INTEGRATION OF STEREO AND MOTION

Ed Sparks & Mike Stephens

Roke Manor Research Ltd
Romsey, Hampshire SO51 0ZN

In extracting three-dimensional positional information from images, both static stereo processing and monocular structure-from-motion processing possess shortcomings for a general purpose vision system. A solution to these problems is to use stereo structure-from-motion processing. This paper addresses the problems of integrating stereo and structure-from-motion data to determine the camera ego-motion and the locations of features by Kalman filtering. Results from monocular and stereo structure-from-motion algorithms are presented.

INTRODUCTION

Since the early days of computer vision, attempts have been made to extract information about the three-dimensional (3D) structure of the observed scene using only the information available from the imagery. Some of the most successful systems developed have been based upon 'meaningful' features extracted from the images. Such features - corner points, edges, regions, etc. - often arise from real world objects. The initial techniques were stereo, using a pair of static cameras, and structure-from-motion, using a single moving camera. However both of these techniques have their short-comings, and this has prompted researchers to try and combine both stereo and motion into a single framework. We are concerned here with developing techniques to coherently integrate the data in two of the most important areas; the 3D location of features, and the calculation of the camera viewpoint (the so-called camera ego-motion). The results are given for the motion of a binocular pair of cameras, though they apply to any number of cameras.

Stereo systems [1,2] use two (or more) cameras in known relative positions (and orientations). The known camera dispositions are used to reduce the search areas in feature matching by using the epi-polar constraint, and once features are matched, in calculating the resulting 3D position by using the camera separation as the baseline for triangulation. The resulting accuracy of the feature positions decreases quadratically with depth (the distance from the cameras). This range limitation means that stereo can be applied only in situations where the objects of interest are close to the cameras. Attempting to overcome this difficulty by increasing the camera separation causes difficulties in matching, and also reduces the number of possible matches because of occlusion.

The monocular structure-from-motion approach uses a single camera moving through the scene [3]. Unlike conventional (ie. snap-shot) stereo, the number of frames is unlimited, allowing the possibility of refining the positions of features by observing them over a wide baseline. The disadvantage is that the camera position for any frame may not be known exactly, necessitating the ego-motion to be determined from the images themselves. This calculation for a single moving camera introduces the speed-scale ambiguity into the system, since from visual data alone the scale of the scene and the magnitude of the motion form an irresolvable ambiguity.

The solution to these problems is to combine the two techniques, that is to use stereo structure-from-motion. The absolute base-line provided by the separation of the stereo cameras enables the speed-scale ambiguity to be resolved, and the long base-line resulting from processing a long image sequence enables good long-range accuracy to be obtained.

The problems involved in matching are not addressed in this paper, but the algorithms we have used are discussed in [3]. Both 2D-2D matching (eg. stereo matching; see Figure 4) and 3D-2D matching will be assumed to have been performed, with the majority of matches correct. Incorrect matches will adversely affect the position of the 3D features involved, but the calculation of ego-motion being dependent upon all the matches may be made robust (see below).

The analysis developed below is for a moving stereo camera head in an otherwise static world. We use feature-point data extracted from images using the corner operator [4], to form a scene representation consisting of 3D feature-points. These are specified by their positions and positional uncertainties expressed as normal probability distribution functions. Straight edges are widely used in extracting 3D information and the results presented below could be reformulated to use them. This is, however, beyond the scope of this paper.

KALMAN FILTERING OF POINT LOCATIONS

At this stage we assume that the camera location and attitude are known correctly from the ego-motion calculation (see below), or at least known to sufficient accuracy. Each extracted image point possesses uncertainty as to its exact location, due mainly to the effects of spatial sampling. We assume therefore that the image position of the true point is described probabilistically by a normal distribution centred upon the located point; this we call the observation error. The resultant distribution of point positions in 3D is of a non-normal form, possessing conical equi-probability density surfaces with apex at the camera pin-hole. To overcome the problems associated with this non-normal

BMVC 1990 doi:10.5244/C.4.24

form, we choose to work in so-called disparity space, with coordinates (X/Z ,Y/Z ,1/Z). Here, the Z axis lies along the camera optical axis, and the X and Y axes lie in the image plane. This gives a feature-point observation a normal 3D form, which can be described by its disparity mean or centroid position, **R**, and a disparity covariance matrix C. Use of disparity space is not essential to the analysis, and may be replaced by a cartesian representation if it makes the reader more comfortable.

Monocular Approach

Let the Kalman Filter (KF) representing a 3D point before the incorporation of the information of the current (t'th) observation have centroid $\mathbf{R}_t$ and co-variance matrix C_t in the disparity-space of the camera local coordinate system. Let the match on the current image be located at $\mathbf{s}_t$, with observation error (ie. co-variance matrix) H_t. The KF update equations are given by [5]

$$C_{t+1} = \left(C_t^{-1} + P\, H_t^{-1}\, P^T \right)^{-1}$$

$$\mathbf{R}_{t+1} = C_{t+1} \left(C_t^{-1} \mathbf{R}_t + P\, H_t^{-1} \mathbf{s}_t \right)$$

where $P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$ is the 2D to 3D projection matrix.

This update is shown graphically in Figure 1, illustrated by the surfaces of constant probability density.

Stereo Approach

For stereo, just the same procedure may be applied, sequentially updating the Kalman Filter of a point from the data from each camera in turn. The conventional stereo approach is to use the stereo matches from the current image to instantiate (ie. start off) a new Kalman Filter for each pair of matched points on the current image, then to combine this with the previous Kalman Filter for the point. As expected from the use of KFs, this may be shown to give an identical result to the sequential approach.

DETERMINING THE EGO-MOTION

The ego-motion is the location and attitude of the camera (or stereo camera head) with respect to a reference coordinate system, and is necessary for the proper integration of data from images comprising a sequence. The ego-motion is defined by a translation **t**, followed by a rotation specified by the vector $\boldsymbol{\theta}$, whose direction is the axis of rotation, and whose magnitude is the angle of rotation. The rotation matrix generated by $\boldsymbol{\theta}$ is

$$A_{ij}(\boldsymbol{\theta}) = \cos(\theta)\, \delta_{ij} + (1-\cos(\theta))\, \hat{\boldsymbol{\theta}}_i \hat{\boldsymbol{\theta}}_j$$

$$- \sin\theta \sum_{k=1}^{3} \varepsilon_{ijk}\, \hat{\boldsymbol{\theta}}_k$$

where ε_{ijk} is the Levi-Civita symbol

This ego-motion is undertaken with respect to a reference coordinate system, which we shall choose to be our best guess at the actual camera location, as this makes the numerical value of the ego-motion small, and permits subsequent linearisation of the equations. The ego-motion is denoted by the 6-vector $\mathbf{q}=(\boldsymbol{\theta},\mathbf{t})$. The optimal estimate of the ego-motion is calculated by finding the vector **q** which brings the projection of the 3D features into best alignment with the observed image points to which they are matched. The error in alignment is measured in terms of a combined probability error distribution of the 3D features and the image points.

Single Camera

To determine the optimal ego-motion, we define a cost function, E(**q**), which measures the mis-alignment of the observed features and the projected 3D features by calculating the squared Mahalanobis distance (ie. number of standard deviations) between the projection $\mathbf{r}_i$ of the 3D feature $\mathbf{R}_i$, and the image point $\mathbf{s}_i$ to which it has been matched. The total cost function is the sum of the squared Mahalanobis distances for all n of the matches:

$$E(\mathbf{q}) = \sum_{i=1}^{n} E_i(\mathbf{q})$$

$$= \sum_{i=1}^{n} (\mathbf{r}_i(\mathbf{q}) - \mathbf{s}_i)^T L_i^{-1}(\mathbf{q})\, (\mathbf{r}_i(\mathbf{q}) - \mathbf{s}_i)$$

where the match covariance, $L_i(\mathbf{q})$, is the sum of $K_i(\mathbf{q})$, the projected covariance matrix of the 3D feature, and H_i, the covariance matrix of the observation $\mathbf{s}_i$.

Minimising the cost function leads to the best estimate of the ego-motion. The minimisation is performed iteratively using the Newton-Raphson method. At each step, k, of the iteration we refine the ego-motion estimate :

$$\mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} - E''^{\,-1}\, \mathbf{E}'$$

where $\mathbf{E}'_{ij} = \partial E(\mathbf{q}^{(k)}) / \partial \mathbf{q}_i^{(k)}$

and $E''_{ij} = \partial^2 E(\mathbf{q}^{(k)}) / \partial \mathbf{q}_i^{(k)} \partial \mathbf{q}_j^{(k)}$

An initial estimate, $\mathbf{q}^{(0)}$, of the motion, may be derived either from some external measurements (eg. odometry), or by motion prediction from earlier frames.

In calculating the partial derivatives of E we assume that the projected co-variance matrix K is only slowly varying with **q**, and that the projection of the 3D point is in good alignment with the observations, ie. that **r**-**s** is small. Noting that the matrix L is symmetric gives, for the m'th matched point,

$$\partial E_m(\mathbf{q}) / \partial \mathbf{q}_i \approx -2\,(\mathbf{r}_m(\mathbf{q}) - \mathbf{s}_m)^T\, L_m^{-1}\, \partial \mathbf{r}_m(\mathbf{q})/\partial \mathbf{q}_i$$

$$\partial^2 E_m(\mathbf{q}) / \partial \mathbf{q}_i \partial \mathbf{q}_j \approx -2\,\partial \mathbf{r}_m^T(\mathbf{q})/\partial \mathbf{q}_i\, L_m^{-1}\, \partial \mathbf{r}_m(\mathbf{q})/\partial \mathbf{q}_j$$

To prevent incorrect matches spoiling the minimisation, a robust deweighting is used. To each point contributing to the total cost function is associated a weight, which falls smoothly to zero as the Mahalanobis distance of that point rises above a few standard deviations. This serves to gracefully disregard incorrectly matched points as the minimising iteration proceeds.

Multiple cameras

When using two (or more) cameras we do not want to calculate a separate ego-motion for both cameras since this fails to capitalise on the strong constraint of relative camera displacement. Nor can we combine together two **q** vectors since they will not in general represent the same motion. We therefore combine the E derivatives, so providing a single estimate of **q**. However, the two cameras in a stereo system do not undergo the same ego-motion, because of their relative displacement. Let the second camera be situated a distance **p** from the first camera, and have a relative attitude expressed by the rotation vector φ. Suppose that the first camera undergoes an ego-motion $\mathbf{q}=(\boldsymbol{\theta},\mathbf{t})$ then the second camera will undergo an ego-motion (relative to its own start position) of

$$\begin{aligned}\mathbf{q}' &= (\boldsymbol{\theta}',\mathbf{t}') \\ &= (\, A^T(\varphi)\boldsymbol{\theta}\ ,\ A^T(\varphi)(\,[A(\boldsymbol{\theta})-I]\,\mathbf{p} + \mathbf{t}\,)\,)\end{aligned}$$

as shown in Figure 2. Thus the cost function we wish to minimise contains the following term from the n' matches obtained from the second camera

$$\begin{aligned}E(\mathbf{q}') &= \sum_{i=1}^{n'} E_i(\mathbf{q}') \\ &= \sum_{i=1}^{n'} (\,\mathbf{r}_i(\mathbf{q}')-\mathbf{s}_i)^T\, L_i^{-1}\,(\,\mathbf{r}_i(\mathbf{q}')-\mathbf{s}_i)\end{aligned}$$

To introduce this additional term into the Newton minimisation, the first and second derivatives with respect to **q** must therefore be calculated. To obtain these differentials, make use of the chain rule of differentiation

$$d/d\mathbf{q} = (\partial \mathbf{q}'/\partial \mathbf{q})\,\partial/\partial \mathbf{q}'$$

Noting that

$$\partial \boldsymbol{\theta}'/\partial \boldsymbol{\theta} = A^T(\varphi)$$

$$\partial \boldsymbol{\theta}'/\partial \mathbf{t} = 0$$

$$\partial \mathbf{t}'/\partial \boldsymbol{\theta} = A^T(\varphi)\,\partial\,[A(\boldsymbol{\theta})\mathbf{p}]\,/\partial \boldsymbol{\theta}$$

$$\partial \mathbf{t}'/\partial \mathbf{t} = A^T(\varphi),$$

the explicit form of the first and second differentials may be determined. These are given in the appendix.

The calculations performed for each matched point (ie. the differentials of E with respect to **q**') are independent of the camera on which they were located. Only the sum of derivatives need be transformed. Thus the transformations, given in the appendix, need only be applied once per camera per iteration of the Newton minimisation.

Stereo Approach

A second approach to calculating the ego-motion first instantiates the stereo matches into 3D to form KFs for each stereo matched pair of points, with centroids $\mathbf{S}_i$ and covariances D_i, the latter constructed from the observation errors, H_i. A total cost function similar to that above may now be defined using the square Mahalanobis distance between the 3D feature, $\mathbf{R}_i$, and the transformed (by the ego-motion, **q**) stereo feature $\mathbf{S}_i$:

$$E(\mathbf{q}) = \sum_{i=1}^{n} \mathbf{X}^T \left(\, C_i + D_i(\mathbf{q})\,\right)^{-1} \mathbf{X}$$

$$\mathbf{X} = (\mathbf{S}_i(\mathbf{q}) - \mathbf{R}_i)$$

where $\mathbf{S}_i(\mathbf{q})= A^T(\boldsymbol{\theta})(\mathbf{S}_i-\mathbf{t})$ is the transformed stereo feature-point position, and $D_i(\mathbf{q}) = A^T(\boldsymbol{\theta})\, D_i\, A(\boldsymbol{\theta})$ is the transformed co-variance matrix (all the above variables are in cartesian space). As before, C_i is the co-variance of the 3D feature $\mathbf{R}_i$.

If we compare the contribution of a single stereo feature, which is the square of the Mahalanobis distance between the stereo feature and the 3D feature, against the value from the squared Mahalanobis distances of the observations which combined to form the stereo feature, we find that they are identical. This shows that the stereo ego-motion determination can be performed in either approach, and the same answer obtained, provided the same matches are used. In practice, the first approach is better, because it also takes account of points seen in only one camera.

RESULTS

A comparison of the performance of monocular and stereo structure-from-motion was made using synthetic data, as this provided knowledge of the true camera motion that we currently lack on real imagery. The camera(s) traversed a sinusoidal path over a flat textured floor, always pointing in the +Z direction. In Figure 3 are shown the true path and the calculated camera ego-motions from both monocular and stereo structure-from-motion, as obtained from the DROID computer vision system (figure 5) [3]. Good motion estimates were provided to both the monocular and stereo systems, so that the difference in the paths is due to feature-point positional noise alone. It is clearly shown that stereo-motion confers the advantages hoped of it.

The main problem with the monocular processing is due to lack of odometry. The breaking of the speed-scale ambiguity on the first two frames of the sequence by specifying the camera speed, 'freezes' a scale factor into the rest of the sequence. In fact, long term drift can occur on the scale, position and orientation of the ego-motion. This can be seen in Figure 3, where the latter portion of the sequence is shrunk in magnitude towards the origin.

Figure 4 shows an example of stereo matching for a single pair of images (matched points shown bright). The matcher is similar to the 2D-2D temporal matcher, [3], which uses weak epi-polar constraints and grey-level attributes.

CONCLUSIONS

For stereo structure-from-motion both the determination of ego-motion and the updating of the Kalman Filters of the 3D points may be undertaken by either the conventional stereo or our alternative approach, and identical results obtained if the same matches are used. However, the conventional stereo approach is computationally more expensive as it involves working in 3D as opposed to 2D. In addition, the conventional stereo approach will give worse results if the matches are reduced by stereo occlusion, by inconsistent feature detection (due to features being close to the detection threshold), or by a reduced stereo overlap region (due for example, to multiple cameras not being used in a conventional stereo configuration, but with limited overlap or different magnifications).

REFERENCES

[1] J. Porill et al *TINA: A 3D vision system for pick and place* AVC 87 pp. 65-72 (1987)

[2] N.Ayache & F.Lustman *Fast and reliable passive trinocular stereovision* ICCV 87 pp. 422-427 (1987)

[3] M.J.Stephens et al *Outdoor vehicle navigation using passive 3D vision* CVPR 89 pp. 556-562 (1989)

[4] C.G.Harris & M.J.Stephens *A combined corner and edge detector* AVC 88 pp.147-152 (1988)

[5] C.G.Harris & J.M.Pike *3D positional integration from image sequences* AVC 87 pp. 233-236 (1987)

APPENDIX - EGO-MOTION DERIVATIVES FOR MULTIPLE CAMERAS

$$\frac{dE}{d\theta_i} = \sum_{j=1}^{3} A_{ji}^{T}(\varphi)\,\frac{\partial E}{\partial \theta'_j} + \sum_{jkl=1}^{3} A_{jk}^{T}(\varphi)\,\frac{\partial A_{kl}(\boldsymbol{\theta})}{\partial \theta_i}\,p_l\,\frac{\partial E}{\partial t'_j}$$

$$\frac{dE}{dt_i} = \sum_{j=1}^{3} A_{ji}^{T}(\varphi)\,\frac{\partial E}{\partial t'_j}$$

$$\frac{d^2E}{dt_i dt_j} = \sum_{kl=1}^{3} A_{ki}^{T}(\varphi)\;A_{lj}^{T}(\varphi)\,\frac{\partial^2 E}{dt'_k \partial t'_l}$$

$$\frac{d^2E}{d\theta_i dt_j} = \sum_{kl=1}^{3} A_{ki}^{T}(\varphi)\,A_{lj}^{T}(\varphi)\,\frac{\partial^2 E}{\partial \theta'_k \partial t'_l} + \sum_{klmn=1}^{3} A_{kl}^{T}(\varphi)\,A_{nj}^{T}(\varphi)\,\frac{\partial A_{lm}(\boldsymbol{\theta})}{\partial \theta_i}\,p_m\,\frac{\partial^2 E}{\partial t'_k \partial t'_n}$$

$$\frac{d^2E}{d\theta_i d\theta_j} = \sum_{kl=1}^{3} A_{ki}^{T}(\varphi)\,A_{lj}^{T}(\varphi)\,\frac{\partial^2 E}{\partial \theta'_k \partial \theta'_l} + \sum_{klmn=1}^{3} A_{kj}^{T}(\varphi)\,A_{lm}^{T}(\varphi)\,\frac{\partial A_{mn}(\boldsymbol{\theta})}{\partial \theta_i}\,p_n\,\frac{\partial^2 E}{\partial \theta'_k \partial t'_l}$$

$$+ \sum_{klmn=1}^{3} A_{ki}^{T}(\varphi)\,A_{lm}^{T}(\varphi)\left(\frac{\partial^2 A_{mn}(\boldsymbol{\theta})}{\partial \theta'_k \partial \theta_j}\,p_n\,\frac{\partial E}{\partial t'_l} + \frac{\partial A_{mn}(\boldsymbol{\theta})}{\partial \theta_j}\,p_n\,\frac{\partial^2 E}{\partial \theta'_k \partial t'_l}\right)$$

$$+ \sum_{klmnrs=1}^{3} A_{kr}^{T}(\varphi)\,A_{lm}^{T}(\varphi)\,\frac{\partial A_{rs}(\boldsymbol{\theta})}{\partial \theta_i}\,\frac{\partial A_{mn}(\boldsymbol{\theta})}{\partial \theta_j}\,p_s\,p_n\,\frac{\partial^2 E}{\partial t'_k \partial t'_l}$$

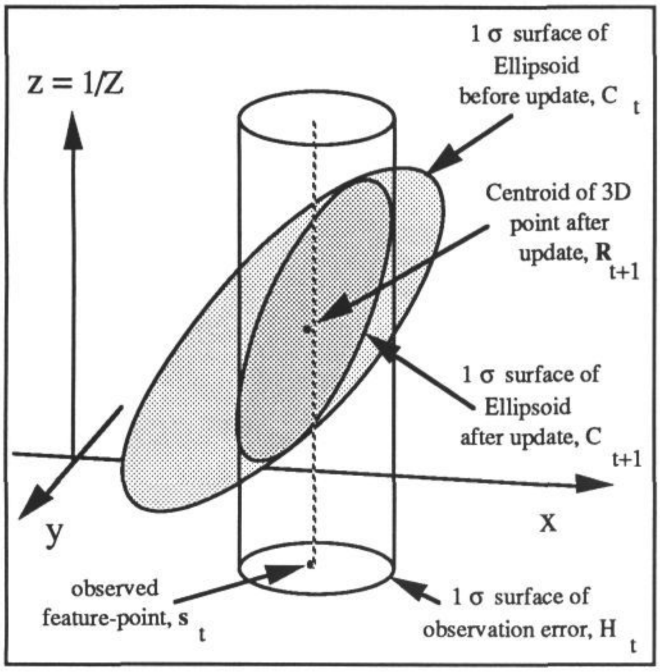
$z = 1/Z$
1 σ surface of Ellipsoid before update, C_t
Centroid of 3D point after update, $\mathbf{R}_{t+1}$
1 σ surface of Ellipsoid after update, C_{t+1}
y
x
observed feature-point, $\mathbf{s}_t$
1 σ surface of observation error, H_t

Figure 1. The Kalman Filter update of a 3D point position.

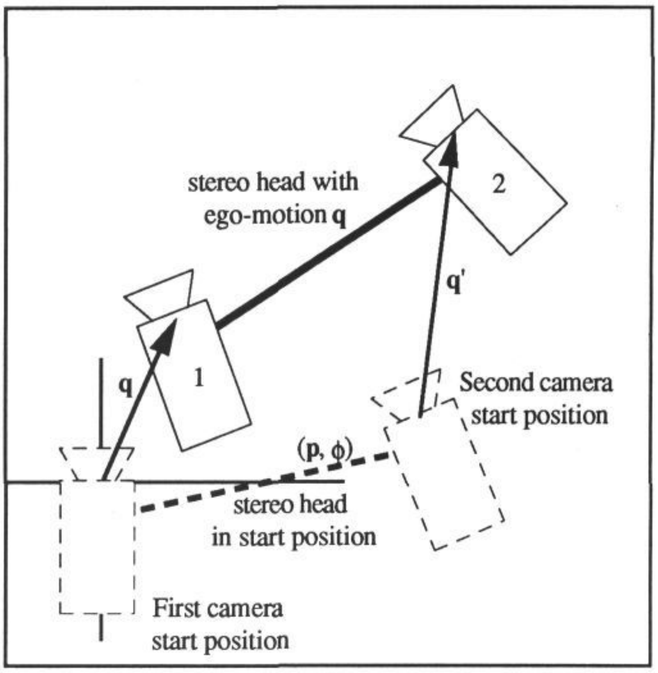
stereo head with ego-motion **q**
2
q'
q
1
Second camera start position
(**p**, ϕ)
stereo head in start position
First camera start position

Figure 2. The motion, **q'**, of a second camera in a stereo configuration.

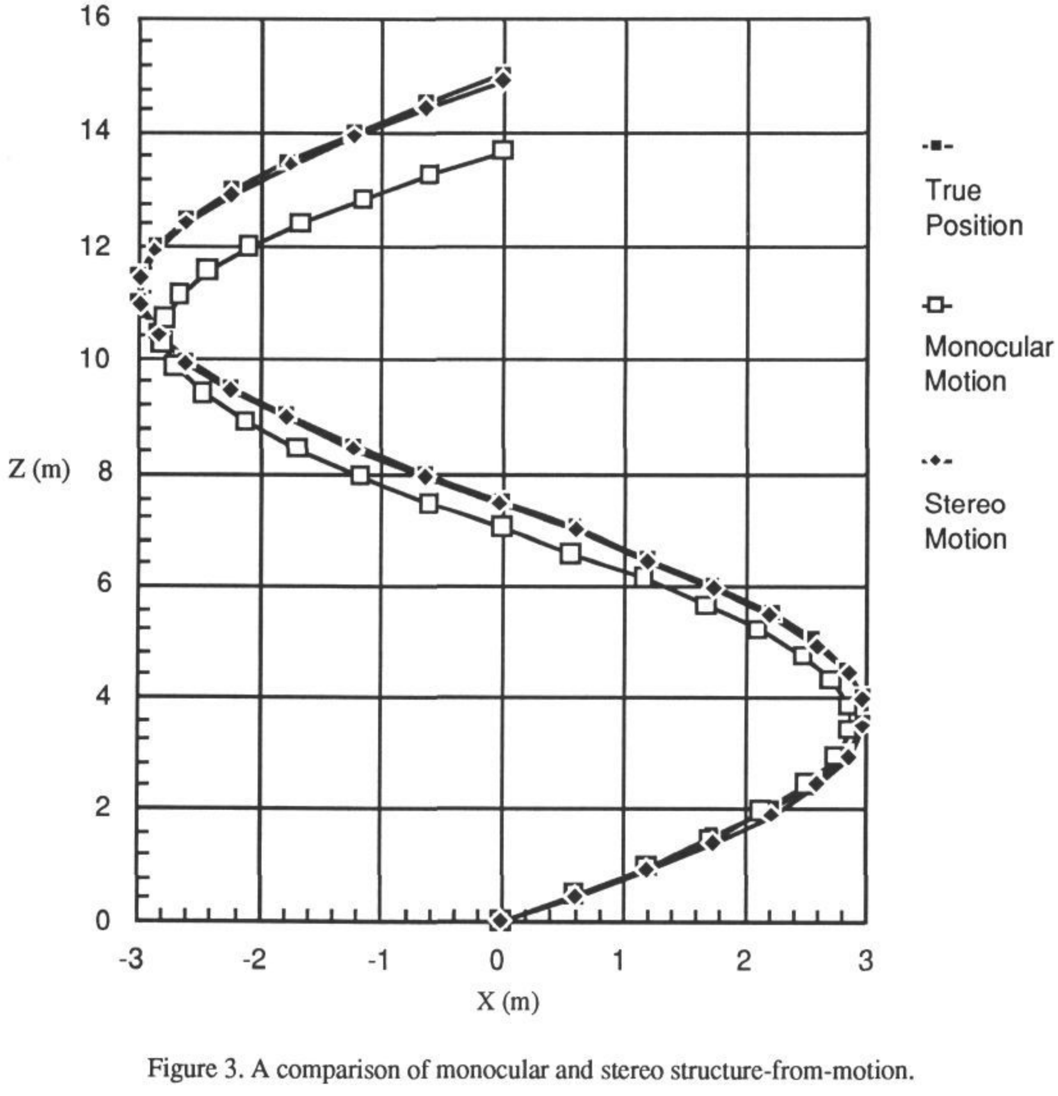

Figure 3. A comparison of monocular and stereo structure-from-motion.

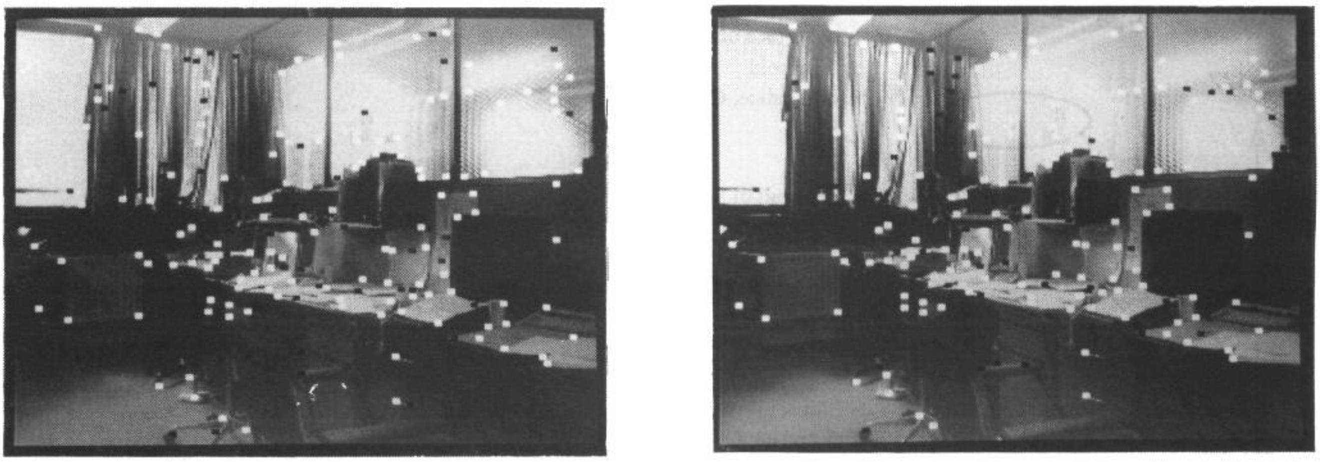

Figure 4. Stereo Matches (shown bright)

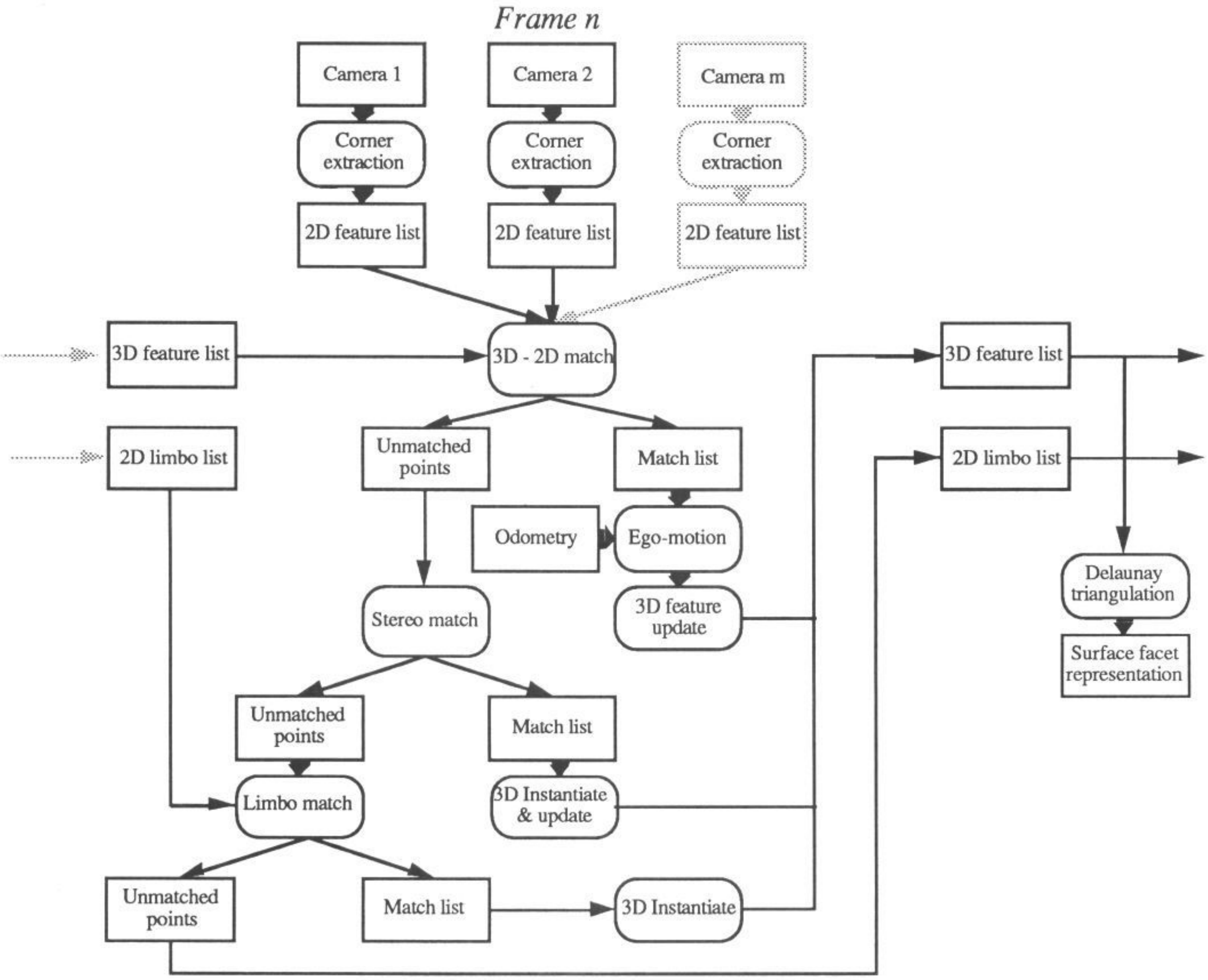

Figure 5. Stereo DROID flowchart