An Interactive System for Constraint-Based Modelling

Duncan Robertson and Roberto Cipolla
Department of Engineering
University of Cambridge
Cambridge , UK
dpr20@eng.cam.ac.uk

Abstract

Interactive techniques for geometric scene modelling typically give good results at the expense of considerable user intervention. This paper describes a working, interactive modelling system that allows the user to build models quickly. Using a few, poorly localised feature correspondences the system generates an initial guess at projection matrices and scene structure that is used as a basis for subsequent automatic matching and triangulation. In addition, the system provides an entirely flexible constraint-based reconstruction strategy that can be used to model parallelism and orthogonality constraints on line directions and plane normals. The working application (called 'PhotoBuilder') can be downloaded via the web page:
http://svr-www.eng.cam.ac.uk/PhotoBuilder/

1 Introduction

There are various approaches to the problem of obtaining geometric models from images [1,2] and extended image sequences [3]. The simplest systems[1] use conventional photogrammetry to obtain a 3D, wire-frame model from feature correspondences defined by hand. This approach has two disadvantages: (i) it is only possible to model polygons entirely visible from at least two viewpoints, and (ii) the accuracy of the model is sensitive to the accuracy of the feature correspondences. In contrast, constraint-based modelling techniques allow improved accuracy and single-view reconstruction by the application of the user's prior knowledge of scene constraints (such as parallelism and orthogonality). Such approaches are readily applicable to architectural scenes. Several constraint-based schemes have been described. [1] is a scheme based on primitives (simple 3D shapes such as prisms and pyramids)[2]. The principle disadvantage of primitive-based schemes is that many real-world scenes cannot be effectively decomposed into such simple geometric shapes. Furthermore, it is frequently impossible to find viewpoints such that whole primitives are visible in a single photograph, particularly where aerial views are unavailable. A more flexible approach to constraint-based modelling is described in [2] (and extended in [4]). This system allows the application of various kinds of constraint without the use of

[1] See http://www.photomodeler.com

[2] A good example of a working system based on this approach is available at http://www.canoma.com

BMVC 2000 doi:10.5244/C.14.54

primitives although it has the disadvantage that line directions and plane normals must be determined before reconstruction using parallel lines defined in the image.

These constraint-based approaches have the disadvantage that they require significant user intervention and the modelling process is time-consuming and slow. In this paper we describe a complete geometric modelling system that uses automation to significantly reduce the time taken to build models, and (optionally) provides an entirely flexible constraint-based reconstruction strategy that can be used to model parallelism and orthogonality constraints on line directions and plane normals.

2 Geometric Framework

From point correspondences $\mathbf{x}_{ij}$ defined in a sequence of images we aim to recover camera projection matrices P_i and scene structure $\mathbf{X}_j$. For a pinhole projective camera, perspective projection from Euclidean 3-space to an image can be conveniently represented in homogeneous co-ordinates by a 3 × 4 camera projection matrix P:

$$\lambda_{ij}\mathbf{x}_{ij} = P_i\mathbf{X}_j \tag{1}$$

where $\mathbf{x}_{ij} = (u, v, 1)^T$, $\mathbf{X}_j=(x, y, z, 1)^T$, and $\mathbf{x}_{ij}$ is the projection of the jth vertex in the ith image. The projection matrix has 11 degrees of freedom and can be decomposed into the orientation and position of the camera relative to a world co-ordinate system:

$$P_i = K_i\left[R_i \mid \mathbf{T}_i\right] \tag{2}$$

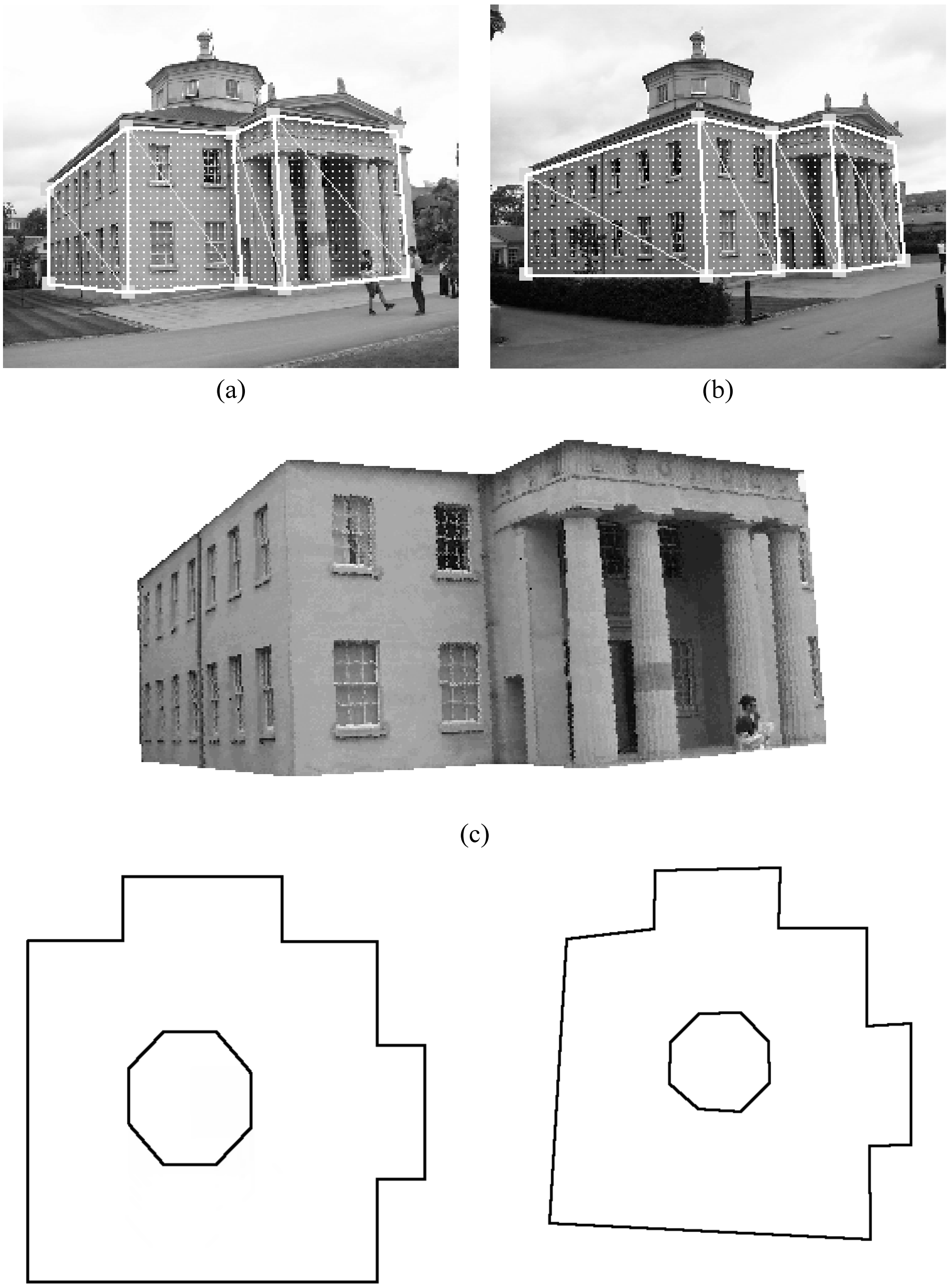
and a 3 × 3 camera calibration matrix K_i, corresponding to the following image plane transformation:

$$K_i = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

We have found the following three-stage modelling sequence to be the most effective. Firstly, an intial guess at camera calibration parameters K_i is determined using vanishing points (Section 2.1). Next projection matrices P_i and scene structure $\mathbf{X}_j$ are determined by ray bundle adjustment (Section 2.2). Finally, scene structure is enhanced using constraints (Section 2.3).

2.1 Camera Calibration

A number of solutions have been proposed to the problem of estimating camera intrinsic parameters K_i. In [5], focal lengths for three or more cameras are estimated by making assumptions of zero skew, square pixels, and principal point in the image centre. Alternatively, based on [6], we can determine up to three camera intrinsic parameters for a single camera using vanishing points according to the following algorithm:



(a) (b)

(c)

(d) (e)

Fig 1: Modelling proceeds in three stages. Firstly camera calibration parameters are determined using the vanishing points of orthogonal sets of parallel lines (a,b). Then scene structure determined using ray bundle adjustment (c). Finally constraints are applied. An outline of a roof is shown after reconstruction (d) with and (e) without constraints.

1. The user identifies three sets of parallel lines in the image, corresponding with three orthogonal directions in the world (see Figure 1(a,b)).

2. The vanishing point for each set of parallel lines is computed.

3. Vanishing points $(u_l, v_l, 1)^{\mathrm{T}}$ corresponding to three orthogonal directions in the world provide the following constraint on $K_iK_i^{\mathrm{T}}$:

$$\begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^{\ 2} & 0 & 0 \\ 0 & \lambda_2^{\ 2} & 0 \\ 0 & 0 & \lambda_3^{\ 2} \end{bmatrix} \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{bmatrix}^T = K_i K_i^{\ T} \qquad (4)$$

   Under the assumption of known aspect ratio and zero skew, this equation can be rewritten to as 6 linear equations (from 6 elements of a symmetric matrix) and solved to recover up to 3 camera intrinsic parameters and the unknown scale factors $\lambda_l^{\ 2}$.

This technique is degenerate for images where two vanishing points lie at infinity. In practice, this is not a significant problem since camera calibration need only be estimated for two viewpoints in order to initialise the reconstruction process (see Section 2.2).

2.2 Reconstruction

The next stage of the modelling process is to define image feature correspondences, automatically or by hand. Ray bundle adjustment is used to optimise camera parameters and scene structure given a good initial guess (Figure 1(b)). We seek to minimise back-projection error for our reconstructed vertices according to the following criteria:

$$\min_{P_i, \mathbf{X}j} \sum_i \sum_j \left\| \mathbf{x}_{ij} - P_i \mathbf{X}_j \right\|^2 \qquad (5)$$

An initial guess at camera external parameters R_i and $\mathbf{T}_i$, and scene structure $\mathbf{X}j$ is determined using an approach similar to that suggested in [7] according to the following algorithm:

1. Choose two views with known camera calibration K_i. Without loss of generality, we may arbitrarily assign the first of these two viewpoints to be the origin of our world co-ordinate system with projection matrix $K_i[I \mid 0]$.

2. Compute an essential matrix for the two views using feature correspondences (see [9]). Decompose the essential matrix into R and $\mathbf{T}$, which provide the projection matrix for the second view. Compute an initial guess at (partial) 3D structure by triangulation (see [10]).

3. Bundle adjust the partial reconstruction to improve projection matrix and structure estimates according to (5). The bundle adjustment strategy we use is based on that described in [8] although we use the Levenberg-Marquardt optimisation technique and additionally optimise some camera intrinsic parameters.

4. Repeat for each remaining view:

i. Determine an approximate projection matrix for the view using the direct linear transformation (DLT) method described in [11]. The new viewpoint allows more scene structure to be determined.
ii. Bundle adjust the partial reconstruction to refine projection matrix and structure estimates.

In practice, inaccurate and few feature matches mean that the order in which remaining views are integrated in step 4(i) is critical since some views may be relatively more ‘degenerate’ than others. Degeneracy occurs, for example, (i) where all vertices lie on a plane parallel with the image plane, or (ii) where features are concentrated in a small part of the image. By solving the DLT using the singular value decomposition, we obtain a good guide to the degeneracy of the solution. Integrating views in order of the number of features already reconstructed is not, in general, a good strategy. We have found that integrating views in a ‘most degenerate last’ order means the reconstruction algorithm converges more quickly in the presence of noise.

2.3 Constrained reconstruction

Some scenes exhibit significant orthogonality and parallelism and in some cases it may be desirable to apply these constraints to the resulting 3D model. Constraints reduce the sensitivity of a model to inaccuracy in the feature correspondences that define its vertices. Furthermore, constraints allow single-view reconstruction and reconstruction of ‘hidden’ vertices that are not visible in the image but can be inferred from knowledge of parallelism and orthogonality.

Our approach to constraint application is an extension to [2]. This approach uses parallel lines defined in the image plane to determine camera pose, plane orientations, and line directions. This allows linear solution for constrained 3D structure and camera position. However, given sufficiently many feature correspondences, camera pose and position can be estimated more accurately by ray bundle adjustment (see Section 2.2). ‘Best-fit’ plane orientations and line directions are then estimated from 3D structure.

Our constraint application algorithm is as follows:

1. Compute projection matrices and structure from feature correspondences using bundle adjustment (see 3.2).

2. Group vertices into lines and planes. Group lines and planes with parallel directions and normals into parallel sets. Group parallel sets into orthogonal sets.

3. Determine a best-fit set of orthogonal directions using the following technique:
   i. From 3D structure estimate, determine best-fit directions for each parallel set using linear least squares.
   ii. Group each set of three approximately orthogonal unit direction vectors into a matrix R.
   iii. Find R' that minimises the Frobenius norm $||R-R'||$ subject to the condition that R' is an orthonormal rotation matrix. This can be simply achieved using singular value decomposition. Let UWV^T be the singular value decomposition of R. Then R' is obtained from $UW'V^T$ where $W'=diag(1,1,1)$.
   iv. Non-linear optimisation using Levenberg-Marquardt.

4. Compute constrained reconstruction by linear least squares.

Given best-fit directions $\mathbf{m}_k$, we estimate scene structure as the solution of a linear equation. Each model constraint provides additional rows in this equation. Table 1 lists the constraints that are employed.

Type	Constraint	n
Image co-ordinate	$(K^{-1}\mathbf{x}-R^{T}\mathbf{T})\times\mathbf{X}=0$	2
Line direction	$(\mathbf{X}_2-\mathbf{X}_1)\times\mathbf{m}_k=0$	2
Triangle normal	$(\mathbf{X}_2-\mathbf{X}_1)\cdot\mathbf{m}_k=0$	2
	$(\mathbf{X}_3-\mathbf{X}_1)\cdot\mathbf{m}_k=0$	
Known 3D co-ordinate	$\mathbf{X}$	3

Table1: Modelling constraints (n is number of independent equations)

We can formulate all such constraints as a linear equation:

$$A\mathbf{X}=\mathbf{b} \tag{5}$$

where $\mathbf{X}$ is a vector comprising the unknown vertex co-ordinates $\mathbf{X}_j$. We may optionally give different constraints different weights within this matrix equation. In practice, we weight the line direction and triangle normal constraints about 10 times higher than the others to ensure square-looking models. Solution is possible via a variety of sparse matrix techniques (or for less than about 200 vertices) by singular value decomposition.

3 Implementation

We have developed a complete, interactive 3D modelling system ('PhotoBuilder') based on the framework described in Section 2. This system allows the user to define by hand image features corresponding to significant geometric vertices in the scene. Pairs of vertices may be joined with lines and triplets of vertices may be grouped to make triangles; these will form the basis of a texture-mapped wire-frame model. Lines and triangles may be grouped into parallel sets, which, in turn may be grouped into orthogonal sets. Optionally, this information is used as a basis for constrained reconstruction (see Section 2.3).

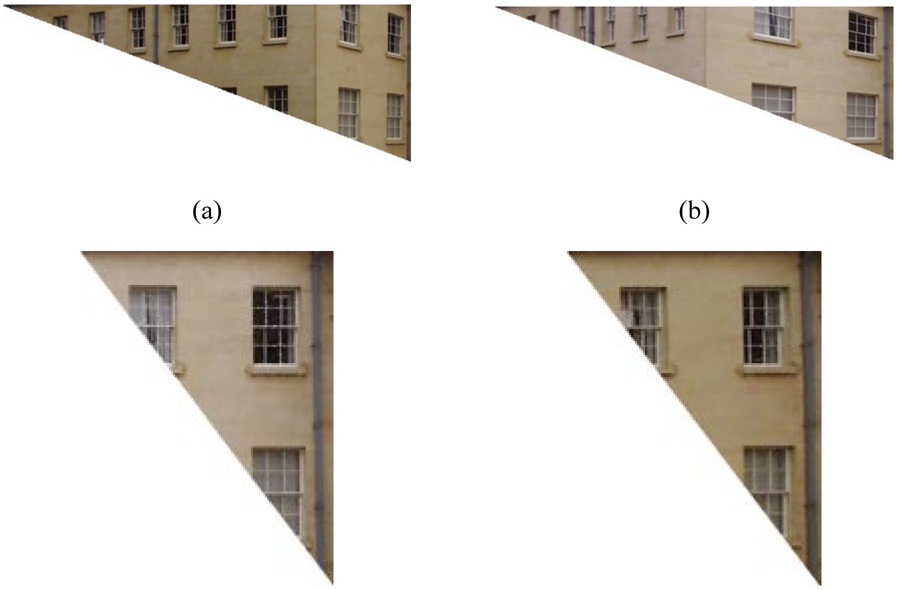
Two additional improvements significantly reduce the time taken to build models: (i) an automatic matching and wire-frame triangulation tool (Section 3.1), and (ii) a guided matching tool used to detect and match extra features automatically. This allows improved estimation of projection matrices (Section 3.2).

3.1 Automatic matching and wire-frame triangulation

User interaction considerably simplifies the problem of 3D modelling. However, large-scale, by-hand feature detection and matching is unreliable, inaccurate, and slow. From an initialisation of a few feature matches identified by hand, our system can determine remaining feature matches and wire-frame triangulation automatically.

Around 10-15 matches are sufficient to allow us to estimate projection matrices using the method described in Section 2.2. Given these estimates, the task is to choose remaining feature matches and wire-frame triangulation such that the resulting model will 'agree' well with image data. This is framed as a search problem. Correctly matched triangles should be warped by an homography from view to view (see Figure 2).

This search is significantly constrained by the knowledge that feature matches must lie on the epipolar line and that triangles of texture will not overlap. Nevertheless, the search space is still large since there are ${}^{n}C_3$ ways of choosing a candidate triangle from n candidate vertices. We limit the problem by allowing the user to assign a maximum of 20 or so features at a time for matching and triangulation. Multi-resolution comparison is used: the majority of possible triangle matches can be ruled out quickly by low resolution comparison. Higher resolution comparison is only used when triangle matches cannot be ruled out at lower resolution. In practice, our system can match 20 or so triangles quite reliably in a few seconds.



(a) (b)

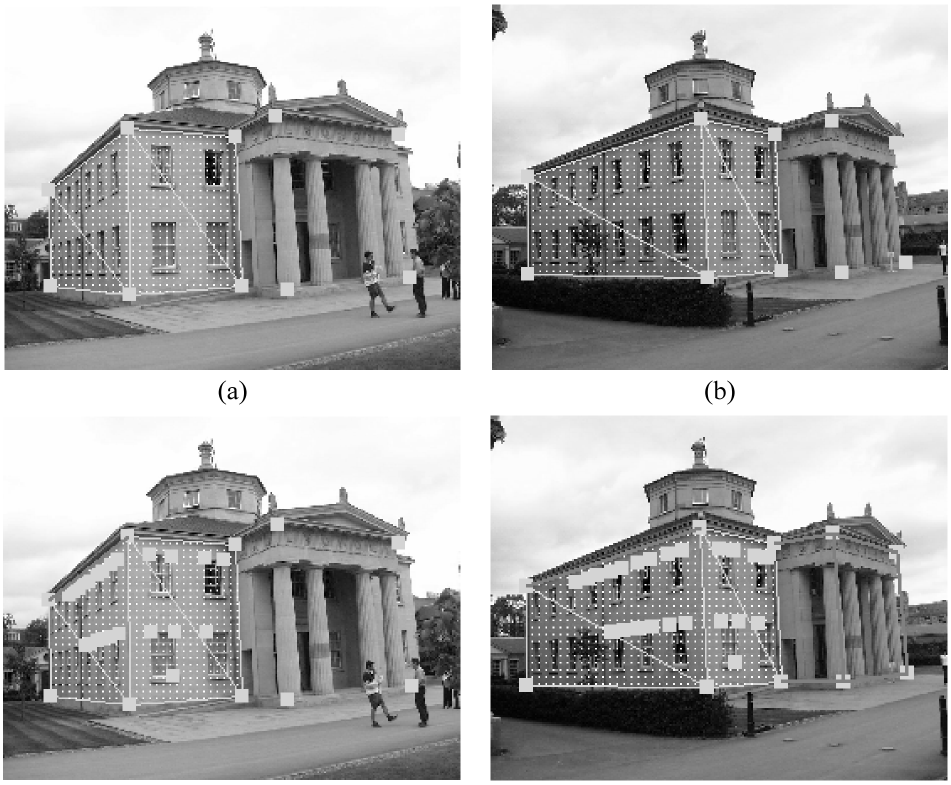
(c) (d)

Figure 2: Texture is projected onto two candidate triangles from two different viewpoints. The first triangle (a,b) does not lie on a plane in the scene and thus projected texture is highly viewpoint dependent. The second triangle (c,d) does lie on a plane surface. RMS pixel intensity errors are (a,b) 52 and (c,d) 34.

3.2 Guided matching

For a given amount of measurement noise, a larger number of feature correspondences gives a more accurate estimate of projection matrices and scene structure. To reduce the amount of time that the user spends identifying feature correspondences by hand, our system detects some feature correspondences automatically. Using a coarse model of the scene (in the form of a texture mapped wire-frame), we can use guided matching to detect extra matches for features lying on (or near to) the surface of the model.

The system uses the corner detector described in [12] to detect strong corner features lying within a matched triangle in two images. Since the triangle should correspond to a plane in the scene, our knowledge of projection matrices and structure allows us to estimate the mapping between points lying within the triangle in the two images. In practice, not all features will be matched since uncertainty in the estimate means there may be more than one possible match for each. Therefore we only use those features that can be matched unambiguously (see Figure 3).



(a) (b)

(c) (d)

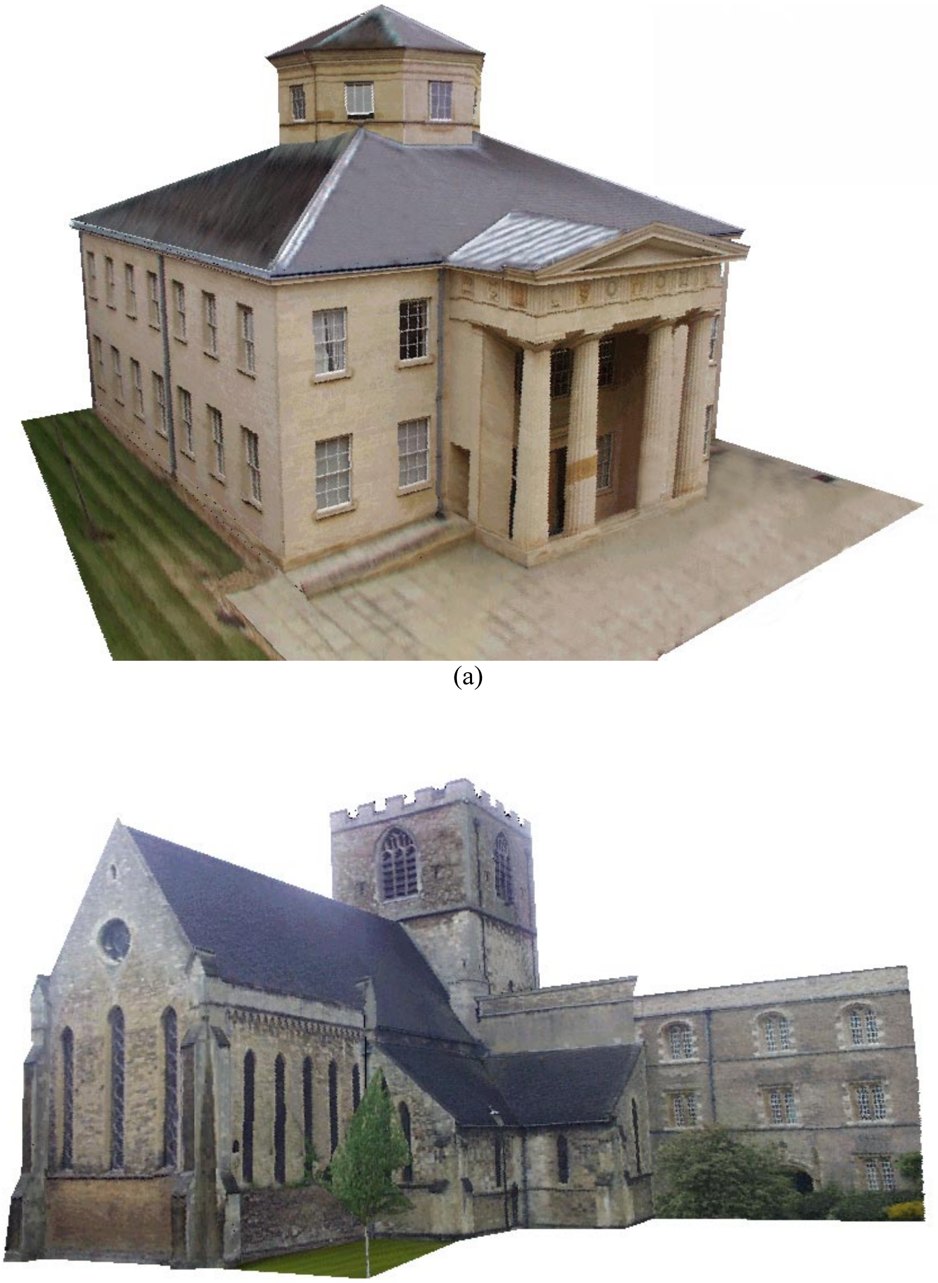
Figure 3: (a,b) Two images with 10 feature correspondences and four triangles, (c,d) 23 extra feature correspondences generated automatically using guided matching.

4 Evaluation

Experiments have been conducted using photographs of a large variety of architectural scenes. Figure 1 shows two out of a set of eight photographs of Downing College Library obtained using an Olympus digital camera. An initial guess for camera calibration was obtained using three sets of parallel lines. Calibration parameters for all the images in the set were then determined accurately by bundle adjustment. Figure 3(a) shows the resulting 3D model.

5 Conclusions

We have presented a method for obtaining geometric scene models from uncalibrated images obtained from a sparse set of viewpoints. Our working, interactive modelling system uses automation to considerably reduce the amount of user intervention required to build models and to improve the accuracy of projection matrix estimates. In addition, the system provides an effective constraint application strategy for use where there is significant prior knowledge of orthogonality and parallelism.



(a)

(b)

Figure 3: 3D models output in VRML format and viewed from new viewpoints. These models were reconstructed using constraints to ensure square corners and to allow reconstruction of points visible in only one image (e.g. the ground plane).

References

[1] P.E. Debevec, C.J. Taylor, and J. Malik. *Modelling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach*. In A CM Computer Graphics (Proceedings SIGGRAPH), pages 11-20, 1996.

[2] H-Y. Shum, M. Han, and R. Szeliski. *Interactive Construction of 3D Models from Panoramic Mosaics*. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, pages 427-433, Santa Barbara, (June) 1998.

[3] P. Beardsley, P. Torr, and A. Zisserman. *3D Model Acquisition from Extended Image Sequences*. In Proc. 4th European Conf. on Computer Vision, Cambridge (April 1996); LNCS 1065, volume II, pages 683-695, Springer-Verlag, 1996.

[4] P. F. Sturm and S. J. Maybank. *A Method for Interactive 3D Reconstruction of Piecewise Planar Objects from Single Images*, In Proc. British Machine Vision Conf., volume I, pages 265-274, 1999.

[5] M. Pollefeys, R. Koch, and L. Van Gool. *Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters*. In Proc. 6th Int. Conf. on Computer Vision, Mumbai, India, 1998.

[6] R. Cipolla, T. Drummond and D. Robertson. *Camera calibration from vanishing points in images of architectural scenes*, In Proc. British Machine Vision Conf., volume II, pages 382-392, 1999.

[7] R.I. Hartley. *Euclidian reconstruction from uncalibrated views*. In J.L. Mundy, A. Zisserman, and D. Forsythe, editors, *Applications of Invariance in Computer Vision*, volume 825 of Lecture notes in Computer Science, pages 237-256, Springer-Verlag, 1994.

[8] C.Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, USA, 4th edition, 1980.

[9] R.I. Hartley. *In defence of the 8-point algorithm*. In Proc. International Conference on Computer Vision, pages 1064-1070,1995.

[10] R.I. Hartley and P. Sturm. *Triangulation*. In American Image Understanding Workshop, pages 957-966, 1994.

[11] I.E. Sutherland. *Three dimensional data input by tablet*. Proceedings of IEEE, Vol 62, No. 4:453-461, April 1974.

[12] C.J. Harris and M. Stephens. *A combined corner and edge detector*. In Alvey Vision Conf., pages 147-151, 1988.