Epipole Estimation Using
Affine Motion Parallax

Jonathan Lawn and Roberto Cipolla

CUED/F-INFENG/TR 138

July 16, 1993

A reprint of a paper to be
presented at BMVC'93

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

Email: jml@eng.cam.ac.uk

Epipole Estimation Using Affine Motion Parallax

Jonathan Lawn and Roberto Cipolla
Department of Engineering,
University of Cambridge,
Cambridge CB2 1PZ, England.

Abstract

Determining the motion of a camera from its image sequences has so far proved very difficult, and no practical algorithms have been found for freely moving cameras. This novel algorithm is based on motion parallax, but uses sparse visual motion estimates to extract the direction of translation of the camera directly, after which determination of the camera rotation and the depths of the image features follows easily. This method can also detect and reject independent motion, and provide a measure of the uncertainty of its estimates.

1 Introduction

Relative motion between a viewer and a scene provides a visual cue for the determination of the scene structure and the viewer motion. If the camera motion is known to a reasonable accuracy, only the *depths* in the scene need be computed (the distances from the camera centre to the points in the scene). However, many systems would benefit from using a freely moving camera with no external motion sensors, which requires that the visual motion must be decomposed to give the camera motion and the scene structure.

There are two common methods for estimating the camera motion from an image pair or sequence. One class attempts to find both the translation and the rotation of the camera simultaneously. For instance, the 3x3 essential matrix [12, 20], a linear representation of the epipolar constraint, can be decomposed into the five egomotion parameters. However this estimation of the mutually dependent variables of the matrix makes no reference to this and it can introduce excessive errors [5]. This approach particularly fails to distinguish between rotations and translations perpendicular to the optic axis, especially when perspective effects are small. The other class of methods finds approximate solutions for one component, either the rotation [16] or the translation [1, 10], by assuming that it is dominant. This is much more direct, but the solutions tend to be biased when the other component is significant. Extracting one component of the egomotion exactly would avoid the disadvantages of both of the above.

It has long been known that the relative visual motions of coincident points can be used to cancel the rotational part of the visual motion, and therefore extract the visual motion due to the viewer translation only [7, 13]. This effectively allows the camera motion to be treated as a pure translation, with all visual motion towards or away from the *epipole*, the intersection of the direction of motion with the imaging surface. The absolute magnitude of the camera translation cannot be found, because of the speed-scale ambiguity, though it can be expressed in terms of the scene depths. Once the epipole is known, the camera rotation can be found from the component of the visual motion orthogonal to it. Unfortunately motion parallax has many shortcomings, particularly the need for dense velocity field measurements at sudden depth changes [8, 17].

Here we present a method that will find the epipole independent of the camera rotation and point depths and that does not require the instantaneous alignment of features or dense image velocity measurements. A method of calculating the uncertainty of the estimates is also presented. Section 2.1 presents the affine motion parallax algorithm, 2.2 explains the novel use to extract the epipole, 2.3 shows how uncertainty analysis can allow optimal estimation, and 2.4 how the formulation of the algorithm can be used to reject spurious visual motion information. The results of an implementation using sparse visual motion information are then presented in Section 3.

2 Theory

2.1 The Affine Motion Parallax Algorithm

This algorithm measures motion parallax without requiring the instantaneous alignment of features by using affine (linear) approximations to projection. Below is given an introduction to the notation, a description of each theory, and then the algorithm that combines them. It is then shown how this can be used to find the epipole from only sparse image velocity measurements.

Visual Motion: Stationary *features* points in space are given in the camera coordinate system by vectors $\underline{X}$ measured from the camera (projection) centre. However only the directions of these vectors can be measured by a camera from a single position, and so the image positions of the points are represented here by the vectors to the intersections of $\underline{X}$ with the unit sphere, $\underline{Q}$, and the image velocities by their velocities, $\dot{\underline{Q}}$. (An alternative is to use a projection plane, which is less elegant because involves the introduction of an extra variable, the plane normal.)

$$\underline{Q} = \frac{\underline{X}}{|\underline{X}|} \tag{1}$$

As the camera moves with translational velocity $\underline{U}$ and angular velocity $\underline{\Omega}$ (also in camera coordinates), the relative positions of the points in space change

$$\dot{\underline{X}} = -\underline{\Omega} \times \underline{X} - \underline{U} \tag{2}$$

($\times$ is the symbol for the vector product), and this causes the image points to move as well. Differentiation of (1) and substitution into (2) gives [2, 15]

$$\dot{\underline{Q}} = -\underline{\Omega} \times \underline{Q} - (\underline{Q} \times \frac{1}{r}\underline{U}) \times \underline{Q} \tag{3}$$

where $r = |\underline{X}|$, the depth. This clearly shows that the visual motion is made up of two components, one depending on the translational velocity of the camera and the scene structure (depths), and one depending on the rotational velocity and only the image positions.

Motion Parallax: If two points in space, a and b, project to the same point on the imaging sphere momentarily, then their image velocities will have identical rotational components (from 3), and the difference in their image velocities, $\dot{\Delta}_{\underline{Q}}$, will depend only on the translational velocity of the projection centre and on their image position and depths. This is called *motion parallax* [13].

$$\dot{\Delta}_{\underline{Q}} \equiv (\dot{\underline{Q}}_a - \dot{\underline{Q}}_b)\,|_{\underline{Q}_a = \underline{Q}_b = \underline{Q}} \tag{4}$$

$$= (\underline{Q} \times \underline{U}) \times \underline{Q}(\frac{1}{r_b} - \frac{1}{r_a}) \tag{5}$$

$$= (I - \underline{Q}\underline{Q}^T)\underline{U}(\frac{1}{r_b} - \frac{1}{r_a}) \tag{6}$$

where $\underline{Q}^T$ is the transpose of $\underline{Q}$. The epipole, $\underline{Q}_E$ is defined by the direction of the camera velocity vector $\underline{U}$, and so they are parallel ($\underline{U} \times \underline{Q}_E = 0$). It therefore follows from (6) that

$$(\dot{\Delta}_{\underline{Q}} \times \underline{Q}).\underline{Q}_E = 0 \tag{7}$$

This implies that the epipole is constrained by each measurement of $\dot{\Delta}_{\underline{Q}}$ to lie on a great circle of the unit projection sphere (of axis $\dot{\Delta}_{\underline{Q}} \times \underline{Q}$), and therefore two measurements in different parts of the visual field can determine its direction. Unfortunately implementation requires sudden depth changes and dense 2D velocity field estimates (eg. from a continuous measurement method or dense corners). The solution will be ill-conditioned if:

- the points used to calculate the parallax are not close to coincident
- there is no significant depth change between the points, giving only a small velocity difference, or
- the field of view is too small to allow good triangulation on the epipole.

Affine Transformations: Weak perspective [18] makes the assumption that there is a linear (*affine*) transformation between the positions of the points in 3D and their projections. From small perturbation analysis of (3), it can be seen that the image velocity is in general a non-linear function of both the image position,$\underline{Q}$, and the scene depth, r, and therefore, for the visual motion to vary linearly, the visual region considered must be small and contain small depth changes.

A point in a certain plane in space will therefore have an image velocity that is a linear function of its image position only. This *affine transformation* can be determined from, for instance, the visual motion of a minimum of three points in the plane, or the deformation of a closed curve in the plane [2].

Affine Motion Parallax: Conventional motion parallax uses two points instantaneously aligned, but this algorithm uses four nearby points. The motion parallax comes from one real point, and one virtual point, taken to be momentarily behind or in front of the real one but at a different depth. The movement of this virtual point is calculated by assuming it is on a plane defined by three other nearby points. The algorithm therefore allows much sparser features to be used than conventional motion parallax methods. This implementation used corners, with the virtual partner for each provided by its three nearest neighbours (see figure 1), but future work will investigate using closed curves to define the affinely deforming planes, and using the curves centroids to define the parallax points.

This algorithm was originally used to determine the axis of rotation of an object in front of a stationary camera [3]. Here it is extended so that a number of measurements of affine motion parallax from distinct small linear regions can determine the epipole.

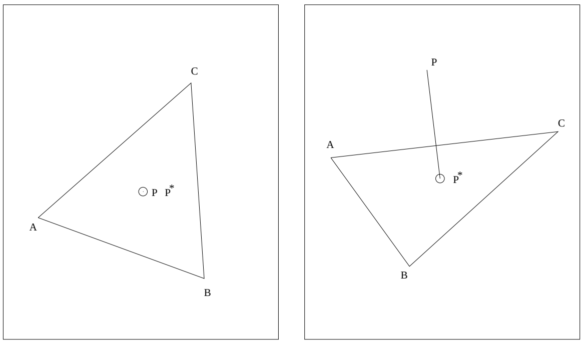

Figure 1: The Affine Motion Parallax Algorithm
Point P^* is defined to be in the plane defined by points A,B and C and momentarily coincident with point P in the image (left). When the viewpoint of the scene changes the displacement of P^* is defined by the locally linear deformation of the image of the plane, which can be determined from the displacements of A,B and C (right). The difference in the displacement of the real point P and the virtual point P^* (resulting from their different depths) is the motion parallax.

2.2 Epipole Determination

Methods that attempt to extract the epipole that rely on the small global perspective effects tend to suffer badly with noisy data. Longuet-Higgins [13] showed as above that two or more local measurements of motion parallax can determine the epipole but true parallax is difficult to measure. Using the weak perspective assumption globally results in parallel motion parallax vectors [11], as it is only valid for small fields of view. However weak perspective assumptions can be used in a number of small regions of a wide view.

In this paper, a novel method is presented that extends the use of affine motion parallax to a number of small neighbourhoods in a full perspective image, so that the relative velocities can be used to determine the epipole. The epipole can be obtained entirely from geometric construction on the image plane, though here it is mapped onto the sphere to avoid infinities. Each neighbourhood generates a motion parallax vector which provides a great circle constraint on the imaging sphere on which the epipole must lie (7), using only sparse visual motion estimates. Two or more separated cases can determine the epipole exactly at the intersection of their constraints. The camera rotation can then be found by a least-squares estimate from the visual motion perpendicular to the epipole, and the scene depths can then be determined from the visual motion towards the epipole not accounted for by this rotation.

2.3 Uncertainty Estimation

The feature measurement errors on the image plane determine the image velocity measurement uncertainty, which then in turn affects the rest of the calculation on to the motion parallax vectors and then the epipole estimate. If all the image positions are represented as normalized 3D vectors to the projection sphere, then all the calculations can be expressed in standard vector arithmetic. Assuming that the uncertainty on each vector (and scalar) in the calculations is small, additive and gaussian, then it can be represented by a covariance matrix [9] which can be calculated with the estimated value. The significance of the motion parallax vectors can now be defined in terms of their uncertainty estimates, for instance. If any point is found to have no significant affine motion parallax, it may well be in the plane of the three points providing its virtual pair, and a new basis can be chosen if necessary.

Epipole estimates can be generated from the normalised vector product of every possible pair of constraint planes normals (axies of the great circles) given by the motion parallax (7). The optimal estimate is then the weighted sum of these estimates with the lowest predicted uncertainty. Intuitively those estimates which have the greatest uncertainty should receive the smallest weights. However the exact weighting depends on the choice of the scalar uncertainty measure to be minimised (see the appendix for the calculations in detail). [1]

Independence of all estimates is assumed during the calculations. However where estimates or constraints come from shared original measurements are combined, independence is lost. Here, the weights calculated for the constraints assume their independence, and therefore in reality will be only approximately optimal. Also the scale of the uncertainty will be wrong, and so the number of times the measurements have been overused on average is taken as the scaling factor for the covariance matrix of the result. (For instance, the estimation of the epipole combines $\frac{1}{2}N(N-1)$ vector products from only N constraint normals, so the calculated variance for the epipole is multiplied by $\frac{1}{2}(N-1)$ to compensate.)

2.4 Detecting Independent Motion

Independent motion is when a point in the scene is not stationary in 3D space, and therefore has a different motion relative to the camera. Incorrect visual motion measurements (caused by bad matching or tracking of features, or by independently moving objects or spurious features) can be removed at the constraint fusion stage. Starting from the best estimate available from two constraints, only those constraints that agree (to within their uncertainty bands) are included. If too few constraints agree then a new initial estimate is needed. Otherwise all those features that were not part of any included constraint can be rejected, and if necessary the calculation can be repeated to improve its accuracy. Other ways of recognizing bad features include checking that their depth is positive and changes smoothly.

3 Implementation

3.1 The Measurement of Visual Motion

This algorithm requires image velocities or displacements at a number of points. *Corners* are 2D image structures and therefore those points in the view that can be tracked most easily. The corner detector used in this implementation is based on Tomasi [19] and Harris [6], ie. looking for small regions of pixels with high intensity gradients in all directions. These algorithms will recognise all forms of

[1] This estimate has also been made in other work, usually in the context of finding vanishing points, and Magee and Aggarwal [14] used a similar system. Kanatani [9] and Collins and Weiss [4] found the smallest eigenvalue of the sum of the outer products of the normal vectors, but by this method the important optimization requires an initial estimate of the epipole, and not just the assumption that they are all reasonably accurate. Others (eg. Hildreth [8]) have found the maximum of a discrete histogram on the sphere, with each 'bin' scoring for each constraint plane that approximately intersects it, but achieving accuracy and producing an uncertainty estimate would require a very dense array of bins and therefore excessive computation.

2D feature and can grade them for localizability, whilst only taking one derivative of the intensity field which increases the immunity to camera noise. Tracking schemes to determine the displacement of each corner in the next frame that were considered for real-time implementation include finding corners and then matching them using their proximities and similarities [6], using intensity gradients to perform an iterative search [19], and making SSD (sum of the squared differences) comparisons. However, reliable tracking is still a difficult problem, and the benefits of smooth real-time motion prediction are lost to an implementation for discrete views. Figure 2 shows the corners found on the first image of the sequence.

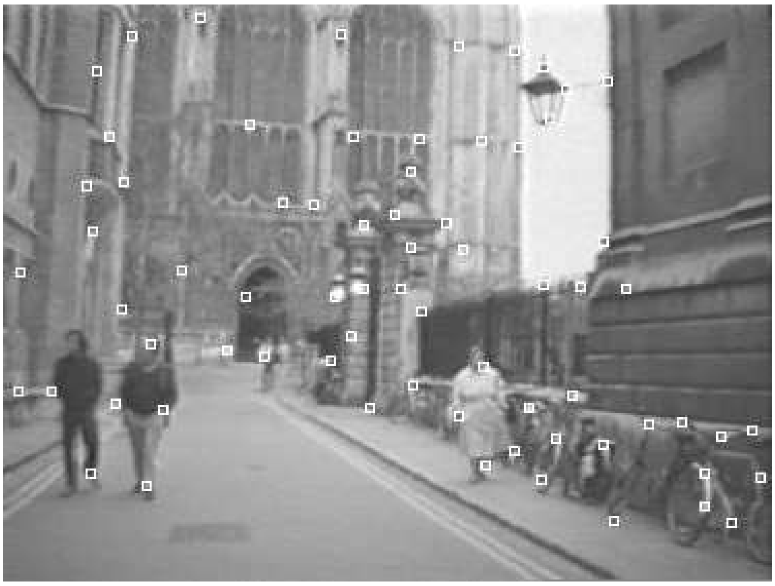

Figure 2: The approach to Kings Chapel North Gate (Frame 1) showing the corners found by Tomasi's algorithm [19].

3.2 Experiments

An epipole finder for pairs of images, which uses the affine motion parallax algorithm given above, was implemented using first hand-picked corners, and then corners found using the detector described above (section 3.1) (but matched by hand to avoid the correspondence problem). The uncertainty tracking system was also implemented starting from a priori estimates of the corner measurement uncertainties, allowing an optimal estimate of the epipole to be found. The results showed that the algorithm produced a robust epipole estimate, within its calculated uncertainty margin and usefully accurate.

The sequence below shows estimates of the epipole being made from a sparse set of corners. There is a realistic amount of camera rotation between the frames as well as a translation towards the gates. It can be seen that the algorithm is not only accurate with only a dozen features, but also fails gracefully as the information given reduces further. Since the reliability of the epipole measurement is also estimated, Kalman filtering of the frame pair estimates would be simple. The figures show an ellipse centred on the optimal estimate indicating the standard

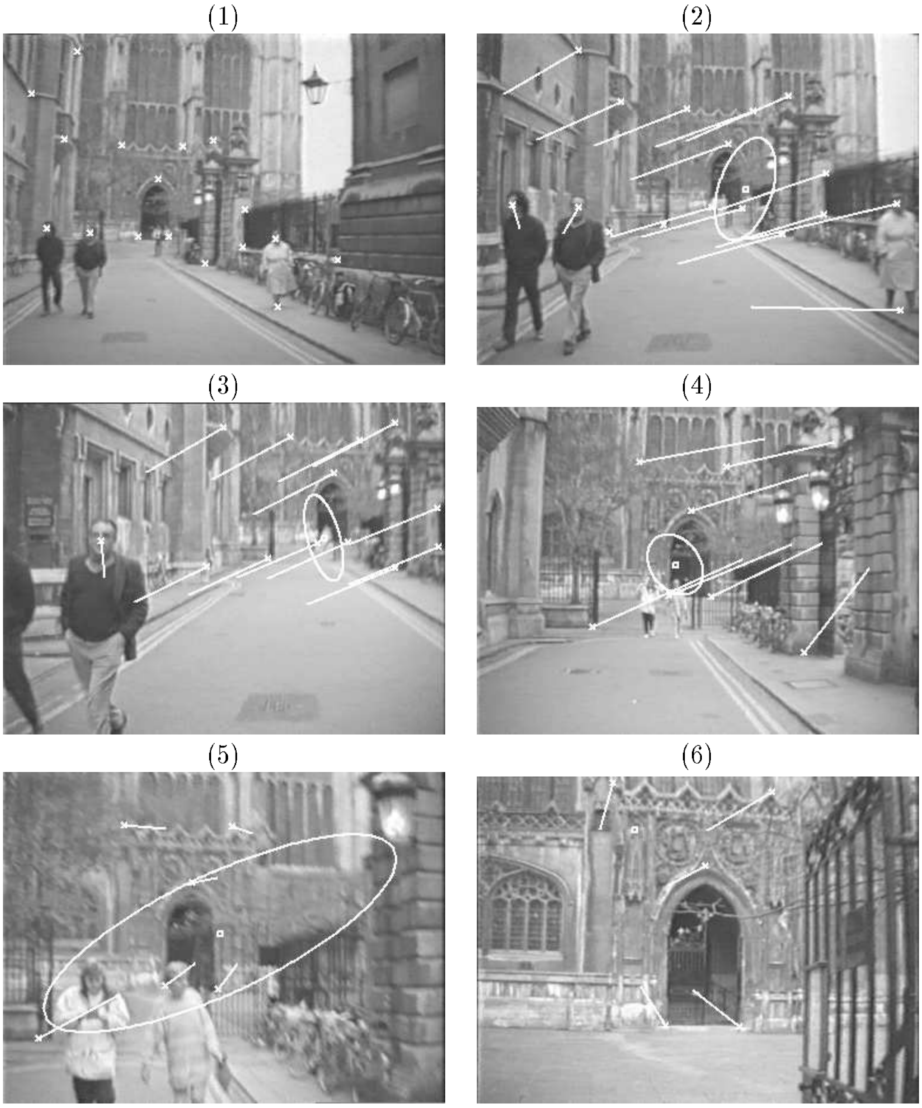

Figure 3: The approach to Kings Chapel North Gate (Frames 1–6) showing the feature movements from the previous frames and the epipole estimates with their 1 s.d. ellipses. The sparse features in frames 3 and 4 still provide good epipole estimates, but the decrease in the information content, caused by the frame 5 having only one feature out of the plane of the wall, is also clearly demonstrated by the increased uncertainty. The last frame contains only points on the wall and can therefore provide only one constraint plane causing the uncertainty to explode.

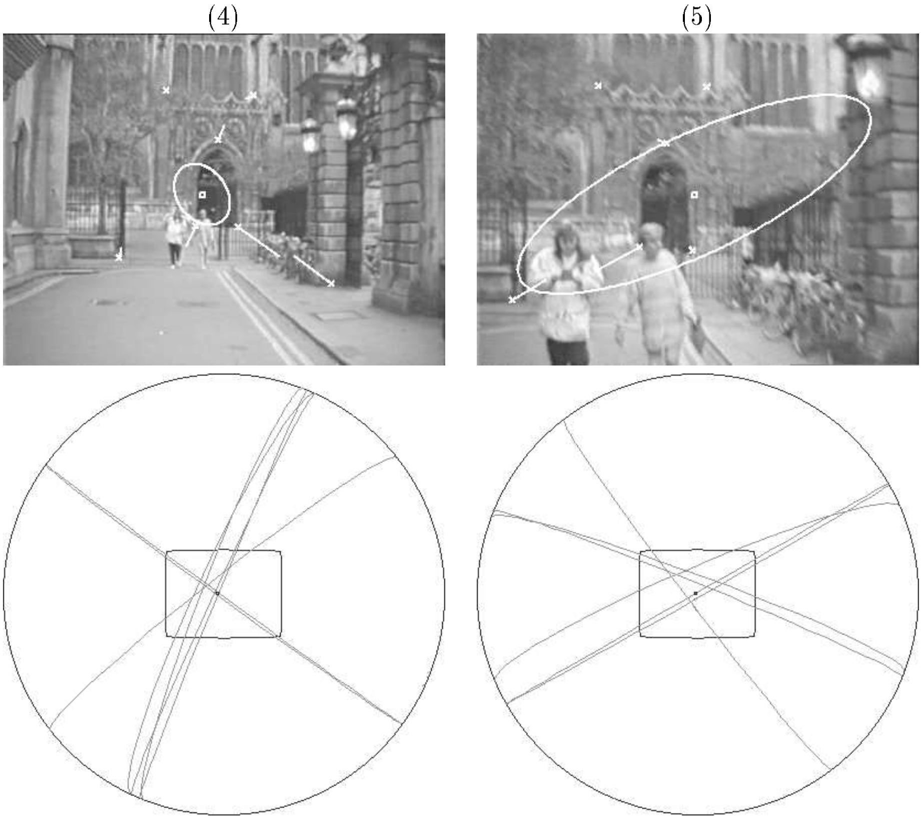

Figure 4: The approach to Kings Chapel North Gate (Frames 4 & 5) showing the affine motion parallax directions calculated for the features with length representing certainty (top), and the constraint planes they provide, plotted as great circles on a hemisphere, and the optimal estimate of the intersection, allowing for the uncertainties of each constraint (bottom). Again the decrease in information content to frame 5 is evident, with only one constraint plane found.

deviation assuming that the feature position estimates error is gaussian with a standard deviation of 1 pixel.

Though small visual regions are being used, the depth changes in many triplets that form the affine bases are too large for the affine approximation to be very accurate. However the estimates of the affine motion parallax they provide will be considerably more accurate than could be obtained from pairs of points, as is done in other parallax algorithms [8, 17], or by assuming that there is no rotation [1, 10]. The discrete motions are approximated by velocities in the calculations.

Figure 3 shows the sampled image sequence, grabbed from videotape. The first three frames have plenty of points and variations in depth, but some of those detected are on independently moving objects (people). The later frames have decreasing numbers of points and depth to demonstrate the gradual failure to find the epipole. Figure 4 shows the affine motion parallax calculated from two frame pairs on the borderline of having insufficient data points.

3.3 Discussion

Some of points chosen in the frame sequence are on independently moving bodies (people) but none can be detected as such by the epipolar constraint (translation causes no visual motion perpendicular to the epipole) because they all move in a plane with the camera. This is very often true for scenes like this: only points that are not at eye-level and are not travelling in the same directions the camera, such as the feet of someone crossing the road, can be detected in this way. All points shown are used in the calculation to show its robustness.

In the later frames the independent points are not present but neither are others leaving an even sparser point set. Despite this the algorithm still recovers the epipole accurately (even when there is only one strong constraint (frame 5) though it is less certain) until only planar points remain (frame 6).

4 Conclusions and Summary

Presented above is a method of extracting the epipole exactly from sparse, noisy and corrupted points by decomposing the image velocities into their translational and rotational components. Once this has been determined, camera rotation and the relative point depths can also be found. The algorithm also provides an estimate of its accuracy, which allows it to be combined optimally with other information and its reliability to be gauged. This method is applicable to any problem where the camera motion is unknown, as it provides an efficient and elegant decomposition of visual motion information. Since it fails gracefully and provides uncertainty estimates, it is considerably more practical than essential matrix methods [12, 20] which can produced dramatically wrong results with noisy data. Results of an initial implementation using corners back up these claims.

Acknowledgements

We are grateful the late Professor Fallside for creating the Speech-Vision-Robotics group. Jonathan Lawn is supported by a SERC student grant.

References

[1] Y. Aloimonos and Z. Duric. Active egomotion estimation : A qualitative approach. In *Proc. 2rd European Conf. on Computer Vision*, pages 497–510, 1992.

[2] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In G. Sandini, editor, *Proc. 2nd European Conference on Computer Vision*, pages 187–202. Springer–Verlag, 1992.

[3] R. Cipolla, Y. Okamoto, and Y. Kuno. Robust structure from motion using motion parallax. In *Proc. 4th Int. Conf. on Computer Vision*, 1993.

[4] R.T. Collins and R.S. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *Proc. 3rd Int. Conf. on Computer Vision*, 1990.

[5] K. Daniilidis and H-H. Nagel. Analytical results on error sensitivity of motion estimation from two views. *Image and Vision Computing*, 8(4):297–303, 1990.

[6] C.G. Harris. In A. Blake and A. Yuille, editors, *Active Vision*, chapter 16. MIT Press, 1992.

[7] H. von. Helmholtz. *Treatise on Physiological Optics*. Dover (New York), 1925.

[8] E.C. Hildreth. Recovering heading for visually guided navigation in the presence of self-moving objects. *Philosophical Transactions of the Royal Society of London, Series B*, 337, 1992.

[9] K. Kanatani. Computational projective geometry. *Computer Vision, Graphics and Image Processing (Image Understanding)*, 54/3:333–348, 1991.

[10] J.J. Koenderink. Optic flow. *Vision Research*, 26(1):161–179, 1986.

[11] J.J. Koenderink and A.J. van Doorn. Affine structure from motion *J. Opt. Soc. America*, 8:377–385, 1991.

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

[13] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. of the Royal Society of London, Series B*, 208:385–397, 1980.

[14] M.J. Magee and J.K. Aggarwal. Determining vanishing points in perspective images. *Computer Vision, Graphics and Image Processing*, 26:256–267, 1984.

[15] S.J. Maybank. The angular velocity associated with the optical flow field arising from motion through a rigid environment. *Proc. Royal Society, London*, A401:317–326, 1985.

[16] J.A. Perrone. Model for the computation of self-motion of biological systems. *J. Opt. Soc. America*, A9(2), February 1992.

[17] J.H. Rieger and D.L. Lawton. Processing differential image motion. *J. Opt. Soc. America*, A2(2):354–360, 1985.

[18] L.G. Roberts. Machine perception of three - dimensional solids. In J.T. Tippet, editor, *Optical and Electro-optical Information Processing.* MIT Press, 1965.

[19] C. Tomasi. *Shape and Motion from Image Streams: a Factorization Method.* PhD thesis, CMU, Sept 1991.

[20] X. Zhuang, T.S. Huang, and R. M. Haralick. A simplification to linear two-view motion algorithms. *Computer Vision, Graphics and Image Processing*, 46:175–178, 1989.

A The Optimal Sum of Estimates

Let $\tilde{\underline{x}}$ be the optimal estimate of $\underline{x}$. It is found from a number of estimates $\underline{x}_i$ by taking the weighted average of them, $\tilde{\underline{x}} = \sum w_i \underline{x}_i$ (where $\sum w_i = 1$), that minimises the expected standard deviation σ. The estimates are assumed to be unbiased, so that their expected value $E[\underline{x}_i] = \underline{x}$, and to have only independent small additive gaussian noise, $\underline{x}_i = \underline{x} + \Delta\underline{x}_i$. This is represented by a known covariance matrices, $V[\underline{x}_i] = E[\Delta\underline{x}_i \Delta\underline{x}_i^T]$. σ^2 is defined as $\sigma_x^2 + \sigma_y^2 + \sigma_z^2$, ie. the trace of $V[\tilde{\underline{x}}]$, and σ_i^2 is the trace of $V[\underline{x}_i]$.

$$\begin{aligned}
\tilde{\underline{x}} &= \sum w_i \underline{x}_i \\
E[\tilde{\underline{x}}] &= E\left[\sum w_i \underline{x}_i\right] \\
\sigma^2[\tilde{\underline{x}}] &= E\left[\left(\sum w_i \Delta\underline{x}_i\right)^2\right] = E\left[\sum (w_i \Delta\underline{x}_i)^2\right] = \sum w_i^2 \sigma_i^2
\end{aligned}$$

We want to minimise $\sigma^2[\tilde{\underline{x}}] = \sum w_i^2 \sigma_i^2$ given that $\sum w_i = 1$.
Therefore minimise $C = \sum(w_i^2 \sigma_i^2) - \lambda(\sum w_i - 1)$ for each w_i (Lagrange) :

$$\begin{aligned}
\frac{dC}{dw_i} &= 2w_i\sigma_i^2 - \lambda = 0 \\
\sum w_i &= \sum \frac{\lambda}{2\sigma_i^2} = \lambda \sum \frac{1}{2\sigma_i^2} = 1 \\
\lambda &= \left(\sum \frac{1}{2\sigma_i^2}\right)^{-1} \text{ and } w_i = \frac{1}{\sigma_i^2 \sum \frac{1}{\sigma_i^2}}
\end{aligned}$$

Each estimate is weighted proportional to the inverse of its variance, $\frac{1}{\sigma_i^2}$.