AUTOMATIC CORRESPONDENCES FOR PHOTOGRAMMETRIC MODEL BUILDING

Dr. Gerhard Roth
Computational Video Group, Institute for Information Technology
National Research Council of Canada
Gerhard.Roth@nrc.ca

Working Group TS ICWG V/III

KEY WORDS: Photogrammetry, Modelling, Matching, Automation, Feature, Semi-Automation.

ABSTRACT

The problem of building geometric models has been a central application in photogrammetry. Our goal is to partially automate this process by finding the features necessary for computing the exterior orientation. This is done by robustly computing the fundamental matrix, and trilinear tensor for all images pairs and some image triples. The correspondences computed from this process are chained together and sent to a commercial bundle adjustment program to find the exterior camera parameters. To find these correspondences it is not necessary to have camera calibration, nor to compute a full projective reconstruction. Thus our approach can be used with any photogrammetric model building package. We also use the computed projective quantities to autocalibrate the focal length of the camera. Once the exterior orientation is found, the user still needs to manually create the model, but this is now a simpler process.

1 INTRODUCTION

The problem of building geometric models has been a central application in photogrammetry. Our goal is to make this process simpler, and more efficient. The idea is to automatically find the features necessary for computing the exterior orientation for a given set of images. The claim is that this simplifies the model building process. To achieve this goal it is necessary to automatically create a reliable set of correspondences between a set of images without user intervention.

It has been shown that under certain conditions it is possible to reliably find correspondences, and to compute the relative orientation between image pairs. But these conditions are that the images are ordered, and have a small baseline, such as those obtained from a video sequence. However, in photogrammetric modeling the input images are not ordered and the baseline is often large. In this case it is necessary to identify which image pairs overlap, and then to compute correspondences between them.

Our solution to this problem contains a number of innovations. First, we use a feature finder that is effective in matching images with a wide baseline. Second, we use projective methods to verify that these hypothesized matches are in fact reliable correspondences. Third, we use the projective approach only to create a set of correspondences, and not to compute a projective reconstruction. At the end of the process these correspondences are sent to a bundle adjustment to find the exterior camera parameters. This makes the generation of the correspondences independent of the bundle adjustment. Therefore our approach can be used with any photogrammetric model building package. The computed projective quantities are the fundamental matrix between image pairs, and the trilinear tensor between image triplets. The supporting matches for the tensors are chained across images that are found to be adjacent to create correspondence chains. We show that this process creates very reliable correspondence chains. Finally, we use the computed projective information to autocalibrate the camera focal length.

Working in projective space has the advantage that camera calibration is not necessary. The other advantage is that it is possible to autocalibrate certain camera parameters, in particular the focal length, from the fundamental matrices. The complexity of computing the fundamental matrix, and trilinear tensor is less than that of the bundle adjustment.

2 DISCUSSION AND RELATED WORK

Model building is a very common industrial Photogrammetric application. The basic methodology has been unchanged for many years; calibrate the camera, choose the 2D projections of the 3D points of interest in different images, and then run the bundle adjustment. Then take the computed 3D data points, and join them together to create the model topology. The process is very manual, but has the advantage that the user can control the exact geometry and topology of the final model. Recently there has been some important recent work in the area of automated model building. However, the problem of finding the appropriate geometry and topology is a difficult and unsolved one. Thus the photogrammetric approach to model building will likely be used for a considerable length of time. However, while it is very flexible it is also very labor intensive.

Is it possible to partly automate the photogrammetric model building process and still leave it's flexibility intact? We believe the answer is yes. Normally the feature points used to build the model are also the ones used to run the bundle adjustment. But what if some other automatic process had already computed the exterior orientation? Then computing the required 3D data points for the model would require only triangulation, instead of the bundle adjustment. The triangulation process is inherently simpler than the bundle adjustment, so this would simplify the process of creating the required 3D geometry. If it were possible to automatically find a set of features between overlapping images, then it would be possible to run the bundle adjustment, and compute the exterior orientation. In the past, such matching has been done only in restricted situations. These restrictions are that the motion between the images is relatively small, and that there is some a-priori process which labels images as overlapping. For example, if the images were taken from a video sequence the baseline between then is small and they are also likely to have significant overlap.

If we consider the set of images that are used in a typical photogrammetry project these two restrictions do not hold. The input to the photogrammetric model building process is a set of unordered images with a wide physical separation, that is with a wide baseline. There have been a number of recent attempts to compute reliable correspondences between such images. The most complete system (Schaffalitzky and Zisserman, 2002) is a wide baseline, unordered matching system that uses affine invariants as the feature descriptor. This system attempts to construct a minimal spanning tree of the adjacency graph and it does not allow this spanning tree to form cycles. However, cycles are very important in the bundle adjustment process because they reduce the propogation of errors. Another approach (Ferrari et al., 2003) tries to expand on the number of correspondences between adjacent views by using the concept of tracks. The method of (Martinec and Pajdla, 2002) also has tracks, but uses a trilinear tensor to increase the number of tracks that have been found. There is also a variety of different descriptors (Baumberg, 2000, Pritchett and Zisserman, 1998) that have been used as features, other than the corner based approaches.

This paper describes a way of solving the wide baseline, unordered image matching problem which can interface with standardized photogrammetric packages. The input is a set of images without any intrinsic camera parameters, and the output is a set of correspondences between these images. These correspondences are then sent to a commercial photogrammetric package, which uses them to compute the exterior orientation of the cameras. While the bundle adjustment process requires camera calibration, the correspondence process does not. The pixel size, and aspect ratio for the camera, can be obtained from the camera manufacturers data. Since many modern digital cameras have a zoom lens this camera parameter is not easily obtained without a formal calibration step. The automatic correspondence process also has the ability to automatically find the focal length of the camera. Thus our system computes the correspondences between images and automatically calibrates the focal length.

Our approach computes the fundamental matrix between image pairs, and the trilinear tensor between image triples. It is based on the SIFT corner detector (Lowe, 1999), which has been shown experimentally to be the most robust corner features (Mikolajczk and Schmid, 2003).. The correspondences that support the trilinear tensor are chained strung together in longer tracks using a breadth first search. This method allows cycles in the image adjacency graph, and also uses a chained trilinear tensor, which makes it different from other approaches. The idea of using chained trilinear tensors was first described in (Roth and Whitehead, 2000), and has been shown to create very reliable correspondences. If there are n input images one would expect the overall complexity of this process to be $O(n^3)$. However, in practice, the complexity is actually $O(m^3)$, where m is the average number of images that have signficant overlap with any given image. We will show experimentally that m is typically about one half of n, the total number of images. Therefore $O(m^3)$ complexity is not excessive because m is relatively small. A typical photogrammetric project has thirty images (n) and there are at most ten to fifteen overlapping images (m) for each of these thirty images. The total processing time to compute all the correspondences in such a case is about fifteen minutes on a 2 gigahertz machine. Finally, as a by product we can use the calcuated fundamental matrices to perform autocalibration of the camera focal length.

3 ALGORITHM STEPS

The input is an unordered set of images. The output is a set of correspondences among these images. The processing is as follows:

```
For every image pair compute
  a fundamental matrix.
For all fundamental matrices
  that have enough supporting matches
  compute an associated trilinear tensor.
Chain the trilinear tensor matches
  using bread first search.
Output the chained matches
  as valid correspondences.
```

4 PROJECTIVE VISION

Structure from motion algorithms assume that a camera calibration is available. This assumption is removed in the projective paradigm. To explain the basic ideas behind the projective paradigm, we must define some notation. As in (Xu and Zhang, 1996), given a vector $\mathbf{x} = [x, y, \ldots]^T$, $\tilde{\mathbf{x}}$ defines the augmented vector created by adding one as the last element. The projection equation for a point in 3D space defined as $\mathbf{X} = [X, Y, Z]^T$ is:

$$s\tilde{\mathbf{m}} = P\tilde{\mathbf{X}}$$

where s is an arbitrary scalar, P is a 3 by 4 projection matrix, and $\mathbf{m} = [x, y]^T$ is the projection of this 3D point onto the 2D camera plane. If this camera is calibrated, then the calibration matrix C, containing the information particular to this camera (focal length, pixel dimensions, etc.) is known. If the raw pixel co-ordinates of this point in the camera plane are $\mathbf{u} = [u, v]^T$, then:

$$\tilde{\mathbf{u}} = C\tilde{\mathbf{m}}$$

where C is the calibration matrix. Using raw pixel co-ordinates, as opposed to actual 2D co-ordinates means that we are dealing with an uncalibrated camera.

Consider the space point, $\mathbf{X} = [X, Y, Z]^T$, and its image in two different camera locations; $\tilde{\mathbf{x}} = [x, y, 1]^T$ and $\tilde{\mathbf{x}}' = [x', y', 1]^T$. Then it is well known that:

$$\tilde{\mathbf{x}}^T E \tilde{\mathbf{x}}' = 0$$

Here E is the essential matrix, which is defined as:

$$E = [t] \times R$$

where t is the translational motion between the 3D camera positions, and R is the rotation matrix. The essential matrix can be computed from a set of correspondences between two different camera positions using linear methods (Longuet-Higgins, 1981). This computational process has been considered to be very ill-conditioned, but in fact a simple pre-processing step improves the conditioning, and produces very reasonable results (Hartley, 1997a). The matrix E encodes the epipolar geometry between the two camera positions. If the calibration matrix C is not known, then the uncalibrated version of the essential matrix is the fundamental matrix:

$$\tilde{\mathbf{u}}^T F \tilde{\mathbf{u}}' = 0$$

Here $\tilde{\mathbf{u}} = [u, v, 1]^T$ and $\tilde{\mathbf{u}}' = [u', u', 1]^T$ are the raw pixel co-ordinates of the calibrated points $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$. The fundamental matrix F can be computed directly from a set of correspondences by a modified version of the algorithm used to compute the essential matrix E. As is the case with the essential matrix, the fundamental matrix also encodes the epipolar geometry of the two images. Once E is known, the 3D location of the corresponding points can be computed. Similarly, once F is known the 3D co-ordinates of the corresponding points can also be computed, but up to a projective transformation. However, the actual supporting correspondences in terms of pixel co-ordinates are identical for both the essential and fundamental matrices. Having a camera calibration simply enables us to move from a projective space into a Euclidean space, that is from F to E.

5 FUNDAMENTAL MATRICES

We are given a set of n input images, and we want to calculate the fundamental matrix between every one of these n^2 image pairs. Consider a single pair of images from these set of n^2 images. We first find corners in each image, then find possible matches between these corners, and finally use these putative matches to compute the fundamental matrix. The number of matching corners that support a given fundamental matrix is a good indication of the quality or correctness of that matrix.

5.1 Corners/interest Points

The first step is to find a set of corners or interest points in each image. These are the points where there is a significant change in image gradient in both the x and y direction. The most common corner algorithm is describe by Harris (Harris and Stephens, 1988). While these corners are invariant to rotation in the camera plane they are sensitive to changes in scale, and also to rotations out of the camera plane. Recently, experiments have been done to test a newer generation of corners that are invariant over a wider set of transformations (Mikolajczk and Schmid, 2003). The results of this test show that SIFT corners perform best (Lowe, 1999). The SIFT operator finds at multiple scales, and describes the region around each corner by a histogram of gradient orientations. This description provides robustness against localization errors and small geometric distortions.

5.2 Matching Corners

Each SIFT corner is characterized by 128 unsigned eight bit numbers which define the multi-scale gradient orientation histogram. To match SIFT corners it is necessary to compare corner descriptors. This is done by simply computing the L^2 distance between two different descriptors. Assume that in one image there are j corners, and in the other there are k corners. Then the goal is for each of these j corners to find the closest of the k corners in the other image under the L^2 norm. This takes time proportional to jk, but since i and j are in the order of one thousand, the time taken is not prohibitive.

However, it is still necessary to threshold these L^2 distances in order to decide if a match is acceptable. Instead of using a fixed threshold for this L^2 distance, a dynamic threshold is computed. This is done by finding the first and second closest corner match. Then we compute the ratio of these two L^2 distances, and accept the match only if the second best match is significantly worse than the first. If this is not the case then the match is considered to be ambiguous, and is rejected. This approach works well in practice (Lowe, 1999), and avoids the use of an arbitrary threshold to decide on whether a pair of corners is a good match.

5.3 Computing the Fundamental Matrix

The next step is to use these potentially matching corners to compute the fundamental matrix. This process must be robust, since it can not be assumed that all of the matches are correct. Robustness is achieved by using concepts from the field of robust statistics, in particular, random sampling. Random sampling is a “generate and test process” in which a minimal set of correspondences, in this case the smallest number necessary to define a unique fundamental matrix (7 points), are randomly chosen (Rousseeuw, 1984, Bolles and Fischler, 1981, Roth and Levine, 1993, Torr and Murray, 1993, Xu and Zhang, 1996). A fundamental matrix is then computed from this best minimal set. The set of all corners that satisfy this fundamental matrix is called the support set. The fundamental matrix F_{ij}, with the largest support set SF_{ij} is returned by the random sampling process. The matching corners (support set) for two typical wide baseline views is shown in Figure 1.

Figure 1: The matching corners for two views

6 TRILINEAR TENSORS

While this fundamental matrix has a high probability of being correct, it is not necessarily the case that every correspondence that supports the matrix is valid. This is because the fundamental matrix encodes only the epipolar geometry between two images. A pair of corners may support the correct epipolar geometry by accident. This can occur, for example, with a checkerboard pattern when the epipolar lines are aligned with the checkerboard squares. In this case, the correctly matching corners can not be found using only epipolar lines (i.e. computing only the fundamental matrix). This type of ambiguity can only be dealt with by computing the trilinear tensor.

Assume that we see the point $\mathbf{X} = [X, Y, Z]^T$, in three camera views, and that 2D co-ordinates of its projections are $\tilde{\mathbf{u}} = [u, v, 1]^T$, $\tilde{\mathbf{u}}' = [u', v', 1]^T$, $\tilde{\mathbf{u}}'' = [u'', v'', 1]^T$. In addition, in a slight abuse of notation, we define $\tilde{\mathbf{u}}_i$ as the i'th element of $\mathbf{u}$; ie. $\mathbf{u}_1 = u$, and so on. It has been shown that there is a 27 element quantity called the trilinear tensor T relating the pixel co-ordinates of the projection of this 3D point in the three images (Shashua, 1995). Individual elements of T are labeled T_{ijk}, where the subscripts vary in the range of 1 to 3. If the three 2D co-ordinates $(\tilde{\mathbf{u}}, \tilde{\mathbf{u}}', \tilde{\mathbf{u}}'')$ truly correspond to the same 3D point, then the following four trilinear constraints hold:

$$
\begin{aligned}
u''T_{i13}\tilde{\mathbf{u}}_i - u''u'T_{i33}\tilde{\mathbf{u}}_i + u'T_{i31}\tilde{\mathbf{u}}_i - T_{i11}\tilde{\mathbf{u}}_i = 0 \\
v''T_{i13}\tilde{\mathbf{u}}_i - v''u'T_{i33}\tilde{\mathbf{u}}_i + u'T_{i32}\tilde{\mathbf{u}}_i - T_{i12}\tilde{\mathbf{u}}_i = 0 \\
u''T_{i23}\tilde{\mathbf{u}}_i - u''v'T_{i33}\tilde{\mathbf{u}}_i + v'T_{i31}\tilde{\mathbf{u}}_i - T_{i21}\tilde{\mathbf{u}}_i = 0 \\
v''T_{i23}\tilde{\mathbf{u}}_i - v''v'T_{i33}\tilde{\mathbf{u}}_i + v'T_{i32}\tilde{\mathbf{u}}_i - T_{i22}\tilde{\mathbf{u}}_i = 0
\end{aligned}
$$

In each of these four equations i ranges from 1 to 3, so that each element of $\tilde{\mathbf{u}}$ is referenced. The trilinear tensor was previously known only in the context of Euclidean line correspondences (Spetsakis and Aloimonos, 1990). Generalization to projective space is relatively recent (Hartley, 1995, Shashua, 1995).

The estimate of the tensor is more numerically stable than the fundamental matrix, since it relates quantities over three views, and not two. Computing the tensor from its correspondences is equivalent to computing a projective reconstruction of the camera position and of the corresponding points in 3D projective space.

6.1 Computing the Trilinear Tensor

We compute the trilinear tensor from the correspondences that form the support set of two adjacent fundamental matrices in the image sequence. Previously we computed the fundamental matrix for every pair of images. Now we filter out those image pairs that do not have more than a certain number of supporting matches. This leaves a subset of these n^2 image pairs that have valid fundamental matrices. Consider three images, i, j and k and their associated fundamental matrices F_{ij} and F_{jk}. Assume that these fundamental matrices have a large enough set of supporting correspondences, which we call SF_{ij} and SF_{jk}. Say a particular element of SF_{ij} is (x_i, y_i, x_j, y_j) and similarly an element of SF_{jk} is (x'_j, y'_j, x'_k, y'_k). Now if these two supporting correspondences overlap, that is if (x_j, y_j) equals (x'_j, y'_j) then the triple created by concatenating them is a member of CT_{ijk}, the possible support set of tensor T_{ijk}. The set of all such possible supporting triples is the input to the random sampling process that computes the tensor. The result is the tensor T_{ijk}, and a set of triples (corner in the three images) that actually support this tensor, which we call ST_{ijk}.

A tensor is computed for every possible triple of images. In theory this is $O(n^3)$, but in practice it is much less. The reason is that only a fraction (usually from 10 to 30 percent) of the n^2 possible fundamental matrices are valid. And from this fraction, an even smaller fraction of the possible triples are valid.

7 CHAIN THE CORRESPONDENCES

The result of this process is a set of trilinear tensors for three images along with their supporting correspondences. Say that we have a sequence of images numbered from 1 to n. Assume the tensors T_{ijk} and T_{jkl} have supporting correspondences $(x_i, y_i, x_j, y_j, x_k, y_k)$ in ST_{ijk} and $(x'_j, y'_j, x'_k, y'_k, x'_l, y'_l)$ in ST_{jkl}. Those correspondences for which (x_j, y_j, x_k, y_k) equals (x'_j, y'_j, x'_k, y'_k) represent the same corner in images i, j, k and l. In such cases we say that this corner identified in T_{ijk} is continued by T_{jkl}.

The goal of this step is to compute the maximal chains of supporting correspondences for a tensor sequence. This is done in a breadth first search using the supporting tensor correspondences as input. Individual correspondences that are continued by a given tensor are chained for as long as is possible. The output of the process is a unique identifier for a 3D corner, and its chain of 2D feature correspondences in a sequence of images. This corner list is then sent directly to the commercial bundle adjustment program Photomodeler (Photomodeler by EOS Systems Inc., n.d.) using a Visual Basic routine that communicates through a DDE interface.

8 AUTOCALIBRATION OF FOCAL LENGTH

To run the bundle adjustment it is necessary to know the camera calibration. However, it was not necessary to know the camera calibration to compute these correspondences. As a side effect of computing these correspondences we have a set of fundamental matrices between input images. It is possible to autocalibrate the camera parameters from these fundamental matrices.

The goal of autocalibration is to find the intrinsic camera parameters directly from an image sequence without resorting to a formal calibration process.

The standard linear camera calibration matrix K has the following entries (Hartley, 1997b):

$$C = \begin{pmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (1)$$

This assumes that the camera skew is $\pi/2$. Here f is the focal length in millimeters, and k_u, k_v the number of pixels per millimeter. The terms fk_u, fk_v can be written as α_u, α_v, the focal length in pixels on each image axis. The ratio α_u/α_v is the aspect ratio. It is often the case that all the camera parameters are known, except the focal length f. The reason is that many digital cameras have a zoom lens, and thus can change their focal length. The other camera parameters are specified by the camera manufacturer.

Thus a reasonable goal of autocalibration process is simply to find the focal length. This can be done reliably from the fundamental matrices that have been computed as part of the procedure to find the correspondences between image pairs (Roth, 2002).

8.1 Autocalibration by Equal Singular Values

If we know the camera calibration matrix K, then the essential matrix E is related to the fundamental matrix by $E = C^tFC$. The matrix E is the calibrated version of F; from it we can find the camera positions in Euclidean space. Since F is a rank two matrix, E also has rank two. However, E has the extra condition that the two nonzero singular values must be equal. This fact can be used for autocalibration by finding the calibration matrix C that makes the two singular values of F as close to equal as possible (Mendonca and Cipolla, 1999). Given two non zero singular values of E: σ_1 and σ_2 ($\sigma_1 > \sigma_2$), then, in the ideal case $(\sigma_1 - \sigma_2)$ should be zero. Consider the difference $(1 - \sigma_2/\sigma_1)$. If the singular values are equal this quantity is zero. As they become more different, the quantity approaches one. Given a fundamental matrix, autocalibration proceeds by finding the calibration matrix K which minimizes $(1 - \sigma_2/\sigma_1)$.

Assume we are given a sequence of n images, along with their fundamental matrices. Then F_i, the fundamental matrix relating images i and $i+1$, has non zero singular values σ_{i1} and σ_{i2}. To autocalibrate from these n images using the equal singular values method we must find the K which minimizes $\sum_{i=1}^{n-1} w_i(1 - \sigma_{i2}/\sigma_{i1})$. Here w_i is a weight factor, which defines the confidence in a given fundamental matrix. The weight w_i is set in proportion to the number of matching 2D feature points that support the fundamental matrix F_i. The larger this number, the more confidence we have in that fundamental matrix. In the case where only the focal length needs to be autocalibrated the minimization of this quantity is a simple one dimensional optimization process.

9 EXPERIMENTS

There are as yet no standardized data sets for testing wide baseline matching algorithms. However, there is one data set that has been used in a number of wide baseline matching papers (Schaffalitzky and Zisserman, 2002, Ferrari et al., 2003, Martinec and Pajdla, 2002), which is the Valbonne church sequence as shown in Figure 2.

Figure 2: Twelve pictures of the Valbonne Sequence

This sequence has a number of views of the church at Valbonne, in France. These views are typical of what would be used in a photogrammetric model building process. This sequence was processed by our software. There were approximately 350 feature points over these twelve images, and each feature point exists in at least five or more images. There are twelve images, so one would expect 12^2 fundamental matrices, and about 12^3 trilinear tensors to be calculated. However, only about fifty percent of the maximum number of fundamental matrices is calculated, and likewise, only thirty percent of the maximum number of trilinear tensors. A rendering of the camera positions and feature points is shown in Figure 3. The RMS residual error of each feature point when it is reprojected into the 2D image is at most 0.8 pixels, and at least 0.1 pixels. Thus we can see that the feature points are computed reliable enough to produce a good 3D reconstruction. The autocalibrated focal length of the camera is 610.00 pixels, while the true focal length is listed as 685 pixels, which is within about 15 percent of the correct value.

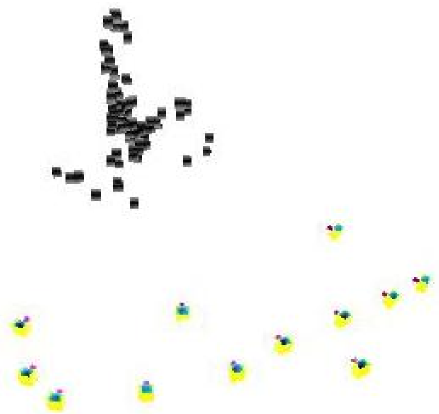

Figure 3: Rendering of camera positions and feature points

10 DISCUSSION

In this paper we have described a system which automatically computes the correspondences for an unordered set of overlapping images. These correspondences are then sent to a bundle adjustment process to compute the extrinsic camera parameters. This process does not require camera calibration, and in fact can autocalibrate the camera focal length. A demonstration version of this code can be found in http://www.cv.iit.nrc.ca/research/PVT.html.

REFERENCES

Baumberg, A., 2000. Wide-baseline muliple-view correspondences. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina, pp. 1774–1781.

Bolles, R. C. and Fischler, M. A., 1981. A ransac-based approach to model fitting and its application to finding cylinders in range data. In: Seventh International Joint Conference on Artificial Intelligence, Vancouver, British Colombia, Canada, pp. 637–643.

Ferrari, V., Tuytelaars, T. and Gool, L. V., 2003. Wide-baseline muliple-view correspondences. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, pp. 348–353.

Harris, C. and Stephens, M., 1988. A combined corner and edge detector. In: Proceedings of the 4th lvey Vision Conference, pp. 147–151.

Hartley, R., 1995. A linear method for reconstruction from lines and points. In: Proceedings of the International Conference on Computer Vision, Cambridge, Mass., pp. 882–887.

Hartley, R., 1997a. In defence of the 8 point algorithm. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19number 6.

Hartley, R., 1997b. Kruppa's equations derived from the fundamental matrix. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(2), pp. 133–135.

Longuet-Higgins, H., 1981. A computer algorithm for reconstructing a scene from two projections. Nature 293, pp. 133–135.

Lowe, D., 1999. Object recognition from scale invariant features. In: International Conference on Computer Vision, IEEE Computer Society, pp. 1150–1157.

Martinec, D. and Pajdla, T., 2002. Structure from many perspective images with occlusions. In: Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part II, Lecture Notes in Computer Science, Vol. 2351, Springer, pp. 355–369.

Mendonca, P. and Cipolla, R., 1999. A simple technique for self-calibration. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, pp. 112–116.

Mikolajczk, K. and Schmid, C., 2003. A performance evaluation of local descriptors. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition.

Photomodeler by EOS Systems Inc., n.d. http:/www,photomodeler.com.

Pritchett, P. and Zisserman, A., 1998. Wide baseline stereo matching. In: Proc. International Conference on Computer Vision, Bombay, India.

Roth, G., 2002. Some improvements on two autocalibratio algorithms based on the fundamental matrix. In: International Conference on Pattern Recognition: Pattern Recognition Systems and Applications, Quebec City, Canada.

Roth, G. and Levine, M. D., 1993. Extracting geometric primitives. Computer Vision, Graphics and Image Processing: Image Understanding 58(1), pp. 1–22.

Roth, G. and Whitehead, A., 2000. Using projective vision to find camera positions in an image seqeuence. In: Vision Interface 2000, Montreal, Canada, pp. 87–94.

Rousseeuw, P. J., 1984. Least median of squares regression. Journal of American Statistical Association 79(388), pp. 871–880.

Schaffalitzky, F. and Zisserman, A., 2002. Multi-view matching for unordered image sets. In: Proceedings of the 7th European Conference on Computer Vision, Copenhagen,Denmark.

Shashua, A., 1995. Algebraic functions for recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(8), pp. 779–789.

Spetsakis, M. and Aloimonos, J., 1990. Structure from motion using line correspondences. International Journal of Computer Vision 4(3), pp. 171–183.

Torr, P. and Murray, D., 1993. Outlier detection and motion segmentation. In: Sensor Fusion VI, Vol. 2059, pp. 432–443.

Xu, G. and Zhang, Z., 1996. Epipolar geometry in stereo, motion and object recognition. Kluwer Academic.