A Framework for 3D Pushbroom Imaging *

CUCS-002-03

Naoyuki Ichimura† and Shree K. Nayar‡

†Information Technology Research Institute,
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Ibaraki 305-8568, Japan
Email: naoyuki.ichimura@aist.go.jp

‡Department of Computer Science,
Columbia University
New York, New York 10027, USA
Email: nayar@cs.columbia.edu

*This work was supported in parts by the JSPS Overseas Research Fellowship.

Abstract

Pushbroom cameras produce one-dimensional images of a scene with high resolution at a high frame-rate. As a result, they provide superior data compared to conventional two-dimensional cameras in cases where the scene of interest can be temporally scanned. In this paper, we consider the problem of recovering the structure of a scene using a set of pushbroom cameras. Although pushbroom cameras have been used to recover scene structure in the past, the algorithms for recovery were developed separately for different camera motions such as translation and rotation. In this paper, we present a general framework of structure recovery for pushbroom cameras with 6 degree-of-freedom motion. We analyze the translation and rotation cases using our framework and demonstrate that several previous results are really special cases of our result. Using this framework, we also show that three or more pushbroom cameras can be used to compute scene structure as well as motion of translation or rotation. We conclude with a set of experiments that demonstrate the use of pushbroom imaging to recover structure from unknown motion.

Contents

1 Introduction

A pushbroom camera, also known as a line scanner/sensor, is an imaging system with a one-dimensional (1D) array of pixels [1]. The use of a 1D detector gives a pushbroom camera several significant advantages over conventional cameras that use two-dimensional (2D) detectors: the pushbroom camera produces data with high resolution and has a high frame-rate. For example, recent off-the-shelf pushbroom cameras have a resolution of 2,048 pixels and a frame-rate of 20 KHz [2]. For these reasons, pushbroom cameras have begun to emerge as a popular alternative to 2D cameras, especially in the fields of remote sensing and visual inspection, where it is convenient to scan the scene of interest [3, 4].

Figure 1(a) shows a scene being scanned by a moving pushbroom camera. At each instant in time, the camera produces a 1D image which represents the brightness of the scene points that intersect the "view plane" of the camera. By concatenating consecutive 1D images, we obtain a 2D image called a pushbroom panorama [5], such as the one in Fig. 1(b). Note that the columns of this image include just spatial information, i.e., a

---

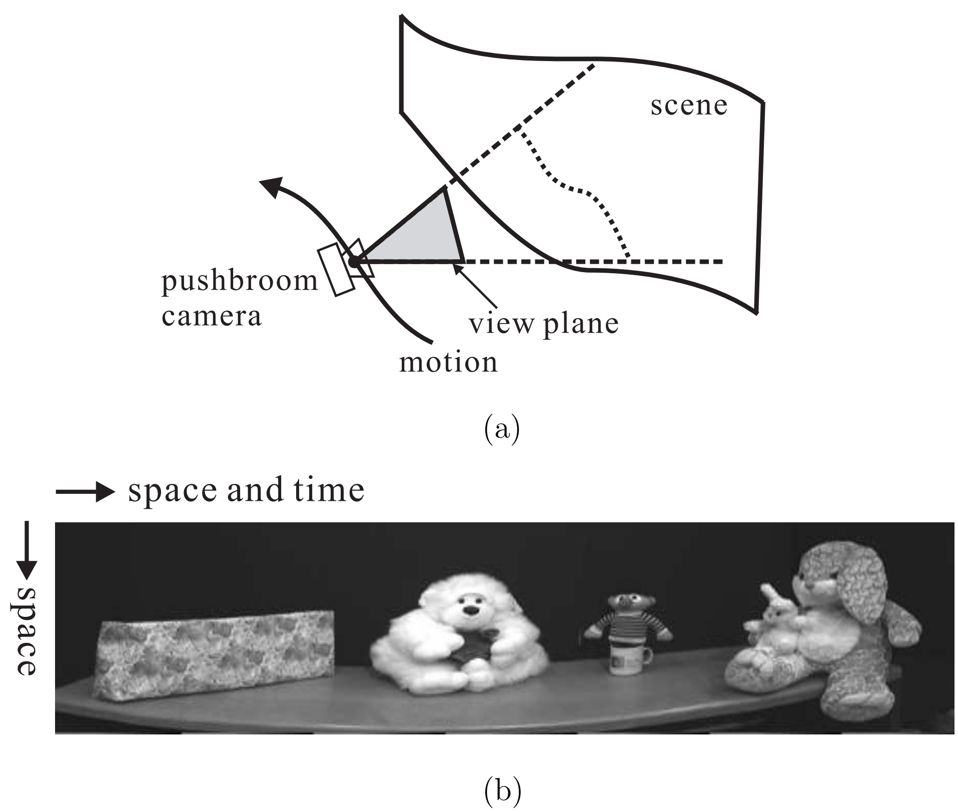


Figure 1: The acquisition of a pushbroom panorama. (a) A moving pushbroom camera sweeps a scene with the "view plane". A 1D image, which represents the brightness of the scene points that intersect the view plane, is produced at each instant of time. (b) An example of a pushbroom panorama created by concatenating the 1D images. The columns of this image include spatial information, while the rows include both spatial and temporal information. This embedding of spatial and temporal information can be exploited to recover *both* scene structure and motion.

1D perspective projection of the scene, while the rows include both spatial and temporal information, i.e., projections of scene points captured at different instants of time. In short, a moving pushbroom camera encodes both spatial and temporal information within a single 2D image. As we shall see, this feature of a pushbroom camera makes it an interesting device in the context of structure and motion recovery.

This paper presents a general framework for the use of pushbroom cameras for recovering the structure of a scene. Before we present this framework, a quick review of previous work is in order. Recently, several investigators have explored structure recovery using pushbroom panoramas. For demonstration purposes, pushbroom cameras were emulated with a conventional 2D camera by extracting two or more lines of pixels. Chai and Shum [6] proposed stereo reconstruction based on parallel projections realized by a camera with 1D and 2D translation. Zhu et. al. [7] considered parallel-perspective stereo mosaics obtained from a camera with 3D translation. Ishiguro [8] et. al., Shum et. al. [9] and Perr et. al. [10] presented approaches using pushbroom panoramas captured by a rotating camera to achieve scene reconstruction. Benosman et al. [11] constructed a stereo system using two rotating pushbroom cameras. It is important to note that all of these previous works are restricted in two ways: (a) they consider only specific motions of the pushbroom cameras, i.e., translation and rotation; (b) all of them require the motion of the pushbroom cameras to be known. In previous work, algorithms for recovering scene structure were developed *separately* for translation and rotation; there exists no common framework which subsumes the various configurations in which pushbroom cameras can be used for structure recovery. In addition, previous methods rely on prior knowledge of camera motion. This information has been obtained using a variety of devices including gyroscopes [3], GPS and INS [7], and optical encoders [8, 9, 10, 11].

The goal of this paper is to explore the conditions under which a set of pushbroom cameras can be used to recover structure. First, we present a general framework for structure recovery using pushbroom cameras. The key contribution here is a general geometric constraint, called the *view plane equation*. We have developed an algorithm based on this constraint which is general in that it can handle 6 degree-of-freedom (DOF) motion of the camera. As special cases, we analyze two specific camera motions, i.e., translation and rotation and show that several previous methods for structure recovery are really special cases of our framework. Next, we apply our framework to the problem of structure from *unknown* translation or rotation. We show that, in these cases, three (or more) pushbroom cameras can be used to recover *both* scene structure and motion. We conclude the paper with a set of experiments that demonstrates the practical utility of our results.

2 General Framework for Structure Recovery

In this section, we derive geometric constraints imposed by a pushbroom camera with 6 DOF motion. Then, we present the fundamental equations for recovering scene structure based on the geometric constraints.

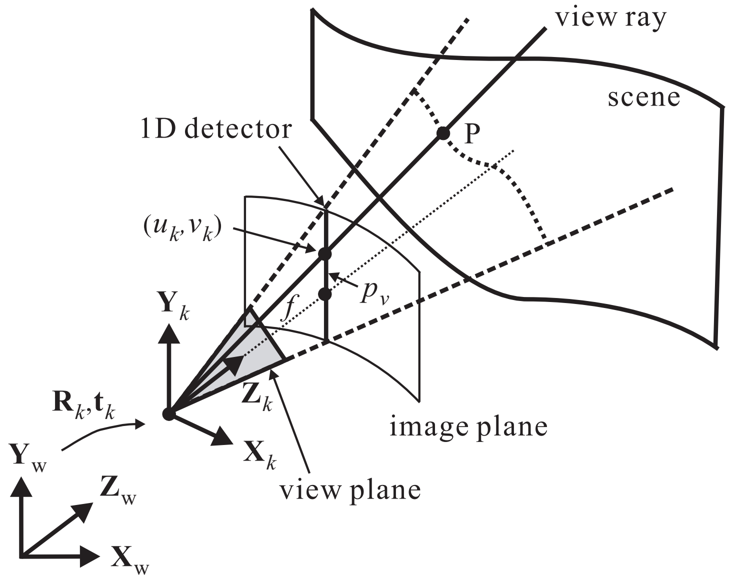


Figure 2: Imaging geometry of a pushbroom camera. The world coordinate system is denoted by the $\boldsymbol{X}_{\rm w}$, $\boldsymbol{Y}_{\rm w}$ and $\boldsymbol{Z}_{\rm w}$ axes. The camera coordinate system at the instant of time kT_s is denoted by the $\boldsymbol{X}_k$, $\boldsymbol{Y}_k$ and $\boldsymbol{Z}_k$ axes. The relationship between the world and camera coordinate systems is given by the rotation matrix $\boldsymbol{R}_k$ and translation vector $\boldsymbol{t}_k$. The world and camera coordinates of a 3D point P are $\boldsymbol{p}_w = ({\rm x_w}, {\rm y_w}, {\rm z_w})^t$ and $\boldsymbol{p}_k = ({\rm x}_k, {\rm y}_k, {\rm z}_k)^t$, respectively. The 1D detector (and hence the view plane) lies on the $\boldsymbol{Y}_k$–$\boldsymbol{Z}_k$ plane. A pushbroom panorama with coordinates (u_k, v_k) is created by concatenating 1D images.

2.1 Imaging Geometry

Figure 2 depicts the imaging geometry of a pushbroom camera. The world coordinate system is denoted by $\boldsymbol{X}_{\rm w}$, $\boldsymbol{Y}_{\rm w}$ and $\boldsymbol{Z}_{\rm w}$. The camera captures a scene at discrete instants of time represented by kT_s, where k ($k = 0, 1, 2, \ldots$) is the time index and T_s is the sampling interval. The camera motion at the time instant kT_s is represented by the rotation matrix $\boldsymbol{R}_k$ and translation vector $\boldsymbol{t}_k$, which define the camera coordinate system given by $\boldsymbol{X}_k$, $\boldsymbol{Y}_k$ and $\boldsymbol{Z}_k$.

The 1D detector of the camera lies on the $\boldsymbol{Y}_k$–$\boldsymbol{Z}_k$ plane and it produces a 1D image at every time instant. By concatenating successive 1D images, we can produce a pushbroom panorama with coordinates (u_k, v_k). The u coordinate represents the discrete time. That is:

$$u_k = kT_s \,. \tag{1}$$

The v coordinate represents spatial information. Consider a 3D scene point P whose camera coordinates are $\boldsymbol{p}_k = ({\rm x}_k, {\rm y}_k, {\rm z}_k)^t$. Its v coordinate is determined by a 1D perspective projection as:

$$v_k = \frac{f {\rm y}_k}{{\rm z}_k} + p_v \,, \tag{2}$$

where, f and p_v are the focal length and the image center, respectively.

2.2 Deriving Geometric Constraints

We denote the world coordinates of the 3D point P as $\boldsymbol{p}_w = (\mathrm{x_w}, \mathrm{y_w}, \mathrm{z_w})^t$. We know that the camera and world coordinates are related to each other as:

$$\boldsymbol{p}_k = (\boldsymbol{R}_k \,|\, -\boldsymbol{R}_k \boldsymbol{t}_k) \begin{pmatrix} \boldsymbol{p}_w \\ 1 \end{pmatrix} , \tag{3}$$

where,

$$\boldsymbol{R}_k = (\boldsymbol{i}_k, \boldsymbol{j}_k, \boldsymbol{k}_k)^t = \boldsymbol{R}_x(\theta_k)\,\boldsymbol{R}_y(\phi_k)\,\boldsymbol{R}_z(\psi_k) , \tag{4}$$

$$\boldsymbol{t}_k = (t_{xk}, t_{yk}, t_{zk})^t . \tag{5}$$

The rows of the rotation matrix, $\boldsymbol{i}_k$, $\boldsymbol{j}_k$ and $\boldsymbol{k}_k$, define the directions of the axes of the camera coordinate system, $\boldsymbol{X}_k$, $\boldsymbol{Y}_k$ and $\boldsymbol{Z}_k$. The rotation matrices $\boldsymbol{R}_x(\theta_k)$, $\boldsymbol{R}_y(\phi_k)$ and $\boldsymbol{R}_z(\psi_k)$ represent rotations of θ_k, ϕ_k and ψ_k about the $\boldsymbol{X}_k$, $\boldsymbol{Y}_k$ and $\boldsymbol{Z}_k$ axes, respectively. The detailed expressions for the rotation matrices are given in Appendix A. The *orientation* and *position* of the camera are represented by the rotation angles and translation vector, respectively.

We can express the camera coordinate x_k by expanding Eq.(3) as follows:

$$\mathrm{x}_k = \boldsymbol{i}_k \cdot \boldsymbol{p}_w - \boldsymbol{i}_k \cdot \boldsymbol{t}_k , \tag{6}$$

where,

$$\boldsymbol{i}_k = (\cos\phi_k \cos\psi_k, -\cos\phi_k \sin\psi_k, \sin\phi_k)^t . \tag{7}$$

Note that, since the view plane (1D detector) lies on the $\boldsymbol{Y}_k$–$\boldsymbol{Z}_k$ plane, we have $\mathrm{x}_k = 0$. Therefore, we have:

$$\boldsymbol{i}_k \cdot \boldsymbol{p}_w = \boldsymbol{i}_k \cdot \boldsymbol{t}_k . \tag{8}$$

The above expression which represents the view plane passing through the 3D point is called the *view plane equation*. It imposes a geometric constraint on the world coordinates $\boldsymbol{p}_w$ and thus it is useful in recovering scene structure.

Another constraint is obtained from the perspective projection of the 3D point to the 1D detector given by Eq.(2). Using y_k and z_k from Eq.(3), we have:

$$\boldsymbol{r}_k \cdot \boldsymbol{p}_w = \boldsymbol{r}_k \cdot \boldsymbol{t}_k , \tag{9}$$

where,

$$\boldsymbol{r}_k = (v_k - p_v)\,\boldsymbol{k}_k - f\boldsymbol{j}_k , \tag{10}$$

$$\boldsymbol{j}_k = (\cos\theta_k \sin\psi_k + \sin\theta_k \sin\phi_k \cos\psi_k, \tag{11}$$

$$\cos\theta_k \cos\psi_k - \sin\theta_k \sin\phi_k \sin\psi_k, -\sin\theta_k \cos\phi_k) , \tag{12}$$

$$\boldsymbol{k}_k = (\sin\theta_k \sin\psi_k - \cos\theta_k \sin\phi_k \cos\psi_k, \tag{13}$$

$$\sin\theta_k \cos\psi_k + \cos\theta_k \sin\phi_k \sin\psi_k, \cos\theta_k \cos\phi_k) . \tag{14}$$

In summary, we have two geometric constraints (Eqs. (8) and (9)) on the world coordinates of a scene point from one pushbroom camera.

2.3 Recovering Structure

We need three constraints to recover the three coordinates $\boldsymbol{p}_w = (\mathrm{x_w}, \mathrm{y_w}, \mathrm{z_w})^t$ of a scene point. As one camera yields only two constraints, we need one more camera. The view plane equations for the two cameras can be written as:

$$\boldsymbol{i}_{k1} \cdot \boldsymbol{p}_w = \boldsymbol{i}_{k1} \cdot \boldsymbol{t}_{k1} \text{ and } \boldsymbol{i}_{k2} \cdot \boldsymbol{p}_w = \boldsymbol{i}_{k2} \cdot \boldsymbol{t}_{k2}\,, \tag{15}$$

where, the time indices k_1 and k_2 represent the instants at which the cameras observe the 3D point. The constraints obtained from the 1D projections are:

$$\boldsymbol{r}_{k1} \cdot \boldsymbol{p}_w = \boldsymbol{r}_{k1} \cdot \boldsymbol{t}_{k1} \text{ and } \boldsymbol{r}_{k2} \cdot \boldsymbol{p}_w = \boldsymbol{r}_{k2} \cdot \boldsymbol{t}_{k2}\,. \tag{16}$$

Using the above expressions, we can recover the coordinates of the 3D point, because we have four constraints and only three unknowns. For example, a matrix equation which has the following form can be constructed from Eq.(15) and the first expression in Eq.(16):

$$\boldsymbol{A}\boldsymbol{p}_w = \boldsymbol{b} \tag{17}$$

where, $\boldsymbol{A} = (\boldsymbol{i}_{k1}, \boldsymbol{i}_{k2}, \boldsymbol{r}_{k1},)^t$ is a 3×3 matrix and $\boldsymbol{b} = (\boldsymbol{i}_{k1} \cdot \boldsymbol{t}_{k1}, \boldsymbol{i}_{k2} \cdot \boldsymbol{t}_{k2}, \boldsymbol{r}_{k1} \cdot \boldsymbol{t}_{k1})^t$ is a 3×1 vector. If the internal parameters and the motion of the cameras are known, then the elements of $\boldsymbol{A}$ and $\boldsymbol{b}$ can be computed and the coordinates $(x_w, y_w, z_w)^t$ can be found.

The matrix $\boldsymbol{A}$ must satisfy Rank$\boldsymbol{A}$=3 to have a solution. This means that the two pushbroom cameras must have different motions*. When the camera motions are identical then we have a degeneracy as Rank$\boldsymbol{A}$ <3 in that case.

If we have more cameras, we simply add more geometric constraints (represented by Eqs.(8) and (9)) and use the least mean square method to recover structure.

The structure recovery described in this section is general in that it can handle 6 DOF camera motion and multiple cameras. In the subsequent sections, we will demonstrate the generality of this result by analyzing specific camera motions, such as translation and rotation.

3 Cases of Translation and Rotation

We consider two specific camera motions, translation and rotation, as special cases of our framework. So far, the analysis of these special cases has been done separately in several previous works [6, 7, 8, 9, 10, 11]. We show that several equations for structure recovery used in the past can be easily derived from our framework.

3.1 Translating Pushbroom Cameras

Figure 3 shows pushbroom cameras with 3D translation. As the orientations of the cameras are time independent (fixed), the camera coordinate system at time k is defined

*The cameras may be rigidly attached to each other but, in that case, they must have different orientations; this naturally results in different motions.

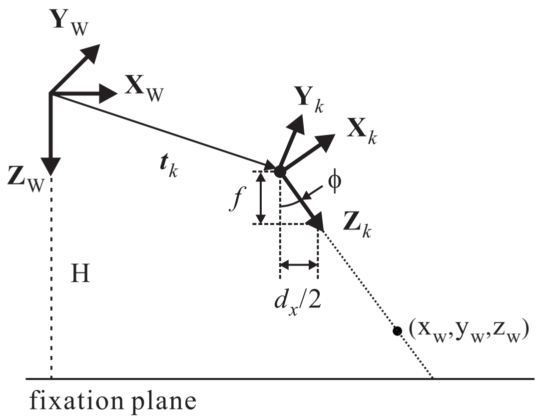


Figure 3: The configuration of a pushbroom camera with 3D translation. The orientation (rotation angles) of the camera is fixed and thus the time dependent motion is only the translation $\boldsymbol{t}_k$.

---

by:

$$\boldsymbol{R}_k = \boldsymbol{R}_x(\theta)\,\boldsymbol{R}_y(\phi)\,\boldsymbol{R}_z(\psi)\ , \tag{18}$$
$$\boldsymbol{t}_k = (t_{xk}, t_{yk}, t_{zk})^t\ . \tag{19}$$

We can reconstruct scene structure using Eqs.(15) and (16) if we have two or more pushbroom cameras with the different motions represented by Eqs.(18) and (19).

We now show that this result can be used to describe previous work on parallel perspective stereo mosaics [6, 7]. Assume that we have two cameras with rotation angles $(0, \phi_1, 0)$ and $(0, \phi_2, 0)$ which satisfy $\phi_1 = -\phi_2$ and $|\phi_1| = |\phi_2| = \phi$. From Eq.(8), we have the following view plane equation of the first camera:

$$\mathrm{x_w}\cos\phi_1 + \mathrm{z_w}\sin\phi_1 = t_{xk1}\cos\phi_1 + t_{zk1}\sin\phi_1\ . \tag{20}$$

The view plane equation for the second camera can be obtained by replacing ϕ_1, t_{xk1} and t_{zk1} by ϕ_2, t_{xk2} and t_{zk2}, respectively. The depth $\mathrm{z_w}$ can be calculated from the intersection of the two view planes as:

$$\mathrm{z_w} = \frac{t_{xk2} - t_{xk1}}{2\tan\phi} + \frac{t_{zk1} + t_{zk2}}{2}\ . \tag{21}$$

In terms of the notation used in [7], the above expression can be written as:

$$\mathrm{z_w} = H\left(1 + \frac{\Delta x}{d_x}\right) + \frac{t_{zk1} + t_{zk2}}{2}\ , \tag{22}$$

where, $\Delta x = x2 - x1$, $x1 = \frac{f}{H}t_{xk1} + \frac{d_x}{2}$, $x2 = \frac{f}{H}t_{xk2} - \frac{d_x}{2}$, f is the focal length, $\tan\phi = \frac{(d_x/2)}{f}$ and H is the height of the fixation plane [7] (see Fig. 3). 1D parallel stereo [6] is a special case of the 3D case, where $t_{yk1} = t_{zk1} = 0$ and $t_{yk2} = t_{zk2} = 0$. Thus, the first term of Eq.(21) is equivalent to the depth equation for 1D parallel stereo given in [6].

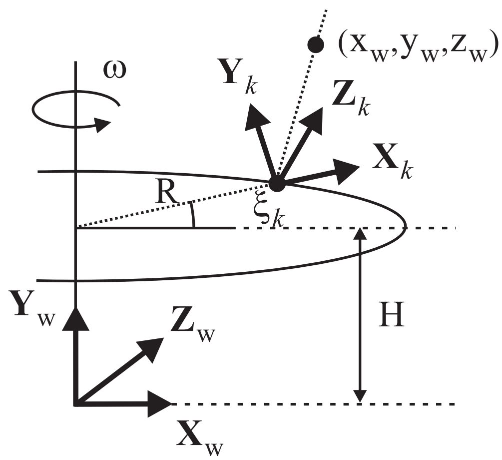


Figure 4: The configuration of a pushbroom camera with rotation. The imaging system is rotated about the $\boldsymbol{Y}_{\mathrm{w}}$ axis with the angular velocity ω. The elevation of the camera is H. The radius of the rotation circle is R. The rotation angle of the camera at time k is ξ_k.

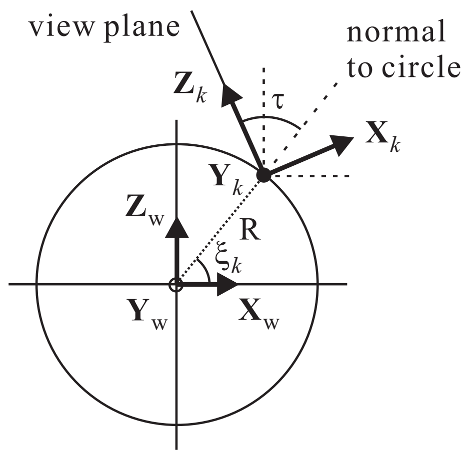


Figure 5: Definition of tilt angle τ. The angle between the normal to the circle and the $\boldsymbol{Y}_k$–$\boldsymbol{Z}_k$ plane remains τ, regardless of the rotation angle ξ_k of the camera.

3.2 Rotating Pushbroom Cameras

Figure 4 shows a pushbroom camera rotating about the $\boldsymbol{Y}_{\mathrm{w}}$ axis with the angular velocity ω. The radius of the circle, i.e. the distance between the center of rotation and the origin of the camera coordinate system, is R. The elevation of the camera is H. The rotation angle ξ_k of the camera at time k is given by:

$$\xi_k = \omega k T_s + \xi_0 \,. \tag{23}$$

where, ξ_0 is the *initial angle* which corresponds to the initial position of the camera. The camera coordinate system at time k is defined by the following motion:

$$\begin{aligned} \boldsymbol{R}_k &= \boldsymbol{R}_x(\theta)\,\boldsymbol{R}_y(\phi_k)\,\boldsymbol{R}_z(\psi)\;, & (24)\\ \boldsymbol{t}_k &= (R\cos\xi_k, H, R\sin\xi_k)^t\;. & (25) \end{aligned}$$

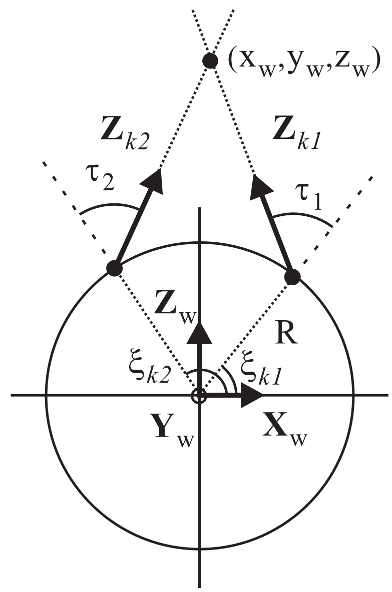


Figure 6: The camera configuration used by omnidirectional stereo [8] and concentric mosaics [9].

Note that only ϕ_k is time dependent among the three rotation angles.

Figure 5 shows the definition of the *tilt angle* τ which is the angle between the normal to the circle and the direction of the view plane. Since the view plane of the camera is restricted to the $\boldsymbol{Y}_k$–$\boldsymbol{Z}_k$ plane of the camera coordinate system, the rotation of the imaging system about the $\boldsymbol{Y}_{\mathrm{w}}$ axis only changes the angle ϕ_k in Eq. (24). Therefore, ϕ_k can be determined from the tilt angle τ and the rotation of the camera ξ_k as:

$$\phi_k = \xi_k - (\pi/2) + \tau \,. \tag{26}$$

We can reconstruct scene structure using two or more pushbroom cameras with different motions given by Eqs.(24) and (25).

The above result includes previous works such as omnidirectional stereo and concentric mosaics [8, 9, 10, 11]. To show this, we derive the equations for structure recovery and epipolar geometry used in these previous works using our view plane equation.

Figure 6 shows the camera configuration used in omnidirectional stereo and concentric mosaics [8, 9]. Let the two cameras have the following rotation angles and translation vectors: $(0, \phi_{k1}, 0)$, $(R\cos\xi_{k1}, H, R\sin\xi_{k1})$, $(0, \phi_{k2}, 0)$ and $(R\cos\xi_{k2}, H, R\sin\xi_{k2})$, where, $\phi_{k1} = \xi_{k1} - (\pi/2) + \tau_1$ and $\phi_{k2} = \xi_{k2} - (\pi/2) + \tau_2$.

In Fig. 6, assume that $\tau_1 = -\tau_2$ and $|\tau_1| = |\tau_2| = \tau$. The view plane equation for the first camera is:

$$\mathrm{x_w}\cos\phi_{k1} + \mathrm{z_w}\sin\phi_{k1} = R\sin\tau_1 \,, \tag{27}$$

$$\phi_{k1} = \xi_{k1} - (\pi/2) + \tau_1 \,. \tag{28}$$

The view plane equation for the second camera can be obtained by replacing ϕ_{k1}, ξ_{k1} and τ_1 by ϕ_{k2}, ξ_{k2} and τ_2, respectively. From these equations, the depth of a 3D point can be determined as:

$$\mathrm{z_w} = \frac{R \sin \tau}{\sin\left(2\tau - (\xi_{k2} - \xi_{k1})\right) / \left(\cos \phi_{k1} + \cos \phi_{k2}\right)} . \tag{29}$$

Although our notations for orientation and position are different from those used in the previous works, Eq.(29) is essentially equivalent to the depth equation derived in [8]; depth is proportional to the radius and the tilt angle, and it is inversely proportional to the difference between the tilt angle and the rotation angles.

We now derive the vertical scale defined in [9]. The vertical scale is the ratio of the image coordinates v_{k1} and v_{k2} given by Eq.(2). It is used to determine whether corresponding features lie on the same scan line of two panoramas. In Appendix B, we show that the vertical scale can be written as:

$$S_{vs} = \frac{v_{k1}}{v_{k2}} = \frac{\sqrt{1 - \left(\frac{R}{r} \sin \tau_2\right)^2} - \frac{R}{r} \cos \tau_2}{\sqrt{1 - \left(\frac{R}{r} \sin \tau_1\right)^2} - \frac{R}{r} \cos \tau_1} , \tag{30}$$

where, $r = \sqrt{\mathrm{x_w}^2 + \mathrm{z_w}^2}$. In summary, Eqs. (22), (29) and (30) demonstrate that several previous results using translation and rotation are really special cases of our approach to pushbroom imaging.

4 Structure from Unknown Translation or Unknown Rotation

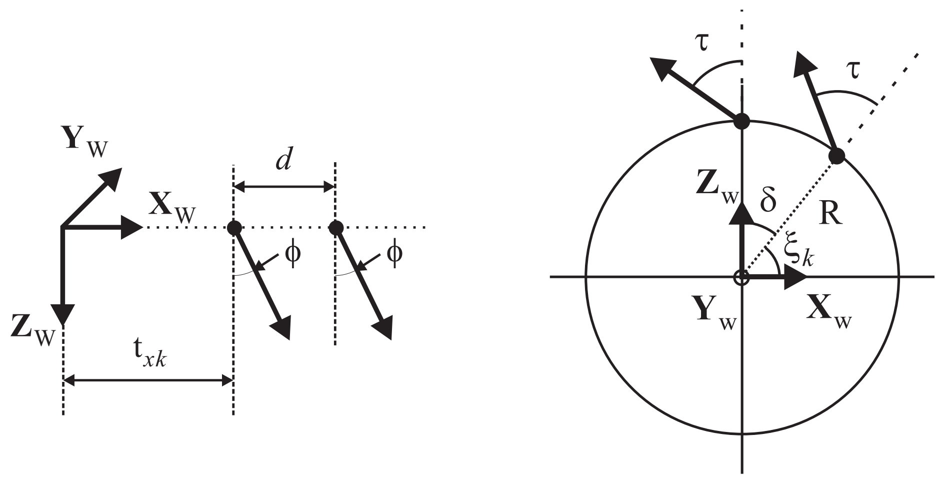
As we discussed in Section 2, an imaging system with two pushbroom cameras is sufficient to compute structure when the motion of the system is known. We now show how three pushbroom cameras are sufficient to compute both structure and motion, when the motion is 1D translation or rotation.

4.1 Recovering Velocity

In the translation and rotation cases, we can classify motion parameters into two groups: time independent and time dependent. In 1D translation, which is a special case ($t_{yk} = t_{zk} = 0$) of Section 3.1, the angles θ, ϕ and ψ are time independent and the translation t_{xk} is time dependent. In the rotation case of Section 3.2, the angles θ, τ and ψ of camera orientation are time independent and the rotation angle ξ_k given by Eq.(23) is time dependent. While we can determine the time independent parameters by off-line calibration, we must recover the time dependent parameters. We show that one additional pushbroom camera is sufficient to achieve this.

In the 1D translation case, the translation t_{xk1} and t_{xk2} of the cameras at the times k_1 and k_2 when the cameras observe the same 3D point are given by:

$$t_{xk1} = v_x k_1 T_s + t_{x01} \ \text{ and } \ t_{xk2} = v_x k_2 T_s + t_{x02} \, , \tag{31}$$



(a) translation (b) rotation

Figure 7: Configurations of two cameras for recovering velocity of (a) translation and (b) rotation. The cameras must have the same motion so that they observe the same 3D scene point from the same position. In both cases (translation and rotation), the velocity of the cameras can be found from the difference of a scene point's observation times, which can be found from feature correspondences between the two acquired pushbroom panoramas and the distance d (for translation) or the angle δ (for rotation) between the two cameras.

---

where, v_x is the velocity of the translation and t_{x01} and t_{x02} are the initial positions of cameras. In the rotation case, the rotation angles ξ_{k1} and ξ_{k2} are given by Eq.(23):

$$\xi_{k1} = \omega k_1 T_s + \xi_{01} \text{ and } \xi_{k2} = \omega k_2 T_s + \xi_{02}, . \tag{32}$$

We need to know the translations and the rotation angles at the times k_1 and k_2 to recover structure. In Eqs.(31) and (32), the initial positions of the cameras, t_{x01}, t_{x02}, ξ_{01} and ξ_{02}, can be obtained from calibration. The sampling interval T_s is known. Note that the indices k_1 and k_2 are column numbers in the two acquired pushbroom panoramas. They can be determined by finding correspondences (feature matching) between the two panoramas. To determine the translations and rotation angles of the cameras, we need only the velocity v_x and angular velocity ω.

Figure 7 shows two cameras with *the same motion and different initial positions* for (a) translation and (b) rotation. That is, the two cameras have the same orientation and are rigidly fixed with respect to each other to have different initial positions. The difference between the initial positions of the cameras is d or δ. As we discussed in Section 2.3, we cannot use these configurations to recover structure since the motions of the two cameras are identical; the rank of $\boldsymbol{A}$ in Eq.(17) is 2 in this case. Interestingly, these configurations can be used to recover the linear or angular velocity.

Since the view plane equations of the two cameras are identical, *they observe the same 3D point from the same position regardless of scene structure*. The difference between the observation time is d/v_x or δ/ω. Therefore, the u coordinates of the projections of the 3D

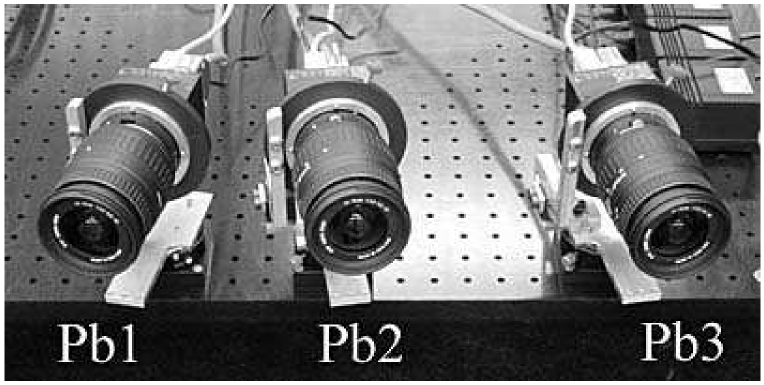


Figure 8: An imaging system with three pushbroom cameras. The orientations of the cameras and the distances between cameras can be adjusted using the positioning stages that the cameras are mounted on. The cameras are synchronized and produce 850 lines of pixels per second and each line has 1024 pixels. The time resolution of this system is very high, 1/850=0.00117[sec].

---

point for the two cameras are:

$$u_{k1} = k_1 T_s, \quad u_{k2} = k_2 T_s = k_1 T_s + \Delta t\,, \tag{33}$$

$$\Delta t = \begin{cases} d/v_x & \text{(translation)} \\ \delta/\omega & \text{(rotation)}\,. \end{cases} \tag{34}$$

Now, we can recover the velocity as:

$$v_x = \frac{d}{\Delta u} \text{ (translation)}\,, \tag{35}$$

$$\omega = \frac{\delta}{\Delta u} \text{ (rotation)}\,, \tag{36}$$

where, $\Delta u = (k_2 - k_1)\, T_s$.

4.2 Structure from Unknown Velocity

In our analysis in Sections 4.1 and 3, we have shown that we can recover both the structure of the scene and the velocity using three or more pushbroom cameras. Two cameras must have the same motion and different initial positions to recover velocity. The other camera(s) must have different motions to recover structure.

5 Experiments

Figure 8 shows the imaging system we have configured with three pushbroom cameras (model L103 from Basler). The orientations of the cameras and the distances between cameras can be adjusted and determined using the positioning stages that the cameras are mounted on. The cameras are synchronized and produce 850 lines of pixels per second and each line has 1024 pixels. That is, the time resolution of this system is 1/850=0.00117[sec].

(a) (b)

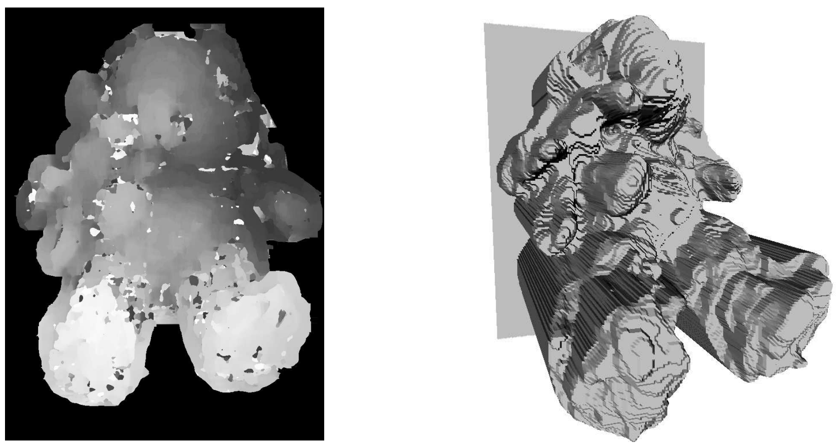
Figure 9: Comparison between the pushbroom panoramas captured by (a) an NTSC camera and (b) a pushbroom camera for an object with the same velocity. The field of views of the two cameras are identical. The former has only 25 columns due to the slow frame rate while the latter has 708 columns. This example shows the clear advantage of using pushbroom cameras for capturing moving objects.

First, we show results of experiments with 1D translation. In the experiments, the cameras Pb1, Pb2 and Pb3 (see Fig. 8) had the orientations $(0, -\phi, 0)$, $(0, \phi, 0)$ and $(0, \phi, 0)$, where $\phi = 7$[deg]. The distances between Pb1 and Pb2, and Pb2 and Pb3 (d in Fig. 7 (a)) were 5 and 8 inches, respectively. To emphasize the importance of the velocity recovery using pushbroom cameras, we moved an object instead of the imaging system. When we move the imaging system, we can use devices for measuring motion such as gyroscopes and optical encoders. We cannot, however, use such devices for moving objects and thus we need to measure velocity only from captured images. Note that, the principle of structure recovery for the moving object case is the same for the moving camera case discussed in Sections 3.1 and 4.1. The object was moved by a robot (AdeptOne Robot [12]).

Figure 9 shows the difference between pushbroom panoramas created by a conventional NTSC camera and a pushbroom camera. The object was moving and we obtained an image with only 25 columns from the NTSC camera while we obtained an image with 708 columns using the pushbroom camera.

Figure 10 shows three pushbroom panoramas of the object generated using our system. The size of each image is 4000×1024[pixel] (the images in Fig. 10 were cropped for display purposes). The first and second images, which were generated by Pb2 and Pb3, were used to recover the velocity of the object. We used an image alignment method to find the velocity. We extracted corners in each image using a corner detector [13] and then calculated the parameters of a translational motion model using corresponding corners. The horizontal translation between the two images shows the time difference Δu in Eq.(35) needed to calculate velocity. Normalized correlation was used to find the correspondences and the RANSAC [14] was applied to remove false matching. The estimated velocity was 434[mm/sec]. This value is exactly the same speed by the robot (ground truth).

Figure 10: Pushbroom panoramas of a moving object produced by the system in Fig. 8. The size of each image is 4000 $\times$ 1024[pixel] (the images were cropped for display purposes). The left and middle panoramas were used to recover velocity, while the middle and right panoramas were used to recover 3D structure.



(a) (b)

Figure 11: Recovered structure (unknown linear velocity). (a) A depth image, where brighter pixels correspond to closer scene points. (b) A VRML model of the moving object as seen from a different viewpoint. This result shows that our method can recover fairly accurate scene structure even when the velocity of the moving object is high and unknown.

This accuracy is due to the very high time resolution of the imaging system, 0.00117[sec]. This result shows the advantage of using temporal information provided by pushbroom panoramas for motion estimation.

Figure 11 shows the recovered structure of the moving object. The structure was computed from the second and third images in Fig. 10 using an area-based stereo matching algorithm that uses normalized correlation. Figure 11 (a) shows a depth image, where brighter pixels correspond to closer scene points. Figure 11 (b) shows a VRML model of the moving object. The depth image and VRML model demonstrate that our method can recover scene structure from unknown velocity using only pushbroom cameras.

We now show results of experiments with rotation. The orientations of the pushbroom cameras were (θ, τ, ψ), (θ, τ, ψ) and $(\theta, -\tau, \psi)$, where $\theta = -15.0$, $\tau = 20.0$ and $\psi = 5.0$[deg], respectively. Note that unlike the previous works discussed in 3.2, none of the orientation angles are 0. The angle between the cameras (δ in Fig. 7 (b)) were 27.0[deg].

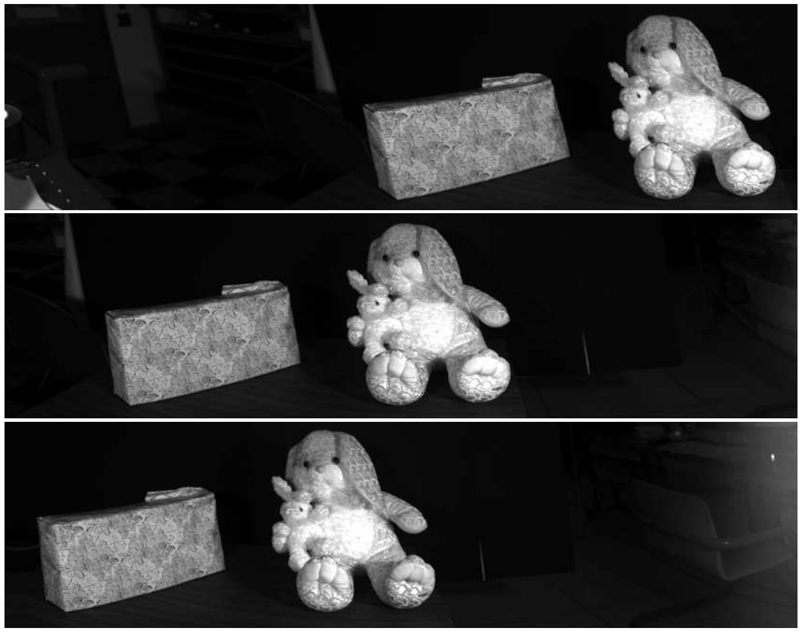


Figure 12: Pushbroom panoramas captured by a set of rotating pushbroom cameras. The size of each image is 5500×1024 [pixel] (the images were cropped for display purposes). The top and middle panoramas were used to recover angular velocity, while the middle and bottom panoramas were used to recover 3D structure.

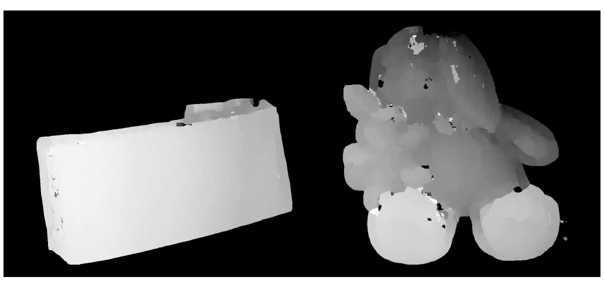
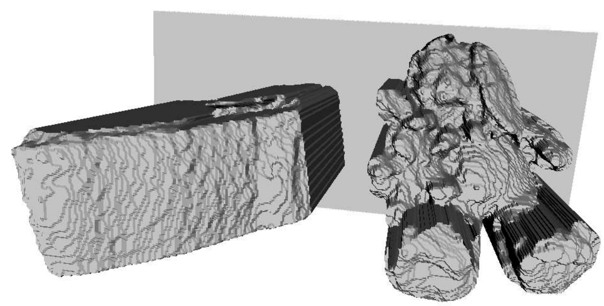



Figure 13: Recovered structure (unknown angular velocity). Top: A depth panorama, where brighter pixels correspond to closer scene points. Bottom: A VRML model of the scene corresponding to a different viewpoint. This result was obtained by rotating the imaging system with unknown angular velocity.

The imaging system was rotated by a DC motor and angular velocity was unknown (no ground truth).

Figure 12 shows the three captured pushbroom panoramas. The size of each image is 5500×1024 [pixel]. The estimated angular velocity obtained from the first and second images was 21.94 [deg/sec]. Using the estimated angular velocity, we were able to reconstruct scene structure as shown in Fig. 13, which was computed from the second and third images.

The results of the experiments demonstrate the usefulness of the method for structure from unknown velocity based on our framework.

6 Summary

In this paper, we have presented a framework for structure recovery using pushbroom cameras. We derived a set of general geometric constraints, including the view plane equation, for a pushbroom camera with 6 DOF motion. The method for structure recovery from unknown velocity using three or more pushbroom cameras for the 1D translation and rotation cases was developed based on these constraints. We showed that the results of several previous works can be viewed as special cases of our approach. Experiments were performed to show the effectiveness of our approach. We believe that the framework presented here will facilitate more applications of pushbroom cameras, especially in fields where high spatial resolution and high frame-rate are important.

A Rotation Matrices

The entities of the rotation matrices used in the transformation between the world and camera coordinate systems in Fig. 2 are as follows:

$$\boldsymbol{R}_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix}, \boldsymbol{R}_y(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix}, \boldsymbol{R}_z(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

B Derivation of Vertical Scale

The vertical scale which shows vertical parallax is the ratio of image coordinates v_{k1} and v_{k2} given by the following equations:

$$v_{k1} = \frac{f\mathrm{y}_{k1}}{\mathrm{z}_{k1}} + p_v, \;\; v_{k2} = \frac{f\mathrm{y}_{k2}}{\mathrm{z}_{k2}} + p_v \,, \tag{37}$$

where, y_{k1}, z_{k1}, y_{k2} and z_{k2} are obtained from Eq.(3) using the orientations and the positions of the cameras. In the special case discussed in Section 3.2, $\mathrm{y}_{k1} = \mathrm{y}_{k2} = \mathrm{y}_{\mathrm{w}}$ (this can be verified by Eq.(3)). Thus the vertical scale is obtained as follows by removing the common term, image center p_v:

$$S_{vs} = \frac{\mathrm{z}_{k2}}{\mathrm{z}_{k1}} \,. \tag{38}$$

From Eq.(3), z_{k1} is obtained as follows:

$$\mathrm{z}_{k1} = -\mathrm{x}_{\mathrm{w}} \sin\phi_{k1} + \mathrm{z}_{\mathrm{w}} \cos\phi_{k1} - R\cos\tau_1 \,. \tag{39}$$

We have the following equation from Eq.(39):

$$z_{k1} = r\sqrt{1 - \sin^2(\alpha - \xi_{k1} - \tau_1)} - R\cos\tau_1 \,, \tag{40}$$

where, $\alpha = \tan^{-1}(\mathrm{z_w}/\mathrm{x_w})$ and $r = \sqrt{\mathrm{x_w}^2 + \mathrm{z_w}^2}$. The view plane equations given by Eq.(27) can be modified using α and r as follows:

$$\sin(\alpha - \xi_{k1} - \tau_1) = -\frac{R}{r}\sin\tau_1 \,. \tag{41}$$

By introducing Eq.(41) into Eq.(40), we have the following representation of z_{k1}:

$$\mathrm{z}_{k1} = r\sqrt{1 - \left(\frac{R}{r}\sin\tau_1\right)^2} - R\cos\tau_1 \,. \tag{42}$$

We can do the same manipulation for z_{k2}. Then Eq.(30) is derived using the condition of $r \neq 0$.

References

[1] R. Gupta and R. I. Hartley: "Linear pushbroom cameras," IEEE Trans. Pattern Anal. & Mach. Intell., Vol.19, No.9, pp.963-975, 1997

[2] http://www.baslerweb.com/

[3] A. Borner: "The optimization of the stereo angle of CCD-line-Scanners," ISPRS Vol.XXXI, Part B1, Commission I, pp.26-30, 1996

[4] A. Gaich, A. Fasching and M. Gruber: "Stereoscopic imaging and geological evaluation for geotechnical modeling at the tunnel site," Felsbau/Rock and Soil Engineering, No.1, pp.15-21, 1999

[5] S. M. Seitz and J. Kim: "The space of all stereo images," Int. J. Computer Vision, Vol.48, No.1, pp.21-38, 2002

[6] J. X. Chai and H. Y. Shum: "Parallel projections for stereo reconstruction," Proc. CVPR'00, Vol.II, pp.493-500, 2000

[7] Z. Zhu, E. M. Riseman and A. R. Hanson: "Parallel-Perspective Stereo Mosaics," Proc. ICCV'01, Vol.I, pp.345-352, 2001

[8] H. Ishiguro, H. Yamamoto and S. Tsuji: "Omni-directional Stereo," IEEE Trans. Pattern Anal. & Mach. Intell., Vol.14, pp.257-262, 1992

[9] H. Y. Shum and R. Szeliski: "Stereo reconstruction from multiperspective panoramas," Proc. ICCV'99, pp.14-21, 1999

[10] P. Peer and F. Solina: "Mosaic-based panoramic depth imaging with a single standard camera," Proc. Workshop on Stereo and Multi-Baseline Vision, pp.75-84, 2001

[11] R. Benosman and J. Devars: "Calibration of the stereovision panoramic sensor," Panoramic Vision, pp.169-180, Springer-Verlag, 2001

[12] http://www.adept.com/

[13] F. Chabat, G.Z. Yang, and D.M. Hansell: "A corner orientation detector," Im. and Vis. Comp., 17, pp. 761-769, 1999.

[14] M. A. Fischler and R. C. Bolles: "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, Vol.24, No.6, pp. 381-395, 1981.