Interactive Reconstruction from an Omnidirectional Image

José Gaspar, Etienne Grossmann, José Santos-Victor[⋆]

Instituto de Sistemas e Robótica,
Instituto Superior Técnico,
Av. Rovisco Pais,
1049-001 Lisboa,
Portugal.

Abstract. We propose a method for 3D reconstruction of structured environments from a single omnidirectional image. It is based on a reduced amount of user information, in the form of 2D pixel coordinates, alignment and coplanarity properties amongst subsets of the corresponding 3D points. Just a few panoramic images are sufficient for building the 3D model, as opposed to a larger number of "normal" images that would be required to reconstruct the same scene [11, 14]. One contribution of the paper is showing how to build these 3D models with an omnidirectional camera equipped with a spherical mirror, in spite of not having a single projection centre. Additionally, we present concisely a simple method for single-view reconstruction.

1 Introduction

The construction of scene models is a well known problem in the computer graphics and in the computer vision communities. While in the former there is traditionally a strong emphasis in using precise user-defined geometric and texture data, in the later the emphasis is more on the direct use of images to automatically correspond and generate depth or shape maps. Recently, many works started to combine with success both approaches in a way well tuned for each purpose [4, 11, 3, 12, 5, 13, 15, 14].

Our motivation comes from the field of tele-operation of mobile robots. Given an image of a structured environment and some user input derived from his understanding (knowledge) of the environment, one can reconstruct a 3D scene model for visualisation or for specifying moving actions for the robot.

Our robot is equipped with an omnidirectional camera that provides a 360° view of the environment in a single image. Omnidirectional images are usually obtained with Catadioptric Panoramic Cameras, which combine conventional cameras (lenses) and convex mirrors. Mirror shapes can be conic, spherical, parabolic or hyperbolic [1, 16, 17]. The wide field of view of omnidirectional vision sensors makes them particularly well suited for fast environmental modelling.

The user input consists of geometric nature supported on his knowledge of parallelism or perpendicularity of lines and planes of the scene. Typically, the user will identify some points in the omnidirectional images and indicate that some subsets of points are collinear or coplanar. We restrict ourselves to lines parallel to one of the canonical axes and planes whose normal is parallel to one of these axes.

In this paper we start by presenting the geometry of the spherical projection model for single projection centre omnidirectional cameras and, also, the geometry of our own omnidirectional camera based on a spherical mirror. Even though a camera with a spherical mirror does not have a single projection centre, we show that it can be approximated by the spherical projection model. Then we present the reconstruction method based on user defined input over the omnidirectional images. Next we describe some preliminary results. Finally, we present our conclusions and future work.

2 Omnidirectional vision geometry

It is well known that some omnidirectional images can be transformed back to an equivalent perspective image, provided that all the projection rays intersect in a single projection centre. This is not the case

[⋆] Email: {jag,etienne,jasv}@isr.ist.utl.pt when an omnidirectional camera equipped with a spherical mirror [7] is used, because it does not have a single projection centre. However, we shall show that it can be approximated by a pin-hole camera. We first describe briefly the spherical projection model, which is known to be equivalent to a pin-hole camera and then show that it can approximate the projection with a spherical mirror.

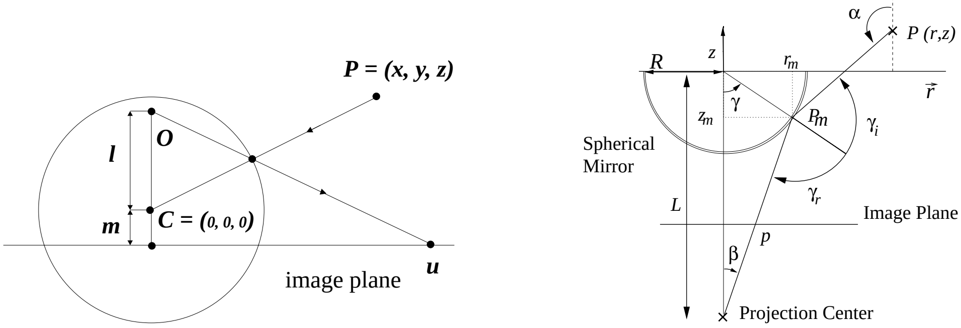
2.1 Using spherical projection for spherical mirrors

The *Spherical Projection Model*, defined by Geyer and Daniilidis in [8], represents in an unified manner several single projection centre systems, like pin-hole cameras and, mostly important, the recent omnidirectional (catadioptric) cameras based on hyperbolic, elliptical or parabolic mirrors.

The spherical projection model combines a mapping to a sphere followed by a projection to a plane. The centre of the sphere lies on the optical axis of the projection to the plane. This allows a reduced representation with two parameters, l and m, representing the distances from the sphere centre to the projection centre, O and to the plane (see Fig.1a). The projection of a point in space (x, y, z) to an image point (u, v) can be written as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{l+m}{l \cdot r - z} \begin{bmatrix} x \\ y \end{bmatrix} = \mathcal{P}(x, y, z; l, m) \tag{1}$$
$$r = \sqrt{x^2 + y^2 + z^2}$$

Each catadioptric camera with a single projection centre can be represented by a set of values l, m. For example, for pin-hole cameras, we have $l = 0$ and $m = 1$, while for cameras with hyperbolic mirrors, l and m are defined by the mirror parameters eccentricity and inter-focal length. The camera intrinsic parameters, image centre and focal length, combine naturally with the model as a two dimensional affine transformation of $[u\ v]^T$.



(a) (b)

Fig. 1. Projections (a) unified model, (b) spherical mirror.

Catadioptric vision sensors based on a spherical mirror, represented in Fig.1b, are modelled essentially by the equation of reflection at the mirror surface, $\gamma_i = \gamma_r$. Adding the pin-hole projection model we obtain the system that allows to compute the projected image point, $p = [u\ v]^T$, by first computing the angle β between the branch of the projection ray from P_m to p and the optical axis of the camera [7]:

$$\begin{cases} \gamma_i = \gamma_r \ \Leftrightarrow \arctan\left(\frac{z - z_m}{r - r_m}\right) + \frac{\pi}{2} - \beta = -2 \arctan\left(r_m / z_m\right) \\ r_m = (z_m + L)\ \tan\beta \\ z_m^2 + r_m^2 = R^2 \end{cases} \tag{2}$$

where P, P_m denote respectively one 3D point and the point of reflection at the surface of the spherical mirror, R is the radius of the mirror and L is the distance between the mirror and the camera.

The intersections of the projection rays PP_m define a continuous set of points distributed in a volume, unlike the spherical projection model where they all converge to a single point. Baker and Nayar [1], showed that the projection centre corresponding to a spherical mirror lies over a spherical surface and varies with scene structure. In spite of the fact that the projection with spherical mirrors can not be represented by the spherical projection model, we will show that for a certain operation range, this model can be a good approximation.

A camera with a spherical mirror cannot be exactly represented by the spherical projection model. In order to find an approximate representation we focus the comparison on the image projection error, instead of analysing the projection centre itself.

Let $\mathcal{P}(x_i, y_i, z_i; \theta)$ denote the spherical projection defined in eq.(1) and $\mathcal{P}_c$ be the projection with a spherical mirror defined in eq.(2). Grouping into θ and θ_c the geometric and intrinsic parameters for the former and latter projections, we want to minimise a cost functional associated to the image projection error:

$$\hat{\theta} = \arg_\theta \min \sum_i \left\| \mathcal{P}(x_i, y_i, z_i; \theta) - \mathcal{P}_c(x_i, y_i, z_i; \theta_c) \right\|^2$$

The minimisation of the functional gives the desired parameters, θ, for the unified projection model, $\mathcal{P}$, that will approximate the real sensor characterised by $\mathcal{P}_c$ and θ_c.

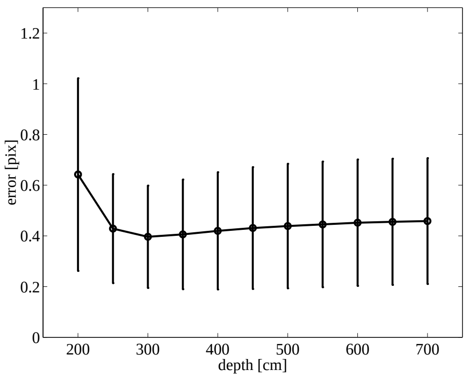


Fig. 2. Mean absolute error between the spherical projection model and the projection with the spherical mirror. Vertical bars indicate the standard deviation. Our omnidirectional images have 500x500 pixels.

Figure 2 shows that the approximation errors measured in the image plane are small by considering 3D points distributed around the sensor at several heights, in a range of 2 to 7m from the camera optical axis.

2.2 Using back-projection to form perspective images

The acquisition of correct perspective images, independently of the scenario, requires that the vision sensor be characterised by a single projection centre [1]. The spherical projection model has, by definition, this property but, due to the intermediate mapping over the sphere, the obtained images are in general not perspective.

In order to obtain correct perspective images, the spherical projection must be first reversed from the image plane to the sphere surface and then, re-projected to the desired plane from the sphere centre. We term the reverse projection as *back-projection*, after Sturm in [15, 14].

The back-projection of an image pixel (u, v), obtained through spherical projection, yields a 3D direction $k \cdot (x, y, z)$ given by the next equations derived from eq.(1):

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{l(l+m) - sign(l+m)\sqrt{(u^2+v^2)(1-l^2)+(l+m)^2}}{u^2+v^2+(l+m)^2} \begin{bmatrix} u \\ v \end{bmatrix} \tag{3}$$

$$z = \pm\sqrt{1 - x^2 - y^2}$$

where z becomes negative if $|l+m|/l > \sqrt{u^2+v^2}$, and positive otherwise. It is assumed, without loss of generality, that (x, y, z) is lying on the surface of the unit sphere.

At this point, it is worth noting that the set $\{(x, y, z)\}$ interpreted as points of the projective plane, already define a perspective image. However for the purpose of displaying or to obtain specific viewing directions further development is needed.

Let R denote the orientation of the desired (pin-hole) camera relative to the frame associated to the results of back-projection, the new perspective image $\{(\lambda u, \lambda v, \lambda)\}$ becomes:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = K \cdot R^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4}$$

where K contains intrinsic parameters and λ is a scaling factor. This is the pin-hole camera projection model [6], when the origin of the coordinates is the camera centre.

2.3 Reconstruction reference frame

Vanishing points, i.e. image points representing scene points at an infinite distance to the camera [2], show important scene directions through which the reconstruction can be conveniently done. A vanishing point is the intersection in the image of the projection of parallel 3D lines. If one has two parallel lines, defined by two points, AB and CD, then the corresponding vanishing point $\mathbf{v}$ is:

$$\mathbf{v} = (A \times B) \times (C \times D) \tag{5}$$

where the points A, B, C and D are in homogeneous coordinates obtained according to the back-projection eq.(3). In case of having more points in each line and more lines in each set, least squares estimates replace the external products in eq.(5) and more robust estimates are obtained for the vanishing point coordinates [14].

Given three (unit-norm) vanishing points, $\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$, representing three world orthogonal directions, it is possible to obtain perspective images in a reference frame built on those directions using eq.(4) with $R = [\mathbf{v_1}\ \mathbf{v_2}\ \mathbf{v_3}]$.

In our case the optical axis of the sensor, z, is vertical which is one of the most frequent (and representative) directions of lines in the scene. The corresponding vanishing point in the reference frame defined by back-projection eq.(3) is simply $[001]^T$, and the new reference frame, associated to the rotation matrix of eq.(4), takes the form of a rotation about z:

$$R = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In what follows, we rotate the image so that the x and y axes of the camera frame coincide with that of the world reference frame. One thus has, in Eq.(4),

$$R = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} = [\mathbf{e}_1\, \mathbf{e}_2\, \mathbf{e}_3] \tag{6}$$

where the $\mathbf{e}_i$ form the canonical basis of $\Re^3$.

3 Reconstruction

We have just shown that, for all practical effects, we can consider that the input image is obtained by a pinhole camera whose orientation coincides with the world reference frame. The user input consists of image data and auxiliary geometric information, as follows:

Image features Image points $[u_1\, v_1]^\top, \ldots, [u_P\, v_P]^\top$, projections of 3D points $[x_1\, y_1\, z_1]^\top, \ldots, [x_P\, y_P\, z_P]^\top$. For example, points 1..16 are illustrated in Figure 3.

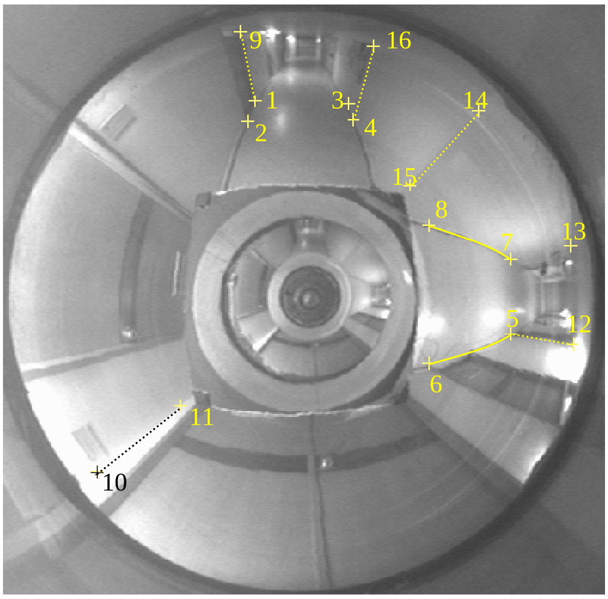


Fig. 3. Typical input image. Some of the points and lines localised by the user are shown.

Auxiliary geometric information

1. Information of 3D alignment of some of the 3D objects that project in the above-mentioned image features. This information can e.g. take the form of lists of indices of image features, lines or points. Also, the 3D direction of each line ("x", "y" or "z") is known. For example, the user could have specified in Figure 3 that the dashed and smoothed lines are respectively vertical and parallel to the "x" axis.
2. Information of coplanarity of some of the 3D objects that project, together with knowledge of the normal of the planes. This can again be expressed in the form of lists of (indices of) image features. For example, in Figure 3, the user could have specified that points (1,2 and 9) lie on a "x=Constant" plane and that points (1,...,8) lie on a horizontal plane.

3.1 Exploiting the user information

The auxiliary geometric information serves to determine what distinct coordinates are that will be estimated.

Let us consider the line between points (1,9) in the input data in Figure 3. Since this line is parallel to the "z" axis, the coordinates of these points are of the form $[C_1, C_2, C_3]$ and $[C_1, C_2, C_4]$ respectively. Note that the first and second coordinates are identical. Then, considering that points (1,2, 3 and 4) lie on a horizontal plane, and points (1, 2, 9) lie on a "x=Constant" plane, one knows that the coordinates of point 2 are of the form $[C_1, C_5, C_4]$. By using in this manner the user-supplied information, it is easy to identify the set of distinct coordinates that will be estimated. This operation is easily automated, using basic set operations, by sequentially inspecting the input data.

As a result, one can determine the number of distinct 3D coordinates $C_1, \ldots, C_M$ and the corresponding 2D features. After having determined the camera orientation, R, and identified the set of distinct coordinates (but not their value), we proceed to show how to obtain a 3D reconstruction from the 2D image features. Figure 4 shows the camera frame, aligned with the world frame.

Linear constraints from 2D lines A 3D line parallel to the i^{th} canonical axis is a set of points :

$$\{\mu \mathbf{e}_i + C\mathbf{e}_{i'} + C'\mathbf{e}_{i''} | \mu \in R\}$$

where C, C' are real constants and i', i'' are distinct indices such that $\{i, i', i''\} = \{1, 2, 3\}$. The projection (Eq. (4)) of a 3D point belonging to this line has the form :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda \left(\mu \mathbf{e}_i + C'\mathbf{e}_{i'} + C'\mathbf{e}_{i''} \right). \tag{7}$$

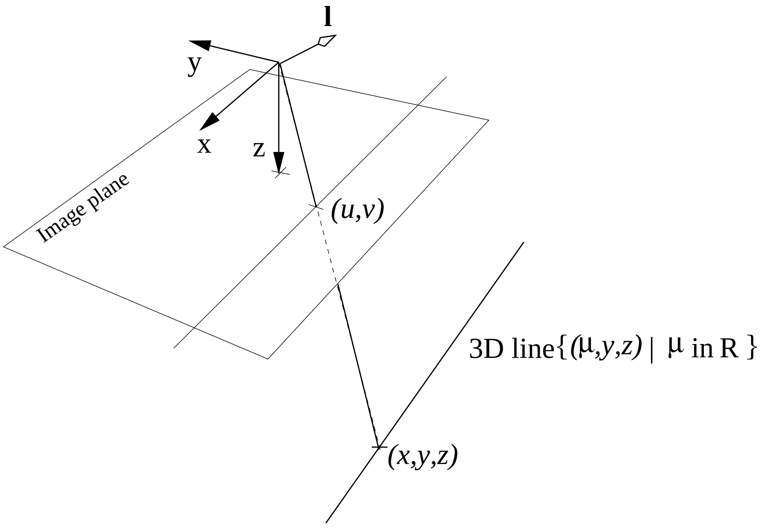


Fig. 4. Perspective projection with equal camera and world reference frame.

A 2D line is represented by a 3-by-1 vector $\mathbf{l}$. The line is the set of 2D points $[u, v]^\top$ such that

$$\mathbf{l}^\top \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0. \tag{8}$$

If the user has located a line $\mathbf{l}$ in the image, resulting from projection, as in Eq. (7), (see Figure 4) we can use $\mathbf{l}$ to build a linear constraint on the unknown elements C, C' : replacing Eq. (7) in Eq. (8) one gets the implication :

$$\mathbf{l}^\top \mathbf{r}_{i'} C + \mathbf{l}^\top \mathbf{r}_{i''} C' = 0\,. \tag{9}$$

This equation is a linear equation in two of the coordinates that we will be estimating. This kind of equation forms the basis of the linear system that we will build and solve.

Linear constraints from 2D points If the projection $[u\ v\ 1]^\top$ of a point $[C, C', C'']^\top$ is observed, it is possible to build the 2D line that passes through that point and any one of the vanishing points. This line is

$$\mathbf{l} = \mathbf{r}_i \times \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}.$$

This 2D line, is moreover the projection of the 3D line

$$\{\mu \mathbf{e}_i + C' \mathbf{e}_{i'} + C'' \mathbf{e}_{i''} | \mu \in R\}\,.$$

This 2D line, which we built rather than observed, can be used to form a constraint Eq. (9) in exactly the same way as an observed line. One verifies that the three constraints given by each point (one constraint per vanishing point) form a system of rank two.

3.2 Solutions to the reconstruction problem

If $\mathbf{C} = [C_1, \ldots, C_N]^\top$ is the vector of the distinct coordinates, by concatenating Equations (9) obtained from the input data, and removing equations that are equal, one obtains a system of M equations:

$$A\mathbf{C} = 0_{M,1} \tag{10}$$

where A is the M-by-N matrix holding the coefficients that multiply the C_i . Each row of A has in general[2] exactly two non-zero elements.

[2] It is more correct to say “almost always” rather than “in general”. We use the later term to avoid having to introduce a probability space on the input data, in which the property holds almost always.

Under some broad conditions, which we make precise in another article [10], one shows that A has corank 1 and thus has a single (up to scale) null vector $\mathbf{C}^*$. The general form of the solution for Eq. (10) is thus :

$$\mathbf{C} = \lambda \mathbf{C}^* \tag{11}$$

where λ is an arbitrary scale factor.

In the presence of noise, A will not be rank-deficient. In order to obtain a solution of the form of Eq. (11), we replace A by the rank-deficient matrix that best approximates it, for the Frobenius norm. This matrix is easily obtained from the singular value decomposition of A [9].

Equation (11) says that, even with a single view, there is no ambiguity in the reconstruction, other than that –well-known– of scale. An important point is that the reconstruction of the whole scene is obtained in a single step.

4 Experimental Results

Figure 3 shows the original image with part of the user input superposed. The user input consists in the 16 points shown and knowledge that some sets of points belong to constant x, y or z, planes, and that some other sets of points belong to lines parallel to x, y or z axes. Table 1 details all the user-defined data. Figure 5 shows this information in a graphic way : planes orthogonal to the x and y axes are in light gray and white respectively, and one horizontal plane is shown in dark gray (the topmost horizontal plane is not shown because it would occlude the other planes)

Axis	**Planes**	**Lines**
x	$(1,2,9,10,11)$, $(3,4,14,15,16)$, $(5,7,12,13)$	
y	$(5,6,10,11,12)$, $(7,8,13,14,15)$, $(1,3,9,16)$	$(1,2)$
z	$(1,2,3,4,5,6,7,8)$, $(9,12,13,16)$	

Table 1. User-defined planes and lines. The numbers here are indexes of image points as shown in figure 3. The first column indicates the axis to which the planes are orthogonal and the lines are parallel.

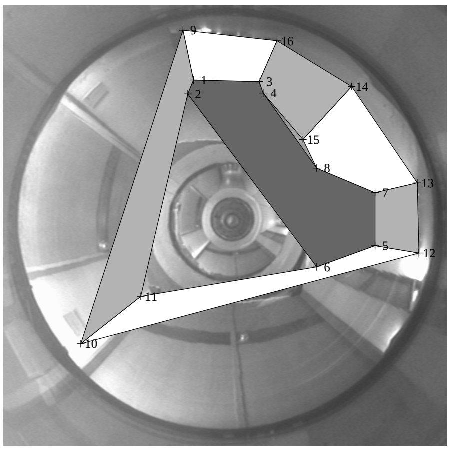


Fig. 5. User defined planes orthogonal to the x axis (light gray), y axis (white) and z axis (dark gray).

The coordinates in the original image were transformed in the equivalent pin-hole model coordinates, and used for reconstruction. Figure 6 shows some views of the texture-mapped resulting reconstruction. These results are interesting in the sense that the required amount of information is small, just a single image and reduced user input, and the whole surroundings of the sensor are obtained.

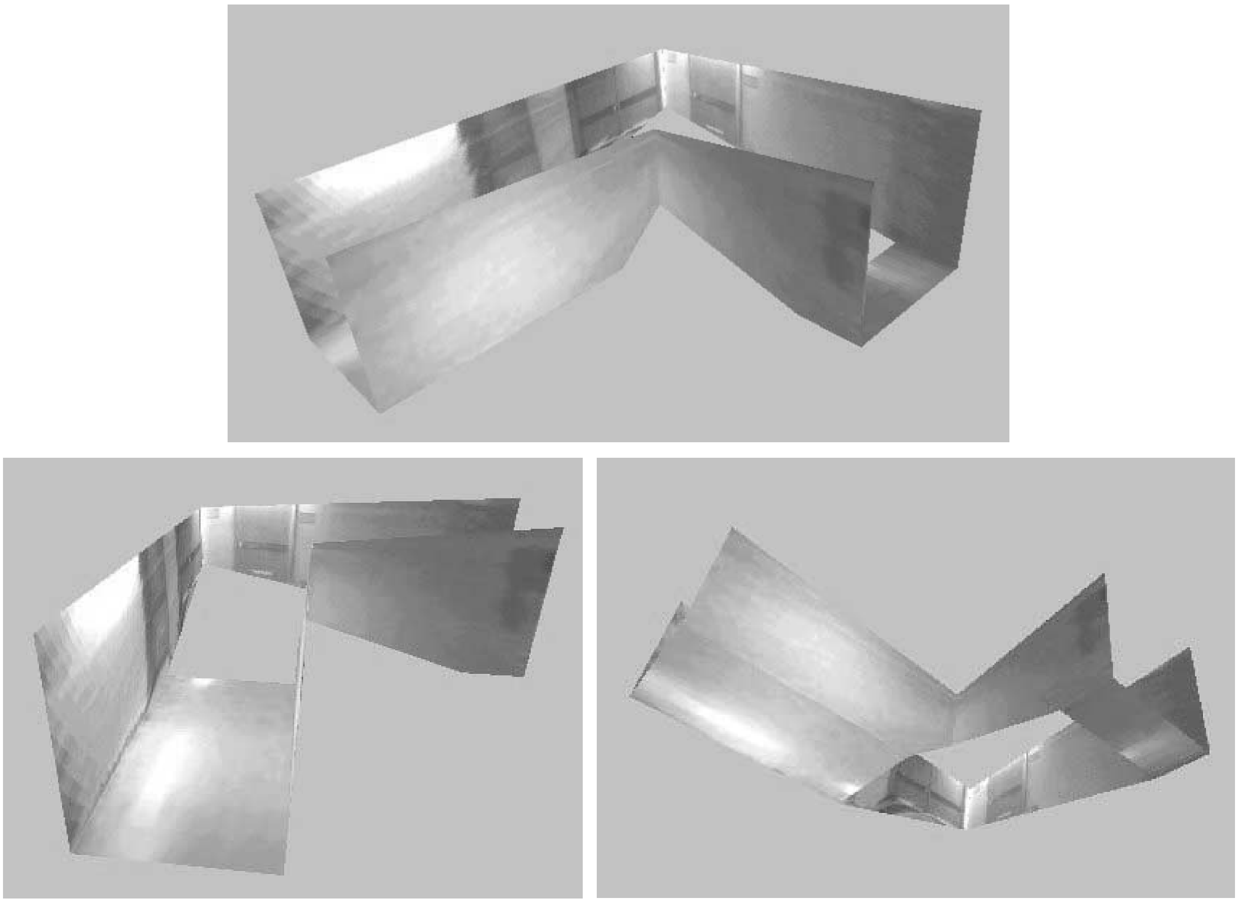


Fig. 6. Views of the reconstructed 3D model.

5 Discussion and Future work

We have shown that a sensor equipped with a spherical mirror can be approximately modelled by spherical projection. Using a single-view reconstruction technique requiring limited user input, one obtains a model of the environment (surroundings) of the robot that carries the sensor.

Such models can be directly used in the context of e.g, tele-operation. A remote user can instruct the robot to move to desired position, simply by manipulating the model to reach the desired view point. Such simple scene models can be transmitted even with low bandwidth connections.

There are a number of ways to further extend this work. On one hand, one can fuse different models together (by 3D-to-3D registration) to create large scene models and to improve the quality of the data (e.g. texture resolution). Another research direction is that of automatically estimating geometric constraints that can be used for 3D reconstruction, hence keeping the user intervention to a minimum. In the future, we plan to use these models for closed-loop navigation, where acquired images should be compared to the model for generating the appropriate commands.

This approach offers a simple procedure for building a 3D model of the scene where a vehicle may operate. Even though the models do not contain very fine details, they can provide the remote user (or the autonomous) robot with a sufficiently rich description of the environment.

Acknowledgements

This work was partly funded by the European Union RTD - Future and Emerging Technologies Project Number: IST-1999-29017, Omniviews; and by Praxis XXI Grant BD /19594 /99.

References

1. Simon Baker and Shree K. Nayar. A theory of single-viewpoint catadioptric image formation. *Int. J. of Computer Vision*, 35(2):175–196, 1999.
2. B. Caprile and V. Torre. Using vanishing points for camera calibration. *IJCV*, 4:127–140, 1990.
3. A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *ICCV*, pages 434–441, 1999.
4. Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *SIGGRAPH*, 1996.
5. Anthony Dick, Phil Torr, and Roberto Cipolla. Automatic 3d modelling of architecture. In *BMVC (vol1)*, pages 372–381, 2000.
6. O. Faugeras. *Three Dimensional Computer Vision.* MIT Press, 1993.
7. J. Gaspar and J. Santos-Victor. Visual path following with a catadioptric panoramic camera. In *7th Int. Symp. on Intelligent Robotic Systems (SIRS'99)*, pages 139–147, Coimbra, Portugal, July 1999.
8. C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems and practical applications. In *ECCV (vol2)*, pages 445–461, 2000.
9. Gene H. Golub and Charles F. Van Loan. *Matrix computations.* Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.
10. E. Grossmann, D. Ortin, and J. Santos-Victor. Reconstruction of structured scenes from one or more views: Nature of solutions. *Submitted to IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2001.
11. Sing Bing Kang and Richard Szelisky. 3-d scene recovery using omnidirectional multibaseline stereo. In *CVPR*, pages 364–370, 1996.
12. D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Proceedings Euro-Graphics (vol.18)*, pages 39–50, 1999.
13. D. Robertson and R. Cipolla. An interactive system for constraint-based modelling. In *BMVC (vol2)*, pages 536–545, 2000.
14. P. Sturm. A method for 3d reconstruction of piecewise planar objects from single panoramic images. In *IEEE OMNIVIS (workshop of CVPR)*, pages 119–126, 2000.
15. Peter F. Sturm and Stephen J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *BMVC*, pages 265–274, 1999.
16. T. Svoboda, T. Pajdla, and V. Hlavác. Epipolar geometry for panoramic cameras. In *ECCV'98*, pages 218–231, Freiburg Germany, July 1998.
17. S. C. Wei, Y. Yagi, and M. Yachida. Building local floor map by use of ultrasonic and omni-directional vision sensor. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 2548–2553, Leuven, Belgium, May 1998.

This article was processed using
the TEX macro package with SIRS2001 style