3D Model Acquisition from Uncalibrated Images

Roberto Cipolla Edmond Boyer
Department of Engineering,
University of Cambridge,
Cambridge CB2 1PZ, UK.

Abstract

In this paper we address the problem of recovering 3D models from uncalibrated images of architectural scenes.

We propose a simple, geometrically intuitive method which exploits strong rigidity constraints such as parallelism and orthogonality present in indoor and outdoor architectural scenes. We show how these simple constraints can be used to calibrate the cameras and to recover the 3×4 projection matrices for each viewpoint. The projection matrices are used to recover partial 3D models of the scene and these can be used to visualise new viewpoints.

Our approach does not need any *a priori* information about the cameras being used. A working system has been designed and implemented to allow the user to interactively build a 3D model from a pair of uncalibrated images from arbitrary viewpoints.

1 Introduction

Considerable efforts have been made to recover photorealistic models of the real world. The most common *geometric* approach is to attempt to recover 3D models from calibrated stereo images [16] or uncalibrated extended image sequences [21, 1, 17] by triangulation and exploiting epipolar [15] and trilinear constraints [10, 20]. An alternative approach consists of visualisation from image-based representations of a 3D scene. This is the preferred method when trying to generate an intermediate viewpoint image given two nearby viewpoints since it does not need to make explicit a 3D model of the scene [22, 19, 7, 12, 6].

Facade [4] – one of the most successful systems for modelling and rendering architectural buildings from photographs – consists of a hybrid geometric and image-based approach. Unfortunately it involves considerable time and effort from the user in decomposing the scene into prismatic blocks, followed by the estimation of the pose of these primitives.

In this paper we propose a much simpler approach to construct a 3D model and generate new viewpoint images by exploiting strong constraints present in the scenes to be modelled. In the context of architectural environments, the constraints which can be used are parallelism and orthogonality. These constraints lead to very simple and geometrically intuitive methods to calibrate the cameras and to recover Euclidean models of the scene from only two images from arbitrary positions.

2 Outline of the algorithm

A 3D model can be recovered from two or more uncalibrated images. Our algorithm consists of the four following stages (see figure 1).

1. We first define a set of primitives – segments and cuboids – for which parallelism and orthogonality constraints are derived. These primitives are precisely localised in the image using the image gradient information.

2. The next step concerns the camera calibration: the intrinsic parameters of the camera are determined for each image. This is done by determining the vanishing points associated with parallel lines in the world. Three mutually orthogonal directions are exploited [2].

3. The motion (a rotation and a translation) between the viewpoints is then computed. The motion combined with the knowledge of the camera parameters allows the recovery of the perspective projection matrices for each viewpoint.

4. The last step consists in using these projection matrices to find more correspondences between the images and then to compute 3D textured triangles that represent a model of the scene.

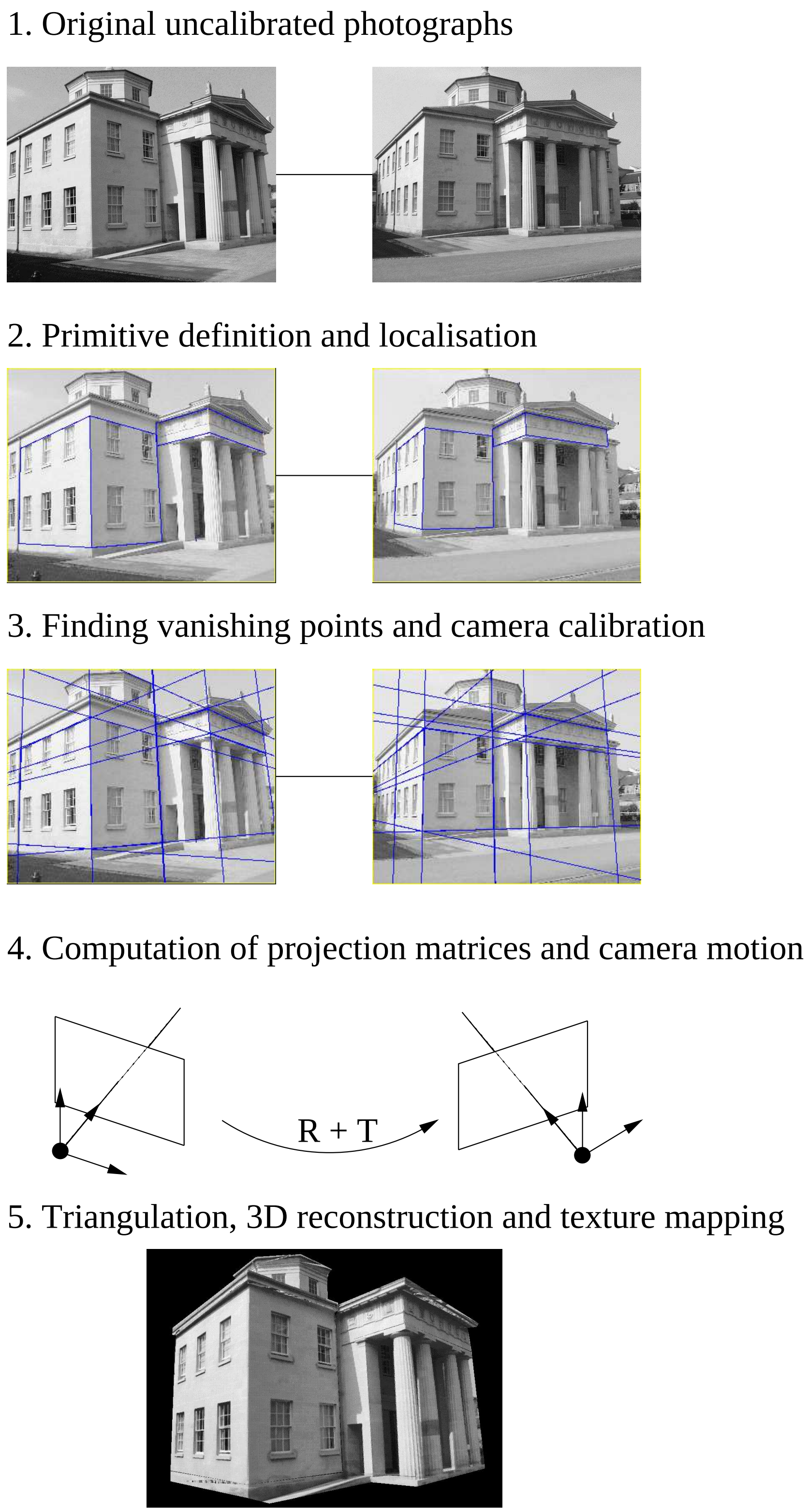

Figure 1: Outline of the algorithm.

3 Geometric Framework

For a pin-hole camera, perspective projection from Euclidean 3-space to an image can be conveniently represented in homogeneous coordinates by a 3×4 camera projection matrix, $\mathbf{P}$:

$$\begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \quad (1)$$

The projection matrix has 11 degrees of freedom and can be decomposed into the orientation and position of the camera relative to a the world co-ordinate system (a 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation vector $\mathbf{T}$:

$$\mathbf{P} = \mathbf{C} \left[\begin{array}{cc} \mathbf{R} & \mathbf{T} \end{array} \right] \quad (2)$$

and a 3×3 camera calibration matrix, $\mathbf{C}$, corresponding to the following transformation:

$$\mathbf{C} = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

where α_u, α_v are scale factors; s is a skew parameter; and u_0, v_0 are the pixel coordinates of the principal point (the intersection of the optical axis with the image plane [5]).

In general, the parameters of the projection matrix can be computed from the measured image positions of at least six known points in the world. This matrix can then be decomposed by QR decomposition into camera and rotation matrices [5].

In our approach the vanishing points corresponding to three mutually orthogonal directions can be used to determine:

1. the camera calibration matrix, $\mathbf{C}$ under the assumption of zero skew and known aspect ratio [2].

2. the rotation matrix $\mathbf{R}$.

3. the direction of translation, $\mathbf{T}$.

We show that the 8 degrees of freedom of the projection matrix for this special case can be determined from three vanishing points corresponding to the projections of 3 points at infinity and a reference point. The projection matrix can thus be recovered from the projection of at least one arbitrary cuboid. Applying the algorithm to two views allows the Euclidean reconstruction of all visible points up to an arbitrary scale.

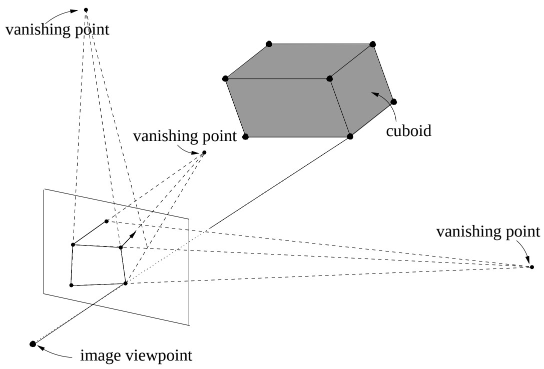

Figure 2: A cuboid and its projection which is defined by five image points and an image direction. The cuboid image edges define three vanishing points in the image plane.

4 Primitives

The first step of the method consists in defining image segments and cuboids by means of which the parallelism and orthogonality constraints can be derived. These constraints are then used to recover the intrinsic and extrinsic cameras parameters and hence the projection matrices for each view.

First, interest points are extracted in the reference views. This can be done by means of a Harris corner detector [8] and also interactively by hand since results from a corner detector may not be sufficient (see figure 3). Such points are then used to define the image projections of primitives: segments and cuboids, which are present in the observed scene.

The definition of a segment projection in an image is straightforward and requires two image points. The image of a cuboid is completely specified by six or more vertices. The interest points, extracted corners and user defined corners which are used to define segments and cuboids, are not necessarily well localised in the image plane. An optimisation is therefore performed to improve their localisations. This optimisation takes into account image gradient information and modifies image point coordinates in order to position primitive edges along image edges.

5 Finding Vanishing Points

Once an initial set of primitives has been defined, parallelism and orthogonality constraints are used through the vanishing points to constrain the intrinsic parameters.

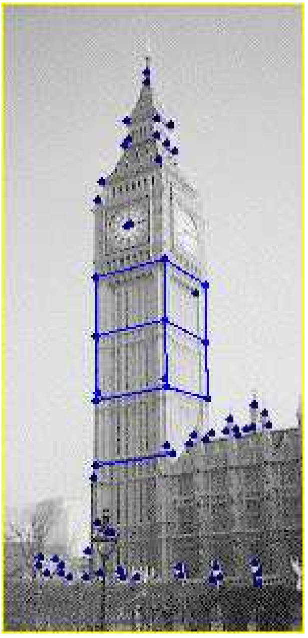

Figure 3: Example of primitive definitions and localisations. Two cuboids and one segment have been interactively defined using extracted corners.

A vanishing point corresponds to the projection of the intersection of parallel lines at infinity. A number of approaches have been proposed to localise precisely this intersection, from the simple calculation of a weighted mean of pairwise intersections [2] to more elaborate approaches involving noise assumption and non-linear criteria [14] (see also [11] for line intersection estimation). However, all these approaches are based on criteria which take into account image distances. Though trying to minimise the distance from the vanishing point to the image lines is geometrically correct, it appears to be numerically unstable in the presence of noise. In contrast, our approach is based on a linear criterion which optimises a three dimensional direction, the dual of a vanishing point in 3D space.

Let $l_1, .., l_n$ be a set of image lines corresponding to parallel space lines and let ν be the vanishing point defined by these lines. Then if we suppose that l_i and ν are expressed in homogeneous form in the image coordinate system (u, v, w), we have the following relation:

$$[l_1, .., l_n]^T \nu = 0. \tag{4}$$

and the null vector of the $3 \times n$ matrix $[l_1, .., l_n]$ gives a solution $\hat{\nu}$ for the vanishing point position. However, the minimised criterion does not necessarily have a physical meaning. By choosing an appropriate normalisation for $(l_1, .., l_n)$, the solution $\hat{\nu}$ can minimise the sum of the Euclidean distances from $\hat{\nu}$ to $[l_1, .., l_n]$. As said before, this solution will not be completely satisfactory in the presence of noise due to poor conditioning of the matrix $[l_1, .., l_n]$.

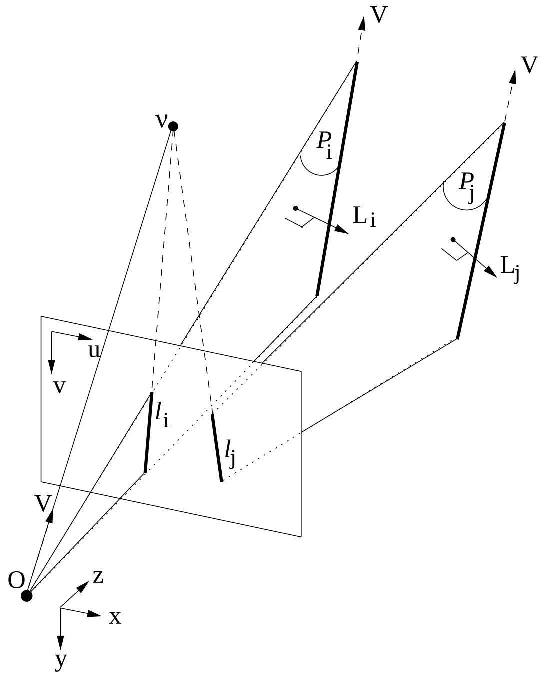

Figure 4: Vanishing point determination. The idea is to find the viewing direction V of the vanishing point ν. This direction should belong to all the planes $\mathcal{P}_i$.

Now suppose that the intrinsic parameters of the camera are known. Then image lines and points can be expressed using *normalised image coordinates*, that is to say in the coordinate system (x, y) associated with the camera retinal plane (see figure 4). Let $(L_1, .., L_n)$ and V be the homogeneous representations of $(l_1, .., l_n)$, ν in the retinal plane. Since V belongs to all the image lines, we still have the relation:

$$[L_1, .., L_n]^T V = 0. \tag{5}$$

However, this relation has also an interpretation in the three dimensional coordinate system (x, y, z) associated with the camera (see figure 4). Indeed, L_i is a vector orthogonal to the plane $\mathcal{P}_i$ spanned by the camera projection centre O and the image line l_i, and V is a three dimensional vector. This gives a physical meaning to the null vector $\hat{V}$ of $[L_1, .., L_n]^T$ which is the space direction closest to all planes

$(\mathcal{P}_1, .., \mathcal{P}_n)$.

Finally, we compute the space direction V as the null vector of the matrix $[L_1, .., L_n]^T$, where $|L_i| = 1$. This is done using a singular value decomposition of $[L_1, .., L_n]^T$ [18]. Experiments show that estimating a space direction is more robust to noise perturbations than estimating an image point. We use this method in the calibration routine described in the next section.

In the case where the camera's intrinsic parameters are not known, an intermediate solution consists of using (4) with prenormalized image point coordinates [9] in order to improve the condition number of the matrix $[l_1, .., l_n]$.

6 Camera Calibration

Having found the vanishing points we now show how they are used to derive constraints on the intrinsic parameters and the latter estimated in practice.

6.1 Constraints

A set of parallel line projections define a *vanishing point* ν in the image plane which is the projection of the intersection parallel lines at infinity. Let V be the space direction of these parallel lines (a unit vector). The vanishing point ν depends only on the direction of the line in space and not its position. The relationship between vanishing point in homogeneous coordinates, and the corresponding space direction in the camera coordinate system is given by:

$$\nu = \mathbf{C}V \qquad (6)$$

Now suppose that we have different vanishing points ν_i, ν_j corresponding to distinct space directions V_i, V_j separated by a known angle β. Then:

$$V_i^T V_j = \cos\beta$$

and from (6):

$$\frac{\nu_i^T \mathbf{C}^{-T}}{|\nu_i^T \mathbf{C}^{-T}|} \cdot \frac{\mathbf{C}^{-1}\nu_j}{|\mathbf{C}^{-1}\nu_j|} = \cos\beta \qquad (7)$$

which provides a constraint on the matrix $\mathbf{C}^{-T}\mathbf{C}^{-1}$ and thus on the intrinsic parameters. A particularly interesting case is when the space directions are orthogonal [23]. Then:

$$\nu_i^T \mathbf{C}^{-T}\mathbf{C}^{-1}\nu_j = 0 \qquad (8)$$

which gives a linear constraint on the parameters of $\mathbf{C}^{-T}\mathbf{C}^{-1}$. It is easy to show both algebraically and geometrically that the following degenerate cases exist:

1. if both directions V_i, V_j are parallel to the image plane then their projections give information on the angle θ between image plane axes u and v only[1].

2. if one of the directions V_i, V_j is orthogonal to the image plane, then their projections determin u_0 and v_0 but do not constrain the other parameters.

The faces of a cuboid define three orthogonal space directions, thus its projection leads to three constraints on the intrinsic parameters allowing the computation of three of them. As noticed before, degenerate cases exist. In particular, information on the parameters α_u and α_v can not be derived from a cuboid which has a face parallel to the image plance as shown in figure 5.

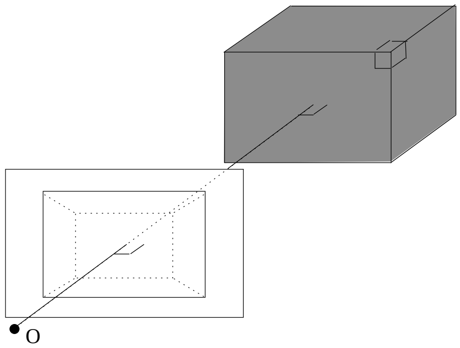

Figure 5: The degenerate case when the cuboid's projection does not provide any constraint on the intrinsic parameters α_u and α_v. One of the cuboid directions is perpendicular to the image plane.

6.2 Recovering intrinsic parameters

Now suppose that the skew parameter s is zero, then, for two vanishing points ν_i, ν_j associated with two orthogonal space directions, (8) be-

[1] θ is linked to the skew parameter by the relation $s = -\cot\theta/\alpha_v$ [5]

comes:

$$\nu_i^T \begin{bmatrix} \frac{1}{\alpha_u^2} & 0 & -\frac{u_0}{\alpha_u{}^2} \\ 0 & \frac{1}{\alpha_v}^2 & -\frac{v_0}{\alpha_v^2} \\ -\frac{u_0}{\alpha_u{}^2} & -\frac{v_0}{\alpha_v{}^2} & \frac{u_0{}^2}{\alpha_u{}^2} + \frac{v_0{}^2}{\alpha_v{}^2} + 1 \end{bmatrix} \nu_j = 0 \tag{9}$$

and if (u_i, v_i, w_i) are the homogeneous image coordinates of ν_i then:

$$\begin{aligned} \frac{(u_i - w_i u_0)(u_j - w_j u_0)}{\alpha_u^2} & + \\ \frac{(v_i - w_i v_0)(v_j - w_j v_0)}{\alpha_v^2} & + \\ w_i w_j &= 0. \end{aligned} \tag{10}$$

The expression above is linear in $\frac{1}{\alpha_u^2}$, $\frac{1}{\alpha_v^2}$ and therefore leads to linear estimation of these parameters if (u_0, v_0) are known. Degenerate cases appears when space directions associated with vanishing points belong to the image plane.

A cuboid defines three orthogonal directions, thus its projection allows, in general, the computation of three of the intrinsic parameters. In our current implementation, we determine the scale parameters from orthogonal directions using this method and under the assumption that the *principal point*, is known (assumed to be in the image centre). The vanishing points can, of course, be used to find the principal point [2] but the following method was found to be more reliable.

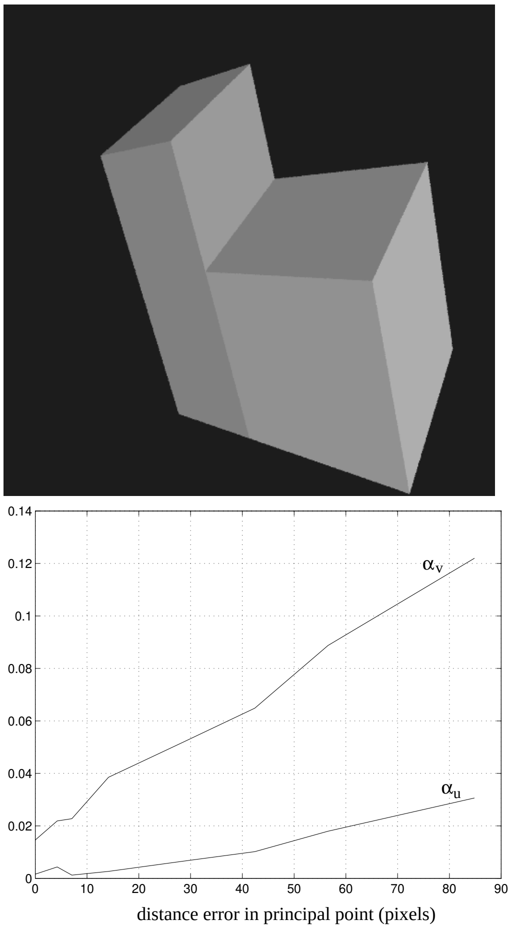

Figure 6: The 512×512 test image and the graph of the relative error in the estimation of α_u and α_v as a function of the error in the principal point position.

Algorithm

The global procedure to evaluate the scale parameters is as follows:

1. Extract lines $\{l_i\}$ of image primitives (cuboid and segment projections).

2. Set (u_0, v_0) to the image centre coordinates. Set (α_u, α_v) initially to $(1, 1)$.

3. Estimate the positions of vanishing points using intrinsic parameters and $\{l_i\}$ as explained in section 5.

4. Estimate scale parameters (α_u, α_v) using (10).

5. Iterate steps 3 and 4 until the estimated values for the scale parameters converge.

Evaluation

We have tested this algorithm on synthetic data in order to evaluate the sensitivity of the estimated parameters to errors in the assumed position of the principal point (u_0, v_0). Figure 6 shows an example test image. Different images were generated with principal points at different distances from the assumed image centre $(256, 256)$. The graph of figure 6 shows relative errors on the estimation of α_u and α_v. The maximum relative error is of 12.5% when the principal point is 85 pixels from the image centre. Large errors in the principal point can therefore be tolerated. Note also the difference between relative errors on α_u and α_v. This is due to the fact that the overall error on these parameters is divided according to the relative proximity of vanishing points to one or the other image axes (i.e. according to the orientation of the cuboid's space directions in the camera coordinate frame).

7 Projection Matrices

In the previous sections we have explained how to estimate intrinsic parameters using the vanishing points. Once the intrinsic parameters are known, it is possible to estimate the relative position and orientation of the cameras, that is the rigid motion between views. This is used to compute consistent projection matrices for all views.

Two different approaches can be used. The conventional approach is to use many pairs of matched points to estimate a fundamental matrix and then to decompose this matrix into a rotation and translation once the intrinsic parameters are known [15, 9]. Note that this method can not be used if the points lie on a cube – a critical surface [15].

An alternative method presented below is to use the vanishing points directly to estimate the rotation (since these are independent of camera translation) and then the translation from pairs of corresponding points using the epipolar constraint.

Rotation

Let ν_1 and ν_2 be two vanishing points in image 1 and 2 respectively, corresponding to the same space direction. Let $V_1 = \mathbf{C}_1^{-1}\nu_1/||\mathbf{C}_1^{-1}\nu_1||$ and $V_2 = \mathbf{C}_2^{-1}\nu_2/||\mathbf{C}_2^{-1}\nu_2||$ be the 3D unit vectors representing this space direction in the camera coordinate frames 1 and 2 respectively. Since vanishing points correspond to a space direction, they do not change their coordinates under viewer translation [2] but only by rotation of the viewpoint. From this observation, we derive:

$$V_2 = \mathbf{R}V_1$$

and for n vanishing points:

$$(V_2^1, .., V_2^n) = \mathbf{R}(V_1^1, .., V_1^n).$$

The rotation matrix can be estimated from at least three vanishing points in both images by SVD decomposition [11].

Translation

Let (x_1, x_2) be the projections of a space point $\mathbf{X}$ in the two images. By the epipolar constraints the vectors $\mathbf{RC}_1^{-1}x_1$, $\mathbf{C}_2^{-1}x_2$ and the direction of translation between the two views, $\mathbf{T}$, are coplanar. Thus:

$$[(\mathbf{RC}_1^{-1}x_1) \wedge (\mathbf{C}_2^{-1}x_2)] \cdot \mathbf{T} = 0,$$

which gives a linear constraint on the translation parameters. Let γ_i be the unit vector orthogonal to the epipolar plane associated with matched point (x_1^i, x_2^i):

$$\gamma_i = \frac{(\mathbf{RC}_1^{-1}x_1^i) \wedge (\mathbf{C}_2^{-1}x_2^i)}{||(\mathbf{RC}_1^{-1}x_1^i) \wedge (\mathbf{C}_2^{-1}x_2^i)||}$$

and let Γ be the matrix defined by:

$$\Gamma = (\gamma_1, .., \gamma_n)^T$$

for n pairs of matched points. The eigenvector of the matrix $\Gamma\Gamma^T$ corresponding to the smallest eigenvalue gives a solution for the direction of translation – the direction which is closest to all the epipolar plane in the least-squares sense. The translation is estimated up to a scalar and the ambiguity on its sign is resolved during the reconstruction stage.

8 3D reconstruction

From the previous step, we have computed motions between the different views, up to scale factors. These motions combined with the intrinsic parameters allow us to compute projection matrices for the different views involved. From these projection matrices, we can determine the epipolar geometry to help find more point correspondences and then the 3D point positions. These points are then used in association with an image point triangulation to obtain 3D structure. This structure is rendered afterwards using a texture mapping procedure and the final model is stored in standard VRML format. In this section, we summarise the different steps involve to produce such a model.

8.1 Reconstruction

Let $\mathbf{P}_1$ and $\mathbf{P}_2$ be the projection matrices for view 1 and 2 respectively. If we now take the coordinate system of the first camera as the 3D space coordinate system then:

$$\mathbf{P}_1 = \mathbf{C}_1 \left[\begin{array}{cc} \mathbf{I} & \mathbf{0} \end{array}\right] \quad (11)$$

$$\mathbf{P}_2 = \mathbf{C}_2 \left[\begin{array}{cc} \mathbf{R} & \mathbf{T} \end{array}\right] \quad (12)$$

where $\mathbf{R}$ and $\mathbf{T}$ now represent the rigid motion between view 1 to view 2. There is an ambiguity in the sign of $\mathbf{T}$. To solve this ambiguity, 3D points must be reconstructed and the sign of the translation is chosen to make sure that the depths are not negative. The estimation of the 3D coordinates of a point $\mathbf{X}$ is straightforward and can be obtained by the least-squares solution from at least two image points.

Since no metric information about the scene are available, reconstruction is performed up to a scale factor.

8.2 Triangulation and texture mapping

The projection matrices are used to reconstruct points individually. New primitives can be added to the original set of primitives, either automatically [1] or interactively. To compute a 3D structure, image points are then triangulated in one view. In order to preserve the information given by primitives segments, the triangulation performed is a constrained triangulation. More precisely, it is a constrained Delaunay triangulation [5] which preserve edges. The 2D triangulation leads to triangular 3D structure that represent the scene.

To render the reconstructed structure, we use a texture mapping algorithm in which a texture map is constructed for each triangular patch in the 3D model. Each element in the map is obtained by projecting the corresponding triangle point onto the image plane and by determining the intensity value by bilinear interpolation.

9 Experimental results

Experiments on synthetic and real data have been conducted. Figures 7 – 9 show some preliminary results for real images of the Downing College library in the University of Cambridge. These images were obtained by an Olympus digital camera. The calibration for these images was performed using two cuboids. Figure 8 shows these cuboids and the corresponding directions which determine the vanishing points. The geometry was computed using the vanishing points the 3D co-ordinates were then recovered.

Figure 9 shows an example of the final textured model. This model consists in fifty textured triangles. Note that the reconstructed structure is visually consistent with the original one even if points are treated individually: reconstructed squares are squares and reconstructed right angles are right angles. This shows that the computed geometry is correct with respect to the imposed constraints. Furthermore, the realism of the 3D model is strongly dependent on the fact the constraints of parallelism and orthogonality are actually verified. These results are extremely promising in scene modelling applications.

10 Conclusions

The techniques presented have been successfully used to interactively build models of architectural scenes from pairs of uncalibrated photographs. The simple but powerful constraints of parallelism and orthogonality in architectural scenes can be used to recover very precise projection matrices with only a few point and line correspondences.

We plan to use these initial estimates of the projection matrices (and hence the epipolar geometry) to automatically match additional features and then to optimise the parameters of the camera motion and 3D structure by standard ray-bundle adjustment. Preliminary results are extremely promising.

Acknowlegments

We acknowledge the financial support of Office Workstation Limited (Edinburgh) and Panasonic, Japan.

References

[1] 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.

[2] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *International Journal of Computer Vision*, 4: 127–139, 1990.

[3] R. Cipolla. The Visual Motion of Curves and Surfaces. In *Phil. Trans. Royal Society*, A356:1103–1121, 1998.

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

[5] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* Artificial Intelligence. MIT Press, Cambridge, 1993.

[6] O. Faugeras and S. Laveau. Representing Three-Dimensional Data as a Collection of Images and Fundamental Matrices for Image Synthesis. In *Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem (Israel)*, pages 689–691, 1994.

[7] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The Lumigraph. In *ACM Computer Graphics (Proceedings SIGGRAPH)*, pages 43–54, 1996.

[8] C. Harris and M. Stephens. A combined corner and edge. In *Proceedings 4th Alvey Vision Conference*, pages 147–151, 1988.

[9] R.I. Hartley. In Defence of the 8-point Algorithm. In *Proceedings of 5th International Conference on Computer Vision, Boston (USA)*, pages 1064–1070, January 1995.

[10] R.I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2): 125–140, 1996.

[11] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice.* Lecture Note, Gunma University (Japan) 1995.

[12] M. Levoy and P. Hanrahan. Light Field Rendering. In *ACM Computer Graphics (Proceedings SIGGRAPH)*, pages 31–42, 1996.

[13] H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. *Nature*, 293: 133–135, September 1981.

[14] D. Liebowitz and A. Zisserman. Metric Rectification for Perspective Images of Planes. In IEEE Proc. CVPR 1998.

[15] Q.T. Luong and O. Faugeras. The Fundamental Matrix: Theory, Algorithms and Stability Analysis. *International Journal of Computer Vision*, 17(1): 43–75, 1996.

[16] P.J. Narayanan, P.W. Rander, and T. Kanade. Constructing Virtual Worlds Using Dense Stereo. In *Proc. of Sixth IEEE Intl. Conf. on Computer Vision, Bombay (India)*, pages 3–10, January 1998.

[17] M. Pollefeys, R. Koch and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown internal camera parameters. In *Proc. of Sixth IEEE Intl. Conf. on Computer Vision, Bombay (India)*, pages 90–95, January 1998.

[18] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C, The Art of Scientific Computing, Second Edition.* Cambridge University Press, 1992.

[19] S.M. Seitz and C.R. Dyer. Toward Image-Based Scene Representation Using View Morphing. In *Proc. of Intl. Conf. on Pattern Recognition, Vienna (Austria)*, January 1996.

[20] A. Shashua. Trilinearity in Visual Recognition by Alignment. In *Proceedings of Third European Conference on Computer Vision, Stockholm, (Sweden)*, pages 479–484, January 1994.

[21] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154,1990.

[22] T. Werner, R.D. Hersh, and V. Hlavac. Rendering Real-World Objects Using View Interpolation. In *Proceedings of 5th International Conference on Computer Vision, Boston (USA)*, pages 957–962, January 1995.

[23] A. Zisserman, D. Liebowitz and M. Armstrong. Resolving ambiguities in auto-calibration. In *Phil. Trans. Royal Society*, A356:1193–1211, 1998.

Figure 7: The original images from uncalibrated cameras.

Figure 8: The two cuboids used for the calibration and the lines used to estimate the vanishing points.

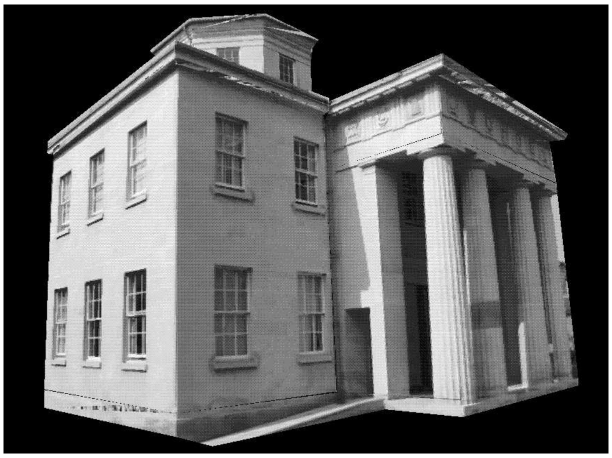

Figure 9: The 3D textured model.