A Method for 3D Reconstruction of Piecewise Planar Objects from Single Panoramic Images

Peter Sturm

IEEE WORKSHOP ON OMNIDIRECTIONAL VISION (WITH CVPR), HILTON HEAD ISLAND, PP. 119–126, JUNE 2000.

A Method for 3D Reconstruction of Piecewise Planar Objects from Single Panoramic Images

Peter Sturm*
INRIA Rhône-Alpes
655 Avenue de l'Europe
38330 Montbonnot St Martin, France
Peter.Sturm@inrialpes.fr

Abstract

We present an approach for 3D reconstruction of objects from a single panoramic image. Obviously, constraints on the 3D structure are needed to perform this task. Our approach is based on user-provided coplanarity, perpendicularity and parallelism constraints. The method is described in detail for the case of a parabolic mirror-based omnidirectional sensor and results are provided.

1 Introduction

Methods for 3D reconstruction from images abound in the literature. A lot of effort has been spent on the development of multi-view approaches allowing for high accuracy and complete modeling of complex scenes. On one hand, research is directed towards completely automatic systems; these are relatively difficult to realize and it is not clear yet if they are ready for use by a non expert. On the other hand, commercial systems exist, but they usually require a high amount of user interaction (clicking on many points in many images).

The guideline of the work described here is to provide an intermediate solution, reconstruction from a single image, that needs relatively little user interaction. Naturally, there are limits on the kind of objects possible to be reconstructed and on the achievable degree of completeness of reconstructions.

Work on reconstruction from single images has been done before, see e.g. [9, 12, 13]. Most of the existing methods were developed for the use of a pinhole camera (with the exception of [12] where mosaics are used). The scope of several of these methods is limited, e.g. the approaches described in [9, 12] only allow to reconstruct planar surfaces whose vanishing line can be determined in the image. One of the two approaches in [9] achieves the reconstruction by measuring heights of points with respect to a ground plane. The drawback of the method is the requirement of the foot point for each 3D point to be reconstructed, i.e. the image of the vertical intersection with the ground plane.

In this paper, we present an approach for 3D reconstruction from a single panoramic image (work on panoramic stereo and ego-motion estimation is described in e.g. [3, 5, 7, 14]). The concrete example of an image acquired with a parabolic mirror-based omnidirectional camera is described, but the method is easily adapted to other omnidirectional sensors. Reconstruction from a single image requires a priori constraints on the 3D structure. We use constraints that are easy to provide: coplanarity of points, perpendicularity of planes and lines, and parallelism of planes and lines. The parallelism and perpendicularity constraints are used to estimate the "directional geometry" of the scene (line directions and plane normals) which forms the skeleton of the 3D reconstruction. Coplanarity constraints are used to complete the reconstruction, via simultaneous reconstruction of points and planes. With the type of information used we are able to reconstruct piecewise planar objects.

The paper is organized as follows. In §2, we describe the camera model. The input to our reconstruction scheme is explained in §3. The basic idea for 3D reconstruction from a single image is outlined in §4. Details on 3D reconstruction are given in §§5 and 6. The complete algorithm is summarized in §7. §8 shows an experimental result and conclusions are given in §9.

2 Camera Model

We use an omnidirectional camera formed by the combination of a parabolic mirror and an orthographic camera whose viewing direction is parallel to the mirror's axis [10]. Orthographic projection can be obtained by using telecentric optics [15]. Geometrically speaking, the projection center of the orthographic camera coincides with the infinite one among the two focal points

*This work is partially supported by the EPSRC funded project GR/K89221 (Vector).

of the paraboloid. Given the image of a point and a small amount of calibration information described below, it is possible to determine the 3D direction of the line joining the original 3D point and the *finite* focal point of the paraboloid. The finite focal point acts as an effective optical center, relative to which correct perspective views of the scene can be created from the panoramic image [1, 11].

In the following, we give formulas needed for calibrating the system and for 3D reconstruction. These formulas are well known [10, 14], but presented here for the sake of completeness.

2.1 Representation of Mirror and Camera

The mirror is a rotationally symmetric paraboloid. Its shape is thus defined by a single parameter a. Without loss of generality, we may represent the paraboloid in usual quadric notation by the following symmetric matrix:

$$\Omega \sim \begin{pmatrix} 4a^2 & 0 & 0 & 0 \\ 0 & 4a^2 & 0 & 0 \\ 0 & 0 & 0 & -2a \\ 0 & 0 & -2a & -1 \end{pmatrix}$$

where $\sim$ means equality up to scale, which accounts for the use of homogeneous coordinates. The mirror's axis is the Z-axis and the finite focal point $\mathbf{F}$ is the coordinate origin, i.e. $\mathbf{F}^\mathsf{T} = (0, 0, 0, 1)$. The parameter a describes the mirror's "curvature".

The viewing direction of the orthographic camera is parallel to the Z-axis, thus the projection matrix can be written as (using homogeneous coordinates):

$$\mathsf{P} \sim \begin{pmatrix} b & 0 & 0 & x_0 \\ 0 & b & 0 & y_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad .$$

The parameter b is the magnification factor of the orthographic projection. The coefficients x_0 and y_0 describe the relative position of the image plane and the mirror, perpendicular to the viewing direction.

2.2 Projection of a 3D Point

Let $\mathbf{Q}$ be a 3D point with coordinates $(X, Y, Z, 1)$. Its projection can be computed as follows. Let $\mathbf{L}$ be the line joining $\mathbf{Q}$ and the mirror's finite focal point $\mathbf{F}$. Among the two intersection points of $\mathbf{L}$ with the mirror Ω, choose the one which lies on the same half-line as $\mathbf{Q}$, with respect to $\mathbf{F}$. The image of $\mathbf{Q}$ is the orthographic projection of this intersection point, giving the image coordinates:

$$\begin{aligned} x &= x_0 + \frac{b}{2a} X \frac{Z + \sqrt{X^2 + Y^2 + Z^2}}{X^2 + Y^2} \\ y &= y_0 + \frac{b}{2a} Y \frac{Z + \sqrt{X^2 + Y^2 + Z^2}}{X^2 + Y^2} \quad . \end{aligned}$$

2.3 Calibration

The above projection equations show that the mirror's shape parameter a and the magnification b of the orthographic projection can be grouped together in a parameter $r = \frac{b}{2a}$ describing the combined system. To calibrate the system, we thus need to estimate the parameters x_0, y_0 and r. These parameters have a simple geometrical meaning: consider the horizontal circle on the paraboloid at the height of the focal point $\mathbf{F}$ (cf. figure 1). The projection of this circle in the orthographic image is exactly the circle ω with center (x_0, y_0) and radius r.

If the mirror's top border does not lie at the height of the focal point (e.g. as shown in figure 1), then we can not directly determine the circle ω in the image. Instead, we fit a circle ω' to the border of the image as shown in figure 2 (a). This circle is cocentric with ω, thus x_0 and y_0 are given by its center. The radius r' of ω', and r are related as follows:

$$r = r' \frac{\cos \alpha}{1 + \sin \alpha}$$

where α is the angle shown in figure 1, which is known by construction.

The calibration procedure has to be done only once for a fixed configuration. Another, more flexible calibration method, is described in [2].

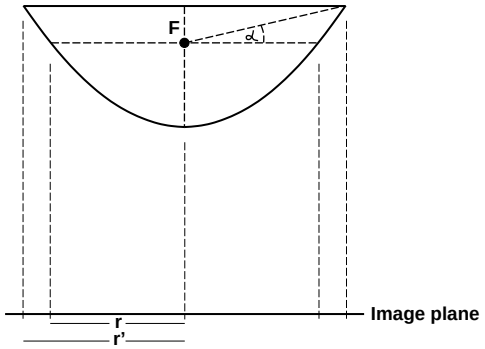


Figure 1: The paraboloidal mirror.

2.4 Backprojection

The most important feature of our mirror-camera system is that from a panoramic image, we may create correct perspective images of the scene, as if they had been observed by a pinhole camera with optical center at $\mathbf{F}$.

This is equivalent to being able to backproject image points via $\mathbf{F}$: we are able to determine the projection ray $\mathbf{L}$ (cf. §2.2) of a point $\mathbf{Q}$, given its image point. Given the calibration parameters, the projection rays are determined in Euclidean space, which is useful for obtaining metric 3D reconstructions as described later.

A projection ray may be represented by its ideal point[1] which can be computed from the image coordinates (x, y) as follows:

$$\begin{pmatrix} \mathbf{B} \\ 0 \end{pmatrix} \sim \begin{pmatrix} 2r(x - x_0) \\ 2r(y - y_0) \\ (x - x_0)^2 + (y - y_0)^2 - r^2 \\ 0 \end{pmatrix} \quad . \tag{1}$$

3 Input

To prepare the description of the 3D reconstruction method, we first explain the (user-provided) input. First of course, the system has to be calibrated, as described in §2.3 (also see figure 2 (a)). The basic primitives for our method are interest points (see figure 2 (b)). Based on interest points, coplanarity, parallelism and perpendicularity constraints are provided as follows.

Lines are defined by two or more interest points and they are grouped together into sets of mutually parallel lines (see figures 2 (c) and (d)). In §5.2 it is described how to compute the direction of a set of parallel lines. In the following, the direction of the ith set of parallel lines will be represented via the ideal point $\left(\mathbf{D}_i^\mathsf{T}, 0\right)^\mathsf{T}$.

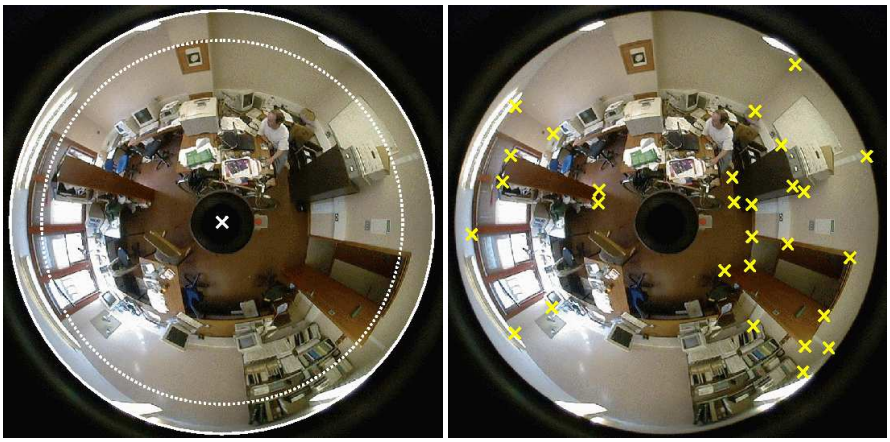
Planes are also defined by interest points and grouped according to parallelism (see figures 2 (e) and (f)). The normal direction of a set of parallel lines can be computed as described in §5.3. The normal of the jth set of parallel planes will be represented via the 3-vector $\mathbf{n}_j$.

Other useful constraints are:

- parallelism of lines and planes, expressed as: $\mathbf{D}_i^\mathsf{T}\mathbf{n}_j = 0$.
- perpendicularity of lines and planes: $\mathbf{D}_i \sim \mathbf{n}_j$.
- perpendicularity of lines: $\mathbf{D}_{i_1}^\mathsf{T}\mathbf{D}_{i_2} = 0$.
- perpendicularity of planes: $\mathbf{n}_{j_1}^\mathsf{T}\mathbf{n}_{j_2} = 0$.

The input data are rather easy to provide interactively, which typically takes 10-15 minutes per image.

[1] By ideal points and ideal lines we denote points and lines at infinity respectively.



(a) Calibration: the dotted line shows the circle ω (see text).

(b) Interest points.

(c) A set of parallel lines.

(d) A set of parallel lines.

(e) A set of parallel planes.

(f) A set of parallel planes.

Figure 2: Illustration of camera calibration and the input used for 3D reconstruction. Crosses of the same color represent interest points belonging to the same line or plane.

4 Basic Idea for 3D Reconstruction

The principal aim here is to reconstruct a set of 3D points and planes. Sets of coplanar 3D points define polygons onto which texture can be mapped for visualization purposes. The reconstruction process is based on operations which are described in the §§5 and 6. Here, we explain that 3D reconstruction from a single image is indeed possible, given the considered types of constraints – coplanarity, parallelism and perpendicularity.

We assume that the image has been calibrated as described in §2.3. Hence, it is possible to backproject image points to 3D, which means that the position of each 3D point is known up to one parameter, its depth. Parallelism and perpendicularity constraints allow us to compute normal directions of some of the planes in the scene (this is described in §§5.2 and 5.3). Hence, these planes are also determined up to one unknown parameter each.

Unless a reference distance in the scene is known, 3D reconstruction can be achieved up to a global scale factor only. We are thus free to arbitrarily fix the position of one point (along its projection ray) or one plane (while preserving its normal). Suppose, we have fixed one point $\mathbf{Q}$. Planes with known normal and which contain $\mathbf{Q}$ (known from the input) are then completely defined. Other points lying on these planes may then be reconstructed, by simply intersecting the backprojection rays with the planes. In turn, other planes may then be reconstructed by fitting them to the already reconstructed 3D points, and so on. This alternation scheme allows to reconstruct objects whose parts are sufficiently "interconnected", i.e. the points on the object have to be linked together via coplanarity or other geometrical constraints.

This discussion shows that it is possible to obtain a 3D reconstruction from one image and constraints of the types considered. However, the alternation scheme just described might not be the best practical solution, since it favors error propagation throughout the reconstruction process. Instead of reconstructing the scene step by step, we thus developed a simple method for simultaneous reconstruction of potentially large sets of points and planes linked together in a way described in §6. Instead of being accumulated, errors are potentially nicely spread over the 3D model. This initial reconstruction is then completed via the alternating point-plane reconstruction scheme outlined above.

In the following section, the basic modules needed for 3D reconstruction are described. The method for simultaneous reconstruction of points and planes is given in §6. The complete algorithm is summarized in §7.

5 Basic Modules for 3D Reconstruction

5.1 Backprojection of Points

Let $\mathbf{q}_p = (x_p, y_p)^\mathsf{T}$ be an image point and $\mathbf{B}_p$ be the 3D direction of the backprojection ray, given by equation (1). Then, the 3D point may be parameterized as

$$\mathbf{Q}_p = \begin{pmatrix} \lambda_p \mathbf{B}_p \\ 1 \end{pmatrix} \quad . \tag{2}$$

The unknown λ_p expresses the distance of $\mathbf{Q}_p$ from the focal point $\mathbf{F}$ and hence defines its position on the projection ray.

5.2 Computation of the Direction of Parallel Lines

Given the input that two or more 3D lines are parallel, we can compute the lines' 3D direction as follows. We suppose that lines are defined by sets of image points (cf. figures 2 (c) and (d)), i.e. a line $\mathbf{l}_{ik}$ is given by points $\mathbf{q}_{ik,1}, \ldots, \mathbf{q}_{ik,n_{ik}}$. For each line, we may compute the 3D interpretation plane, i.e. the plane spanned by the focal point $\mathbf{F}$ and the 3D line. This plane is given by the backprojection rays of the image points. If more than two points are given, a least squares fit is done to determine the plane: the normal is computed as the right singular vector Λ_{ik} associated to the least singular value [6] of the following matrix:

$$\begin{pmatrix} \mathbf{B}_{ik,1}^\mathsf{T} \\ \mathbf{B}_{ik,2}^\mathsf{T} \\ \cdots \\ \mathbf{B}_{ik,n_{ik}}^\mathsf{T} \end{pmatrix}_{n_{ik} \times 3} \quad .$$

The interpretation plane is then given by $\left(\Lambda_{ik}^\mathsf{T}, 0\right)^\mathsf{T}$.

Given the interpretation planes of two or more parallel 3D lines, we may determine the lines' direction as the ideal point obtained by intersection of the interpretation planes' ideal lines. If more than two interpretation planes are given, a least squares fit is done as above. In the following, let the direction of the ith set of parallel lines be represented by the ideal point $\left(\mathbf{D}_i^\mathsf{T}, 0\right)^\mathsf{T}$.

5.3 Computation of the Normal Direction of a Set of Parallel Planes

Planes are depicted by the user by indicating sets of coplanar points in the image (cf. figures 2 (e) and (f)). Given the knowledge that a plane is perpendicular to a set of parallel lines the plane's normal is directly given by the direction of these lines, which may be determined as described above.

A second method to compute a plane's normal is the use of parallelism constraints: given the knowledge that a plane is parallel to two or more sets of mutually parallel lines, then the planes' normal vector $\mathbf{n}_j$ can be computed in a least squares manner: $\mathbf{n}_j$ is obtained as the right singular vector associated to the least singular value of the matrix consisting of the row vectors $\mathbf{D}_i^\mathsf{T}$.

If, by one of these methods, we are able to compute the normal $\mathbf{n}_j$ of a plane Π_j, then the plane may be represented as:

$$\Pi_j = \begin{pmatrix} \mathbf{n}_j \\ d_j \end{pmatrix} \quad , \tag{3}$$

i.e. the plane's position is determined up to its (oriented) distance $-\frac{d_j}{||\mathbf{n}_j||}$ from the focal point $\mathbf{F}$. In the following, we suppose that the normal vectors $\mathbf{n}_j$ have unit norm.

5.4 Other Modules

Our method requires basically two other reconstruction modules, the backprojection of a point onto a 3D plane and the fitting of a plane to a set of 3D points, possibly including ideal points.

Backprojecting a point onto a plane. Backprojection of a point $\mathbf{Q}_p$ onto a plane Π_j is done by computing λ_p via:

$$\lambda_p = -\frac{d_j}{\mathbf{n}_j^\mathsf{T}\mathbf{B}_p} \quad .$$

Fitting a plane to a set of points. Several cases may be considered. In the general case, the cost function to be minimized is the sum of squared distances (we omit here indices referring to the plane):

$$g = \sum_{p=1}^{n} \left(d^2 + 2\left(\mathbf{n}^\mathsf{T}\mathbf{B}_p\right)\lambda_p d + \left(\mathbf{n}^\mathsf{T}\mathbf{B}_p\right)^2 \lambda_p^2 \right) \quad . \tag{4}$$

Nullifying the partial derivatives leads to a linear homogeneous equation system in the unknowns d and $\mathbf{n}$ that is readily solved.

If we already know the plane's normal $\mathbf{n}$, we obtain the following closed form solution for the unknown d:

$$d = -\frac{\sum_{p=1}^{n}\left(\mathbf{n}^\mathsf{T}\mathbf{B}_p\right)}{\sum_{p=1}^{n} 1} \quad .$$

6 Simultaneous Reconstruction of Points and Planes

The coplanarity constraints provided by the user are in general overconstrained, i.e. several points may lie on more than one plane. This means that, due to image noise, it is difficult to obtain a 3D reconstruction that satisfies all the constraints exactly. This may be achieved by constrained optimization, but there might be no batch method of doing so. Thus, in the following we describe a direct least squares solution for reconstructing a subset of object planes and points, minimizing the sum of squared distances between planes and points. Usually, the subsets of planes and points that may be reconstructed this way cover already a large part of the object.

Consider sets of coplanar points, $S_j = \{\mathbf{Q}_{j,1}, \ldots, \mathbf{Q}_{j,n_j}\}$. A point may belong to more than one set S_j. Let Π_j be the plane corresponding to the point set S_j. In the following, we only consider planes with known normal direction.

We say that two planes Π_{j_1} and Π_{j_2} are *connected* if they share a point, i.e. if the intersection of S_{j_1} and S_{j_2} is non empty. This relationship may be visualized by a graph, whose vertices are planes, with edges being drawn between connected planes. We choose a largest subgraph of connected planes (full connection is not required). Let S'_j be the point sets of the selected planes, points lying on one plane only having been eliminated.

We now show how the considered planes and points may be reconstructed simultaneously in a least squares manner. Reconstruction is done via the determination of the scalars λ and d, as given in equations (2) and (3). Let $\mathbf{Q}$ be a point lying on plane Π. The squared distance between them is given by:

$$\left(d + (\mathbf{n}^\mathsf{T}\mathbf{B})\lambda\right)^2 \quad .$$

We minimize the sum of squared distances for pairs of planes and points. The general form of the cost function is given in the following equation:

$$g = \sum_j \sum_{p,\mathbf{Q}_p \in S'_j} \left(d_j^2 + 2\left(\mathbf{n}_j^\mathsf{T}\mathbf{B}_p\right)\lambda_p d_j + \left(\mathbf{n}_j^\mathsf{T}\mathbf{B}_p\right)^2 \lambda_p^2 \right) \quad .$$

The partial derivatives (divided by 2) of the cost function are given by:

$$\frac{\sigma g}{\sigma d_j} = \left(\sum_{p,\mathbf{Q}_p \in S'_j} 1 \right) d_j + \sum_{p,\mathbf{Q}_p \in S'_j} \left(\left(\mathbf{n}_j^\mathsf{T}\mathbf{B}_p\right)\lambda_p \right)$$

$$\frac{\sigma g}{\sigma \lambda_p} = \sum_{j,\mathbf{Q}_p \in S'_j} \left(\left(\mathbf{n}_j^\mathsf{T}\mathbf{B}_p\right) d_j \right) + \left(\sum_{j,\mathbf{Q}_p \in S'_j} \left(\mathbf{n}_j^\mathsf{T}\mathbf{B}_p\right)^2 \right) \lambda_p \quad .$$

Nullifying these equations leads to a homogeneous linear equation system in the unknowns d_j and λ_p, giving the least squares solution. The solution is defined up to scale, as expected, since reconstruction can only be done up to scale.

The equation system has the nice structure shown in equation (5), where

$$\begin{aligned} C_{jp} &= \begin{cases} \mathbf{n}_j^\mathsf{T}\mathbf{B}_p & \text{if } \mathbf{Q}_p \in S'_j, \\ 0 & \text{else.} \end{cases} \\ D_j &= \sum_{p,\mathbf{Q}_p \in S'_j} 1 \\ L_p &= \sum_{j,\mathbf{Q}_p \in S'_j} \left(\mathbf{n}_j^\mathsf{T}\mathbf{B}_p\right)^2 \quad . \end{aligned}$$

The equation system is relatively well conditioned since the $\mathbf{n}_j$ and $\mathbf{B}_p$ are unit vectors and the D_j entries (the number of points on plane j) are usually not much larger than 10. Special sparse solution methods may be used like e.g. in [4], but for small problems (the size of the matrix is the number of planes plus the number of points, which is usually at most a few dozens for single images) we simply use singular value decomposition [6].

$$\left(\begin{array}{cccc|cccc} D_1 & & & & C_{11} & C_{12} & \cdots & C_{1n} \\ & D_2 & & & C_{21} & C_{22} & \cdots & C_{2n} \\ & & \ddots & & \vdots & \vdots & & \vdots \\ & & & D_m & C_{m1} & C_{m2} & \cdots & C_{mn} \\ \hline C_{11} & C_{21} & \cdots & C_{m1} & L_1 & & & \\ C_{12} & C_{22} & \cdots & C_{m2} & & L_2 & & \\ \vdots & \vdots & & \vdots & & & \ddots & \\ C_{1n} & C_{2n} & \cdots & C_{mn} & & & & L_n \end{array}\right) \left(\begin{array}{c} d_1 \\ d_2 \\ \vdots \\ d_m \\ \hline \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{array}\right) = \left(\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right) \quad . \tag{5}$$

7 Complete Algorithm

1. Calibrate the system (cf. §2.3).
2. Backproject all points up to scale, i.e. compute the vectors $\mathbf{B}_p$ (cf. §5.1). Scale the $\mathbf{B}_p$ to unit norm and use extended coordinates for 3D points:
   $\mathbf{Q}_p^\mathsf{T} = (\lambda_p(\mathbf{B}_p)^\mathsf{T}, 1)$.
3. Compute line directions and normal directions of planes (cf. §§5.2 and 5.3).
4. Partition the planes with known normal in sets of planes which are connected by at least one point (in a transitive manner).
5. Choose the largest partition.
6. Simultaneously reconstruct plane and point positions as described in §6. Use only points that lie on more than one plane in the actual partition (other points do not add useful redundancy).
7. Backproject the other points that lie on planes in the actual partition (cf. §5.4).
8. Reconstruct a plane not reconstructed yet by fitting it to 3D points (cf. §5.4). Each point provides one equation (and the possibly known normal direction two). Choose the plane with the most (independent) equations.
9. Backproject points lying on the plane just reconstructed.
10. If there are planes not reconstructed yet, go to step 8.

Note that this process is done completely automatically.
From the 3D reconstruction, we may create textured VRML models (see an example in §8).

8 Example

Figure 4 on the last page shows rendered views of a textured 3D model obtained from the image shown in figure 3 on the following page. The input image was obtained with the CycloVision ParaShot system and an Agfa ePhoto 1680 camera. Texture maps were created from the panoramic image using the projection equations in §2.2 and bicubic interpolation [8]. With other images, similar results were obtained.

9 Conclusion

We have presented a method for interactive 3D reconstruction of piecewise planar objects from a single panoramic view. The method was developed for a sensor based on a parabolic mirror, but its adaptation to other sensors is straightforward. 3D reconstruction is done using geometrical constraints provided by the user, that are simple in nature (coplanarity, perpendicularity and parallelism) and may be easily provided without any computer vision expertise.

The major drawback of single-view 3D reconstruction is of course that only limited classes of objects may be reconstructed and that the reconstruction is usually incomplete. The major advantages however are that it is a quick way of obtaining 3D models, that it is rather easy to implement and to use and that due to user interaction and the small size of the problem the reconstruction process becomes very reliable, compared to more automatic multi-view systems. Also, using geometrical constraints on the scene structure is always a good idea in order to obtain realistic 3D models. 3D models from single images might be used to register perspective views of scene details in order to obtain high resolution global 3D models.

One advantage of our method compared to other approaches is that a wider class of objects can be reconstructed (especially, there is no requirement of disposing of two or more ideal points for each plane). The simultaneous reconstruction of several planes and several points that forms the starting point of our method makes it likely that errors are nicely spread over the whole 3D model, compared to more sequential approaches.

Please contact the author for getting a paper version with color figures.

References

[1] S. Baker and S.K. Nayar, "A Theory of Catadioptric Image Formation," *Proceedings International Conference on Computer Vision, Bombay*, pp. 35-42, January 1998.

[2] C. Geyer and K. Daniilidis, "Catadioptric Camera Calibration," *Proceedings International Conference on Computer Vision, Kerkyra, Greece*, pp. 398-404, September 1999.

[3] J. Gluckman and S.K. Nayar, "Ego-Motion and Omnidirectional Cameras," *Proceedings International Conference on Computer Vision, Bombay*, pp. 999-1005, January 1998.

[4] R.I. Hartley, "Euclidean Reconstruction from Uncalibrated Views," *Proceeding of the* DARPA–ESPRIT *Workshop on Applications of Invariants in Computer Vision, Azores, Portugal*, pp. 187-202, October 1993.

[5] H. Ishiguro, M. Yamamoto and S. Tsuji, "Omni-Directional Stereo," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 257-262, February 1992.

[6] A. Jennings, J.J. McKeown, *Matrix Computation,* 2nd edition, Wiley, 1992.

[7] S.B. Kang and R. Szeliski, "3-D scene data recovery using omnidirectional multi-baseline stereo," *Proceedings* IEEE *Conference on Computer Vision and Pattern Recognition, San Francisco*, pp. 364-370, June 1996.

[8] R.G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," IEEE *Transactions on Acoustics, Speech, and Signal Processing,* Vol. 29, No. 6, pp. 1153-1160, December 1981.

[9] D. Liebowitz, A. Criminisi and A. Zisserman, "Creating Architectural Models from Images," *Proceedings EuroGraphics*, vol. 18, pp. 39-50, September 1999.

[10] S.K. Nayar, "Catadioptric Omnidirectional Camera," *Proceedings* IEEE *Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pp. 482-488, June 1997.

[11] V.N. Peri and S.K. Nayar, "Generation of Perspective and Panoramic Video from Omnidirectional Video," *Proceedings* DARPA *Image Understanding Workshop, New Orleans*, May 1997.

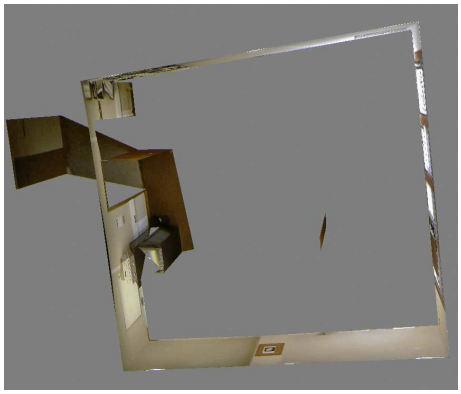
[12] H.-Y. Shum, R. Szeliski, S. Baker, M. Han, P. Anandan, "Interactive 3D Modeling from Multiple Images Using Scene Regularities," *SMILE Workshop, Freiburg, Germany*, pp. 236-252, June 1998.

[13] P. Sturm and S. Maybank, "A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images," *Proceeding British Machine Vision Conference, Nottingham*, pp. 265-274, September 1999.

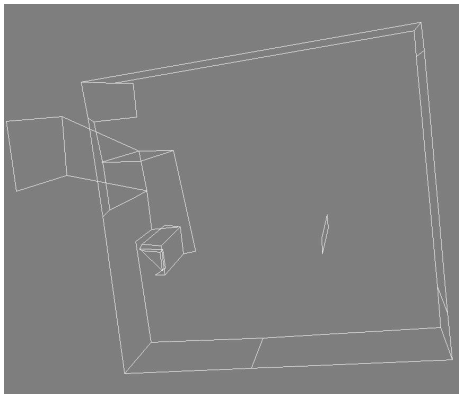
[14] T. Svoboda, *Central Panoramic Cameras – Design, Geometry, Egomotion,* PhD Thesis, Faculty of Electrical Engineering, Czech Technical University, Prague, September 1999.

[15] M. Watanabe and S.K. Nayar, "Telecentric Optics for Computer Vision," *Proceedings European Conference on Computer Vision, Cambridge*, pp. 439-451, April 1996.

Figure 3: The input image.



(a) An overhead view of the scene. On the left hand side, a part of a hallway outside the office that has been reconstructed, is visible.



(b) A wireframe model of the reconstruction.

(c) A view from below the floor.

(d) A view from below and outside the reconstructed office.

(e) Objects not visible in the input image obviously lead to holes in the 3D model.

Figure 4: Results.