Image Mosaicing using Sequential Bundle Adjustment

Philip F. McLauchlan and Allan Jaenicke[1],
School of EE, IT and Mathematics,
University of Surrey,
Guildford GU2 5XH.
Email P.McLauchlan@eim.surrey.ac.uk,
A.Jaenicke@viswiz.com

Abstract

We describe the construction of accurate panoramic mosaics from multiple images taken with a rotating camera, or alternatively of a planar scene. The novelty of the approach lies in *(i)* the transfer of photogrammetric bundle adjustment techniques to mosaicing; *(ii)* a new representation of image line measurements enabling the use of lines in camera self-calibration, including computation of the radial and other non-linear distortion; and *(iii)* the application of the Variable State Dimension Filter (VSDF) to obtain efficient sequential updates of the mosaic as each image is added.

We demonstrate that our method achieves better results than the alternative approach of optimising over pairs of images.

1 Introduction

Mosaicing is a common and popular method of effectively increasing the field of view of a camera, by allowing several views of a scene to be combined into a single view. To use the technique one must assume that either the camera is rotating in a stationary scene, or that the scene is approximately planar. In these cases there is no parallax induced by the motion, and so no truly 3D effects are involved, simplifying the task of combining views. Existing methods are based on either image-to-image warping [4, 9, 8], or corner feature location and matching [2]. There are also commercial mosaicing systems such as the RealViz "Stitcher" software. In all cases the aim is to recover the *homographies* that map images to each other, and thence allow the images to be transformed and combined in a single coordinate system. When seamless, high-resolution mosaics are required it makes sense to consider the techniques that are used in full 3D structure-from-motion and photogrammetry, which have been developed over many years for high-accuracy metrology. That is the main goal of this paper.

The most relevant technique is *bundle adjustment* [1], which is a photogrammetric technique to combine multiple images of the same scene into an accurate 3D reconstruction. Initial estimates of the 3D location of features in the scene must first be computed, as well as estimates of the camera locations. Then bundle adjustment applies an iterative algorithm to compute optimal values for the 3D reconstruction of the scene and camera positions, by minimising the log-likelihood of the overall feature projection errors using a least-squares algorithm.

[1] Now with Vision Wizards Ltd.

BMVC 2000 doi:10.5244/C.14.62

It is quite straightforward to modify the usual bundle adjustment algorithms for image mosaicing. Consider the advantages of this procedure:–

- The result is statistically optimal given the usual assumptions concerning the projection errors (unbiased, Gaussian, known variance).
- All geometric constraints among the features are enforced. This is most important for dense mosaics where each part of the scene is viewed in several (i.e. more than two) images, as we shall demonstrate in our results.
- Self-calibration is a common feature of bundle adjustment algorithms, and different assumptions (e.g. fixed intrinsic parameters/varying focal length/varying focal length & zoom) may be incorporated without difficulty.
- Features other than points may be incorporated, for instance lines and curves.
- Since 2D bundle adjustment is basically a reduction of existing methods from 3D to 2D, it is fairly easy to implement given existing 3D reconstruction tools.

In order to apply bundle adjustment techniques there must be a clear distinction between the "image" and "scene" representations, so that a single scene reconstruction can be computed from multiple images. In order to use image-based techniques one would have to define an "abstract" image, for instance in spherical/cylindrical coordinates, to be computed from multiple projections. The "parameters" of this abstracted image would be the pixel values at each point in the coordinate space, in other words the mosaic itself. There is no doubt that such a method could produce good results, but computationally it would be prohibitive because of the size of the parameter space.

We employ the Variable State Dimension Filter (VSDF) [6, 5] to implement sequential bundle adjustment. The VSDF supports standard sparse matrix techniques that are used to accelerate bundle adjustment iterations. The sequential feature of the VSDF is important for two reasons: firstly it allows long sequences to be registered in a reasonable length of time, and secondly, when using the user interface to build the mosaic, the results can be viewed as each image is processed and added to the mosaic. This greatly enhances the user-friendliness of the software as well as aiding interactive control, such as manual error correction.

Another novel aspect of our work is the use of lines in self-calibration, in particular calibration of the non-linear distortion parameters, for which in the past points have always been used. To accomplish this requires a new model for the projection of straight lines into images, which we discuss in detail in section 3. The method can be generalised to self-calibration under general 3D viewing conditions.

2 Projection Model

We consider first the case of a rotating camera. We represent the scene as fixed features in a scene-centred coordinate frame, based on which the images provide observations. Thus a point in the scene is represented by a direction, which we implement as a unit vector $\mathbf{X} = (X\ Y\ Z)^\top$. Then the projection of point $\mathbf{X}$ onto the image plane x, y is in homogeneous coordinates

$$\begin{aligned} \mathbf{p} &= K\mathbf{d}(R\mathbf{X}) \qquad (1) \\ \text{or} \quad \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} &= \begin{pmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{d} \left(\begin{pmatrix} R_{XX} & R_{XY} & R_{XZ} \\ R_{YX} & R_{YY} & R_{YZ} \\ R_{ZX} & R_{ZY} & R_{ZZ} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \right) \end{aligned}$$

where

- $\mathbf{p}$ contains the homogeneous image coordinates of the projected point, so that $x = p_x/p_z$, $y = p_y/p_z$.
- K is a calibration matrix containing the camera focal lengths f_x, f_y and the image centres x_0, y_0:
- R is a 3×3 rotation matrix describing the rotation of the camera.
- $\mathbf{X}$ is the scene point and $\|\mathbf{X}\|^2 = 1$.
- $\mathbf{d}(.)$ is a distortion function describing the non-linear image distortion, which we represent as two terms of radial distortion:

$$\mathbf{d}(\mathbf{X}) = \begin{pmatrix} (1 + K_1 r^2 + K_2 r^4) \begin{pmatrix} X \\ Y \end{pmatrix} \\ Z \end{pmatrix}$$

  where $r^2 = \frac{X^2+Y^2}{Z^2}$ and K_1, K_2 are the radial distortion parameters.

The camera rotation R is represented using the local rotation approach, employed in 3D reconstruction by Taylor & Kriegman [11]. To reconstruct the mosaic, considering point features only at this stage, the initial registration provides the bundle adjustment algorithm with estimates of the camera motion $R_{(j)}$ for each image j. The feature detection and matching stages then provide observations x, y for each feature i in images j, which are used along with the initial $R_{(j)}$ to compute initial estimates of the scene point $\mathbf{X}_i$. The bundle adjustment then adjusts the estimates of motion $R_{(j)}$, scene structure $\mathbf{X}_i$ and (optionally) calibration parameters K and K_1, K_2.

2.1 Projective Model for Points

In the case of reconstructing a plane, or when the calibration is not known even approximately, or finally that non-linear distortions are either neglected or compensated for in the image, we can apply the "uncalibrated" projective method that reconstructs the mosaic up to a projective transformation. The point projection equation is then

$$\mathbf{p} = P\mathbf{X} \tag{2}$$

The matrix P is a 3×3 matrix defining a homography from projective 2D scene coordinates into projective 2D image coordinates. It is clear that the $P_{(j)}$'s and $\mathbf{X}_i$'s can be recovered up to a general homographic ambiguity. The scale freedoms in each $P_{(j)}$ are removed by imposing the constraint $\|P_{(j)}\|_F^2 = 1$ where $\|.\|_F$ denotes the Frobenius matrix norm.

3 Line Features

We employ an infinite line representation, which similarly to points can be represented as a unit 3-vector $\mathbf{L}$ such that $\mathbf{L}.\mathbf{X} = 0$ and $\|\mathbf{L}\|^2 = 1$. Under a linear camera model (i.e. no radial distortion), this would project to the line $\mathbf{l}.\mathbf{p} = 0$ where

$$\mathbf{l} = K^{-\top} R\mathbf{L} \tag{3}$$

Equation (3) would then define the line projection model, just as equation (1) describes the point projection model. With non-linear distortion, however, the scene line projects to a curve, forming the well-known "barrel" or "pin-cushion" effects in the image. This complication, a qualitative change in the projection equation for lines, has prevented the direct use of lines in non-linear self-calibration methods up to now. However we can circumvent this problem by modifying the line observation model. We build observations of lines by going back to each edge *point* that was used to fit the line. One can construct a measurement for each edge point $\mathbf{p}$ by considering the corresponding scene point $\mathbf{X}$ to lie on the scene line $\mathbf{L}$. For each edge observation we introduce an angle parameter θ that describes where on the line $\mathbf{L}$ the point $\mathbf{X}$ lies. Then the edge point observation is written as an observation depending on both $\mathbf{L}$ and θ. In this way line reconstruction can proceed in a similar manner to point reconstruction.

We must first define a reference zero for the angle θ. We do this by constructing another line $\mathbf{L}_0$ perpendicular to $\mathbf{L}$. We then define θ by the equation

$$\mathbf{X} = \cos\theta(\mathbf{L} \times \mathbf{L}_0) + \sin\theta[(\mathbf{L} \times \mathbf{L}_0) \times \mathbf{L}] \tag{4}$$

relating $\mathbf{L}$, $\mathbf{L}_0$ and θ to the point on the line $\mathbf{X}$. With this construction we have automatically $\mathbf{L}.\mathbf{X} = 0$. The final line projection model is obtained by combining equations (1) and (4). This results in a projection equation relating the line $\mathbf{L}$, angle θ, calibration parameters K, K_1, K_2 and rotation R to the edge point image position $\mathbf{p}$. This model is unusual because a θ parameter is attached to each edge point observation. It might be thought that this would considerably increase the computation involved in updating the reconstruction, but in fact sparse matrix techniques can be applied and not much extra computation is involved. It is impractical to build an observation for *every* edge point, and so instead a set of "representative" points are chosen. In the results we show later we used just the line endpoints.

3.1 Projective Model for Lines

The projective line projection equation is

$$\mathbf{l} = P^{-\top}\mathbf{L} \tag{5}$$

Again the scene line $\mathbf{L}$ is normalised to $||\mathbf{L}||^2 = 1$. Without the non-linear distortion there is no need to introduce the representative points on the image lines in order to relate scene to image; the image line parameters $\mathbf{l}$ may be used directly, first reducing them to a non-redundant representation in a coordinate-frame independent manner (details omitted).

4 Feature Matching

Given an initial approximate rotation $R_{(k)}$ of a new image k and existing scene, the feature matching attempts to match the point and line features in the new image to the existing point and line scene vectors $\mathbf{X}_i$ and $\mathbf{L}_i$. The main assumption is that the solution involves a *small* modification to the latest rotation $R_{(k)}$ or homography $P_{(k)}$. The first stage therefore involves a search around each projected scene point $\mathbf{p} = K\mathbf{d}(R_{(k)}\mathbf{X})$ and projected line $\mathbf{l} = K^{-\top}R\mathbf{L}$ for potential matching image features in the new image k (or $\mathbf{p} = P\mathbf{X}$ and $\mathbf{l} = P^{-\top}\mathbf{L}$ for the projective model). In the case of lines we first remove the non-linear distortion from the line endpoints using the latest distortion parameter estimate. Once candidate matches have been computed, RANSAC is used to find a rotation $R_{(k)}$ or homography $P_{(k)}$ consistent with a large number of scene feature/image feature matches.

5 Bundle Adjustment

Bundle adjustment is an iterative Gauss-Newton method, implementing non-linear least squares, computing the mean of the likelihood or posterior distribution (depending on whether prior knowledge is present or not), and taking advantage of sparseness in the system information matrix to speed up its inversion. The VSDF actually uses the Levenberg-Marquardt algorithm, which is a modification of Gauss-Newton. To apply the VSDF method we have to define the vector of state parameters $\mathbf{x}$, and an observation model which relates the state parameters to the image observations. The state $\mathbf{x}$ will be constructed from all the unknowns in the system: the n_p scene points $\mathbf{X}_i$ and n_l lines $\mathbf{L}_i$, the camera rotations $R_{(j)}$, the calibration parameters K, K_1, K_2, and the line observation angle parameters $\theta_{i(j)}$. These are all bundled together into $\mathbf{x}$ as follows:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_z \\ \mathbf{x}_l \\ \mathbf{x}_d \\ \mathbf{x}_f \end{pmatrix}, \quad \text{where} \quad \mathbf{x}_z = \begin{pmatrix} \theta_{1(1)} \\ \vdots \\ \theta_{1(k)} \\ \vdots \\ \vdots \\ \theta_{n(k)} \end{pmatrix}, \; \mathbf{x}_l = \begin{pmatrix} \mathbf{L}_1 \\ \vdots \\ \mathbf{L}_{n_l} \\ \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_{n_p} \end{pmatrix}, \; \mathbf{x}_d = \begin{pmatrix} \mathbf{r}_{(1)} \\ \vdots \\ \vdots \\ \mathbf{r}_{(k)} \end{pmatrix}, \; \mathbf{x}_f = \begin{pmatrix} f_x \\ f_y \\ x_0 \\ y_0 \\ K_1 \\ K_2 \end{pmatrix}.$$

where

- The "observation" parameters $\mathbf{x}_z$ are attached to individual observations, in this case the angles $\theta_{i(j)}$.
- The "local" parameters $\mathbf{x}_l$ are the scene structure parameters, in this case 2D points $\mathbf{X}_i$ and lines $\mathbf{L}_i$ in homogeneous coordinates.
- The "dynamic" parameters $\mathbf{x}_d$ represent the camera motion over time. Each 3-vector $\mathbf{r}_{(j)}$ represents the local rotation parameters for image j.
- The "fixed" parameters $\mathbf{x}_f$ represent the camera calibration. Here we assume that the calibration is the same for each image; if not then $\mathbf{x}_f$ would consist of multiple sets of calibration parameters.

We now turn to the observation model, which defines the way in which the state parameters relate to the image observations. The VSDF software requires that the projection equations 1, 2 and 5 be translated into routines to evaluate the image point/line coordinates given the relevant parts of the state vector $\mathbf{x}$ for each observation. This projected feature is then compared with the actual (i.e. matched) image feature, the difference or "innovation" being used to update the state vector. The routines should also evaluate the Jacobians of the projection with respect to $\mathbf{x}$ for use by the Gauss-Newton iterations. Other routines are used to set up the initial state vector values given an initial "batch" of image observations along with any prior knowledge, for instance of calibration, and also to initialise new scene features and camera rotations/homographies when in VSDF sequential mode. Details of the VSDF algorithm are given in [5].

The VSDF uses the above routines to build the linear system used in Levenberg-Marquardt iterations. The linear system matrix turns out to be sparse, and this is exploited to compute an efficient solution using the recursive partitioning algorithm [10]. In fact the computational complexity for this problem is theoretically $\mathcal{O}(k^2(k + n_p + n_l))$, i.e. proportional to the number of features n_p and n_l.

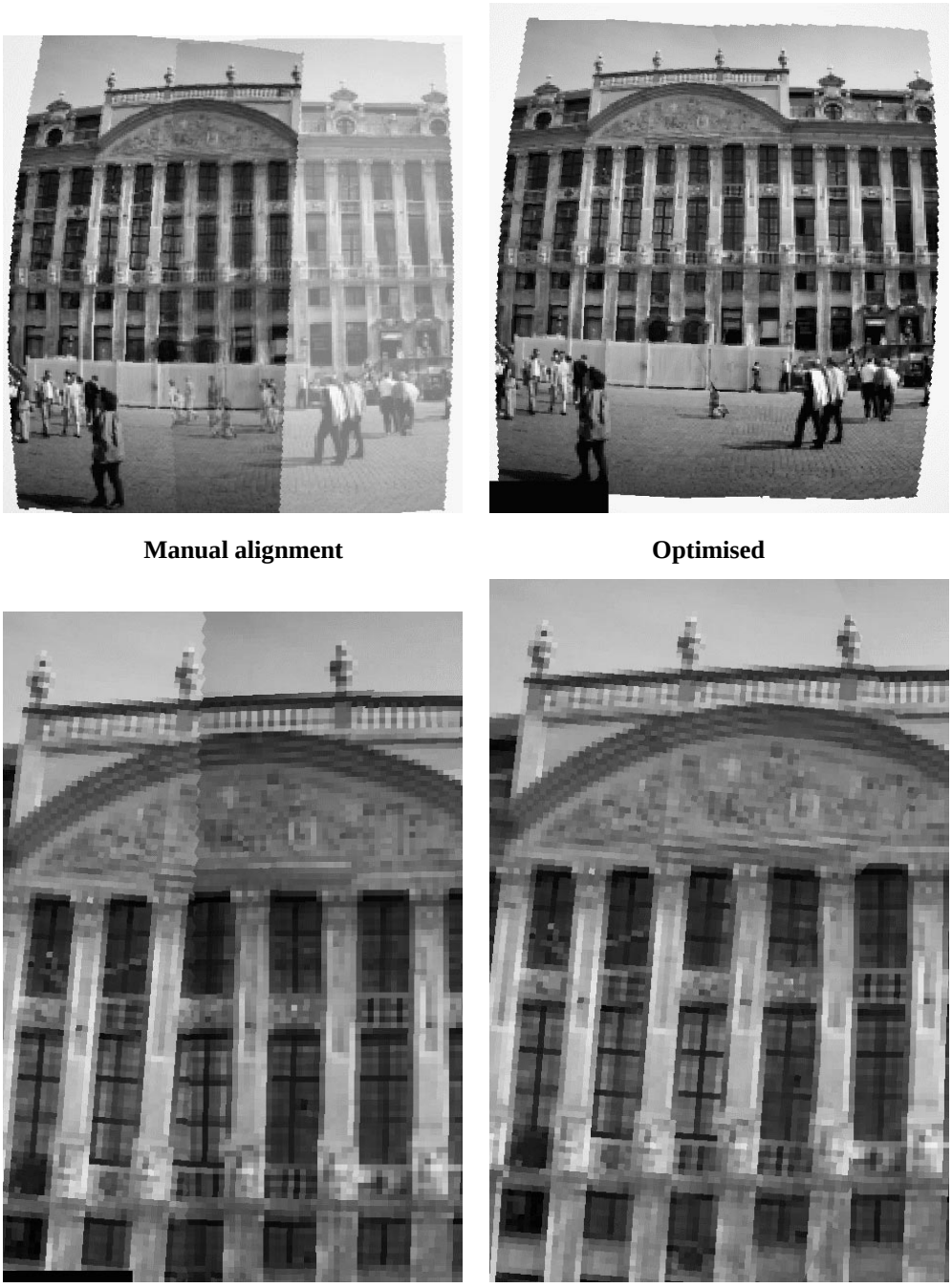
Figure 1: Five images of Market Square in Brussels, with corner and line image features.

In the case of uncalibrated projective reconstruction, there are no angles $\theta_{i(j)}$ and so no observation state vector $\mathbf{x}_z$. Also there are no calibration parameters, so $\mathbf{x}_f$ is also absent. The local state vector $\mathbf{x}_l$ is made up of the scene points $\mathbf{X}_i$ and lines $\mathbf{L}_i$, and the dynamic state vector contains the elements of the $P_{(j)}$ matrices.

6 Results

We present here results for mosaic reconstruction from point and line features for two image sequences, one of separate pictures taken with a still camera, the other a long video sequence. Five overlapping images from Market Square in Brussels are shown in figure 1. To locate point features we use the Plessey corner detector [3], and for lines edge detection followed by Hough transform line fitting [7]. We build up the mosaic image by image, registering the images approximately by hand, and then matching features and optimising using the software. Figure 2 illustrates the process for the first two images. The result of stitching the five images together is shown in figure 3.

While these results are good, other systems can also perform well on such images. More challenging is the long video sequence of which we show a sample in figure 4. Here the danger is that for a long sequence of closely spaced images, the registration error will build up as the sequence is processed. Any system based on exploiting pairwise overlaps between images would have difficulties here, because to maintain good registration overlaps between non-adjacent images have to be utilised, and it is not at all clear how to select the image pairs for this purpose. The sequence is especially difficult because the camera initially pans left, but at around frame 110 it backtracks to the right, finishing beyond the initial position. Thus correct registration involves matching images between the start and end of the sequence. Our system circumvents this problem by constraining image features not to each other but to abstracted scene point/line features $\mathbf{X}$/$\mathbf{L}$, a procedure which guarantees, just as in the case of 3D reconstruction, that all available geometric constraints are exploited. In figure 5a we show the results of stitching 216 images together, using a linear (pinhole) camera model. The non-linear image distortion is clear from the marked curvature of the tennis court baseline. Indeed registration breaks down slightly in some areas of the tennis court. This is mainly because most features are detected among the spectators, so that the registration is best there. When the distortion is modelled, the result is much better, both in removing the curvature and registering all



Manual alignment **Optimised**

Figure 2: The first two images from the Market Square sequence stitched together. On the left is the manual alignment, on the right the optimised registration. The bottom pair shows closeups of the join between the two images in the two cases.

Figure 3: The five images from the Market Square sequence stitched together. The visible image boundaries are not registration errors, but brightness variations between and within images, which we do not currently model. Indeed the centre of each image is significantly brighter than the periphery.

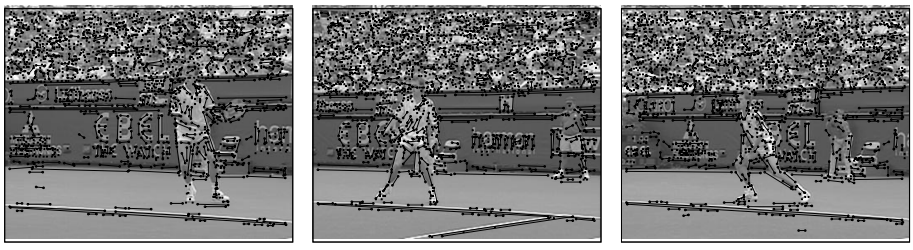


Image 1 **Image 40** **Image 80**

Image 120 **Image 160** **Image 200**

Figure 4: Six images from a video sequence of Stefan Edberg. The images were digitised at frame rate, so the the 216 images of the sequence represent about eight seconds of real-time video.

a

b

c

Figure 5: The Stefan sequence stitched together **(a)** using a linear (pinhole) camera model; **(b)** incorporating non-linear radial distortion parameters K_1 and K_2, and **(c)** using the accumulated homographies of image pairs (linear camera model). The total processing time is each case was about three hours on an Ultra-Sparc.

images correctly, as can be seen in figure 5b. The distortion parameters are recovered by the sequential VSDF algorithm, initialising K_1 and K_2 to zero. Note that Stefan himself has disappeared. Such a clean separation of a moving foreground object is only possible with a densely overlapping sequence. Finally figure 5c shows what happens when we only match features over adjacent image pairs. The RANSAC matching algorithm is unchanged; only the manner in which the matched data is employed is simplified. As expected, the errors in the homographies accumulate over the sequence, to the detriment of the registration.

7 Conclusions

We have designed a sequential semi-automatic mosaicing system for building high-accuracy mosaics from multiple images with a rotating camera, or images of a planar scene. The system is completely automatic when processing a sequence of closely spaced images such as from video. The system is flexible in that it can compute either a calibrated (optionally with self-calibration) or uncalibrated mosaic. It has performed well on an image sequence that would be difficult to register accurately using alternative methods.

8 Acknowledgements

Allan Jaenicke was funded by a Nuffield Foundation undergraduate research bursary.

References

[1] K.B. Atkinson. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, Caithness, Scotland, 1996.

[2] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 885–891, June 1998.

[3] C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf., Manchester*, pages 147–151, 1988.

[4] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proc. 5th Int'l Conf. on Computer Vision, Boston*, pages 605–611, 1995.

[5] P. F. McLauchlan. The variable state dimension filter. Technical Report VSSP 4/99, University of Surrey, Dept of Electrical Engineering, December 1999.

[6] P.F. McLauchlan and D.W. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proc. 5th Int'l Conf. on Computer Vision, Boston*, pages 314–320, June 1995.

[7] J. Kittler P. L. Palmer and M. Petrou. An optimising line finder using a hough transform algorithm. *CVGIP: Image Understanding*, 67(1):1–23, 1997.

[8] H. S. Sawnhey and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 450–456, 1997.

[9] H.Y. Shum and R. Szeliski. Panoramic image mosaics. Technical Report MSR-TR-97-23, Microsoft, 1997.

[10] C.C. Slama, C. Theurer, and S.W. Henriksen, editors. *Manual of Photogrammetry*. American Society of Photogrammetry, 1980.

[11] C.J. Taylor and D.J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1021–1032, November 1995.