Shape and Motion from Image Streams: a Factorization Method

2. Point Features in 3D Motion

Carlo Tomasi   Takeo Kanade

January 1991

CMU-CS-91-105

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was sponsored by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597.

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

Abstract

In principle, three orthographic images of four points are sufficient to recover the positions of the points relative to each other (shape), and the viewpoints from which the images were taken (motion).
In practice, however, the solution to this structure-from-motion problem is reliable only when the viewing direction changes considerably between images. This conflicts with the difficulty of establishing correspondence between images over long-range camera motions.
Image streams, long sequences of images covering a wide motion in small steps, allow solving this conflict by using tracking for correspondence and redundancy for increased reliability in the structure-from-motion computation.
This report is the second of a series on a new factorization method for the computation of shape and camera motion from a stream of images. While the first report considered a camera moving on a plane, we now extend theory, analysis and experiments to general, three-dimensional motion.
In our method, we represent feature points in an image stream by a $2F \times P$ measurement matrix, which gathers the horizontal and vertical coordinates of the P points tracked through F frames. If coordinates are measured with respect to their centroid, we show that under orthography the measurement matrix is of rank 3.
Using this fact, we cast structure-from-motion as a matrix factorization problem, which we solve with an algorithm based on Singular Value Decomposition. Our algorithm gives accurate results, without relying on any smoothness assumption for either shape or motion.

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.

The first report, already published as CMU-CS-90-166, illustrates the idea in the case of planar motion, in which images are single scanlines.

The present report, number 2, extends the idea to three-dimensional camera motion and full image streams. It also carries out a somewhat more systematic error analysis, and discusses an experiment with a real stream. The method used to track points from frame to frame is described in detail in report number 3.

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.

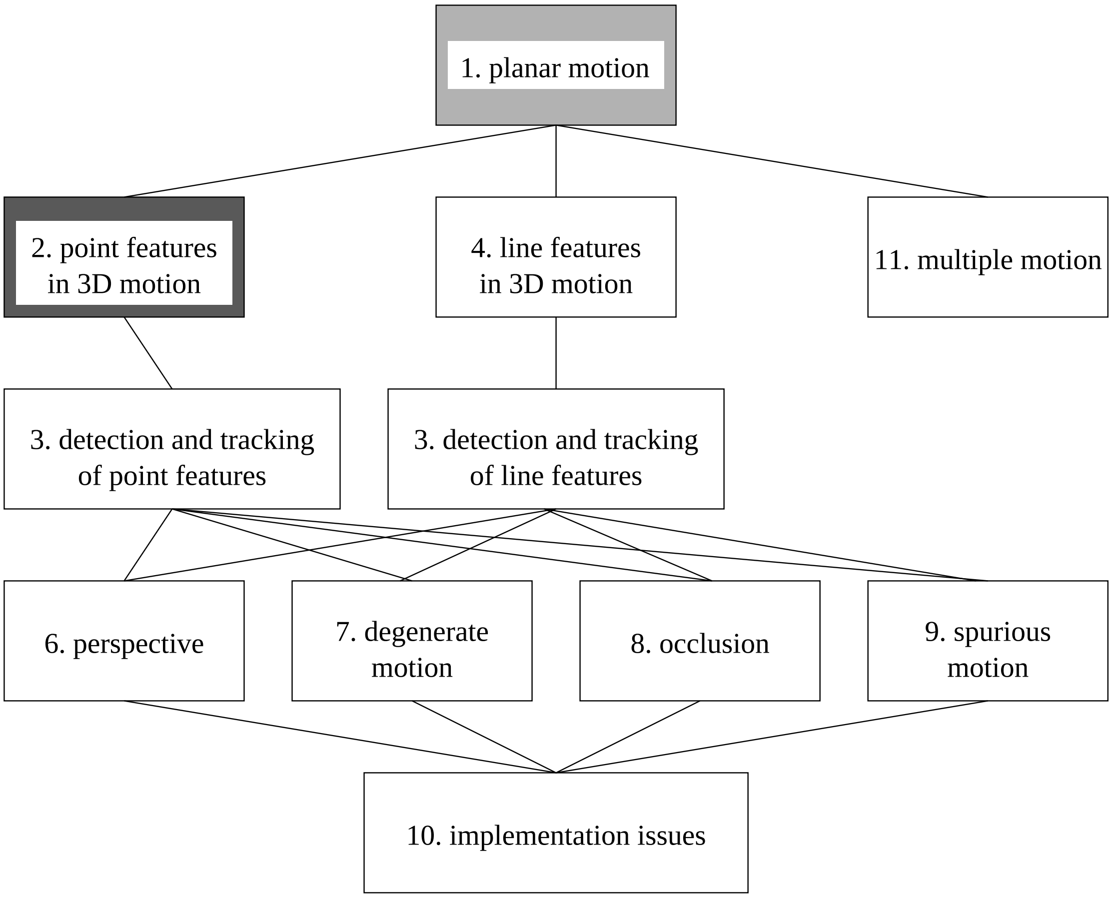


Figure 0.1: The reports in the series. Number 1 was published as CMU-CS-90-166.

Chapter 1

Introduction

In principle, three orthographic images of four points are sufficient to recover *shape* and *motion*, that is, the positions of the points relative to each other and the viewpoints from which the images were taken.

In practice, however, the solution to this structure-from-motion problem is reliable only when the viewing direction changes considerably from image to image. For a given level of image noise and scene distance, this implies a substantial camera motion to achieve good performance.

Too wide a camera motion, on the other hand, poses two fundamental problems which limit the range of acceptable motions from above: establishing correspondence between images and avoiding occlusion of features from image to image.

To summarize, correspondence and occlusion are lesser problems with short-range motions, while reliability calls for long-range motion. Typically, for the classical structure-from-motion problem, characterized by few frames and points, these requirements are not compatible.

Image streams, that is, long sequences of image frames covering a wide motion in small steps, allow bridging the gap between short and long range motion. On the one hand, features can be, thus establishing correspondence between distant frames through the solution of many, simple correspondence problems between consecutive frames. On the other hand, the redundancy of information intrinsic in a stream improves the resiliency to noise, and allows dealing with streams spanning a shorter range of viewing angles.

A successful attack to the structure-from-motion problem based on image streams depends on our ability to deal with the great mass of data in an image stream in a systematic and computationally efficient way for the recovery of shape and motion.

In [Tomasi and Kanade, 1990a] (also presented at ICCV90, [Tomasi and Kanade, 1990b]), we showed that, under the assumption of orthographic projection, we can directly *factor* the information contained in an image stream into shape and motion. However, we made the assumption that the camera move on a plane, in order to simplify the mathematical treatment. In the present report, we remove this assumption.

More specifically, we show that an image stream can be represented by a $2F \times P$ measurement matrix, which gathers the horizontal and vertical coordinates of P points tracked through F frames. If image coordinates are measured with respect to their centroid, we prove the following *rank principle*: under orthography, the measurement matrix is of rank 3. As a consequence of this principle, we show that measurement matrix can be factored into the product of two slender matrices of size $2F \times 3$ and $3 \times P$, respectively, where the first matrix encodes motion, the second shape.

The extension from planar to arbitrary camera motion is relatively straightforward, as far as the computation of shape and motion is concerned. The measurement matrix becomes twice as large, and one must incorporate the constraint that the columns and the rows in every image are mutually orthogonal.

Outline of the Report

In the next chapter we examine how our work relates with relevant results in the literature. This review extends somewhat the one given in [Tomasi and Kanade, 1990a].

We then show how to build the measurement matrix from a stream of full images (chapter 3), prove that the measurement matrix is of rank 3 (chapter 4), and show how to use this result to factor the measurement matrix into shape and camera rotation (chapter 5).

Chapter 6 describes some experiments on synthetic and real image streams, and chapter 7 summarizes the differences with respect to the planar case.

Chapter 2

Relations with Previous Work

In this chapter, we place our algorithm within the context of the literature on structure-from-motion, by pointing out papers which pursue closely related approaches, and by stating our major contributions to the field.

Some shape-from-motion algorithms address the most general problem, others only part of it. Approaches differ in the assumptions they make about the world, the imaging model, and the motion. They require different types of input primitives, work on varying numbers of images, and produce different forms of output.

We address the problem of shape-from-motion in its entirety, in that we make no assumption on either shape or motion. In this regard, our work can be compared with that of Thompson [Thompson, 1959], one of the earliest solutions to the two-frame problem, and Ullman [Ullman, 1979], who proposed an automated solution for four points and three frames. Our work can be contrasted to that of Baker et al. [Bolles *et al.*, 1987], and that of Matthies et al. [Matthies *et al.*, 1989], in that they recover depth from known motion.

Heel [Heel, 1989] produces dense maps and recovers both depth and motion, but he restricts the latter to pure translation. Both Heel and Matthies use incremental methods, where new images are processed as they become available, while we process a whole sequence at once. This is a disadvantage of our method, which deserves future investigation.

The only restrictions we make are the static (rigid) world and the orthographic imaging model. Ullman [Ullman, 1984] considers a less restrictive world model, allowing for non-rigid and rubbery motion, but is more interested in understanding biological systems than in achieving high precision. With Ullman [Ullman, 1979] we use an orthographic projection model, while Prazdny [Prazdny, 1980], Bruss and Horn [Bruss and Horn, 1983], Adiv [Adiv, 1985], Waxman and Wohn [Waxman and Wohn, 1985], and more recently Heeger and Jepson [Heeger and Jepson, 1989] and [Spetsakis and Aloimonos, 1989] assume a perspective model. On the other hand, Pradzny, Bruss and Horn, and Adiv cast the solution as a general search in the space of possible motions, which is computationally expensive, Waxman and Wohn use second order derivatives of image intensity, which are sensitive to noise, and Spetsakis and Aloimonos need a two-frame algorithm to initialize the search. The work by Heeger and Jepson is discussed below.

We make no assumption about the motion, except that it must contain a sufficiently large rotation component around the scene. Frequent limitations in the literature are to pure translation [Lawton, 1983], [Jain, 1983], [Matthies *et al.*, 1989], cases where one of the motion components is known [Horn and Weldon, 1988], smooth motion [Broida *et al.*, 1990], or constant motion [Debrunner and Ahuja, 1990]. On the other hand, Lawton, Jain, Matthies et al., Horn, Broida et al. propose solutions for the perspective model, and Matthies et al. and Broida et al. have incremental algorithms.

Our solution is multi-frame, and can therefore achieve a lesser sensitivity to noise than other approaches. Ullman works with three frames in [Ullman, 1979], but extends it to an arbitrary number in [Ullman, 1984]. Tsai and Huang [Tsai and Huang, 1984] work on two frames at a time, and Heeger and Jepson [Heeger and Jepson, 1989] use instantaneous image velocities.

The results by Heeger and Jepson [Heeger and Jepson, 1989] and Debrunner and Ahuja [Debrunner and Ahuja, 1990] are similar to ours at a conceptual level, in that they are also based on the bilinear nature of the projection equation. Heeger and Jepson, on the other hand, use essentially two frames at a time, and the perspective imaging model. Debrunner and Ahuja limit the type of motion, as said above, and use a different mathematical formalism.

We use Singular Value Decomposition as our major tool for solution. Also Tsai and Huang [Tsai and Huang, 1981], [Tsai *et al.*, 1982] use SVD, but only as an intermediate technical step to decompose their "essential parameter matrix". In contrast, we use SVD to decompose the measurement matrix, corresponding to an image stream, into shape and motion. Thus, SVD is the fundamental core of our method, and is the computational counterpart to the rank principle proven in chapter 4. Thus, the relation with Tsai and Huang is only superficial.

To summarize, our algorithm is general, in that it does not assume *a priori* knowledge of either shape or motion, and makes no assumptions on the latter. It assumes rigid shape, and an orthographic projection model. It requires features to be extracted from images and tracked over time in the image stream. The algorithm produces motion and relative shape (the 3D coordinates of the tracked feature points relative to each other).

The major contributions of our approach are two. One is the conceptual result of the rank principle, described in chapter 4, which captures precisely and simply the nature of the redundancy of an image stream. The other contribution is the computational efficiency and simplicity of our matrix factorization method, which is based on the well-behaved algorithm of Singular Value Decomposition [Golub and Reinsch, 1971].

Chapter 3

The Measurement Matrix

In this chapter we show how to transform an image stream into a matrix collecting the feature coordinates to be fed to the algorithm that computes shape and motion. This assumes the existence of a method for tracking features from frame to frame, which will be described in detail in report number 3 in our series.

Suppose that we track P feature points over F frames in the image stream, resulting in a sequence of image coordinates $\{(u'_{fp}, v'_{fp}) \mid f = 1, \ldots, F,\ p = 1, \ldots, P\}$.

The horizontal coordinates u'_{fp} of those features are written into an $F \times P$ matrix U': there is one row per frame, and one column per feature point. Similarly, an $F \times P$ matrix V' is built from the vertical coordinates v'_{fp}.

The rows of the matrices U' and V' are then registered by subtracting from each entry the centroid of the entries in the same row:

$$\begin{aligned} u_{fp} &= u'_{fp} - \overline{u}_f \\ v_{fp} &= v'_{fp} - \overline{v}_f \ , \end{aligned} \tag{3.1}$$

where

$$\begin{aligned} \overline{u}_f &= \frac{1}{P} \sum_{p=1}^{P} u'_{fp} \\ \overline{v}_f &= \frac{1}{P} \sum_{p=1}^{P} v'_{fp} \ . \end{aligned}$$

This produces two new $F \times P$ matrices $U = [u_{fp}]$ and $V = [v_{fp}]$. The matrix

$$W = \left[\frac{U}{V}\right]$$

is called the *measurement matrix*. This is the input to our shape-and-motion algorithm.

Some of the features disappear during tracking, because of occlusion. Some others change in appearance so much that they are discarded as unreliable. Only the features that survive from the first to the last frame are used in the shape and motion recovery stage. In the future, we plan to investigate how to modify our algorithm to deal with a variable number of feature points over the image stream.

Chapter 4

The Rank of the Measurement Matrix

This chapter introduces the fundamental principle on which our shape-and-motion algorithm is based: the $2F \times P$ matrix W of the registered image coordinates of P points tracked through F frames is higly rank-deficient.

The orientation of the camera reference system corresponding to frame number f is determined by a pair of unit vectors, $\mathbf{i}_f$and $\mathbf{j}_f$, pointing along the scanlines and the columns of the image respectively, and defined with respect to a world reference system with coordinates x, y, and z (see figure 7.1). Under orthography, all projection rays are then parallel to the cross product of $\mathbf{i}_f$and $\mathbf{j}_f$:

$$\mathbf{k}_f = \mathbf{i}_f \times \mathbf{j}_f \ .$$

The position of the camera reference system is determined by the position of the *image center*, defined as the point on the image plane with respect to which all image coordinates are measured.

The projection (u'_{fp}, v'_{fp}) of point $\mathbf{s}'_p = (x'_p, y'_p, z'_p)^T$ onto frame f is then given by the equations

$$\begin{aligned} u'_{fp} &= \mathbf{i}_f \cdot (\mathbf{s}'_p - \mathbf{t}_f) \\ v'_{fp} &= \mathbf{j}_f \cdot (\mathbf{s}'_p - \mathbf{t}_f) \ , \end{aligned}$$

where $\mathbf{t}_f$is the vector from the world origin to the image center of frame f.

We can now write expressions for the entries u_{fp} and v_{fp} of the measurement matrix by substituting the projection equations above into the registration equations (3.1). For the horizontal coordinates we have

$$\begin{aligned} u_{fp} &= u'_{fp} - \overline{u}_f \\ &= \mathbf{i}_f \cdot (\mathbf{s}'_p - \mathbf{t}_f) - \frac{1}{P}\sum_{p=1}^{P} \mathbf{i}_f \cdot (\mathbf{s}'_p - \mathbf{t}_f) \\ &= \mathbf{i}_f \cdot \left(\mathbf{s}'_p - \frac{1}{P}\sum_{p=1}^{P} \mathbf{s}'_p \right) \\ &= \mathbf{i}_f \cdot \mathbf{s}_p \ , \end{aligned}$$

where

$$\mathbf{s}_p = \frac{1}{P}\sum_{p=1}^{P} \mathbf{s}'_p$$

is the centroid of the scene points in space. Thus, the fact that the projection of the centroid is the centroid of the projections allows us to remove translation from the projection equations.

We can write a similar equation for the registered vertical image projection v_{fp}. To summarize,

$$\begin{aligned} u_{fp} &= \mathbf{i}_f \cdot \mathbf{s}_p \\ v_{fp} &= \mathbf{j}_f \cdot \mathbf{s}_p \ , \end{aligned} \tag{4.1}$$

where $\mathbf{s}_p = (x_p, y_p, z_p)$ gathers the coordinates of scene point number p with respect to the centroid of all the points being tracked.

Notice that, in our formulation, translation and rotation are referred to a world-centered system of reference. This is different from the camera-centered reference system usually used for the perspective projection equations. Also, while camera rotation in a camera-centered system supplies no shape information under perspective, translation (in either frame) is useless for shape recovery under orthography. This should come to no surprise: a set of images contains shape information only if the images are taken from different viewpoints, that is, when the center of projection moves betwen images. Under perspective, motion of the center of projection means translation of the camera. Under orthography, the center of projection is at infinity, and moving the center of projection means changing the direction of the projection rays.

Because of the two sets of $F \times P$ equations (4.1), the measurement matrix W can be expressed in a matrix form:

$$W = MS \tag{4.2}$$

where

$$M = \begin{bmatrix} \mathbf{i}_1^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_1^T \\ \vdots \\ \mathbf{j}_F^T \end{bmatrix} \tag{4.3}$$

represents the camera motion, and

$$S = \begin{bmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_P \end{bmatrix} \tag{4.4}$$

is the shape matrix. In fact, the rows of M represent the orientations of the horizontal and vertical camera reference axes throughout the sequence, while the columns of S are the coordinates of the P feature points with respect to their centroid.

Since M is $2F \times 3$ and S is $3 \times P$, the matrix projection equation (4.2) implies the following fact.

The Rank Principle

> *Without noise, the measurement matrix W is at most of rank three.*

The rank principle expresses the simple fact that the $2F \times P$ image measurements are highly redundant. Indeed, they could all be described concisely by giving F frame reference systems and P point coordinate vectors, if only these were known.

Geometrically, the rank principle expresses an incidence property. In fact, we can view the projection of point p onto frame f as the intersection of three planes: a "vertical" plane through $\mathbf{s}_p$ and orthogonal to the unit vector $\mathbf{i}_f$, a "horizontal" plane through $\mathbf{s}_p$ and orthogonal to the unit vector $\mathbf{j}_f$, and the image plane. The two projection equations (4.1) say that the "vertical" planes through point p belong to a star [1] of planes, and so do the "horizontal" planes.

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

[1] A star of planes is the set of planes passing through a fixed point.

Chapter 5

The Factorization Method

When noise corrupts the images, the measurement matrix W 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. The next section introduces this extension, using the concept of Singular Value Decomposition [Golub and Reinsch, 1971] to introduce the notion of approximate rank.

Section 5.2 then points out the properties of the motion matrix M in the projection equation (4.2) that must be enforced to uniquely determine the shape and motion solution. Finally, section 5.3 outlines the complete shape-and-motion algorithm.

5.1 Approximate Rank

Assuming [1] that $2F \geq P$, the matrix W can be decomposed [Golub and Reinsch, 1971] into a $2F \times P$ matrix L, a diagonal $P \times P$ matrix Σ, and a $P \times P$ matrix R,

$$W = L \Sigma R \ , \tag{5.1}$$

such that $L^T L = R^T R = R R^T = \mathcal{I}$, and $\sigma_1 \geq \ldots \geq \sigma_P$. Here, $\mathcal{I}$ is the $P \times P$ identity matrix, and the *singular values* $\sigma_1, \ldots, \sigma_P$ are the diagonal entries of Σ. This is the *Singular Value Decomposition* (SVD) of the matrix U.

[1] This assumption is not crucial: if $2F < P$, everything can be repeated for the transpose of W.

If we now partition the matrices L, Σ, and R as follows:

$$
\begin{aligned}
L &= \left[\begin{array}{c|c} \underbrace{L'}_{3} & \underbrace{L''}_{P-3} \end{array}\right] \}_{2F} \\
\Sigma &= \left[\begin{array}{c|c} \Sigma' & 0 \\ \hline 0 & \Sigma'' \end{array}\right] \begin{array}{l} \}_{3} \\ \}_{P-3} \end{array} \\
& \quad\;\; \underbrace{}_{3}\;\;\underbrace{}_{P-3} \\
R &= \underbrace{\left[\begin{array}{c} R' \\ \hline R'' \end{array}\right]}_{P} \begin{array}{l} \}_{3} \\ \}_{P-3} \end{array} \quad ,
\end{aligned}
\tag{5.2}
$$

we have

$$
L\Sigma R = L'\Sigma' R' + L''\Sigma'' R'' \; .
$$

Let W^* be the ideal measurement matrix, that is, the matrix we would obtain in the absence of noise. Because of the rank principle, the non-zero singular values of W^* are at most three. Since the singular values in Σ are sorted in non-increasing order, Σ' must contain all the singular values of W^* that exceed the noise level. As a consequence, the term $L''\Sigma''R''$ must be due entirely to noise, and the product $L'\Sigma'R'$ is the best possible rank-3 approximation to W^*.

We can now restate our key point.

The Rank Principle for Noisy Measurements

> *All the shape and motion information in W is contained in its three greatest singular values, together with the corresponding left and right eigenvectors.*

Thus, the best possible approximations to the ideal measurement matrix W^* is the product

$$
\hat{W} = L'\Sigma' R'
$$

where the primes refer to the partition (5.2). With the definitions

$$\begin{aligned}\hat{M} &= L'[\Sigma']^{1/2} \\ \hat{S} &= [\Sigma']^{1/2}R' ,\end{aligned}$$

we can also write

$$\hat{W} = \hat{M}\hat{S} . \tag{5.3}$$

The two matrices $\hat{M}$ and $\hat{S}$ are of the same size as the desired motion and shape matrices M and S: $\hat{M}$ is $2F \times 3$, and $\hat{S}$ is $3 \times P$. However, the decompositions (5.3) are not unique. In fact, if A is *any* invertible 3×3 matrix, the matrices $\hat{M}A$ and $A^{-1}\hat{S}$ are also a valid decomposition of W, since

$$(\hat{M}A)(A^{-1}\hat{S}) = \hat{M}(AA^{-1})\hat{S} = \hat{M}\hat{S} = \hat{W} .$$

Thus, $\hat{M}$ and $\hat{S}$ are in general different from M and S. A striking fact, however, is that, except for noise, the matrix $\hat{M}$ is a linear transformation of the true motion matrix M, and the matrix $\hat{S}$ is a linear transformation of the true shape matrix S. In fact, in the absence of noise, M and $\hat{M}$ both span the column space of the measurement matrix $W = W^* = \hat{W}$. Since that column space is three-dimensional, because of the rank principle, M and $\hat{M}$ are different bases for the same space, and there must be a linear transformation between them.

Whether the noise level is low enough that it can be ignored at this juncture depends also on the camera motion and on shape. Notice, however, that the singular value decomposition yields sufficient information to make this decision: the requirement is that the ratios between the third and the fourth largest singular values of W be sufficiently large.

5.2 The Metric Constraints

To summarize, the matrix $\hat{M}$ is a linear transformation of the true motion matrix M. Likewise, $\hat{S}$ is a linear transformation of the true shape matrix S. More specifically, there exists a 3×3 matrix A such that

$$\begin{aligned}M &= \hat{M}A \\ S &= A^{-1}\hat{S} .\end{aligned} \tag{5.4}$$

In order to find A it is sufficient to observe that the rows of the true motion matrix M are unit vectors, and that the first F are orthogonal to corresponding F in the second half. These *metric constraints* yield the over-constrained, quadratic system

$$\begin{aligned} \hat{\mathbf{i}}_f^T A A^T \hat{\mathbf{i}}_f &= 1 \\ \hat{\mathbf{j}}_f^T A A^T \hat{\mathbf{j}}_f &= 1 \\ \hat{\mathbf{i}}_f^T A A^T \hat{\mathbf{j}}_f &= 0 \end{aligned} \tag{5.5}$$

in the entries of A. This is a simple data fitting problem which, though non-linear, can be solved efficiently and reliably.

A last ambiguity needs to be resolved: if A is a solution of the metric constraint problem, so is AR, where R is any orthonormal matrix. In fact,

$$\begin{aligned} \hat{\mathbf{i}}_f^T (AR)(R^T A^T) \hat{\mathbf{i}}_f &= \hat{\mathbf{i}}_f^T A (R R^T) A^T \hat{\mathbf{i}}_f \\ &= \hat{\mathbf{i}}_f^T A A^T \hat{\mathbf{i}}_f \\ &= 1 , \end{aligned}$$

and likewise for the remaining two constraint equations. Geometrically, this corresponds to the fact that the solution is determined up to a rotation, since the orientation of, say, the first camera reference system with respect to the world reference system is arbitrary. This arbitrariness can be removed, if desired, by rotating the solution so that the first frame is represented by the identity matrix.

5.3 Outline of the Complete Algorithm

Based on the development in the previous sections, we now have a complete algorithm for the computation of shape and rotation from the measurement matrix W derived from a stream of images. To summarize, the motion matrix M and the shape matrix S defined in equations (4.3) and (4.4) can be computed as follows.

1. Compute the singular-value decompositions of W:

$$W = L \Sigma R \, .$$

2. Define

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

   where the primes refer to the block partitioning defined in (5.2).

3. Compute the matrix A in equations (5.4) by imposing the metric constraints (equations (5.5)).

4. Compute the motion matrix M and the shape matrix S as

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

5. If desired, align the first camera reference system with the world reference system by finding the rotation matrix R' that minimizes the residue

$$\left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - R' \begin{bmatrix} \mathbf{i}_1 & \mathbf{j}_1 & \mathbf{k}_1 \end{bmatrix} \right\| ,$$

   where the columns of the identity matrix on the left represent the axis unit vectors of the world reference system, $\mathbf{i}_1$ and $\mathbf{j}_1$ are the first and $F+1$-st row of M, and $\mathbf{k}_1 = \mathbf{i}_1 \times \mathbf{j}_1$. This is an absolute orientation problem, and can be solved by the procedure described in [Horn *et al.*, 1988].

Chapter 6

Experiments

This chapter describes experiments on the computation of shape and motion.

First, the three-dimensional shape and motion computation is tested on simulations, in order to assess the ability of the algorithm to cope with varying levels of noise and with different numbers of frames and feature points.

Second, experiments on real images are presented. The complete shape and motion computation is demonstrated with general camera motion, for which accurate ground truth was measured with a mechanical positioning platform.

Simulations

In our simulations, we generated points in space randomly within a cube, with uniform distribution for each coordinate. We simulated camera motion with the camera always keeping those points in the field of view. Because of the assumption of orthographic projection, the distance between the camera and the object cannot be computed by the algorithm. Instead, the algorithm computes the remaining components of translation, along the image plane, in the registration phase (equations (3.1)) and the rotation of the camera around the centroid of the feature points. As discussed in chapter 4, under orthography, only the rotation component contains shape information. Consequently, we ignore the translation. In the simulations, rotation is characterized by equal amounts of pitch, yaw and roll. We added Gaussian noise to the images with various standard deviations, in order to test the robustness of the algorithm.

Figures 7.2 through 7.4 show the simulation results. In each pair of graphs, the first shows the shape error and the second shows the rotation error.

The shape error is defined as the root-mean-squared error on the computed point coordinates, averaged over all points, and divided by the side-length of the cube.

We measured the angle distance between true and computed rotation as the smallest angle necessary to make the two rotations coincide. The rotation error is then defined as the root-mean-squared angle distance error, averaged over all frames, and divided by the total rotation angle.

The effect of image noise is shown in figure 7.2 for three different point-set sizes. For the curves in these two figures, the camera motion was 30 degrees. For the noise levels in the abscissas we assume a 512×512 image. Similar diagrams are shown in figure 7.3, but for varying stream lengths. The total camera rotation was kept at 30 degrees, so shorter streams correspond to greater motions between frames.

Figure 7.4 shows the effects of the total angle of rotation. Both the number of points and the number of frames were set to 50, and plots are given for a few different image noise levels.

Qualitatively, the diagrams show what we would expect: errors increase with more image noise, fewer points or frames, and a smaller range of viewing angles.

Quantitatively, we observe that even for noise values as high as three pixels standard deviation the shape and motion errors are less than one half of one percent. This is demonstrated again in the next section, which describes experiments on real image streams.

The noise we found in practice with real streams is on the order of a tenth of a pixel. The simulations show that for those noise levels we can expect good performance even with as few as ten points and viewing angles of five degrees.

Real Image Streams

In this section we describe an experiment on a real stream of images of a small plastic model of a building. The camera is a Sony CCD camera with a

200 mm lens, and is moved by means of a high-precision positioning platform. Figure 7.5 shows the setup. The motion of the camera is such as to keep the building withing the field of view throughout the stream. Some frames in the stream are shown in figure 7.6. Camera pitch, yaw, and roll around the model are all varied as shown by the dashed curves in figures 7.12 through 7.11.

Figure 7.7 shows the trajectories left by a subset of 50 features tracked through 150 frames. For feature tracking, we extended the method described in [Lucas and Kanade, 1981] to allow also for the automatic selection of image features. We will describe the method in report number 3 in our series. The entire set of 430 features is displayed in figure 7.8, overlayed on the first frame of the stream. Of these features, 42 were abandoned during tracking because their appearance changed too much. The remaining 388 features are displayed in figure 7.9, superimposed on the last frame of the sequence.

The solid curves in figures 7.10, 7.11, and 7.12 compare the rotation components computed by the algorithm with the values measured mechanically from the mobile platform. In each figure, the top diagram shows the computed and the measured rotation components, superimposed, while the bottom diagram shows the difference between the two.

The errors are everywhere less than 0.4 degrees. The computed motion follows closely also rotations with curved profiles, such as the roll profile between frames 1 and 20 (figure 7.11), and faithfully preserves all discontinuities in the rotational velocities. This is a consequence of the fact that no assumption was made on the camera motion: the algorithm does not smooth the results. If the rows of the measurement matrix were permuted, thus simulating a camera jumping back and forth, the computed yaw, roll, and pitch values would be permuted correspondingly, yielding discontinuous plots.

Between frames 60 and 80, yaw and pitch are nearly constant. This means that the image stream contains almost no shape information along the optical axis during that subsequence, since the camera is merely rotating about its optical axis. This demonstrates that it is sufficient for the stream *as a whole* to be taken during non-degenerate motion. The algorithm can deal without difficulty with streams that contain degenerate subsequences, because the information in the stream is used all at once in our method.

We are not yet able to account for the residue error of about 0.3 degrees in the yaw and pitch components. This might be due to lens distortion, errors in the determination of the camera's aspect ratio, poor calibration of the mechanical measurements of motion, or errors due to our algorithm.

The shape results are shown qualitatively in figure 7.13, which shows the computed shape viewed from above. The view in figure 7.13 is similar to that in figure 7.14, included for visual comparison. Notice that the walls, the windows on the roof, and the chimneys are recovered in their correct positions.

To evaluate the shape performance quantitatively, we measured some distances on the actual house model with a ruler, and compared them with the distances computed from the point coordinates in the shape results. Figure 7.15 shows the selected features superimposed on the first frame of the sequence, with the number assigned to them by our feature detection algorithm. The diagram in figure 7.16 shows the distances between pairs of features, both as measured on the actual model and as computed from the results of our algorithm. The results of the algorithm were scaled so as to make the computed distance between feature 117 and 282 equal to the distance measured on the model. Lengths are in millimeters. The measured distances between the steps along the right side of the roof (7.2 mm) were obtained by measuring five steps and dividing the total distance (36 mm) by five. The differences between computed and measured results are of the order of the resolution of our ruler measurements (one millimeter).

Chapter 7

Conclusion

In this report, we extended our factorization method for the recovery of shape and motion from a stream of images to unrestricted camera motion.

We were initially surprised to observe that the algorithm performs better in three dimensions than in two, in the sense that its performance degrades more gracefully as the number of frames and/or feature points decreases, or as the range of viewing directions becomes smaller.

A posteriori, however, this fact is easy to explain. If the camera motion contains both pitch and yaw, then shape can be recovered from *either* the horizontal or the vertical image coordinates. Computing shape and motion from *both* sets of measurements at once is an additional source of redundancy, which improves the performance. In other words, the measurement matrix is of rank three, but only half of it is necessary in principle. In a related context, the learning of shapes from images, Poggio says it succinctly: "1.5 snapshots are sufficient" [Poggio, 1990] to recover the structure.

In practice, going from two to three dimensions, the additional difficulties in the method are two. First of all, one has to know the aspect ratio of the camera pixels in order to write the measurement matrix. Of course, this was not necessary in the planar case. However, this is a simple calibration to perform.

Secondly, and much more importantly, features are harder to track in a full image than in a single scanline. Our tracking method, based on previous work by Lucas and Kanade [Lucas and Kanade, 1981] yielded accurate displacement measurements. Although there may be situations where the algorithm can fail, we did not encounter any in our experiments. We will discuss the features selection and tracking method in report number 3 of our series.

Bibliography

[Adiv, 1985]
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.

[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.

[Broida *et al.*, 1990]
T. 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.

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

[Debrunner and Ahuja, 1990]
C. H. DEBRUNNER AND N. AHUJA. A direct data approximation based motion estimation algorithm. In *Proceedings of the 10th International Conference on Pattern Recognition*, pages 384–389, Atlantic City, NJ, June 1990.

[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.

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

[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.

[Horn and Weldon, 1988]
B. K. P. Horn and E. J. Weldon Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 1988.

[Horn *et al.*, 1988]
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.

[Jain, 1983]
R. Jain. Direct computation of the focus of expansion. *IEEE Pattern Analysis and Machine Intelligence*, 5:58–63, 1983.

[Lawton, 1983]
D. T. Lawton. Processing translational motion sequences. *Computer Graphics and Image Processing*, 22:116–144, 1983.

[Lucas and Kanade, 1981]
B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981. More details can be found in B. D. Lucas, *Generalized Image Matching by hte Method of Differences*, PhD thesis, Carnegie Mellon University, 1984.

[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.

[Poggio, 1990]
T. Poggio. 3d object recognition: on a result of Basri and Ullman. Technical report, IRST, Trento, Italy, March 1990.

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

[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, 1990a]
C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. Technical Report CMU-CS-90-166, Carnegie Mellon University, Pittsburgh, Pa, September 1990.

[Tomasi and Kanade, 1990b]
C. Tomasi and T. Kanade. Shape and motion without depth. In *Proceedings of the Third International Conference in Computer Vision (ICCV)*, Osaka, Japan, December 1990.

[Tsai and Huang, 1981]
R. Y. Tsai and T. S. Huang. Estimating three-dimensional motion parameters of a rigid planar patch. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-29:1147–1152, December 1981.

[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.

[Tsai *et al.*, 1982]
R. Y. Tsai, T. S. Huang, and W. L. Zhu. Estimating three-dimensional motion parameters of a rigid planar patch, ii: Singular value decomposition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30:525–534, August 1982.

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

[Ullman, 1984]
S. Ullman. Maximizing rigidity: the incremental recovery of 3-d structure from rigid and rubbery motion. *Perception*, 13:255–274, 1984.

[Waxman and Wohn, 1985]
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.

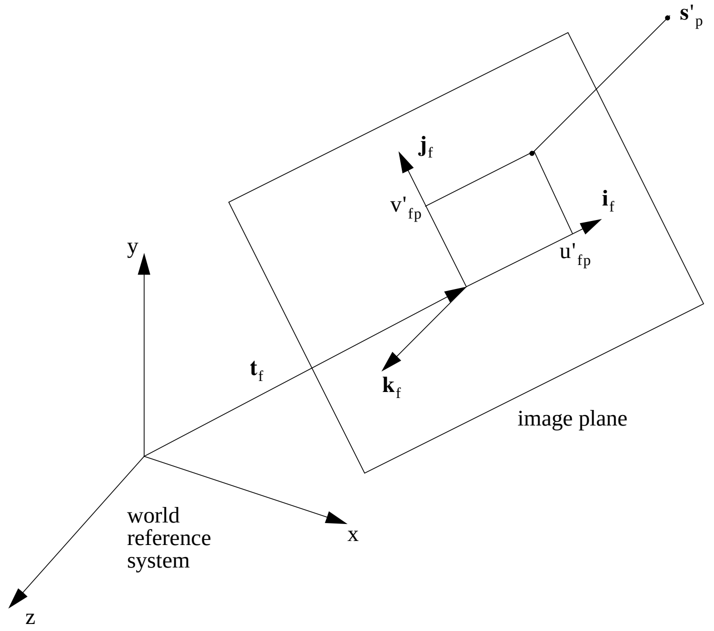


Figure 7.1: The image and world reference systems.

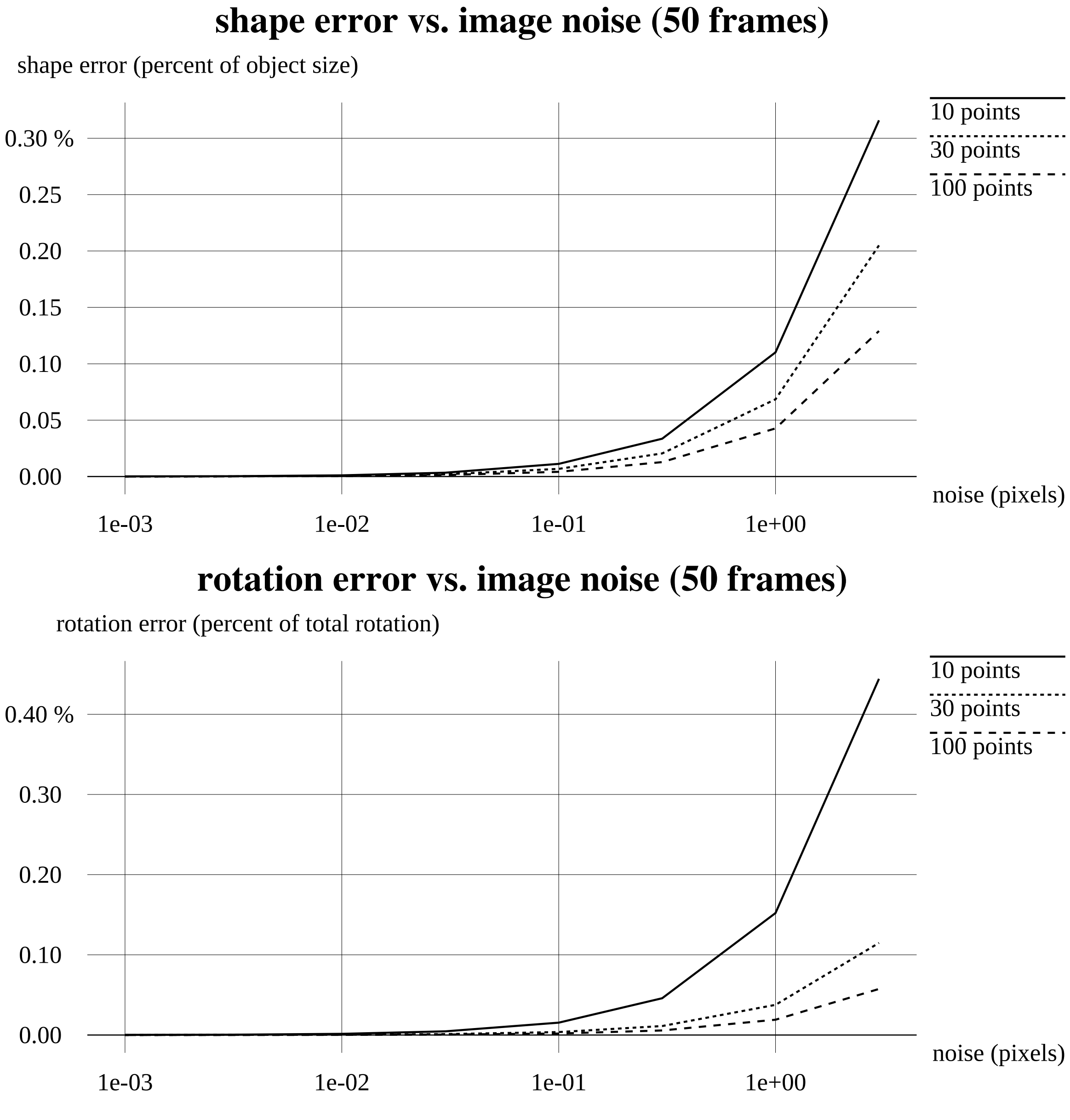


Figure 7.2: Relative shape and rotation error versus image noise for three different point-set sizes. Gaussian noise standard deviation figures assume a 512×512 image.

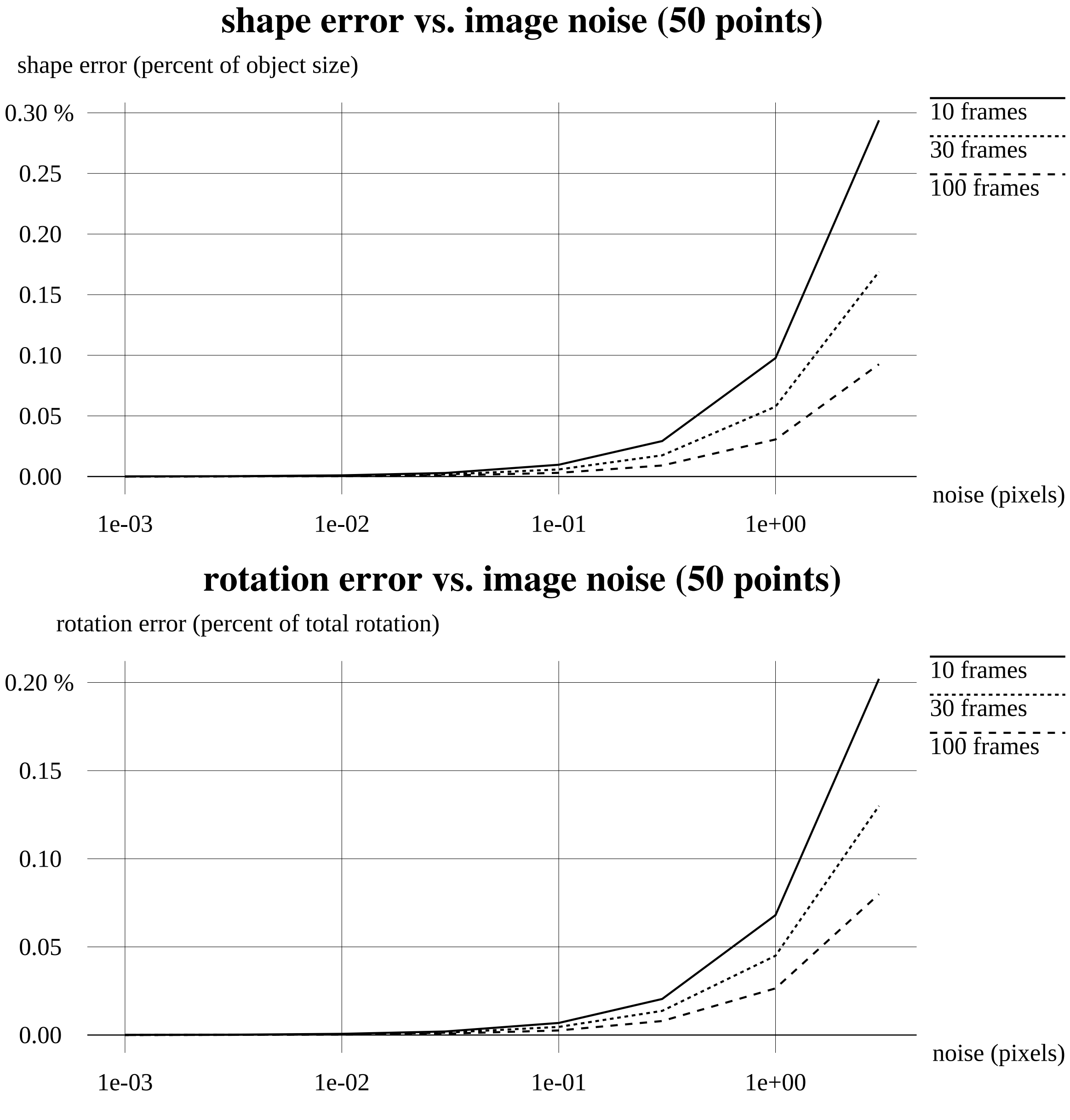


Figure 7.3: Relative shape and rotation error versus image noise for three different stream lengths. Gaussian noise standard deviation figures assume a 512×512 image. Shorter streams correspond to greater motions between frames. See text for details.

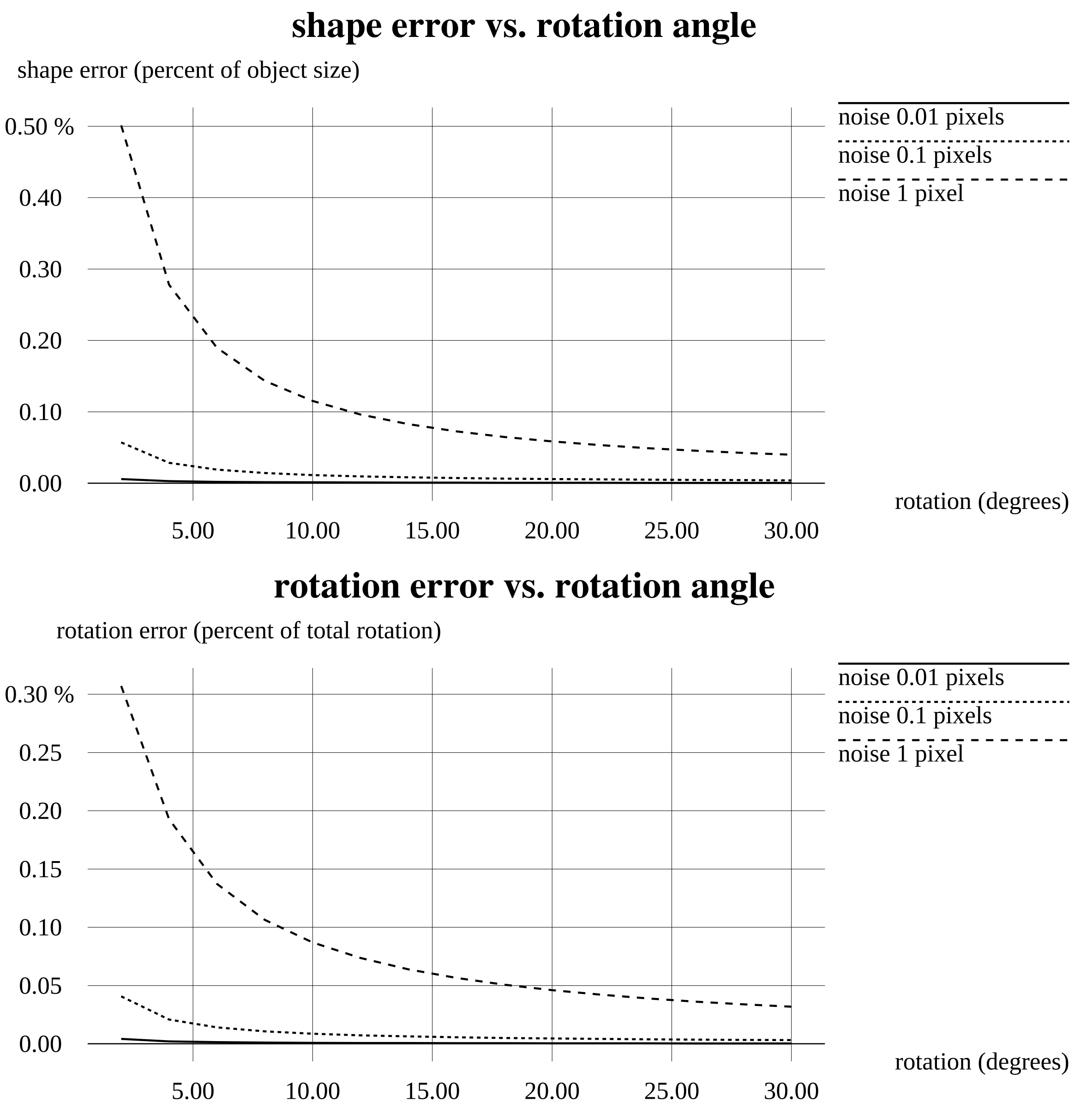


Figure 7.4: Relative shape and rotation error versus total camera rotation angle for various noise levels. Gaussian noise standard deviation figures assume a 512×512 image. See text for details.

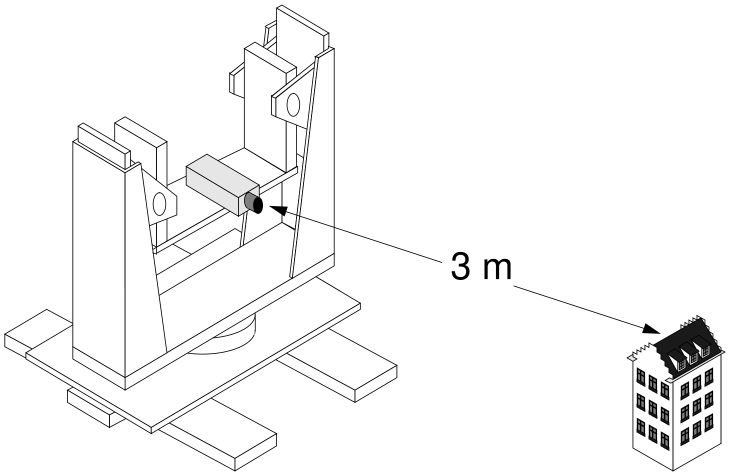


Figure 7.5: The setup used in the experiment. The drawing is not to scale.

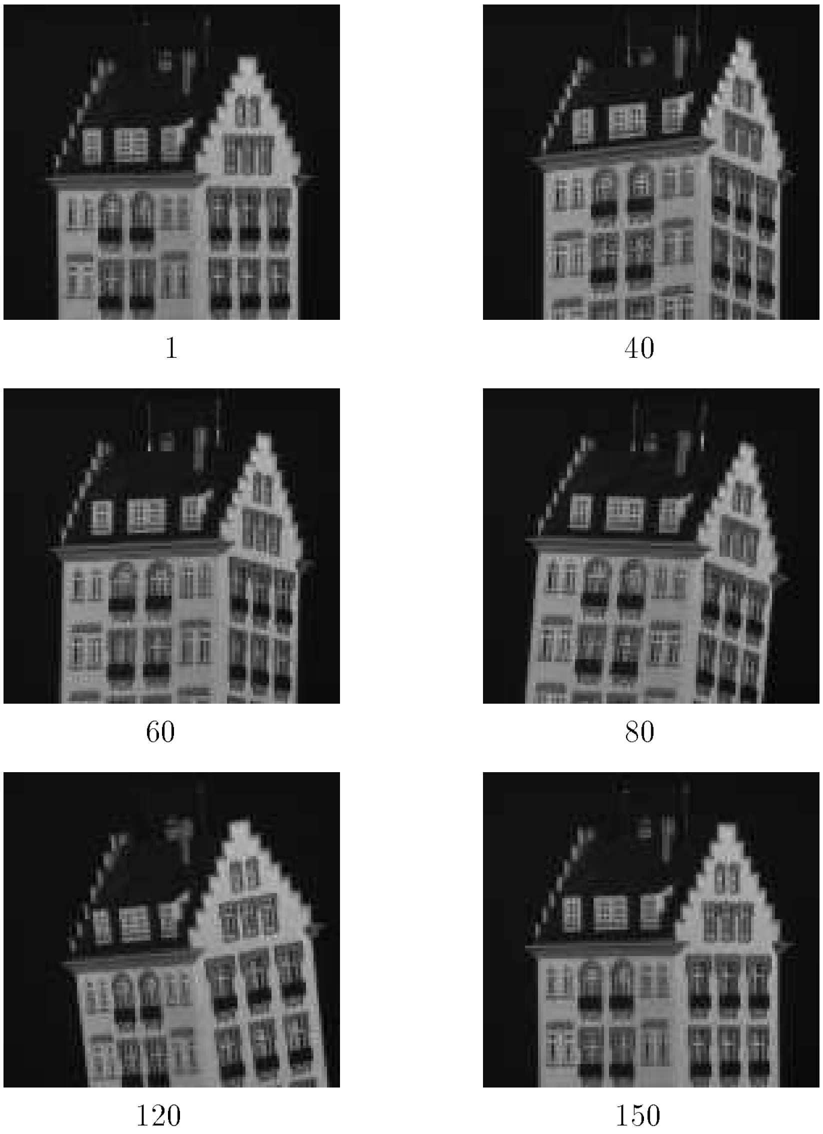


Figure 7.6: Some frames in the stream.

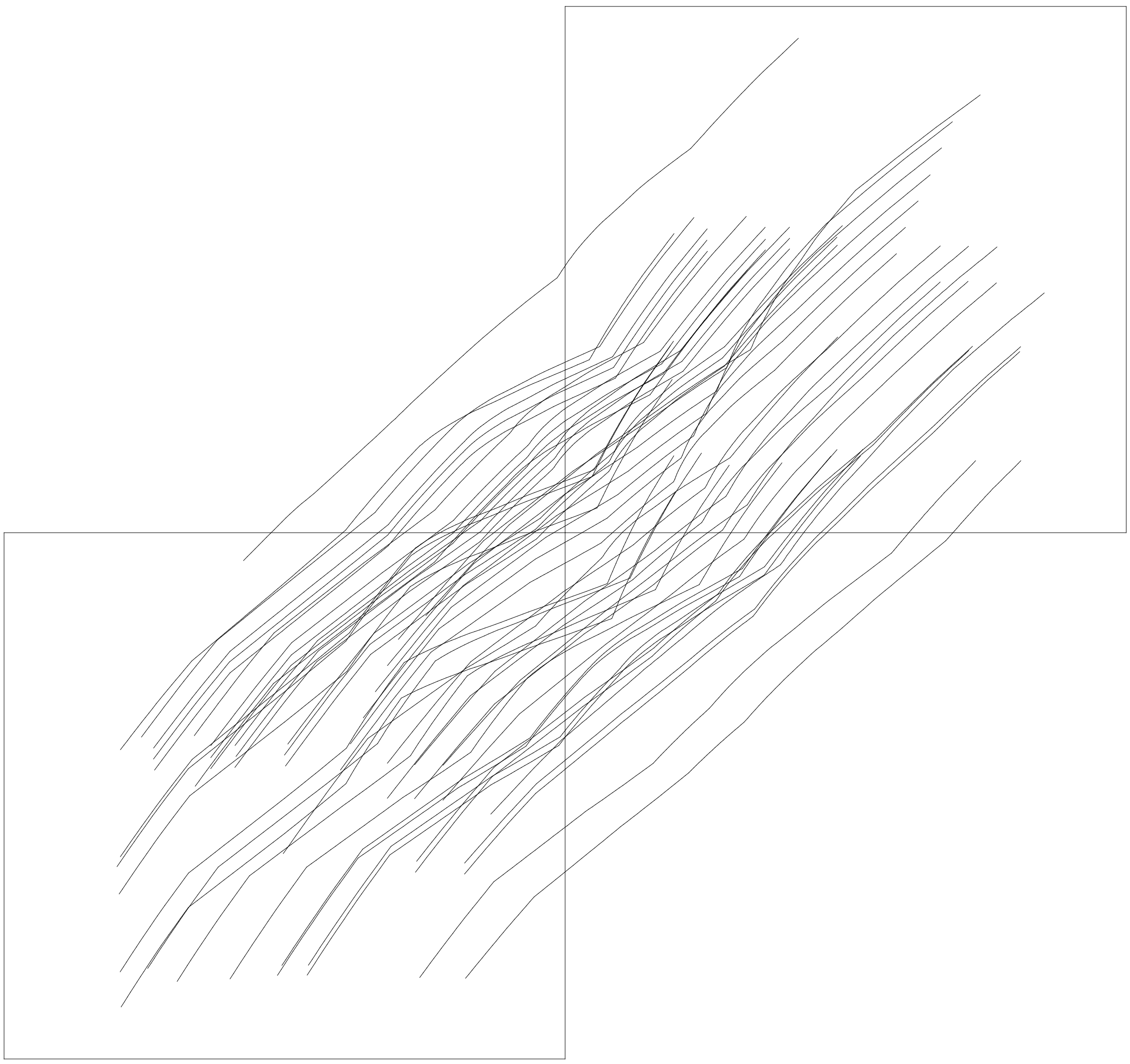


Figure 7.7: Tracks of 50 randomly selected features, from frame 1 to frame 151 of a real stream.

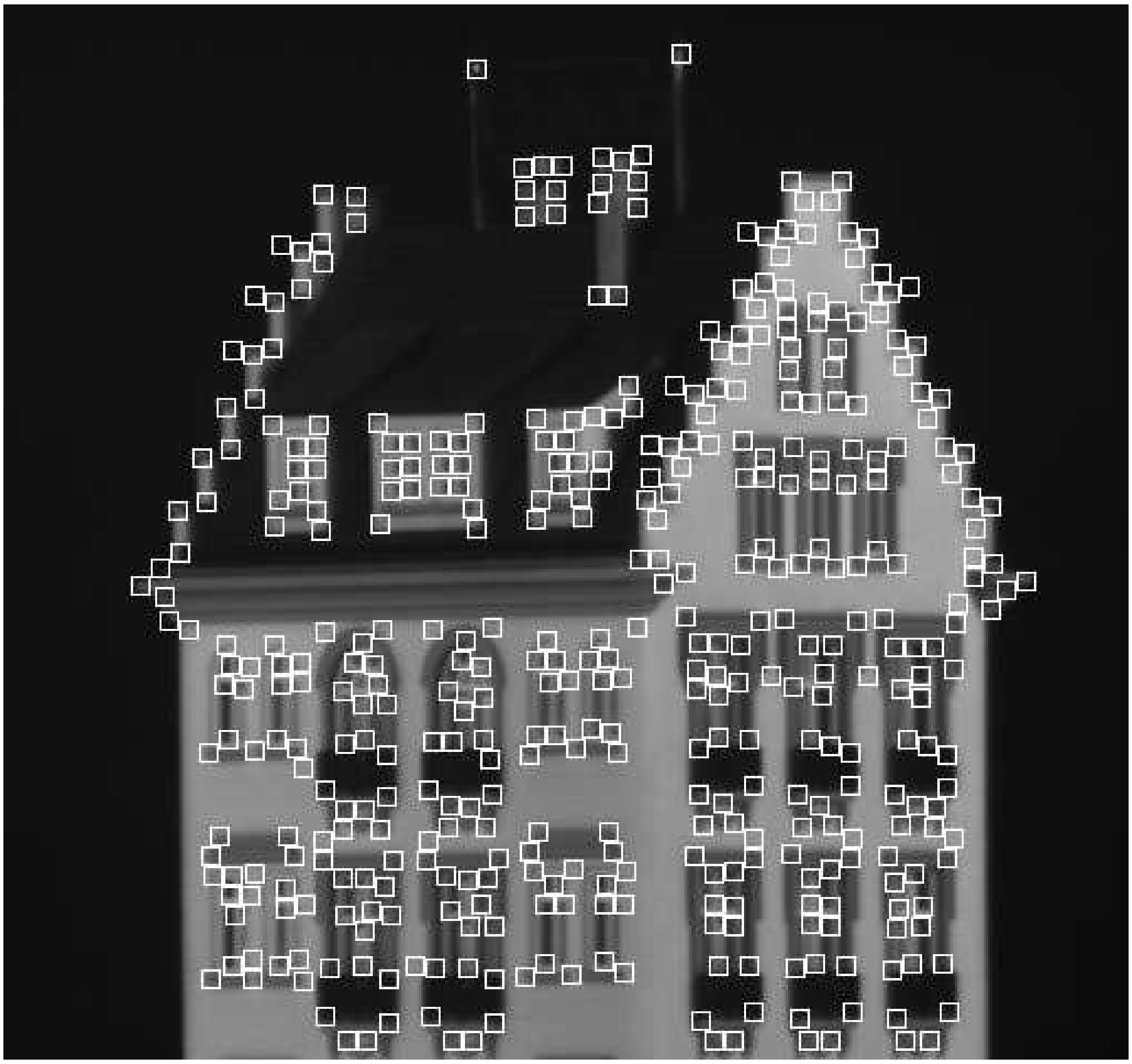


Figure 7.8: The 430 features selected by the automatic detection method, shown superimposed on the first frame of the stream.

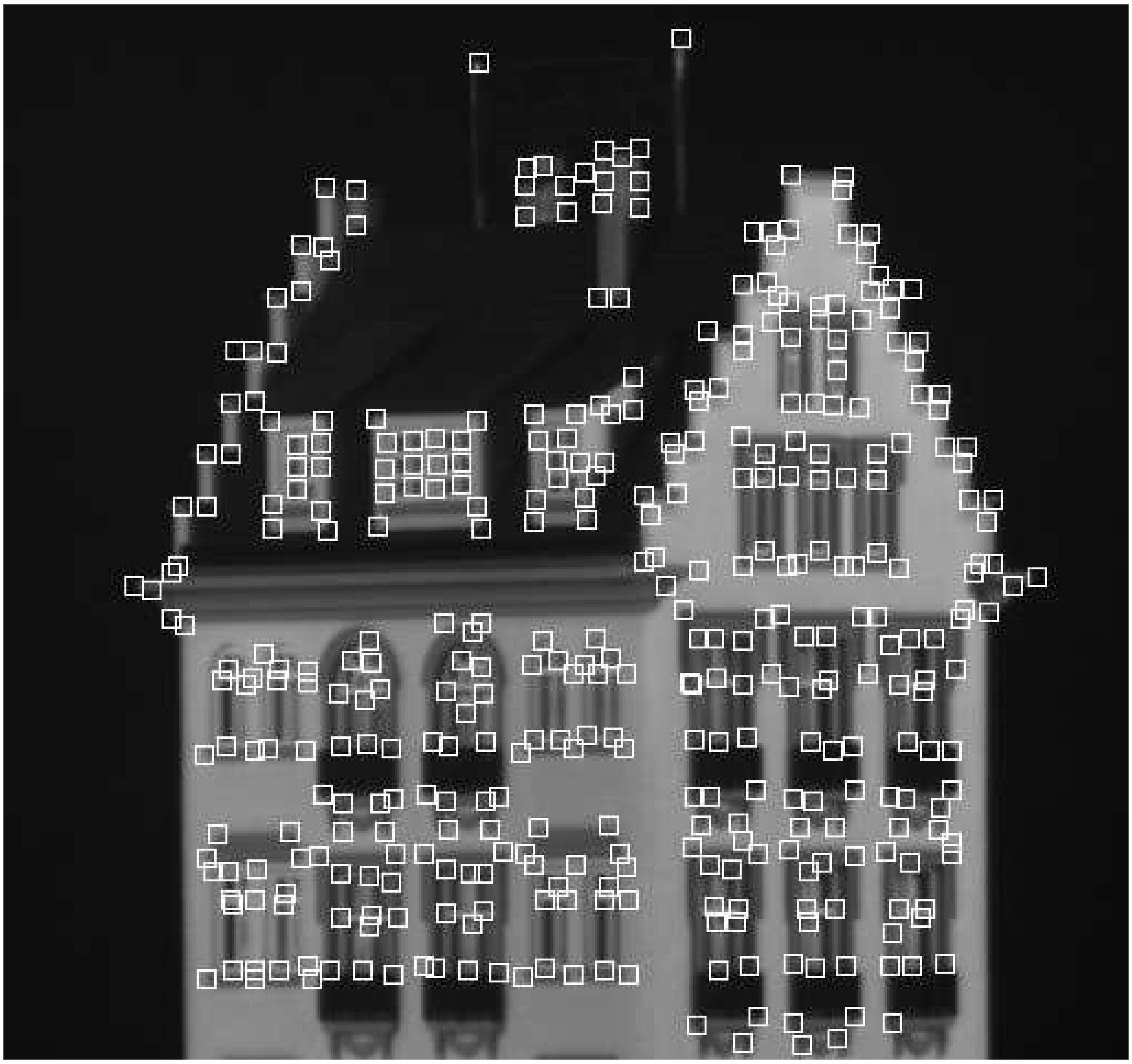


Figure 7.9: The 388 features surviving tracking through 150 frames, shown superimposed on the last frame of the stream.

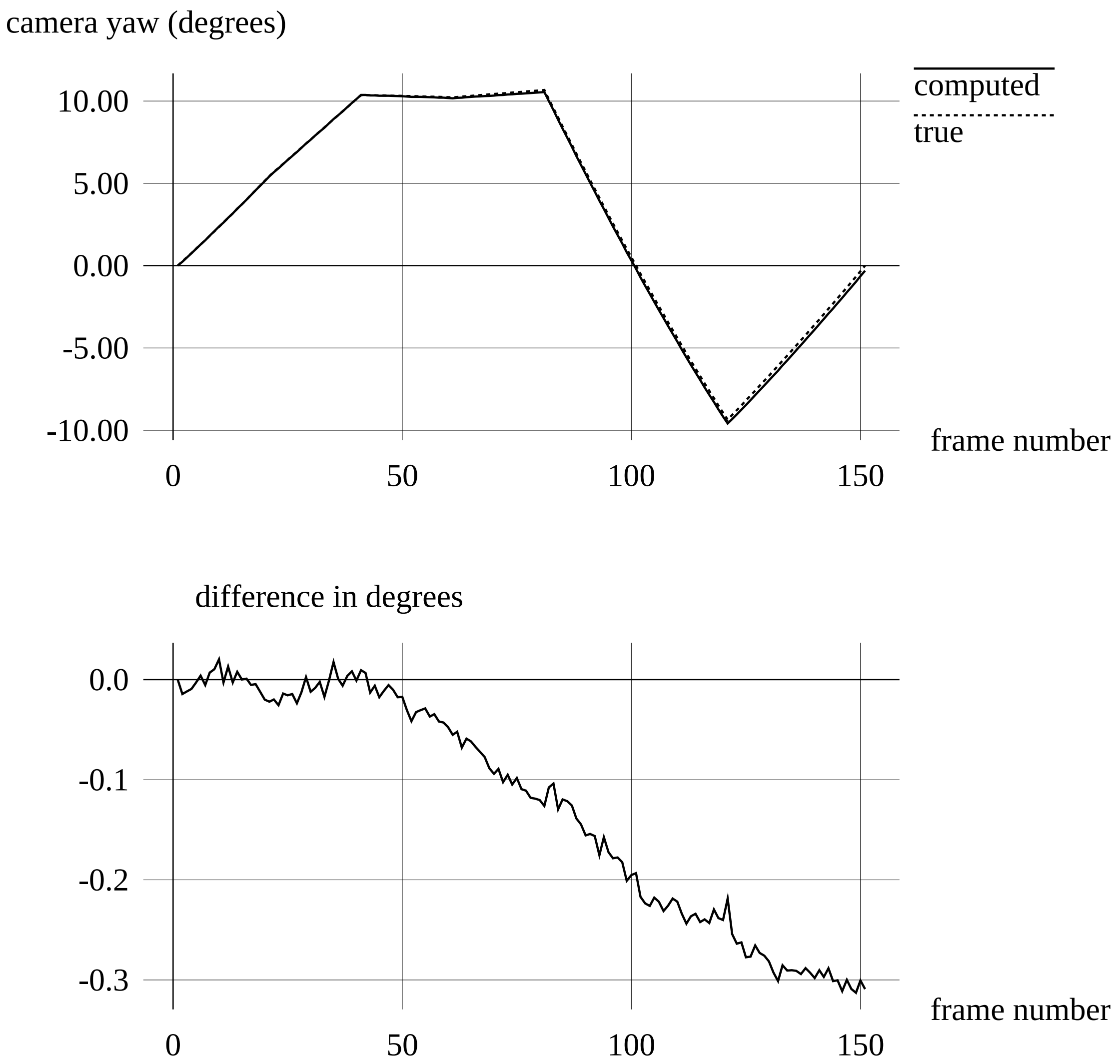
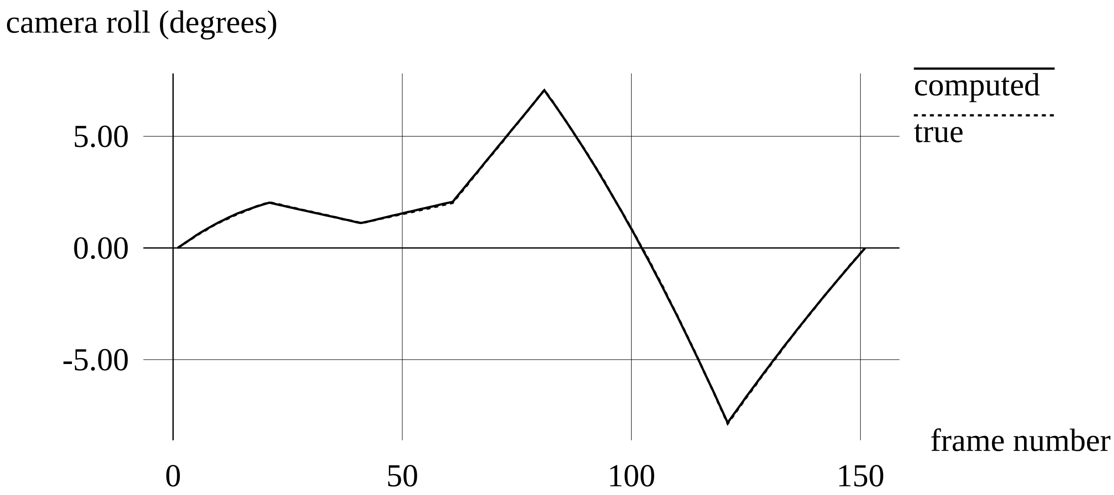


Figure 7.10: Results for the camera yaw. The top graph shows true and computed camera yaw, superimposed, versus the frame number. The bottom graph is a blow-up of the difference between the two plots. The rotation is around the centroid of the feature points.



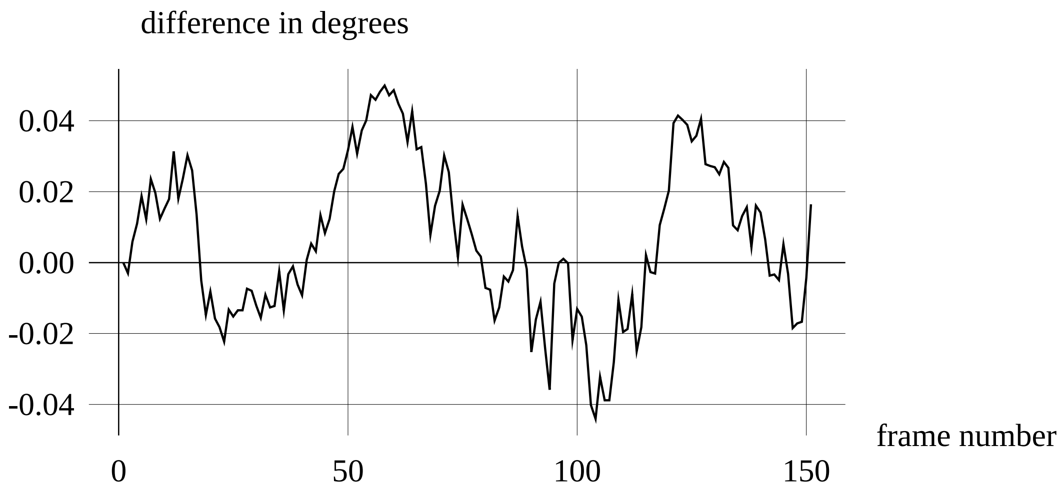


Figure 7.11: Results for the camera roll. The top graph shows true and computed camera roll, superimposed, versus the frame number. The bottom graph is a blow-up of the difference between the two plots. The rotation is around the centroid of the feature points.

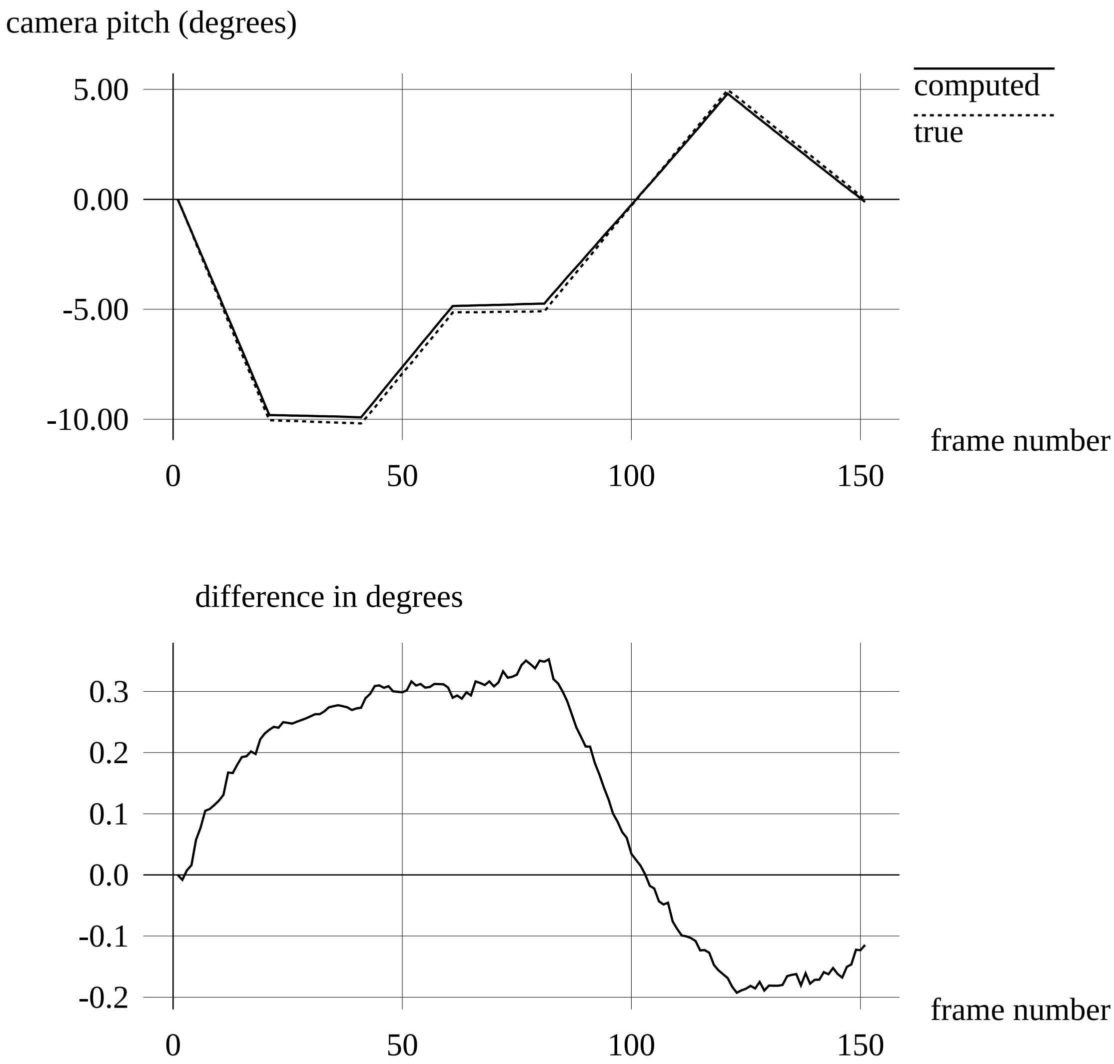


Figure 7.12: Results for the camera pitch. The top graph shows true and computed camera pitch, superimposed, versus the frame number. The bottom graph is a blow-up of the difference between the two plots. The rotation is around the centroid of the feature points.

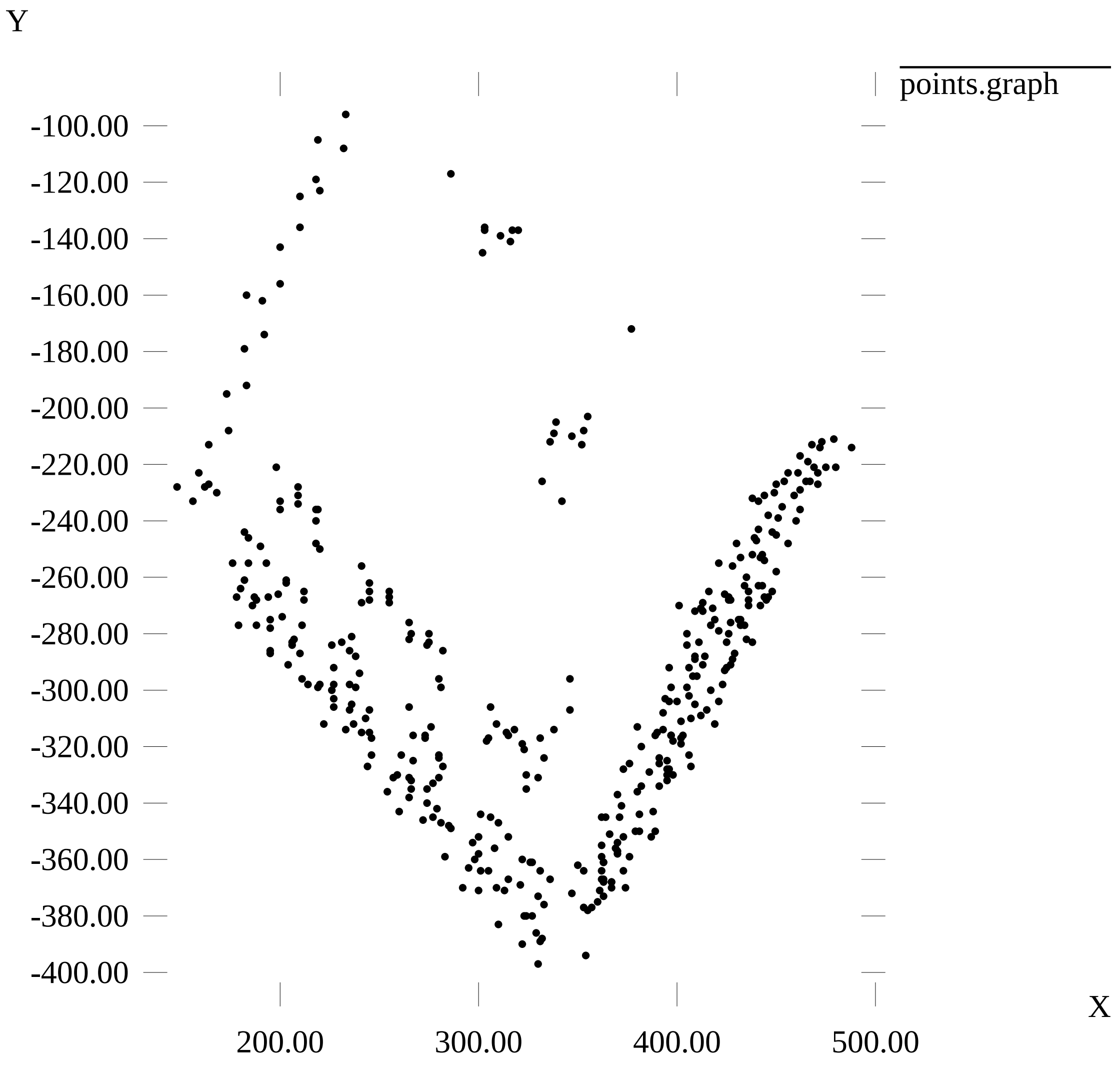


Figure 7.13: A view of the computed shape from approximately above the building (compare with figure 7.14). Notice the correct location of the walls, the windows on the roof, and the chimneys.

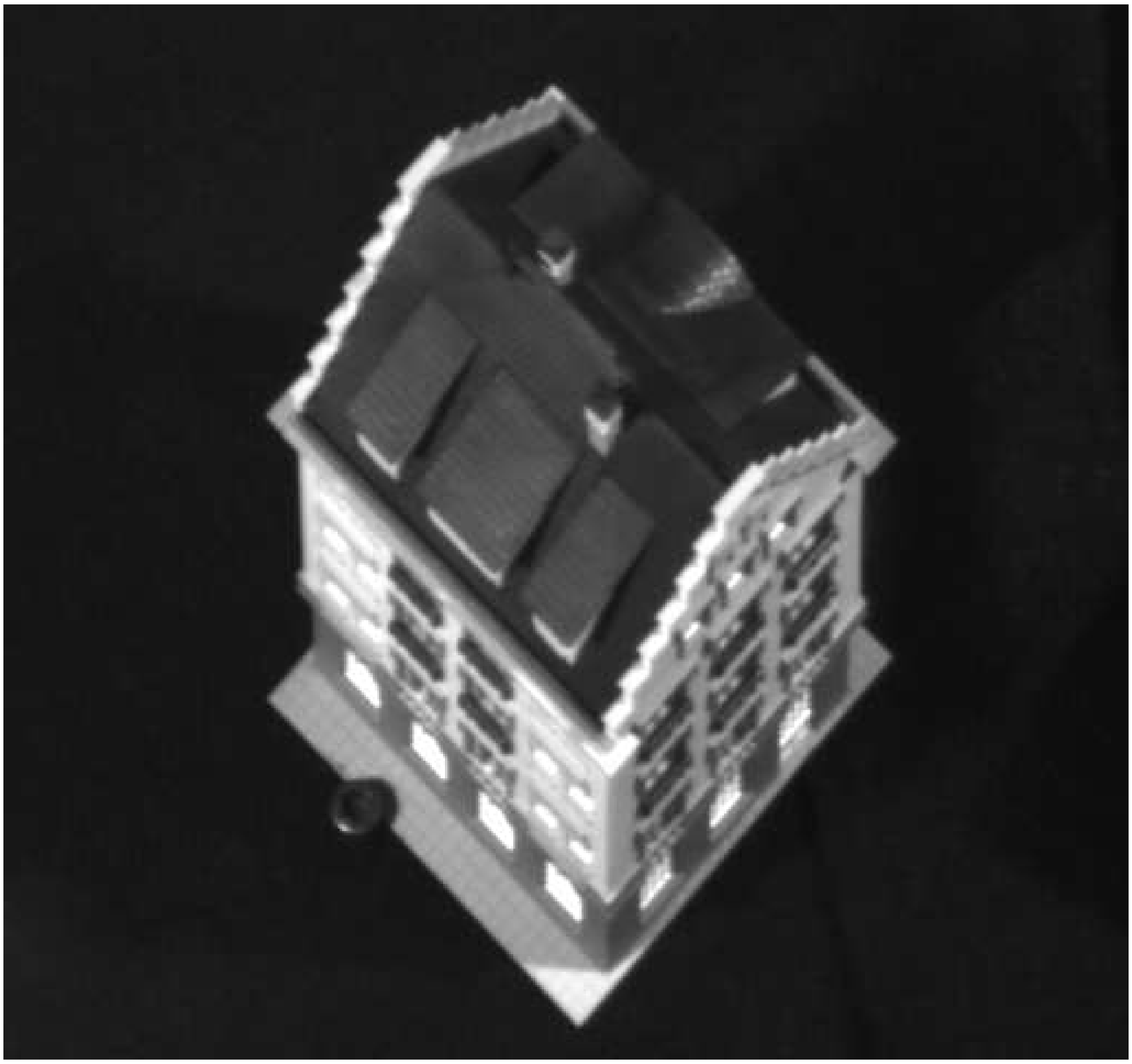


Figure 7.14: A real picture from above the building, similar to figure 7.13. This figure and figure 7.13 were not precisely aligned, but are intended for qualitative comparison.

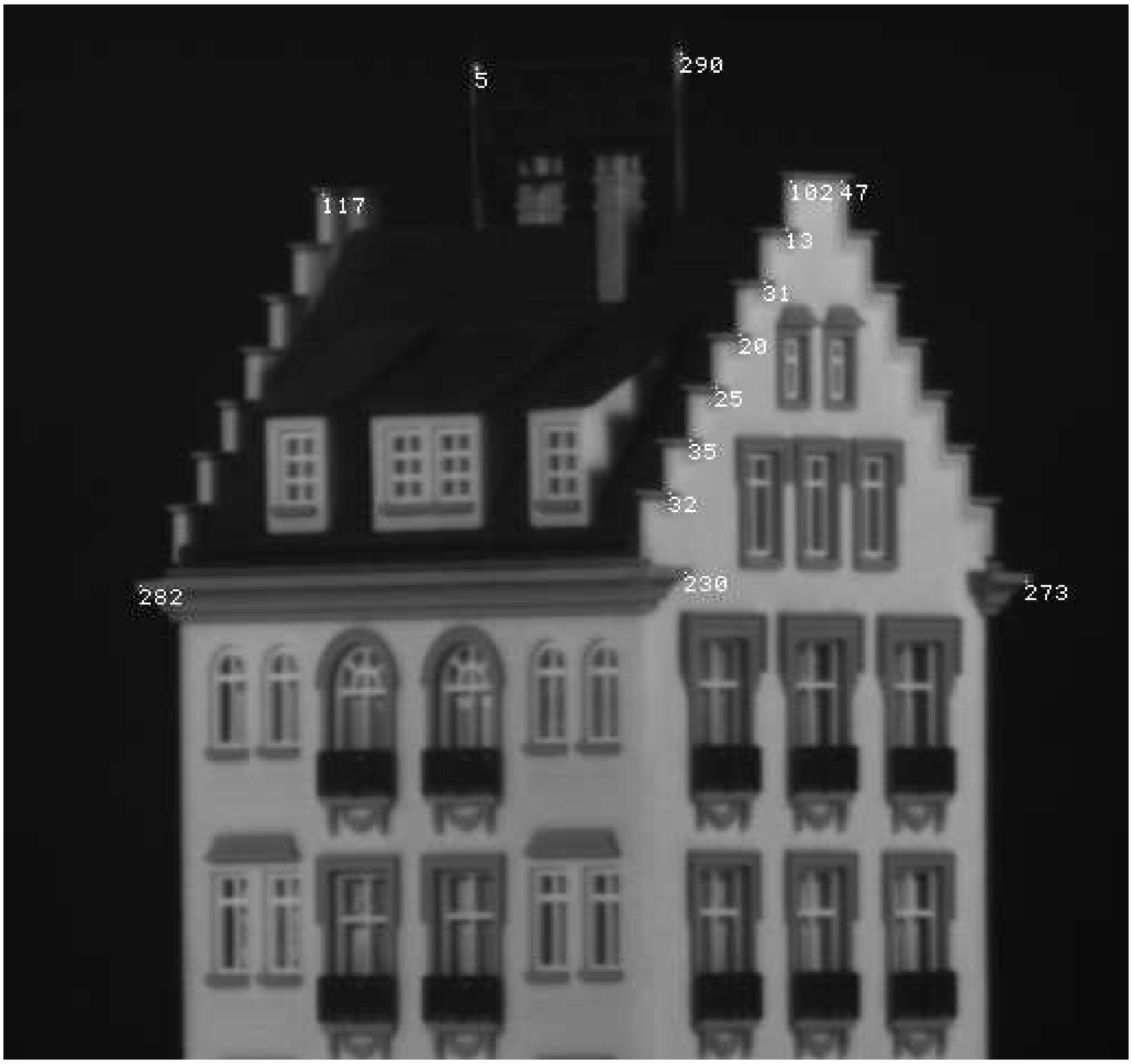


Figure 7.15: For a quantitative evaluation, distances between the features show in the picture were measured on the actual model, and compared with the computed results. The comparison is shown in figure 7.16.

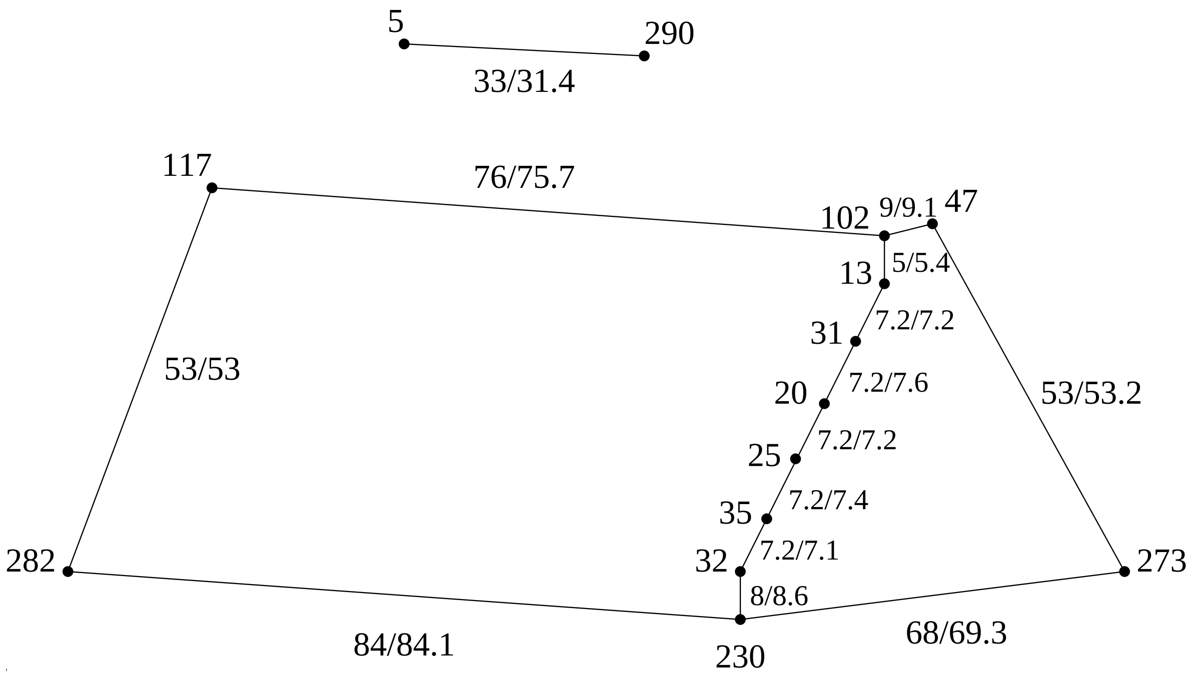


Figure 7.16: Comparison between measured and computed distances for the features in figure 7.15. The number before the slash is the measured distance, the one after is the computed distance. Lengths are in millimeters. Computed distances were scaled so that the computed distance between features 117 and 282 is the same as the measured distance.