Sixth International Symposium on
Robotics Research, MIT Press, 1993

Input Redundancy and Output Observability in the Analysis of Visual Motion

Carlo Tomasi
Computer Science Department
Cornell University
Ithaca, NY 14853

Abstract

Determining structure and motion from the images produced by a camera on a moving robot is a nonlinear and potentially poorly conditioned computation. A reliable system must use redundant data so that small random errors in the inputs cancel out statistically to produce better outputs. Furthermore, the output quantities to be computed must be well observable, that is, they should have well-measurable effects on the images. I demonstrate the importance of these two principles with three system for the analysis of visual motion: the factorization method for structure and motion under orthography, a system to compute camera motion from narrowly spaced frames, and a global, multiframe, and multifeature method for the reconstruction of structure and motion from sequences taken under perspective.

1 Introduction

To function effectively in the world, a robot must be able to perceive the three-dimensional structure of its environment and to determine its own motion therein. Therefore, an important goal of computer vision is to build reliable systems for the computation of structure and/or motion from the images produced by a camera on a moving robot. If the world is stationary and if feature points can be tracked from image to image, the computation of structure and motion becomes a purely geometric problem. It is, however, a nonlinear and potentially poorly conditioned one, for which reliability is a hard goal to achieve.

One necessary ingredient for reliability is *redundancy*. The computation of structure and motion from redundant data becomes an overdetermined minimization problem, and small random errors in the inputs cancel out statistically to produce better results. Although this fact is widely recognized, however, much work in motion vision still takes the minimalist approach of identifying the smallest number of images and feature points needed to determine the solution. This approach is a heritage from the days when a computer could neither hold nor process more than a couple of images at a time; it led to interesting theoretical statements but to no working system.

A less recognized ingredient is the *observability* of the output quantities. If two unknown quantities, outputs of the computation, lead to the same effects on the image, they cannot be distinguished just from image measurements. More relaxedly, with imperfect measurements, two unknown quantities that have very similar effects on the image are hard to distinguish. For instance, a camera translating a small step to the right will induce an image change that is similar to that caused by the camera panning a little to the right instead. Camera rotation and translation, then, will be hard to tell apart in this situation. Similarly, the images produced by an object that is, say, onehundred meters away are not much different from those of the same object ninety-nine meters away. Consequently, the distance from camera to scene will be hard to determine with good accuracy. The notion of observability has been known well and explicitly in systems theory (see for instance [10]): if the output of a system is the same for two given states, the two states cannot be distinguished by just looking at the outputs.

Redundant data and observable outputs are the two main strands of the thread through my research in the interpretation of visual motion. On the one hand, using redundant data leads to global minimization problems in large spaces, which call for special computational techniques. On the other, the need to restrict computation

[0]This research was supported by the National Science Foundation under contract IRI-9201751

to observable quantities requires to reinvent the underlying problem representation. In the following, I illustrate these two points through three pieces of work. The first and the last describe two methods for the computation of structure and motion from image sequences, under orthographic and perspective projection respectively. The one in-between is a method for the computation of camera motion from two narrowly spaced frames.

All these methods combine the two ingredients of input redundancy and output observability. In the factorization method, the first case study, arbitrarily many points and frames are used to determine structure and motion under orthography. This projection model is valid for faraway scenes, and depth, hard to observe, does not appear in the equations. In the second example, as many points as feasible are used to determine inter-frame camera translation under perspective. Rotation, difficult to distinguish from translation, is eliminated by introducing image *deformations*, as opposed to image motion. Furthermore, depth, hard to determine from two close frames even for nearby scenes, is eliminated by subspace techniques from linear algebra. In the last illustration, a multiframe, multifeature, and global method is presented for the computation of both structure and motion under perspective projection. Again, deformations are used as the input, thereby removing rotation from the output representation and leading to a more reliable computation.

2 The Factorization Method

In principle, the structure of a scene can be computed from a sequence of images by first estimating camera motion and depth, and then inferring structure from the depth values. In practice, however, when objects are distant from the camera relative to their size, this computation is ill-conditioned. First, the translation component along the optical axis is difficult to determine because the changes that it produces are small. Second, structure values are very sensitive to noise if they are computed as the small differences between large depth values. These difficulties can be circumvented by inferring structure directly from variations in the relative position of image features, without computing depth as an intermediate step.

In Ullman's original proof of existence of a solution [23] for the structure from motion problem under orthography, as well as in the perspective formulation in [15], the coordinates of feature points in the world are expressed in a world-centered system of reference. Since then, however, this choice has been replaced by most computer vision researchers with that of a camera-centered representation of structure [14], [4], [22], [1], [24], [2], [9], [6], [7], [12], [17], [3]. With this representation, the position of feature points is specified by their image coordinates and by their depths, defined as the distances between the camera center and the feature points, measured along the optical axis. Unfortunately, although a camera-centered representation simplifies the equations for perspective projection, it makes structure estimation difficult, unstable, and sensitive to noise.

There are two fundamental reasons for this. First, when camera motion is small, effects of camera rotation and translation can be confused with each other, as explained in the introduction. Any attempt to recover or differentiate between these two motions, though doable mathematically, is naturally sensitive to noise. Second, the computation of structure as relative depth, for example, the height of a building as the difference of depths between the top and the bottom, is sensitive to noise, since it is a small difference between large values.

In the factorization method [21] both difficulties disappear because the problem is reformulated in world-centered coordinates, unlike the conventional camera-centered formulation. This new (old – in a sense) formulation links object-centered structure to image motion directly, without using retinotopic depth as an intermediate quantity, and leads to a simple and well-behaved solution. Furthermore, the mutual independence of structure and motion in world-centered coordinates makes it possible to cast the structure-from-motion problem as a factorization problem, in which a matrix representing image measurements is decomposed directly into camera motion and object shape.

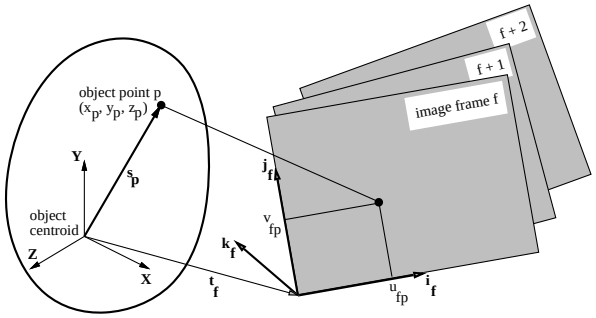

Figure 1: The basic notation used in the factorization method.

In the method, we represent an image sequence as a

$2F \times P$ measurement matrix W, which is made up of the horizontal and vertical coordinates of P points tracked through F frames (see figure 1). If image coordinates are measured with respect to their centroid, we prove the *rank theorem*: under orthography, the measurement matrix is of rank 3. As a consequence of this theorem, we show that this matrix can be factored into the product of two matrices R and S where R is a $2F \times 3$ matrix that represents camera rotation, and S is a $3 \times P$ matrix which represents structure in a coordinate system attached to the object centroid. The two components of the camera translation along the image plane are computed as averages of the rows of W. More specifically, the factorization works in two steps that compute affine and then Euclidean structure and motion:

1. Compute the singular-value decomposition $W = O_1 \Sigma O_2$, truncated to the first three singular values and the corresponding eigenvectors, of the centered measurement matrix W.

2. Find a linear transformation Q that transforms the intermediate matrices $\hat{R} = O_1(\Sigma)^{1/2}$ and $\hat{S} = (\Sigma)^{1/2} O_2$ into rotation R and structure S by imposing the constraint that scanlines and columns are orthogonal in each image.

When features appear and disappear in the image sequence due to occlusions or tracking failures, the resultant measurement matrix W is only partially filled in. The factorization method can handle this situation by growing a partial solution obtained from an initial full submatrix into a full solution with an iterative procedure.

The following experiment illustrates the factorization method with a sequence of real images. In the experiment, a hand holds a cup and rotates it in front of the camera by about ninety degrees. Figure 2 shows four out of the 240 frames of the stream. A total of 207 features was selected. Figure 3 shows the image trajectory of 60 randomly selected features.

Figures 4 and 5 show a front and a top view of the cup and the visible fingers as reconstructed by the factorization method. The shape of the cup was recovered, as well as the rough shape of the fingers. A more detailed reconstruction would be possible if more features were available.

In conclusion, the factorization method exploits the redundancy of the measurement matrix to counter the noise sensitivity of structure-from-motion. Furthermore, the orthographic projection model eliminates depth from the output quantities to be determined, leading to stable and reliable results.

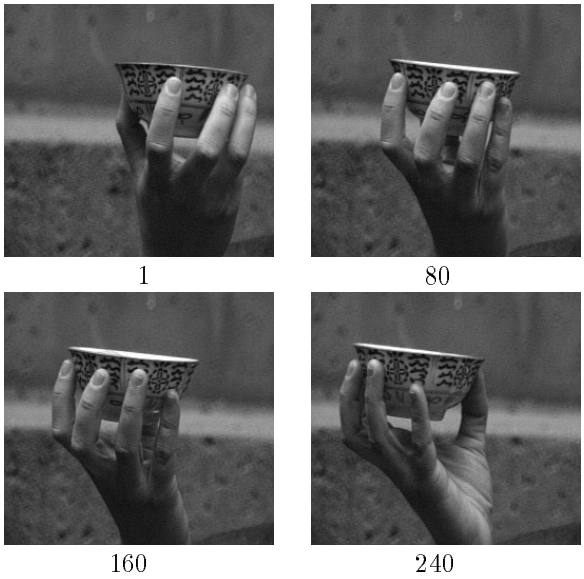

Figure 2: Four out of the 240 frames of the cup image stream.

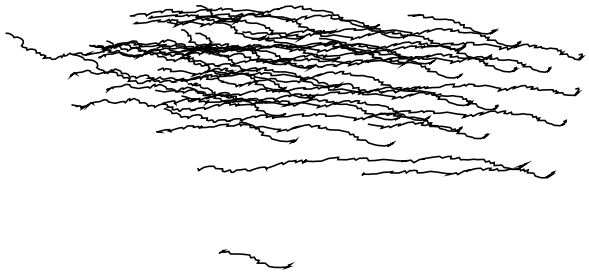

Figure 3: Tracks of 60 randomly selected features from the cup stream.

3 Egomotion from Image Deformations

The second case study concerns the computation of camera motion from a pair of narrowly spaced images. Redundancy is guaranteed by using typically tens or hundreds of features, and the quantities sacrificed to observability are the rotation of the camera and the depths of the feature points.

The input to most methods for computing the direction of camera motion from two successive frames is the motion of points in the image. This motion depends

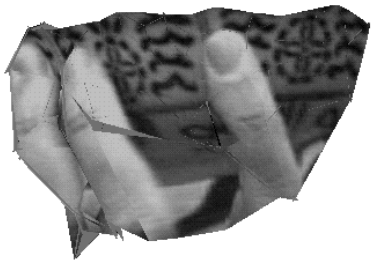

Figure 4: A front view of the cup and fingers, with the original image intensities mapped onto the resulting surface.

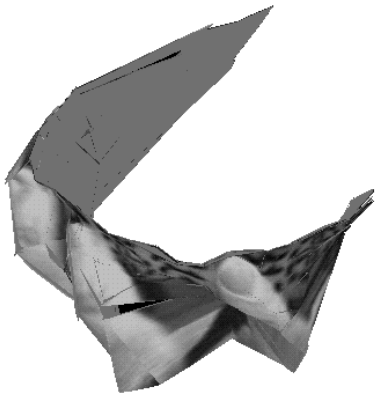

Figure 5: A view from above of the cup and fingers.

on both the camera rotation and the camera translation. However, rotation is both perceptually uninteresting and computationally harmful. It is uninteresting because it does not change the viewpoint and therefore it does not help undoing perspective projection. It is harmful, because it makes translation much harder to estimate: telling how much of a point's motion in the image is caused by translation and how much by rotation is a typically ill-conditioned problem.

To circumvent this difficulty, we proposed [18] to measure image *deformation* instead of image motion. Given two points in the world, the change in the angle formed by their projection rays as the camera moves is a deformation. More specifically, consider two points $\mathbf{P}$ and $\mathbf{Q}$ in space, as in figure 6. As the viewer moves from $\mathbf{C}$ to $\mathbf{C}'$, the magnitude α of the angle $\mathbf{PCQ}$ formed by the projection rays changes to $\mathbf{PC}'\mathbf{Q}$. The *image deformation* is measured for this problem by $\dot{\alpha}$, the time derivative of α. The angle α is given by

$$\alpha = \arccos(\mathbf{p}^T \mathbf{q}) \tag{1}$$

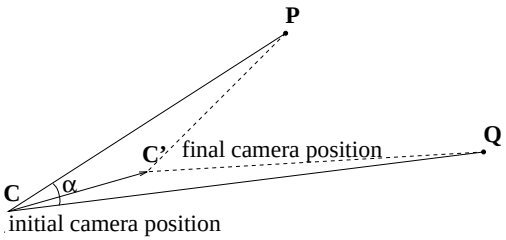

Figure 6: As the viewer moves, the angle α between projection rays $\mathbf{CP}$,$\mathbf{CQ}$ varies. This variation is the *image deformation*.

where $\mathbf{p}$ and $\mathbf{q}$ are two unit vectors from the viewer center to the points $\mathbf{P}$ and $\mathbf{Q}$. Taking the derivative of α with respect to time yields the following measurement equation:

$$b = \mathbf{t}^T A \mathbf{d} \tag{2}$$

where the scalar

$$b = \frac{\dot{\alpha}}{\sin \alpha} \tag{3}$$

is a quantity that can be measured from two or more images, the vector

$$\mathbf{d} = \begin{bmatrix} |\mathbf{P}|^{-1} \\ |\mathbf{Q}|^{-1} \end{bmatrix}$$

collects the reciprocals of the two unknown depth values, the 3×2 matrix A is a known function of image position, and the vector $\mathbf{t}$ is the unknown camera translation velocity.

Clearly, rotation does not affect deformation, since the center of projection does not change with a rotation of the camera around it; the image deforms if and only if the camera center translates. In other words, rather than measuring how the image points *move* in the field of view, the traditional approach, we measure how the image *deforms* over time.

Under perfect arithmetic and noiseless images, the distinction between image motion and image deformation is immaterial, since deformation is computed from point positions in successive images, that is, from image motion. In reality, however, when noise mars the images and finite arithmetic makes computations approximate, using deformations can turn a failure into a success. In fact, the camera's direction of heading is computed from deformation by minimizing a residual function that has a deeper minimum than the one based on image motion. Figure 7 shows that even very small rotations flatten the minimum of the motion-based residual function considerably, leading to a minimization that is more sensitive to noise.

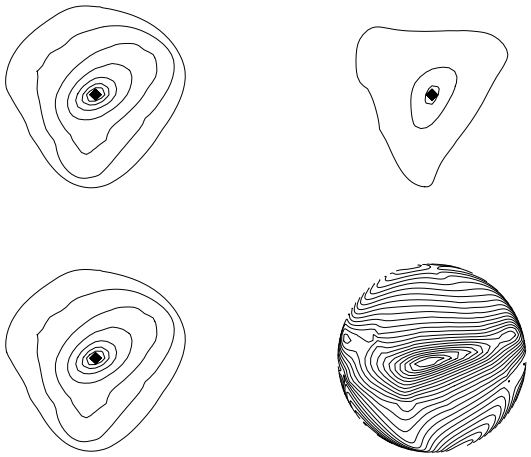

Figure 7: The deformation-based residual (left plots) and the traditional motion-based residual (right) for pure translation (top) and with an added rotation of one degree (bottom).

Based on this idea, we developed a method to compute the camera's direction of heading from image deformations. The method determines the angular deformations of the edges of a Delaunay triangulation over a set of automatically extracted feature points (see figure 8), and equation (2) is repeated once per edge in the triangulation. The depths of the points in space, poorly constrained by the measurements, are eliminated to define a residue function that depends only on the camera's direction of heading. This function is shown for two subsequent frame pairs in figure 9, where global minima are in the first quadrant, local minima in the third. The global minima correspond to the correct direction of heading, and are found by using a variation of Golub and Pereira's variable projection method [5], [16] that we extended to nonconvex functions [18].

The computational advantages of our method are substantial. First, motion is reduced to the two degrees of freedom of interest, rather than the usual five, thereby removing motion's poorly observable components and leading to a more stable solution. Second, the particular minimization technique guarantees graceful degradation of the solution with increasing image noise (see figure 10). Again, using redundant data and restricting the computation to observable quantities proved to be the winning strategy.

Figure 8: Deformations are computed for the feature pairs connected by the edges of a Delaunay triangulation.

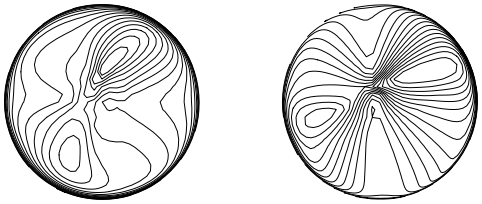

Figure 9: Contour plots of the residue for the computation of egomotion. Global minima are in the first quadrant, local minima in the third.

4 Structure and Motion from Perspective Images

In the introduction, we saw that the sensitivity of the reconstruction process forces us to face the task of computing structure and motion from image sequences in its entirety, not a few images and features at a time. At the same time, however, we need to understand the landscape of the solution space. In fact, the computation is nonlinear, and we must steer our minimization engine toward the global minimum. The main contribution of this line of research [19] is a new description of image sequences that achieves both goals. The proposed representation considers at once *all* the possible images of a given set of feature points. In this representation, the locus of all possible images of a fixed set of features in the world is a three-dimensional locus, called the *picture locus*, in a space with roughly as many dimensions

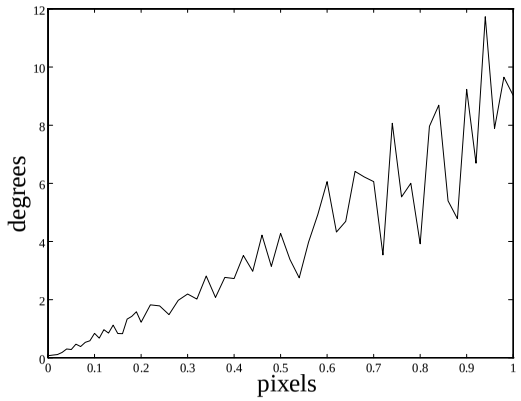

Figure 10: Direction of heading error versus feature position uncertainty.

as there are features in the set. The images in a specific sequence are then points on the picture locus.

Conversely, one can determine a *trail locus* by instead fixing a number of camera positions and collecting the images of a single feature in the world. For each feature in the world, the image measurements from the given camera positions represent the trail that that feature left in the images as the camera covered those positions over time. When the world feature is changed, the trail vector moves on the trail locus. The notions of trail and picture locus turn out to be dual of each other, and the method used to determine structure from the picture locus can be used to compute motion from the trail locus.

The subspaces tangent to the two loci at the origin are shown to represent all the pictures and trails that would be obtained if orthographic projection were used instead of perspective. This link with orthographic projection is the key to a practical solution of the global minimization problem mentioned above. In fact, the introduction of the picture and trail loci and their tangent subspaces allows splitting reconstruction into two phases: the minimization of two *convex* functions in two large spaces, followed by the minimization of a nonconvex function in a *small* space. The first stage yields world structure and camera motion up to an affine transformation, while the second computes the correct Euclidean metric. Working in a small space is crucial to the feasibility of the nonconvex minimization stage. The number of unknowns can be reduced by using the image *deformations* introduced in section 3.

To make visualization easier, we consider only a flat, two-dimensional world, in which images are single scanlines, but the concepts hold also in three dimensions. Measurements from one image are collected in a vector (figure 11). As the observer moves, this vector traces a trajectory on a surface of the third degree in a space of as many dimensions as there are measurements in one image. Thus, the picture locus becomes a picture *surface*.

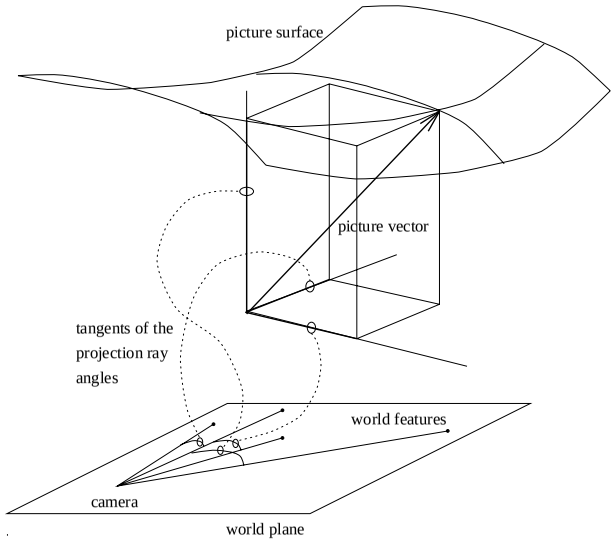

Figure 11: The components of a point on the picture surface (a picture vector) are the tangents of the projection ray angles.

The picture surface depends only on the position of the features in the world, and different camera motions generate different trajectories on the same picture surface. Similarly, one can determine a *trail surface* by instead fixing a number of camera positions and collecting into a trail vector the images of a feature moving around in the planar world.

More specifically, on the plane, pairs of features can be formed easily by keeping one feature fixed and varying the second feature in the pair. Thus, one feature serves as a landmark and the image positions of the other features are specified by the angles between their projection rays and that of the landmark feature. The tangent t of each angle is given by

$$t = \frac{uz - wx}{1 - ux - wz} \; . \tag{4}$$

where (x, z) is the position of the feature in the world and

$$K = (u, w) = C/|C|^2 \tag{5}$$

is the vector obtained by reflecting the camera coordinates C across the unit circle. Notice that this equation

is symmetric in structure and motion: the equation does not change with the replacements

$$u \quad \leftrightarrow \quad x \tag{6}$$

$$w \quad \leftrightarrow \quad -z \tag{7}$$

Consequently, if we find a method to recover structure, we can use a similar method to recover motion as well.

With $P+1$ world feature points, an image from reflected camera position $K = (u, w)$ yields a set of P measurements $t_1, \ldots, t_P$:

$$t_p = \frac{uz_p - wx_p}{1 - ux_p - wz_p} \tag{8}$$

that can be collected into one vector $\mathbf{t} = (t_1, \ldots, t_P)$. This vector can be viewed as a point in a P-dimensional space. As the camera moves, the point $\mathbf{t}$ moves within this space. The locus of all possible points $\mathbf{t}$ for a fixed set of world features is a surface, traced by the parameters u, w and whose P components are given in parametric form by equation (8). This surface is called the *picture surface*. Notice that the picture surface does not depend on camera position, since it represents the images of the given features from all possible camera positions.

As an example, figure 12 shows a region of the picture surface for the four features $S_0 = (0, 0)$, $S_1 = (0, 4, 0.8)$, $S_2 = (0.7, 0.1)$, $S_3 = (0.2, 0.5)$ of figure 12 when the camera moves in the region defined by the rectangle with vertices $K_0 = (-1, -1)$ and $K_1 = (-1, -0.5)$ in the K plane, corresponding to camera positions C on the grid in figure 13. The grid of camera positions in figure 13 is in one-to-one correspondence with the grid on the picture surface of figure 12. Surfaces for more features cannot be visualized (except by projecting them to subspaces), but are still two-dimensional objects, because they are traced by two parameters.

Because of the symmetry in motion and structure, the surface of figure 12 can also be interpreted as a trail surface. Mathematically, this corresponds to fixing the camera positions in equation (4) rather than the world features, as done in equation (8). This yields figure 14, where now circles represent camera positions and the grid points are the varying position of a feature in the world.

Given a set of image measurements of $P+1$ feature points in F image frames, the picture surface can be found by linear fitting by determining the coefficients of a system of $P-2$ equations in the P coordinates $t_1, \ldots, t_P$ of the image vector $\mathbf{t}$. The harder problem

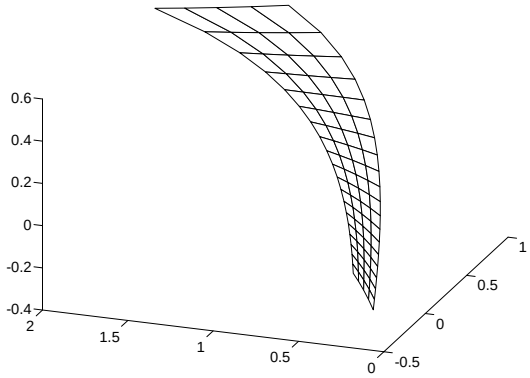

Figure 12: The picture surface for the four features in figure 13. The patch displayed here corresponds to the camera positions shown in figure 13.

is then to find the coordinates of the world features corresponding to a given picture surface. In fact, the coefficients of the system mentioned above are a complicated function of the coordinates x_p, z_p of the world features. The brute-force approach to this problem leads to a nonlinear constrained minimization problem of difficult solution. To avoid this problem, we now introduce an important result about the picture surface (see [19] for a proof).

> **Theorem (Orthographic Plane)** *The plane tangent to the picture (trail) surface at the origin represents all the images of the same world features (the trails from the same camera positions) under orthography, up to a scale factor.*

This theorem is important because any two distinct orthographic images of a given set of features are the x and z coordinates of the features in the world except only for an affine transformation. In other words, we just need to pick any two points (not colinear with the origin) on the orthographic plane to obtain structure up to an affine transformation. Thus, structure and motion can be computed from a sequence of images as follows:

1. Collect all image measurements into one matrix, with one picture per row and one trail per column.
2. Fit a picture surface to the rows of the matrix and a trail surface to its columns.
3. Any two points on the tangent planes to the two

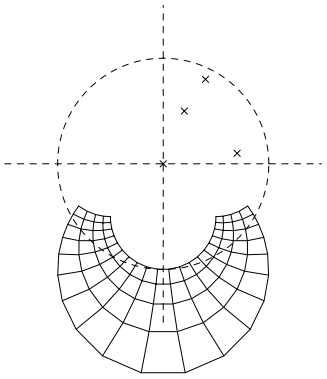

Figure 13: When the inverse camera coordinates K defined in equation (5) vary in the rectangle with vertices $K_0 = (-1, -1)$ and $K_1 = (-1, -0.5)$, the camera positions C move on the grid in this figure. The four crosses represent four features in the world, with the point at the origin being the reference feature.

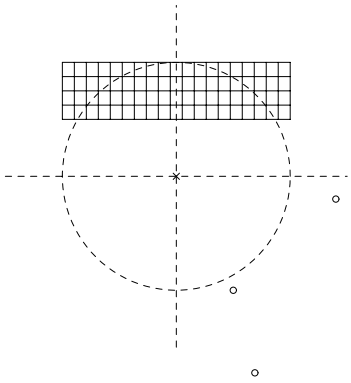

Figure 14: The surface of figure 12 can also be interpreted as the *trail* surface of the situation in this figure. The reference feature is still at the origin (cross).

surfaces represent structure and motion up to two different affine transformations.

4. Replace the affine coordinates into the perspective projection equations to compute the parameters of the affine transformations, thus determining the correct Euclidean metric for structure and motion, up to a scale factor. This last minimization step is nonlinear, but occurs in the small space of transformation parameters.

Thus, a linear stage for affine structure is followed by a nonlinear stage to determine the Euclidean metric. Because of this, the proposed method can be seen on one hand as a successor of techniques based on essential matrices pioneered by Longuet-Higgins [11], independently reinvented by Tsai and Huang [22] and surveyed in [13]; and on the other hand it is a successor of the factorization method described in section 2. However, essential matrices work on two frames at a time, thereby either introducing a hard correspondence problem when the two frames are distant or leading to a poorly conditioned reconstruction when they are close. The multiframe factorization method, on the other hand, works only under orthographic projection, which limits its applicability to distant scenes and narrow fields of view. The current method, in contrast, is multiframe, multifeature, and works for perspective images. In addition, in contrast to multiframe and multifeature *local* methods such as [17], our method is global, in that it does not require an initial estimate of structure or motion.

Figure 15 shows the result of a simulation with noisy images. Both true and computed structure and motion are shown. Noise on the image feature coordinates is Gaussian with a standard deviation of 0.5 pixels. In the simulation, both features and camera positions are scattered randomly, each in one quadrant of the plane. The two points at the origin and along the positive horizontal axis (at $(1, 0)$) are the reference points, and their computed values are therefore exact.

The two plots in figure 16 show the structure and motion errors for increasing levels of noise. Ten features and camera positions are used in all experiments, and each setting is repeated ten times with different random samples to produce ensemble averages. Structure errors are measured as the ratio between the average error per feature and the size of the bounding box of the true feature positions. A similar measure is used for the camera position errors.

Even with relatively few points and viewing positions, performance is good for subpixel noise levels. When the standard deviation of noise increases beyond one pixel, performance degrades sharply but continuously. We point out that in feature tracking the position of features can usually be determined with an accuracy of 0.1 or so pixels [20] for typical 512 by 512 images. From the plots of figure 16 we see that the corresponding structure and motion errors are a fraction of one percent.

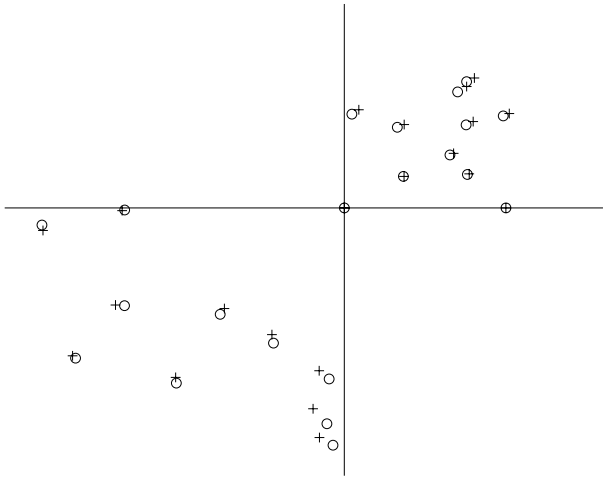

Figure 15: True (circles) and computed (crosses) structure and motion with simulated data. Camera positions are in the lower-left quadrant, feature points in the upper-right one.

Work is in progress to evaluate the method on real image sequences.

Conceptually, our method is an extension to the factorization method presented in section 2: the planes spanned by the rows and columns of the measurement matrix are now replaced by two surfaces, called the *picture surface* and the *trail surface*, which become three-dimensional algebraic varieties in the three-dimensional case.

Besides yielding a method for the computation of structure and motion from image sequences, the geometric characterization of image sequences presented here provides insight and understanding. It captures the essence of the redundancy inherent in a sequence of images, suggests computational techniques, and establishes a clear and useful relation between perspective and orthographic projection. The requirement of observable outputs is fulfilled by the use of deformations instead of feature positions, which eliminates camera rotation from the model.

5 Conclusion

The three case studies presented in this paper demonstrate the importance of input redundancy and output observability in the analysis of visual motion. The success achieved by systems that compute structure from *known* motion led initially to believe that the same techniques could be applied without major changes to the simultaneous estimation of structure and motion. Although some results have been obtained [7],[8], their accuracy and stability needed further attention. Two main difficulties can be identified in this task. First, the computation of structure and motion is inherently sensitive to noise. Being a nonlinear computation, techniques that worked well for known motion are not directly applicable. Second, some combinations of the output quantities are poorly observable. Most notably, small variations of large depth values have often effects on the images that are comparable or even smaller than the positional accuracy of the best feature trackers available.

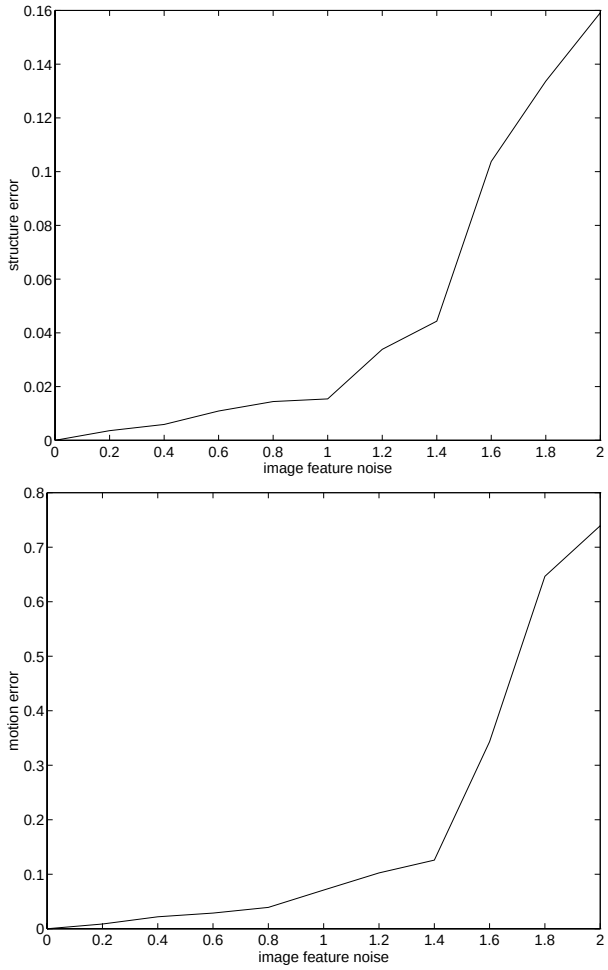

Figure 16: Errors in the computed structure (top) and motion (bottom) for increasing levels of image feature noise, measured in pixels for a 512 by 512 image. See text for the units of the vertical axes.

Also, small rotations and translations can produce mutually similar image changes, and are therefore hard to tell apart. Because of these difficulties, one cannot hope to write equations that hold in the absence of noise and solve them blindly when their parameters come from real images. On the contrary, special attention must be paid to the sensitivity of the computation to errors in the input measurements. On the one hand, the input should be redundant to minimize the effects of noise. On the other, the output should be observable if it is to be computed reliably from the data. Approaches based on the picture and trail surfaces, of which factorization turns out to be a special case, are a good way to handle large amounts of images and features in a uniform notation, and provide a fitting set of conceptual and computational tools for the exploitation of redundancy. Better observable outputs, on the other hand, can be obtained by eliminating camera rotation whenever possible, by the introduction of image deformations, and feature depths whenever they are large with respect to the overall translation of the camera.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Pattern Analysis and Machine Intelligence*, 7:384–401, 1985.

[2] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[3] T.J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, July 1990.

[4] A. R. Bruss and B. K. P. Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21:3–20, 1983.

[5] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, April 1973.

[6] D. J. Heeger and A. Jepson. Visual perception of three-dimensional motion. Technical Report 124, MIT Media Laboratory, Cambridge, Ma, December 1989.

[7] J. Heel. Dynamic motion vision. In *Proceedings of the DARPA Image Understanding Workshop*, pages 702–713, Palo Alto, Ca, May 23-26 1989.

[8] J. Heel. Direct estimation of structure and motion from multiple frames. AI Memo 1190, MIT, Cambridge, MA, March 1990.

[9] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, July 1988.

[10] R. E. Kalman, P. L. Falb, and M. A. Arbib. *Topics in Mathematical System Theory*. McGraw-Hill Book Co., New York, NY, 1969.

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

[12] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–236, September 1989.

[13] S. Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag, Berlin Heidelberg, 1993.

[14] K. Prazdny. Egomotion and relative depth from optical flow. *Biological Cybernetics*, 102:87–102, 1980.

[15] J. W. Roach and J. K. Aggarwal. Computer tracking of objects moving in space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):127–135, April 1979.

[16] A. Ruhe and P. Å. Wedin. Algorithms for separable nonlinear least squares problems. *SIAM Review*, 22(3):318–337, July 1980.

[17] M. E. Spetsakis and J. (Yiannis) Aloimonos. Optimal motion estimation. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 229–237, Irvine, California, March 1989.

[18] C. Tomasi and J. Shi. Direction of heading from image deformations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR93)*, pages 422–427, New York, NY, June 1993.

[19] C. Tomasi. The geometry of rigid visual motion under perspective projection: the planar case. TR in preparation, Cornell University, Ithaca, NY, 1993.

[20] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method - 3. detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, Pittsburgh, PA, April 1991.

[21] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal on Computer Vision*, 9(2):137–154, 1992.

[22] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):13–27, January 1984.

[23] S. Ullman. *The Interpretation of Visual Motion*. The MIT Press, Cambridge, Ma, 1979.

[24] A. M. Waxman and K. Wohn. Contour evolution, neighborhood deformation, and global image flow: planar surfaces in motion. *International Journal of Robotics Research*, 4:95–108, 1985.