# Shape and Motion from Image Streams: a Factorization Method

## 1. Planar Motion

Carlo Tomasi Takeo Kanade

September 1990
CMU-CS-90-166

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was supported by the Defense Advanced Research Projects Agency (DOD) and monitored by the Avionics Laboratory, Air Force Wright Aeronautical Laboratories, Aeronautical Systems Division (AFSC), Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-87-C-1499, ARPA Order No. 4976, Amendment 20.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA or the U.S. government.

**Keywords:** computer vision, motion, shape, time-varying imagery

## Abstract

Inferring the depth and shape of remote objects and the complete camera motion from a stream of images is possible in principle, but is an ill-conditioned problem when the objects are distant with respect to their size.

To overcome this difficulty, we have developed a factorization method to decompose an image stream directly into object shape and camera motion, without computing depth as an intermediate step.

The factorization method is explored in a series of technical reports, going from basic principles through implementation. This is the first report in the series, and presents the basic concepts in the case of planar motion, in which images are single scanlines.

In this situation, an image stream can be represented by the $F \times P$ matrix of the image coordinates of $P$ points tracked through $F$ frames. We show that under orthographic projection this measurement matrix is of rank 3.

Using this observation, we develop an algorithm to recover shape and camera motion, based on the singular value decomposition of the measurement matrix.

Noise is defeated by applying a well-conditioned computation to the highly redundant input represented by an image stream. No assumptions are made about smoothness or regularity of the camera motion, and even sudden jumps in the camera velocity are faithfully reproduced in the computed output.

# Preface

In principle, the stream of images produced by a moving camera allows the recovery of both the shape of the objects in the field of view, and the motion of the camera. Traditional algorithms recover depth by triangulation, and compute shape by taking differences between depth values. This process, however, becomes very sensitive to noise as soon as the scene is more than a few focal lengths away from the camera. Furthermore, if the camera displacements are small, it is hard to distinguish the effects of rotation from those of translation: motion estimates are unreliable, and the quality of the shape results deteriorates even further.

To overcome these problems, we have developed a factorization method to decompose an image stream directly into object shape and camera motion, without computing depth as an intermediate step. The method uses a large number of frames and feature points to reduce sensitivity to noise. It is based on the fact that the incidence relations among projection rays can be expressed as the degeneracy of a matrix that gathers all the image measurements.

To explore this new method, we designed a series of eleven technical reports, as shown in figure 1, going from basic theory to implementation.

This first report illustrates the idea in the case of planar motion, in which images are single scanlines. We introduce the factorization method, and test a complete algorithm on a real image stream.

Report number 2 extends the idea to three-dimensional camera motion and full image streams. It assumes that point features can be tracked over several image frames. Report number 3 describes how to extract and track point features.

If point features are too sparse to give sufficient shape information, line features can be used either instead or in addition, as discussed in report number 4. Report number 5 shows how to extract and track line features.

The performance of our shape-and-motion algorithm is rather atypical. Because it does away with depth and capitalizes on the diversity of viewpoints made

possible by long image streams, it performs best when the scene is distant and the motion of the camera is complex. Report number 6 examines what happens when objects are close to the camera, and perspective foreshortening occurs. Report number 7 shows how to deal with degenerate types of motion.

Occlusion can be handled by our method, and is treated in report number 8.

A basic assumption of our shape-and-motion algorithm is that only the camera moves. In some cases, however, a few points move in space with respect to the others, for instance, due to reflections from a shiny surface. Report number 9 examines how to detect these cases of spurious motion.

Our factorization algorithm deals with the whole stream of images at once. For some applications this is undesirable. Report number 10 proposes an implementation that can work with an indefinitely long stream of images.

Report number 11 considers a more radical departure from the assumption of a static scene than spurious motion. If several bodies are moving independently in the field of view of the camera, our factorization method can be used to count the number of moving bodies.

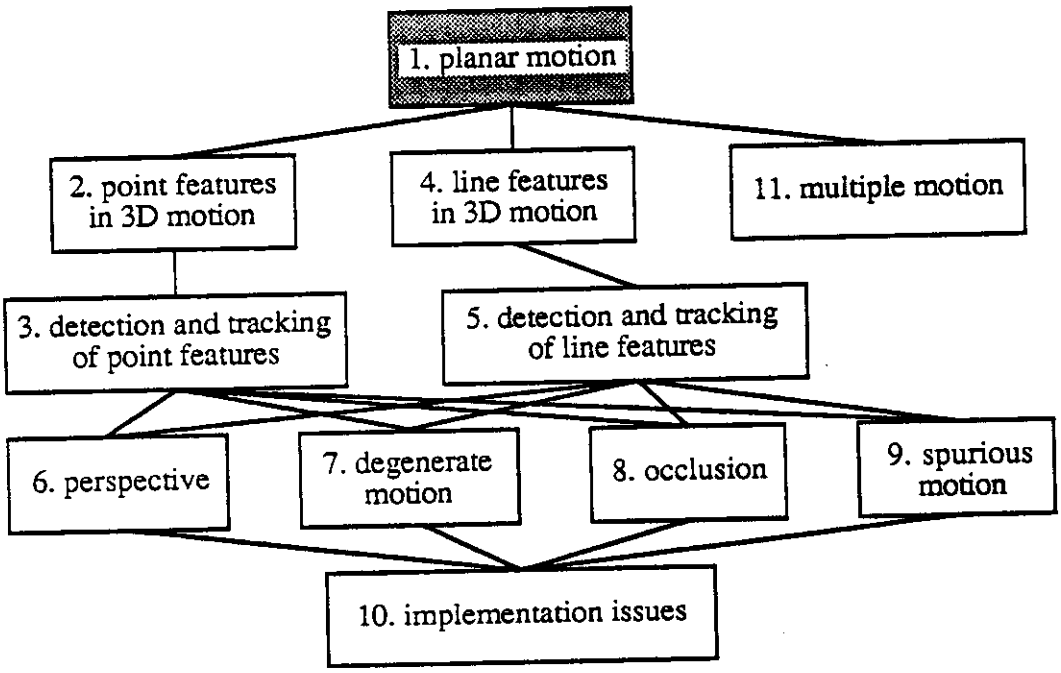


Figure 1: The technical report in the series. Arcs suggest reading paths.

# Contents

# Chapter 1

# Introduction

In principle, the shape of an object can be computed from a stream of images by first estimating camera motion and depth, and then inferring shape 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 image changes that it produces are small. Second, shape 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 shape directly from variations in the relative position of image features, without computing depth as an intermediate step.

In this report, we show that shape and camera rotation can be inferred precisely from many features and frames, without assuming any model for the motion, and reduce the computation to decomposing a matrix of image measurements.

The resulting algorithm, tested in simple situations, gives remarkably precise motion and shape estimates, without introducing smoothing effects into the result.

For simplicity, we will limit our consideration to one epipolar plane at a time, and assume that motion occurs in that plane. In other words, our images are single scanlines.

Our theory is based on the observation that the incidence relations among projection rays can be expressed as the degeneracy of a matrix that gathers all the image measurements. To our knowledge, this observation has not previously appeared in the literature.

Since we use many, closely spaced frames, the results are insensitive to noise,

and the correspondence problem is simplified. Previous multi-frame approaches usually assume a motion model to combine estimates of the camera position over many frames. Typically, this model is some form of motion smoothness. In our method, on the other hand, we assume only the invariant of shape constancy over time.

As an illustration of our theory, we used our algorithm to recover the shape of a one-dollar silver coin (about 4 cm in diameter) placed at 3.5 meters from a real moving camera with a long lens. The total rotation of the camera was 30 degrees around the coin (and in the midplane of the coin). The error in the computed angle of camera rotation was always less than a tenth of one degree, and usually substantially smaller. The error in the shape of the coin was always less than 1.5 percent of its diameter, and typically considerably smaller. The small errors due to the effect of perspective are also analyzed.

In the following, we introduce our scenario, summarize the results, and sketch the relations of our work with previous literature on the subject. Chapter2 introduces the degeneracy principle mentioned above. Chapter3 shows how to use it to decompose the measurement matrix into shape and camera rotation. The experimental results in Chapter4 show the ability of the algorithm to deal with jerky rotations without smoothing its output. The conclusion (Chapter5) compares direct shape algorithms with algorithms that base the computation of shape on that of depth, and shows the former ones to be superior for remote scenes.

# The Scenario

The world is still, and the camera moves in a plane, where it can freely rotate and/or translate. $P$ feature points, far away from the camera, are visible in a given scanline, parallel to the plane of motion. Since the frames are taken frequently, it is easy to track the features from frame to frame. As the camera moves, it is panned so as to keep the features in the field of view.

After $F$ frames, an $F \times P$ matrix $U$ of image measurements is available. This matrix is the input to the algorithm.

This scenario approximates what happens with a camera on an airplane, with suitable control mechanisms to align the camera scanlines with the direction of flight, and to keep the same object within the field of view. Because objects are distant from the camera, we can assume orthographic projection.

## The Results

This report first shows that if the measurements are noise-free, the image coordinate matrix $U$ is highly degenerate: its rank is 3. As a result, $U$ can be decomposed into the product of two smaller matrices: an $F \times 3$ matrix that encodes the $F$ camera positions, and a $P \times 3$ matrix that encodes the positions of the $P$ world points.

When noise corrupts the measurements, the rank of $U$ can be defined in an approximate sense, and is still 3.

The noisy matrix $U$ is factored by Singular Value Decomposition [Golub and Reinsch, 1971], which is known to be efficient and numerically well behaved. If more points and frames are used than prescribed by equation-counting arguments (which require a minimum of three points and three frames), the effects of noise can be reduced.

The resulting shape and motion algorithm is simple and efficient, and has been implemented and tested on objects as distant as one hundred times their size (see Chapter4). The rotation errors are always smaller than one tenth of a degree. The relative precision in the computed shape is of the order of the *relative depth range*, defined as the ratio between the size of the object and its distance from the camera.

The good performance of our algorithm derives from the fact that shape is obtained directly, without using depth as an intermediate result. In traditional approaches, depth is first computed by triangulation. For remote objects, the quality of depth estimates by triangulation is very sensitive to noise, and degrades as the realtive range decreases. Consequently, the shape estimates degrade even faster, since the computation of shape from depth is itself ill-conditioned.

In our approach, instead, no triangulation is done. Depth becomes irrelevant, and the results are highly accurate.

## Relations with Previous Work

Our goal is to compute world point coordinates, relative to each other, and camera motion from multiple image frames.

Our algorithm does what photogrammetrists for more than thirty years have done by hand and with two frames at a time [Thompson, 1959]. Ullman proposed an automated solution to this problem eleven years ago [Ullman, 1979], and called it *structure-from-motion*. He also considered only two frames at once, and as few points as theoretically possible.

Most of the initial efforts in this area have been devoted to finding closed-form solutions with a minimal or nearly-minimal number of points and/or frames (see, for instance, [Longuet-Higgins, 1981]).

In general, structure-from-motion is hard to solve. The major difficulty is the inherent sensitivity of shape and motion to noise in the image, especially when objects are distant. If depth is explicitly represented as an intermediate stage in the computation, performance degrades with reductions in the relative depth range. For instance, the algorithm presented in [Tsai and Huang, 1984] works very well for close objects (which is the intended goal of that algorithm), but the performance is likely to degrade when objects become more remote, and the relative depth range becomes smaller.

The remedy is to by-pass the computation of depth, as we do in this report, to remove the main cause of ill-conditioning.

Even with a well-conditioned algorithm, however, noise degrades performance. Few points and/or few frames give bad results, regardless of how good the math is. Our algorithm allows using many frames and many points, thus exploiting redundancy to counteract noise. If frames are closely spaced, the correspondence problem is also made easier to solve.

Many, tightly spaced frames have been used in [Bolles *et al.*, 1987] and [Matthies *et al.*, 1989], but only for the inference of depth when the motion of the camera is known. Determining shape and motion simultaneously, on the other hand, has been often suspected of being practically infeasible.

In [Spetsakis and Aloimonos, 1989], an interesting algorithm is presented for the case of unknown motion, using several frames and points and a perspective projection model. In spirit, our approach is akin to theirs: the projection lines of the same world point are a bundle (or pencil) of lines, and the resulting incidence relations between them allow casting the computation of shape and motion as a minimization problem. When applied to remote objects, however, their solution suffers from the same ill-conditioning problem discussed above, since depth is explicitly represented in their model.

# Chapter 2

# The Decomposition Principle

This chapterintroduces the fundamental principle on which our shape-and-motion algorithm is based: the $F \times P$ matrix of the image coordinates of $P$ points tracked through $F$ frames is higly rank-deficient.

As we stated in the introduction, we consider only one scanline per frame, and assume that the camera moves in a plane parallel to the scanline. In this plane, we define an arbitrary orthogonal system of coordinates $(X, Z)$.

The images are orthographic projections of $P$ points, tracked through $F$ frames. The measurements $u_{fp}$ can then be collected in an $F \times P$ matrix

$$U = \begin{bmatrix} u_{11} & \cdots & u_{1P} \\ \vdots & & \vdots \\ u_{F1} & \cdots & u_{FP} \end{bmatrix} .$$

From figure 3.1 we see that the projection $u_{fp}$ of point $p$ onto frame $f$ is given by the equation

$$u_{fp} = c_f X_p + s_f Z_p + t_f \ , \tag{2.1}$$

where $c_f$ and $s_f$ are the cosine and sine of the angle $\alpha_f$ that frame $f$ forms with the $X$ axis. The scalar $t_f$ is the projection onto the $f$-th image of the vector that joins the world origin with the origin of the $f$-th frame.

We can now collect all of the $F \times P$ equations (2.1) in matrix form:

$$U = MS \tag{2.2}$$

where

$$M = \begin{bmatrix} c_1 & s_1 & t_1 \\ \vdots & & \vdots \\ c_F & s_F & t_F \end{bmatrix} \tag{2.3}$$

is the motion matrix, and

$$S = \begin{bmatrix} X_1 & \cdots & X_P \\ Z_1 & \cdots & Z_P \\ 1 & \cdots & 1 \end{bmatrix} \tag{2.4}$$

is the shape matrix.

Since $M$ is $F \times 3$ and $S$ is $3 \times P$, we have just proven the following fact.

# The Rank Principle

*Without noise, the rank of the measurement matrix $U$ is at most three.* [1]

Appendix A discusses the degenerate cases in which the rank of $U$ is even smaller than three. These degeneracies correspond to all-aligned points or to special types of motion. They can always be detected, and treated as special cases. Consequently, we can simplify our treatment and assume that the rank principle is satisfied in a strong sense: the rank of $U$ is exactly three.

Intuitively, the rank principle expresses the simple fact that the $F \times P$ image measurements are redundant. Indeed, they could all be described more concisely by giving $F$ frame angles and $P$ points, if only these were known.

Geometrically, the rank principle expresses an incidence property. In fact, if we replace $X_p$ and $Z_p$ in the projection equation (2.1) by the generic coordinates $X$ and $Z$, we obtain the equation of the projection line of point $p$ onto frame $f$:

$$u_{fp} = c_f X + s_f Z + t_f \ .$$

Equation (2.1) and, equivalently, the rank principle, say that there is a point that belongs to these lines for all values of $f$. In other words, the projection lines of a given point form a pencil.

In the next chapter, we show how to use the rank principle to determine the motion and shape matrices $M$ and $S$.

[1] In [Tomasi and Kanade, 1990], all image coordinates were measured with respect to those of a reference feature. In that case, $t_f$ was always zero, so the rank of the measurement matrix was two.

# Chapter 3

# The Algorithm

When noise corrupts the images, the measurement matrix $U$ will not be exactly of rank 3. However, the rank principle can be extended to the case of noisy measurements in a well-defined manner. Section3.1 introduces this extension, using the concept of Singular Value Decomposition (SVD) [Golub and Reinsch, 1971] to introduce the notion of approximate rank.

However, although the rank principle is the key to our algorithm, it is not the whole story. In Section3.2, we show that, based on the rank principle, the matrices $M$ and $S$ are determined only up to an arbitrary affine warping of the plane. Therefore, in Section3.2 we also point out the additional constraints needed to complete the solution.

Section3.3 outlines the complete shape-and-motion algorithm.

## 3.1 Approximate Rank

Assuming [1] that $F \geq P$, the matrix $U$ can be decomposed [Golub and Reinsch, 1971] into an $F \times P$ matrix $L$, a diagonal $P \times P$ matrix $\Sigma$, and a $P \times P$ matrix $R$,

$$U = L\Sigma R \,, \tag{3.1}$$

such that

$$\begin{aligned} L^T L = R^T R = RR^T &= I \\ \sigma_1 \geq \ldots &\geq \sigma_P \,. \end{aligned}$$

[1] This assumption is not crucial: if $F < P$, everything can be repeated for the transpose of $U$.

Here, $I$ is the $P \times P$ identity matrix, and the *singular values* $\sigma_1, \ldots, \sigma_P$ are the diagonal entries of $\Sigma$. This is called the *Singular Value Decomposition* (SVD) of the matrix $U$.

We can now restate our key point.

## The Rank Principle for Noisy Measurements

*The first three singular values of the noisy measurement matrix $U$ are much greater than the others:*

$$\sigma_1 \, , \, \sigma_2 \, , \, \sigma_3 \gg \sigma_4, \ldots, \sigma_P \, . \tag{3.2}$$

It can be shown [Forsythe *et al.*, 1977] that the rank-3 matrix $U^*$ that is closest to $U$ in the $L_2$-norm sense can be obtained by setting to zero all the singular values after the third in the decomposition:

$$U^* = L^* \Sigma^* R^* \, , \tag{3.3}$$

where $L^*$ collects the first three columns of $L$, $\Sigma^*$ is the first third-order principal minor of $\Sigma$, and $R^*$ gathers the first three rows of $R$.

## 3.2 The Metric Constraints

Golub and Reinsch [Golub and Reinsch, 1971] give an efficient and well-behaved algorithm to compute the singular value decomposition of a matrix. We use that algorithm to obtain a decomposition of the measurement matrix $U$.

The singular value decomposition of a matrix is unique because the left and right factors $L$ and $R$ are required to be orthonormal. However, this does not mean that there is only one way to decompose the measurement matrix $U$ into $M$ and $S$. Since the rank principle expresses an incidence relation, it only determines the two matrices $M$ and $S$ up to an affine transformation of the plane. In fact, if $A$ is *any* invertible $3 \times 3$ matrix, the matrices $MA$ and $A^{-1}S$ are also a valid decomposition of $U$, since

$$(MA)(A^{-1}S) = M(AA^{-1})S = MS = U \, .$$

Therefore, if we want to find $M$ and $S$ from the measurement matrix $U$, we need additional constraints. We approach the problem by first decomposing $U$ into

two matrices $\hat{M}$ and $\hat{S}$ of the appropriate sizes via the SVD algorithm. Based on equation (3.3), we can define, for instance, the two matrices

$$\begin{aligned} \hat{M} &= L^*(\Sigma^*)^{1/2} \\ \hat{S} &= (\Sigma^*)^{1/2}R^* . \end{aligned} \tag{3.4}$$

Then, we can complete the solution by finding the matrix $A$ that transforms $\hat{M}$ and $\hat{S}$ into the actual motion and shape matrices $M$ and $S$:

$$\begin{aligned} M &= \hat{M}A^{-1} \\ S &= A\hat{S} . \end{aligned}$$

The matrix $A$ can be found by looking at the structure of the motion and shape matrices. The first and second column of $M$ gather cosines and sines of the frame angles (see equation (2.3)), and must therefore be normalized. Furthermore, the third row of $S$ contains all ones (equation (2.4)). These are *metric* constraints, as opposed to the incidence constraints expressed by the rank principle.

Formally, let us partition $A$ and $A^{-1}$ into rows and columns, respectively:

$$\begin{aligned} A &= \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{bmatrix} \\ A^{-1} &= \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} = \begin{bmatrix} B^T & \mathbf{b}_3 \end{bmatrix} , \end{aligned} \tag{3.5}$$

where $B^T$ gathers the first two columns $\mathbf{b}_1$ and $\mathbf{b}_2$ of $A^{-1}$. Then, the metric constraints above can be written as follows:

$$\begin{aligned} \hat{\mathbf{m}}_f^T B^T B \hat{\mathbf{m}}_f &= 1 \\ \mathbf{a}_3 \hat{\mathbf{s}}_p &= 1 , \end{aligned} \tag{3.6}$$

where $\hat{\mathbf{m}}_f$ and $\hat{\mathbf{s}}_p$ are the $f$-th row and $p$-th column of $\hat{M}$ and $\hat{S}$, respectively. These two equations say that the points $\hat{\mathbf{m}}_f$ are on a cylinder in a three-dimensional space, and that the points $\hat{\mathbf{s}}_p$ are on a plane in a three-dimensional space. The two equations are not independent, since $\mathbf{a}_3$ and $B^T$ are submatrices of $A$ and $A^{-1}$, respectively. If we write out the product of $A$ and $A^{-1}$ as partitioned above, we see that the coupling can be expressed by the following equation:

$$\mathbf{a}_3 B^T = \begin{bmatrix} 0 & 0 \end{bmatrix} . \tag{3.7}$$

Enforcing the pair of equations (3.6) leads to an overconstrained problem, and we can find the cylinder and the plane by data fitting.

In doing this, we encounter two difficulties. First, fitting a cylinder is a non-linear problem. Second, the two fitting problems are coupled through equation (3.7).

However, a well-behaved algorithm for our problem can be found by first determining a good approximation to the solution, and then refining the latter with a numerical function-minimization routine. This two-stage solution of the metric equations has proven to be accurate and robust in our experiments and simulations.

## 3.3 Outline of the Algorithm

The incidence and metric constraints expressed by the rank principle and by the cylinder and plane equations (3.6) are all we need for our algorithm. In conclusion, given an image measurement matrix $U$, the algorithm for computing the motion matrix $M$ and the shape matrix $S$ defined in equations (2.3) and (2.4) can be summarized as follows.

1. Compute the singular value decomposition of $U$:

$$U = L\Sigma R \ .$$

2. Define the initial decomposition of $U$ into two matrices as follows:

$$\begin{aligned} \hat{M} &= L^*(\Sigma^*)^{1/2} \\ \hat{S} &= (\Sigma^*)^{1/2}R^* \ , \end{aligned}$$

   where $L^*$ collects the first three columns of $L$, $\Sigma^*$ is the first third-order principal minor of $\Sigma$, and $R^*$ gathers the first three rows of $R$.

3. Simultaneously fit a cylinder to the rows of $\hat{M}$ and a plane to the columns of $\hat{S}$ by minimizing the error criterion

$$e(\mathbf{a}_3, B) = \sum_{f=1}^{F}(\hat{\mathbf{m}}_f^T B^T B \hat{\mathbf{m}}_f - 1)^2 + \sum_{p=1}^{P}(\mathbf{a}_3\hat{\mathbf{s}}_p - 1)^2$$

   subject to the constraint

$$\mathbf{a}_3 B^T = \begin{bmatrix} 0 & 0 \end{bmatrix} \ .$$

4. Complete the matrix $A$ and its inverse from their submatrices $\mathbf{a}_3$ and $B$ by solving the system

$$AA^{-1} = I .$$

5. Compute the motion matrix $M$ and the shape matrix $S$ as

$$\begin{aligned} M &= \hat{M}A^{-1} \\ S &= A\hat{S} . \end{aligned}$$

The details of the fitting algorithm in step 3 and of the matrix completion of step 4 are described in appendices B and C, respectively.

# Chapter 4

# An Experiment

We implemented the algorithm described in the previous chapter, and applied it on several image streams.

The experiment described in this chapterillustrates the rank principle, demonstrates the good quality of the results, and quantifies the influence of perspective effects on the accuracy of the motion estimates.

The key parameter for the evaluation of performance is the relative depth range, which we defined as the ratio of the object size along the optical axis and the distance between camera and object. In a nutshell, the conclusion drawn from our experiments is that the relative errors in the computed shape are of the same order as the relative depth range. Consequently, modeling inaccuracies that are small with respect to the latter can be ignored.

We put a one-dollar coin (about 4 cm in diameter) approximately 3.5 meters away from a Sony CCD camera with a 300 mm Tokina lens. Thus, the relative depth range was $4/350 \approx 0.011$. Figure 3.2 shows the setup.

The camera was moved in the plane of the coin, so that only the edge of the coin was visible in every frame. The motion was roughly circular around a point in the vicinity of the coin. Only the rotation component was controlled with an accurate positioning mechanism, so that precise ground truth was available for performance evaluation. Translation was such as to keep the coin in the field of view, but was otherwise uncontrolled.

The edge of the coin was approximately aligned with the image scanlines, thus yielding easy-to-track image features (the thin vertical notches on the coin's edge). The first 101 frames were taken in steps of 0.1 degrees between consecutive frames. After that, the velocity was doubled to 0.2 degrees per frame, and 100

more frames were taken. Thus, the overall rotation was 30 degrees. The resulting 201 scanlines are stacked together in figure 3.3, top to bottom. This figure is what is called an epipolar plane in [Bolles *et al.*, 1987].

The image was filtered with a thirteen-tap finite-impulse-response approximation to the Laplacian of a Gaussian, and the 104 zero crossings of the result, shown in figure 3.4, were used as features in the experiment.

The measurement matrix was thus $201 \times 104$ in size. All of the processing, including feature extraction and linking, matrix decomposition, and motion and shape computation, took about three minutes on a VAX 8800.

The rank principle is illustrated graphically by the similarity of figures 3.4 and 3.5. To obtain figure 3.5, we decomposed the matrix $U$ representing the crossings, set to zero all the singular values except the first three, and reconstructed the measurement matrix. Thus, figure 3.5 represents the rank-3 matrix $U^*$ of equation (3.3). The rank principle says that the only differences between figure 3.4 and figure 3.5, under orthography, are due to noise.

The singular values are plotted in figure 3.6; without noise, and if the projection were exactly orthographic, only the first three values would be different from zero.

Figure 3.7 shows the computed and the true rotation. The error is always smaller than one tenth of one degree, and almost everywhere substantially smaller than that. The algorithm assumes no motion models, and does no smoothing. As a result, the sharp change in rotational velocity after frame 100 is faithfully preserved in the motion output.

Figure 3.8 shows the shape results, and the best circular fit to them. The accuracy of shape is of the order of the relative depth range (1 percent), even if variations in depth during the motion of the camera were of the order of the coin size.

In spite of image noise, perspective effects and unmodeled small variations in depth, the quality of both shape and motion results is remarkably good.

To get an idea of how perspective effects influence the accuracy of the results, we tested our algorithm on a stream of simulated, noise-free images similar to those of our coin experiment. A circular object with 10 features is placed at various depths from the camera. For each depth, a pinhole camera moves and rotates by 30 degrees in 30 steps. Figure 3.9 plots the relative error in the total computed rotation as a function of the relative depth range. While algorithms based on depth give worse motion estimates as objects are moved farther away, our algorithm improves (for a constant total rotation angle), because it approximates orthography better and better.

# Chapter 5

# Conclusion: Depth versus Shape Algorithms

The algorithm presented in this report infers the shape of remote objects and the rotation of the camera. It is a *shape* algorithm. It does not compute the depth of the scene.

Algorithms such as the ones described in [Tsai and Huang, 1984], [Heel, 1989], [Spetsakis and Aloimonos, 1989], on the other hand, represent depth explicitly, and compute it from the image stream. They are *depth* algorithms.

Depth algorithms give a more complete answer. They compute all components of motion, up to a scale factor, and the depth information they supply allows, in principle, computing shape as well.

However, depth algorithms do not work if objects are very distant from the camera with respect to their size. When the relative depth range is very small, as for instance in aerial cartography and reconnaissance, the completeness of depth algorithms is not only useless, but harmful. A shape algorithm gives a more stable and accurate answer, because it computes shape and camera rotation directly from image deformations, without using depth as an intermediate step.

Our factorization method is conceptually simple and leads to an accurate algorithm. The remaining technical reports in this series, outlined in the preface, will address the technical problems which are to be solved to make the algorithm into a practical module of a vision system.

# Bibliography

[Bolles *et al.*, 1987]
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.

[Forsythe *et al.*, 1977]
G. E. FORSYTHE, M. MALCOLM, AND C. B. MOLER. *Computer Methods for Mathematical Computations*. Prentice-Hall, Englewood Cliffs, NJ, 1977.

[Golub and Reinsch, 1971]
G. H. GOLUB AND C. REINSCH. Singular value decomposition and least squares solutions, In *Handbook for Automatic Computation*, volume 2, chapter I/10, pages 134–151. Springer Verlag, New York, NY, 1971.

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

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

[Matthies *et al.*, 1989]
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.

[Spetsakis and Aloimonos, 1989]
M. E. SPETSAKIS AND J. Y. ALOIMONOS. Optimal motion estimation. In

*Proceedings of the IEEE Workshop on Visual Motion*, pages 229–237, Irvine, California, March 1989.

[Thompson, 1959]
E. H. THOMPSON. A rational algebraic formulation of the problem of relative orientation. *Photogrammetric Record*, 3(14):152–159, 1959.

[Tomasi and Kanade, 1990]
C. TOMASI AND T. KANADE. Shape and motion without depth. Technical report CMU-CS-90-128, Carnegie Mellon University, Pittsburgh, Pa, May 1990.

[Tsai and Huang, 1984]
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.

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

# Appendix A

# Degeneracies of the Measurement Matrix

If both the scene and the camera motion are sufficiently complex, the measurement matrix $U$ is exactly of rank 3. On the other hand, special object shapes and/or particular types of camera motion can further reduce the rank of $U$.

In this chapterwe show that the object shape is degenerate if and only if all feature points are aligned, and that camera motion is degenerate if and only if it is such that all optical axes pass through the same point.

## Shape degeneracies

We now interpret the determinants

$$\Delta_{fg}^{(pq)} = \det \begin{bmatrix} u_{fp} & u_{fq} \\ u_{gp} & u_{gq} \end{bmatrix}$$

in terms of intrinsic geometric parameters which describe the relative position of the three world points, and of the angles between frames.

It follows immediately from this interpretation that a necessary and sufficient condition for the existence of at least one non-zero determinant of the type above is that there be at least three non-aligned points, and at least two distinct frames.

Since we consider only shape degeneracies, we can set $t_f = 0$ for all $f$. This is equivalent to saying that the camera moves by pure rotation, as shown in figure 3.10.

Let $d_p$ and $\gamma_p$ be the magnitude and phase of the vector which joins the center of rotation, chosen as the world origin, with object point number $p$:

$$\begin{aligned} d_p &= \sqrt{X_p^2 + Z_p^2} \\ \gamma_p &= \arctan_2(Z_p, X_p) \end{aligned}$$

(see figure 3.10).

Here $\arctan_2$ is the two-argument inverse tangent function, which differs from the one-argument function in that it returns the angle in the appropriate quadrant, and has no singularities:

$$\arctan_2(y,x) = \begin{cases} \arctan(y/x) & \text{if } x > 0 \\ \text{sign}(y)(\pi - \arctan|y/x|) & \text{if } x < 0 \\ 0 & \text{if } x = y = 0 \\ \text{sign}(y)\pi/2 & \text{if } x = 0,\ y \neq 0 \end{cases}$$

Furthermore, let $\psi_{fg}$ be the angle between frame $f$ and frame $g$, measured counterclockwise from $f$ to $g$ (figure 3.10), and let $\gamma_{pq} = \gamma_p - \gamma_q$.

Then, if $u_{fp}$ is the projection of point $p$ onto frame $f$, we have

$$\Delta_{fg}^{(pq)} = \det \begin{bmatrix} u_{fp} & u_{fq} \\ u_{gp} & u_{gq} \end{bmatrix} = d_p d_q \sin\gamma_{pq} \sin\psi_{fg} \ .$$

**Proof**

We introduce the angles $\omega_{fp}$ between frame $f$ and the line from the origin to point $p$; the determinant $\Delta_{fg}^{(pq)}$ is easily expressed in terms of these angles:

$$\begin{aligned} \Delta_{fg}^{(pq)} &= \det \begin{bmatrix} d_p \cos\omega_{fp} & d_q \cos\omega_{fq} \\ d_p \cos\omega_{gp} & d_q \cos\omega_{gq} \end{bmatrix} \\ &= d_p d_q (\cos\omega_{fp}\cos\omega_{gq} - \cos\omega_{fq}\cos\omega_{gp}) \\ &= \frac{d_p d_q}{2}[\cos(\omega_{fp} + \omega_{gq}) + \cos(\omega_{fp} - \omega_{gq}) \\ &\quad - \cos(\omega_{fq} + \omega_{gp}) - \cos(\omega_{fq} - \omega_{gp})] \ . \end{aligned}$$

If we now observe that

$$\begin{aligned} \omega_{fq} &= \omega_{fp} + \gamma_{pq} \\ \omega_{fp} &= \psi_{fg} + \omega_{gp} = \psi_{fg} + \omega_{gq} - \gamma_{pq} \end{aligned}$$

we can write

$$\begin{aligned}
\omega_{fp} + \omega_{gq} &= \omega_{fq} + \omega_{gp} = 2\omega_{fp} - \psi_{fg} + \gamma_{pq} \\
\omega_{fp} - \omega_{gq} &= \psi_{fg} - \gamma_{pq} \\
\omega_{fq} - \omega_{gp} &= \psi_{fg} + \gamma_{pq} ,
\end{aligned}$$

so that

$$\begin{aligned}
\Delta_{fg}^{(pq)} &= \frac{d_p d_q}{2}[\cos(\psi_{fg} - \gamma_{pq}) - \cos(\psi_{fg} + \gamma_{pq})] \\
&= d_p d_q \sin\gamma_{pq} \sin\psi_{fg} ,
\end{aligned}$$

as promised.

# Motion degeneracies

The motion matrix $M$ defined in equation (2.3) is of rank smaller than three if and only if one column is a linear combination of the other two. We consider separately two cases, depending on whether the two vectors $\mathbf{c} = (c_1 \ \ldots \ c_F)^T$ and $\mathbf{s} = (s_1 \ \ldots \ s_F)^T$ are mutually dependent.

The vectors $\mathbf{c}$ and $\mathbf{s}$ are dependent only when all inter-frame rotations are integer multiples of $\pi/4$. The only interesting case of this type occurs when the camera moves by pure translation. In this case, all optical axes pass through the same point at infinity.

If, on the other hand, $\mathbf{c}$ and $\mathbf{s}$ are mutually independent, the motion matrix $M$ (and therefore the measurement matrix $U$)is of rank two if and only if there are two numbers $\alpha$ and $\beta$ such that

$$t_f = \alpha c_f + \beta s_f \ . \tag{A.1}$$

For a generic point $(\hat{X}, \hat{Z})$, the projection equation (2.1) can be rewritten in the following form:

$$t_f = u_{fp} - \hat{X} c_f - \hat{Z} s_f \ .$$

By comparing this equation with equation (A.1), we see that for the latter to hold there must be a (possibly invisible) point with coordinates $\hat{X} = -\alpha$ and $\hat{Z} = -\beta$ that is always projected to the origin, that is, such that $u_{fp} = 0$.

Since the projection ray of a point that projects to the origin is the optical axis, this proves that motion is degenerate if and only if all optical axes pass through the same point.

# Appendix B

# Simultaneous Cylinder and Plane Fitting

This Appendix elaborates on step 3 of our algorithm: the minimization of the error criterion

$$e(\mathbf{a}_3, B) = e_m(B) + e_s(\mathbf{a}_3)$$

where

$$e_m(B) = \sum_{f=1}^{F} (\hat{\mathbf{m}}_f^T B^T B \hat{\mathbf{m}}_f - 1)^2$$

and

$$e_s(\mathbf{a}_3) = \sum_{p=1}^{P} (\mathbf{a}_3 \hat{\mathbf{s}}_p - 1)^2$$

subject to the constraint

$$\mathbf{a}_3 B^T = \begin{bmatrix} 0 & 0 \end{bmatrix} . \tag{B.1}$$

We compute the solution in the following steps:

- find the cylinder $\hat{B}^T \hat{B}$ that minimizes $e_m(B)$
- find the plane $\hat{\mathbf{a}}_3$ that minimizes $e_s(\mathbf{a}_3)$
- minimize $e(\mathbf{a}_3, B)$ numerically, using $\hat{\mathbf{a}}_3$ and $\hat{B}$ as a starting point, and enforcing the constraint of equation (B.1).

We now examine the first and the third step in some detail. The second step, fitting a plane, is trivial.

## Fitting the Cylinder

Fitting a cylinder $\mathbf{m}^T B^T B \mathbf{m} = 1$ to a set of three-dimensional data $\hat{\mathbf{m}}_1, \ldots, \hat{\mathbf{m}}_F$ is a non-linear problem in the entries of the matrix $B$.

Consequently, we use the same strategy as above: we first find a good approximation to the solution, and then we refine it numerically.

The approximation can be found by first fitting a quadratic form $\mathbf{m}^T Q \mathbf{m} = 1$ to the data. Thus, rather than finding a cylinder, we find an ellipsoid. This is a linear problem in the entries of the symmetric matrix $Q$, and can be solved easily. We then decompose the result, $\hat{Q}$, and set its smallest eigenvalue to zero. The decomposition yields a first approximation to $\hat{B}$. In this way, instead of finding the optimal cylinder, we obtain the cylinder that is closest to the optimal ellipsoid. From there, we can reach the optimal cylinder by numerical minimization of $e_m(B)$. Our experiments indicate that this last step is hardly necessary: the cylinder obatained by suppressing the smallest eigenvalue of $\hat{Q}$ is almost the same as the optimal cylinder.

## Refining the Minimum of $e(\mathbf{a}_3, B)$

We now have a cylinder $\hat{B}^T\hat{B}$ and a plane $\hat{\mathbf{a}}_3$ which separately minimize the two error functions $e_m(B)$ and $e_s(\mathbf{a}_3)$. However, $\hat{B}$ and $\hat{\mathbf{a}}_3$ may not satisfy the constraint (B.1) exactly.

In order to enforce equation $(B.1)$, and at the same time minimize the global error function $e(\mathbf{a}_3, B)$, we use the constraint to write $\mathbf{a}_3$ as a function of $B$. Equation (B.1) says that $\mathbf{a}_3$ is orthogonal to both rows, $\mathbf{b}_1$ and $\mathbf{b}_2$, of $B$, so we can write

$$\mathbf{a}_3 = a_3(\mathbf{b}_1 \times \mathbf{b}_2)$$

where $\times$ denotes the cross product, and $a_3$ is a scalar.

As a result, we obtain a function $e'(a_3, B)$ of only seven variables, rather than nine. The minimization of $e'$ is now unconstrained.

To complete the task we need the derivatives of the cylinder residue function $\gamma = \mathbf{m}^T B^T B \mathbf{m} - 1$ and the plane residue function $\pi = a_3(\mathbf{b}_1 \times \mathbf{b}_2)\mathbf{s} - 1$ with respect to the unknown parameters $a_3$ and $B$, for use in a standard minimization routine.

Simple algebraic manipulation shows the derivatives to be

$$\frac{\partial \gamma}{\partial B} = 2B\mathbf{m}\mathbf{m}^T$$

(a $2 \times 3$ matrix of derivatives), and

$$\begin{aligned}
\frac{\partial \pi}{\partial \mathbf{b}_1} &= a_3(\mathbf{b}_2 \times \mathbf{s}) \\
\frac{\partial \pi}{\partial \mathbf{b}_2} &= -a_3(\mathbf{b}_1 \times \mathbf{s}) \\
\frac{\partial \pi}{\partial a_3} &= \det \begin{bmatrix} \mathbf{s} \\ B \end{bmatrix} .
\end{aligned}$$

# Appendix C

# Completion of the Matrix $A$ and its Inverse

This Appendix shows how to complete the matrix

$$A = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \end{bmatrix}$$

and its inverse

$$A^{-1} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix} = \begin{bmatrix} B^T & \mathbf{b}_3 \end{bmatrix}$$

given their submatrices $\mathbf{a}_3$ and $B$. This is step 4 of our algorithm.

The $3 \times 3$ matrix equation

$$AA^{-1} = I$$

can be expanded into nine scalar equations:

$$\begin{array}{lll} \mathbf{a}_1\mathbf{b}_1 = 1 & \mathbf{a}_1\mathbf{b}_2 = 0 & \mathbf{a}_1\mathbf{b}_3 = 0 \\ \mathbf{a}_2\mathbf{b}_1 = 0 & \mathbf{a}_2\mathbf{b}_2 = 1 & \mathbf{a}_2\mathbf{b}_3 = 0 \\ \mathbf{a}_3\mathbf{b}_1 = 0 & \mathbf{a}_3\mathbf{b}_2 = 0 & \mathbf{a}_3\mathbf{b}_3 = 1 \, . \end{array}$$

The two equations

$$\mathbf{a}_3\mathbf{b}_1 = 0 \quad \text{and} \quad \mathbf{a}_3\mathbf{b}_2 = 0$$

contain only known quantities. They coincide with the constraint equation (B.1), and can be ignored here. Since the unknown scalars are still nine (the entries of $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{b}_3$), we need two more equations.

These two degrees of freedom derive from the fact that the origin of the world coordinate system was left unspecified. Rather than constraining the origin to be at $(0, 0)$, we use these degrees of freedom to improve the noise performance of the shape result as follows.

The shape matrix is computed as $S = A\hat{S}$ in the last step of our algorithm. Of the three rows of $\hat{S}$, the third is the most sensitive to noise, because it corresponds to the smallest singular value of the decomposition (3.1). Consequently, it is advantageous to avoid using that row in the final result. This can be accomplished by requiring the third entries of $\mathbf{a}_1$ and $\mathbf{a}_2$ to be zero:

$$\begin{aligned} \mathbf{a}_1\mathbf{v} &= 0 \\ \mathbf{a}_2\mathbf{v} &= 0 , \end{aligned}$$

where $\mathbf{v} = (0, 0, 1)^T$.

We now have the nine equations we need. The six homogeneous equations express orthogonality, and we can use them to find the directions (unit vectors) $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$ of the unknown vectors $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{b}_3$. From $\mathbf{a}_1\mathbf{b}_2 = 0$ and $\mathbf{a}_1\mathbf{v} = 0$ we deduce that $\mathbf{a}_1$ is orthogonal to both $\mathbf{b}_2$ and $\mathbf{v}$, so that its unit vector is

$$\mathbf{w}_1 = \frac{\mathbf{b}_2 \times \mathbf{v}}{|\mathbf{b}_2 \times \mathbf{v}|} .$$

Similarly, for the unit vector of $\mathbf{a}_2$, the two equations $\mathbf{a}_2\mathbf{b}_1 = 0$ and $\mathbf{a}_2\mathbf{v} = 0$ yield

$$\mathbf{w}_2 = \frac{\mathbf{b}_1 \times \mathbf{v}}{|\mathbf{b}_1 \times \mathbf{v}|} .$$

From these two results, and equations $\mathbf{a}_1\mathbf{b}_3 = 0$ and $\mathbf{a}_2\mathbf{b}_3 = 0$, we obtain the unit vector of $\mathbf{b}_3$:

$$\mathbf{w}_3 = \frac{\mathbf{w}_1 \times \mathbf{w}_2}{|\mathbf{w}_1 \times \mathbf{w}_2|} .$$

The signed magnitudes $\alpha_1$, $\alpha_2$, and $\beta_3$ of $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{b}_3$ can now be found from the non-homogeneous equations

$$\begin{aligned} \mathbf{a}_1\mathbf{b}_1 &= 1 \\ \mathbf{a}_2\mathbf{b}_2 &= 1 \\ \mathbf{a}_3\mathbf{b}_3 &= 1 , \end{aligned}$$

which yield

$$\begin{aligned} \alpha_1 &= 1/(\beta_1 \cos\theta_1) \\ \alpha_2 &= 1/(\beta_2 \cos\theta_2) \\ \beta_3 &= 1/(\alpha_1 \cos\theta_3) \ , \end{aligned}$$

where $\beta_1$, $\beta_2$, $\alpha_3$ are the magnitudes of the known vectors $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{a}_3$, and $\theta_1$, $\theta_2$, $\theta_3$ are the angles between $\mathbf{a}_1$ and $\mathbf{b}_1$, $\mathbf{a}_2$ and $\mathbf{b}_2$, $\mathbf{a}_3$ and $\mathbf{b}_3$, that is,

$$\begin{aligned} \cos\theta_1 &= \mathbf{w}_1 \frac{\mathbf{b}_1}{|\mathbf{b}_1|} \\ \cos\theta_2 &= \mathbf{w}_2 \frac{\mathbf{b}_2}{|\mathbf{b}_2|} \\ \cos\theta_3 &= \mathbf{w}_3 \frac{\mathbf{a}_3}{|\mathbf{a}_3|} \ . \end{aligned}$$

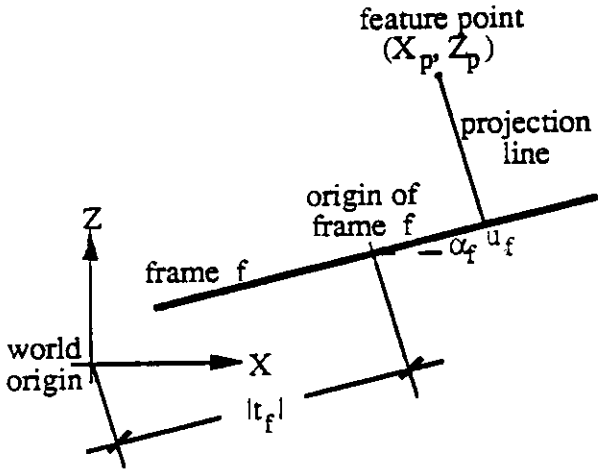


Figure 3.1: The basic geometry.

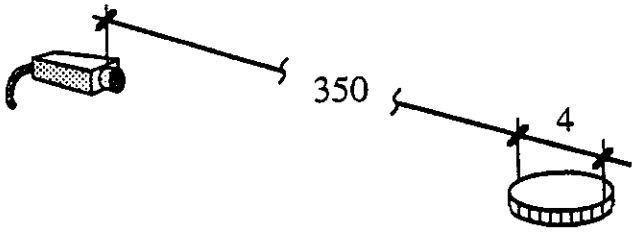


Figure 3.2: The setup in our experiment. Measures are in centimeters.

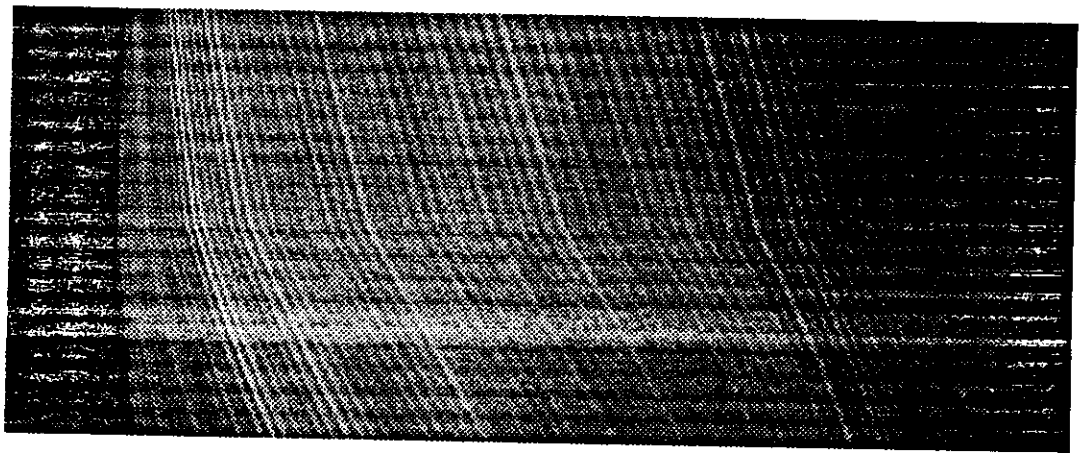

Figure 3.3: The input to the algorithm; each scanline is a new frame, and represents the edge of a one-dollar coin seen from a new angle. In [Bolles *et al.*, 1987], a figure like this is called an epipolar plane. We use it to recover shape and rotation, instead of depth given known motion.

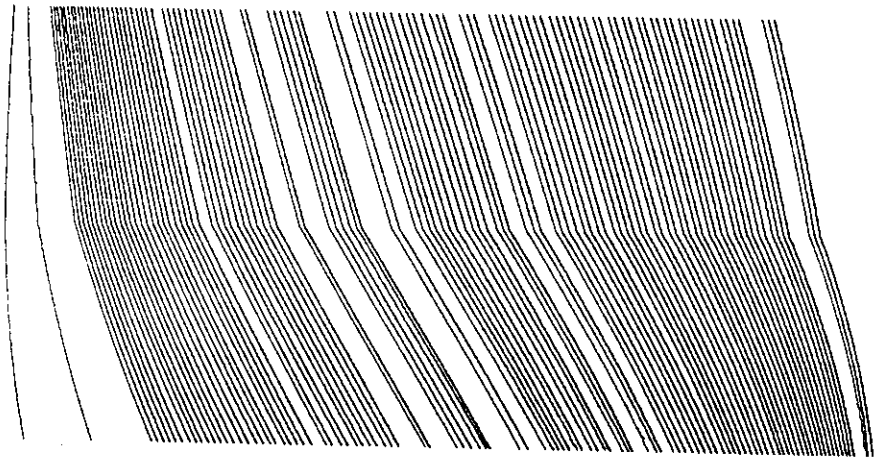

Figure 3.4: The zero crossings from figure 3.3.

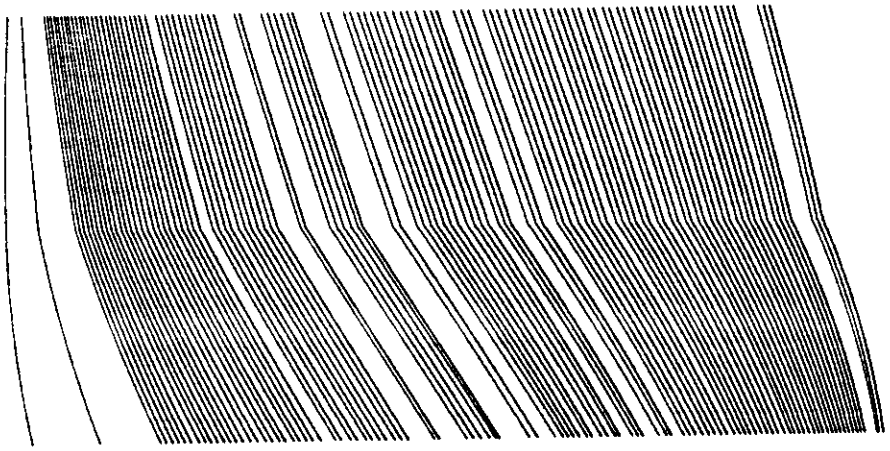

Figure 3.5: Zero crossings reconstructed after suppressing all but the first three singular values of the measurement matrix.

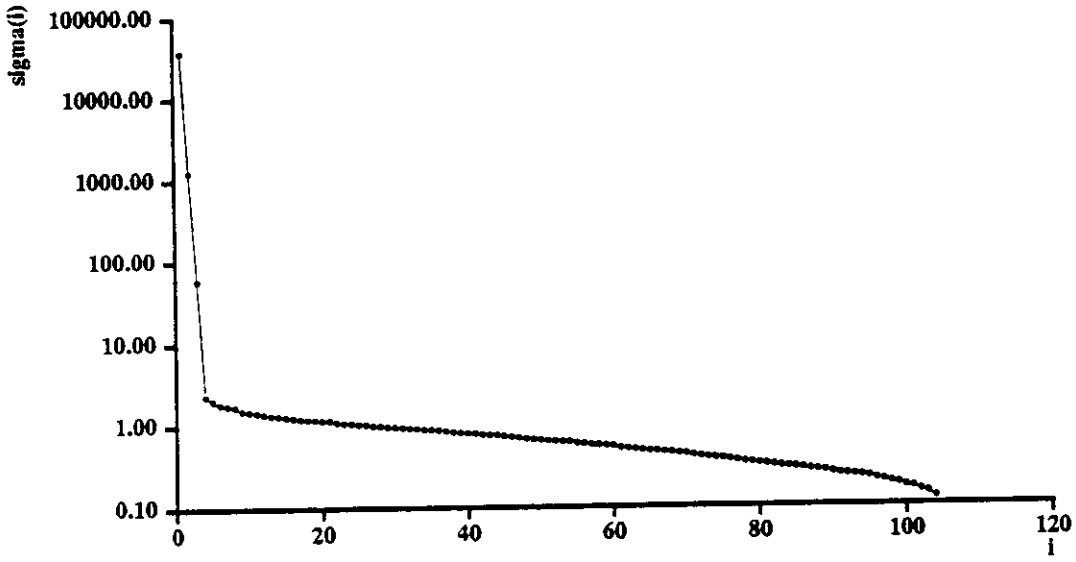


Figure 3.6: Singular values of the measurement matrix. Notice the logarithmic scale along the ordinate axis.

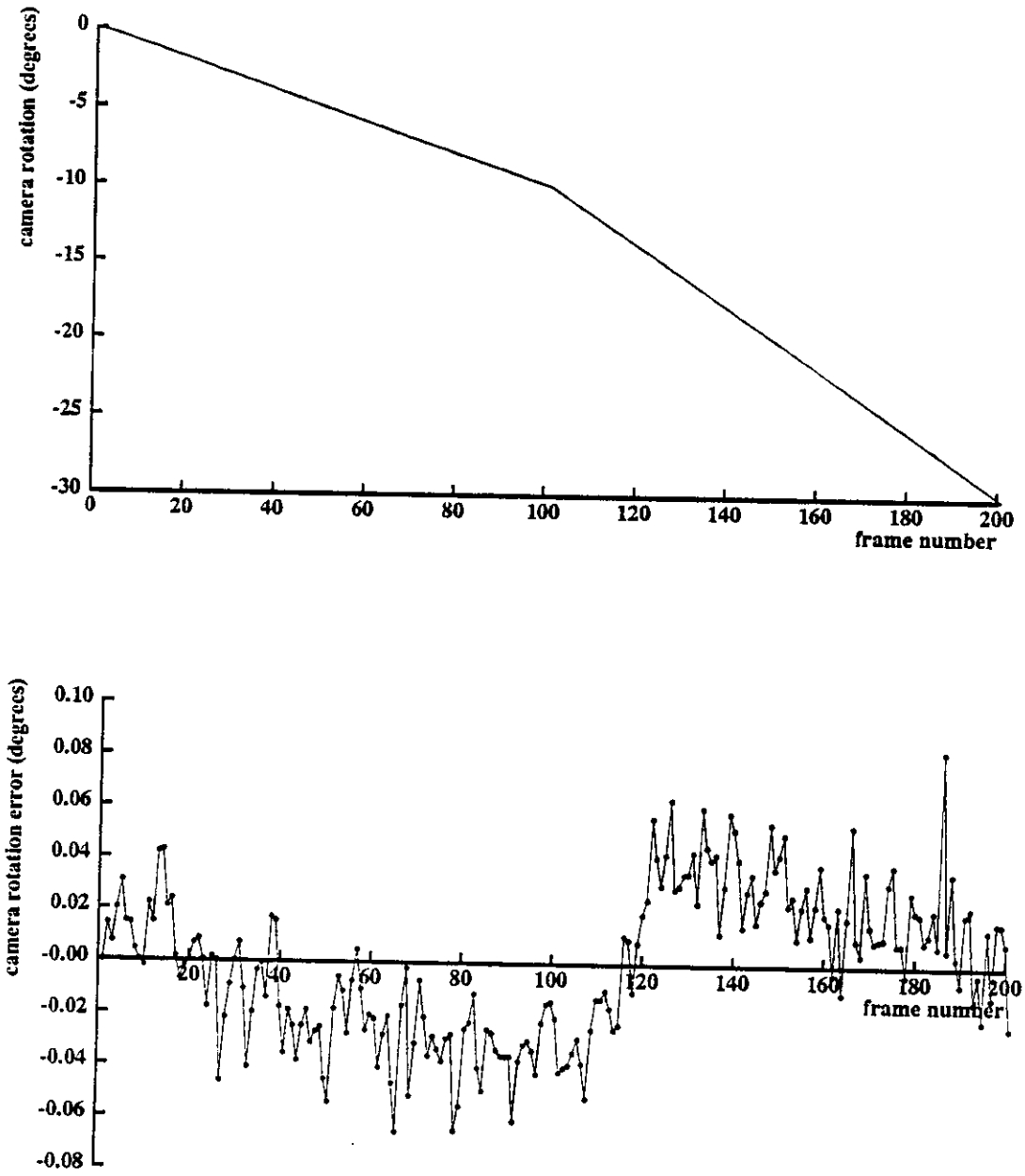


Figure 3.7: Camera rotation. In the top plot, both the computed (solid) and the true (dashed) rotation are plotted, but the difference is so small that they can hardly be distinguished. In the plot below, the difference between the two graphs is enlarged.

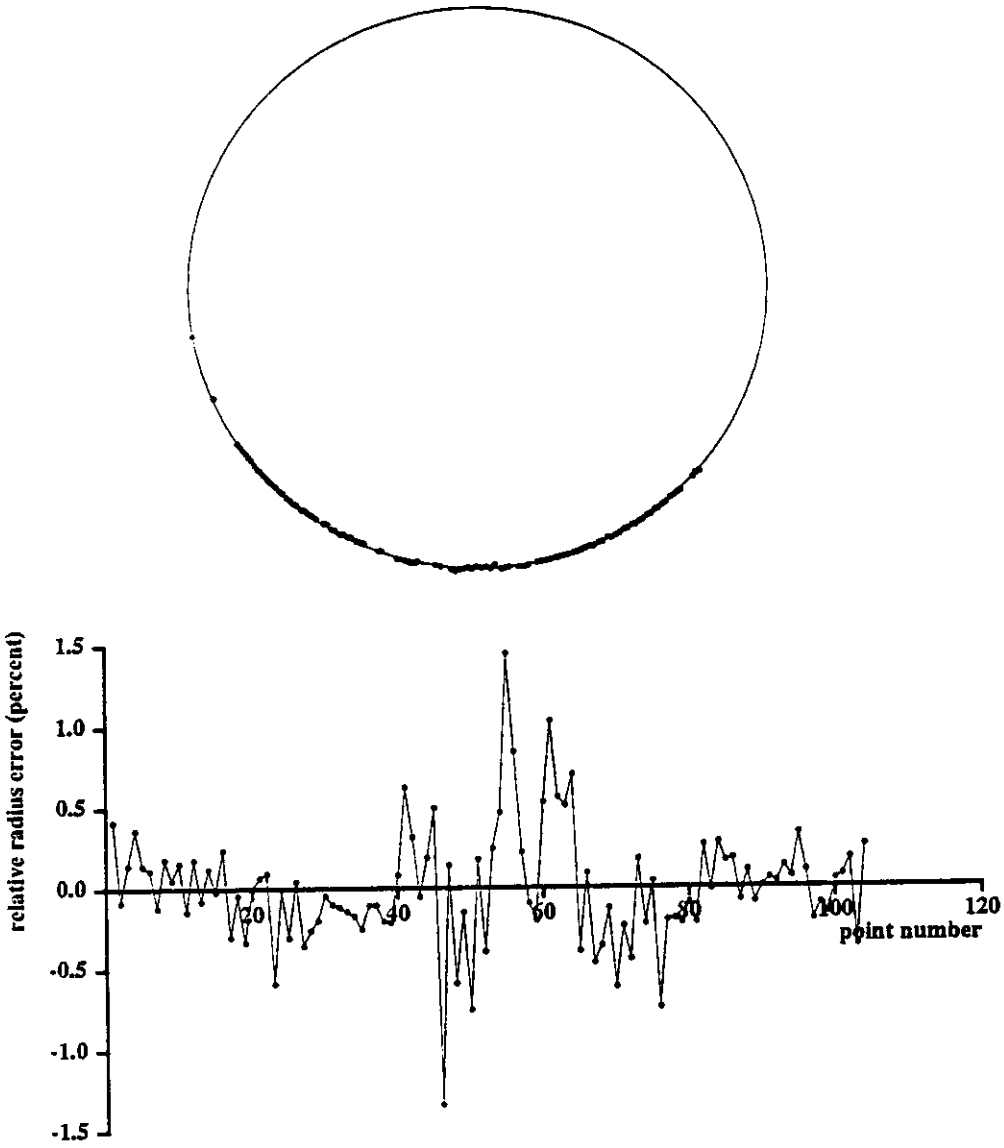


Figure 3.8: Shape. The top figure shows the computed shape (dots) of a one-dollar coin, with the best fit circle. The bottom figure magnifies the difference between true and computed shape values along the radius of the coin.

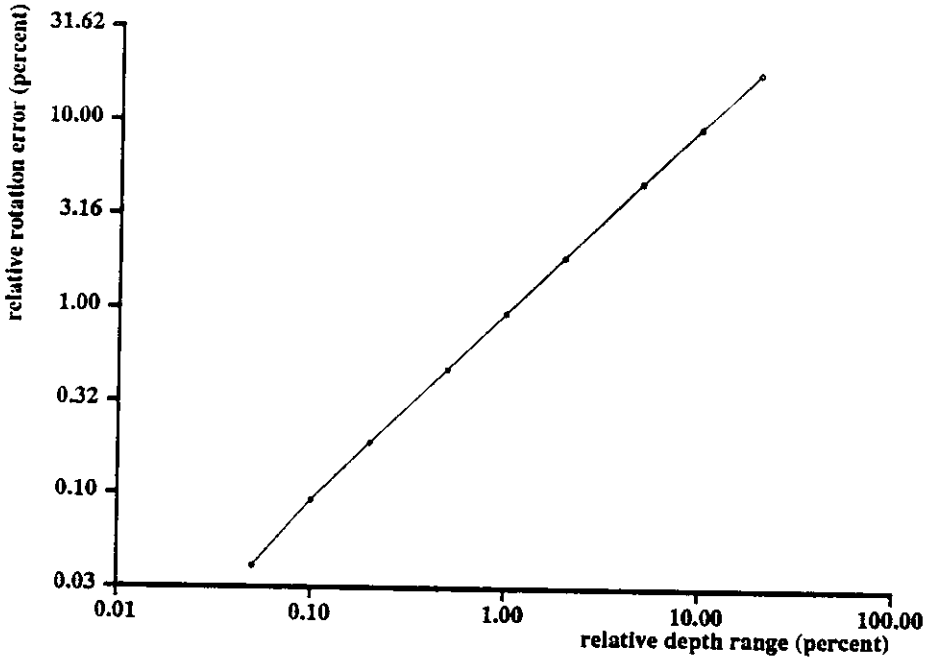


Figure 3.9: The motion error due to perspective distortion decreases when the relative depth range becomes smaller. These results were obtained by simulating noise-free images of a circular object with 10 features, and a pin-hole camera rotating by 30 degrees in 30 frames.

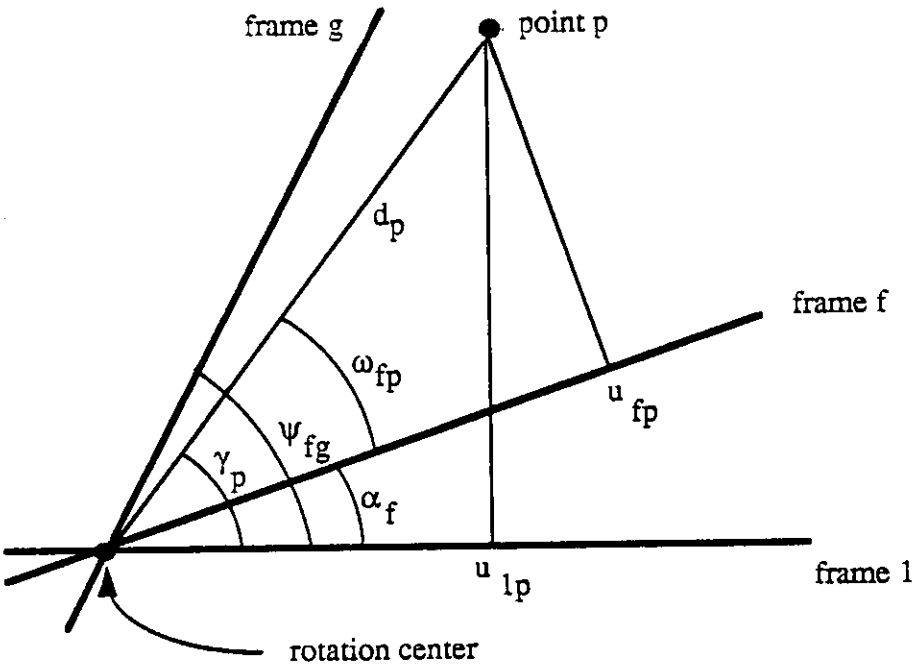


Figure 3.10: The angles defined in appendix A.