??, ??, 1–29 (??)

Sequential Updating of Projective and Affine Structure from Motion

P A BEARDSLEY, A ZISSERMAN AND D W MURRAY*

[pab,az,dwm]@robots.oxford.ac.uk

Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK

Received ??. Revised ??.

Abstract. A structure from motion algorithm is described which recovers structure and camera position, modulo a projective ambiguity. Camera calibration is not required, and camera parameters such as focal length can be altered freely during motion. The structure is updated sequentially over an image sequence, in contrast to schemes which employ a batch process. A specialisation of the algorithm to recover structure and camera position modulo an affine transformation is described, together with a method to periodically update the affine coordinate frame to prevent drift over time. We describe the constraint used to obtain this specialisation.

Structure is recovered from image corners detected and matched automatically and reliably in real image sequences. Results are shown for reference objects and indoor environments, and accuracy of recovered structure is fully evaluated and compared for a number of reconstruction schemes. A specific application of the work is demonstrated — affine structure is used to compute free space maps enabling navigation through unstructured environments and avoidance of obstacles. The path planning involves only affine constructions.

Keywords: Structure from motion, Projective structure, Affine structure, Path-planning, Navigation

1. Introduction

The recovery of structure from motion is a sufficiently mature field for working systems to have been applied to the navigation of mobile vehicles (Ayache 1991; Harris 1987; Harris & Pike 1987; Zhang & Faugeras 1992). All of these systems employ a calibrated camera and recover 3D Euclidean structure. In more recent structure from motion research, an emphasis has been on the use of uncalibrated cameras and the recovery of projective structure, that is, structure modulo a projective transformation (Mohr et al. 1993; Szeliski & Kang 1993; Hartley 1993).

There are a number of advantages in not requiring camera calibration. First, structure recovery will not be adversely affected by any errors in the supposed calibration or sensitive to small changes that occur due to vibrations or focusing. Second, intrinsic camera parameters can be altered freely during motion, for example focal length can be changed by zooming. Third, calibration may not be available, at least initially. For example if the source of the image sequence is an uncalibrated video.

However, a drawback of the algorithms proposed thusfar for projective structure recovery is that they operate off-line in batch mode, employing all the images of a sequence in a single computation to determine structure and camera projection matrices. In this paper, in contrast, we present and apply an algorithm which recovers projective structure sequentially, updating the structure as each successive image is captured.

Projective structure can be specialised to affine structure, and further to Euclidean structure, given suitable constraints on the camera, its motion, or the scene. We explore such specialisations, and consider the cases listed below. In those cases where camera motion is utilised, this is required

* This work was supported by EPSRC grant no. GR/H77668.
Paul Beardsley's current address is Mitsubishi Electric Research Lab, 201 Broadway, Cambridge, MA 02139, USA, email: beardsley@merl.com.

only at the initialisation stage (the first two images of a sequence) since it is at this stage that the coordinate frame is fixed. Thereafter, camera motion is not constrained or required by the processing,

- unknown camera calibration and unknown camera motion, recovering projective structure (§3.2).
- approximately known camera calibration and approximately known camera motion, recovering Quasi-Euclidean projective structure (§3.3);
- unknown but fixed calibration and pure translation of the camera, recovering affine structure (§7.1);
- approximately known fixed calibration and pure translation of the camera, giving Quasi-Euclidean affine structure (§7.1); and
- full calibration and known camera motion, giving strictly Euclidean structure (§6.1).

The concept of "Quasi-Euclidean" structure is introduced to indicate structure which remains strictly projective (or in §7.2 affine) but which is "close" to being Euclidean in the sense that there is only a small skew from the strictly Euclidean structure. We compare the accuracy and stability of the recovered structure for the different cases, investigate the constraints needed to attain a Quasi-Euclidean frame, and compare the quality of structure recovered in Quasi-Euclidean and non-Quasi-Euclidean frames.

Affine structure provides an interesting intermediate type between projective and Euclidean structure. In computational terms, projective structure is most straightforward to obtain, requiring only image correspondences, while Euclidean structure is more difficult, requiring strong constraints such as fixed camera intrinsic parameters [9]. On the other hand, projective structure contains the least geometrical information about the physical scene, while Euclidean structure fully encodes the physical geometry. Affine structure offers a useful compromise between difficulty of computation and information content.

Invariants available from affine structure include ratios of lengths on parallel line segments, ratios of areas on parallel planes, ratios of volumes, and centroids. These are all useful sources of information for tasks which involve interaction with the environment: for instance, ratios can be used for the computation of time-to-contact, and the centroid of a set of data points can be used for fixation [33] or grasping [19]. Another affine invariant is the mid-point locus between a set of points, a basic mechanism in path planning algorithms for navigation [22]. Thus although it is traditional for path-planning to be described in terms of Euclidean structure, many of the techniques will work perfectly well when supplied with affine structure. We demonstrate this point, and the quality of recovered affine structure, by navigating a camera to a specified target where the direct path is blocked by unmodelled objects.

The visual primitives used in this work are image corners, detected and matched in a sequence taken by a camera moving through a static scene and used to generate 3D coordinates for the corresponding points in the scene. The value of corner features for navigation has been demonstrated in the 'DROID' system which computed Euclidean structure using calibrated cameras (Harris 1987; Harris & Pike 1987). . Corner features are well localised, stable and abundant in imagery from a wide variety of indoor and outdoor scenes which avoid extremes of texture density and regularity (such as smooth and untextured objects where there are occluding contours but few corners, or dense texture regions which give excessive numbers of similar corners). A further significant advantage of image corners is their mathematical tractability, both for theoretical results and numerical computation. The corner-based approach described here complements the methods for computing free-space and navigating around curved objects described in (Blake et al. 1991; Blake et al. 1992).

The rest of the paper is arranged as follows. Section 2 introduces the theory and notation used in the paper. Sections 3 and 4 cover the computation of projective structure from two images, and the updating of projective structure through an image sequence. Section 5 details the matching process and how it integrates with the structure recovery and Section 6 gives experimental assessments of recovered projective structure. Section 7 describes the specialization of the algorithm required to compute affine structure and gives associated experimental results, while Section 8

demonstrates the use of affine structure in path planning for navigation. The final section, Section 9, draws overall conclusions and summarises the important practical issues arising from the work.

2. Camera models and projective representations

We now introduce the camera models and notation used in the rest of the paper. The notation and mathematical framework draw heavily on those found in (Faugeras 1992; Hartley 1992; Mundy & Zisserman 1992).

Perspective projection from 3D projective space $\mathcal{P}^3$ to the image plane $\mathcal{P}^2$ is modelled by a 3×4 matrix $\mathtt{P}$

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

where $\mathbf{x} = (x, y, 1)^\top$ and $\mathbf{X} = (X, Y, Z, 1)^\top$ are the homogeneous coordinates of an image point and 3D point respectively. For homogeneous quantities '=' indicates equality up to a non-zero scale factor.

The camera optical centre $\mathbf{Q} = (\mathbf{t}^\top, 1)^\top$ projects as $\mathtt{P}\mathbf{Q} = \mathbf{0}$, and it is convenient to partition the projection matrix $\mathtt{P}$ as

$$\mathtt{P} = [\mathtt{M} | \; -\mathtt{M}\mathbf{t}] \tag{2}$$

This partitioning is valid provided the left 3×3 matrix $\mathtt{M}$ is not singular, which requires the optical centre not to lie on the plane at infinity. In a Euclidean coordinate frame, $\mathtt{P}$ can be decomposed as

$$\mathtt{P} = \mathtt{C}[\mathtt{R} | -\mathtt{R}\mathbf{t}] \tag{3}$$

where $\mathtt{R}$ and $\mathbf{t}$ are the rotation and translation of the camera in the Euclidean frame. $\mathtt{C}$ is a 3×3 matrix encoding the camera intrinsic parameters

$$\mathtt{C} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where α_u, α_v give the focal length in pixels along the x and y axes respectively, and (u_0, v_0) is the principal point.

For two cameras with $\mathbf{x}_1 = \mathtt{P}_1\mathbf{X}$ and $\mathbf{x}_2 = \mathtt{P}_2\mathbf{X}$, corresponding points in the two images satisfy the epipolar constraint

$$\mathbf{x}_2^\top \mathtt{F} \mathbf{x}_1 = 0 \tag{5}$$

where $\mathtt{F}$ is the 3×3 *fundamental matrix*, with maximum rank 2. The epipolar line in image 2 corresponding to $\mathbf{x}_1$ is $\mathbf{l}_2 = \mathtt{F}\mathbf{x}_1$, and similarly in image 1 corresponding to $\mathbf{x}_2$ is $\mathbf{l}_1 = \mathtt{F}^\top \mathbf{x}_2$, where $\mathbf{l}_i$ are homogeneous line vectors. Partitioning $\mathtt{P}_1$ and $\mathtt{P}_2$ as in equation (2) facilitates a number of equivalent representations of $\mathtt{F}$

$$\begin{aligned} \mathtt{F} &= \mathtt{M}_2^{-\top}[\mathbf{t}_1 - \mathbf{t}_2]_\times \mathtt{M}_1^{-1} \\ &= [\mathtt{M}_2(\mathbf{t}_1 - \mathbf{t}_2)]_\times \mathtt{M}_2\mathtt{M}_1^{-1} \\ &= \mathtt{M}_2^{-\top}\mathtt{M}_1^\top [\mathtt{M}_1(\mathbf{t}_1 - \mathbf{t}_2)]_\times \end{aligned} \tag{6}$$

where $[\mathbf{v}]_\times$ denotes the vector product matrix

$$[\mathbf{v}]_\times = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$$

such that $[\mathbf{v}]_\times \mathbf{x} = \mathbf{v} \times \mathbf{x}$. Consider a 3D projective transformation of the world coordinates, $\mathbf{X}' = \mathtt{H}\mathbf{X}$, where $\mathtt{H}$ is a non-singular 4×4 matrix. Image measurements are unaffected by this transformation, and this can be used to obtain the transformation of the perspective projection matrix:

$$\mathbf{x} = \mathtt{P}\mathbf{X} = \mathtt{P}\mathtt{H}^{-1}\mathbf{X}' = \mathtt{P}'\mathbf{X}' \; . \tag{7}$$

Thus, the perspective projection matrix $\mathtt{P}$ is transformed to $\mathtt{P}' = \mathtt{P}\mathtt{H}^{-1}$ under the transformation $\mathtt{H}$. This freedom of projective world frame allows us to choose a canonical camera matrix $\mathtt{P}_1 = [\mathtt{I} | \mathbf{0}]$, where $\mathtt{I}$ is the 3×3 identity matrix. Given some arbitrary coordinate frame in which $\mathtt{P}_1$ has the form $\mathtt{P}_1 = [\mathtt{M}_1 | -\mathtt{M}_1\mathbf{t}_1]$, the canonical form can always be reached by setting $\mathtt{H}^{-1}$ in equation (7) to be the affine transformation

$$\mathtt{H}^{-1} = \begin{bmatrix} \mathtt{M}_1^{-1} & \mathbf{t}_1 \\ \mathbf{0}^\top & 1 \end{bmatrix} \; .$$

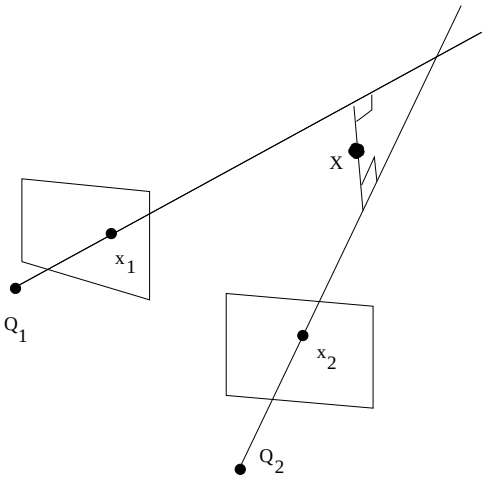


Fig. 1. Backprojected rays may be skew due to noise in the image measurements. In a Euclidean frame the midpoint of the perpendicular between the rays can then be used to give an estimate of the 3D point, but in a projective frame the concept of midpoint is invalid.

The canonical form for $\mathtt{P}_1$ implies that in position 1 the world coordinate origin is at the camera optical centre and the camera and world coordinate frames are aligned.

3. Projective stereo

The first aim of the processing is to use correspondences between corners $\mathbf{x}$ in a sequence of images to recover the structure $\mathbf{X}$ of the scene, modulo a projective transformation. That is, if the Euclidean structure of the scene is $\mathbf{X}_E$, the recovered structure is

$$\mathbf{X} = \mathtt{H}\mathbf{X}_E$$

where $\mathtt{H}$ is a non-singular 4×4 matrix which is the same for all points, but undetermined.

This section examines the initialisation of projective structure from just two images (typically the first pair in the sequence), a process called projective stereo.

3.1. Initialising the projective coordinate frame and structure

To establish a projective coordinate frame some previous methods for projective reconstruction from two or more images have selected a five point basis from the 3D points (Faugeras 1992; Mohr et al. 1993). The problem with this procedure is that if even one of the basis points is poorly localised in an image, the accuracy of the entire reconstruction degrades. Furthermore, care has to be exercised to ensure that the basis points are not collinear or coplanar. We follow more closely the approach of Hartley et al. (1992) (see also [24]) and utilise *all* corner matches in determining the projective frame, by specifying the perspective projection matrices $\mathtt{P}_1$ and $\mathtt{P}_2$ for two images.

A simple geometric argument demonstrates that this serves to fix the frame: once $\mathtt{P}_1$ has been set, each 3D point is constrained to lie on a ray backprojected from optical centre 1; fixing $\mathtt{P}_2$ then constrains each 3D point to lie at the intersection point of a pair of backprojected rays, i.e. the coordinates of the 3D points are fixed uniquely (Faugeras 1992; Hartley et al. 1992).

Unfortunately, localisation error ("noise") in the feature positions perturbs the back projected rays, and they will almost certainly not meet at a point, as sketched in Figure 1. A number of ways to allow for noise have been proposed (a comparison of different methods is given in (Rothwell 1995)). Possible approaches include

1. Given the image point $\mathbf{x}_1$ in image 1, compute the epipolar line $\mathbf{l}_2 = \mathtt{F}\mathbf{x}_1$ in image 2. Compute $\mathbf{x}_{2p}$, the orthogonal projection of $\mathbf{x}_2$ onto $\mathbf{l}_2$. Use $\mathbf{x}_1$ and $\mathbf{x}_{2p}$ to obtain the backprojected rays, which are guaranteed to be coincident at a point in 3D space since they are in the same epipolar plane.
2. Use the method in (Hartley & Sturm 1995) to compute corrected image points which are *(a)* at minimum squared distance from the actual points and *(b)* exactly agree with the epipolar geometry. This is computationally more expensive than (1) because it involves solving a polynomial of degree 6.
3. Instead of working on the image plane as in (1) and (2), work in 3D space. Compute the 3D point which minimises the sum of the square distances of the 3D point to each backprojected ray. This is the midpoint of the perpendicular between the two rays.

A disadvantage of method (1) is that all the error is assumed to be in the second image, while the disadvantage of (2) is that it involves an expensive

non-linear computation. Method (3) is in contrast a simple linear computation which allows for error in each image.

Although (3) is the most attractive option, such an approach is strictly valid only in a Euclidean coordinate frame where distance and perpendicularity are measurable, and it cannot be applied meaningfully in an arbitrary projective frame. In order to make use of method (3) while working with projective structure, we employ a *Quasi-Euclidean* projective frame. This frame is strictly projective but is "close" to Euclidean in the sense that the projective structure is within a small skew of the true Euclidean structure.

3.2. Setting a projective coordinate frame

We first describe a method for determining an arbitrary projective coordinate frame, and in the following section address the modifications which produce a Quasi-Euclidean frame. The algorithm has three principal steps.

Setting a Projective Frame

Step 1: Set the first projection matrix to the canonical form $\mathtt{P}_1 = [\mathtt{I}|\mathbf{0}]$.

Step 2: Determine the fundamental matrix $\mathtt{F}$. Find the epipole in the second image using $\mathtt{F}^\top\mathbf{e}_2 = 0$. Compute $\mathtt{M}_2 = [\mathbf{e}_2]_\times\mathtt{F}$.

Step 3: Set the second projection matrix

$$\mathtt{P}_2 = [\mathtt{M}_2 + \mathbf{e}_2\mathbf{b}^\top | \; c\mathbf{e}_2]$$

where $\mathbf{b}$ and c are an arbitrary 3-vector and scalar respectively.

The freedom in Step 1 to set $\mathtt{P}_1$ to its canonical form has been explained in §2. The practical issues involved in determining the fundamental matrix from a set of image correspondences will be elaborated on in §5.2, and the decomposition of $\mathtt{F}$ is given in Appendix A.

Lemma 1 and its accompanying proof in Appendix A show that $\mathtt{P}_2$ has four degrees of freedom. Different choices for $\mathbf{b}$ correspond to different choices of projective coordinate frame, introducing different amounts of projective skew away from the Euclidean frame. To minimise skew as much as possible by using the calibration information to hand we now consider recovering a Quasi-Euclidean frame.

3.3. Setting a Quasi-Euclidean frame

In a strictly Euclidean frame, valid choices for the projection matrices of a camera with intrinsic parameters $\mathtt{C}$ in successive positions related by rotation matrix $\mathtt{R}$ and translation vector $\mathbf{t}$ are, from equation (3), $\mathtt{P}_1^E = \mathtt{C}[\mathtt{I}|\mathbf{0}]$ and $\mathtt{P}_2^E = \mathtt{C}[\mathtt{R}|-\mathtt{R}\mathbf{t}]$. To establish a Quasi-Euclidean frame, $\mathtt{P}_1$ and $\mathtt{P}_2$ are set "close" to the form of $\mathtt{P}_1^E$ and $\mathtt{P}_2^E$, using approximate values of the camera intrinsics $\mathtt{C}^*$ and rotation $\mathtt{R}^*$.

The algorithm has three steps, which are modifications of the ones already described above.

Setting a Quasi-Euclidean Frame

Step 1: Normalize the image coordinates $\mathbf{x} \leftarrow (\mathtt{C}^*)^{-1}\mathbf{x}$ in both images. Set $\mathtt{P}_1 = [\mathtt{I}|\mathbf{0}]$.

Step 2: Determine the fundamental matrix $\mathtt{F}$. Find the epipole in the second image using $\mathtt{F}^\top\mathbf{e}_2 = 0$. Compute $\mathtt{M}_2 = [\mathbf{e}_2]_\times\mathtt{F}$.

Step 3: Using the form

$$\mathtt{P}_2 = [\mathtt{M}_2 + \mathbf{e}_2\mathbf{b}^\top | \; c\mathbf{e}_2]$$

choose c arbitrarily, but choose the value of $\mathbf{b}$ so that the term $\mathtt{M}_2 + \mathbf{e}_2\mathbf{b}^\top$ most closely approximates the estimated rotation $\mathtt{R}^*$.

The trivial normalization in Step 1 associates the effect of the camera intrinsics with the image coordinates, not with the camera matrices. The first camera matrix can then be assigned the canonical form while still being consistent with the goal of attaining a Quasi-Euclidean frame. In practice, the normalisation involves setting the homogeneous vector for an image point to be

$$\mathbf{x} = \left(\frac{u - u_0^*}{\alpha_u^*}, \frac{v - v_0^*}{\alpha_v^*}, 1\right)^\top \tag{8}$$

where (u, v) is the pixel position of the image point, and α_u^* etc are elements of $\mathtt{C}^*$ whose form was given in equation (4).

Step 2 is unchanged, but Step 3 is modified. We seek a matrix in the four dimensional subspace $\lambda\mathtt{M}_2 + \mathbf{e}_2\mathbf{b}^\top$ of $\mathcal{P}^8$ which is as close as possible to $\mathtt{R}^*$ (where λ is a scalar which is used here to make explicit the presence of the homogeneous scale factor). The subspace is spanned by the basis matrices $\mathtt{M}_2$, $\mathbf{e}_2(b_1, 0, 0)$, $\mathbf{e}_2(0, b_2, 0)$, and $\mathbf{e}_2(0, 0, b_3)$. The matrix $\mathtt{R}_p^*$ in this subspace which is closest to $\mathtt{R}^*$ is determined by the standard method of orthogonal projection of $\mathtt{R}^*$ onto the subspace. Then $\mathtt{P}_2$ is set equal to $[\mathtt{R}_p^* | \, c\mathbf{e}_2]$.

Note that the scalar c in the final column of $\mathtt{P}_2$ can be chosen arbitrarily as it merely determines the overall scale of the structure computed in the Quasi-Euclidean frame. More interesting is that although the approximate camera rotation $\mathtt{R}^*$ is used, the approximate camera translation $\mathbf{t}^*$ is not: the epipole $\mathbf{e}_2$ provides all the information about the direction of translation needed for setting $\mathtt{P}_2$.

With the projection matrices set we compute 3D structure in the Quasi-Euclidean frame in a further step as follows.

Quasi-Euclidean ctd.

Step 4: For all corresponding image points $(\mathbf{x}_1, \mathbf{x}_2)$, backproject the two rays and determine the 3D structure as the midpoint $\mathbf{X}_M$ of their mutual perpendicular.

Consider a point $\mathbf{X}$ which projects in two images $i = 1, 2$ as

$$\mathbf{x}_i = \mathtt{P}_i\mathbf{X} = [\mathtt{M}_i | -\mathtt{M}_i\mathbf{t}_i]\mathbf{X} \; . \tag{9}$$

Each backprojected ray is defined by two 3D points, the optical centre $\mathbf{Q}_i$ and the ray's intersection with the plane at infinity $\mathbf{X}_i^\infty$. The optical centre is given by $\mathbf{Q}_i = (\mathbf{t}_i^\top, 1)^\top$, while $\mathbf{X}_i^\infty$ is found from equation (9) by

$$\mathbf{X}_i^\infty = \begin{pmatrix} \mathtt{M}_i^{-1}\mathbf{x}_i \\ 0 \end{pmatrix} \; .$$

Since $\mathbf{X}$ lies on both backprojected rays

$$\begin{aligned} \mathbf{X} &= \begin{pmatrix} \mathbf{t}_1 \\ 1 \end{pmatrix} + \lambda_1 \begin{pmatrix} \mathbf{X}_1^\infty \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{t}_2 \\ 1 \end{pmatrix} + \lambda_2 \begin{pmatrix} \mathbf{X}_2^\infty \\ 0 \end{pmatrix} \end{aligned}$$

where $\lambda_{1,2}$ are unknown scalars. This is an overconstrained system of three equations in two unknowns which, because the backprojected rays will be skew due to noise, will not have a consistent solution. We obtain the midpoint $\mathbf{X}_M$ of the perpendicular between the rays by solving

$$\mathbf{X}_M = \left(\sum_{i=1,2} [\mathtt{I} - \mathbf{D}_i\mathbf{D}_i^\top] \right)^{-1} \left(\sum_{i=1,2} \mathbf{t}_i - \sum_{i=1,2} (\mathbf{t}_i^\top \mathbf{D}_i)\mathbf{D}_i \right)$$

where $\mathbf{X}_M$ is a 3-vector, and $\mathbf{D}_i = (X_{i1}^\infty, X_{i2}^\infty, X_{i3}^\infty)^\top$ is normalised to unit magnitude. (Note that the formula extends to the intersection of n rays by summing over $i = 1..n$.)

3.4. Projective skew in the Quasi-Euclidean frame

If the approximate camera calibration $\mathtt{C}^*$ and approximate camera rotation $\mathtt{R}^*$ are perfect, the resulting Quasi-Euclidean coordinate frame is strictly Euclidean; otherwise the frame is subject to a projective skew. To obtain quantitative information about the extent of projective skew, it is possible to develop an expression for the transformation $\mathtt{H}$ between a Quasi-Euclidean projective frame obtained by the method in §3.3 and a strictly Euclidean frame. The transformation is a function of the true and approximated camera intrinsics, and the true and approximated rotation matrix, as described in Appendix B. Here we provide some typical numerical examples of actual and approximated camera information, and compute the resulting projective skew.

(a) Example 1. The true camera parameters and motion are $\alpha_u = 660$ pixel, aspect ratio $\alpha_v/\alpha_u = 1.5$, and $(u_0, v_0) = (260, 263)$ for a 512×512 image, a rotation of 2° to the left around the vertical axis, and a translation $\mathbf{t}$ of 1 unit

with direction along the optical axis. The assumed parameters are $\alpha_u^* = 500$ pixels, $\alpha_v^*/\alpha_u^* = 1.4$, $(u_0^*, v_0^*) = (256, 256)$ and $\mathtt{R}^* = \mathtt{I}$. Using equation (B1) in Appendix B, the transformation between the Quasi-Euclidean frame and a strictly Euclidean frame is

$$\mathtt{H}_1 = \begin{pmatrix} 1 & 0 & -0.01 & 0 \\ 0 & 0.97 & -0.01 & 0 \\ 0 & 0 & 1.3 & 0 \\ -0.02 & 0 & 0 & 1 \end{pmatrix}$$

(a) Example 2. True camera parameters and motion are as in example 1. The assumed parameters are $\alpha_u^* = 1$ pixel, $\alpha_v^*/\alpha_u^* = 1$, $(u_0^*, v_0^*) = (0, 0)$, and $\mathtt{R}^* = \mathtt{I}$. In this case, the transformation between the initialised frame and a strictly Euclidean frame is

$$\mathtt{H}_2 = \begin{pmatrix} 1.00 & 0 & -260.42 & 0 \\ 0 & 0.65 & -171.05 & 0 \\ 0 & 0 & 641.0 & 0 \\ -0.02 & 0 & 20.36 & 1 \end{pmatrix}$$

Note that $\mathtt{H}_1$ is much closer to an identity matrix than $\mathtt{H}_2$, with the top-left 3×3 matrix close to the identity matrix, and the elements of the bottom-left 1×3 row having small size relative to unity, indicating less projective distortion. Thus, making very approximate but sensible guesses can result in the transformation $\mathtt{H}$ approaching $\mathtt{I}$ i.e. a Quasi-Euclidean frame which is nearly Euclidean.

In practice, we have obtained approximate camera calibration using both naïve calibration methods (such as imaging a fronto-parallel ruler and using similar triangle constructions to obtain estimates of focal length and aspect ratio), and self-calibration (Armstrong et al. 1994; Faugeras et al. 1992; Hartley 1994) . To obtain approximate camera motion, we have used odometry from the robot arm or mobile vehicle carrying the camera, or assumed zero rotation and set $\mathtt{R}^* = \mathtt{I}$. Any combination of the above has proved sufficient to obtain a reasonable Quasi-Euclidean coordinate frame. Later, we show that using the Quasi-Euclidean frame makes a significant contribution to accurate reconstruction, especially when the quantization error in features is appreciable.

4. Sequential updating of projective structure

Whilst Section 3 dealt with projective stereo, the computation of projective structure from just a pair of images, this section discusses the updating of structure throughout an image sequence. The algorithm is described in four parts:

- the computation of the perspective projection matrix for the latest image;
- the updating of structure based on the latest image;
- the refinement of the estimate of the projection matrix; and
- the initialisation of new structure.

4.1. Computing P

The first two images in a sequence are processed as in Section 3. Now consider the general case when structure is known for image $(i-1)$ and processing is about to begin on image i. Matching of corners between images $(i-1)$ and i provides a correspondence between existing 3D points and the new observations in image i. These correspondences are used to compute the perspective projection matrix $\mathtt{P}_i$ for image i. The process is fully described in Section 5.3 where details of the matching are also given.

4.2. Updating structure

Structure updating is achieved using an Iterated Extended Kalman Filter (IEKF) with a separate filter operating on each 3D point. This approach is well-tried in Euclidean structure from motion algorithms [11], but here we are applying it within a projective framework.

We adopt the notation q_i to indicate a quantity q at timestep i, and $\hat{q}_{(i|j)}$ to denote an estimate of q at timestep i conditioned on observations up to and including timestep j. At image i for example, the estimate of a point's 3D position $\mathbf{X}$ is $\hat{\mathbf{X}}_{(i|i-1)}$ and $\hat{\mathbf{X}}_{(i|i)}$ before and after the update respectively.

Because the structure is assumed static, the state transition equation is simply

$$\mathbf{X}_i = \mathbf{X}_{(i-1)} = \mathbf{X} \ .$$

The observation equation is

$$\mathbf{x}_i = \mathbf{h}(\mathbf{X}) + \mathbf{w}_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \frac{\mathtt{P}_i\mathbf{X}}{\mathtt{P}_i[3]\mathbf{X}} + \mathbf{w}_i \quad (10)$$

where $\mathbf{x}_i = (x, y)^\top$ is a corner in image i (here, note, a 2-vector), $\mathbf{X} = (X, Y, Z, 1)^\top$ is the corresponding 3D point, and $\mathbf{w}_i$ is temporally uncorrelated zero-mean Gaussian noise. $\mathtt{P}_i[3]$ is a 4-vector for the third row of $\mathtt{P}_i$ taken from

$$\mathtt{P}_i = \begin{pmatrix} \mathtt{P}_i[1] \\ \mathtt{P}_i[2] \\ \mathtt{P}_i[3] \end{pmatrix} = \begin{pmatrix} P_{i11} & P_{i12} & P_{i13} & P_{i14} \\ P_{i21} & P_{i22} & P_{i23} & P_{i24} \\ P_{i31} & P_{i32} & P_{i33} & P_{i34} \end{pmatrix}$$

The prediction equations for the estimated state and covariance are (Bar-Shalom & Fortmann 1988)

$$\begin{aligned} \hat{\mathbf{X}}_{(i|i-1)} &= \hat{\mathbf{X}}_{(i-1|i-1)} \\ \mathtt{A}_{(i|i-1)} &= \mathtt{A}_{(i-1|i-1)} \end{aligned}$$

and the update equations for the state vector and covariance matrix are

$$\hat{\mathbf{X}}_{(i|i)} = \hat{\mathbf{X}}_{(i|i-1)} + \mathtt{W}\nu_i \quad (11)$$

$$\mathtt{A}_{(i|i)} = \mathtt{A}_{(i|i-1)} - \mathtt{W}\mathtt{S}_i\mathtt{W}^\top \quad , \quad (12)$$

where the Kalman gain matrix, innovation vector, and innovation covariance are

$$\begin{aligned} \mathtt{W} &= \mathtt{A}_{(i|i-1)}\nabla\mathbf{h}_\mathbf{X}^\top\mathtt{S}_i^{-1} \\ \nu_i &= \mathbf{x}_i - \mathbf{h}(\hat{\mathbf{X}}_{(i|i-1)}) \\ \mathtt{S}_i &= \nabla\mathbf{h}_\mathbf{X}\mathtt{A}_{(i|i-1)}\nabla\mathbf{h}_\mathbf{X}^\top + \mathtt{R}_i \end{aligned}$$

respectively, and $\mathtt{R}$ is the covariance matrix for the observed image points $\mathbf{x}$.

The Jacobian $\nabla\mathbf{h}_\mathbf{X}$ of the non-linear observation equation (10) is evaluated at $\hat{\mathbf{X}}_{(i|i-1)}$

$$\nabla\mathbf{h}_\mathbf{X} = \begin{pmatrix} \frac{\partial h_x}{\partial X} & \frac{\partial h_x}{\partial Y} & \frac{\partial h_x}{\partial Z} \\ \frac{\partial h_y}{\partial X} & \frac{\partial h_y}{\partial Y} & \frac{\partial h_y}{\partial Z} \end{pmatrix}$$

whose jk-th element is

$$\left(\frac{\partial h_j}{\partial X_k}\right) = \frac{P_{ijk}}{\mathtt{P}_i[3]\mathbf{X}} - \frac{P_{i3k}\mathtt{P}_i[j]\mathbf{X}}{(\mathtt{P}_i[3]\mathbf{X})^2} \quad .$$

Within an IEKF, the update cycle in equations (11,12) is repeated for a number of iterations with $\nabla\mathbf{h}_\mathbf{X}$ evaluated at the current value of $\hat{\mathbf{X}}_{(i|i)}$ on each iteration. (In our work three iterations have proved sufficient.)

4.3. Refining P *and computing camera position*

Once structure has been updated, $\mathtt{P}_i$ is recomputed, but this time using the updated 3D points with the observations in image i. The optical centre $\mathbf{Q}_i$ is then computed from the linear system $\mathtt{P}_i\mathbf{Q}_i = \mathbf{0}$.

4.4. Initialising new structure

The processing in this section has dealt with updating the position of existing 3D points. Of course during a sequence new feature points will appear. Once the second observation of a new point is obtained, the projection matrices for the two images can be used to recover its 3D position using the projective stereo method of Section 3.3.

5. Integration of matching and structure recovery

Thusfar we have presented the theory required to establish a projective frame and to compute 3D structure within it *assuming* the availability of a set of corner matches between successive images in a sequence. This section addresses the issue of how to obtain corner matches. Correspondence matching is carried out automatically in a three stage process, and no knowledge of camera calibration or camera motion (apart from a threshold on maximum disparity) is assumed.

Image corners are extracted to sub-pixel accuracy using the corner detector of Harris & Stephens (1988). In stage 1, each corner in the first image is matched against potential matches in the second image, subject only to a threshold on maximum disparity i.e. each corner has a search area which is a circle as shown in Figure 2(a). The match with strongest cross-correlation is accepted (an implicit assumption in using cross-correlation for matching is that cyclorotation around the principal axis is small). The radius of the search area can be up to 50 pixels, so there is a relatively large chance of obtaining a mismatch. Matches from stage 1 are

passed into stage 2, which begins with the computation of the fundamental matrix (which encodes the epipolar geometry) by a robust method. The effects are twofold - firstly mismatches can be identified and removed, because they do not agree with the epipolar geometry; secondly matching can be resumed on all unmatched corners, but the search area is reduced to an epipolar line as shown in Figure 2(b). Matches from stage 2 are passed into stage 3, which is employed when matching images k to $(k+1)$ of a sequence for $k \geq 2$. These matches provide a correspondence between existing 3D structure and the corners in image $(k+1)$, enabling the computation of the camera matrix $\mathtt{P}_{(k+1)}$ for image $(k+1)$. Once $\mathtt{P}_{(k+1)}$ has been found, matching can be carried out on unmatched 3D points, with a search area around the projected 3D point as shown in Figure 2(c).

Parameters such as cross-correlation and outlier thresholds which are used in the matching are supplied at the start of a sequence but are updated at the end of processing each image according to the current matching statistics.

Each of the matching stages is now described in more detail.

5.1. Stage 1: unguided matching

This initial, unguided matching stage is used to obtain a small number of highly reliable seed matches to be passed onto stage 2 and the computation of the fundamental matrix F.

As sketched in Figure 2(a), potential matches for an image feature at $\mathbf{x}_1$ in image 1 are sought within a radius of 30–40 pixels (for 256×256 image) of $\mathbf{x}_1$ in image 2. The matching strength for each is determined by measuring cross-correlation of image intensity over a 7×7 pixel patch, and the best match is accepted subject to a threshold, which is set deliberately high to minimise incorrect matches at this stage. Typically this stage will yield some 100–120 matches for 250–300 corner features in image 1.

Deriving initial matches

For each corner i in image 1:

1: Generate a list of of potential matches with corners j such that $|\mathbf{x}_{1i} - \mathbf{x}_{2j}| < r$, where $r \sim 30$–40 pixels.

2: For each corner j in the list:

2.1: Derive a matching strength S_{ij} using cross-correlation.

2.2: If $S_{ij} >$ threshold $\wedge\, S_{ij} > S_{ij^*}$, set j to be the best match $j^* \leftarrow j$.

If multiple corners in image 1 match the same corner in image 2, the match of highest strength is taken.

5.2. Stage 2: using epipolar geometry

The stage 1 matches are passed here. These matches are used to compute the fundamental matrix F [23] using a random sampling algorithm to mitigate the effects of outlying mismatches. Use of random sampling for this computation has been described in (Torr et al. 1994), (Deriche et al. 1994), (Zhang 1995), and a survey of robust methods is given in (Torr 1995).

F is computed using an iterative linear algorithm. This is satisfactory when F is being used to guide matching but for the most accurate requirements, such as the use of F in frame initialization, a non-linear technique is also employed in the computation as discussed later in Section 5.4. The non-linear technique also enables the application of the constraint that F has rank 2.

Once F is computed, correspondence matching is resumed for unmatched corners. The cross-correlation threshold used in §5.1 for acceptance of a match $\mathbf{x}_{1i}$ to $\mathbf{x}_{2j}$ is made more lenient, while the search area for $\mathbf{x}_{2j}$ is restricted to a band about the epipolar line generated using $\mathtt{F}\mathbf{x}_{1i}$. Using corners computed to sub-pixel accuracy, the typical distance of a point from its epipolar line is ∼0.2pixels. (If point positions are found only to pixel accuracy, this figure increases to ∼0.8pixels, whereas if positions are found by intersecting lines, as they are when using the reference object of Section 6, the figure falls to $\sim$ 0.02 pixels.) After this stage, there are typically 150–180 matches for 250–300 corners.

The algorithm in detail is as follows:

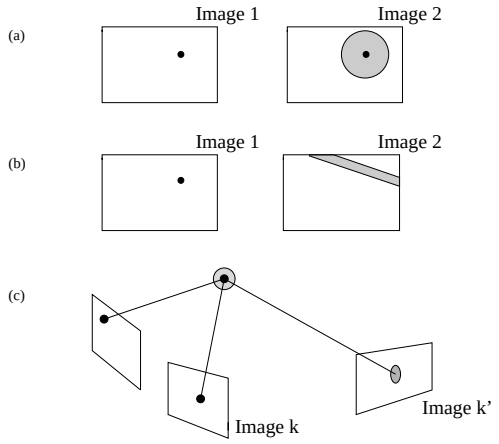


Fig. 2. Successive refinement of the search area during correspondence matching. (a) During unguided matching, the search area is limited only by maximum disparity. (b) Search area along an epipolar line. (c) Search area around a projected 3D point.

Deriving F to guide matching

Step 1: Select a random sample of 8 matches from the initial set.

Step 2: For each match $\mathbf{x}_{1i} \leftrightarrow \mathbf{x}_{2i}$ use $\mathbf{x}_{2i}^{\top}\mathtt{F}\mathbf{x}_{1i}$ to generate a homogeneous equation in the unknown elements $\mathbf{f} = (f_1, \ldots f_9)$ of F. Each equation gives a row in the 8×9 matrix B such that $\mathtt{B}\mathbf{f} = \mathbf{0}$, where each row of B is normalised to unit magnitude. Solve for $\mathbf{f}$ using SVD, and assemble $\mathbf{f}$ into F.

Step 3: For each match in the full set, determine $d_1 = d_\perp(\mathtt{F}\mathbf{x}_1, \mathbf{x}_2)$ and $d_2 = d_\perp(\mathtt{F}^{\top}\mathbf{x}_2, \mathbf{x}_1)$, where $d_\perp()$ returns the perpendicular distance between a point and the epipolar line. If d_1 and d_2 are below an outlier threshold (typically 1.25 pixels), mark the match as accepted.

Step 4: If the accepted matches form less than some percentage of the total (typically 75%), return to step 1. Otherwise, use all n accepted matches to construct a $n\times9$ matrix B′ such that $\mathtt{B}'\mathbf{f} = \mathbf{0}$, where each row of B′ is normalised to unit magnitude.

Step 5: From $\mathtt{B}'\mathbf{f} = \mathbf{0}$, compute $\mathbf{f}$ by SVD and assemble $\mathbf{f}$ into F.

Step 6: For every match being utilised in B′, determine the rms distance $\sqrt{((d_1^2 + d_2^2)/2)}$, and weight the corresponding equation in the matrix B′ by its inverse. The weighting is truncated to zero if the distance is greater than the outlier threshold used in Step 3.

Step 7: Repeat steps 5 and 6 until there is neglible change in the residuals computed from the current value of F, or a maximum iteration count (typically 6) is reached. On the final iteration, mark corner features which are further away from their epipolar line than the outlier threshold as unmatched.

5.3. Stage 3: use of 3D projective structure

The stage 2 matches are passed here. This stage is employed when matching images k to $(k+1)$ of a sequence for $k \geq 2$. For clarity, we put $(k+1) = k'$.

Corners $\mathbf{x}_{ki}$ in image k which have associated 3D coordinates $\mathbf{X}_i$ and which are matched to corners $\mathbf{x}_{k'i}$ in image k' provide a correspondence between $\mathbf{X}_i$ and $\mathbf{x}_{k'i}$. Each correspondence obeys the relationship $\mathbf{x}_{k'i} = \mathtt{P}_{k'}\mathbf{X}_i$. The processing to compute $\mathtt{P}_{k'}$ from these correspondences is closely analogous to that in the previous section.

Once $\mathtt{P}_{k'}$ has been computed, correspondence matching is continued for unmatched corners $\mathbf{x}_{ki}$ which have associated 3D coordinates $\mathbf{X}_i$. As sketched in Figure 2(c), the search area in image k' is determined by projecting the uncertainty ellipsoid of the 3D point. The r.m.s. distance between projected 3D points and their corresponding image points is some 0.3 pixels for corners obtained by corner detection (and 0.02 pixels for points found by line intersection on the reference object). After this stage, there are typically 180-190 matches for 250–300 corners.

In detail, the algorithm is as follows:

Deriving P to guide matching

Step 1: Identify the set of stage 2 matches which provide a correspondence between 3D points $\mathbf{X}_i$ and corners $\mathbf{x}_{k'i}$ as described above.

Step 2: Take a random sample of 6 correspondences from the identified correspondences.

Step 3: For each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_{k'i}$ in the sample, use the relationship $\mathbf{x}_{k'i} = \mathtt{P}_{k'}\mathbf{X}_i$ to

generate two homogeneous equations in the unknown elements $\mathbf{p} = (p_1, \ldots p_{12})$ of $\mathtt{P}_{k'}$. (Each correspondence gives three linear homogeneous equations in the unknown elements of $\mathtt{P}_{k'}$ and in the unknown homogeneous scale factor; eliminating the scale factor leaves two linear homogeneous equations.) These two homogeneous equations contribute two rows to a 12×12 matrix $\mathtt{D}$ such that $\mathtt{D}\mathbf{p} = \mathbf{0}$ where each row of $\mathtt{D}$ is normalised to unit magnitude. Solve for $\mathbf{p}$ using SVD and assemble $\mathbf{p}$ into $\mathtt{P}_{k'}$.

Step 4: For every correspondence in the full set, use $\mathtt{P}_{k'}$ to project the uncertainty ellipsoid of $\mathbf{X}_i$ onto the image plane, and if $\mathbf{x}_{k'i}$ lies within the 95% confidence limit (see [39]) mark the correspondence as accepted.

Step 5: If the percentage of accepted correspondences in the full set is less than a threshold (typically 75%), return to step 2. Otherwise, use all n accepted correspondences to construct a $2n\times12$ matrix $\mathtt{D}'$ such that $\mathtt{D}'\mathbf{p} = \mathbf{0}$, where each row of $\mathtt{D}'$ is normalised to unit magnitude.

Step 6: From $\mathtt{D}'\mathbf{p} = \mathbf{0}$, compute $\mathbf{p}$ by SVD and assemble $\mathbf{p}$ into $\mathtt{P}_{k'}$.

Step 7: For every correspondence being utilised in $\mathtt{P}_{k'}$, determine the image plane distance $\| \mathbf{x}_{k'i} - \mathtt{P}_{k'}\mathbf{X}_i \|$, and weight the two associated equations in the matrix $\mathtt{D}'$ by its inverse. The weighting is truncated to zero if $\mathbf{x}_{k'i}$ lies outside the confidence region used in Step 4.

Step 8: Repeat steps 6 and 7 until there is neglible change in the residuals computed from the current value of $\mathtt{P}_{k'}$, or a maximum iteration count (typically 6) is reached. On the final iteration, corners which lie outside their associated projected confidence region are judged to be incorrect matches, and are marked as unmatched.

5.4. Non-linear refinement of $\mathtt{F}$ *and* $\mathtt{P}$

The linear estimate of the fundamental matrix proves sufficient for processing arising in the course of an image sequence when $\mathtt{F}$ is being used only to guide correspondence matching. For frame initialisation however it is worth the computational expense of refining the estimate using non-linear optimisation methods: we used the Powell method [32]. The error measure minimised to obtain $\mathtt{F}$ is the sum of the squares of the perpendicular distances of each matched point from its epipolar line [9]

$$e = \sum_{\text{matches } ij} [d_\perp(\mathbf{x}_j, \mathtt{F}\mathbf{x}_i)^2 + d_\perp(\mathbf{x}_i, \mathtt{F}^\top\mathbf{x}_j)^2]$$

thus minimising an image plane distance rather than an algebraic error as in the linear computation. The improvement obtained in the average corner-epipolar line distance is typically small, about 0.01 pixels, but this may result in a movement of many tens of pixels in the position of the epipoles obtained from $\mathtt{F}$.

The non-linear optimisation also serves a second purpose: unlike linear processing it permits enforcement of the constraint that $\text{Rank}(\mathtt{F}) = 2$ [23], by making the third row of $\mathtt{F}$ a linear combination of the first two rows [8].

Turning to $\mathtt{P}$, because of its use in computing and updating the 3D structure, the highest accuracy estimate is required and therefore non-linear refinement is always used. The error measure minimised is the sum of the squares of the image distances between corners on the image plane and the projection $\mathtt{P}\mathbf{X}$ of the 3D structure,

$$e = \sum \| \mathbf{x} - \mathtt{P}\mathbf{X} \|^2$$

6. Results for projective SFM

The primary questions addressed in the experimental work are:

- How does the quality of recovered structure in a projective system with uncalibrated cameras compare with that from a Euclidean system which utilises full and accurate camera calibration and estimates of camera motion?
- How much skew is there between the Quasi-Euclidean frame of Section 3.3 and a Euclidean frame, when utilising approximate camera calibration and motion to initialise the Quasi-Euclidean frame?
- How does the quality of structure compare in a Quasi-Euclidean frame with that in a frame with a large projective skew?

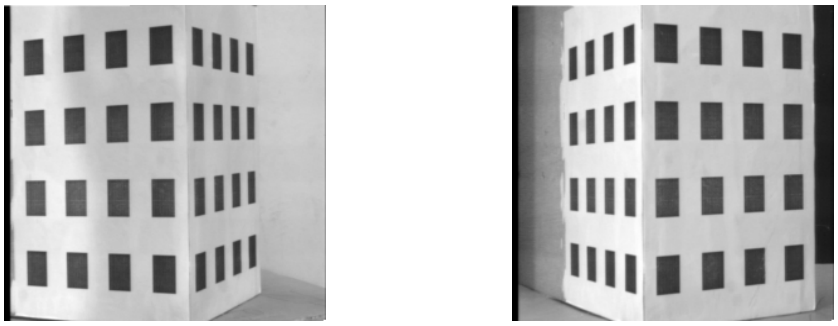


Fig. 3. The first and last images from a sequence of fifteen images of the reference object.

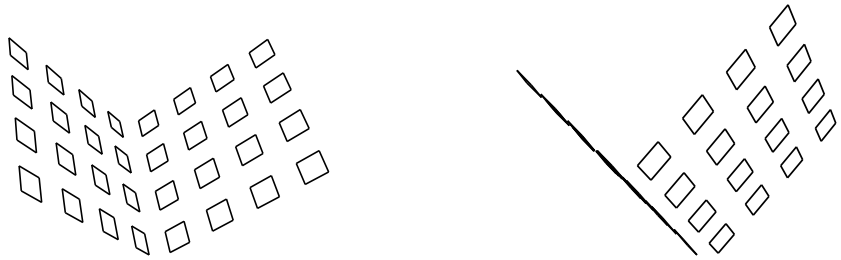


Fig. 4. Structure of the reference object in a Quasi-Euclidean coordinate frame (connectivity has been added to the point structure for illustration). Although Euclidean relationships such as perpendicularity are not preserved exactly in a Quasi-Euclidean frame, it is evident that they are approximately true. The right-hand figure is viewed with one of its planes edge-on to show coplanarity in the recovered structure.

Experiments have been carried out in two types of environment, the first a camera mounted on a robot arm viewing a reference object made of two perpendicular Tsai calibration grids, and the second a camera mounted on a robot arm or a mobile vehicle viewing an indoor laboratory scene. The first environment allows us to make quantitative assessments of the recovered structure, and we do so in two ways: by measuring projective invariants directly from the recovered structure, and by transforming to a strictly Euclidean coordinate frame (using the known Euclidean structure of the reference object) and measuring Euclidean invariants. The reference object provides a common reference coordinate frame, allowing proper comparison between the quality of the projective structure with that obtained from conventional Euclidean algorithms. For the indoor laboratory environment, results are presented in the Quasi-Euclidean frame allowing qualitative assessment.

The intrinsic parameters of the camera used in the experiments are given in example 1 in §3.4. The approximate camera parameters used to set up the Quasi-Euclidean frame in §6.1 and §6.4 are also given in example 1 in §3.4. The approximate camera parameters used to set up the non Quasi-Euclidean frame in §6.2 are given in example 2 in §3.4.

For the sequence in Figure 3, the camera is moving in a horizontal circular arc while fixating on the reference object some 80cm away. The translation between each image is about 2cm and the rotation is about 2° around a vertical axis. For the sequence in Figure 10, each motion is a translation in the horizontal plane of about 50cm and a rotation of about 3° around a vertical axis. Part of a sequence from the navigation experiments which are described in §8 appears in Figure 6. Typical motions during navigation are a translation of about 3cm and a rotation of 2°-3°.

6.1. Reference object in the Quasi-Euclidean frame

The first and last images from a sequence of fifteen of the reference object are shown in Figure 3. To obtain the "best feasible" structure for the reference object, point positions on the grid

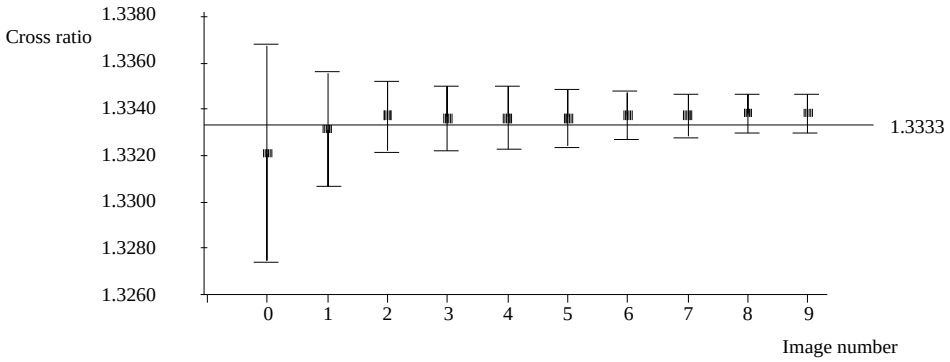


Fig. 5. The mean values and standard deviations determined for the 32 cross ratios computed from the recovered projective structure updated during an image sequence. The expected ratio of 4/3 is shown as a solid line.

Table 1. A comparison with expected geometric values of results obtained using the present projective algorithm, the DROID Euclidean algorithm, and the affine specialisation (discussed in Section 7). Coplanarity is a mean value for the two faces of the reference object. Distance ratio is the ratio of two equal lengths on the reference object. For the projective structure the cross-ratio measurement was made before transformation to the Euclidean frame, and the remaining measures after. For the affine structure, the cross-ratio and distance ratio measurements were made before transformation to the Euclidean frame, and the remaining measures after. 128 points were used to compute the transformation to the Euclidean frame. The point error is the average distance between a transformed point and the veridical Euclidean point, in the Euclidean frame.

Point in Sequence	Measure	Expected value	Projective	Affine	DROID
After 2 images	Point error (cm)	0.0	0.2	0.3	0.3
	Collinearity	0.0	0.003	0.005	0.006
	Coplanarity	0.0	0.004	0.006	0.007
	Cross-ratio	4/3	1.332 ± 0.006	1.333 ± 0.003	1.332 ± 0.005
	Distance ratio	1.0	0.999 ± 0.012	1.002 ± 0.009	1.000 ± 0.013
After 20 images	Point error (cm)	0.0	0.1	0.2	0.2
	Collinearity	0.0	0.002	0.002	0.004
	Coplanarity	0.0	0.002	0.003	0.004
	Cross-ratio	4/3	1.333 ± 0.002	1.333 ± 0.001	1.333 ± 0.002
	Distance ratio	1.0	1.000 ± 0.004	1.000 ± 0.006	0.999 ± 0.007

are determined not from the corner detector, but by intersecting lines. Figure 4 shows the structure of the reference object recovered in a Quasi-Euclidean frame. Although Euclidean relationships such as perpendicularity are not preserved, it is evident that the violation is small.

The cross-ratio is a projective invariant, and can be measured directly from the recovered structure. Four equally spaced collinear points have a cross-ratio of 4/3. Thirty-two such cross-ratios are computed for each image of a sequence, and the results plotted in Figure 5. The measured cross-ratio improves with the sequential update, and converges to the predicted value.

Because the reference grid has known structure, it is possible to transform the recovered structure to a strictly Euclidean coordinate frame. The transformation can be determined using the coordinates of five or more points in the Quasi-Euclidean and Euclidean frames [35] (we employ all 128 points on the reference object in a least-squares computation), where direct physical measurement on the reference object provides the Euclidean coordinates.

Comparison between the expected and measured values in columns 3 and 4 of Table 1 provides an overall assessment of the quality of the recovered projective structure, by showing cross-ratios computed before transformation to the Eu-

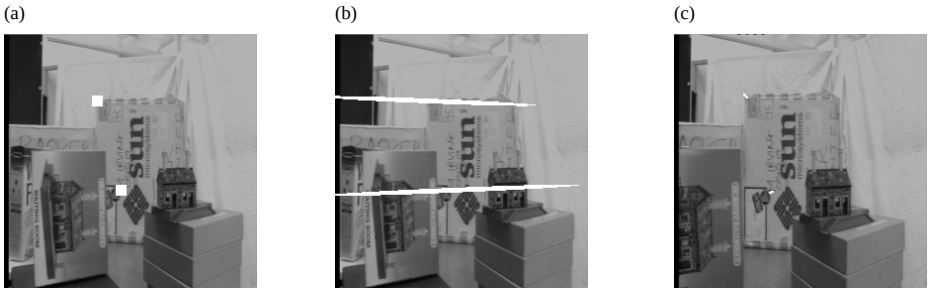


Fig. 6. (a) The large white blocks mark two example corners. (b) Uncertainty ellipsoids for the recovered 3D points, projected to ellipses on the image plane, after one update in the sequential update scheme. (c) The ellipses after four updates. The ellipses have shrunk rapidly as the uncertainty in the 3D points is reduced by successive observations, and the major and minor axes are about 6 pixels and about 1 pixel (approximately 0.4° and 0.1°), respectively.

clidean coordinate frame and other measurements like collinearity and coplanarity made after the transformation to ensure that all such measurements are in a single reference coordinate frame. The collinearity measure $L = (\sigma_j^2 + \sigma_k^2)^{1/2}/\sigma_i$ and the coplanarity measure $P = \sigma_k/(\sigma_i^2 + \sigma_j^2)^{1/2}$ of a set of points are obtained by using SVD to obtain the principal axes i, j, k together with the variance σ_i, σ_j, σ_k of point positions along each principal axis. A straight line is thus expected to have $L = 0$ and a plane to have $P = 0$. Note that all measures converge as more images are considered.

Column 6 of Table 1 also provides a comparison with a local implementation of the DROID system (Harris 1987; Harris & Pike 1987) which computes Euclidean structure directly, requiring, of course, exact camera calibration and approximate camera motion. Evidently there is no significant difference between the quality of our projective algorithm and the DROID Euclidean algorithm, though we note again that no camera calibration is required in the projective case.

As we discussed in Section 3.4, varying the approximate values of camera intrinsic parameters and camera rotation used to set up the Quasi-Euclidean frame produces different amounts of projective skew. To test for the effects of skew, values for the camera intrinsics used in setting up the Quasi-Euclidean frame were varied up to 20% from their true values, and the camera rotation was approximated by setting the rotation to zero. It was found that this level of variation had no effect on the assessments listed in Table 1.

Finally, Figure 6 shows some examples of the uncertainty ellipsoids for recovered 3D points, projected onto the image plane. Each example is computed by taking the uncertainty ellipsoid for a 3D point at image i and projecting it onto image $i + 1$, finding the ellipse which has a 95% likelihood of containing the new observation of the 3D point[39]; such ellipses are used to define search areas for correspondence matching (Section 5.3).

6.2. Reference object in a non Quasi-Euclidean frame

Section 6.1 dealt with structure recovered in a Quasi-Euclidean frame. In this sub-section, structure and camera position are computed in a coordinate frame which has a large projective skew away from being Euclidean. An example transformation $\mathtt{H}_P$ between such a frame and a Euclidean frame was given earlier in Section 3.4. Figure 7 shows structure recovered for the reference object in such a frame. The transformation away from the true Euclidean form is evident in the projective skew of the grid itself, and also in the camera positions relative to the reference object. Figure 8 shows the structure and camera positions from Figure 7 after transformation to a Euclidean frame.

6.3. Comparison of structure in the Quasi-Euclidean and non Quasi-Euclidean frames

A comparison between measurements on 3D structure in a Quasi-Euclidean frame and in a frame with a large projective skew is given in Table 2,

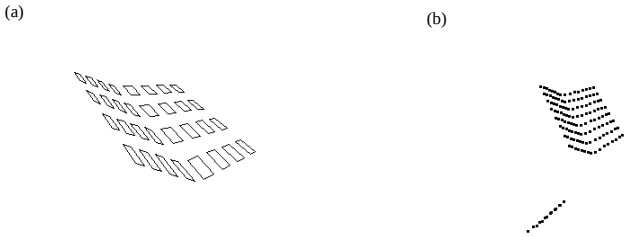


Fig. 7. (a) Structure of the reference object in a coordinate frame with a large projective skew — coplanarity and collinearity are preserved as expected, but the structure is skewed along one plane, and the angle between the two planes is greater than 90° (connectivity has been added to the point structure for illustration). (b) View of the computed structure together with the computed camera positions in the frame with large projective skew. Compare with the plan view after transformation to the Euclidean frame in Figure 8.

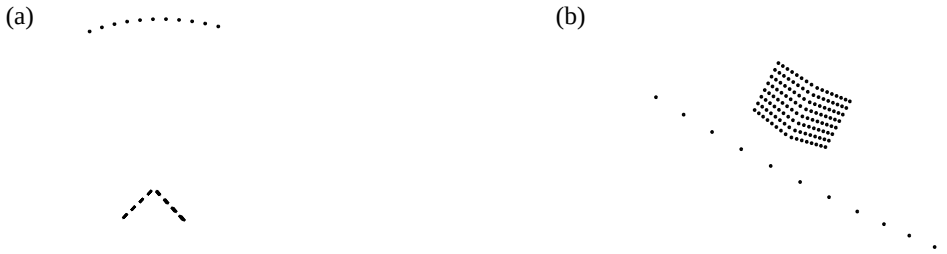


Fig. 8. (a) Plan view of the reference object viewed edge-on (lower) and the arc of successive camera positions in a circle (upper) after transformation to a Euclidean frame. Note the perpendicularity of the planes of the reference object. (b) View from behind the arc of camera positions.

and shows the superiority of the Quasi-Euclidean frame. Measurements are given after 2 images when the structure has just been initialised, after 3 images when there has been one update, and after 10 images. There is a further partition of results according to the level of localisation accuracy in the image points utilised to compute the structure. Three levels of localisation accuracy were explored. First, using line intersection to compute point positions, an accuracy of 0.02 pixels is obtained. Secondly, the positions are rounded to the nearest 0.1 pixels and the structure recomputed, and finally rounded to the nearest integer pixel and the structure again recomputed.

One trend evident in the results is that in all measures the accuracy is better in the Quasi-Euclidean frame. A second trend is that the difference in accuracy between the Quasi-Euclidean and non Quasi-Euclidean diminishes as information from more images is integrated, although when using the reference object no new structure is being introduced between frames. A third observation is that as the localisation accuracy is reduced to nearest pixel, the non Quasi-Euclidean reconstruction cannot be continued: the initial structure is so erroneous that the computed perspective projection matrix results in predicted image positions differing from their veridical positions by greater than 10 pixels.

Figure 9 gives a more detailed graphical comparison of of the point error (the first measure in Table 2) in the structure recovered in a Quasi-Euclidean frame and in a non Quasi-Euclidean frame with a large projective skew for a range of corner localisation accuracies. The recovered structure is transformed to the Euclidean coordinate frame of the reference object, and the average distance between transformed points and the veridical positions of points on the reference object measured. The point errors are plotted over a sequences of 11 images. Notice that at a localisation error of 1 pixel no improvement in the Quasi-Euclidean structure is discernible over this time scale. At the finer localisations, the Quasi-Euclidean always out-performs the non Quasi-Euclidean reconstruction.

Table 2. A comparison between measurements on 3D structure in a Quasi-Euclidean frame and in a frame with a large projective skew, showing the superiority of the Quasi-Euclidean frame. Measurements are shown at three times during the sequences — after 2 images when the structure has just been initialised, after 3 images, and after 10 images — and for three resolutions of image point positions — to 0.02 pixels, 0.1 pixels and 1 pixel. The blank entries indicate a failure to recover structure, as described in the text.

Measure	Exp val	Quasi-Euclidean	Non Quasi-Euclidean	Quasi-Euclidean	Non Quasi-Euclidean	Quasi-Euclidean	Non Quasi-Euclidean
		Resolved to 0.02 pixel		*Resolved to 0.1 pixel*		*Resolved to nearest pixel*	
At 2 images:							
Pt error (cm)	0.0	0.18	0.75	0.39	2.49	3.69	4.14
Collinearity	0.0	0.004	0.015	0.028	0.092	0.046	0.17
Coplanarity	0.0	0.007	0.021	0.019	0.093	0.039	0.10
Cross-ratio	4/3	1.337 ± 0.007	1.337 ± 0.011	1.339 ± 0.024	1.334 ± 0.059	1.328 ± 0.019	1.24 ± 0.24
Distance ratio	1.0	1.016 ± 0.020	1.003 ± 0.026	1.017 ± 0.059	0.984 ± 0.194	0.945 ± 0.048	0.90 ± 0.24
At 3 images:							
Pt error (cm)	0.0	0.18	0.79	0.39	2.53	3.41	—
Collinearity	0.0	0.004	0.016	0.028	0.095	0.170	—
Coplanarity	0.0	0.007	0.021	0.018	0.093	0.149	—
Cross-ratio	4/3	1.337 ± 0.007	1.338 ± 0.012	1.339 ± 0.025	1.334 ± 0.064	1.290 ± 0.098	—
Distance ratio	1.0	1.015 ± 0.020	1.005 ± 0.031	1.017 ± 0.060	0.989 ± 0.209	0.951 ± 0.065	—
At 10 images:							
Pt error (cm)	0.0	0.08	0.06	0.09	0.30	3.63	—
Collinearity	0.0	0.002	0.002	0.004	0.007	0.072	—
Coplanarity	0.0	0.003	0.003	0.003	0.012	0.065	—
Cross-ratio	4/3	1.335 ± 0.002	1.335 ± 0.002	1.335 ± 0.003	1.334 ± 0.003	1.317 ± 0.046	—
Distance ratio	1.0	1.007 ± 0.008	1.009 ± 0.009	1.007 ± 0.010	1.005 ± 0.008	0.942 ± 0.044	—

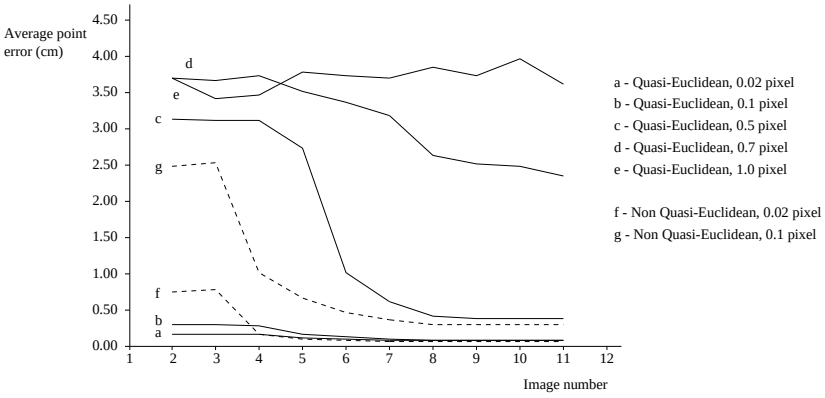


Fig. 9. Comparison of the time evolution of the point error in the 3D structure recovered in the Quasi-Euclidean frame and in a non Quasi-Euclidean frame with a large projective skew, for a range of corner localisation accuracies. The method of reconstruction by determining the midpoint of two backprojected rays produces structure of significantly better quality in the Quasi-Euclidean frame.

6.4. Structure from motion of a mobile vehicle

Figures 10(a and b) show the first and last images of a twelve image sequence taken by a camera mounted on a mobile vehicle which translated forward along a corridor while turning to the left. The maximum depth of the scene is about 7m. Corners were obtained using the sub-pixel corner detector discussed earlier. Figures 10(c and d) show the structure recovered in a Quasi-Euclidean frame rendered from two different vantage points. Notice that Euclidean relationships such as perpendicularity of the side wall and floor are approximately correct, and the quality of the reconstruction remains high even at the most distant points.

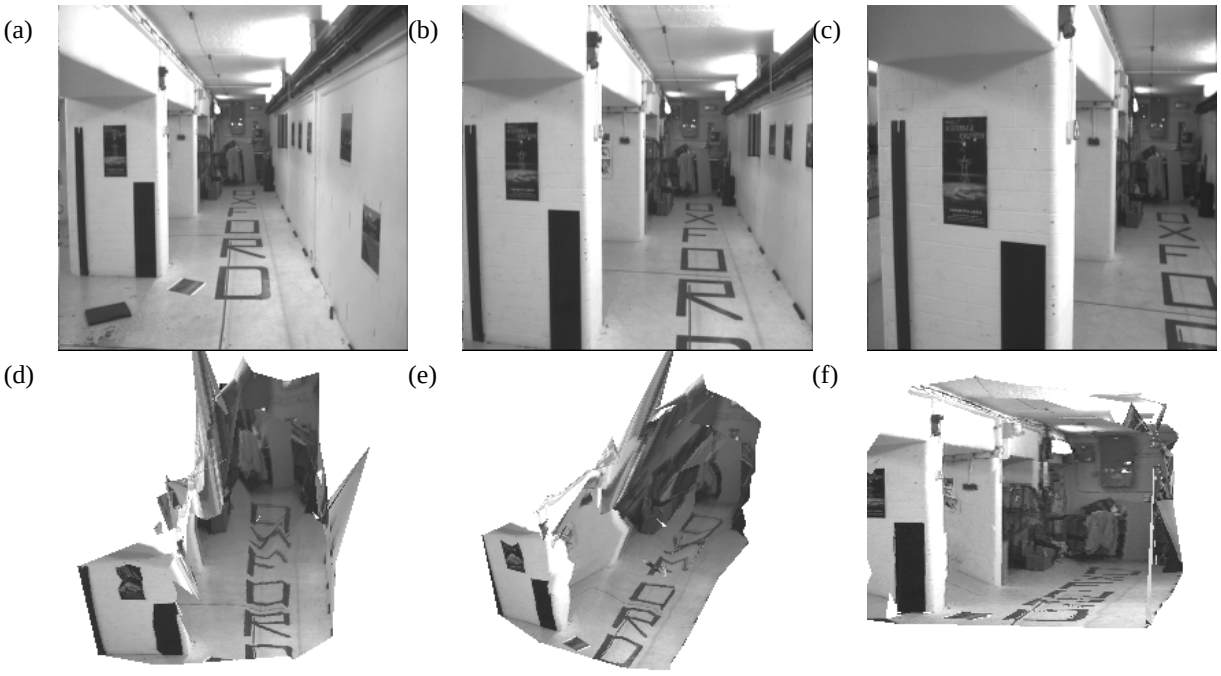


Fig. 10. (a)-(c) Three images of a sequence taken by a camera mounted on a mobile vehicle as it moves forward and turns left. (d)-(f) Recovered 3D structure in the Quasi-Euclidean frame, viewed from novel viewpoints not obtained during the sequence. (The overlaid image texture is created by mapping between Delaunay triangulations of the 2D image corners.)

6.5. Summary of results

The questions posed at the start of this section can now be related to the experimental results.

- There is effectively no difference in the quality of the structure recovered by the Quasi-Euclidean projective algorithm and the strictly Euclidean DROID system, although the latter utilises accurate camera calibration. Table 1 shows that both at initialisation and at a later stage in the sequence, all the evaluation measures are similar. Note the accuracy of recovered structure: the structure is accurate to within 1mm, after transformation to a Euclidean frame, for an object at a distance of 80cm.
- The initialisation of a suitable Quasi-Euclidean frame is not sensitive to the particular approximation of camera calibration and motion (Section 6.1).
- The Quasi-Euclidean frame produces superior structure to a coordinate frame which has a large projective skew away from Euclidean, as shown in Figure 9. This is a consequence of the method used to initialise 3D points. The method has the requirement that the coordinate frame is Quasi-Euclidean (whereas there are approaches which avoid this as discussed in Section 3.3), but offers the most straightforward and computationally efficient way of handling error in the image measurements.
- Sequential update of the structure over time using an iterated EKF provides a way of integrating many observations of a point in a computationally efficient way. There is no guarantee of optimality or convergence with an EKF but empirically the quality of the recovered structure improves under the sequential update as demonstrated in Figures 5 and 9, and the structure has been found to be stable provided the number of gross mismatched corners is reduced using the outlier detection methods described in Section 5.

7. Affine Structure from Motion

The previous sections of the paper, particularly Section 3, have dealt with the computation of projective structure. Here we specialise our approach to recover affine structure. The objective of the affine structure from motion algorithm is to use correspondences between corners $\mathbf{x}$ in a sequence of images to recover the structure $\mathbf{X}$ of the scene, modulo an affine transformation. That is, if the Euclidean structure of the scene is $\mathbf{X}_E$, the recovered structure is

$$\mathbf{X} = \mathtt{H}_A \mathbf{X}_E$$

where $\mathbf{X} = (X, Y, Z, 1)^\top$, $\mathbf{X}_E = (X_E, Y_E, Z_E, 1)^\top$, and $\mathtt{H}_A$ is an affine transformation which is undetermined but the same for all points:

$$\mathtt{H}_A = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

with $\mathtt{A}$ a non-singular 3×3 matrix and $\mathbf{t}$ a 3-vector.

An affine coordinate frame differs from a projective coordinate frame because the *plane at infinity* π_∞ has been identified (Semple & Kneebone 1952).

Our approach is a variation on a result of Moons et al. (1994) who showed that affine structure can be obtained from a perspective camera with fixed intrinsic parameters undergoing pure translational motion. Unlike their method which is based on a small fixed number of image points, we use all available image points to set up the affine frame. Note that affine structure is being obtained from perspective images, and there is no need to assume affine imaging conditions; that is, there is no need to assume weak or paraperspective cameras. (Sequential computation of affine structure under affine imaging conditions is described in [26].)

7.1. Setting an affine coordinate frame

This section describes initialisation of an affine coordinate frame. Unlike the projective case where there are several significant modifications to obtain a Quasi-Euclidean frame, only one modification to the basic processing is required to obtain a Quasi-Euclidean affine coordinate frame. To maintain consistency with the projective case, we will call this Step 0 in the following listing:

Setting an Affine Frame

Step 0 (optional): If the coordinate frame is to be Quasi-Euclidean, normalize the image coordinates $\mathbf{x} \leftarrow (\mathtt{C}^*)^{-1}\mathbf{x}$ in both images.

Step 1: Set $\mathtt{P}_1 = [\mathtt{I}|\mathbf{0}]$.

Step 2: Determine the fundamental matrix $\mathtt{F}$ (a skew matrix here). Use it to determine the epipole $\mathbf{e}_2$ in the second image.

Step 3: Set $\mathtt{P}_2 = [\mathtt{I}|\mathbf{t}] = [\mathtt{I}|\lambda\mathbf{e}_2]$

Step 4: Backproject the rays and determine the midpoint of mutual perpendicular.

At Step 2, the epipole $\mathbf{e}_2$ is obtained as before from the fundamental matrix computed for the two images. Here, however, $\mathtt{F}$ has a special form because the intrinsic parameters are fixed and the camera motion between image 1 and 2 is a pure translation. For this special situation, and in a Euclidean frame, the perspective projection matrices for the images 1 and 2 have the form

$$\mathtt{P}_1 = \mathtt{C}[\mathtt{R}|\mathbf{t}_1] \quad \text{and} \quad \mathtt{P}_2 = \mathtt{C}[\mathtt{R}|\mathbf{t}_2] \ .$$

From equation (5), the fundamental matrix is then

$$\mathtt{F} = [\mathtt{CR}]^{-\mathsf{T}}[\mathbf{t}_1 - \mathbf{t}_2]_\times[\mathtt{CR}]^{-1}$$

which is of the form $\mathtt{F} = \mathtt{A}^{\mathsf{T}}\mathtt{SA}$ where $\mathtt{S}$ is skew-symmetric. It follows that $\mathtt{F}$ is skew symmetric. Further, since $\mathtt{F}$ is unaffected by a projective transformation of the world frame, the same argument holds in any coordinate frame, not just a Euclidean one. Because $\mathtt{F}$ is skew symmetric, it has only three distinct homogeneous elements or two degrees of freedom, as opposed to seven degrees of freedom in the general case. This substantial reduction in the number of unknowns makes the computation more efficient and better conditioned. The skew form also means that the rank 2 condition on $\mathtt{F}$ is imposed automatically during the linear computation. (As we saw earlier, this is not possible for the general form, where the rank 2 condition must be imposed in a non-linear step.)

Note that at Step 2, unlike the projective case, we do not need to compute $\mathtt{M}_2$ and use it in setting $\mathtt{P}_2$. Instead, for a camera with fixed intrinsic parameters undergoing a pure translation, the much simpler choices of $\mathtt{P}_1 = [\mathtt{I}|\mathbf{0}]$ and $\mathtt{P}_2 = [\mathtt{I}|\lambda\mathbf{e}_2]$ are valid. This sets the plane at infinity to $\pi_\infty = (0, 0, 0, 1)^{\mathsf{T}}$, the conventional value for an affine frame. This may be verified using Lemma 2 in Appendix C.

7.2. Updating the affine coordinate frame

The previous section addressed the initialisation of an affine coordinate frame. Here we describe a method for transforming an *existing* arbitrary projective coordinate frame into an affine frame, using a pure translational motion of the camera. This can be used to update an existing affine frame which has "drifted" slightly over time. Strictly this should be unnecessary because the coordinate frame is fixed once it has been set up, but in practice the need to recompute the frame can arise for two reasons. First, as structure is updated the plane at infinity may drift due to error. Secondly, the motion made in order to determine the plane at infinity might not be pure translation, and the error which arises in an individual measurement can be overcome by making repeated measurements.

Updating an Affine Frame

Step 1. At the current step k determine $\mathtt{P}_k$ for the camera position in the established coordinate frame.

Step 2. Keeping the intrinsic parameters fixed, make the camera undergo pure translation. Determine $\mathtt{P}_{k+1}$ for the new camera position in the established coordinate frame.

Step 3. Transform the coordinate frame so that $\mathtt{P}_k$ takes the canonical form $\mathtt{P}'_k = [\mathtt{I}|\mathbf{0}]$. Apply the same transformation to $\mathtt{P}_{k+1}$ to obtain $\mathtt{P}'_{k+1} = [\mathtt{M}_{k+1}|\ \mathbf{t}^*]$.

Step 4. Decompose $\mathtt{M}_{k+1}$ into $[\lambda\mathtt{I} + \mathbf{t}^*\mathbf{v}^{\mathsf{T}}]$ using Lemma 2, where λ is a scale factor, and $\mathbf{v}$ a vector. The plane at infinity is $\pi_\infty = (\mathbf{v}^{\mathsf{T}}, 1)$.

Step 5. Transform the whole coordinate frame once more so that the plane at infinity takes its conventional form of $\pi_\infty = (0, 0, 0, 1)$.

Step 4 exploits Lemma 2 which, with its proof, is given in Appendix C. Also in Appendix C is the method for decomposing $\mathtt{M}_{k+1}$ so that $\mathbf{v}$ can be found.

7.3. Results for affine SFM

Experiments similar to those for projective structure in Section 6 were carried out. Quasi-Euclidean affine structure is computed for two image sequences, one of the reference object and the other of an indoor scene. Assessment is by (i) measurement of affine invariants directly from the recovered structure and (ii) measurements on the structure after transformation to a Euclidean frame.

The reference object. The ratio of distances on parallel lines is an affine invariant. Ratios were measured from the affine structure of the reference object for thirty-two triples of equidistant collinear points, each triple defining a ratio of unity. The variation of the mean and standard deviation over an image sequence is shown in Figure 11, and the value is evidently converging to the expected value of unity. This value, and the other error measures, are compared with results from the projective and Euclidean DROID algorithms in Table 1.

Indoor scene. Figure 12 shows results for an indoor sequence. The camera was translated laterally in front of a scene comprising a variety of boxes. Two views of the recovered structure computed in a Quasi-Euclidean affine frame are shown, one from above, and the other laterally from the right and behind. We shall explore this structure further in the next section as we use it to drive an affine path planning algorithm.

7.4. Summary

The experimental results for affine structure echo the conclusions already listed in Section 6: viz. that the quality of structure is the same for uncalibrated and calibrated systems (see Table 1), and that structure improves over time.

However, it is worth recalling our introductory remarks about the significant advantages affine structure presents over projective structure in terms of the extra invariants available, invariants which appear to offer more scope for interaction with the environment than does the fundamental invariant in projective structure, the cross-ratio. In the next section we demonstrate the use of midpoint invariance, and the quality of the affine reconstruction, by using the structure to carry out path-planning for navigation.

8. Navigation in Affine Space

The affine SFM scheme provides the basis for navigation through an unknown environment populated with unmodelled obstacles. We investigate to what extent affine structure can be used for a task traditionally carried out with Euclidean information. The experimental setup is a camera mounted on a robot arm moving in a horizontal plane and rotating around a vertical axis. The objective is to reach a target position specified in the robot's coordinate frame. The area in between the start and target positions is unknown and may contain obstacles as illustrated in Figure 13.

8.1. Structure recovery

Processing begins with initialisation of a Quasi-Euclidean affine coordinate frame as described in Section 7, and sequential update of affine scene structure with initialisation of newly appearing points using the methods from Section 4. Remaining stages involve incremental acquisition of free space regions, path planning through the free space, and finally control of the robot.

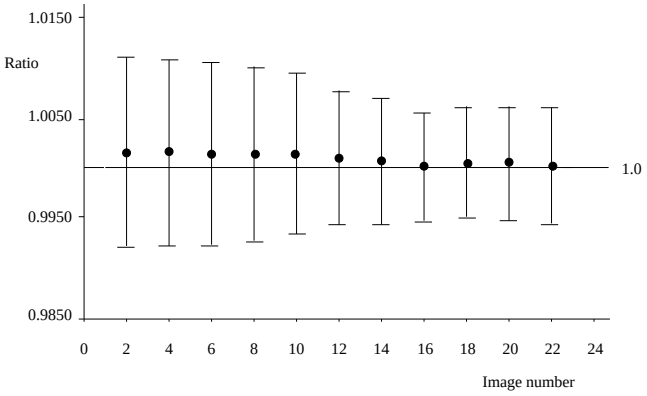


Fig. 11. Ratios computed from recovered affine structure against image number. The points and error bars show the mean and standard deviation for a fixed number (thirty-two) of ratio values computed at each image. The horizontal line shows the expected value of unity for the ratio.

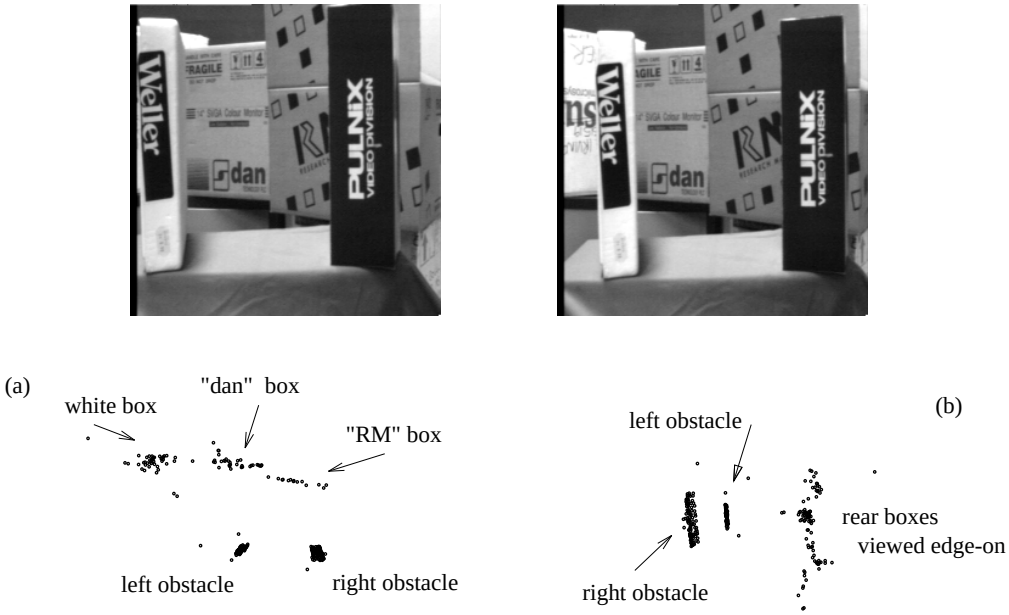


Fig. 12. Two images from a sequence with structure recovered in a Quasi-Euclidean affine frame. (a) Plan view of recovered structure. (b) View from the right and to the rear of the obstacles.

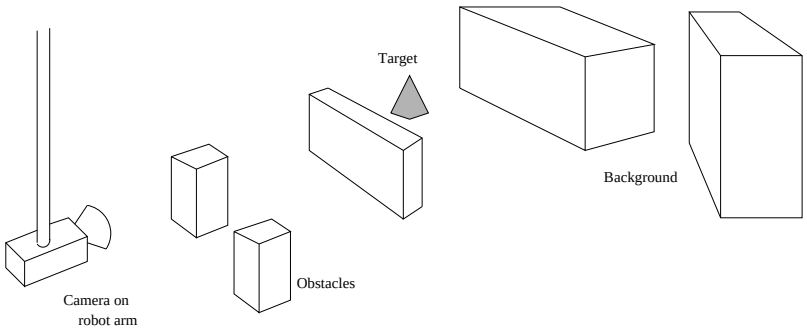
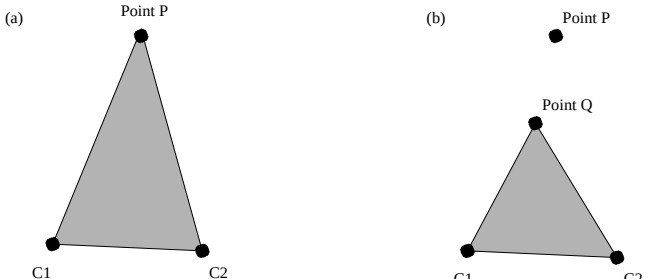


Fig. 13. The experimental setup. A camera carried by a robot arm manouevres to a target through an environment of unmodelled obstacles.



arm or mobile vehicle which is constrained to execute motion in the plane and is well modelled by a vertical cylinder. The recovered structure and camera positions are projected onto the ground plane using a method described in Section 8.4.

Computation of free space is complicated by the fact that the recovered structure consists solely of points so there is no representation of continuous surfaces, and thus no notion of objects or the free space between objects. We begin with the assumptions first that recovered points lie on surfaces and so are not isolated in space, and second that points cover each surface with sufficient density to make the surface detectable — that is, there are no large homogeneous regions on surfaces (the latter assumption is defined more rigorously below). A simple occlusion test is then used to detect free space as sketched in Figure 14a. Consider first the 3D information prior to projection to the ground plane. If a scene point P is visible continuously as the camera moves from C_1 to C_2 (this may be over several images rather than between consecutive images), then there is no occluding surface in the triangle defined by C_1PC_2. The projection of this free space triangle to the ground plane defines a free space triangle on the 2D map.

A necessary modification of this test, shown in Figure 14b, is that the projection of C_1PC_2 onto the ground plane is accepted as free space only if no other projected 3D point Q lies within that triangle. The modification is required for a number of reasons. Firstly, point Q might arise from a low object O in the foreground while P is a point which is visible above and to the rear of O, in which case the projected C_1PC_2 clearly should not be accepted as free space because it overlays O; this situation relates to the assumption made at the start of the paragraph that O must generate points in "sufficient density" to indicate its presence and prevent the acceptance of free space triangles which overlay it. Secondly, the modification deals with the case of concave objects where P arises from a point within a concavity and Q is a point on the convex hull of the projected object i.e. we prevent the marking of the inside of the concavity as free space on the projected map. Thirdly and finally, the modification is conservative, and prevents the acceptance of free space triangles when there is a mismatched or badly localised point present.

The complete free space map is the union of all accepted triangles — thus the more corners there are in the images, the more detailed will be the computed free space. An alternative approach to free space computation involving the use of points to construct a polyhedral approximation to an object is described in [10]. The identification of obstacles directly from range data for map building in a navigation system is described in [21].

8.3. Path planning

Path planning involves the determination of a route to the target, passing only through areas which have been confirmed to be free space. Use of the midline, an affine construct, through an area of free space is fundamental to the adopted approach. The explanation below is given with reference to Figure 15 which shows actual free space maps computed during the processing (further examples are given in [4]).

Figure 15(a) is a schematic plan view of the environment, where there is no direct route from the initial camera position to the target because of the presence of obstacles $O1, O2, O3$. Figure 15(b) shows the free space map computed as described in Section 8.2 after several small lateral camera motions have been executed, and affine structure computed. The free space extends forward from the camera, is truncated at $O1$ and $O2$, but a central lobe extends through the gap between the obstacles. The midline of the lobe is computed, and the trajectory from the current camera position to a point on the midline, and then along the midline, is checked to see how far the camera can proceed.

Figure 15(c) shows the free space after the camera has moved through the gap between the obstacles O_1 and O_2. A lateral camera motion has been carried out at the new position, and affine structure computed for the newly visible parts of the scene. Newly detected free space has been used to incrementally enlarge the free space map, the new area being truncated by obstacle O_3 to the left and terminating at the obstacle to the rear of the scene. The midline of the new lobe of free space is computed and the camera proceeds

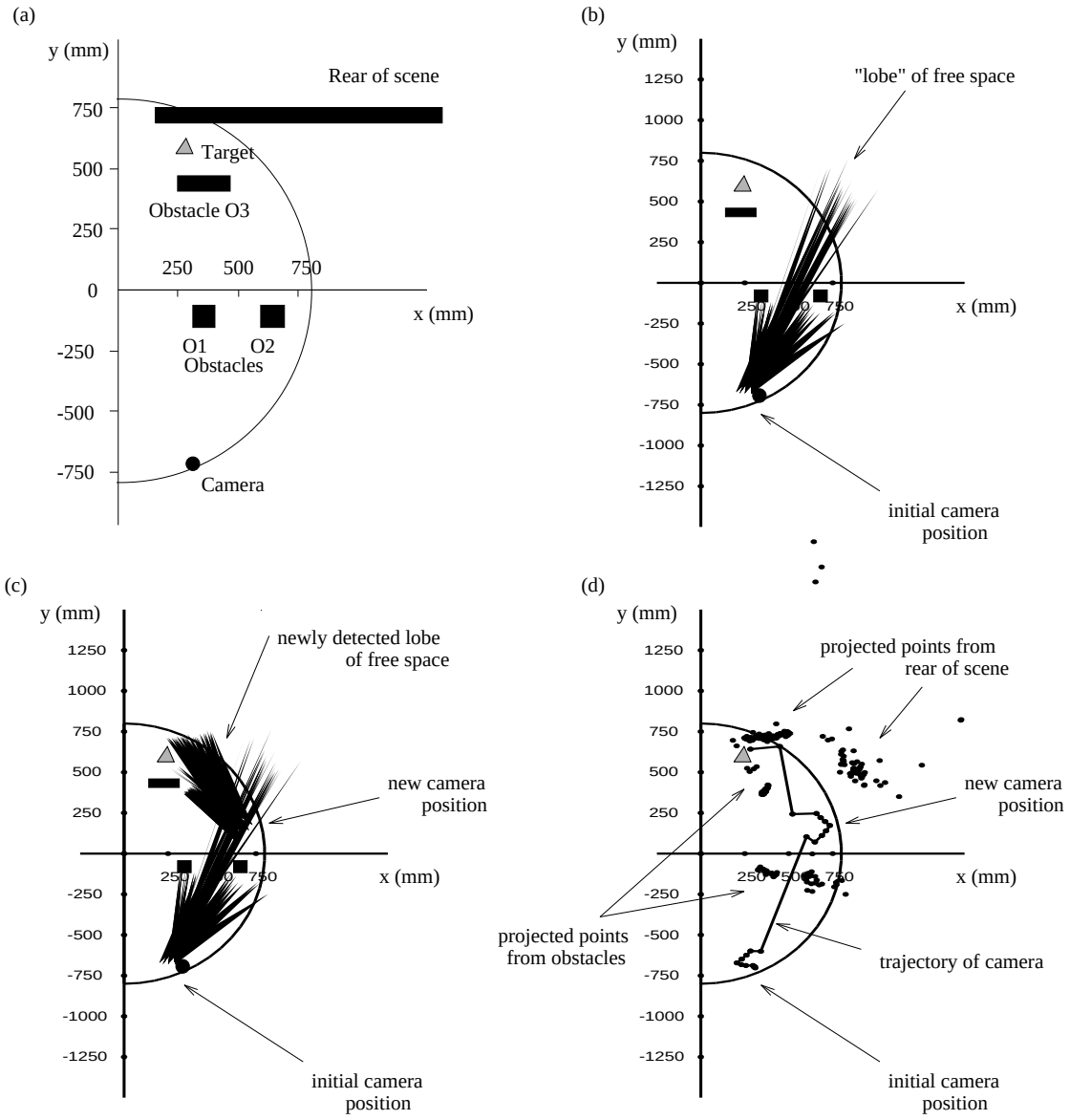


Fig. 15. (a) Schematic plan view of the experimental layout showing the initial camera position, obstacles $O1, O2, O3$, the target position, and the rear of the scene. The semi-circle indicates the workspace of the robot arm. (b) Free space (black) computed after the initial camera motions, extending forward from the camera which is at the lower part of the figure. The left and right-hand sides of the free space are terminated at about $y = -100$ where there are obstacles but the central "lobe" of free space extends through the gap between the obstacles to about $y = 350$. (c) Updated free space map after the robot has proceeded through the gap and rechecked the target. (d) Projection of 3D structure and camera positions (connected line) onto the ground plane. The full trajectory of the camera from the initial to the target position is shown.

to the target. Figure 15(d) shows the computed affine structure (isolated points) and the camera trajectory (connected line).

8.4. Computation of mappings

Two mappings which arise in the navigation processing are the 3D-2D projection of 3D structure to the ground plane in the affine frame, and a 2D-2D transformation between the ground plane in the affine frame and the ground plane in the Euclidean robot frame.

3D-2D projection to the ground plane. Projection of 3D point positions to the ground plane in the affine coordinate frame requires knowledge of the vertical direction. This is in fact readily available since the camera is mounted such that its y-axis is vertical, and the axes of the camera and the affine coordinate frames are aligned at initialisation (Section 7.1). Thus the vertical direction is aligned with the Y-axis of the affine frame, and a 3D point $\mathbf{X} = (X, Y, Z, 1)$ projects simply to $\mathbf{X}_P = (X, Z, 1)$ on the ground plane.

2D-2D transformation between affine and robot frames. A full 3D transformation between the affine coordinate frame and the robot coordinate frame is not needed, since motions are in a horizontal plane and information about height above the ground plane is not relevant. Thus it suffices to obtain a 2D transformation for the ground plane. The transformation can be found given the coordinates of three or more non-collinear points on the ground plane in the affine frame, and their corresponding position on the ground plane in the robot coordinate frame. Computation of the transformation utilises optical centre positions computed in the affine frame in the normal course of the SFM processing, with the corresponding positions in the robot frame being provided by the robot, and no special calibration is necessary. All computed camera positions are utilised in a least-squares linear computation.

The primary use of the transformation between the affine and robot frames is to enable motions in the affine frame to be mapped to Euclidean commands for the robot. This could potentially be avoided because the robot could be controlled by visual servoing alone. For example, with no calibration the robot could be driven to rotate until a certain point (for instance an affine invariant such as a centroid) was at the middle of the image. This has not been addressed since the focus of the work so far has been on the computation of 3D structure. In addition, the transformation has two further uses: first, the dimensions of the robot assembly which carries the camera are specified as Euclidean measurements; and, secondly, the target position for the robot motion is specified as a Euclidean position in the robot coordinate frame.

9. Conclusion

This paper has demonstrated the initialisation of projective and affine structure from image sequences, with an accuracy similar to a system using calibrated cameras. The work has been implemented in a system which has been extensively tested on real images, with automatic and reliable correspondence matching, and the use of robust techniques to detect outliers. The recovery of projective and affine structure is increasingly well-understood, but its use in practice raises interesting problems about what can be achieved when Euclidean measurements are not available. Here affine structure has been applied to path planning.

The possibility of utilising a constraint such as translational motion (Moons et al. (1994) to obtain affine structure underlies a spectrum of possibilities for investigation, ranging from fully calibrated stereo heads through to cameras of unknown intrinsic parameters and motion. The precision of the constraints and the stage at which they are introduced interplay to determine the type of the recovered structure and motion — projective, affine, or Euclidean — and its accuracy. This echoes the idea of stratification introduced by Koenderink and van Doorn [20].

Here we have concentrated on the uncalibrated end of this spectrum. There are many remaining questions concerning the constraints required, and the stage at which they are introduced, when specialising structure. For example, there are various ways to specialise projective structure to affine; by translation as demonstrated, or by identifying distant points [8]. Further specialisation to scaled Euclidean structure is possible by camera

self-calibration (Faugeras 1992; Hartley 1993) , or by other constraints on lengths and angles (Mohr et al. 1994). The interaction and application of such constraints offers numerous possibilities for extension of the ideas presented here.

Our contribution has been to provide a mechanism — via the Quasi-Euclidean frame — for incorporating poor or partial camera calibration. It has been demonstrated that in practice this approximate information is sufficient to obtain a reconstruction which is only slightly skewed away from metric structure.

Finally, the work has highlighted a number of issues which proved to be of key importance in experimental terms, although their importance was not always immediately evident in the mathematical theory.

Sensitivity to mismatches. The veracity of the fundamental matrix `F` and the perspective projection matrix `P` is severely affected by mismatched corners. It is crucial to remove mismatches since, even if they appear to be having only a small effect in individual computations, they cause a cumulative degradation over time as the structure is updated. As described in Section 5, we employ a three-stage process in which mismatches are identified as outliers in the computation of `F` and `P`.

Camera motion. The conditioning of the computation of `F` becomes poorer as the distance between the camera optical centres gets smaller. We have utilised large (rather than infinitesimal) motions between images of 1-3cm.

Wide-angle lens. Use of a wide angle lens (a field of view of about 50°) leads to better camera localisation because rays from the optical centre to the scene have good divergence; it also makes it easier to fix each new camera position in the ongoing coordinate frame because many points remain in view between images.

Forward motion. Simple forward motion produces poor structure because rays from the camera to a 3D point change angle slowly (relative to the effect of a lateral motion) resulting in large error in the computed point position. To avoid this, forward motion paths could be "dithered" with lateral movements (stereo would be of obvious benefit in this role). Instead, a 3D point is not initialised from its first two observations if the angle between the backprojected rays is below a threshold (2° estimated in the Quasi-Euclidean frame), but the observations are accumulated until the angle exceeds the threshold. Only then is initialisation carried out, using *all* the backprojected rays in a generalisation of the 2-ray scheme: the 3D point is found as that which minimises the sum of the square distances between the 3D point to each backprojected ray.

Critical surfaces. The problem of critical surfaces in structure from motion is well-known (see for example [25]). A special case of the general form of a critical surface arises in our environment when a planar surface fills (or nearly fills) the field of view. This suggests a need to explicitly test for critical surfaces and switch to alternative processing when detected.

Homogeneous coordinates. The arbitrary homogeneous component in a homogeneous vector is typically chosen as unity — e.g., an image corner (u, v) is represented as $(u, v, 1)$. Increased stability is achieved if the third component is chosen to be of the same order of magnitude as u and v (Section 3.3). Hartley describes an automatic procedure for achieving this initialisation (Hartley 1995). Similar remarks apply to points in 3D.

Corner matching. The cross-correlation used to measure strength of match between corners is initially on raw image intensity to avoid unnecessary computation. However, if the matching between a pair of images appears to be failing at any stage (which is tested by examining whether the ratio of number of matches to total number of corners is below a threshold), then the matching is restarted with cross-correlation on normalised intensity. The normalisation is effectively done by dividing the pixel patch at a corner by its mean intensity. The initiation of normalisation occurs for two reasons in practice: (i) changing illumination is an obvious effect which will cause matching

on raw image intensity to fail; and (ii) the automatic gain control of the camera may adjust the grey-level intensity across the whole image in response to some event such as a bright area appearing on the image periphery; again, this will prevent matching on raw intensity.

Acknowledgements

Acknowledgements

This work was supported by Grant GR/H77668 from the UK EPSRC, by Esprit Grant BRA 6448 'VIVA' from the EC, and by and the Newton Institute, Cambridge, under SERC Grant GR/G59981. The authors have profited from discussions with Richard Hartley, Jitendra Malik, Joe Mundy, Charlie Rothwell and with colleagues in the Robotics Research Group, particularly Andrew Blake, Mike Brady, Phil McLauchlan, Ian Reid, Larry Shapiro, Phil Torr and Bill Triggs. Adrian Cox provided considerable advice on operating the Adept robot arm.

Appendix A
Lemma 1

LEMMA 1 *Given two cameras with distinct optical centres and fundamental matrix* $\mathtt{F}$*, and the perspective projection matrix for the first camera in the canonical form* $\mathtt{P}_1 = [\mathtt{I}|\mathbf{0}]$*, then* $\mathtt{P}_2$ *has the general form*

$$\mathtt{P}_2 = [\mathtt{M}_2 + \mathbf{e}_2\mathbf{b}^\top | \, c\mathbf{e}_2]$$

where $\mathbf{e}_2$*, the epipole in the second image, satisfies* $\mathtt{F}^\top\mathbf{e}_2 = \mathbf{0}$*;* $\mathtt{M}_2$ *is a particular solution of* $\mathtt{F} = [\mathbf{e}_2]_\times\mathtt{M}_2$*; and* $\mathbf{b}$ *and* c *are an arbitrary 3-vector and scalar respectively.*

Proof:

This proof follows closely Hartley's proof of the uniqueness of decompositions of the fundamental matrix [14]. Suppose $\mathtt{P}_2 = [\mathtt{M}_2 | \, \mathbf{t}_2]$ and $\mathtt{P}'_2 = [\mathtt{M}'_2 | \, \mathbf{t}'_2]$ are two possible $\mathtt{P}_2$ matrices consistent with $\mathtt{F}$. Since $\mathtt{P}_1 = [\mathtt{I}|\mathbf{0}]$, the optical centre of the first camera has coordinates $\mathbf{Q}_1 = (0,0,0,1)^\top$. The epipole in the second image is the projection of the optical centre. Applying $\mathtt{P}_2$ and $\mathtt{P}'_2$ we have

$$\begin{aligned} \mathtt{P}_2\mathbf{Q}_1 &= \mathbf{t}_2 = \lambda\mathbf{e}_2 \\ \mathtt{P}'_2\mathbf{Q}_1 &= \mathbf{t}'_2 = \lambda'\mathbf{e}_2 \ . \end{aligned}$$

Thus $\mathbf{t}_2 = \mathbf{t}'_2 = \mathbf{e}_2$ up to a scale factor.

Next from equation (5) for the fundamental matrix defined from projection matrices, we have:

$$\mathtt{F} = [\mathbf{e}_2]_\times\mathtt{M}_2 = [\mathbf{e}_2]_\times\mathtt{M}'_2 \ .$$

It follows that $[\mathbf{e}_2]_\times(\mathtt{M}_2 - \mathtt{M}'_2) = \mathtt{0}$ and so $\mathtt{M}_2 - \mathtt{M}'_2 = \mathbf{e}_2\mathbf{b}^\top$.

Thus, including the overall scale factor, there are five homogeneous parameters or four DOF in $\mathtt{P}_2$. Values for $\mathtt{M}_2$ and $\mathbf{e}_2$ are obtained by decomposing $\mathtt{F}$ as described next. ■

A.1. Decomposing F

The first step is to obtain $\mathbf{e}_2$ from the equation $\mathtt{F}^\top\mathbf{e}_2 = \mathbf{0}$. If $\mathtt{F} = (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)$, where $\mathbf{f}_i$ are the columns of $\mathtt{F}$, then $\mathbf{f}_i \cdot \mathbf{e}_2 = 0$. Thus the epipole can be computed as $\mathbf{e}_2 = \mathbf{f}_1 \times \mathbf{f}_2$.

The second step is to compute a particular $\mathtt{M}_2$. Note, that there is not a unique decomposition since (from the proof of the lemma) if $\mathtt{N}$ is a particular solution, then so is $\mathtt{N} + \mathbf{e}_2\mathbf{d}^\top$ where $\mathbf{d}$ is an arbitrary 3-vector. The equation $\mathtt{M}_2 = [\mathbf{e}_2]_\times\mathtt{F}$ provides a particular solution. This can be verified by substitution: $\mathtt{F} = [\mathbf{e}_2]_\times\mathtt{M}_2 = [\mathbf{e}_2]_\times[\mathbf{e}_2]_\times\mathtt{F} = \lambda\mathtt{F}$ where λ is a scalar. This holds since for each column c_1, c_2, c_3 of $\mathtt{F}$, $\mathbf{e}_2 \times (\mathbf{e}_2 \times c_i) = \lambda c_i$ with λ the same scalar for each c_i.

Appendix B
Transformation between Euclidean and Quasi-Euclidean frames

We develop theory to quantify the residual projective skew in the Quasi-Euclidean frame.

Consider two cameras with the same intrinsic parameters $\mathtt{C}$, separated by a rotation $\mathtt{R}$ and translation $\mathbf{t}$. Then a Euclidean coordinate frame is obtained by setting the perspective projection matrices to

$$\begin{aligned} \mathtt{P}_1^E &= \mathtt{C}[\mathtt{I}|\mathbf{0}] = [\mathtt{C}|\mathbf{0}] \\ \mathtt{P}_2^E &= \mathtt{C}[\mathtt{R}| - \mathtt{R}\mathbf{t}] = [\mathtt{CR}| - \mathtt{CR}\mathbf{t}] = [\mathtt{CR}|\lambda\hat{\mathbf{e}}_2] \end{aligned}$$

where $\lambda = \| - \mathtt{CR}\mathbf{t}\|$, $\hat{\mathbf{e}}_2$ is the epipole in image 2 normalised so that the sum of its squared components is unity, and it has been made explicit that

the last column of $\mathtt{P}_2$ is a multiple of $\hat{\mathbf{e}}_2$. (For the purposes of explanation, we will keep $\mathtt{C}$ explicit here unlike the approach in §3.3).

Now consider a projective coordinate frame set up with

$$\begin{aligned}\mathtt{P}_1 &= [\mathtt{C}^*|\mathbf{0}]\\ \mathtt{P}_2 &= [\mathtt{C}^*\mathtt{R}_p^*|\mu\hat{\mathbf{e}}_2]\end{aligned}$$

where μ is the scale of the reconstruction.

We now show that the transformation $\mathbf{X} = \mathtt{H}^{-1}\mathbf{X}^E$ between the Euclidean and projective frames is

$$\mathtt{H} = \begin{bmatrix} \frac{\lambda}{\mu}\mathtt{C}^{-1}\mathtt{C}^* & \mathbf{0} \\ \mathbf{v}^\top & 1 \end{bmatrix} \qquad \text{(B1)}$$

where

$$\mathbf{v}^\top = \mu^{-1}\hat{\mathbf{e}}_2^\top(\mathtt{C}^*\mathtt{R}_p^* - \mathtt{C}\mathtt{R}\mathtt{C}^{-1}\mathtt{C}^*)\ . \qquad \text{(B2)}$$

Proof: Under the transformation $\mathbf{X} = \mathtt{H}^{-1}\mathbf{X}_E$, equation (7) shows the projection matrices transform as

$$\begin{aligned}\mathtt{P}_1^E\mathtt{H} &= \mathtt{P}_1 \qquad \text{(B3)}\\ \mathtt{P}_2^E\mathtt{H} &= \mathtt{P}_2\ . \qquad \text{(B4)}\end{aligned}$$

$\mathtt{H}$ can be written as

$$\mathtt{H} = \begin{bmatrix} \mathtt{A} & \mathbf{b} \\ \mathbf{c}^\top & d \end{bmatrix}$$

where $\mathtt{A}$ is a 3×3 matrix, $\mathbf{b}$ and $\mathbf{c}$ are 3-vectors, and d a scalar. From equation (B3) it follows that $\mathtt{A} = \mathtt{C}^{-1}\mathtt{C}^*$ and $\mathbf{b} = \mathbf{0}$ i.e.

$$\mathtt{H} = \begin{bmatrix} \mathtt{C}^{-1}\mathtt{C}^* & \mathbf{0} \\ \mathbf{c}^\top & d \end{bmatrix},$$

and from equation (B4)

$$[\mathtt{C}\mathtt{R}\mathtt{C}^{-1}\mathtt{C}^* + \lambda\hat{\mathbf{e}}_2\mathbf{c}^\top|d\lambda\mathbf{e}_2] = [\mathtt{C}^*\mathtt{R}_p^*|\mu\mathbf{e}_2]\ . \qquad \text{(B5)}$$

Pre-multiplying the left 3×3 matrices of equation (B5) by $\hat{\mathbf{e}}_2^\top$ gives

$$\mathbf{c}^\top = \lambda^{-1}\hat{\mathbf{e}}_2^\top(\mathtt{C}^*\mathtt{R}_p^* - \mathtt{C}\mathtt{R}\mathtt{C}^{-1}\mathtt{C}^*),$$

and the final column gives $d = \mu/\lambda$. An overall scaling gives the form of $\mathtt{H}$ in equations (B1) and (B2). ■

Appendix C

Lemma 2

LEMMA 2 *Given two camera matrices* $\mathtt{P}_1 = [\mathtt{I}|\mathbf{0}]$ *and* $\mathtt{P}_2 = [\mathtt{M}|\mathbf{t}^*]$ *for identical cameras related by a pure translation, matrix* $\mathtt{M}$ *can be decomposed as* $\mathtt{M} = \lambda\mathtt{I} + \mathbf{t}^*\mathbf{v}^\top$ *where* λ *is a scale factor, and* $\pi_\infty = (\mathbf{v}^\top, 1)^\top$ *is the equation of the plane at infinity.*

Proof:

The proof is based on the approach given in (Mundy & Zisserman 1994) . The cameras have identical intrinsic parameters, and their positions differ only by a pure translation. Thus, assuming image coordinates have been normalised according to equation (8), the projection matrices for a Euclidean frame are

$$\begin{aligned}\mathtt{P}_1^E &= [\mathtt{R}|-\mathtt{R}\mathbf{t}_1]\\ \mathtt{P}_2^E &= [\mathtt{R}|-\mathtt{R}\mathbf{t}_2]\end{aligned}$$

The Euclidean structure $\mathbf{X}^E$ and projective structure $\mathbf{X}$ are related by $\mathbf{X}^E = \mathtt{H}\mathbf{X}$, where

$$\mathtt{H} = \begin{bmatrix} \mathtt{A} & \mathbf{s} \\ \mathbf{v}^\top & 1 \end{bmatrix}\ .$$

The projection matrices in the projective frame are

$$\begin{aligned}\mathtt{P}_1 &= \mathtt{P}_1^E\mathtt{H}\\ \mathtt{P}_2 &= \mathtt{P}_2^E\mathtt{H}\ . \qquad \text{(C1)}\end{aligned}$$

The $\mathtt{P}_1$ equality gives

$$\lambda[\mathtt{I}|\mathbf{0}] = \mathtt{R}[\mathtt{A} - \mathbf{t}_1\mathbf{v}^\top|\mathbf{s} - \mathbf{t}_1]$$

where λ is an arbitrary scale factor. Hence,

$$\begin{aligned}\mathbf{s} &= \mathbf{t}_1\\ \mathtt{R}\mathtt{A} &= \lambda\mathtt{I} + \mathtt{R}\mathbf{t}_1\mathbf{v}^\top\ .\end{aligned}$$

From the $\mathtt{P}_2$ equality (equation (C1)), if follows that

$$\begin{aligned}\mathbf{t}^* &= \mathtt{R}[\mathbf{t}_1 - \mathbf{t}_2]\\ \mathtt{M} &= \lambda\mathtt{I} + \mathbf{t}^*\mathbf{v}^\top\ .\end{aligned}$$

This completes the first part of the proof. It only remains to demonstrate that $\pi_\infty = (\mathbf{v}^\top, 1)^\top$

is the equation of the plane at infinity. The point transformation matrix is $\mathbf{X} = \mathtt{H}^{-1}\mathbf{X}^E$, hence the plane transformation matrix is $\mathtt{H}^\top$ [35]. The coordinates of the plane at infinity in the Euclidean frame are $\pi_\infty^E = (0,0,0,1)^\top$. Therefore the coordinates in the projective frame are

$$\begin{aligned}\pi_\infty &= \mathtt{H}^\top \pi_\infty^E \\ &= (\mathbf{v}^\top, 1)^\top\end{aligned}$$

■

C.1. C.1 Solving for λ and $\mathbf{v}$

Given $\mathtt{M}$ and $\mathbf{t}^*$ we now describe how to obtain λ and $\mathbf{v}$ from

$$\mathtt{M} - \lambda \mathtt{I} = \mathbf{t}^* \mathbf{v}^\top \ .$$

This is an eigenvector problem: the matrix $\mathbf{t}^*\mathbf{v}^\top$ is rank one, so λ must be a repeated eigenvalue of $\mathtt{M}$. Call this eigenvalue λ_1, associated with eigenvectors $\mathbf{e}_a$ and $\mathbf{e}_b$, and the remaining eigenvalue λ_2 associated with eigenvector $\mathbf{e}_c$. Then

$$\begin{aligned}(\mathtt{M} - \lambda_1 \mathtt{I})\mathbf{e}_c &= (\lambda_2 - \lambda_1)\mathbf{e}_c \qquad \text{(C2)} \\ &= (\mathbf{t}^*\mathbf{v}^\top)\mathbf{e}_c = \mathbf{t}^*(\mathbf{v}\cdot\mathbf{e}_c)\end{aligned}$$

Hence $\mathbf{e}_c$ is parallel to $\mathbf{t}^*$. For the other eigenvectors $(i = a, b)$

$$(\mathtt{M} - \lambda_1 \mathtt{I})\mathbf{e}_i = \mathbf{t}^*\mathbf{v}^\top\mathbf{e}_i = \mathbf{t}^*(\mathbf{v}\cdot\mathbf{e}_i) = \mathbf{0} \ .$$

Hence $\mathbf{e}_a, \mathbf{e}_b$ are both perpendicular to $\mathbf{v}$, and therefore

$$\mathbf{v} = \mu \mathbf{e}_a \times \mathbf{e}_b \qquad \text{(C3)}$$

where μ is an unknown scale. This scale μ is determined from equation (C2) as follows:

$$\mathbf{v}\cdot\mathbf{e}_c = \frac{(\lambda_2 - \lambda_1)\|\mathbf{e}_c\|^2}{\mathbf{t}^*\cdot\mathbf{e}_c}$$

and, taking the scalar product with $\mathbf{v}$ in equation (C3),

$$\mu = \frac{(\lambda_2 - \lambda_1)\|\mathbf{e}_c\|^2}{(\mathbf{t}^*\cdot\mathbf{e})_c[\mathbf{e}_c, \mathbf{e}_a, \mathbf{e}_b]} \ .$$

References

1. N. Ayache. 1991. *Artificial vision for mobile robots.* MIT Press, Cambridge, 1991.
2. M. Armstrong, A. Zisserman, and P.A. Beardsley. 1994. Euclidean reconstruction from uncalibrated images. *Proc. British Machine Vision Conference*, 1994.
3. Y. Bar-Shalom and T.E. Fortmann. 1988. *Tracking and Data Association.* Academic Press, 1988.
4. P.A. Beardsley, A.P. Zisserman, and D.W. Murray. 1994. Navigation using affine structure and motion. In *Proc. 3rd European Conference on Computer Vision*, pages 85–96. Springer-Verlag, 1994.
5. A. Blake, M. Brady, R. Cipolla, Z. Xie, and A.R. Zisserman. 1991. Visual navigation around curved objects. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2490–2495, 1991.
6. A. Blake, A. Zisserman, and R. Cipolla. 1992. Visual exploration of free-space. In Blake and Yuille, editors, *Active Vision.* MIT Press, 1992.
7. R. Deriche, Z. Zhang, Q.T. Luong, and O. Faugeras. 1994. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In *Proc. 3rd European Conference on Computer Vision*, pages 567–576. Springer-Verlag, 1994.
8. O.D. Faugeras. 1992. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. 2nd European Conference on Computer Vision*, pages 563–578. Springer-Verlag, 1992.
9. O.D. Faugeras, Q.T. Luong, and S.J. Maybank. 1992. Camera self-calibration: theory and experiments. In *Proc. 2nd European Conference on Computer Vision*, pages 321–334. Springer-Verlag, 1992.
10. O.D. Faugeras. 1993. *Three-dimensional computer vision: a geometric viewpoint.* MIT Press, 1993.
11. C.G. Harris. 1987. Determination of ego-motion from matched points. In *Third Alvey Vision Conference*, pages 189–192, 1987.
12. C.G. Harris and J.M. Pike. 1987. 3D positional integration from image sequences. In *Third Alvey Vision Conference*, pages 233–236, 1987.
13. C.G. Harris and M. Stephens. 1988. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, 1988.
14. R. Hartley. 1992. Invariants of points seen in multiple images. GE internal report, to appear in PAMI, GE CRD, Schenectady, NY 12301, USA, 1992.
15. R.I. Hartley, R. Gupta, and T. Chang. 1992. Stereo from uncalibrated cameras. *Proc. Conference Computer Vision and Pattern Recognition*, 1992.
16. R.I. Hartley. 1994. Euclidean reconstruction from uncalibrated views. In J.L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of invariance in computer vision*, pages 237–256. Springer-Verlag, 1994.
17. R.I. Hartley. 1995. In defence of the 8-point algorithm. In E. Grimson, editor, *Proc. 5th International Conference on Computer Vision*, Cambridge, MA, June 1995.
18. R.I. Hartley and P. Sturm. 1995. Triangulation. In *Proc. Conf. Computer Analysis of Images and Patterns, Prague, Czech Republic*, 1995.

19. N. Hollinghurst and R. Cipolla. 1993. Uncalibrated stereo hand-eye coordination. In *Proc. British Machine Vision Conference 93*, pages 389–398, 1993.
20. J.J. Koenderink and A.J. VanDoorn. 1991. Affine structure from motion. *J. Opt. Soc. Am. A*, 8(2):377–385, 1991.
21. D. Langer, J.K. Rosenblatt, and M. Hebert. 1994. An integrated system for autonomous off-road navigation. In *Proc. IEEE Conf. Robotics and Automation*, pages 414–419. IEEE, 1994.
22. J.C. Latombe. 1991. *Robot motion planning.* Kluwer Academic Publishers, 1991.
23. Q.T. Luong, R. Deriche, O. Faugeras, and T. Papadopoulo. 1993. On determining the fundamental matrix. Technical report 1894, INRIA, Sophia-Antipolis, France, 1993.
24. Q.T. Luong and T. Vieville. 1994. Canonic representations for the geometries of multiple projective views. In *Proc. 3rd European Conference on Computer Vision*, pages 589–597. Springer-Verlag, 1994.
25. S.J. Maybank. 1993. *Theory of reconstruction from image motion.* Springer-Verlag, Berlin, 1993.
26. P.F. McLauchlan, I.D. Reid, and D.W. Murray. 1994. Recursive affine structure and motion from image sequences. In *Proc. 3rd European Conference on Computer Vision*, pages 217–224. Springer-Verlag, 1994.
27. R. Mohr, F. Veillon, and L. Quan. 1993. Relative 3D reconstruction using multiple uncalibrated images. *Proc. Conference Computer Vision and Pattern Recognition*, pages 543–548, 1993.
28. R. Mohr, B. Boufama, and P. Brand. 1994. Accurate projective reconstruction. In J.L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of invariance in computer vision*, pages 257–276. Springer-Verlag, 1994.
29. T. Moons, L. van Gool, M. van Diest, and A. Oosterlinck. 1994. Affine structure from perspective image pairs obtained by a translating camera. In J.L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of invariance in computer vision*, pages 297–316. Springer-Verlag, 1994.
30. J.L. Mundy and A.P. Zisserman. 1992. *Geometric invariance in computer vision.* MIT Press, 1992.
31. J.L. Mundy and A. Zisserman. 1994. Repeated structures: Image correspondence constraints and ambiguity of 3D reconstruction. In J.L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of invariance in computer vision*, pages 89–106. Springer-Verlag, 1994.
32. W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. 1988. *Numerical Recipes in C.* Cambridge University Press, 1988.
33. I. D. Reid and D. W. Murray. 1993. Tracking foveated corner clusters using affine structure. In *Proc. 4th International Conference on Computer Vision*, pages 76–83, Los Alamitos, CA, 1993. IEEE Computer Society Press.
34. C.A. Rothwell, G. Csurka, and O. Faugeras. 1995. A comparison of projective reconstruction methods for pairs of views. In E. Grimson, editor, *Proc. 5th International Conference on Computer Vision*, Cambridge, MA, June 1995.
35. J.G. Semple and G.T. Kneebone. 1952. *Algebraic projective geometry.* Oxford University Press, 1952.
36. R. Szeliski and S.B. Kang. 1993. Recovering 3D shape and motion from image streams using non-linear least squares. DEC technical report 93/3, DEC, 1993.
37. P.H.S. Torr, P.A. Beardsley, and D.W. Murray. 1994. Robust vision. In *Proc. British Machine Vision Conference 94*, 1994.
38. P.H.S. Torr. . *Motion segmentation and outlier detection.* PhD thesis, Dept. of Engineering Science, University of Oxford, 1995.
39. Z. Zhang and O. Faugeras. 1995. *3D Dynamic Scene Analysis.* Springer-Verlag, 1992.