Factoring Image Sequences into Shape and Motion

Carlo Tomasi and Takeo Kanade
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Recovering scene geometry and camera motion from a sequence of images is an important problem in computer vision. If the scene geometry is specified by depth measurements, that is, by specifying distances between the camera and feature points in the scene, noise sensitivity worsens rapidly with increasing depth.

In this paper, we show that this difficulty can be overcome by computing scene geometry directly in terms of shape, that is, by computing the coordinates of feature points in the scene with respect to a world-centered system, without recovering camera-centered depth as an intermediate quantity.

More specifically, we show that a matrix of image measurements can be factored by Singular Value Decomposition into the product of two matrices that represent shape and motion, respectively.

The results in this paper extend to three dimensions the solution we described in a previous paper for planar camera motion.

1 Introduction

Recovering scene geometry and camera motion from a sequence of images is an important problem in computer vision. It admits a solution [18], [13] for perfect images, but is very sensitive to noise [2]. In this paper, we observe that this sensitivity is due in part to the representation of shape as a depth map, and show that by reformulating the problem in world-centered coordinates can lead to a simpler and better-behaved solution.

In Ullman's proof of existence of a solution [18], as well as in the perspective formulation in [13], the coordinates of feature points in the world are expressed in a world-centered coordinate system.

However, this choice has been replaced by most computer vision researchers with that of a retinotopic, or camera-centered, representation of shape [12], [5], [17], [1], [19], [3], [9], [7], [8], [11], [14], [4]. With this representation, the position of feature points is specified by their image coordinates and their depths, defined as the distances between the camera center and the feature points, measured along the optical axis.

Unfortunately, although a camera-centered representation simplifies the equations for perspective projection, it makes shape estimation harder and, for increasingly distant scenes, eventually impossible. This is due to two reasons. First, the computation of shape via depth is very sensitive to noise for remote objects: since even large changes in depth produce small changes in the image, computing small depth differences from image variations is virtually impossible with any amount of image noise.

Second, as the camera moves, the camera-centered feature coordinates change. This leads to the difficult problem of relating depth values in different camera coordinate systems to each other in the presence of motion uncertainties (see for instance [11], [8]).

In this paper, we show that both difficulties disappear if feature corodinates are expressed with respect to a world-centered frame. With this formulation, object-centered shape can be linked to image motion directly, without using retinotopic depth as an intermediate quantity.

Furthermore, the mutual independence of shape and motion in world-centered coordinates makes it possible to cast structure-from-motion as a factorization problem, in which a matrix representing image measurements is decomposed directly into camera rotation and object shape.

More specifically, an image sequence 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 theorem*: under orthography, the measurement matrix is of rank 3. As a consequence of this theorem, we show that the measurement matrix can be factored into the product of two matrices of size

TH0390-5/91/0000/0021/$01.00 © 1991 IEEE

$2F \times 3$ and $3 \times P$, respectively, where the first matrix encodes camera rotation, the second shape.

The rank theorem captures precisely the nature of the redundancy of an image sequence, and allows dealing with a large number of points and frames in a conceptually simple and computationally efficient way to reduce the effects of noise. The resulting algorithm is based on Singular Value Decomposition, which is numerically well-behaved and stable.

We first introduced this factorization method in [15], where we treated the simple case of single-scanline images in a flat, two-dimensional world. We now develop the idea into a working system for arbitrary camera motion in three dimensions, and full, two-dimensional images.

In the next section we show how to build the measurement matrix from an image sequence, prove that the measurement matrix is of rank 3, and show how to use this result to factor the measurement matrix into shape and camera rotation. Section 3 describes an illustrative experiment on a real image sequence. To reduce the printing cost of extra pages in these proceedings, we refer to [16] for a discussion of the relation of our work with relevant results in the literature.

2 The Factorization Method

In the next Subsection, we show how to represent an image stream as a measurement matrix collecting the feature coordinates to be fed to the algorithm that computes shape and motion. We then introduce the main result on the rank of the measurement matrix in the absence (Subsection 2.2) and presence (Subsection 2.3) of noise. Subsection 2.4 shows that the motion and shape result is essentially unique, and Subsection 2.5 summarizes the factorization method.

To track features from frame to frame, we used a method based on [10], which we extended to allow for the automatic selection of features. The description of both detection and tracking are beyond the scope of this paper.

2.1 The Measurement Matrix

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

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.

The horizontal feature coordinates u'_{fp} 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:

$$\begin{aligned} u_{fp} &= u'_{fp} - \overline{u}_f \\ v_{fp} &= v'_{fp} - \overline{v}_f \ , \end{aligned} \tag{1}$$

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.

2.2 The Rank Theorem

We now analyze the relation between camera motion, shape, and the entries of the measurement matrix W. This analysis leads to the key result that W is highly rank-deficient (the rank theorem).

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. 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 origin of the camera reference system is at the center of the image.

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 (1). For the horizontal coordinates we have

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

where $\frac{1}{P}\sum_{q=1}^{P} \mathbf{s}'_q$ 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 compute the visible components of translation, and remove them from the projection equations.

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

$$\begin{aligned} u_{fp} &= \mathbf{i}_f \cdot \mathbf{s}_p \\ v_{fp} &= \mathbf{j}_f \cdot \mathbf{s}_p \ , \end{aligned} \tag{2}$$

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.

Because of the two sets of $F \times P$ equations (2), the measurement matrix W can be expressed in a matrix form:

$$W = MS \tag{3}$$

where

$$M = \begin{bmatrix} \mathbf{i}_1^T \\ \vdots \\ \mathbf{i}_F^T \\ \mathbf{j}_1^T \\ \vdots \\ \mathbf{j}_F^T \end{bmatrix} \tag{4}$$

represents the camera motion, and

$$S = \begin{bmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_P \end{bmatrix} \tag{5}$$

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 (3) implies the following **rank theorem**.

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

The rank theorem expresses the 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.

When noise corrupts the images, the measurement matrix W will not be exactly of rank 3. However, the rank theorem can be extended to the case of noisy measurements in a well-defined manner. The next subsection introduces this extension, using the concept of Singular Value Decomposition [6] to introduce the notion of approximate rank.

2.3 Approximate Rank

Assuming [1] that $2F \geq P$, the matrix W can be decomposed [6] into a $2F \times P$ matrix L, a diagonal $P \times P$ matrix Σ, and a $P \times P$ matrix R,

$$W = L\Sigma R \ , \tag{6}$$

such that $L^T L = R^T R = RR^T = \mathcal{I}$, 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.

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

$$\begin{aligned} L &= \underbrace{\left[\ L' \ \middle|\ L''\ \right]}_{3 \quad P-3} \}2F \\ \Sigma &= \underbrace{\left[\begin{array}{c|c} \Sigma' & 0 \\ \hline 0 & \Sigma'' \end{array}\right]}_{3 \quad P-3} \begin{array}{l} \}3 \\ \}P-3 \end{array} \\ R &= \underbrace{\left[\begin{array}{c} R' \\ \hline R'' \end{array}\right]}_{P} \begin{array}{l} \}3 \\ \}P-3 \end{array} \ , \end{aligned} \tag{7}$$

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.

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

Because of the rank theorem, 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 **rank theorem 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 (7). With the definitions

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

we can also write

$$\hat{W} = \hat{M}\hat{S} . \tag{8}$$

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 decomposition (8) is 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. Indeed, 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 theorem, 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 ratio between the third and the fourth largest singular values of W be sufficiently large.

2.4 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

$$\begin{aligned} M &= \hat{M}A \\ S &= A^{-1}\hat{S} . \end{aligned} \tag{9}$$

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

$$\begin{aligned} \hat{\mathbf{i}_f}^T AA^T\hat{\mathbf{i}_f} &= 1 \\ \hat{\mathbf{j}_f}^T AA^T\hat{\mathbf{j}_f} &= 1 \\ \hat{\mathbf{i}_f}^T AA^T\hat{\mathbf{j}_f} &= 0 \end{aligned} \tag{10}$$

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,

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

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.

2.5 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) and (5) can be computed as follows.

1. Compute the singular-value decomposition $W = L\Sigma R$.

2. Define $\hat{M} = L'(\Sigma')^{1/2}$ and $\hat{S} = (\Sigma')^{1/2}R'$, where the primes refer to the block partitioning defined in (7).

3. Compute the matrix A in equations (9) by imposing the metric constraints (equations (10)).

4. Compute the motion matrix M and the shape matrix S as $M = \hat{M}A$ and $S = A^{-1}\hat{S}$.

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 [9].

3 An Experiment

In this chapter, we illustrate the factorization method with an experiment on a real sequence of images. The images depict 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. Some frames in the sequence are shown in figure 1. Camera pitch, yaw, and roll around the model are all varied as shown by the dashed curves in figure 2. The translation of the camera is such as to keep the building withing the field of view throughout the sequence.

For feature tracking, we extended the method described in [10] to allow also for the automatic selection of image features. The entire set of 430 features is displayed in figure 3, overlaid on the first frame of the sequence. Of these features, 42 were abandoned during tracking because their appearance changed too much. The remaining 388 features are used in the computation of shape and motion.

The plots in figure 2 compare the rotation components computed by the algorithm (solid curves) with the values measured mechanically from the mobile platform (dashed curves). The differences are magnified in figure 4.

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 (second plot in figure 2), 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.

Between frames 60 and 80, yaw and pitch are nearly constant. This means that the image sequence 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 sequence *as a whole* to be taken during non-degenerate motion. The algorithm can deal without difficulty with sequences that contain degenerate subsequences, because the information in the sequence is used all at once in our method.

The shape results are shown qualitatively in figure 5, which shows the computed shape viewed from above. The view in figure 5 is similar to that in figure 6, 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 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 8 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).

In order to be able to measure the ground truth even more precisely than is possible in laboratory experiments, we studied the performance of our method with a series of simulations, in which we corrupted the image measurements with Gaussian noise. The findings indicate that even for noise levels as high as three pixels standard deviation the algorithm converges to within 1 percent of the correct motion and shape estimates, provided that there are at least, say, fifty

frames and fifty features, and that the camera rotates at least five degrees around the scene. More details about the simulations can be found in [16].

4 Conclusion

Formulating the structure-from-motion problem in terms of shape and motion, rather than depth and motion, has two important advantages. First, shape is no more computed by taking small differences between large depth values, but is directly related to the image displacements. This greatly improves the conditioning of the problem, yielding both better shape and better motion estimates.

Second, with the shape-and-motion formulation we compute two mutually independent quantities: while depth is a function of both scene geometry and camera motion, shape and motion are independent of each other. This key observation leads to our factorization method. Writing an image sequence as the product of motion and shape results into a well-behaved algorithm that capitalizes on the intrinsic redundancy of the sequence to achieve good performance in the presence of noise.

References

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

[2] J. L. Barron, A. D. Jepson, and J. K. Tsotsos. The feasibility of motion and structure from noisy time-varying image velocity information. *IJCV*, 5(3):239–269, 1990.

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

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

[5] A. R. Bruss and B. K. P. Horn. Passive navigation. *CVGIP*, 21:3–20, 1983.

[6] G. H. Golub and C. Reinsch. *Singular Value Decomposition and Least Squares Solutions*, volume 2, chapter I/10, pages 134–151. Springer Verlag, New York, NY, 1971.

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

[8] J. Heel. Dynamic motion vision. In *DARPA IU Workshop*, pages 702–713, Palo Alto, CA, 1989.

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

[10] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *7th Intl. Joint Conf. on Artificial Intelligence*, 1981.

[11] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *IJCV*, 3(3):209–236, 1989.

[12] K. Prazdny. Egomotion and relative depth from optical flow. *Biol. Cyb.*, 102:87–102, 1980.

[13] J. W. Roach and J. K. Aggarwal. Computer tracking of objects moving in space. *IEEE Trans. on PAMI*, PAMI-1(2):127–135, 1979.

[14] M. E. Spetsakis and J. Aloimonos. Optimal motion estimation. In *IEEE Workshop on Visual Motion*, pages 229–237, Irvine, CA, 1989.

[15] C. Tomasi and T. Kanade. Shape and motion without depth. In *ICCV*, Osaka, Japan, 1990.

[16] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method - 2. point features in 3d motion. Technical Report CMU-CS-91-105, Carnegie Mellon University, Pittsburgh, PA, January 1991.

[17] Roger Y. Tsai and Thomas S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. on PAMI*, PAMI-6(1):13–27, 1984.

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

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

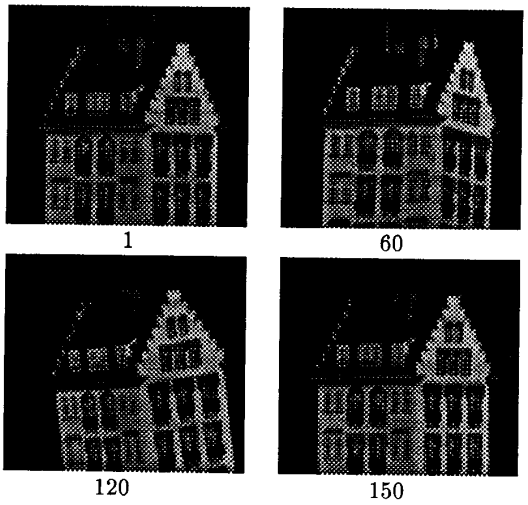

Figure 1: Some frames in the sequence. The whole sequence is 150 frames.

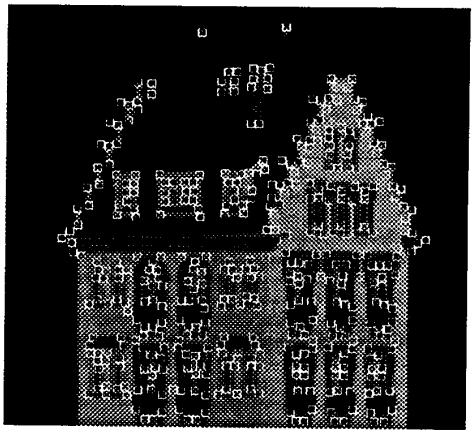

Figure 3: The 430 features selected by the automatic detection method.

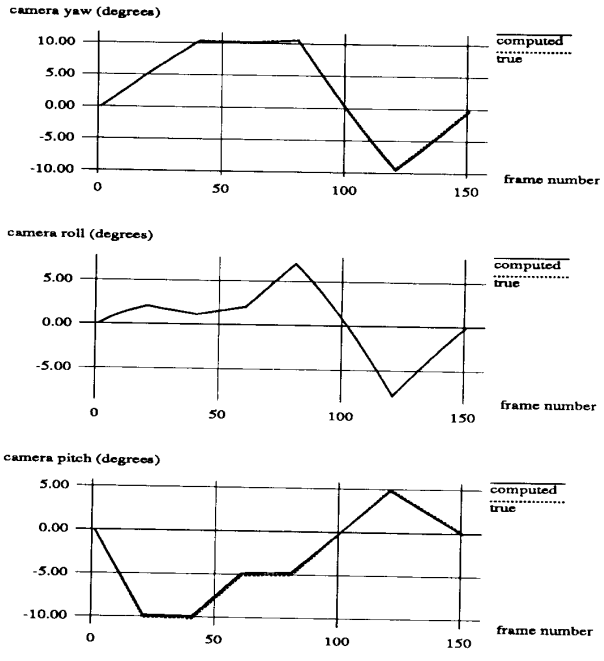

Figure 2: True and computed camera yaw, roll, pitch.

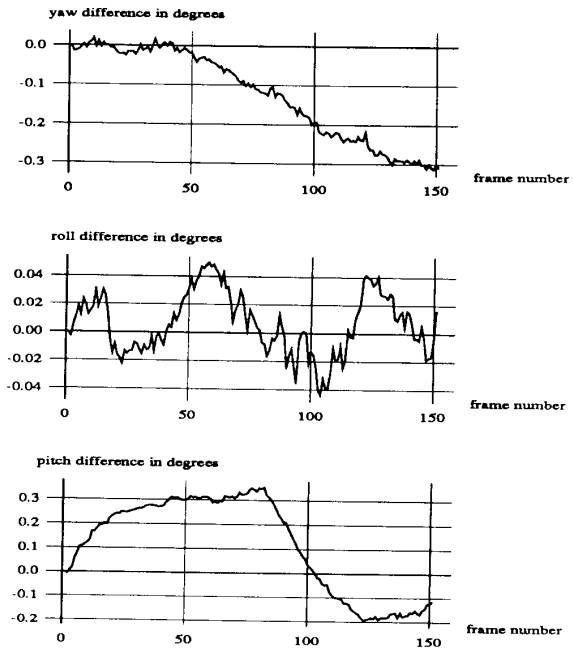

Figure 4: Blow-up of the errors in figure 2.

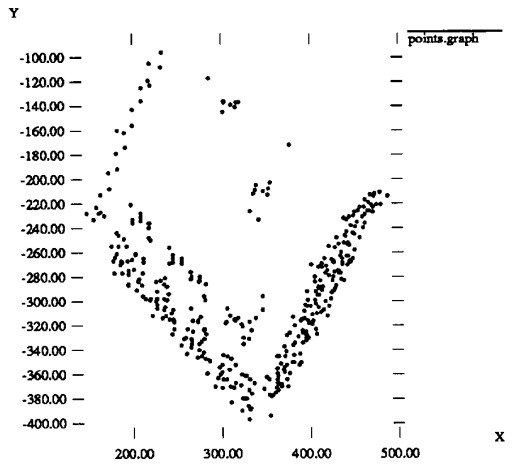

Figure 5: A view of the computed shape from approximately above the building (compare with figure 6).

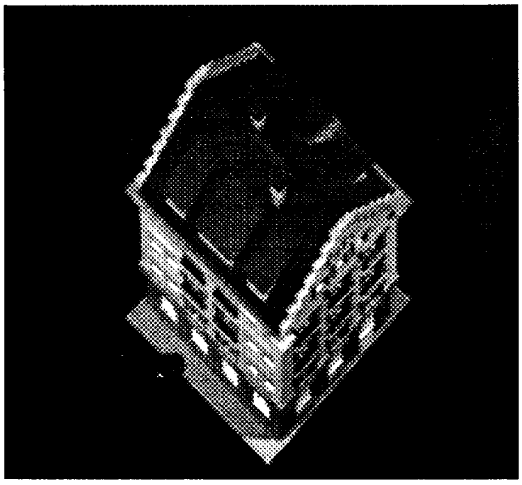

Figure 6: A real picture from above the building, similar to figure 5.

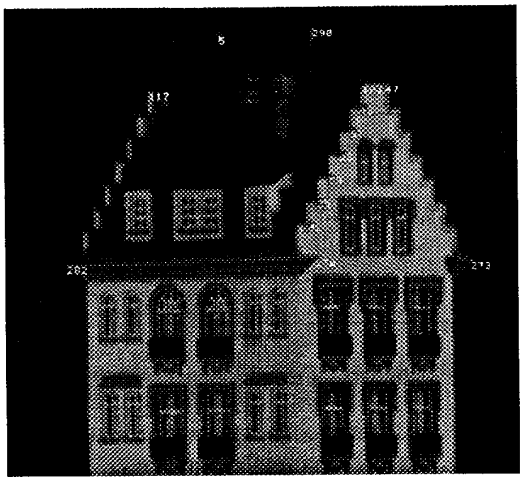

Figure 7: For a quantitative evaluation, distances between the features show in the picture were measured on the actual model, and compared with the computed results. The comparison is shown in figure 8.

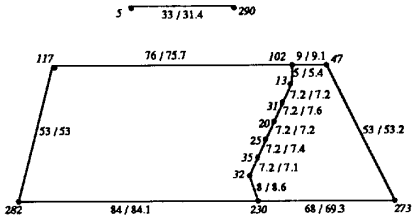

Figure 8: Comparison between measured and computed distances for the features in figure 7. The number before the slash is the measured distance, the one after is the computed distance. Lengths are in millimeters. Computed distances were scaled so that the computed distance between features 117 and 282 is the same as the measured distance.