AUTOMATIC 3D MODEL ACQUISITION AND GENERATION OF NEW IMAGES FROM VIDEO SEQUENCES

Andrew Fitzgibbon and *Andrew Zisserman*
Dept. of Engineering Science, University of Oxford,
19 Parks Road, Oxford OX1 3PJ, UK
e-mail: {awf,az}@robots.ox.ac.uk

ABSTRACT

We describe a method to completely automatically recover 3D scene structure together with 3D camera positions from a sequence of images acquired by an unknown camera undergoing unknown movement. Unlike "tuned" systems which use calibration objects or markers to recover this information, and are therefore often limited to a particular scale, the approach of this paper is more general and can be applied to a large class of scenes. It is demonstrated here for interior and exterior sequences using both controlled-motion and hand-held cameras.

The paper reviews Computer Vision research into structure and motion recovery, providing a tutorial introduction to the geometry of multiple views, estimation and correspondence in video streams. The core method, which simultaneously extracts the 3D scene structure and camera positions, is applied to the automated recovery of VRML 3D textured models from a video sequence.

1 INTRODUCTION

As virtual worlds demand ever more realistic 3D models, attention is being focussed on systems that can acquire graphical models from real objects. This paper describes a method for processing a sequence of images acquired by an unknown camera undergoing unknown movement to completely automatically recover 3D scene structure together with 3D camera positions. We employ Structure and Motion recovery results from the photogrammetry and computer vision literature, where it has been shown that there is sufficient information in perspective projections of a static cloud of 3D points and lines to determine the 3D structure as well as the camera positions *from image measurements alone*.

The core system is an automatic process which can be thought of, at its simplest, as converting a camcorder to a sparse range sensor. Together with more standard graphical post-processing such as triangulation of sparse 3D point and line sets, and texture mapping from images, the system becomes a "VHS to VRML" converter — to acquire a realistic model of a 3D scene, a user must simply video it. The primary application is as a simple, automatic, accurate, and quick means of model acquisition to populate virtual worlds. Figure 1 shows a schematic overview of the system.

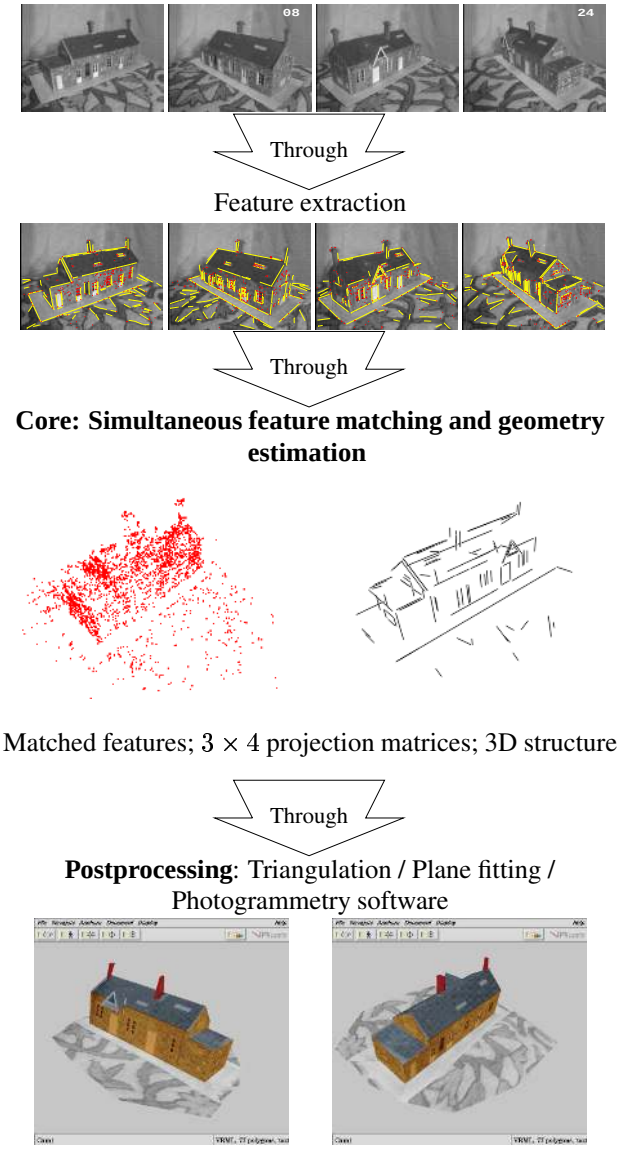

Figure 1: Overview of the system. Four frames from the 32-frame input video sequence are shown at the top; views of the automatically acquired VRML model are shown at the bottom.

The key advantage of the approach we adopt is that no information other than the images themselves is required *a priori*: the camera pose is computed automatically from texture in the viewed 3D scene, so that neither calibration patterns nor 3D control points are required.

1.1 Background

Although the general framework for uncalibrated structure from motion has been in place for some time [6, 14, 17] only recently have general acquisition systems come near to becoming a reality. This is because a combination of image processing, projective geometry for multiple views [13, 23, 25], and robust statistical estimation [28, 29] has been required in order to succeed at automating structure and motion algorithms [1, 16].

Tomasi and Kanade's acquisition system [26] has much in common with ours, taking uncalibrated views and converting them to 3D structure. However there are several important differences: first, a simplified projection model is used, in our case the most general projection model applies. Significant perspective effects in the Kanade system (giving rise to vanishing points etc) will degrade the results. Second, their system uses a simple point tracker to find matches and does not employ robust statistics and rigid geometry for tracking—this severely limits camera motions and the type of acquisition scenes.

1.2 The scope of the approach

The limitations of the approach of this paper can essentially be summarized by saying that the images must be sufficiently "interesting"—if the scene has no significant texture (to be defined more precisely later), then the feature based methods we use will have too few 2D measurements to work with; and second, that the camera motion between images needs to be relatively small, in particular rotation about the optical axis should be limited—otherwise the cross-correlation techniques used to match the features between images will fail. Happily, this restricted motion is the typical motion between frames of a video sequence, and the system is tuned for such data. We also require that the 3D scene be largely static, although smaller independently moving objects—shadows, highlights, passing cars and the like—are tolerated because of the use of robust statistics.

The advantage of a video sequence, where the distance between camera centres (the baseline) for successive frames is small, is that correspondence between successive images is simplified because the images are similar in appearance. The disadvantage is that the 3D structure is estimated poorly due to the small baseline. However, this disadvantage is ameliorated by tracking over many views in the sequence so that the effective baseline is large. The accurate position of the 3D point or line is then computed by a bundle adjustment [24] over all views in which it appears.

2 THE CORE METHOD: CAMERAS FOR EACH FRAME, AND 3D POINTS AND LINES

The core method is now described—the uncalibrated structure and motion algorithm. The core method is automatic, requiring no manual intervention at any stage. The house sequence of figure 1 will be used to illustrate the method throughout this paper.

The key ideas are that the images of 3D entities (points, lines) satisfy relationships which are induced by the geometry of cameras viewing a rigid scene [7, 15]. These relationships are represented by tensors; in the two-view case the tensor is the fundamental matrix. These tensors can be computed from the image coordinates of a sufficient number of corresponding entities alone. The camera positions are then determined from the tensors, and given the cameras and correspondences the 3D structure can be recovered.

Sections 3 to 5 describe the core system: the 2D feature extraction process, the geometry of multiple-view tensors, and the statistical estimation of the tensors from the 2D features.

3 FEATURE EXTRACTION

In order to recover the 3D entities, their 2D images must be extracted from the input sequence. Two types of image primitives are used—interest points ("corners") and line segments—extracted independently in each frame of the sequence using standard computer vision algorithms. These algorithms have the desirable property that the features they produce are generally the images of real 3D point and line features in the scene.

Corners are detected to sub-pixel accuracy using the Harris corner detector [12]. Line segments are detected by: Canny edge detection at sub-pixel accuracy[4]; edge linking; segmentation of the chain at high curvature points; and finally, straight line fitting to the resulting chain segments. The straight line fitting is by orthogonal regression, with a tight tolerance to ensure that only actual line segments are extracted, i.e. that curves are not piecewise linear approximated. Further implementation details are given in [1], and examples are shown in figure 2b.

4 THE GEOMETRY OF MULTIPLE VIEWS: REVIEW

We work in projective 2- and 3- space, representing geometric objects in homogeneous coordinates. In general bold uppercase is used for homogeneous 4-vectors $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1)^\top$ and bold lowercase for image 3-vectors $\mathbf{x} = (x, y, 1)^\top$. Note that equations involving homogeneous primitives are defined only up to scale. This review is based on the following papers and books [2, 6, 9, 13, 14, 15, 18].

Perspective Projection A camera maps a point in 3D to a 2D image plane. The mapping is perspective (or central) projection, and is represented by a 3×4 projection matrix, P, which projects a 3D point $\mathbf{X}$ to its 2D image $\mathbf{x}$:

$$\mathbf{x} = \mathsf{P}\mathbf{X} \tag{1}$$

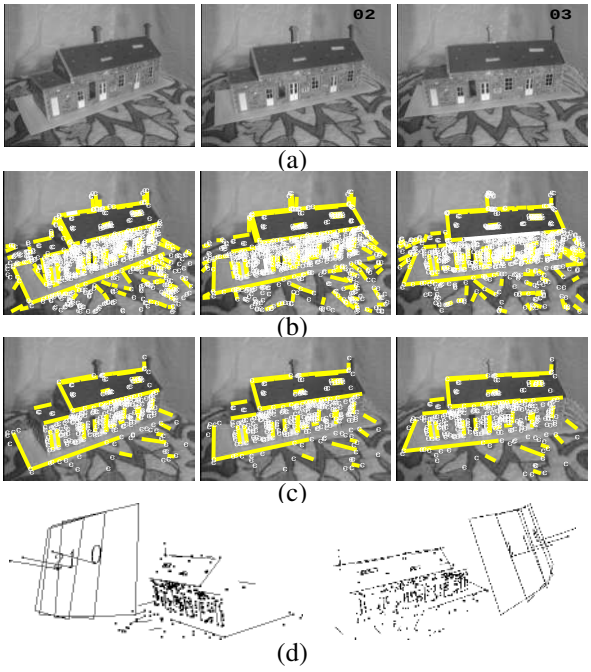

Figure 2: **Image triplet processing**: The workhorse of the system, converting a passive, uncalibrated, camera into a sparse range sensor. (a) The first three images of a 32-image sequence where the camera circumnavigates a toy house. (b) Point (white) and line (grey) features extracted from the sequence. (c) features matched across these three views. (d) Visualization of the recovered 3D structure and cameras.

The 3×4 projection matrix has 12 elements but is only defined up to an overall scale (because it appears in homogeneous equations), and so has only 11 degrees of freedom. It may be computed from the correspondence of 6 or more 3D points and their images. The null-space of P, i.e. $\mathbf{C}$ such that $\mathsf{P}\mathbf{C} = \mathbf{0}$, is the centre of projection of the camera.

Multiple-View Geometry Suppose there are n views, with the cameras represented by projection matrices $\{\mathsf{P}_i\}_{i=1}^n$. A 3D point $\mathbf{X}$ will project to a (different) 2D point $\mathbf{x}_i = \mathsf{P}_i\mathbf{X}$ in each view. These 2D points are *corresponding* features—they are images of the same 3D feature. It is assumed always that the scene is *rigid*, that is the world does not deform between views. Then the motion of the camera induces multiple view relations which are satisfied by any corresponding image points. Corresponding lines are defined in an analogous manner, again with rigidity inducing multiple view relations for lines. The multiple view relations for two and three views are described in the following subsections.

4.1 Two-View Geometry: The Fundamental Matrix

Triangulation Suppose the projection matrices, P and P' say, are known for two views, then the 3D point $\mathbf{X}$ can be computed from its images $\mathbf{x}$ and $\mathbf{x}'$. Each image point places two constraints on $\mathbf{X}$ as

$$\mathbf{x} = \mathsf{P}\mathbf{X} \qquad \mathbf{x}' = \mathsf{P}'\mathbf{X}$$

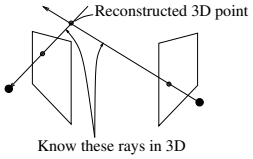

Figure 3: The principle of triangulation. The known projection matrices P and P' back project image points to 3D rays on which the 3D point lies. The 3D point position is recovered by intersecting the rays.

Figure 4: The epipolar line of a point (in the first view) is the image (in the second view) of the ray passing through the point in the first. The two images from the example sequence show a point $\mathbf{x}$ selected in the first generating the line $\mathsf{F}\mathbf{x}$ in the second. The epipolar line of the 2D point in the first view passes through the image of the 3D point in the second view. The F matrix for these two views was computed automatically by the algorithm described in section 5.1.

and these four constraints (over-) determine $\mathbf{X}$. This is *triangulation*, and is illustrated in Figure 3. It is the basis for all algorithms which recover 3D structure from 2D images.

Epipolar Geometry and the Fundamental Matrix The images of a 3D point in two views obey a simple linear relationship. As shown in figure 4, corresponding points must lie on each other's *epipolar lines*. This constraint is represented in homogeneous coordinates using the *fundamental matrix*:

$$\mathbf{x}'^{\top}\mathsf{F}\mathbf{x} = 0 \qquad (2)$$

where F is a 3×3 matrix of rank two. This is the bilinear relation in the homogeneous coordinates of the corresponding points in two images. The projective geometry of this 2-view relation is shown in figure 4.

The fundamental matrix is independent of the scene structure $\mathbf{X}$, depending only on the camera motion and internal parameters. Moreover, because the fundamental matrix directly relates image points, it can be computed from image correspondences alone: 7 point correspondences determine F (there are one or three solutions). In turn, from F, the projection matrices may be determined subject to the choice of an arbitrary basis for projective 3-space.

4.2 Three-view Geometry: The Trifocal Tensor

For a triplet of images, let the image of a 3D point $\mathbf{X}$ be $\mathbf{x}^1$, $\mathbf{x}^2$ and $\mathbf{x}^3$ in the first, second and third images respectively, and similarly the images of a line are $\mathbf{l}^1$, $\mathbf{l}^2$ and $\mathbf{l}^3$.

Corresponding points in three images, and corresponding lines in three images, satisfy trilinear relations which are encapsulated in the trifocal tensor $\mathcal{T}$, a $3 \times 3 \times 3$ homogeneous

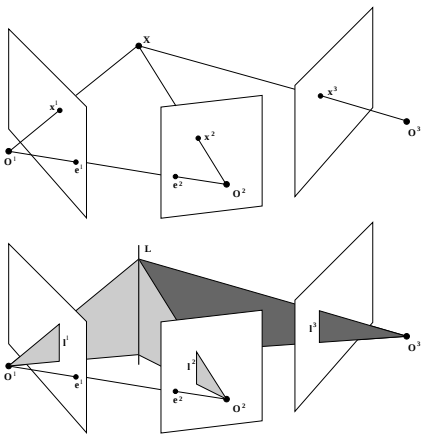

Figure 5: **Trifocal geometry**. Rays backprojected from corresponding image points in the first and second view intersect, and thus determine, the 3D point. The position of the corresponding point in the third view is computed by projecting this 3D point onto the image. Similarly lines backprojected from the first and second image intersect in the 3D line, the projection of this line in 3-space to the third image determines its image position.

Figure 6: **Trifocal line transfer**. Corresponding lines in the first two images (the roof edge marked in black) predict the infinite line in the third.

tensor. Using the tensor a point can be transferred to a third image from correspondences in the first and second:

$$x_l^3 = x_i^2 \sum_{k=1}^{k=3} x^1{}_k \mathcal{T}_{kjl} - x_j^2 \sum_{k=1}^{k=3} x^1{}_k \mathcal{T}_{kil},$$

for all $i, j = 1 \ldots 3$. Similarly, a line can be transferred as

$$l_i^1 = \sum_{j=1}^{j=3} \sum_{k=1}^{k=3} l_j^2 l_k^3 \mathcal{T}_{ijk}$$

i.e. the same tensor can be used to transfer both points and lines. The geometry of these 3-view relations is shown in figures 5 and 6.

The trifocal tensor can be computed from 6 corresponding image points over 3 views (there are one or three solutions). Given the image relation $\mathcal{T}$, the projection matrices $\mathsf{P}_1, \mathsf{P}_2, \mathsf{P}_3$ for the three views can be extracted, again subject to the choice of basis in projective 3-space.

4.3 Recovering the 3D structure and cameras

Given a set of image correspondences $\{\mathbf{x}_i\} \leftrightarrow \{\mathbf{x}'_i\}$, sufficient to determine the fundamental matrix, the corresponding object space coordinates $\{\mathbf{X}_i\}$ may be computed up to a homography of 3-space.

In more detail suppose the Euclidean coordinates of the actual (i.e. true) set of points are $\mathbf{X}_i^{\mathrm{E}}$, then from the image correspondences between two views alone, a projective reconstruction $\mathbf{X}_i$ can be obtained which is related to $\mathbf{X}_i^{\mathrm{E}}$ as

$$\mathbf{X}_i = \mathtt{H}\mathbf{X}_i^{\mathrm{E}}$$

where $\mathtt{H}$ is a 4×4 homography matrix which is unknown but the same for all points. The camera matrices of the reconstruction are also determined up to the same ambiguity:

$$\mathtt{P} = \mathtt{P}^{\mathrm{E}}\,\mathtt{H}^{-1} \qquad \mathtt{P}' = \mathtt{P}'^{\mathrm{E}}\,\mathtt{H}^{-1}$$

where the cameras are defined by $\mathbf{x}_i = \mathtt{P}^{\mathrm{E}}\mathbf{X}_i^{\mathrm{E}}$, $\mathbf{x}'_i = \mathtt{P}'^{\mathrm{E}}\mathbf{X}_i^{\mathrm{E}}$ for the Euclidean coordinates, and $\mathbf{x}_i = \mathtt{P}\mathbf{X}_i$, $\mathbf{x}'_i = \mathtt{P}'\mathbf{X}_i$ for the projective reconstruction. To remove this ambiguity, autocalibration techniques [8, 19] are used.

5 CORRESPONDENCE AND ESTIMATION: REVIEW

In the following subsections we describe two robust matching schemes applicable to a camera moving through a scene that is largely static. In the two view case the objective is to simultaneously estimate the fundamental matrix and a consistent set of point correspondences; in the three view case the objective is to simultaneously estimate the trifocal tensor and a consistent set of point correspondences over the three views. No *a priori* information on camera internal parameters or motion is assumed other than a threshold on the maximum disparity between images. The methodology for matching is essentially the same in both cases.

5.1 Matching corners between image pairs

The two-view matching problem is representative of all the simultaneous matching and geometry estimation problems. In the two view case, the pertinent geometric relation that we wish to estimate is the 7 degree-of-freedom Fundamental Matrix, and the primitives matched are 2D corners corresponding to 3D point features. The algorithm is summarized as follows:

- Extract seed correspondences by simple image-based matching.
- Use robust estimation to compute the F that has the greatest number of consistent correspondences.
- Generate more correspondences by guided matching using the the newly computed F.
- And repeat steps 2 and 3 until the number of matches stabilizes.
- Compute the Maximum Likelihood Estimate of $\mathtt{F}$.

The following paragraphs describe in greater detail the implementation of each of these steps.

Seed correspondences by unguided matching Given a corner at position (x, y) in the first image, the search for a match considers all corners within a region centred on (x, y) in the second image with a threshold on maximum disparity. The strength of candidate matches is measured by cross-correlation on corner neighbourhoods. The threshold for match acceptance is deliberately conservative at this stage to minimize incorrect matches.

Robust computation of the epipolar geometry The aim then is to obtain a set of "inliers" consistent with the geometric constraint using a robust technique — RANSAC has proved the most successful [10, 27, 28, 29]: A putative fundamental matrix (up to three solutions) is computed from a random set of seven corner correspondences (the minimum number required to compute a fundamental matrix). The support for this fundamental matrix is determined by the number of correspondences in the seed set within a threshold distance of their epipolar lines. This is repeated for many random sets, and the fundamental matrix with the largest support is accepted. The outcome is a set of corner correspondences consistent with the fundamental matrix, and a set of mismatches (outliers). The fundamental matrix is then reestimated using all of its associated inliers to improve its accuracy.

Guided matching The aim here is to obtain additional matches consistent with the geometric constraint. The constraint provides a far more restrictive search region than that used for unguided matching. Consequently, a less severe threshold can be used on the matching attributes. In this case, matches are sought for unmatched corners searching only epipolar lines. This generates a larger set of consistent matches.

Maximum Likelihood Estimation Given a statistical model for the measurement error, that the observed features have been perturbed by a Gaussian noise process, *Maximum Likelihood Estimation* (MLE) can be developed for both the fundamental matrix and the correspondences.

Suppose $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$ are the measured points, then the MLE involves obtaining a fundamental matrix $\hat{\mathtt{F}}$ *and* corrected correspondences $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i\}$ that minimize

$$\mathcal{D} = \sum_i d(\hat{\mathbf{x}}_i, \mathbf{x}_i)^2 + d(\hat{\mathbf{x}}'_i, \mathbf{x}'_i)^2$$

subject to $\hat{\mathbf{x}}'^{\top}_i \hat{\mathtt{F}} \hat{\mathbf{x}}_i = 0$, where the notation $d(\mathbf{x}, \mathbf{y})$ is the Euclidean image distance between $\mathbf{x}$ and $\mathbf{y}$. Minimization of $\mathcal{D}$ requires a *consistent* parametrization of $\mathtt{F}$, i.e. one where the constraints on the matrix elements are imposed — in this case that $\det \mathtt{F} = 0$. The minimization is carried out using the Levenberg-Marquardt algorithm [20].

Typical results Typically the number of corners used in a 768×576 image of an indoor scene is about 500, the number of seed matches is about 200, and the final number of matches is about 250. Using corners computed to sub-pixel accuracy, the average distance of a point from its epipolar line is $\sim$0.2-0.4 pixels.

5.2 Matching points between image triplets

The same basic steps are used over image triplets, with the geometric constraint provided by the trifocal tensor. Briefly, putative point matches (Harris corners) are first obtained for the consecutive image pairs, one/two and two/three, by simultaneously computing epipolar geometry and matches consistent with this estimated geometry as described above. From these seed matches the trifocal tensor is robustly fitted. The number of point correspondences in each random sample is now reduced to six, as six point triplets are enough to determine the trifocal tensor. New matches are found (*guided matching*) which are consistent with the fitted $\mathcal{T}$. Fitting and guided matching are repeated until the number of matched points stabilises. The improvements over [1] include:

1. Parametrizing the trifocal tensor such that it obeys all the constraints between the tensor elements [28].
2. Maximum-Likelihood Estimation (MLE) of $\mathcal{T}$ via bundle adjustment.

Typical results Typically the number of seed matches over a triplet is about 100 corners. The final number of matches is about 180. Using corners computed to sub-pixel accuracy, the typical distance of a corner from its transferred position is $\sim$1 pixel.

5.3 Matching lines between image triplets

Line matching is notoriously difficult over image pairs as there is no geometric constraint equivalent to the fundamental matrix for point correspondences. The following scheme matches lines over triplets using the geometric constraint provided by the trifocal tensor computed as above from point correspondences, and also a photometric constraint based on intensity cross-correlation for neighbourhoods long the lines.

In detail there are two stages of verification for line matches over an image triplet. First, a geometric verification. Given the trifocal tensor and putatively corresponding lines in two images, the corresponding line in the third image is determined. A line segment should be detected at the predicted position in the third image. Second, a photometric verification. The basic idea is to treat each line segment as a list of points to which neighbourhood correlation is applied as a measure of similarity. Only the point to point correspondence is required, and this is provided by epipolar geometry. Details are given in [21].

Typical results Typically there are 200 lines in each image and a third of these are matched over the triplet. The line transfer error is generally less than a pixel. In practice the two stages of verification eliminate all but a couple of mismatches.

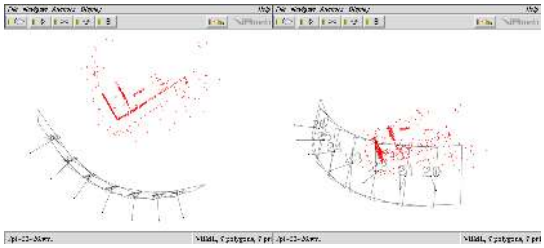

Figure 7: **Registered triplets**. Registered cameras and structure for 7 frames (five triplets) of the example sequence.

Figure 8: **Example sequences**: Model house (32 frames); Dinosaur on turntable (36 frames); Castle, hand-held camera (25 frames); Basement, camera on a vehicle (12 frames).

5.4 From triplets to sequences

The computation of the trifocal tensor and the concomitant point and line correspondences provides accurate and reliable 3D structure and camera positions from each successive triplet of views in the sequence.

These image triplets are then merged in order to extract structure and camera motion for the entire sequence. This problem is similar to that of registering range images into a consistent frame and the approach taken is broadly related to the iterated closest point (ICP) algorithm [3]. The problem here differs from ICP in two ways. First, rather than solving for a scaled Euclidean transformation, as in the calibrated (e.g. range image) case, a projective transformation of 3-space, represented as a 4×4 homogeneous transformation matrix, must be determined. Second, the correspondence problem is rendered trivial in this case by the existence of the image feature correspondences. Further details are supplied in [11]. An example of the registered views and structure is shown in figure 7.

6 EXAMPLES

Several example sequences are shown in figure 8. The following descriptions illustrate several applications of the core structure and motion recovery system. First the sequences are discussed, with the points of note being identified, and then some applications of the system are presented, with reference to the example sequences.

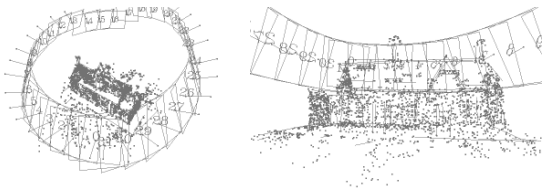

Figure 9: **Model house:** 3D point and line structure plus cameras represented by their (numbered) image planes.

Figure 10: **Dinosaur:** 3D point structure for the Dinosaur sequence.

6.1 Model house sequence

This is a 32 frame sequence obtained from a low resolution monochrome Pulnix camera. The model is rotated on a turntable so that effectively the camera circumnavigates the object. No information concerning the camera motion is used at any stage. In particular the angular rotation between views is irregular. The fact that the sequence is closed *is* used to refine the recovered structure. The automatically extracted point and line structure is shown in figure 9. Because the model is known to be rotating on a turntable, the quality of the recovered structure can be assessed by observing the positions of the recovered cameras, which should lie in a circle. Of course the model could be improved by imposing the constraint that the cameras lie in a circle, and this is planned in the near future.

6.2 Dinosaur sequence

This sequence is again a closed turntable sequence, but of a non-polyhedral object. Feature extraction is performed on the luminance component of the colour signal. No reliable lines are extracted on this object so only points are used. Again note (Fig. 10) the circularity of the recovered cameras. Again, no knowledge of the circular motion was used, in order to more thoroughly exercise the system.

6.3 Castle sequence

This sequence is taken with a standard SLR camera, by a cameraman walking around the grounds of a Belgian castle. The images have been digitized to PAL resolution and presented to the system. There is significant lighting variation between the first and final frames, and the sequence contains non-rigid components (passing pedestrians and moving trees). Figure 11 shows that structure and motion are successfully recovered despite these impediments.

6.4 Basement sequence

A camera was mounted on a mobile robot for this sequence. The robot moves along the floor turning to the left. The forward translation in this sequence makes structure recovery

Figure 11: **Castle:** Computed cameras and 3D point structure. The plan view shows the accuracy of the self calibration.

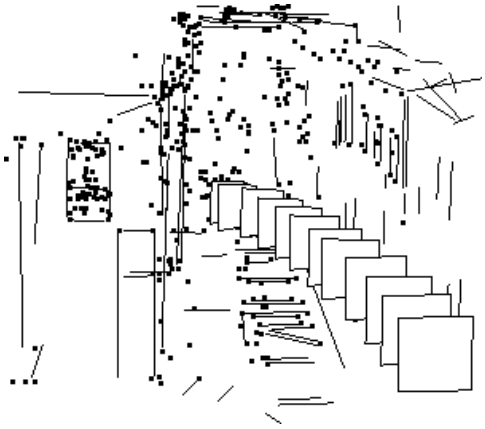

Figure 12: **Basement:** Computed cameras and 3D structure. Digital camera mounted on autonomous guided vehicle (AGV). Forward motion is a difficult case due to the small interocular baseline. In this case the combining of all views gives greatly improved structure over the sequential system.

difficult, due to the small baseline for triangulation. In this situation, the benefit of using all frames in the sequence is significant. Figure 12 shows the recovered structure.

7 VRML MODEL CONSTRUCTION

Having the complete point and line structure, we now describe how to convert the sparse 3D features into a form suitable for graphical rendering.

To produce triangulated structure for the polyhedral examples in this paper, planes are automatically extracted from the 3D data using the RANSAC technique: random 3-point subsets of the data are selected to define planes, and the number of 3D points which are less than a user-specified distance from each plane are counted. The plane with the greatest number of consistent points is stored, and the data points which were consistent with it removed from the structure. Repeating this process extracts the largest planes from the dataset, and the process is terminated when the required number of planes have been found.

The RANSAC procedure, by its nature, will ignore small-

Figure 13: **Final model**. Two views of the VRML model obtained after plane fitting and photogrammetric modelling.

Figure 14: **Basement**: Texture mapped planar model built from 11 views of the basement sequence. Left: VRML model of the scene with the cameras represented by their image planes (texture mapped with the original images from the sequence). Right: a rendering of the scene from a novel viewpoint different from any in the sequence.

scale structure in the data, but is an ideal starting point for photogrammetric techniques such as the Debevec *et al.* architectural system [5]. A simplified version of their approach is used here to add the chimneys and porch back into the model.

The planes are textured by selecting (automatically) the image from the sequence which is most fronto-parallel to that plane, and then texture mapping from the appropriate polygonal image region. As the texture mapping from the image to the plane is via an affine transformation, it is necessary to first warp the image to remove any projective distortion. Again this correction is automatic. Figure 13 shows the final texture-mapped model. Figure 14 shows the results of the same process applied to the point and line data of the basement sequence.

Non-polyhedral objects For the non-polyhedral objects, the surface extraction problem is more difficult, mainly due to the sparsity of the data. However, the dinosaur sequence is easily approached by segmenting the (blue) background and intersecting the cones formed by the occluding contours, and results are shown in figure 15.

8 FUTURE DEVELOPMENTS

We have presented a system that will take sequences of images from an uncalibrated camera or cameras, and will automatically recover camera positions and 3D point and line

Figure 15: **Dinosaur**: Reconstruction from occluding contours.

structure from these sequences. We are currently extending the core system to include space curves [22].

The system can be used as a pre-process to a number of computer graphics algorithms. For example building a lumigraph or for light field rendering. Since the depth is known for each image there is also the opportunity in film and video post-production for techniques to employ this. Examples are "blue-screening" based on depth (Z-keying); depth based optical blurring to simulate depth of field effects; changing the lighting of a videoed scene; and, augmenting the video (AR).

Acknowledgements

We are grateful for permission to use the castle sequence supplied by the University of Leuven and the dinosaur sequence supplied by the University of Hannover. Financial support was provided by ACTS project VANGUARD.

References

[1] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proc. ECCV*, LNCS 1064/1065, pages 683–695. Springer-Verlag, 1996.

[2] P. Beardsley, A. Zisserman, and D. W. Murray. Navigation using affine structure and motion. In *Proc. ECCV*, LNCS 800/801, pages 85–96. Springer-Verlag, 1994.

[3] P. J. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE T-PAMI*, 14(2):239–256, Mar 1992.

[4] J. Canny. A computational approach to edge detection. *IEEE T-PAMI*, 8(6):679–698, 1986.

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image- based approach. In *Proceedings, ACM SIGGRAPH*, pages 11–20, 1996.

[6] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. ECCV*, LNCS 588, pages 563–578. Springer-Verlag, 1992.

[7] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[8] O. Faugeras, Q. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *Proc. ECCV*, LNCS 588, pages 321–334. Springer-Verlag, 1992.

[9] O. D. Faugeras and B. Mourrain. On the geometry and algebra of point and line correspondences between n images. In *Proc. ICCV*, pages 951–962, 1995.

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.

[11] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. ECCV*, 1998.

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

[13] R. I. Hartley. A linear method for reconstruction from lines and points. In *Proc. ICCV*, pages 882–887, 1995.

[14] R. I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. CVPR*, 1992.

[15] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 1998. (to appear).

[16] S. Laveau. *Geometry of a system of N cameras. Theory, estimation, and applications.* PhD thesis, INRIA, 1996.

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

[18] J. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[19] M. Pollefeys, R. Koch, and L. Van Gool. Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. ICCV*, pages 90–96, 1998.

[20] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[21] C. Schmid and A. Zisserman. Automatic line matching across views. In *Proc. CVPR*, pages 666–671, 1997.

[22] C. Schmid and A. Zisserman. The geometry and matching of curves in multiple views. In *Proc. ECCV*, 1998.

[23] A. Shashua. Trilinearity in visual recognition by alignment. In *Proc. ECCV*, volume 1, pages 479–484, May 1994.

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

[25] M. E. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *IJCV*, 4(3):171–183, 1990.

[26] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *IJCV*, 9(2):137–154, 1992.

[27] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Proc SPIE Sensor Fusion VI*, pages 432–443, Boston, Sep 1993.

[28] P. H. S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15:591–605, 1997.

[29] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.