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

Peter F Sturm* and Stephen J Maybank
Computational Vision Group, Department of Computer Science
The University of Reading, Whiteknights, PO Box 225
Reading, RG6 6AY, United Kingdom
{P.F.Sturm,S.J.Maybank}@reading.ac.uk

Abstract

We present an approach for 3D reconstruction of objects from a single 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. These are used to calibrate the image and perform 3D reconstruction. The method is described in detail 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) or a special camera setup (e.g. using structured light).

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. However, our results suggest that such a minimal solution for reconstruction might be quite useful, e.g. for visualization and augmented reality purposes.

Work on reconstruction from single images has been done by others. Shum et al. describe a method similar to ours in [6]. Their method, however, allows to reconstruct only planes whose vanishing lines can be computed from two or more sets of parallel lines, whereas our method can also reconstruct arbitrary planes, thus leading to a wider class of objects that may be reconstructed. Liebowitz et al. describe two different methods for single-view 3D reconstruction in [5]. The first method 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. This puts a limit to the nature of objects that may be reconstructed. The second method of Liebowitz et al. requires, like the method by Shum et al., the computation of the vanishing lines of all planes to

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

BMVC 1999 doi:10.5244/C.13.27

be reconstructed. Their method also appears to be less straightforward than the one we describe in this paper, e.g. it performs intermediate rectifications of the images of planar patches that might perhaps be omitted.

Reconstruction from single images requires geometrical constraints on the 3D structure of the observed object. The approach described in this paper is based on three types of constraints: coplanarity of points, perpendicularity of directions or planes and parallelism of directions or planes. Perpendicularity constraints are used to calibrate the image. Together with parallelism constraints they provide the vanishing geometry of the scene which forms the skeleton of the 3D reconstruction. Coplanarity constraints are used to complete the reconstruction, via alternating reconstruction of points and planes.

The paper is organized as follows. In §2, we describe our camera model and the computation of vanishing points and lines. Details on calibration and 3D reconstruction are given in §§3 and 4 respectively. The complete algorithm is summarized in §5. §6 gives an example of how the algorithm works and presents some results. Conclusions are given in §7.

2 Background

2.1 Camera Model

We use perspective projection to model cameras. A projection may be represented by a 3×4 projection matrix P that maps points of 3-space to points in 2-space: $\mathbf{q} \sim \mathsf{P}\mathbf{Q}$. Here, $\sim$ means equality up to a non zero scale factor, which accounts for the use of homogeneous coordinates. Since we consider a single view and may chose the 3D reference frame arbitrarily, we align it with the camera, leading to the simple projection matrix $\mathsf{P} \sim (\ \mathsf{K}\ |\mathbf{0})$. Here, K is the 3×3 calibration matrix:

$$\mathsf{K} = \begin{pmatrix} \tau f & s & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} .$$

In general, we distinguish 5 intrinsic parameters for the perspective projection model: the (effective) focal length f, the aspect ratio τ, the principal point (u_0, v_0) and the skew factor s accounting for non rectangular pixels. The skew factor is usually very close to 0 and we ignore it in the following.

2.2 Vanishing Points and Lines

We compute vanishing points as the least squares solution for the intersection of sets of images of parallel 3D line segments. The information of line segments being parallel is provided by the user.

Vanishing lines are determined from vanishing points and parallelism constraints. The assumption that a vanishing point $\mathbf{v}$ belongs to a 3D direction parallel to a plane implies that $\mathbf{v}$ lies on the vanishing line $\mathbf{l}$ of that plane. Hence, two or more vanishing points parallel to a plane define its vanishing line.

A vanishing point $\mathbf{v}$ that belongs to the 3D direction *perpendicular* to a plane, completely defines the vanishing line (if the image is calibrated): $\mathbf{l} \sim \mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}\mathbf{v}$.

3 Calibration

In the following, we derive calibration equations that are based on vanishing points of pairs of perpendicular directions. This approach is well known (cf. e.g. [1]); we briefly describe it and give then a closed-form solution for the focal length which is usually the only intrinsic parameter that we calibrate in practice.

Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the vanishing points of two perpendicular 3D directions. Let the ideal points of the 3D directions be written as: $(\mathbf{V}_1^\mathsf{T}, 0)^\mathsf{T}$ and $(\mathbf{V}_2^\mathsf{T}, 0)^\mathsf{T}$. From the projection equations $\mathbf{v}_1 \sim \mathsf{K}\mathbf{V}_1$ and $\mathbf{v}_2 \sim \mathsf{K}\mathbf{V}_2$ we compute the ideal points as:

$$\begin{aligned} \mathbf{V}_1 &\sim \mathsf{K}^{-1}\mathbf{v}_1 \\ \mathbf{V}_2 &\sim \mathsf{K}^{-1}\mathbf{v}_2 \ . \end{aligned}$$

The 3D directions being perpendicular means that $\mathbf{V}_1^\mathsf{T}\mathbf{V}_2 = 0$, hence:

$$\mathbf{v}_1^\mathsf{T}\mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}\mathbf{v}_2 = 0 \ . \tag{1}$$

This equation is homogeneous linear in the coefficients of the symmetric matrix $\omega \sim \mathsf{K}^{-\mathsf{T}}\mathsf{K}^{-1}$ (which represents the image of the Absolute Conic). Having determined ω, using equations (1) or other means, the calibration matrix K may be computed uniquely using Cholesky decomposition [4].

Each pair of perpendicular vanishing points gives one constraint on the intrinsic parameters in K. In a man-made environment, we will typically observe three pairs of mutually perpendicular vanishing points, sometimes more, sometimes only a single pair. This puts a limit on the number of intrinsic parameters that may be computed. The aspect ratio τ can often be assumed to be known. Depending on the number of calibration equations, we may estimate the principal point and the focal length. For the experiments described later, we assumed that the principal point is in the center of the image, and only estimated the focal length.

The equations for the focal length are particularly simple. We may decompose the calibration matrix in its known and unknown parts:

$$\mathsf{K} = \mathsf{K}_1\mathsf{K}_2 = \begin{pmatrix} \tau & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \ .$$

Transforming the vanishing points by K_1:

$$\mathbf{v}'_p \sim \mathsf{K}_1^{-1}\mathbf{v}_p$$

we obtain points $\mathbf{v}'_p$ for which the calibration equation (1) takes on the simple form:

$$(\mathbf{v}'_1)^\mathsf{T} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f^2 \end{pmatrix} \mathbf{v}'_2 = 0 \ . \tag{2}$$

The least squares solution for a set of equations (2) is given by:

$$f^2 = -\frac{\sum_{\mathbf{v}'_p \perp \mathbf{v}'_q} v'_{p,3} v'_{q,3} (v'_{p,1} v'_{q,1} + v'_{p,2} v'_{q,2})}{\sum_{\mathbf{v}'_p \perp \mathbf{v}'_q} (v'_{p,3} v'_{q,3})^2} \ .$$

Before solving for f, the $\mathbf{v}'_p$ should be normalized to unit norm.

What we observe is that vanishing points that lie on the ideal line in the image are useless for focal length calibration (the term $v'_{p,3}v'_{q,3}$ in the denominator is zero here). This means that the vanishing points that correspond to directions that are parallel to the image plane, are useless, so we need at least two finite vanishing points that are perpendicular. These considerations tell us how to position the camera to successfully calibrate it. Note that vanishing points at infinity will not badly influence the determination of the focal length: the term in the denominator will be zero and the term in the numerator very close to zero, so they won't affect the sums in the equation. Infinite vanishing points might be used for the determination of other intrinsic parameters, if required.

The images used in our experiments (see figures 3 and 4) only show small amounts of optical distortion. However, wide-angle views which are likely to be used for single-view 3D reconstruction, might require distortion removal prior to calibration and reconstruction. The automatic method by Devernay and Faugeras [2] might be used. Distortion removal can also be achieved in a very simple way by manually adjusting the dominant first coefficient of radial distortion, by the aid of a slider provided by the graphical user interface, such as to make line segments in the image roughly straight.

4 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.

We assume that vanishing points and lines have been computed where possible and that the image has been calibrated as described in the previous section. This enables us to backproject image points to 3D along their projection rays – a 3D point whose image is given by $\mathbf{q}$, has coordinates:

$$\mathbf{Q} = \begin{pmatrix} \lambda \mathbf{q}' \\ 1 \end{pmatrix} \quad , \tag{3}$$

where $\mathbf{q}' \sim \mathsf{K}^{-1}\mathbf{q}$ and $\mathbf{q}'$ has unit norm. The unknown λ expresses the distance of $\mathbf{Q}$ from the optical center and hence defines its position on the projection ray.

If we know the vanishing line of a 3D plane, its position is also defined up to one unknown. Let $\mathbf{l}$ be the vanishing line and $\mathbf{n} \sim \mathsf{K}^{\mathsf{T}}\mathbf{l}$ such that $\mathbf{n}$ has unit norm. Then, the 3D position of any plane whose vanishing line is $\mathbf{l}$, is given by:

$$\Pi = \begin{pmatrix} \mathbf{n} \\ d \end{pmatrix} \quad . \tag{4}$$

The vector $\mathbf{n}$ is the plane's normal and d the plane's distance from the optical center.

Unless a reference distance in the scene is known, 3D reconstruction can be achieved up to a global scale factor only. Hence, we are free to set the position of one point (along its projection ray) or one plane (while preserving its vanishing line). Suppose, we have fixed one point $\mathbf{Q}$. The position of planes with known vanishing lines and containing $\mathbf{Q}$ is then completely defined. Other points lying on these planes may then be reconstructed, by simply intersecting the projection rays with the planes. In turn, other planes may then be reconstructed using the available 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.

In the following, we describe an extension of this basic reconstruction scheme. Basically, we bootstrap the alternating point-plane reconstruction scheme via the simultaneous reconstruction of a set of points and a set of planes that are linked together in a way described below. This is the central part of our reconstruction method. Other modules used for reconstruction are described in §4.2. The complete algorithm is given in §5 and the way it works is illustrated in §6.

4.1 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 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 (cf. the example in §6).

Consider sets of images of coplanar points, $S_r = \{\mathbf{q}_{i_{r,1}}, \ldots, \mathbf{q}_{i_{r,n_r}}\}$. A point may belong to more than one set S_r. Let Π_r be the plane corresponding to the set S_r. In the following, we only consider planes with known vanishing lines.

We say that two planes Π_{r_1} and Π_{r_2} are connected if they share a point, i.e. if the intersection of S_{r_1} and S_{r_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'_r 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 (3) and (4). Let $\mathbf{Q}$ be a point lying on plane Π. The squared distance between them is given by:

$$\left(d + (\mathbf{n}^\mathsf{T}\mathbf{q}')\lambda\right)^2 \quad .$$

We want to minimize the sum of squared distances for pairs of planes and points. The cost function is thus:

$$g = \sum_r \sum_{p=1}^{n_r} \left(d_r^2 + 2\left((\mathbf{n}_r)^\mathsf{T}\mathbf{q}'_{i_{rp}}\right)\lambda_{i_{rp}} d_r + \left((\mathbf{n}_r)^\mathsf{T}\mathbf{q}'_{i_{rp}}\right)^2 \lambda_{i_{rp}}^2 \right) \quad .$$

Its partial derivatives are (divided by 2):

$$\frac{\sigma g}{\sigma d_r} = \left(\sum_{p=1}^{n_r} 1\right) d_r + \sum_{p=1}^{n_r} \left((\mathbf{n}_r)^\mathsf{T}\mathbf{q}'_{i_{rp}}\right)\lambda_{i_{rp}}$$

$$\frac{\sigma g}{\sigma \lambda_p} = \sum_{r, \mathbf{q}_p \in S'_r} \left((\mathbf{n}_r)^\mathsf{T}\mathbf{q}'_p\right) d_r + \left(\sum_{r, \mathbf{q}_p \in S'_r} \left((\mathbf{n}_r)^\mathsf{T}\mathbf{q}'_p\right)^2\right)\lambda_p$$

Nullifying these leads to a homogeneous linear equation system in the unknowns d_r 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 following nice structure:

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

where:

$$D_r = \sum_{p=1}^{n_r} 1 \qquad C_{rp} = (\mathbf{n}_r)^\mathsf{T} \mathbf{q}'_p \qquad L_p = \sum_{r, \mathbf{q}_p \in S'_r} \left((\mathbf{n}_r)^\mathsf{T} \mathbf{q}'_p \right)^2$$

Special sparse solution methods may be used like e.g. in [3], 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.

4.2 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 onto a plane is done by computing λ_p via:

$$\lambda_p = -\frac{d_r}{(\mathbf{n}_r)^\mathsf{T} \mathbf{q}'_p} \quad .$$

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):

$$g = \sum_{p=1}^{n} \left(d^2 + 2 \left(\mathbf{n}^\mathsf{T} \mathbf{q}'_p \right) \lambda_p d + \left(\mathbf{n}^\mathsf{T} \mathbf{q}'_p \right)^2 \lambda_p^2 \right) \quad . \tag{5}$$

Nullifying the partial derivatives leads to a linear homogeneous equation system in the unknowns d and $\mathbf{n}$. If we already know the plane's normal $\mathbf{n}$, we obtain the following closed form solution for the unknown d:

$$d = -\frac{\sum_{p=1}^{n} \left(\mathbf{n}^\mathsf{T} \mathbf{q}'_p \right)}{\sum_{p=1}^{n} 1} \quad .$$

5 Complete Algorithm

1. Compute vanishing points and lines (cf. §2.2).
2. Calibrate (cf. §3).
3. Backproject all points up to scale, i.e. compute the vectors $\mathbf{q}'_p \sim \mathsf{K}^{-1}\mathbf{q}_p$. Scale the $\mathbf{q}'_p$ to unit norm and use extended coordinates for 3D points: $\mathbf{Q}_p^\mathsf{T} = (\lambda_p(\mathbf{q}'_p)^\mathsf{T}, 1)$.
4. From vanishing lines, compute plane normals (cf. equation (4)).
5. Partition the planes with known normal in sets of planes which are connected by at least one point (in a transitive manner).
6. Choose the largest partition.
7. Reconstruct plane and point positions (distances from origin) as described in §4.1. Use only points that lie on more than one plane in the actual partition.
8. Backproject points that lie on exactly one plane in the actual partition (cf. §4.2).
9. Reconstruct a plane not reconstructed yet by fitting it to 3D points (cf. §4.2). Each point provides one equation and a vanishing line two. Choose the plane with the most equations.
10. Backproject points lying on the plane just reconstructed.
11. If there are planes not reconstructed yet, go to step 9.

We may optimize the reconstruction, respecting the geometric constraints. We have coded such a bundle adjustment procedure, but in practice there is virtually no improvement in the quality of the reconstruction, so we usually omit this step. From the 3D reconstruction, we automatically create textured VRML models (see examples in §6).

6 Sample Run and Examples of 3D Models

Figure 1 explains the user-provided input to our algorithm for the example shown in figure 3. On the left hand side, the different directions present in the 3D object are represented via the dotted line segments which are used for computing the vanishing points. Additionally, the user should flag perpendicular directions. On the right hand side, 5 groups of parallel planar patches are shown, i.e. groups of patches sharing the same vanishing line. The edges in the middle of the graph show which directions "belong" to which groups of planes. For example, for each of the second, third and fifth groups of planes, we have two vanishing points, allowing us to compute the vanishing lines.

Some of the steps taken by the reconstruction algorithm for our example are shown in figure 2. The upper left figure shows the result of the initial step described in §4.1. Note that a large part of the object is already reconstructed. The upper right figure shows points that are backprojected onto the reconstructed planes (cf. step 8 of the algorithm). The subsequent rows of figures show on the left a reconstructed plane (dark) and the (bold) points used to reconstruct it (step 9 of the algorithm). On the right, backprojected points are shown using bright circles (step 10). The reconstruction of the whole 3D model for this example required 2 additional steps of alternating plane-point reconstruction, not shown here.

Figures 3 and 4 on page 10 show examples of 3D models obtained with our method. The first image in each figure is the original image from which reconstruction was obtained. The other images show rendered views of created VRML models. Texture maps were taken from the original images. For the model shown in figure 4, additional texture maps, taken from frontoparallel views of two of the walls were used to enhance resolution.

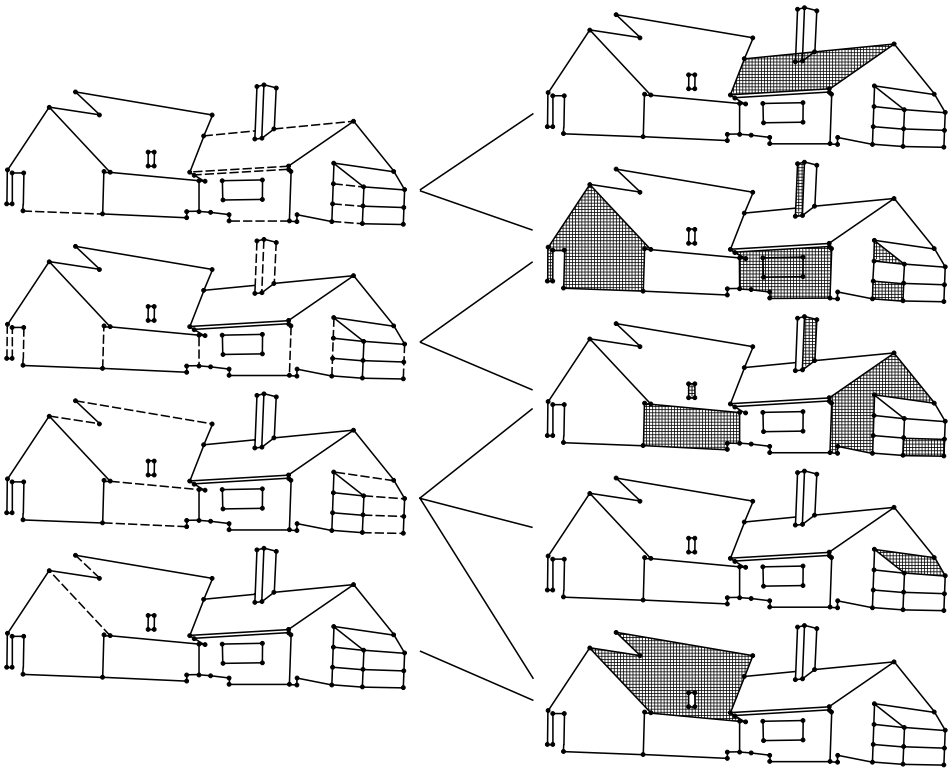

Figure 1: User-provided input: directions of lines and planes.

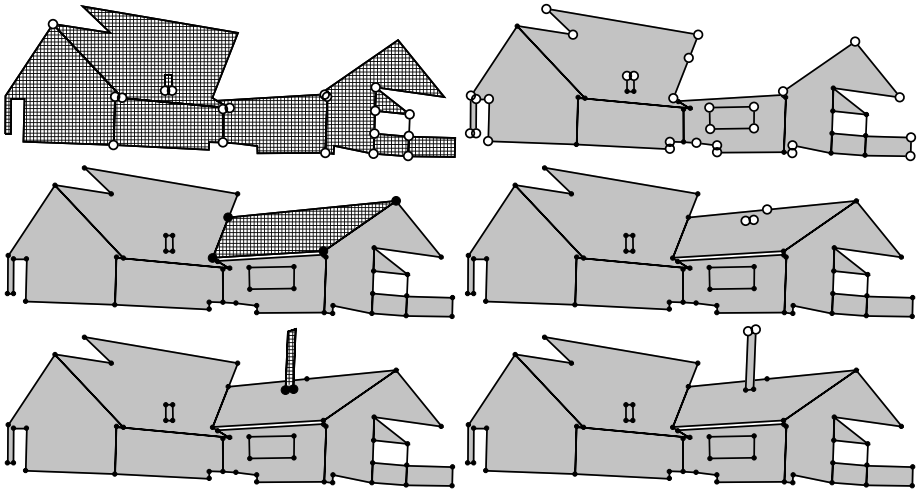

Figure 2: First 6 steps of the reconstruction process.

7 Conclusion

We have presented a method for interactive 3D reconstruction of piecewise planar objects from a single view. Camera calibration and 3D reconstruction are 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.

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 vanishing 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 like [5].

There are several extensions that may be added to our basic method. For example, other primitives than points and planes might be used, like lines, spheres or cylinders. Other types of geometrical constraints, like e.g. ratios of distances, can be added, enlargening the class of objects that can be reconstructed. Also, it might be worth trying to stitch together two or more 3D models obtained from single, possibly non-overlapping views (e.g. from the back and the front of a house), to get complete 3D models.

We already adapted our method to the use of panoramic images, obtained using a parabolic mirror. Thus, we are able to create 360° 3D models from one image, usually of the interior of a room.

Please contact the first author to get hard copies with color figures and images.

References

[1] B. Caprile and V. Torre, "Using Vanishing Points for Camera Calibration," *International Journal on Computer Vision*, Vol. 4, pp. 127–140, 1990.

[2] F. Devernay and O.D. Faugeras, "Automatic calibration and removal of distortion from scenes of structured environments," *Proceedings of the* SPIE *Conference on Investigate and Trial Image Processing, San Diego, California, USA*, Vol. 2567, SPIE - Society of Photo-Optical Instrumentation Engineers, August 1995.

[3] 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.

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

[5] D. Liebowitz, A. Criminisi and A. Zisserman, "Creating Architectural Models from Images," *Proceedings EuroGraphics*, to appear, September 1999.

[6] 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.

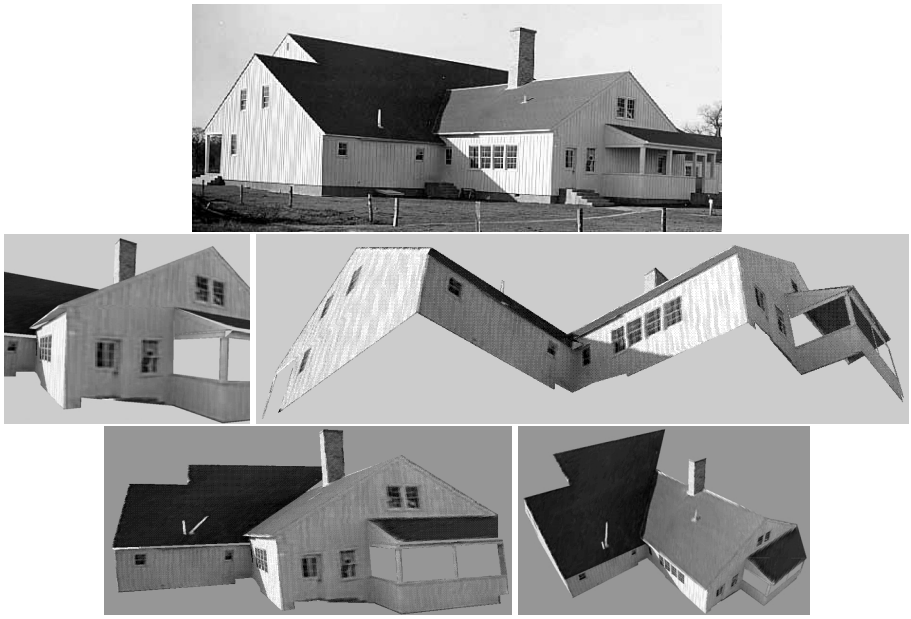

Figure 3: Original image and rendered views of 3D VRML model.

Figure 4: Original image and rendered views of 3D VRML model.