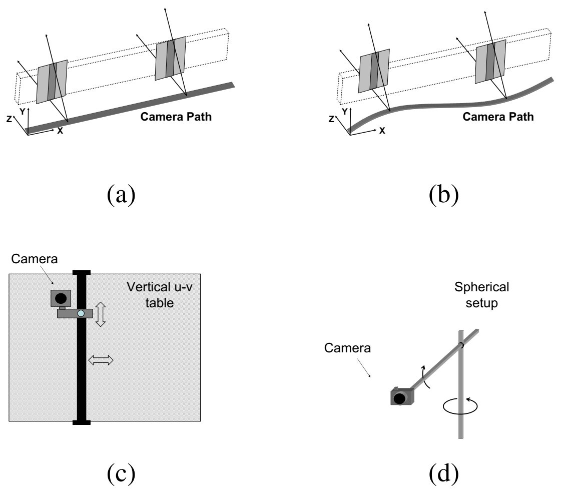


Figure 1: Example for cases yielding 1D and 2D motions.
(a) 1D motion - The camera moves along a straight line.
(b) 1.5D motion - Like (a), with vertical jumps. This kind of motion is common for sequences taken by a translating car.
(c) 2D motion - Traditional light field capturing device. The camera can move to arbitrary locations along the u-v table.
(d) 2D motion - Camera moves on a surface of a sphere. When tilting is forbidden, the camera moves on a cylindrical surface and the motion is 1D.

1.1 EPI Analysis

The space-time volume (or the *epipolar volume*) is constructed by stacking all input images into a single volume. It was first introduced by Bolles et. al. [2] who found out that when a camera moves in a constant velocity, *x*-*t* slices of the *x*-*y*-*t* volume (EPI) consist of straight lines. They used this observation to extract 3D information for the EPI images. Later papers use the same representation, but present more robust depth estimations and better handling of occlusions [11, 8]. In [17], the orientations of lines in the EPI plane are used for many applications: mosaicing, modeling, and visual navigation. A recent application of slicing the EPI volume to simulate a virtual walk-through was presented in [18].

The EPI representation is most natural when the camera translation is parallel to the image plane, and is perpendicular to the viewing direction. The EPI representation can also be used when the viewing direction is not perpendicular to the direction of translation after image rectification [6]. A possible rectification for the case of forward camera motion was described in [13].

1.2 Time Warping: From Time To Location

When the camera's velocity and the sampling rate are constant, the time of frame capture is proportional to the location of the camera along the camera path. In this case, the image features are arranged in the EPI plane (an *x*-*t* slice of the *x*-*y*-*t* volume) along straight lines, since the projections of each 3D point are only along a straight line in this plane. Each line represents a different image feature corresponding to a point in the 3D world, and the orientation of this line is inversely proportional to the depth of that point. Points at infinity, for example, will create straight lines parallel to the t axis, since their projection into the image is constant, and does not change with camera translation. Closer points move faster in the image, and the line representing them will have a small angle with the x axis.

When the velocity or the sampling rate of the camera varies, the time of frame capture is no longer proportional to the camera location. In this case the image features are no longer arranged on straight lines in the EPI plane.

The lines in the EPI plane can be straightened by "Time Warping". In time warping the image location along the time axis is replaced with the camera's location along the x axis. When the camera locations along the x axis are unknown, any time warping that will make the EPI lines straight must have time spacing which is proportional to the camera locations along the x axis.

2 Geometrical Analysis

Given a set of images whose optical centers reside on a plane, all images can be represented as a set of rays [10, 4]. Each ray can be determined by its intersection with two planes. Commonly, one plane is the camera plane *u*-*v*, and the second plane is the image plane *s*-*t*.

A perspective image is a set of rays which intersect the camera plane in a single point. When the optical axis of the camera is perpendicular to the camera plane (i.e - a frontal view), the camera plane is expressed by the 3D coordinates: $(u, v, 0)$, and the image plane is expressed by the 3D coordinates: (s, t, f). Each image can be considered as a sampling of a 2D slice in the continuous 4D function $L(u, v, s, t)$.

In our notation each image has its own *x-y* coordinate system, and the 4D volume is represented by *x-y-u-v*. Using the image coordinate system is a natural generalization of the space-time volume *x-y-t*. Also, in local image coordinates the slopes of the epipolar lines are inversely proportional to the depth. This is equivalent to the light-field notations with the *s-t* plane at infinity.

2.1 Geometrical Analysis of the Light Field Space

Let I_n be the n^{th} frame, and let (u_n, v_n) be its optical center. The 3D point $P = (X, Y, Z)$ is projected to its image coordinates:

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X - u_n \\ Y - v_n \end{pmatrix} \quad (1)$$

Without loss of generality, $(u_0, v_0) = (0, 0)$ and thus:

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - \frac{f}{Z} \begin{pmatrix} u_n \\ v_n \end{pmatrix} \quad (2)$$

From Eq. 2 it can be shown that the projections of the point P onto these images reside on a plane in the 4D light field space. The parameterized representation of this plane is given by:

$$\begin{pmatrix} x \\ y \\ u \\ v \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ 0 \\ 0 \end{pmatrix} + u \begin{pmatrix} -\frac{f}{Z} \\ 0 \\ 1 \\ 0 \end{pmatrix} + v \begin{pmatrix} 0 \\ -\frac{f}{Z} \\ 0 \\ 1 \end{pmatrix} \quad (3)$$

The slope of this plane is identical in both the u and the v directions as both are determined by f/Z, and only a single slope parameter is needed to describe the plane.

Taking an *x-u* (*y-v*) slice of the 4D space will produce an EPI image which is constructed from the rows (columns) of the images taken under a translation along the u axis (v axis). This is a reduction to the traditional representation of the epipolar plane image.

We will describe the use of this 4D representation to recover the 2D motion of a camera moving in a plane. This is done by "shifting" the estimated u, v coordinates of each frame so that the image features will be arranged in the 4D volume *x-y-u-v* along 2D planes. When this goal is achieved after warping the u and v axes, the estimated u, v coordinates of the frames become proportional to the 2D locations of the optical centers of the camera. In the case of 1D motion this is called "Time Warping".

2.2 Spherical Camera Motion

When a sequence is taken by a hand held camera or using a tripod, the camera's translation can be better estimated by a spherical surface.

In this case, the camera translates and rotates simultaneously. Since the rotation is proportional to the translation in all directions, we can achieve an adequate model by treating the motion identically to the case of a camera translating on a plane, with the two unknowns u and v. In this case, however, the slopes of the EPI feature lines (or planes) are a combination of the translation component (which is depth dependent) and the rotational component (which is depth independent). In the extreme case of a purely rotating camera ($R = 0$), the slopes of the feature lines are depth independent, so there is no parallax.

When the camera can rotate about the z axis, e.g. when the camera is hand-held, a rotational component α can be added as a third unknown.

3 The Alignment Scheme

The alignment process uses both motion parameters and shape parameters. The motion parameters are the translation and rotation of the camera, which vary for each frame. The shape parameters are the slopes of the lines in the EPI plane for a camera translating along a straight line, or the slopes of the planes in the light field space for a camera translating in a plane. The slopes of the lines and the planes in the EPI domain are related to the depth of the corresponding 3D points, and thus they are constant for each scene point at all frames.

To compute the locations of the optical centers such that image features will reside on straight lines (or on planes) we used the following scheme, alternating between estimation of shape and estimation of motion:

1. Choose a frame I, and initial from it a set $S = \{I\}$. Assume that the shape parameters corresponding to this image are spatially uniform.

2. Compute motion parameters (translation components and optionally rotation components) by aligning a new image to the existing set S.

3. Add the registered frame to set S.

4. Estimate the shape parameters (the slopes of EPI lines or the slope of EPI planes) for S.

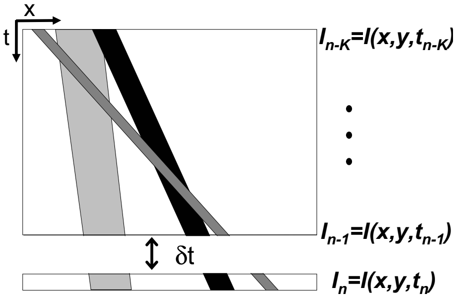

x
t
$I_{n\text{-}K}=I(x,y,t_{n\text{-}K})$
$I_{n\text{-}1}=I(x,y,t_{n\text{-}1})$
δt
$I_n=I(x,y,t_n)$

Figure 2: Time warping: When the scene has depth variations there is only a single placement t_n of the new frame I_n for which the EPI lines remain straight. This displacement is independent in the slopes of the lines, and thus it is depth invariant.

5. Return to 2. Repeat until reaching the last frame of the sequence.

Fig. 2 demonstrate this scheme for the case of a camera translating along a straight line. The process begins by estimating the motion parameters between a pair of frames under the assumption of uniform depth. This is equivalent to the regular 2D parametric registration.

4 Adding a new Frame to the Panorama

4.1 Estimating the Shape Parameters

Estimating the shape parameters should only be done for a subset of image points, as they are used to compute only a few motion parameters. The process can be formulated in the following way: Let k be the index of the frame for which we estimate the shape and let $T_{n,k} = (u_n - u_k, v_n - v_k)^t$ be the translation of the optical center of the camera between the n^{th} and the k^{th} frames.

Following [7], The shape parameter $d = d(x, y, k)$ in the image point (x, y) minimizes the following error function:

$$Err(d) = \sum_{n \neq k} w_n^d \cdot \sum_{x,y \in W} (d \cdot \bigtriangledown I^t \cdot T_{n,k} + I_n - I_k)^2, \quad (4)$$

Where $\bigtriangledown I$ is the gradient of the image I_k in the point (x, y), and W is a small window around (x, y). (We used a 5x5 window). The minimum of this quadratic equation is obtained by:

$$d = -\frac{\sum_{n \neq k} w_n^d \cdot \sum_{x,y} \bigtriangledown I^t \cdot T_{n,k} \cdot (I_n(x,y) - I_k(x,y))}{\sum_{n \neq k} w_n^d \cdot \sum_{x,y} (\bigtriangledown I^t \cdot T)^2} \quad (5)$$

The weights w_n^d determine the influence of each frame on the shape estimation. Most of the weights are set to zero, except for frames which are close in time or in space (five closest frames in current implementation).

For each window in I_k, the computation described above is repeated iteratively until convergence, where in each iteration, the relevant regions in all the frames $\{I_n\}$ with $w_n^d \neq 0$ are warped back towards I_k according to $T_{n,k}$ and the current estimate of d.

As we do not need to estimate the shape parameters for every pixel, we reject points for which the shape computation may be wrong using the following criteria:

1. We do not use points with a small gradient in the direction of motion. The threshold is selected according to the desired number of points to use.

2. We do not use points for which the iterative shape computation algorithm fails to converge.

4.2 Depth Invariant Alignment

The alignment concept is demonstrated in Fig. 2. The motion parameters should align an input image with the line or plane formed by image features in the preceding frames. We use a slight modification of the Lucas-Kanade direct $2D$ alignment as described in [1].

Assume that all the images $I_0 \ldots I_{k-1}$ have already been aligned and let the k^{th} frame be the new frame being aligned. We also know of the shape parameters $d(x, y, n)$ for $n < k$. To compute the motion parameters of the new frame, we minimize the error function: (Sometimes the term I_t is used to denote the difference between images).

$$Err(p, q) = \sum_{n \neq k} w_n^a \cdot \sum_{x,y} (p \frac{\partial I_n}{\partial x} + q \frac{\partial I_n}{\partial y} + I_n - I_k)^2, \quad (6)$$

where the displacement p, q of each point is given by:

$$\begin{array}{c} p(x,y,n) = (u_n - u_k) \cdot d(x,y,n) \\ q(x,y,n) = (v_n - v_k) \cdot d(x,y,n). \end{array} \quad (7)$$

Note the use of the derivatives $\frac{\partial I_n}{\partial x}$ and $\frac{\partial I_n}{\partial y}$ which are estimated from I_n rather then from I_k, since we haven't computed $d(x,y,k)$ yet, and therefore we must align I_k to the rest of the images.

The coefficients w_n^a are also used to weight the importance of each frame in the alignment. For example, frames which are far off, or contain fewer information should probably receive smaller weights. For each image whose location u_n, v_n is unknown we set $w_n^a = 0$.

Currently we align a new frame using about 3 preceding frames. When the camera is translating on a plane we use several additional frames which are not neighbors in time but whose optical centers are close. In this way we reduce the drift in the motion computations.

4.2.1 Handling rotations

When the camera can also rotate, image displacements are a combination of the translational component, which is depth dependent, and the rotational component which is depth independent. Assuming small camera rotations and using the approximation $cos(\alpha) \approx 1$ and $sin(\alpha) \approx \alpha$ the following motion model is obtained:

$$\begin{aligned} p(x,y,n) &= (u_n - u_k) \cdot d(x,y,n) + a - \alpha \cdot y \\ q(x,y,n) &= (v_n - v_k) \cdot d(x,y,n) + b + \alpha \cdot x. \end{aligned} \tag{8}$$

a and b denote the small pan and tilt which induce an approximately uniform displacement in the image. α denotes small camera rotation about the z axis. For larger rotations, or when the focal length is small, full rectification can be used.

Using Eq. 8 with the error function in Eq. 6, and setting to zero the derivative with respect to the motion parameters (the camera shift u, v and the rotational components α, a, b), gives five linear equations with five unknowns.

If the camera is restricted to translate along a straight line (without loss of generality this line is horizontal), then $v_n = v_k = 0$, and we are left with fewer unknowns - one unknown for a purely translating camera, and four unknowns for a camera translation and rotating.

5 Mosaicing a Panorama

A panoramic image is a *t*-*y* slice in the epipolar volume ($1D$ camera motion) or a *u*-*v* slice in the $4D$ light field space ($2D$ camera motion). Since the samples of the epipolar volume along the time axis, or the samples of the light field space along the u, v axes, are very sparse, interpolation in used to create a continuous panorama.

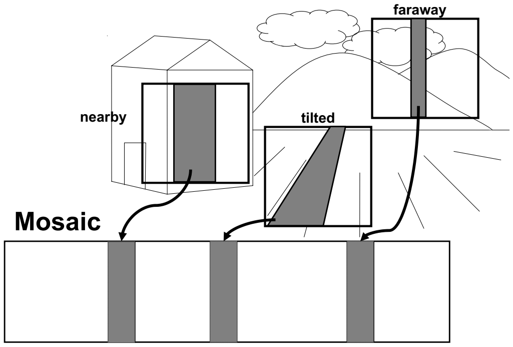


Figure 3: To produce a seamless mosaic, strips should be scaled according to scene depth. The width of the strip in the mosaic image is a function of the camera motion. Relative to image motion, image strips of nearby objects should be wider, while image strips of faraway objects should be narrower. Depth estimate can vary along the strip.

The basic concept of mosaicing with strips suggests that each pixel of the mosaic will be taken from the nearest frame in the sequence. When the camera is translating along a straight line, this implies the use of vertical strip from each image, whose width is proportional to the current motion [13]. In the more general case, the mosaic panorama is generated from a collection of patches [14]. These patches are determined by the Voronoi diagram, created from the set of all camera centers. An example of such a diagram is shown in Fig. 8(c).

Using the ego-motion of the camera, a possible mosaicing approach is to synthesize a parallel (or pushbroom) projection from the given input images [15, 16]. Practically, when input images are dense, this process can be approximated by the traditional mosaicing framework of warping and pasting strips with simple image scaling. This is demonstrated in Fig. 3 for the $1D$ case - the shape of each image strip is determined by the depth estimation at the strip. The image strips are warped to fit the mosaic strips determined by the ego-motion of the camera.

The scale is determined in a similar manner to the slope estimation described in Eqs. 4-5 with two exemptions:

- We do not estimate the slope at each point, but rather fit a single slope (or a polynomial one) for the entire

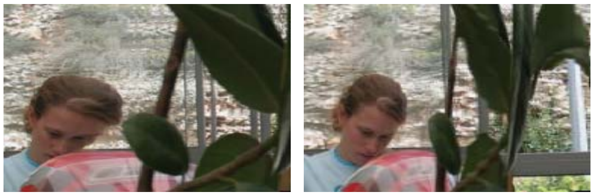
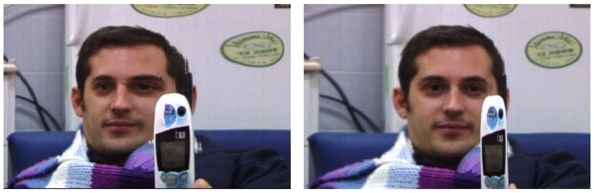


Figure 6: Mosaicing from a camera on a circular trajectory. 2D alignment shows (left) distortions around depth discontinuities (girl's head), which disappear with EPI analysis (right).



(a) (b)

Figure 7: Mosaicing from a translating camera. Most of the motion distortions caused when using 2D image alignment (a) disappear when using the proposed time warping (b).

slice.

- We use only the preceding and succeeding frames, as a continuous stitching is more important than an accurate slope.

When the image strips are small, no scaling was needed to produce continuous results (See for example Fig. 5).

6 Examples

The examples in Fig. 4 and in Fig. 5 were taken from a moving car, where in Fig. 5 the camera had substantial vibrations. The differences between the mosaic images obtained by $2D$ image alignment and the mosaic images obtained by time warping is evident.

The sequence shown in Fig. 6 was taken by a camera mounted on a panning tripod.

The sequence shown in Fig. 8 was taken by a camera mounted on a freely rotating tripod, as demonstrated in Fig.1(d).

7 Discussion

Distortions can be observed in the mosaic images created by our method, such as the thinning of the mobile phone in Fig. 7(b). These distortions are not due to inaccuracies in motion computation, but are mostly unavoidable in mosaicing from translating cameras, giving the pushbroom projection. In this case, close objects become thinner and far away objects become wider in the mosaic.

The main advantage of our approach relative to 2D image alignment is being depth invariant. As a result, the mosaics created using time warping are consistent for each distance, while in 2D alignment an object might change its geometry, as happened for the head in Fig. 7(a).

The depth invariance is most relevant when synthesising new viewpoints. Objects change viewing direction when our method is used, but shrink and expand when 2D alignment is used. Depth invariance is also essential with 2D camera motion. Frames which were taken far apart in time can be neighbors in the resulting mosaic. Therefore, only depth invariant methods will have a chance to succeed when the camera path covers areas of different depths.

8 Concluding Remarks

Most applications of mosaicing with image-based alignment are used only for purely rotating cameras or for aerial images, assuming that ego-motion computation is too complicated and unstable. In this work we expand the application areas of mosaicing to restricted, but very common, camera translations. In such cases the ego-motion computation can be represented in the space-time volume, and is shown to be simple and elegant. We further propose that our depth invariant alignment, along with the space-time representation, can also be used in view synthesis applications and generation of panoramic stereo images.

References

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

[2] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV*, 1(1):7–56, 1987.

[3] O. Faugeras. Three-dimensional computer vision : A geometric viewpoint. The MIT Press, Cambridge, 1993.

[4] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. *SIGGRAPH*, 30:43–54, 1996.

[5] K. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *MOTION91*, pages 156–162, 1991.

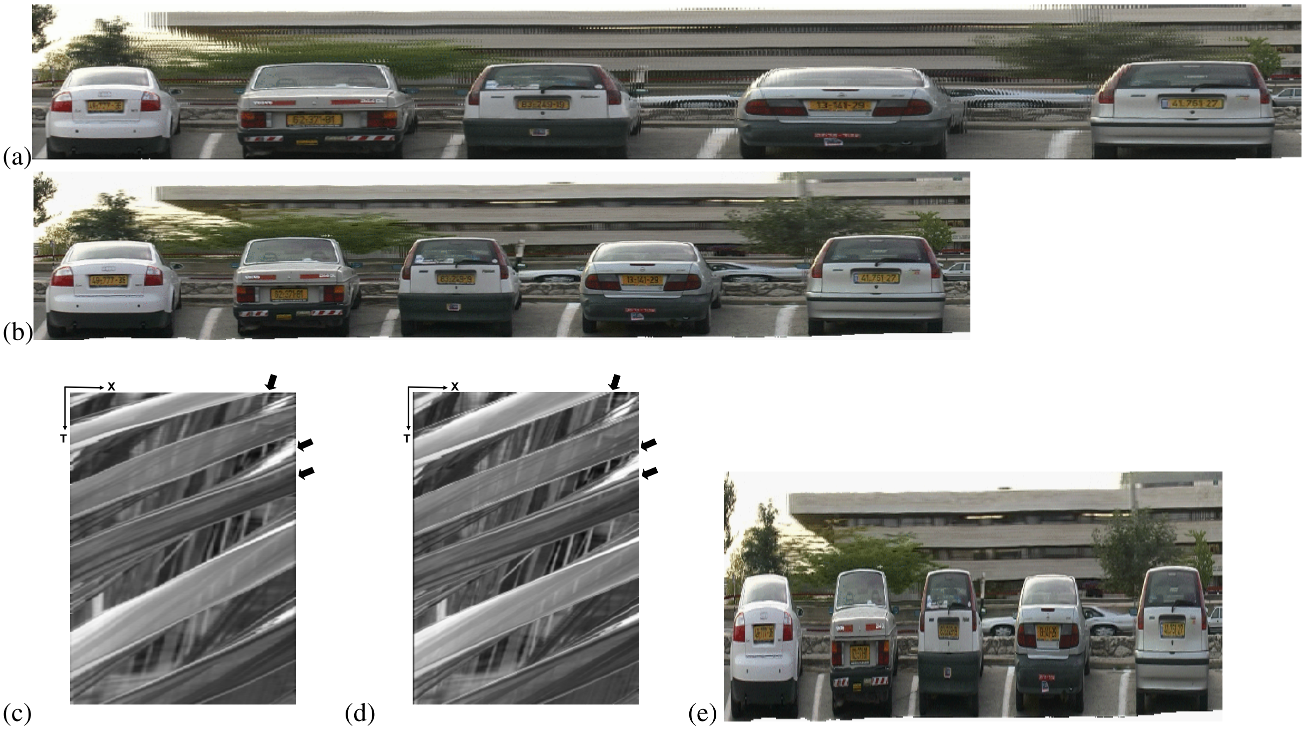


Figure 4: Mosaicing from a translating camera.
(a) Using 2D parametric image alignment. Distortions occur when image motion alternates between far and near objects.
(b) Using proposed EPI method, all cars are properly scaled.
(c)-(d) EPI planes for the mosaic in (a) and (b). Marked EPI line are straight in (d).
(e) Using different scales of the EPI mosaic, one can "focus" on different depth layers. Here, we focus on the far building. Usually, the scale is automatically determined so that the slope of the dominant depth in the EPI image will be one.

[6] R. Hartley. Theory and practice of projective rectification. *IJCV*, 35(2):1–16, November 1999.

[7] M. Irani, P. Anandan, and M. Cohen. Direct recovery of planar-parallax from multiple frames. *PAMI*, 24(11):1528–1534, November 2002.

[8] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR01*, pages I:103–110, 2001.

[9] H. Kawasaki, M. Murao, K. Ikeuchi, and M. Sakauchi. Enhanced navigation system with real images and real-time information. In *ITSWC2001*, October 2001.

[10] M. Levoy and P. Hanrahan. Light field rendering. *SIGGRAPH*, 30:31–42, 1996.

[11] M. Okutomi and T. Kanade. A multiple-baseline stereo. *PAMI*, 15(4):353–363, April 1993.

[12] S. Ono, H. Kawasaki, K. Hirahara, and K. I. Masataka Kagesawa. Ego-motion estimation for efficient city modeling by using epipolar plane range image. In *ITSWC2003*, November 2003.

[13] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *PAMI*, 22(10):1144–1154, October 2000.

[14] H. S. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. *ECCV*, 2:103–119, 1998.

[15] H. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *ICCV98*, pages 953–958, 1998.

[16] Z. Zhu, A. Hanson, H. Schultz, and E. Riseman. Parallel-perspective stereo mosaics. In *ICCV01*, pages I: 345–352, 2001.

[17] Z. Zhu, G. Xu, and X. Lin. Panoramic epi generation and analysis of video from a moving platform with vibration. In *CVPR99*, pages II: 531–537, 1999.

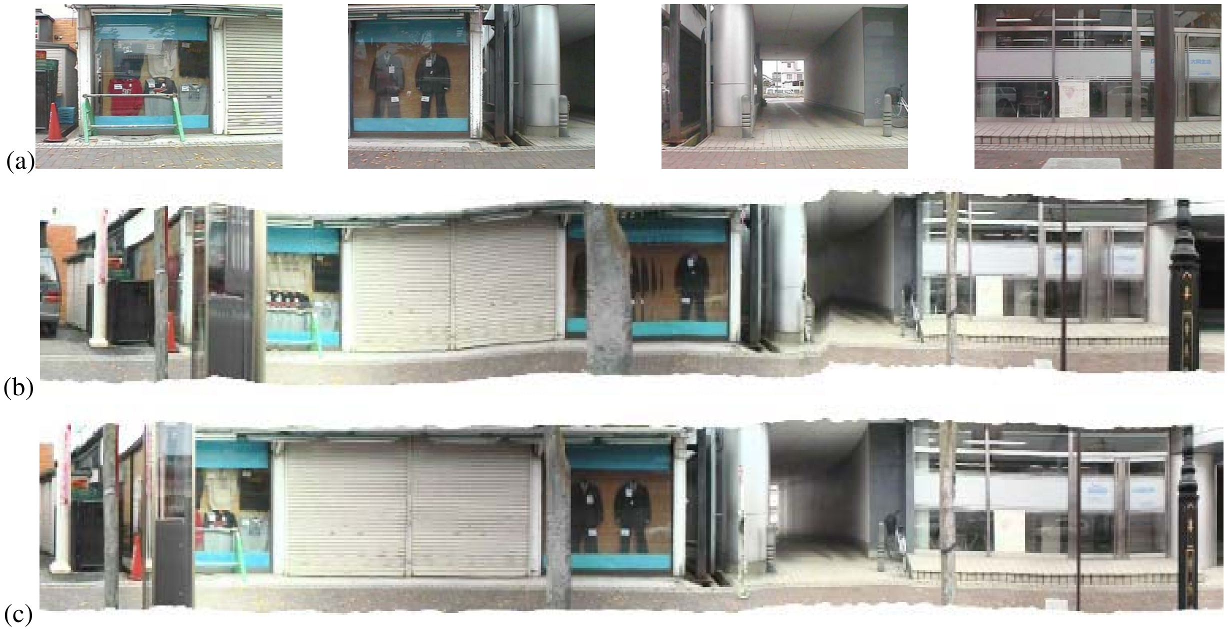


Figure 5: Mosaicing results for a sequence taken by a translating car with strong vibrations.
(a) Several images from the sequence.
(b) The resulting panoramic mosaic using $2D$ alignment. Distortions are clearly seen.
(c) The resulting panoramic mosaic using time warping.

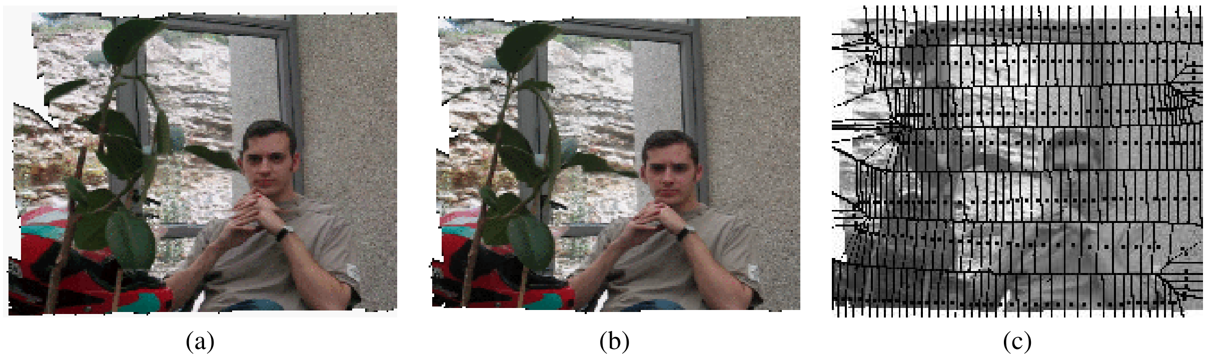


Figure 8: 2D mosaicing with a camera moving on the surface of a sphere as demonstrated in Fig.1.d.
(a) Using 2D image alignment. Visible distortions in foreground and background. (b) Using proposed EPI alignment. Distortions are substantially reduced. The respective Voronoi diagram and the camera centers are plotted in (c).

[18] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. *PAMI*, pages 741–754, June 2003.