Stereo Reconstruction from Multiperspective Panoramas

Yin Li, Heung-Yeung Shum, *Senior Member*, *IEEE*, Chi-Keung Tang, *Member*, *IEEE Computer Society*, and Richard Szeliski, *Senior Member*, *IEEE*

Abstract—A new approach to computing a panoramic (360 degrees) depth map is presented in this paper. Our approach uses a large collection of images taken by a camera whose motion has been constrained to planar concentric circles. We resample regular perspective images to produce a set of multiperspective panoramas and then compute depth maps directly from these resampled panoramas. Our panoramas sample uniformly in three dimensions: rotation angle, inverse radial distance, and vertical elevation. The use of multiperspective panoramas eliminates the limited overlap present in the original input images and, thus, problems as in conventional multibaseline stereo can be avoided. Our approach differs from stereo matching of single-perspective panoramic images taken from different locations, where the epipolar constraints are sine curves. For our multiperspective panoramas, the epipolar geometry, to the first order approximation, consists of horizontal lines. Therefore, any traditional stereo algorithm can be applied to multiperspective panoramas with little modification. In this paper, we describe two reconstruction algorithms. The first is a cylinder sweep algorithm that uses a small number of resampled multiperspective panoramas to obtain dense 3D reconstruction. The second algorithm, in contrast, uses a large number of multiperspective panoramas and takes advantage of the approximate horizontal epipolar geometry inherent in multiperspective panoramas. It comprises a novel and efficient 1D multibaseline matching technique, followed by tensor voting to extract the depth surface. Experiments show that our algorithms are capable of producing comparable high quality depth maps which can be used for applications such as view interpolation.

Index Terms—Multiperspective panorama, epipolar geometry, stereo, correspondence, tensor voting, plane sweep stereo, multibaseline stereo.

1 Introduction

Traditional stereo reconstruction begins with two calibrated perspective images taken with pinhole cameras. To reconstruct the 3D position of a point in the first image, its corresponding point in the second image has to be found before applying triangulation. Perspective cameras have the property that corresponding points lie on straight lines, which are called epipolar lines. In order to simplify the search for correspondences, the two images can optionally be rectified so that epipolar lines become horizontal.

In this paper, we are interested in computing a dense depth map with a large field of view (e.g., 360 degrees) for applications such as large environment navigation. In traditional stereo, the field of view of the reconstructed depth map is usually small. Combining intermediate depth maps obtained from overlapping stereo pairs may be a plausible solution to widen the field of view [13]. Unfortunately, accumulation error can quickly add up. An alternative approach is to apply stereo algorithms to panoramic images, bypassing the need for merging intermediate representations. In [11], a multibaseline stereo algorithm is proposed that employs omni-directional panoramic images. The epipolar constraints, however, are no longer straight lines [18].

Recently, multiperspective panoramas [31] have been proposed to reconstruct large environments. Unlike conventional images, multiperspective panoramas capture parallax effects as each column of pixels is taken from a different perspective point. It has been shown in [31] that the imaging geometry of multiperspective panoramas can be greatly simplified for depth reconstruction since the epipolar geometry, to the first order, consists of horizontal lines. On the other hand, multibaseline stereo using several images has produced better depth maps by averaging out noise and reducing ambiguities [12], [20]. Space-sweep approaches, which project multiple images onto a series of imaging surfaces (usually planes), also explore significant data redundancy for better reconstruction (e.g., [5], [15], [27], [32]). Space-sweep approaches in general need to discretize the scene volume and, therefore, sampling strategies ([3], [33]). Though the use of multiple images for stereo reconstruction improves accuracy, the computation cost may become an issue.

In this paper, we present a new approach to computing dense 3D reconstruction from a large collection of images. First, we constrain our camera motion to planar concentric circles [30]. This constraint is practical and can easily be satisfied using a number of simple camera rigs. For each concentric circle, we take one or more columns of pixels from each input image (or use line scan sensors such as linear pushbroom cameras [7]) and rebin these into *multiperspective panoramas*. Rather than using the original perspective images, we perform stereo reconstruction from these resampled and rebinned multiperspective panoramas.

- *Y. Li and C.-K. Tang are with the Computer Science Department, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: liyin@ust.hk, cktang@cs.ust.hk.*
- *H.-Y. Shum is with Microsoft Research Asia, 3F, Beijing Sigma Center, No. 49, Zhichun Road, Haidian District, Beijing 100080, China. E-mail: hshum@microsoft.com.*
- *R. Szeliski is with Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399. E-mail: szeliski@microsoft.com.*

Manuscript received 18 Sept. 2002; revised 13 May 2003; accepted 25 July 2003. Recommended for acceptance by R. Kumar.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117405.

In previous work, [8] observed that such panoramas capture the range information, although neither epipolar analysis nor dense stereo was provided in [8]. In [21] and [28], it was shown that the only existing epipolar geometry that can be shared by three or more views is the planar epipolar geometry, which is the requirement to produce epipolar plane images (EPI). One way to produce it is by linearly translating the pushbroom camera. However, in this paper, we show that in many cases the epipolar geometry of multiperspective panoramas can be well approximated by horizontal epipolar lines, as in the conventional stereo algorithms. This allows a wide range of preexisting stereo matching algorithms to be applied without much modification to multiperspective panoramas representation. In this paper, we develop two algorithms to reconstruct depth from such multiperspective panoramas. The first is a *cylinder sweep stereo* algorithm. The second is a multibaseline stereo matching on *locally linear epipolar plane images* that makes use of the approximate linear epipolar geometry and perform *dense 3D reconstruction*. Both algorithms output a panoramic depth map. We demonstrate experimentally that our approach produces high-quality reconstruction results.

Preliminary versions of this paper have appeared in [17], [31]. This paper explains the epipolar geometry for multiperspective panoramas, describes and compares the two algorithms we developed for computing panoramic depth maps, and presents experimental results.

1.1 Overview of Multiperspective Panoramas

Our multiperspective panoramas are a special case of the more general *multiperspective panoramas for cel animation* [34], and are actually very similar to *multiple-center-of-projection images* [24], *manifold mosaics* [23], and *circular projections* [22]. They are also closely related to images obtained by *pushbroom cameras* [7]. Unlike these approaches, however, we constrain our camera motions to be along one or more radial paths around a fixed rotation center and always take the same column from a given camera to generate a multiperspective panorama. As we show in this paper, this results in an epipolar geometry that, in most cases, is well approximated by traditional horizontal epipolar lines.

The idea of resampling and rebinning has recently become popular in image-based rendering. For example, the Lumigraph and Lightfield resample all captured rays and rebin them into a 4D two-plane parameterization [6], [16]. Rebinning the input images produces a representation that facilitates the desired application. For the Lumigraph, this is the resynthesis of novel views. For our multiperspective panoramas, the application is 3D stereo reconstruction of depth maps to be associated with panoramic images in order to support a "look around and move a little" kind of viewing scenario [30].

The advantage of extracting depth from multiperspective panoramas is that it provides a 360 degree field of view. This wide field of view dense depth map enables the application of view interpolation between panoramas. In spite of this very wide field of view, adapting an existing stereo algorithm to multiperspective panoramas stereo is not difficult. Some approaches [11], [18] use stereo matching of single-perspective panoramic images taken at several different locations. However, the sampling of corresponding pixels is nonuniform in the radial and angular directions, resulting in biased stereo reconstructions. For instance, points along the baseline of two panoramas cannot be reconstructed. Moreover, the epipolar geometry for panoramic images is complicated because epipolar curves are sine curves [18]. In contrast, the epipolar geometry of multiperspective panoramas can often be well approximated by horizontal lines (Section 3). Therefore, traditional stereo matching algorithms can be used with few modifications. In this paper, we propose two algorithms to extract dense depth map from multiperspective panoramas. The first one is a cylinder sweep stereo algorithm, while the second is multibaseline stereo with locally linear EPIs.

1.2 Outline of this Paper

The organization of this paper is as follows: In Section 2, we introduce how multiperspective panoramas are captured and how they are generated from regular perspective images. In Section 3, we analyze and show that the epipolar geometry of multiperspective panoramas, to first order, consists of horizontal lines. Two stereo reconstruction methods are described in Sections 4 and 5, respectively. One is a novel cylinder sweep method. The other is a multibaseline stereo using epipolar plane images. In Section 6, we show the experimental results of both algorithms and, in Section 7, we compare the features of two algorithms. Finally, we conclude in Section 8.

2 Multiperspective Panorama Imaging

To generate multiperspective panoramas, the camera motion is constrained to planar concentric circles. A multiperspective panorama is formed by selecting the same column from each of the original perspective images taken at different positions along a circle. Multiperspective panoramas differ from conventional single-perspective panoramas in that each column in a multiperspective panorama is taken from a different optical center.

Two camera rigs have been proposed to capture multiperspective panoramas [30]. These multiperspective panoramas have also been used to synthesize novel view images without 3D information. There exist, however, vertical distortions in the synthesized images without depth correction.

We first review these two camera rigs: *concentric panoramas* and *swing panoramas*. The concentric panorama is theoretically simple to construct and, thus, very suitable for synthetic scene and experimental evaluation. Swing panorama is a practical set-up for capturing real multiperspective panoramas.

2.1 Concentric Panoramas

The first proposed capture device uses several slit cameras (or regular cameras where only a few columns of pixels are kept) mounted on a rotating bar as shown in Figs. 1a and 1b. It is also possible to use only a single slit camera and to slide it to different locations along the bar before rotation. A multiperspective panorama is constructed by collecting all slit images at all rotation angles. We call these images *concentric panoramas*. Fig. 11a in Section 6 shows a synthetically generated concentric panorama. Figs. 11e, 11f, and 11g show detail views from several input panoramas illustrating the parallax effects. We can also use a more general configuration of one or more cameras oriented at various angles other than along the tangent direction and sample one or more columns

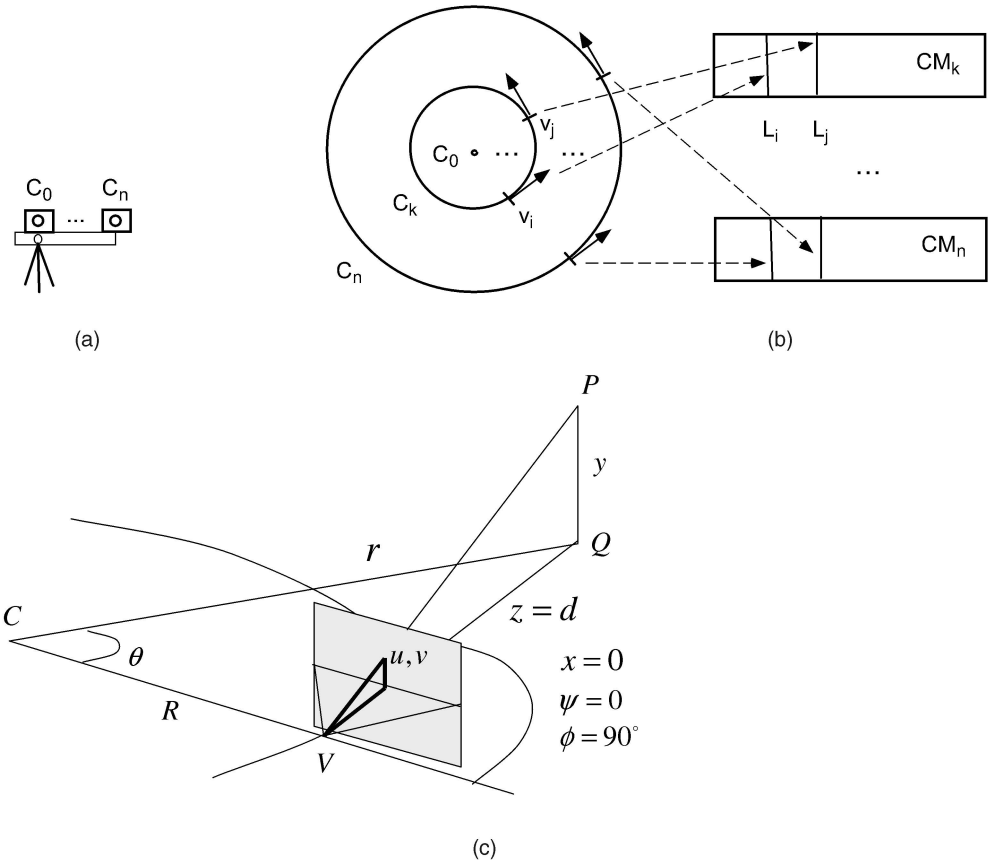


Fig. 1. Concentric panoramas: (a) acquisition rig, (b) rebinning process, and (c) imaging geometry of concentric panoramas.

of pixels from each camera to construct the concentric panoramas.

The general imaging geometry and trigonometric relationships for a single concentric panoramas are shown in Fig. 1c. Here, C is the center of rotation, around which the camera rotates with radius of R. The camera is located at V and the selected column of pixels is looking at a scene point P, which is at an in-plane distance r from the center C and at an in-plane distance d from the camera V. The current angle of rotation, denoted by θ, varies between 0 degrees and 360 degrees. In this configuration, the horizontal dimension of a panoramic image is indexed by θ. The other (vertical) axis in the panoramic image is indexed by the elevation or row number y.

The plane of pixels captured by this particular panoramic imager forms an angle ϕ with the *swing line* in the normal direction connecting V and C. Let ψ be the in-plane viewing angle of P in camera V, i.e., $\psi = \tan^{-1}((x - x_c)/f)$, where x is the column number in the input image, x_c is the center column, and f is the focal length in pixels. In the concentric panorama configuration (Fig. 1c), $\phi = 90^\circ$ and $\psi = 0$. Note that, if ϕ is not 90°, the resulting panorama has an effective radius $R \sin \phi$. We vary R to obtain different concentric panoramas, depending on the number of samples K.

2.2 Swing Panoramas

The other design is to swing a regular camera mounted on a rotating bar looking outwards, as shown in Figs. 2a and 2b. In this case, different columns are used to construct different multiperspective panoramas. For example, a rebinned panoramic image may have 2,160 columns with $1/6$ degree increment, each one taken from the same column in successive frames of a given input video. We call such panoramas *swing panoramas* (Peleg et al. call these *circular projections* [22]). A video sequence of F frames of image size $W \times H$ can be rebinned into (up to) W panoramas with size $F \times H$, as illustrated in Fig. 6.

The general imaging geometry and trigonometric relationship for a multiperspective panorama is shown in Fig. 2c. Definitions of ϕ and ψ are same as in concentric panoramas. Different swing panoramas are formed by changing the ψ angle, which results in panoramas of different ψ by the rebinning construction (Fig. 2b). When multiple columns are selected from a camera pointing outward from C in the swing panoramas configuration (Fig. 2c), we have $\phi = \psi$.

2.3 Summary and Comparison

To summarize:

- Each panoramic image is indexed by (θ, y);
- Each panoramic image's acquisition geometry is parameterized by (R, ϕ, ψ);
- For swing panoramas, $\psi = \phi$, $\phi_k = \tan^{-1}(kw), k = -K \ldots K$;
- For concentric panoramas, $\psi = 0, \phi = \pm 90^\circ, R_k = kR_1$, $k = 0 \ldots K$.

3 Epipolar Geometry of Multiperspective Panoramas

Conventional panoramic images taken at different locations can be used to compute 360 degree field of view stereo.

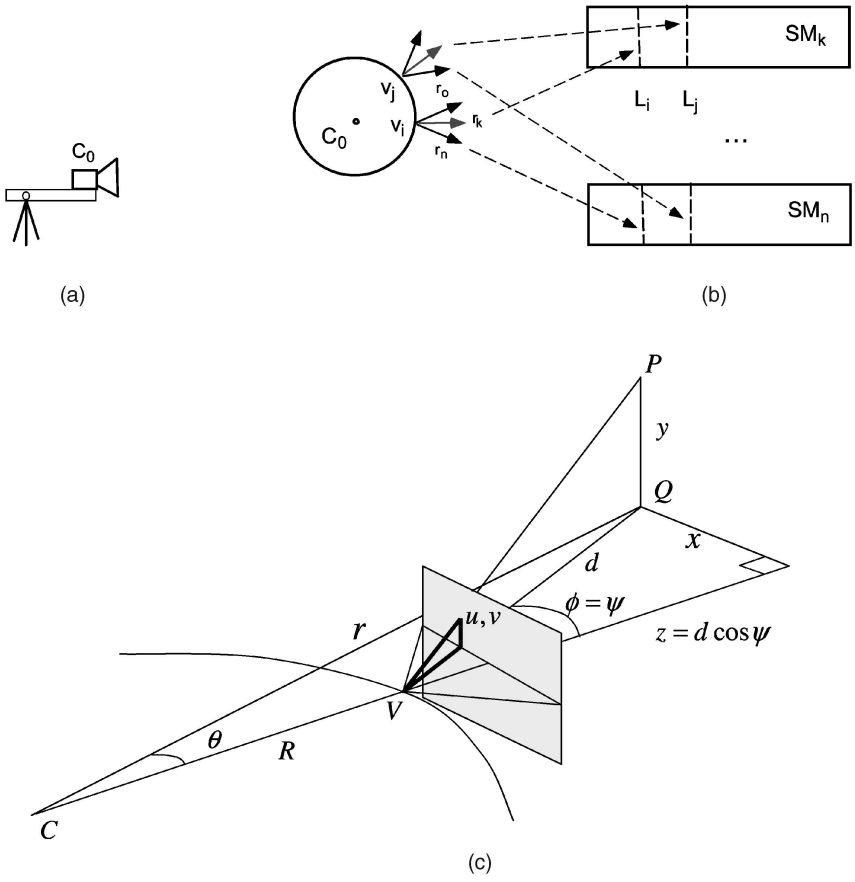


Fig. 2. Swing panoramas. (a) acquisition rig, (b) rebinning process, and (c) Imaging geometry of swing panorama.

However, they have a nonlinear epipolar geometry. Therefore, the matching window shape has to be carefully designed and the multibaseline stereo has to be redesigned. In contrast, the epipolar geometry of multiperspective panoramas can be well approximated by horizontal lines under reasonable and practical constraints. In fact, for a symmetric pair of multiperspective panoramas, the epipolar geometry is exactly the same as horizontal parallel motion. Consequently, a fixed matching window shape combined with a linear 1D search can be used during stereo matching.

In this section, we analyze the epipolar geometry of multiperspective panoramas. First, we derive the formulas for the horizontal and vertical parallax of a point located on a cylindrical surface of radius r (Figs. 1c and 2c).

3.1 Horizontal Parallax

Using the basic law of sines for triangles, we have

$$\frac{R}{\sin(\phi-\theta)} = \frac{r}{\sin(180^\circ-\phi)} = \frac{d}{\sin\theta} \tag{1}$$

or

$$\theta = \phi - \sin^{-1}\left(\frac{R}{r}\sin\phi\right). \tag{2}$$

Therefore, the horizontal parallax $\Delta\theta_{2:1} = \theta_2 - \theta_1$ for a point at a distance r seen in two panoramic images I_1 and I_2 consists of a constant factor $\phi_2 - \phi_1$ and two terms depending on r. If we (circularly) shift each panoramic image by ϕ_i, the first factor drops out, leaving

$$\Delta\theta_{2:1} = \sin^{-1}\left(\frac{R_1}{r}\sin\phi_1\right) - \sin^{-1}\left(\frac{R_2}{r}\sin\phi_2\right). \tag{3}$$

3.2 Vertical Parallax

The vertical parallax can be derived using the following observation. Recall that, according to the laws of perspective projection, the appearance (size) of an object is inversely proportional to its distance along the optical axis, e.g., $x = fX/Z, y = fY/Z$. For pixels at a constant distance r from C and, therefore, at a constant distance from V along the optical axis, the vertical scaling can be computed directly from this distance $Z = d\cos\psi$ (Figs. 1c and 2c). Here, d is the in-plane distance between V and P and $d\cos\psi$ is the distance along the optical axis (as we mentioned before, typically $\psi = \phi$ or $\psi = 0$).

We can write this scale factor as $s = f/Z = f/(d\cos\psi)$. Using the law of sines (1) again, we can compute the change of scale between two panoramic images as

$$s_{2:1} = \frac{s_2}{s_1} = \frac{d_1\cos\psi_1}{d_2\cos\psi_2} = \frac{\sin\theta_1/\sin\phi_1}{\sin\theta_2/\sin\phi_2}\frac{\cos\psi_1}{\cos\psi_2}.$$

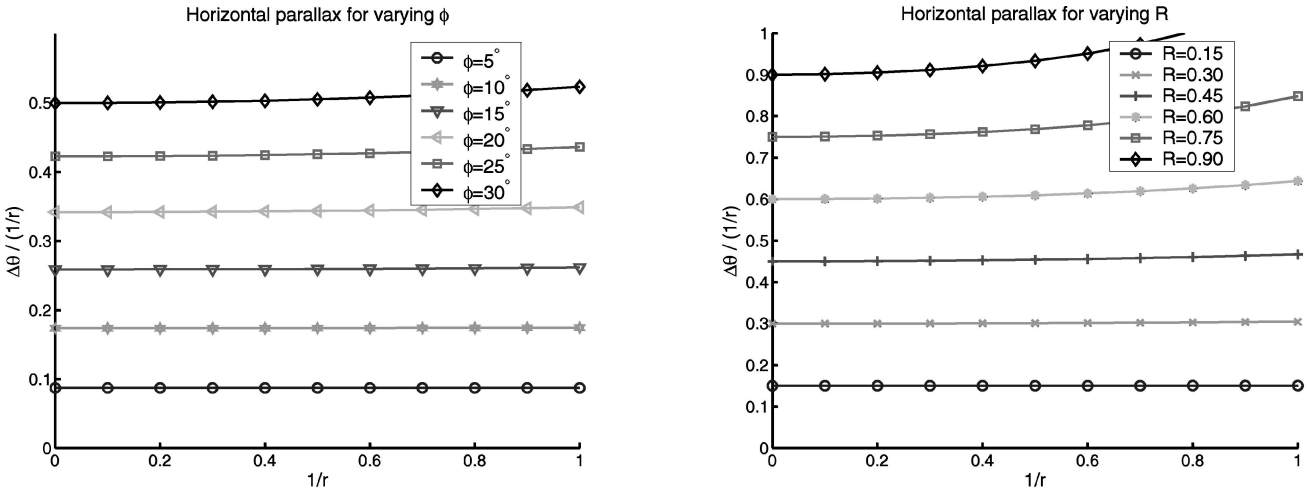


Fig. 3. Plot of horizontal parallax (divided by $1/r$) for varying values of ϕ and R.

Expanding $\sin\theta$, where θ is given by (2), we obtain

$$\sin\theta = \sin\phi\cos\left(\sin^{-1}(\frac{R}{r}\sin\phi)\right) - \cos\phi\sin\left(\sin^{-1}(\frac{R}{r}\sin\phi)\right)$$
$$= \sin\phi\left[\sqrt{1-(\frac{R}{r}\sin\phi)^2} - \frac{R}{r}\cos\phi\right].$$

We can thus rewrite the scale change as

$$s_{2:1} = \frac{\sqrt{1-(\frac{R_1}{r}\sin\phi_1)^2} - \frac{R_1}{r}\cos\phi_1}{\sqrt{1-(\frac{R_2}{r}\sin\phi_2)^2} - \frac{R_2}{r}\cos\phi_2}\frac{\cos\psi_1}{\cos\psi_2}. \quad (4)$$

The first factor in this equation depends on r and goes to 1 as $r \to \infty$. The second factor is a global scale that compensates for the off-axis angle of a given column.

3.3 Epipolar Geometry of Symmetric Multiperspective Panoramas

An imaging configuration of special interest is a two-frame stereo arrangement where the optical rays are symmetric with respect to the swing line $\overline{CV}$, i.e., $\phi_1 = -\phi_2$ (we also assume that $\psi = \phi$ or $\psi = 0$). This occurs when, for example, the left and right columns of a swing stereo sequence [22], [29] are taken or when you fix two cameras at equal distances but at opposite ends of a rotating beam (concentric panorama).

An important conclusion of (4) is that, for a *symmetric pair* of multiperspective panoramas as defined above, the epipolar geometry consists of *horizontal lines*. That is, $s_{2:1} = 1$. An informal proof could be obtained by drawing another point P' in Fig. 1c at an angle $-\phi$ and observing that $z = d\cos\psi$ is the same for both viewing rays. This conclusion first appeared in [31] and was later generalized in [28]. In [21], epipolar surfaces similar to [28] are classified, where the epipolar geometry for multiperspective images are generalized.

A direct consequence of selecting such a pair of panoramic images is that *any* traditional stereo algorithm [26] can be used. More general camera configurations, e.g., those which do not constrain cameras to planar motion, are discussed in [29].

3.4 Small Disparity and Angle Approximations

In practice, we would like to use more than two images in order to obtain a more accurate and robust correspondence [5], [20]. In this section, we study whether the epipolar geometry is sufficiently close to a horizontal epipolar geometry so that conventional multiimage stereo matching algorithms such as EPI analysis [4], SSSD [20], and plane sweep [5], [12], [32] can be used.

A requirement for a conventional multibaseline stereo algorithm to be applicable is that the location of pixels at different depths should be explained by a collection of pinhole cameras. In our case, we further restrict our attention to the horizontal epipolar geometry, in which case, the horizontal parallax as specified in (3) must be of the form

$$\Delta\theta_{k:0} = m_k f(r),$$

i.e., we have a fixed linear relationship between horizontal parallax and some common functions of r for all images. The horizontal parallax equation given in (3) does *not* exactly satisfy this requirement. However, if either R/r or $\sin\phi$ is small in both images, we obtain

$$\Delta\theta_{k:0} \approx \frac{R_k}{r}\sin\phi_k - \frac{R_0}{r}\sin\phi_0 = [R_k\sin\phi_k - R_0\sin\phi_0]r^{-1}. \quad (5)$$

Therefore, the inverse of r plays the same role as inverse depth (*disparity* [20]) does in multibaseline stereo.

Fig. 3 plots (3) for horizontal parallax as a function of r^{-1} for various values of ϕ and R. The left plot shows the ratio of $\Delta\theta_{k:0}$ (divided by $1/r$) to $1/r$ (since it is very hard to tell the deviation from linearity by eye) for $\phi_0 = 0$ (central column) and varying ϕ_k for swing panoramas. The right plot shows the ratio of $\Delta\theta_{k:0}$ (divided by $1/r$) to $1/r$ for $R_0 = 0$ (no parallax) and varying R_k for concentric panoramas with $\phi_k = 90^\circ$ and $\psi_k = 0$. As we can see from these plots, the linear approximation to parallax is quite good as long as the nearest scene point does not get too close, e.g., no closer than 50 percent of R for moderate focal lengths.

A second requirement for assuming a horizontal epipolar geometry is that the vertical parallax needs to be negligible (preferably under one pixel). For images of about 240 lines (a single field from NTSC video), we would like $|\Delta y| \leq$

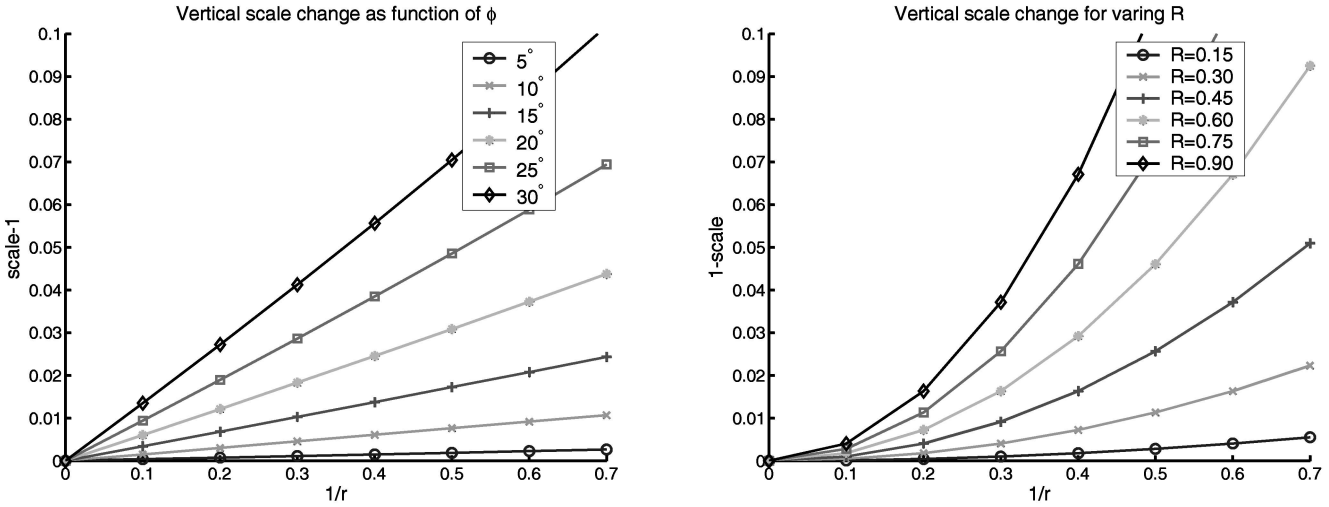


Fig. 4. Plot of vertical parallax (scale change—1) for varying values of ϕ and R.

$120|s_{2:1} - 1| < 1$ (120 is half of the image height), i.e., $|s_{2:1} - 1| < 0.008$.

We can approximate the vertical parallax (4) under two different conditions. For swing stereo ($R_1 = R_2$, $\psi = \phi$), assume that $\phi_2 = 0$ (central column) and ϕ_1 is small. We can expand (4) to obtain

$$\begin{aligned} s_{2:1} &\approx \cos\psi_1 \left[1 - \frac{R^2}{2r^2}\sin^2\phi_1 - \frac{R}{r} + \frac{R}{2r}\sin^2\phi_1\right]\left[1 + \frac{R}{r}\right] \\ &\approx \cos\psi_1 \left[1 + \frac{R}{2r}\sin^2\phi_1 - \frac{R^2}{r^2}\right]. \end{aligned} \tag{6}$$

Thus, once the global scale change by $\cos\psi_1$ (which is independent of depth) is compensated for, we have a vertical parallax component that is linear in R/r and quadratic in $\sin\phi$.

For concentric panoramas, $\phi = 90^\circ$ and $\psi = 0$. Therefore,

$$s_{2:1} = \frac{\sqrt{1 - R_1^2/r^2}}{\sqrt{1 - R_2^2/r^2}} \approx 1 + \frac{1}{r^2}\left(R_2^2 - R_1^2\right).$$

The vertical parallax is inversely proportional to squared distance r and proportional to the difference in squared radii R.

Fig. 4 plots the exact formula (4) for vertical parallax as a function of r^{-1} for various values of ϕ and R. The left plot shows scale change $s_{k:0} - 1$ for $\phi_0 = 0$ (central column) and varying ϕ_k for swing panoramas. The right plot shows $s_{k:0} - 1$ for $R_0 = 0$ (no parallax) and varying R_k for concentric panoramas with $\phi_k = 90^\circ$ and $\psi_k = 0$. As we can see from these plots, the amount of vertical parallax is very small (< 1 percent) if the field of view is moderately small or the ratio of the nearest scene point to the variation in radial camera positions is large.

To wrap up this section, we have the following result: When the disparity R/r and/or off-axis angle ϕ are small, we obtain a nearly horizontal epipolar geometry (classic multi-baseline stereo [20]), after compensating once for the vertical geometry scaling of each image. For our *swing panoramas*, we can further make use of the above to derive the following valid approximations, which is useful in later discussion.

1. The epipolar geometry is approximately horizontal, i.e., $s_{2:1}$ is constant up to order $O(\frac{R}{r}\phi^2)$. This can be explained from the above result and (6).
2. The ratio of horizontal disparity among multiperspective panoramas, i.e., $\frac{\Delta\theta}{\Delta\phi}$, is linear to $\frac{R}{r}$ up to order $O(\phi)$. For swing panoramas, notice that $\psi = \phi$, $\cos\psi = 1 - O(\phi^2)$, and $R_k = R_0 = R$. From (5), $\Delta\theta = \Delta\sin\phi \cdot \frac{R}{r} = \cos\phi\Delta\phi \cdot \frac{R}{r} = \Delta\phi\frac{R}{r}(1 - O(\phi^2))$. The conclusion can be drawn since (5) is obtained from (3) by the first order approximation.

For example, typically, we use a camera with 40 degree horizontal FOV lens to take images with the horizontal resolution of 400 pixels. To make the approximations in 1 and 2 valid and practical for swing panoramas, we take the panorama rebinned from the 20th column ($x = 20$); according to Fig. 2c, we have $\phi = \tan^{-1}(20\tan(20^\circ)/200) \approx 2^\circ \approx 0.04$, which corresponds to a very small $s - 1$, as shown in Fig. 4a.

4 Cylinder Sweep Stereo

In this section, we develop a novel multi-image cylinder sweep stereo reconstruction algorithm that generalizes the concept of plane sweep stereo. Because of the special structure of our concentric panoramas, the cylinder sweep algorithm only requires horizontal translations and vertical rescalings of the panoramic images during matching.

The cylinder sweep stereo algorithm uses a few multi-perspective panoramas to perform matching. Plane-sweep and space coloring/carving stereo algorithms have recently become popular because they support true multi-image matching [5], enable reasoning about occlusion relationships [12], [14], [15], [27], [32], and are more efficient than traditional correlation-based formulations [10]. Traditional stereo matching algorithms pick a window around each pixel in a reference image and then find corresponding windows in other images at every candidate disparity (searching along an epipolar line). Plane sweep algorithms consider each candidate disparity as defining a plane in space and project all images to be matched onto that plane, using a planar perspective transform (homography) [1], [5], [32]. A per-pixel fitness metric (e.g., the variance of the corresponding collection of pixels) is first computed and then aggregated

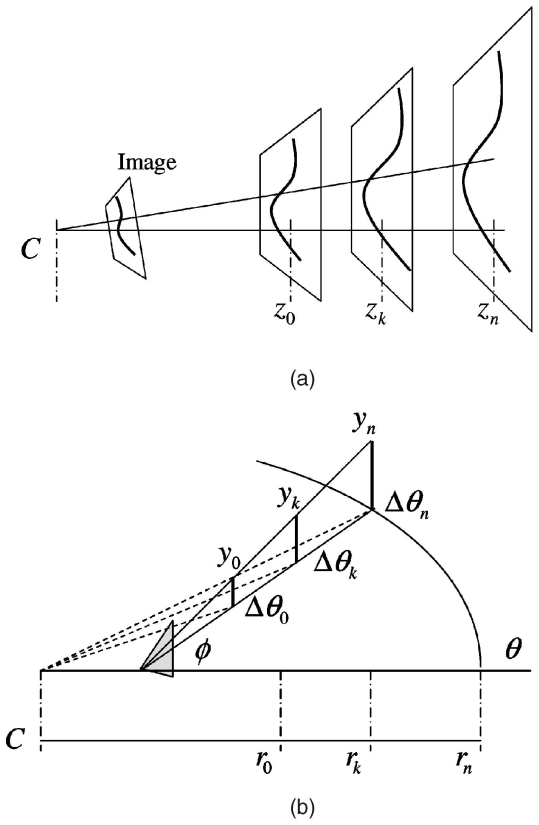


Fig. 5. Illustration of (a) plane sweep algorithm and (b) cylinder sweep algorithm.

spatially using an efficient convolution algorithm (e.g., a moving average box filter) or some other techniques [25], [26]. After all the cost functions have been computed, a winning disparity can be chosen. If the planes are processed in a front-to-back order, occlusion relationships can also be inferred [27].

Our novel *cylinder sweep* algorithm works similarly. Instead of projecting images onto planes, however, we project our multiperspective panoramas onto cylinders of varying radii r. The transformations that map panoramas onto these cylinders and, hence,onto each other, are particularly simple, i.e., the reprojection of each panoramic image onto a surface of radius r consists of a horizontal translation and a vertical scaling. This conclusion follows directly from (3) and (4) for horizontal and vertical parallax. A less formal but more intuitive argument is to just observe that the image of any column of a cylinder at a fixed radius r seen by a concentric pushbroom camera is just a scaled version of the pixels lying on that cylinder. Since our representation has no preferred direction, the shift between the panoramic image and the cylinder must be the same for all pixels.

4.1 The Cylinder Sweep Algorithm

The traditional plane sweep algorithm projects images onto a set of parallel planes, which are swept through space along a line normal to each plane, as shown in Fig. 5a. The volume of interest in space is bounded by two planes $Z = z_{min}$ and $Z = z_{max}$. The volume is sampled by a sweeping plane at a discrete number of equally spaced Z intervals within the limits z_{min} to z_{max}. The projection of images onto the candidate plane $Z = z_i$ is simply a planar projective transformation (homography).

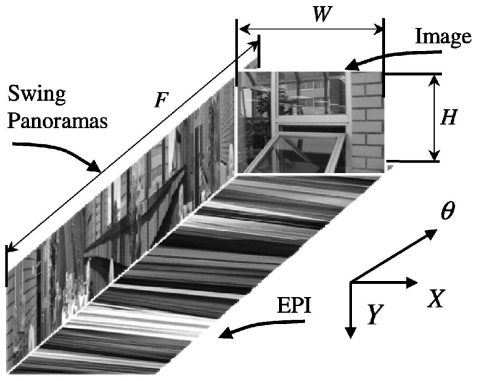


Fig. 6. An EPI is produced by extracting a horizontal slice as shown. Each EPI corresponds to a 2D potential inverse depth image of a scanline in the corresponding multiperspective panorama.

For our novel cylinder sweep algorithm, the input multiperspective panoramas are reprojected onto concentric cylinders instead of planes. Once the projection is done, the same strategy can be applied on these data as in the traditional plane sweep algorithms [5]. Details of the reprojection process are shown in the following and illustrated in Fig. 5b.

We start with a set of multiperspective panoramas, (R_i, ϕ_i, ψ_i), $i = 1, \cdots, K$, which can be either concentric or swing panoramas. K is the number of panoramas used, e.g., $K = 7$ in our experiments. Recall that each panorama is parameterized by (θ, y). We perform horizontal shifting for each panorama with an offset equal to $\Delta\theta = \phi_i$ to satisfy the precondition of (3).

Then, the projection from each panorama onto each sweeping plane is simply one horizontal shift plus one vertical scaling. As the shift and scaling is relative, without losing generality, we can always fix one panorama without shifting and scaling, say (R_0, ϕ_0, ψ_0). In practice, in order to take the advantage of symmetric property, we usually choose the panorama at $\phi_0 = 0$ in swing panoramas or $R_0 = 0$ in concentric panoramas.

For each candidate sweeping cylinder with radius of $r = r_j$, $j = 1, \cdots, N$, horizontal shifting is performed, with an offset equals to $\Delta\theta_{j:0}$, according to (3). Then, we perform the vertical scaling $y_i = y_c + (y_0 - y_c) \times \Delta s_{i:0}$, where y_c is the scaling center, i.e., the intersection of this vertical scan line with sweeping plane, and $\Delta s_{i:0}$ is given by (4).

Since a 360 degree panorama is considered, there is no view clipping in the horizontal direction. Hence, the amount of horizontal shifting does not matter. In contrast, in the traditional plane sweep algorithm, the input images have to be viewed at a restricted range (usually a subset of the camera's field of view) after projection.

Given that the input set of multiperspective panoramas is well captured, the horizontal shift offset and vertical scaling factor are fixed constants for each multiperspective panorama. However, the cameras do not have to lie on the same plane. A sufficient condition is that they should be on the circles that share the same axis. If a camera is on a different plane, y_c does not have to be the center of images, a vertical shift offset can be added in addition to the vertical scaling described above. This enables us to stack cameras vertically in order to exploit vertical parallax, if desired [2].

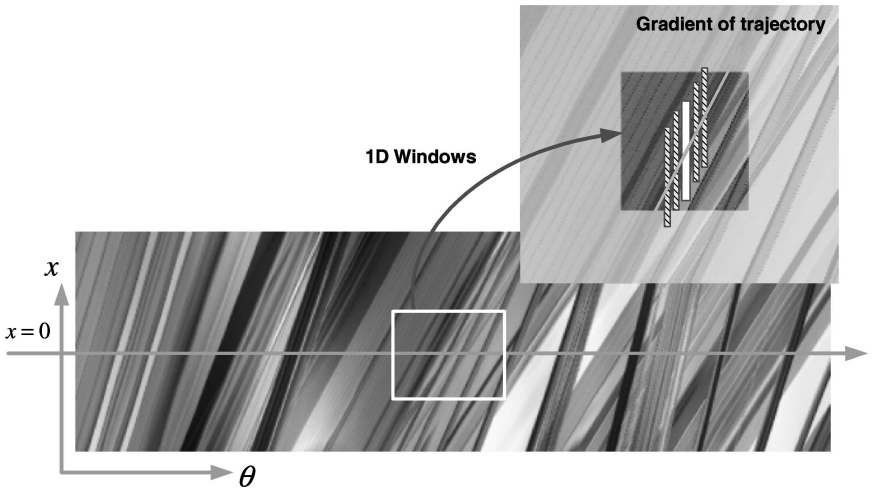


Fig. 7. An example EPI from the real scene of the balcony in Fig. 16. Note the linear patterns. The vertical bars in the inset are 1D windows that are used to estimate the line gradient, which indicate inverse depth.

Once the reprojection is done, the same algorithm as in the traditional plane sweep stereo can be applied to obtain a 360 degree depth map.

4.2 Efficient Algorithm for Swing Panoramas

For swing panoramas, we can develop an even more efficient projection algorithm. According to the results at the end of Section 3.4, there exists a linear relationship between the disparity $\Delta\theta$ and the inverse radius $\zeta = \frac{R}{r}$. We can discretize the sweeping cylinder equally in inverse radius, instead of equally in the radius r. This is particularly useful for outdoor scenes in which $z_{max} = \infty$.

If the panoramas are not far from the center, i.e., ϕ is small, we can further ignore the vertical scaling for swing panoramas. Therefore, the reprojection part of the cylinder sweep algorithm is simply a horizontal shift, which can be efficiently performed.

5 Dense Depth Estimation from Locally Linear EPI

In this section, we describe a multibaseline stereo algorithm that makes use of the approximate horizontal epipolar geometry. The multibaseline stereo algorithm [20] has been proposed to achieve robust reconstruction by taking advantage of the inherent data redundancy in a large collection of images.

In our case of multiperspective panoramas, an approximate EPI[1] is obtained by concatenating corresponding scanlines, as shown in Fig. 6, where x indicates the pixel location along the scan line and θ represents the rotation angle of the camera. One enlarged EPI is depicted in Fig. 7. A straight line in the EPI indicates the locus or trajectory of an image point. By the results presented in Section 3.4, we obtain the following:

$$\frac{\Delta\theta_{k:0}}{\Delta x_{k:0}} \approx \frac{R}{r}. \tag{7}$$

1. EPIs, or epipolar plane images, first proposed by Bolles et al. [4], are extracted from parallel motion image sequence, each of which consists of pixels sharing a common epipolar line in other input images.

Note that the left-hand side of (7) is exactly a line gradient in an EPI, as depicted in Fig. 7. Therefore, the depth estimation from multiperspective panoramas can be translated into slope estimation of the straight lines in an EPI. (See [9] for more recent work in EPI processing.)

This linear relationship further implies that matching only requires a 1D search in a single EPI. It can be implemented as 1D convolution using a *constant* 1D search window. No rectification is needed. Further, since (7) is linear, we simply quantize the inverse depth uniformly without introducing bias or negligence. Hence, the dimensionality of our scene reconstruction problem is reduced from 3D to 2D.

Note that the above derivation pertains to a single EPI in which the slopes detected (along $x = 0$ in Fig. 7) represent the inverse depth of the corresponding scanline. The panoramic depth map can be generated by considering the corresponding EPIs of other scanlines as well, as shown in Fig. 6.

Thus, using multibaseline stereo and a 1D matching window, a panoramic depth map can be reconstructed by estimating line gradients in approximate EPIs derived from rebinned multiperspective panoramas.

5.1 Computing Potential Inverse Depth Image

In this section, we design a multibaseline stereo algorithm based on the above. An EPI is indexed by x and θ, as shown in Fig. 7. Following the traditional multibaseline stereo algorithm [20], we evaluate the similarity with multiple 1D matching window inside individual EPIs.

Let $I(x, \theta)$ be an EPI. Given any θ, we define a 1D window, $W(\theta)$, to be $W(x, \theta) = \{I(x + x_i, \theta) | x_i \in [-w, w] \text{ for integer } w\}$, where x is the center of the 1D window. The typical value of w is 10 in our experiments. Suppose we slide this 1D window along a certain direction and compute the consistency of pixel colors between this 1D window and the overlapping pixels. The line gradient is equal to the direction that produces the maximum consistency.

Let θ_0 be the location of the 1D window centered at $x = 0$. To compute color consistency, we compute sum of squared difference, or SSD, which is defined by

$$SSD(\zeta, \theta)|_{\theta_0} = \sum_{i=-w}^{w} [I(x_i + (\theta - \theta_0)\zeta, \theta) - I(x_i, \theta_0)]^2. \tag{8}$$

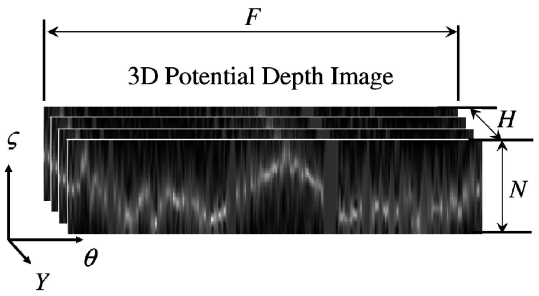


Fig. 8. Three-dimensional potential depth image.

With multiple panoramas, we compute $SSSD$ (sum of SSD) for the reference image at θ_0 in a neighborhood of size M as color consistency evaluation, which is defined as

$$SSSD(\zeta)|_{\theta_0} = \sum_{n=-M}^{M} SSD(\zeta, \theta_n)|_{\theta_0}. \quad (9)$$

Note that the definition of (9) is analogous to the $SSSD$ used in [20] for multibaseline matching and that the matching presented in this section is performed on an EPI. A small $SSSD(\cdot)$ indicates a high color consistency. The typical value of M we use is 5.

By considering more EPIs, a 3D potential inverse depth image is built, as shown in Fig. 8. Each voxel in this volume is the similarity at a given image pixel (y, θ) at a given inverse depth ζ, represented as $P(y, \theta, \zeta)$. It is normalized for each pixel, s.t. $\sum_\zeta P(y, \theta, \zeta) = 1$. The brightest locations indicate the most probable inverse depth surface. A plausible solution for our depth estimation can thus be translated into extracting a salient surface from the 3D potential depth image, assuming the scene is opaque,

$$S = \bigcup_{Y,\theta} \{(Y, \theta, \zeta_n) | P(Y, \theta, \zeta_n) \geq P(Y, \theta, \zeta_i), \forall i = 1 \cdots N\},$$

where N is the number of the quantized (inverse) depths. However, this straightforward winner-takes-all algorithm usually produces many outliers, as shown by the noisy image of Fig. 9a.

It is worth noting that the use of 1D matching windows in our stereo algorithm will inevitably run into the aperture problem. Our solution is to keep a set of possible inverse depth maps at the initial reconstruction stage; the aperture problem is then solved by an adaptive smoothing criterion in the first pass of tensor voting [19], which also removes wrong matches and handles depth discontinuities. The uniqueness constraint is then applied by the second pass of tensor voting.

5.2 Extracting the Inverse Depth Surface

A two-pass algorithm based on tensor voting [19] is proposed. Given the initial set of matches from the 3D potential inverse depth image $P(Y, \theta, \zeta)$, our objectives are two-fold:

1. Remove wrong matches and infer smooth features that are possibly missed due to the aperture problem associated with a 1D matching window, and
2. Infer missing matches and compute the inverse depth with maximum support.

At the same time, we want to preserve depth discontinuities and occlusion boundaries. Two passes of tensor voting are used. The first pass propagates the *continuity constraint* to achieve Step 1. After removing wrong matches, a reliable set of inverse depths are obtained. The second pass implements Step 2 by applying the *uniqueness constraint* along the line of sight. The terminology and exact algorithms we use in this section are explained in Appendix A.

5.2.1 Pass One: Continuity Constraint

In the first pass, S is first computed, where S is the set of voxel locations whose $P(\cdot)$ is the maximum among all values along the line of sight. The algorithm is described below along with a running example.[2]

1. Compute S given by (1). Encode S into a set of default ball tensors. All eigenvalues are made equal to its $P(\cdot)$, as illustrated in Fig. 9a.
2. Compute V, the set of voxel locations whose associated $P(\cdot) \geq p_1$ and $S \cap V = \emptyset$. We also encode V into a set of ball tensors, with all the eigenvalues equal to their respective $P(\cdot)$s.
3. The encoded S and V vote with the ball voting field.
4. S collects votes by tensor addition. The resulting eigensystem is computed.
5. A subset of points in S, whose normalized surface saliencies exceed p_2, is obtained, Fig. 9b.

The choice of p_1 is not critical and is set at 0.01 in our experiments. The choice of p_2 is not critical either since we shall collect votes in every voxel location in the 3D image in pass 2. It is 0.1 in our experiments. Figs. 9a and 9b, respectively, depict the S before and after pass 1. Note that both smooth structures and depth discontinuities are preserved simultaneously, while most of the outliers are eliminated. We use a homogeneous ball tensor in our experiments, where the three dimensions are all related to image resolution. It is a reasonable assumption that the scale differences on the three dimensions are not large.

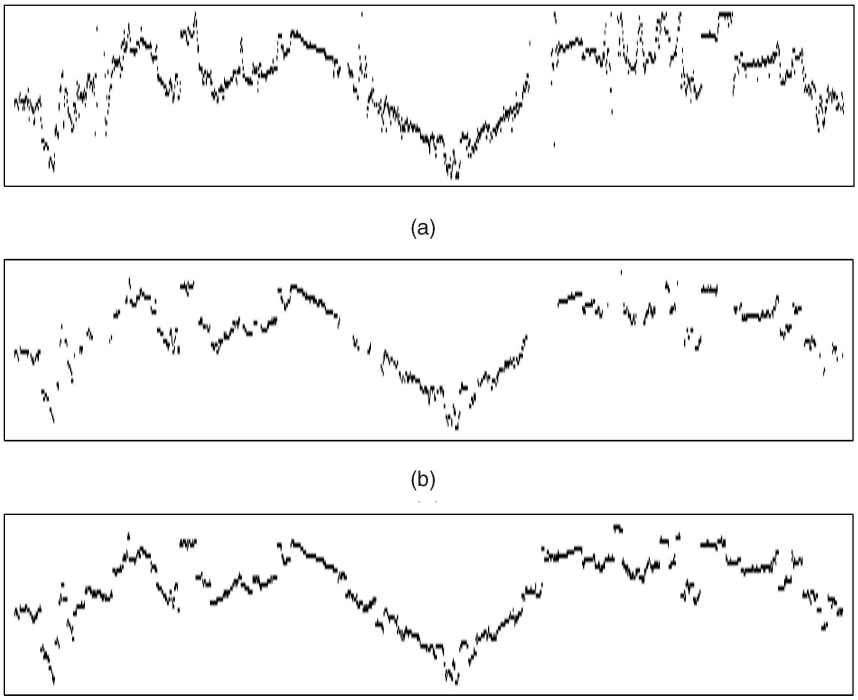
Let $\overline{S} \subset S$ be the resulting set shown in Fig. 9b, which provides more reliable evidence. In pass 2, we resample the whole 3D volume using $\overline{S}$ by computing a generic tensor vote at all quantized inverse depths.

5.2.2 Pass Two: Uniqueness Constraint

In pass two, we apply the uniqueness constraint along the line of sight by voting for the *maximum inverse depth*: The inverse depth that receives the maximum support from the $\overline{S}$.

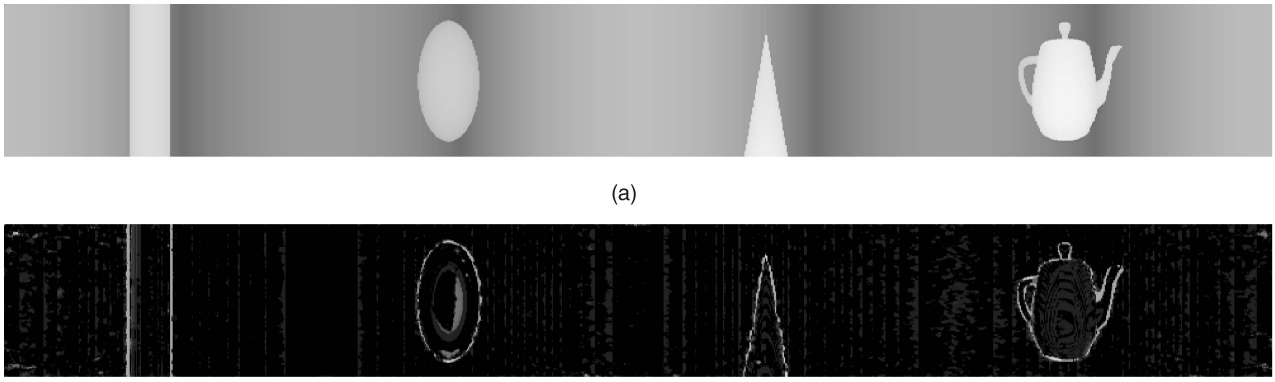
1. Each point in $\overline{S}$ is initially encoded as a ball tensor, with three eigenvalues set to its surface saliency $ssal = \lambda_{max} - \lambda_{mid}$, which is obtained in the first pass. In doing so, voters with higher surface saliency are preferred.
2. Each encoded ball casts a ball vote in its neighborhood to resample the whole 3D volume. For every (Y, θ), we compute *all* N tensor votes, received at (Y, θ, ζ_1), $(Y, \theta, \zeta_2), \cdots, (Y, \theta, \zeta_n)$. A voxel not in $\overline{S}$ will assume a zero tensor initially. In this voting pass, we also use the same σ as the scale of analysis we used in the previous stage. Since $\overline{S} \subset S$, $|\overline{S}| < |S|$. Therefore, $\overline{S}$ is now sparser. There may exist some location (Y, θ) where all N tensor votes received at $(Y, \theta, \zeta_1), (Y, \theta, \zeta_2), \cdots,$

2. For better visualization, we use a 2D example, which is one slice obtained from the 3D volume.



(a) (b) (c)

Fig. 9. Running example. (a) The candidate set S with maximum $SSSD$ along the line of sight. (b) Outlier removal and discontinuity preservation by applying the smoothness constraint. (c) Missing details are filled in by applying the uniqueness constraint.



(a) (b)

Fig. 10. Error map corresponding to Fig. 13. (a) The ground truth panoramic depth map rendered by Discrete 3D Max. (b) The error map in pseudocolor mode.

$(Y, \theta, \zeta_n))$ are zeros. We usually have about 1-5 percent of such locations. To deal with this, we progressively increase σ by one (approximately equals to three voxels) until at least one of these N votes obtain nonzero votes.

3. When the whole (Y, θ, ζ_n) volume has collected all nonzero votes, we apply the uniqueness constraint: for each (Y, θ), we return $\zeta_{Y,\theta}$, that receives the maximum support, or the largest surface saliency along the line of sight:

$$\zeta_{Y,\theta} = \{\zeta_n | ssal(Y, \theta, \zeta_n) \geq ssal(Y, \theta, \zeta_i), i = 1 \cdots N\}.$$

Fig. 9c shows one slice of our result. Note that each column consists of one and only one solution that corresponds to the maximum salient inverse depth.

6 Experiments

We perform experiments on some challenging synthetic and real data to evaluate our approaches.

6.1 Cylinder Sweep Stereo Algorithm

Fig. 11 shows the stereo reconstruction results by our Cylinder Sweep algorithm from seven concentric panoramas that were synthesized with a slit camera rotating along circles of different radii $(0.4, 0.5, \ldots, 1.0)$. The right and the center panoramas are shown in Figs. 11a and 11b, respectively. Horizontal parallax can be observed from close-up of regions of three original panoramas shown in Figs. 11f, 11g, and 11h. Because a small field of view (24 degrees) camera was used, these concentric panoramas have negligible vertical parallax. Using the estimated depth map shown in Fig. 11c, we synthesize a panorama shown in Fig. 11d by warping Fig. 11b

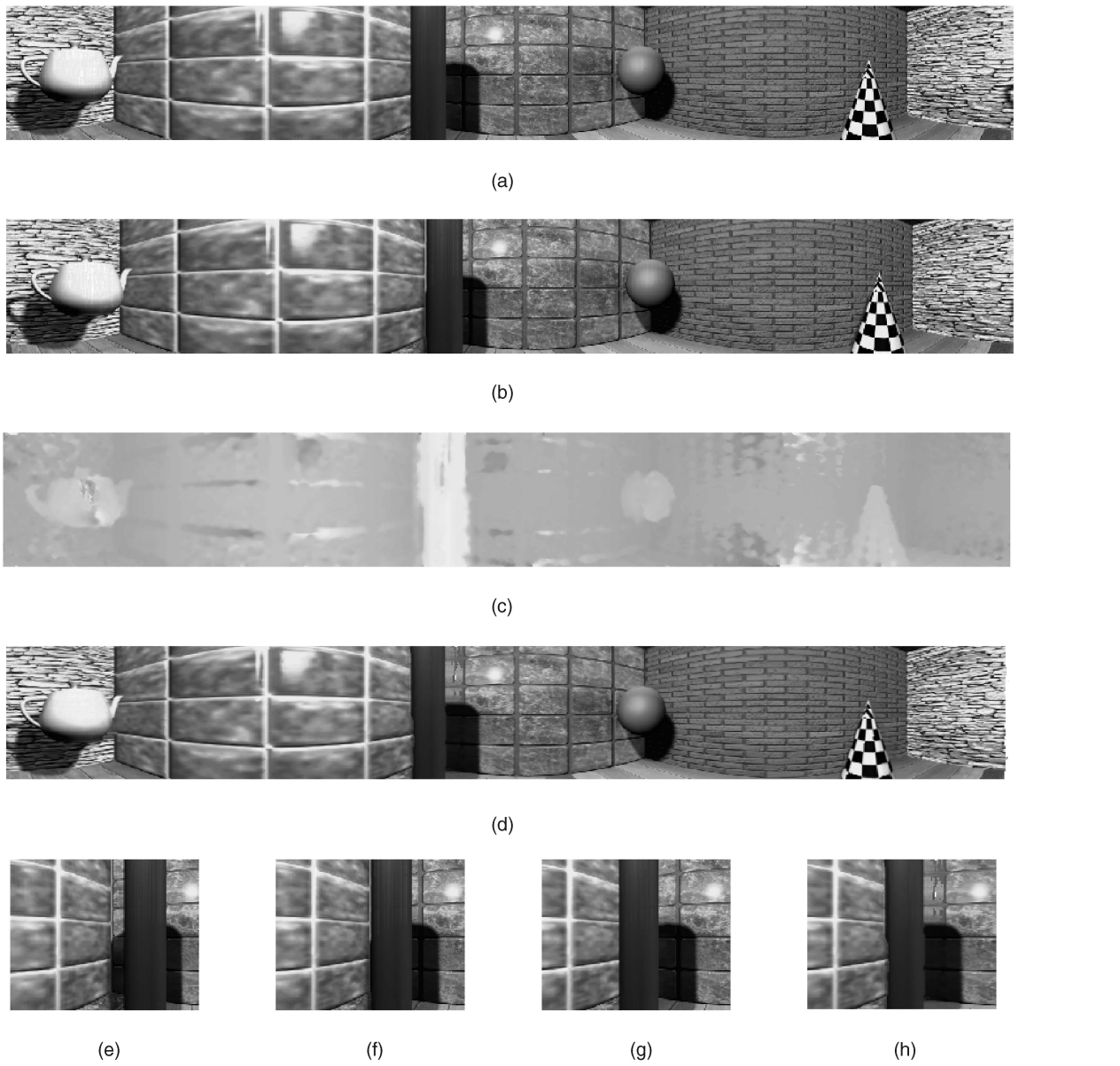


Fig. 11. Concentric panoramic stereo results: (a) an input panorama (rightmost camera, $R = 1.0$), (b) another input panorama (center camera, $R = 0.7$), (c) estimated depth map for the center panorama, (d) panorama resynthesized from center panorama and depth, (e), (f), (g) close-up of input panoramas (note the horizontal parallax), and (h) close-up of resynthesized panorama.

to the same camera parameters as in Fig. 11a using the depth map Fig. 11c. The new panorama is almost indistinguishable from Fig. 11a except in the regions where significant occlusion occurs as shown in the close-up Fig. 11h. Notice that the reflection of the spotlight is synthesized well even though its depth estimation is clearly wrong.

Two rebinned panoramas from a real swing sequence of a lab scene are shown in Figs. 12a and 12b. We used a digital video camera in portrait mode with a field of view of around $27^\circ \times 36^\circ$ and a digitized image size of 240×320 pixels. The camera was mounted off-center on a plate rotated with a stepper motor that provided accurate rotation parameters. After scaling the images vertically by $\cos\psi$, we found that there was a small (0.5 pixel) drift remaining between the panoramas. This was probably caused by a slight rotation of the camera around its optical axis. In practice, compensating for such registration errors is not difficult: A simple point tracker followed by linear regression can be used to obtain the best possible horizontal epipolar geometry. The panoramas after vertical scaling and drift compensation are shown in the close-up regions in Figs. 12e, 12f, and 12g. As you can see, very little vertical parallax remains. The reconstructed depth map is shown in Figs. 12c and a synthesized *novel* panorama (from an *extrapolated* viewpoint) and its close-up are shown in Figs. 12d and 12f, respectively.

6.2 Multibaseline Algorithm on Approximate EPI

We capture the following three swing multiperspective panorama using the setup shown in Fig. 2 using a camera of horizontal FOV of 40 degrees. The synthetic scene with ground true depth image is rendered by Discrete 3DMax and the real scenes are captured using a commercial digital video camera.

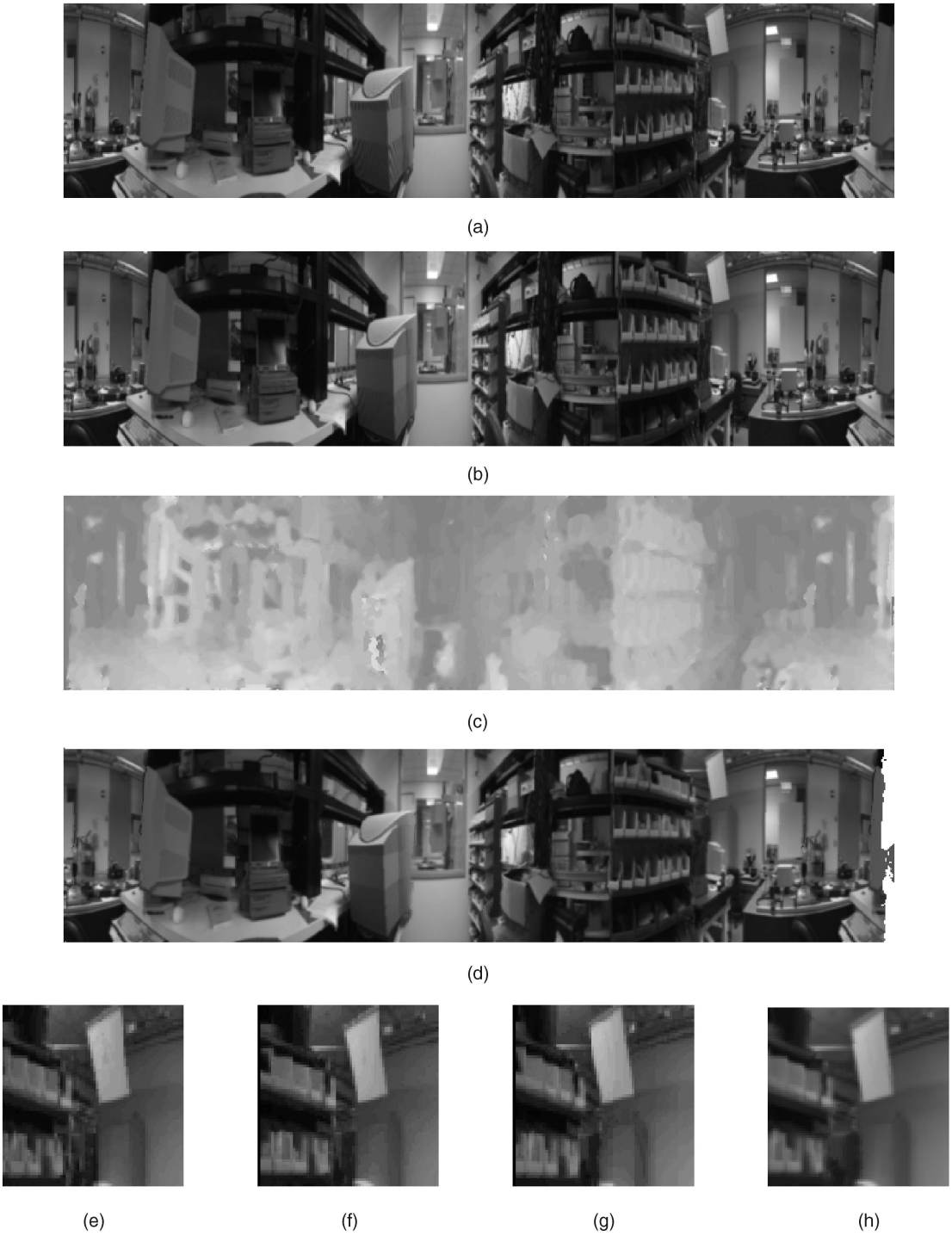


Fig. 12. Swing panoramic stereo results: (a) an input panorama from left column, (b) input panorama from center column, (c) estimated depth map for the center panorama, (d) novel panorama extrapolated from center panorama and depth, (e), (f), (g) close-up of input panoramas, and (h) close-up of extrapolated panorama.

Figs. 13a and 13b show a 360 degree multiperspective panorama for a synthetic *Virtual Room* and its corresponding dense depth map by our method. The multiperspective panorama in Fig. 13a is then reprojected to a novel viewpoint where occlusions between objects (e.g., teapot, ball) and the walls are clearly visible. Due to the cylindrical mapping, the walls appear curved. Using the depth map shown in Fig. 13d, the teapot can be observed from a novel viewpoint at a lower viewing angle as in Fig. 13e. To demonstrate the high-quality reconstruction of the virtual room, we show the top-down view and the top-side view of the Euclidean reconstruction in Figs. 13f and 13g, respectively. Note that the four reconstructed walls are perpendicular and four objects keep their respective shapes very well. The reconstruction quality with 40 swing panoramas (Fig. 13) is much better than that of the cylinder sweep algorithm (Fig. 11), which only uses seven concentric panoramas.

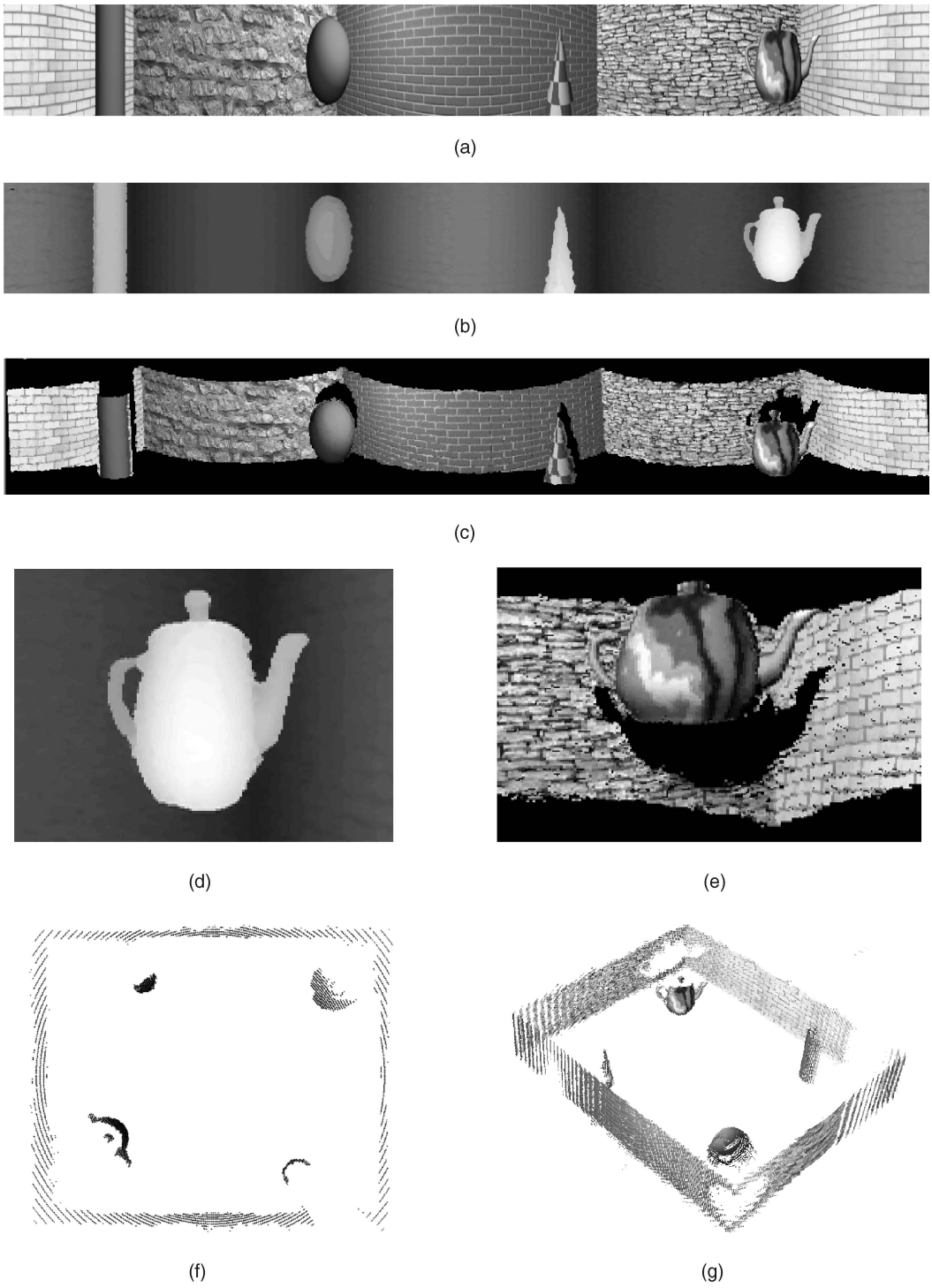


Fig. 13. Virtual Room (synthetic scene): (a) and (b), respectively, show the multiperspective panorama and its corresponding inverse depth map, (c) a novel view of the panorama, (d) depth map of the teapot, (e) a novel view of teapot reprojected with the depth map, (f) and (g) the reconstructed room at top-down and top-side views.

Fig. 10 compares our result on the same synthetic scene in Fig. 13 with the ground truth geometry obtained from Discrete 3D Max. Fig. 10a is the ground truth panorama depth map, which encodes inverse radius as gray level. Fig. 10b shows the error map. The pure black region indicates error that is less than one quantization step (64 steps for whole depth range). The dark gray region means one step error. The bright regions mean that the errors are larger than one step. This error map shows that our algorithm performs well at textured region, while occlusion boundaries suffer some "fattening" problem as well.

Figs. 14 and 15 show the acceptable results and graceful degradation of our approach in two complex real scenes, with severe depth discontinuities, camera noise, and low image resolution. Figs. 14a and 15a show two multiperspective panoramas and Figs. 14b and 15b show the corresponding

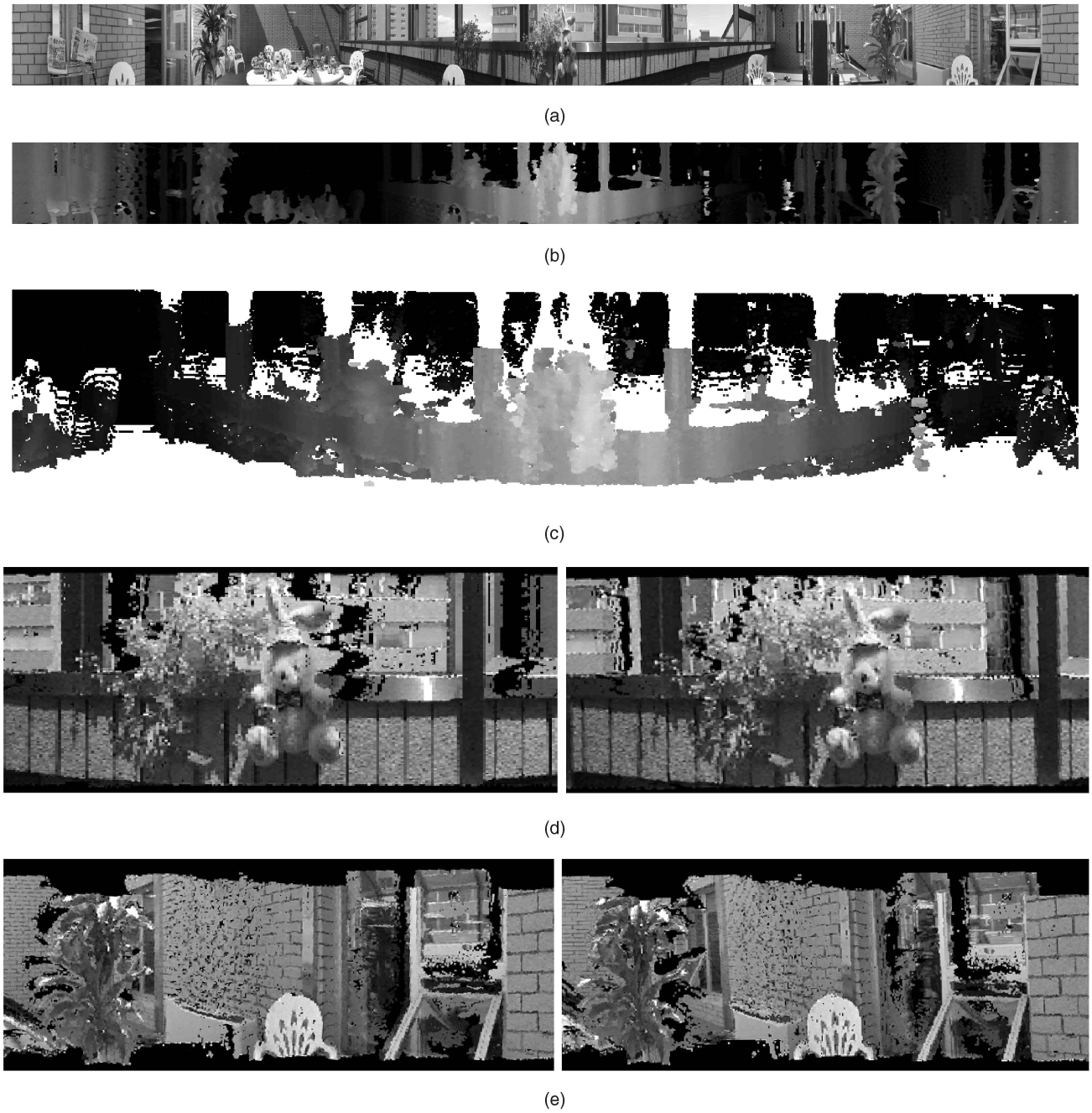


Fig. 14. Balcony (real scene with depth discontinuities, textureless objects, and mirror reflection): (a) Shows one multiperspective panorama, (b) shows the corresponding inverse depth map, (c) is the depth map from a novel viewpoint. (d), (e), (f), and (g) are two close-up pairs of the textured reprojected views. Note the depth inside the window in (f) and (g) is reconstructed as a faraway background due to the mirror reflection.

depth maps automatically computed by our method. In Figs. 14c and 15c , the reprojected depth maps from a novel viewpoint, show the good quality of our reconstruction. Figs. 14 and 15d, 15e, 15f, and 15g show the close-up texture mapped views at novel viewpoints. Note that this is a very challenging outdoor scene with many textureless regions and occlusions. Figs. 14f and 14g show the depth inside the window is reconstructed as a faraway background because of mirror reflection. Note that, while depth discontinuities can still be preserved to a large extent, as shown in Figs. 15d and 15f, the occlusion boundaries are not very well localized.

7 Discussion

In this section, we discuss the properties of the two algorithms we proposed. We have summarized the comparison in Table 1 to help one choose between different algorithms.

The first (cylinder sweep) algorithm does not require the multiperspective panoramas to be rebinned from nearby columns because it can use a nonlinear epipolar geometry. Of course, using an approximate horizontal epipolar geometry, the cylinder sweeping step can be further simplified into a horizontal shifting one. In contrast, the second (multibaseline stereo with locally linear EPI) algorithm

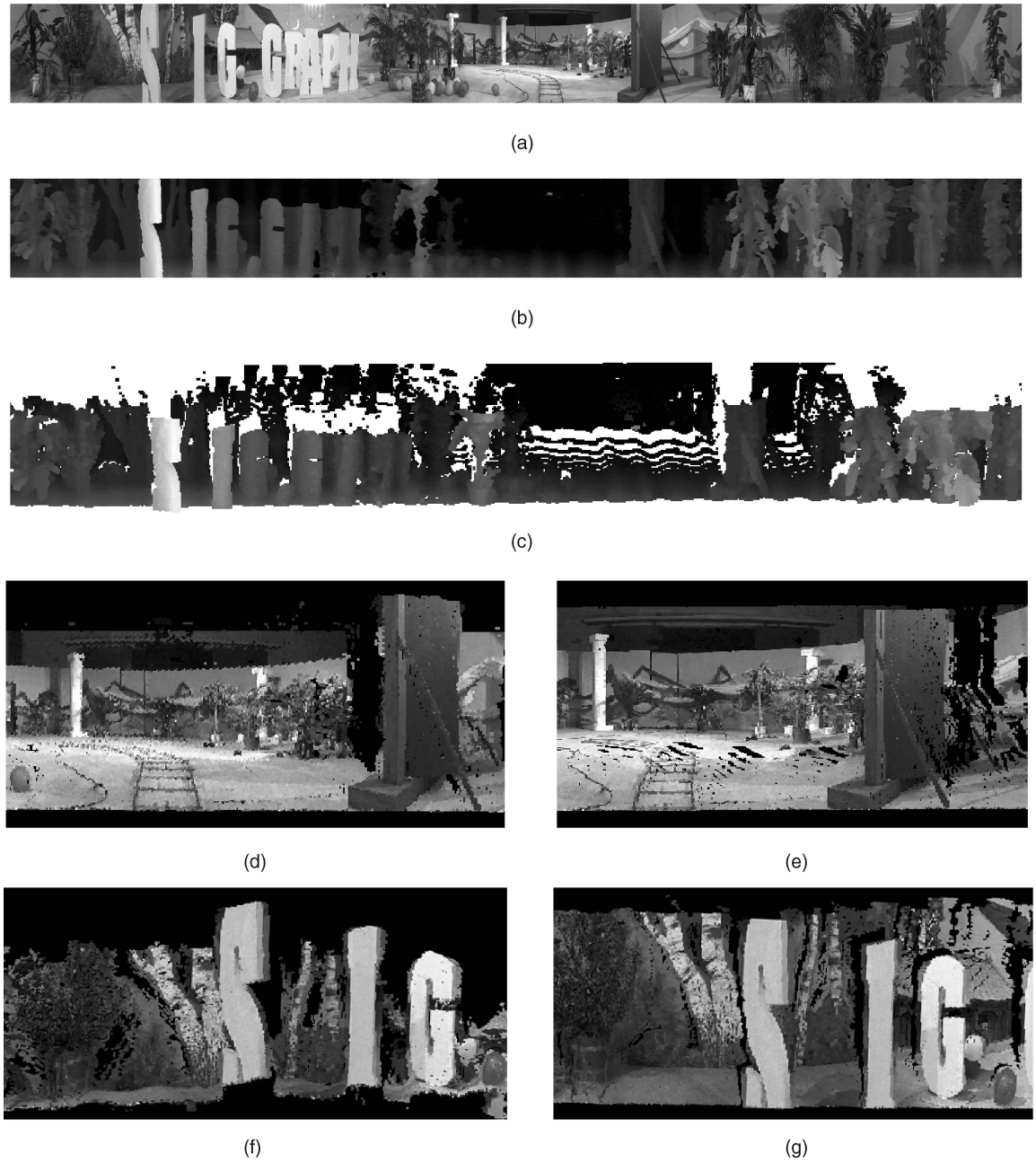


Fig. 15. Film Studio (real scene): (a) Shows one multiperspective panorama, (b) shows the corresponding inverse depth map, (c) is the depth map from a novel viewpoint. (d), (e), (f), and (g) are two close-up pairs of the textured reprojected views. Note the severe depth discontinuity behind the door in (d) and (e) and the standing characters in (f) and (g). Although the depth discontinuities can be preserved to a large extent, the occlusion boundaries are not well located due to the fattening effect.

selects the multiperspective panoramas from nearby columns since it relies on the approximate horizontal epipolar geometry and the local linear relationship of disparity with inverse radius.

Both algorithms output 360 degree field of view dense depth maps. Although the space and time complexities of both algorithms are linear in the number of pixels and the number of discrete depths (in the case of the first algorithm, we may need to perform iterative optimization), the second algorithm makes better use of the approximate horizontal epipolar geometry by decomposing the matching problem into subproblems inside each EPI image.

The first algorithm does offer more flexibility than the second algorithm since there is actually nothing preventing the cameras from being located in different (parallel) planes so long as their rotation axes are all the same (i.e., the camera motions are *coaxial*, rather than concentric). The only difference, in this case, is the addition of some extra depth-dependent *vertical* parallax, which can easily be accommodated in both traditional stereo algorithms and in our novel cylinder sweep algorithm.

Capturing swing panoramas only uses one camera that undergoes simple rotation. Therefore, real scenes are usually captured as swing panoramas, as shown in our experiments.

TABLE 1
Comparison of Our Proposed Algorithms

Properties	Cylinder Sweep Algorithm	Locally Linear EPI Algorithm
Derived from conventional stereo	Plane Sweep Stereo	Multi-baseline Stereo
Applicable imaging device	Both Swing and Concentric Panoramas	Only Swing Panoramas
Horizontal epipolar approximation	Optional for efficiency	Required
Usage of panoramas	Arbitrary	From nearby columns
Typical number of panoramas	Several	Tens to hundreds
Dense depth map inference	Cylinder sweep and SSD	Two-pass Tensor Voting
Typical running time†	A few minutes	Half an hour

† The inputs to our experiments are multiperspective panoramas with an image size of about 2000×300 with 100 levels of depth. The output is a 360° field of view dense depth map. A Pentium III 550MHz PC was used to perform our experiments.

8 Conclusions

In this paper, we have introduced a novel representation, multiperspective panoramas, that efficiently captures the parallax available in a scene or environment. Multiperspective panoramas, either concentric panoramas or swing panoramas, are constructed by resampling and rebinning perspective images from one or more off-center rotating cameras.

Multiperspective panoramas are ideally suited for stereo reconstruction of 3D scenes. Instead of using many original images, only several rebinned multiperspective panoramas need to be used. Unlike a collection of single-perspective panoramas taken from different locations, there are no preferred directions or areas where the matching fails because the disparity vanishes.

We have also shown, both analytically and experimentally, that the epipolar geometry of multiperspective panoramas is often well approximated by a traditional horizontal epipolar geometry. This property allows us to apply traditional multibaseline and multiview stereo algorithms without any modification. We have shown experimentally that good stereo reconstructions can be obtained from such panoramas and that the original parallax in the scene can be recreated from just one panorama and one panoramic depth map. It is also possible to extrapolate novel views from original panoramas and the recovered depth map.

We have developed two novel reconstruction algorithms. One uses a cylinder sweep algorithm. The other uses multibaseline stereo on approximate EPIs. The cylinder sweep algorithm only requires us to translate and scale the panoramas during the matching phase, while 1D matching is sufficient for the other algorithm. We have analyzed and compared the two methods.

The novel representation of concentric multiperspective panoramas and good quality stereo reconstruction from such panoramas suggest a powerful new way of modeling and rendering a large environment. Instead of using a single global model for the whole environment, we envision using a collection of local models for overlapping subregions of the environment. Each subregion is represented by a small set of multiperspective panoramas and their associated depth maps. At each subregion, the user is free to "look around and move a little" inside a circular region using the local panoramas and depth maps. As the user moves from subregion to another, a different local model is activated. We are developing novel rendering algorithms based on this representation that will bring the "third dimension" back into panoramic photography and the viewing and exploration of virtual environments.

Appendix A
Tensor Voting

Tensor voting [19] uses a second order symmetric *tensor* for data representation and a *voting* methodology for data communication. Each input site is encoded as a tensor, propagating preferred direction in a neighborhood. In essence, we collect a large number of tensor votes at each input point in order to attenuate the effect of outlier noise and analyze their direction consistency simultaneously. A high agreement in the normal direction indicates a high surface saliency. A high *dis*agreement in the normal direction indicates a surface orientation discontinuity. If only a small number of inconsistent votes are received, the point is an outlier. We now introduce the terminology used in this appendix.

A.1 Representation as Tensors

A point in the 3D space can assume one of the following: a surface patch, a discontinuity, or an outlier. A point on a smooth surface is very certain about its surface normal orientation (or stick tensor) while at a point junction at which surfaces intersect has absolute orientation uncertainty (indicated by a ball tensor). A second order symmetric tensor in 3D is used to represent this continuum. This tensor representation can be visualized as an ellipsoid (Fig. 16). To describe it, we use an eigensystem with three unit eigenvectors $\hat{V}_{max}$, $\hat{V}_{mid}$, and $\hat{V}_{min}$ and three eigenvalues $\lambda_{max} \geq \lambda_{mid} \geq \lambda_{min}$. An ellipsoid can thus be expressed as: $(\lambda_{max} - \lambda_{mid})\mathbf{S} + (\lambda_{mid} - \lambda_{min})\mathbf{P} + \lambda_{min}\mathbf{B}$, where $\mathbf{S} = \hat{V}_{max}\hat{V}_{max}^T$ defines a stick tensor, $\mathbf{P} = \hat{V}_{max}\hat{V}_{max}^T + \hat{V}_{mid}\hat{V}_{mid}^T$ defines a plate tensor, and $\mathbf{B} = \hat{V}_{max}\hat{V}_{max}^T + \hat{V}_{mid}\hat{V}_{mid}^T + \hat{V}_{min}\hat{V}_{min}^T$ gives a ball tensor (Fig. 16). These tensors define

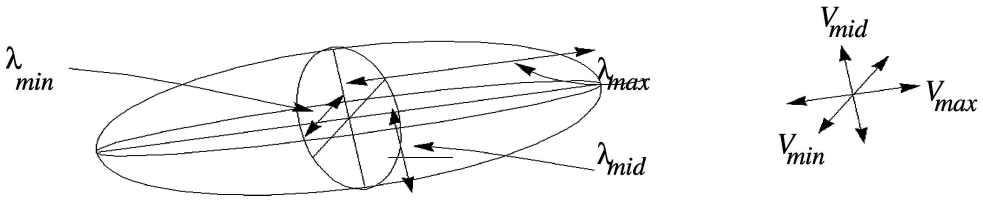


Fig. 16. A second order symmetric tensor in 3D.

the three basis tensors for any ellipsoid. $\lambda_{max} - \lambda_{mid}$ is used to indicate *surface saliency* [19]. See Fig. 16.

A.2 Tensor Decomposition

The eigenvectors encode orientation (un)certainties: Stick tensor, indicating certainty along a single direction, encodes surface normal orientation. Uncertainties are abstracted by two other tensors: Curve junction is produced from two intersecting surfaces, where the uncertainty in orientation only spans a single plane perpendicular to the tangent of the junction curve, and is thus described by a plate tensor. At point junctions where more than two intersecting surfaces are present, a ball tensor is used since there is no preferred orientation. The eigenvalues encode the magnitudes of orientation (un)certainties, or the size of the ellipsoid.

In this paper, we define $\lambda_{max} - \lambda_{mid}$ to be our surface saliency at each tensor, with $\hat{V}_{max}$ indicating the normal direction.

We perform eigensystem decomposition and derive the following geometric interpretation to measure feature saliencies, with the associated directions:

- *Surface-ness*: Surface saliency is measured by $\lambda_{max} - \lambda_{mid}$, with $\hat{V}_{max}$ indicating the normal direction.
- *Curve-ness*: Curve saliency is measured by $\lambda_{mid} - \lambda_{min}$, with $\hat{V}_{min}$ indicating the tangent direction.
- *Junction-ness*: Junction saliency is measured by λ_{min}. There is no preferred direction associated with point junctions.

A.3 Data Communication by Voting

First, we encode the input into a set of *default tensors*: If the voxel contains an input point, we associate it with a 3D default ball tensor, having all $\lambda_{max} = \lambda_{mid} = \lambda_{min}$, and $\hat{V}_{max} = [1\ 0\ 0]^T$, $\hat{V}_{mid} = [0\ 1\ 0]^T$, and $\hat{V}_{min} = [0\ 0\ 1]^T$. Otherwise, if the voxel does not contain an input point, it is associated with a *zero tensor* (i.e., zero eigenvalues and zero eigenvectors). These input tensors *cast* votes or are made to align (by translation and rotation) with predefined *voting fields*. In particular, we describe the *ball voting field* here, which is used for depth estimation in this paper. One slice on the x-y plane of this 3D tensor field is shown in Fig. 17. It is a dense isotropic field without any orientation preference, which propagates all possible directions in a neighborhood with equal likelihood. The neighborhood size is determined by the scale of analysis or, equivalently, the size of the voting field.

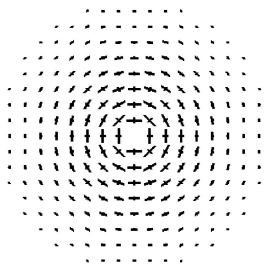


Fig. 17. One slice of the 3D ball voting field, which propagates all directions in equal likelihood in a neighborhood.

When each input point has cast its tensor vote to its neighboring voxels, by aligning with the ball voting fields (or *votes with the ball voting field*), each voxel in the volume receives a set of tensor votes. These votes are collected, using *tensor addition*, as a 3×3 covariance matrix of second order moment collection of all the vote contribution. Upon eigensystem analysis, we obtain a generic saliency tensor or ellipsoid, encoding preferred normal orientation and discontinuity information by the stick and the ball tensors, respectively.

Acknowledgments

The authors would like to thank Dr. Sing Bing Kang for his many constructive suggestions. They would also like to thank Gang Xu, Tao Feng, and Zhouchen Lin for many helpful discussions while Y. Lin was at Microsoft Research, Asia. This work is supported by the Research Grant Council of the Hong Kong Special Administrative Region, China under grant number HKUST6193/02E.

References

[1] S. Baker, R. Szeliski, and P. Anandan, "A Layered Approach to Stereo Reconstruction," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 434-441, June 1998.

[2] R. Benosman and J. Devars, "Panoramic Stereovision Vensor," *Panoramic Vision: Sensors, Theory, and Applications,* pp. 161-168, New York: Springer, 2001.

[3] S. Birchfield and C. Tomasi, "A Pixel Dissimilarity Measure that Is Insensitive to Image Sampling," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 4, pp. 401-406, Apr. 1998.

[4] R.C. Bolles, H.H. Baker, and D. Marimont, "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion," *Int'l J. Computer Vision,* vol. 1, pp. 7-55, 1987.

[5] R.T. Collins, "A Space-Sweep Approach to True Multi-Image Matching," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 358-363, June 1996.

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

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

[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, Feb. 1992.

[9] D. Jelinek and C.J. Taylor, "View Synthesis with Occlusion Reasoning Using Quasi-Sparse Feature Correspondences," *Proc. Seventh European Conf. Computer Vision,* vol. 2, pp. 463-478, May 2002.
[10] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A Stereo Machine for Video-Rate Dense Depth Mapping and Its New Applications," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 196-202, June 1996.
[11] S.B. Kang and R. Szeliski, "3-D Scene Data Recovery Using Omnidirectional Multibaseline Stereo," *Int'l J. Computer Vision,* vol. 25, no. 2, pp. 167-183, Nov. 1997.
[12] S.B. Kang, R. Szeliski, and J. Chai, "Handling Occlusions in Dense Multiview Stereo," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 103-110, Dec. 2001.
[13] R. Koch, M. Pollefeys, and L. Van Gool, "Multi Viewpoint Stereo from Uncalibrated Video Sequences," *Proc. Fifth European Conf. Computer Vision,* vol. 1, pp. 55-71, June 1998.
[14] V. Kolmogorov and R. Zabih, "Multi-Camera Scene Reconstruction via Graph Cuts," *Proc. Seventh European Conf. Computer Vision,* vol. 3, pp. 82-96, May 2002.
[15] K.N. Kutulakos and S.M. Seitz, "A Theory of Shape by Space Carving," *Proc. Seventh Int'l Conf. Computer Vision,* pp. 307-314, Sept. 1999.
[16] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proc. ACM SIGGRAPH '96,* pp. 31-42, 1996.
[17] Y. Li, C.-K. Tang, and H.-Y. Shum, "Efficient Dense Depth Estimation from Dense Multiperspective Panoramas," *Proc. Eighth Int'l Conf. Computer Vision,* pp. 119-126, 2001.
[18] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System," *Proc. ACM SIGGRAPH '95,* pp. 39-46, 1995.
[19] G. Medioni, M. Lee, and C. Tang, *A Computational Framework for Feature Extraction and Segmentation.* Amsterdam: Elsevier Science, 2000.
[20] M. Okutomi and T. Kanade, "A Multiple Baseline Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 4, pp. 353-363, Apr. 1993.
[21] T. Pajdla, "Stereo with Oblique Cameras," *Int'l J. Computer Vision,* vol. 47, nos. 1-3, pp. 161-170, Apr. 2002.
[22] 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.
[23] S. Peleg and J. Herman, "Panoramic Mosaics by Manifold Projection," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 338-343, June 1997.
[24] P. Rademacher and G. Bishop, "Multiple-Center-of-Projection Images," *Proc. ACM SIGGRAPH '98,* pp. 199-206, July 1998.
[25] D. Scharstein and R. Szeliski, "Stereo Matching with Nonlinear Diffusion," *Int'l J. Computer Vision,* vol. 28, no. 2, pp. 155-174, July 1998.
[26] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int'l J. Computer Vision,* vol. 47, no. 1, pp. 7-42, May 2002.
[27] S. Seitz and C. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring," *Int'l J. Computer Vision,* vol. 25, no. 3, Nov. 1999.
[28] S. Seitz and J. Kim, "The Space of All Stereo Images," *Int'l J. Computer Vision,* vol. 48, no. 1, pp. 21-38, June 2002.
[29] H. Shum, A. Kalai, and S. Seitz, "Omnivergent Stereo," *Proc. Int'l Conf. Computer Vision,* pp. 22-29, 1999.
[30] H.-Y. Shum and L.-W. He, "Rendering with Concentric Mosaics," *Proc. ACM SIGGRAPH '99,* pp. 299-306, 1999.
[31] H.-Y. Shum and R. Szeliski, "Stereo Reconstruction from Multiperspective Panoramas," *Proc. Seventh Int'l Conf. Computer Vision,* pp. 14-21, 1999.
[32] R. Szeliski and P. Golland, "Stereo Matching with Transparency and Matting," *Int'l J. Computer Vision,* special issue for Marr Prize papers, vol. 32, no. 1, pp. 45-61, Aug. 1999.
[33] R. Szeliski and D. Scharstein, "Symmetric Sub-Pixel Stereo Matching," *Proc. Seventh European Conf. Computer Vision,* vol. 2, pp. 525-540, May 2002.
[34] D.N. Wood, et al.. "Multiperspective Panoramas for Cel Animation," *Proc. ACM SIGGRAPH '97,* pp. 243-250, Aug. 1997.

Yin Li received the BSc degree from the University of Science and Technology of China in 1997 and the MSc degree from the Institute of Automation, Chinese Academy of Science in 2000. He is pursuing the PhD degree in the Computer Science Department, Hong Kong University of Science and Technology. He has been with the Visual Computing Group of Microsoft Asia since 1999. His research interests include image-based modeling and rendering, and interactive computer vision.

Heung-Yeung Shum received the PhD degree in robotics from the School of Computer Science, Carnegie Mellon University in 1996. He worked as a researcher for three years in the vision technology group at Microsoft Research, Redmond, Washington. In 1999, he moved to Microsoft Research Asia, where he is currently a senior researcher and the assistant managing director. His research interests include computer vision, computer graphics, human computer interaction, pattern recognition, statistical learning, and robotics. He serves on the editorial boards of *IEEE Transactions of Circuit and System Video Technology*, the *IEEE Transactions of Pattern Analysis and Machine Intelligence*, and *Graphical Models*. He is a senior member of the IEEE and the general cochair of the 10th International Conference of Computer Vision, Beijing, 2005.

Chi-Keung Tang received the MS and PhD degrees in computer science from the University of Southern California (USC), Los Angeles, in 1999 and 2000, respectively. He has been with the Computer Science Department at the Hong Kong University of Science and Technology since 2000, where he is currently an assistant professor. He is also an adjunct researcher at the Visual Computing Group of Microsoft Research, Asia, working on various exciting research topics in computer vision and graphics. His research interests include low to mid-level vision such as segmentation, correspondence, shape analysis, and vision and graphics topics such as image-based rendering and medical image analysis. He is a member of the IEEE Computer Society.

Richard Szeliski received the PhD degree in computer science from Carnegie Mellon University, Pittsburgh, in 1988. He is a senior researcher in the Interactive Visual Media Group at Microsoft Research, where he is pursuing research in 3D computer vision, video scene analysis, and image-based rendering. His current focus is on constructing photorealistic 3D scene models from multiple images and video. He joined Microsoft Research in 1995. Prior to Microsoft, he worked at Bell-Northern Research, Schlumberger Palo Alto Research, the Artificial Intelligence Center of SRI International, and the Cambridge Research Lab of Digital Equipment Corporation. He has published more than 100 research papers in computer vision, computer graphics, medical imaging, neural nets, and parallel numerical algorithms, as well as the book *Bayesian Modeling of Uncertainty in Low-Level Vision*. He is on the editorial board of the *International Journal of Computer Vision* and has served as program chair for ICCV 2001, organizer of the ICCV '99 Workshop on Vision Algorithms, cochair of the SPIE Conferences on Geometric Methods in Computer Vision, and associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He is a senior member of the IEEE.