Minimal Metric Structure and Motion from Three Affine Images

Marc-André Ameller **Adrien Bartoli** **Long Quan**
INRIA Rhône-Alpes, 655, av. de l'Europe
38334 St. Ismier cedex, France. *first.last@inria.fr*

Abstract

Structure and motion from minimal data is essential to bootstrap robust methods based on random sampling such as RANSAC or LMS. Let us consider the affine camera model and make the hypotheses of zero skew and unity aspect ratio. In this case, at least 4 points in 3 images are necessary to recover structure and motion. We propose a parametrization based on metric structure rather than camera motion parameters which have been previously used. The structure of 4 points is computed in closed-form by solving a quadratic equation. Unstable configurations are also investigated. Experimental results on simulated data and real images demonstrate that our algorithm allows fast estimation when included in a robust estimation process.

Key words: affine camera, structure and motion

1 Introduction

Obtaining 3D structure and motion from image feature correspondences only is a fundamental task in computer vision. Particularly, reconstructing 3D points or estimating relative camera motions from minimal data is of primary importance for various numerical estimation procedures such as robust algorithms. In this paper, we propose an algorithm to obtain metric reconstruction from three affine images with the minimum of four point correspondences. While structure and motion from two affine views is ambiguous, it becomes possible when three or more are available. For that reason, this topic has been studied in the case of point [8, 7] or line [1] features. A calibrated affine camera may be modeled by either orthographic, weak perspective or para-perspective projection [4].

In this paper, a polynomial formulation of this minimal reconstruction problem is proposed based on a direct parametrization of the metric structure of a set of four points. Using tools from algebraic geometry, a closed-form solution to reconstruction and unstable cases are all given in the same framework. The main advantage of the method is that it gives directly stable structure by solving a quadratic equation rather than the less well-defined motion parameters.

The paper is organized as follow. Section 2 summarizes the affine camera model. Section 3 formulates the problem in terms of a polynomial system. Experimental results on simulated and real data are respectively given in sections 4 and 5.

2 The affine camera model

The general perspective camera model is described by a 3×4 matrix. Consider a point $M \sim (X, Y, Z, T)^T$ of the projective space, expressed in homogeneous coordinates. Its projection $m \sim (x, y, z)^T$ on the image (considered as a projective plane) is given by the formula:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \sim \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} \\ p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix},$$

where $\sim$ means "equal up to scale". In this paper, we deal with the affine camera model. This means that points at infinity in 3D space are projected to points at infinity in the image. This implies that:

$$p_{3,1} = p_{3,2} = p_{3,3} = 0.$$

The projection equation can then be reduced to:

$$\frac{1}{z} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} p'_{1,1} & p'_{1,2} & p'_{1,3} \\ p'_{2,1} & p'_{2,2} & p'_{2,3} \end{pmatrix} \begin{pmatrix} X/T \\ Y/T \\ Z/T \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}.$$

Note that this is now a strict equality. Using decompostion QR [5]:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} 1 & \epsilon \\ 0 & \xi \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \quad (1)$$

where $\mathbf{R}$ is a rotation, λ is a scale factor, ξ is the aspect ratio, and ϵ is the skew factor. Camera calibration is equivalent to the knowledge of λ, ξ and ϵ. If ϵ and ξ are known, we can perform a change of image coordinates so that the projection writes:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} R_1^T \\ R_2^T \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{2}$$

where R_1^T and R_2^T are two rows of a 3×3 rotation matrix. In practice, we assume $\epsilon = 0$ and $\xi = 1$. These assumptions are valid for most modern cameras.

3 Metric reconstruction

3.1 Problem formulation

Let us consider three affine cameras, each one represented by its projection matrix:

$$\mathbf{P}_i = \lambda_i \begin{pmatrix} {R_1^i}^T \\ {R_2^i}^T \end{pmatrix},$$

where $i \in \{1, 2, 3\}$. For each camera i, we can define an arbitrary plane $\mathcal{P}_i$, which represents its focal plane (the plane where 3D points are orthogonally projected). These planes are defined up to a translation. We expressed the recovered structure and motion in the first camera frame, which implies $\mathbf{P}_1 = (\mathbf{I}_2|\mathbf{0}_2)$.

The minimal configuration. Let us count the number of degrees of freedom of the problem. The first camera may be fixed arbitrarily, and for each additionnal camera, 6 additionnal d.o.f. corresponding to a 3 d.o.f. rotation and a 2 d.o.f. translation (translation is meaningful only parallel to the image plane), and a 1 d.o.f. scale factor. Next, we consider the number of constraints that might be given by each corresponding points: each point gives 6 constraints and 3 unknowns. The problem with 3 cameras and n points may have a solution if and only if

$$3n + 6 \times (3 - 1) \leq 6 \times n.$$

This implies that we need at least 4 points to solve the problem with 3 cameras. This is the minimal configuration.

3.2 Solving the Problem

The fundamental equation. Now, for each point M in space, we denote $\delta_i(M)$, the distance between M and $\mathcal{P}_i$. Then for each pair (M_p, M_q) of points in space, and for the i-th camera, using the Pythagore theorem, we have:

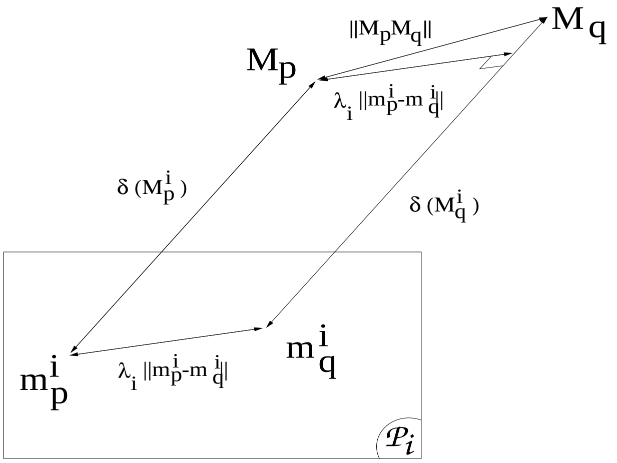

Figure 1. Deriving equation (3)

$$|\delta_i(M_p) - \delta_i(M_q)|^2 + \lambda_i^2 \|m_p^i - m_q^i\|^2 \quad = \|M_pM_q\|^2, \tag{3}$$

where $m_p^i = \mathbf{P}_i M_p$, and $m_q^i = \mathbf{P}_i M_q$ (see figure 1). This equation have interesting properties.

For our reconstruction problem, the calibration of the camera is known so that $\mathbf{P}_i$ could be written as in (2). The coordinates of corresponding points in images are known. Only the distances $\|m_p^i - m_q^i\|$ are known.

Eliminating scale factors. It is clear that the problem is defined up to a global scaling factor, i.e. multiplying all distances by any non-zero constant does not change the 3D structure, λ_1 can therefore be fixed to unity. Using 4 corresponding points, the affine fundamental matrix can be computed [6] between cameras 1 and i, for each $i \neq 1$. The scale factors λ_i can also be computed from the affine fundamental matrices as given in [6].

Reduction of the fundamental equation. For simplicity, we introduce the following notations :

$$\begin{array}{rcc} \alpha_{i,p,q} & = & \lambda_i^2 \|m_p^i - m_q^i\|^2 \\ X_p & = & \delta_1(M_p) \\ Y_p & = & \delta_2(M_p) \\ Z_p & = & \delta_3(M_p). \end{array}$$

Equation (3) is then rewritten as:

$$\begin{array}{rcl} (X_p - X_q)^2 + \alpha_{1,p,q} & = & \|M_pM_q\|^2 \\ (Y_p - Y_q)^2 + \alpha_{2,p,q} & = & \|M_pM_q\|^2 \\ (Z_p - Z_q)^2 + \alpha_{3,p,q} & = & \|M_pM_q\|^2. \end{array}$$

Writing the above equations for cameras i and j, ($i \neq j$), and substracting them, we obtain:

$$\begin{array}{rcl} (X_p - X_q)^2 + \alpha_{1,p,q} & = & (Y_p - Y_q)^2 + \alpha_{2,p,q} \\ (X_p - X_q)^2 + \alpha_{1,p,q} & = & (Z_p - Z_q)^2 + \alpha_{3,p,q}. \end{array}$$

Remembering that X_p, Y_p, Z_p are defined up to the choice of an arbitrary plane $\mathcal{P}_i$, then up to a translation, we can set $X_0 = Y_0 = Z_0 = 0$.

Consider a system given by 4 points in 3 images, it is described by the following polynomials:

$$\begin{aligned}
P_{1,2} &= X_1^2 - Y_1^2 + k_{1,2} \\
P_{1,3} &= X_2^2 - Y_2^2 + k_{1,3} \\
P_{1,4} &= X_3^2 - Y_3^2 + k_{1,4} \\
P_{2,3} &= (X_1 - X_2)^2 - (Y_1 - Y_2)^2 + k_{2,3} \\
P_{2,4} &= (X_1 - X_3)^2 - (Y_1 - Y_3)^2 + k_{2,4} \\
P_{3,4} &= (X_2 - X_3)^2 - (Y_2 - Y_3)^2 + k_{3,4} \\
Q_{1,2} &= X_1^2 - Z_1^2 + l_{1,2} \\
Q_{1,3} &= X_2^2 - Z_2^2 + l_{1,3} \\
Q_{1,4} &= X_3^2 - Z_3^2 + l_{1,4} \\
Q_{2,3} &= (X_1 - X_2)^2 - (Z_1 - Z_2)^2 + l_{2,3} \\
Q_{2,4} &= (X_1 - X_3)^2 - (Z_1 - Z_3)^2 + l_{2,4} \\
Q_{3,4} &= (X_2 - X_3)^2 - (Z_2 - Z_3)^2 + l_{3,4}, \qquad (4)
\end{aligned}$$

where:

$$\begin{aligned}
k_{i,j} &= \alpha_{1,i-1,j-1} - \alpha_{2,i-1,j-1}, \\
l_{i,j} &= \alpha_{1,i-1,j-1} - \alpha_{3,i-1,j-1}.
\end{aligned}$$

The goal is to compute X_i, Y_j and Z_k from $k_{i,j}$ and $l_{i,j}$. We first look at unstable cases for which small errors on the input data may give unreasonably large errors on the solution. We finally give a complete algorithm for structure and motion recovery.

3.3 Instability Conditions

To study the unstable cases, we introduce a new polynomial system obtained with pertubated measurement coefficients $k_{i,j}$ and $l_{i,j}$:

$$\begin{aligned}
k_{i,j}^{meas} &= k_{i,j} + \delta k_{i,j} \\
l_{i,j}^{meas} &= l_{i,j} + \delta l_{i,j}.
\end{aligned}$$

Let the new solution corresponding to these pertubated measurements be:

$$\begin{aligned}
X_i^{meas} &= X_i + \delta X_i \\
Y_i^{meas} &= Y_i + \delta Y_i \\
Z_i^{meas} &= Z_i + \delta Z_i.
\end{aligned}$$

Assume X_i, Y_i, Z_i are the true solution of the original system, after a few algebraic manipulations, the pertubated system can be reduced in matrix form to:

$$\delta\mathbf{K} = \mathbf{N}\delta\mathbf{X} + \varepsilon$$

where

$$\delta\mathbf{K} = \begin{pmatrix} \delta k_{1,2} \\ \delta k_{1,3} \\ \delta k_{2,3} \\ \delta l_{1,2} \\ \delta l_{1,3} \\ \delta l_{2,3} \end{pmatrix} \quad \text{and} \quad \delta\mathbf{X} = \begin{pmatrix} \delta X_1 \\ \delta X_2 \\ \delta Y_1 \\ \delta Y_2 \\ \delta Z_1 \\ \delta Z_2 \end{pmatrix}.$$

The matrix $\mathbf{N}$ has entries linear in X_i, Y_j, Z_k and ε is a vector whose coefficients are linear in $\delta k_{i,j}$ and $\delta l_{i,j}$. Unstable cases occur when $\mathbf{N}$ is singular, i.e. the determinant of $\mathbf{N}$ vanishes. In this case, small changes in $\delta\mathbf{K}$ induce large variations in $\delta\mathbf{X}$. After expansion and factorisation, we obtain:

$$\det(\mathbf{N}) = 64(Z_2Y_1 - Y_2Z_1)(Z_2X_1 - X_2Z_1)(Y_2X_1 - Y_1X_2).$$

The unstable configurations may then be geometrically interpreted as the following two cases:

- the three camera planes are linearly dependent (i.e. the three projection planes in space do not intersect in a common point);
- all planes formed by any three space points intersect any image plane in a line parallel to the intersection lines of the camera planes.

This can be easily proved by assuming that $\mathbf{i}_k$ is a unit vector otrthogonal to the plane $\mathcal{P}_k$ with $k \in \{1, 2, 3\}$.

We first examine the case where the three vectors $\mathbf{i}_k$ are linearly dependent. This is equivalent to the case of two images. The third image can be deduced from the two first ones.

The second case is when the three vectors $\mathbf{i}_k$ are linearly independent. As the vectors $\{\mathbf{i}_1, \mathbf{i}_2, \mathbf{i}_3\}$ can be defined as an orthogonal basis, it is clear that points M_0, M_1, M_2 have $(0, 0, 0)$, (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) as coordinates in this basis. The instability condition $det(M) = 0$ is then equivalent to the fact that there exists $k \in \{1, \ldots, 3\}$ such that the projection of $\mathbf{M}_0\mathbf{M}_1$ and $\mathbf{M}_0\mathbf{M}_2$ on $\mathcal{P}_k$ are collinear.

3.4 Structure and Motion Algorithm

The complete Structure and motion algorithm can now be described as follows. The key idea is to use polynomial resultants [2] to reduce the system to an univariate polynomial:

1. **Computation of the coefficients.** Compute the scale factors with the affine fundamental matrices [6]. Obtain the coefficients $k_{i,j}$ and $l_{i,j}$.

2. **Elimination of the unknowns.** Compute symbolically the resultant R_1 of $P_{1,2}$ and $P_{2,3}$ in Y_1. Compute symbolically the resultant R_2 of R_1 and $P_{1,3}$ in Y_2. Set $R_{1,2} = \sqrt{R_2}$ (R_2 is a squared polynomial). Compute formally the resultant S_1 of $Q_{1,2}$ and $Q_{2,3}$ in Z_1. Compute formally the resultant S_2 of S_1 and $Q_{1,3}$ in Z_2. Set $S_{1,2} = \sqrt{S_2}$ (S_2 is a squared polynomial). Compute formally the resultant T of $R_{1,2}$ and $S_{1,2}$ in X_2.

Finally, we obtain the polynomial T which is an univariate polynomial of degree 2 in X_1^2. $T = 0$ can be easily solved in closed-form.

The algorithm gives directly the 3D coordinates of points in space, the projection matrices can then be computed from them.

4 Experimental results

4.1 Maximum Likelihood Estimation

The method described in this paper is minimal, in other words, there is no cost function and the results obtained are exact, depending only on input data. However, our algorithm involved solving a polynomial of degree 2 as well as a set of linear equations, which might induce numerical instabilities. For that purpose, we devise the Maximum Likelihood Estimator (MLE), for further refining the previously computed solution. In more detail, it consists in minimizing the Root Mean Squares of the reprojection residuals using an adequat parameterization of camera matrices and scene points. Let us denote the set of parameters as Θ. This set contains 12 motion parameters which are two rotations and translations and $3n$ structure parameters, where n is the number of points considered. Camera matrices are formed according to equations (1). The MLE is then given by $argmin_{\Theta}\mathcal{C}(\Theta)$ where the cost function is defined as:

$$\mathcal{C}(\Theta) = \sum_{i=1}^{i=3}\sum_{j=1}^{j=n} d^2(m_j^i, \hat{m}_j^i),$$

where $d^2(.,.)$ denotes the squared Metric distance in the image, m_j^i denotes points measured in the images and $\hat{m}_j^i$ denotes the reprojected points. The optimization is conducted using the Levenberg-Marquardt method [3].

We insist on the fact that, theoretically, both results should be the same. Comparing the solution provided by our algorithm and the Maximum Likelihood Estimator will only tells us about the numerical stability of the algorithm. The MLE is equivalent to bundle adjustment.

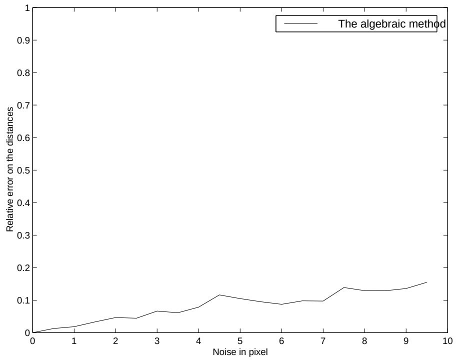

Figure 2. The median of the errors with different noise levels on simulated data.

4.2 Simulated data

Minimal data. The simulation protocol is set up as follows. We choose at random 3 affine cameras, and four 3D points. We then compute the projections, and add noise on them. With the projections we can compute the 3D structure, which can be compared to the original points in space. This gives us one possible evaluation of reconstruction errors. This experiment is repeated 100 times with different noise levels from 0 to 10 pixels. Figure 2 shows the results. We can observe that the error is approximately linear in the added noise level.

RANSAC Estimation. We now take 157 3D points from a calibration object (Figure 4). We project them with computed projections, and then add some noise on the obtained images. Then, we compute a solution with RANSAC. First, we compare the results of the method with those obtained after a bundle adjustement on the motion computation. The bundle adjustment is made with the data computed by the method. Next, we do the same but the bundle adjustement is on structure and motion instead of only motion. The results illustrated in Figure 3 show that the obtained results are almost as good as the optimal solutions from bundle adjustements.

4.3 Real images

Target images. We carried out recontruction experiments from images of a target. With an appropriate software, we match points (there are no outliers in those images). The structure of the target gives us the affine calibration of the camera (We have seen that between the three images the calibration changes of less than 0.5%). The target is an object composed of three planes, two pairs of them forming a right angle (figure 4). At first we have seen that the average reprojection error is less than 0.5 pixel.

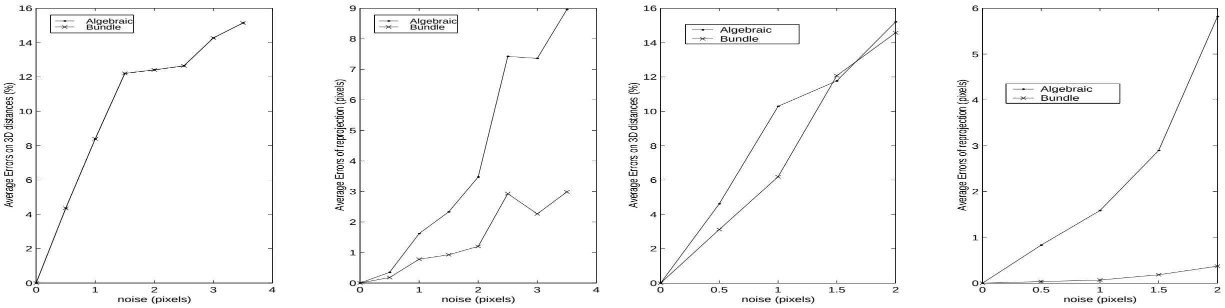

Figure 3. The two first graphs give a comparison between the new method followed by a bundle adjustement on motion parameters. The two next graphs give a comparison between the new method followed by a bundle adjustement on motion and structure. Note that the two first curves are undistinguishables.

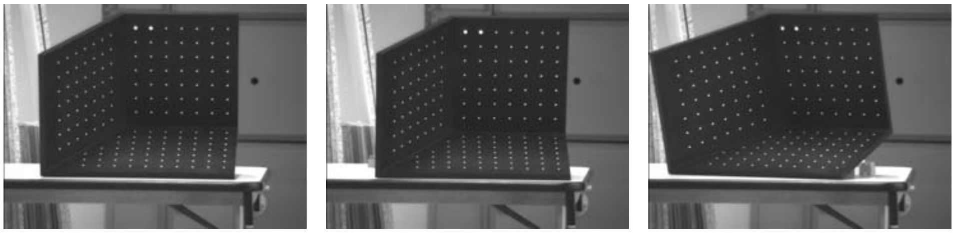

Figure 4. Three images of the target.

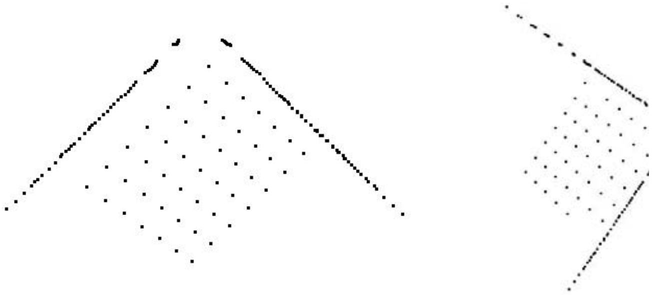

Figure 5. The two right angles of the reconstructed target.

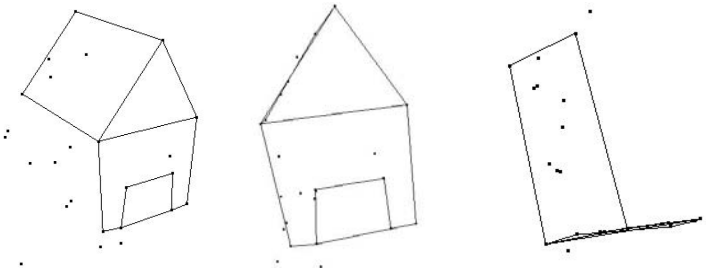

Figure 7. The house with matched points.

In fact, because the reconstruction is Metric we expect to see the right angles between the planes in the reconstruction. Figure 5 shows the two top views of these angles.

House images. We have tested the algorithm on the house image sequence. We took three images of a house, detect points of interest with the Harris detector, match them with cross-correlation between adjacent images and triplet registration, see figure 6. We can see that there is a lot of bad matches. We first detect the outliers with a RANSAC method, so as to eliminate them. Next, we compute a model that minimizes the reprojection error over 100 trials. For the model, see figure 7.

5 Conclusion

We have presented a method for Metric reconstruction from calibrated affine cameras. The method works with minimal data, i.e. 4 points in 3 images. The stucture of 4 points in space is computed in closed-form rather than the motion parameters used by other researchers. We also studied the instability of minimal reconstruction. Experimental results show that the method is stable. Another important point is that the method is fast as it needs only to solve a polynomial of degree 2. As a comparison, bundle

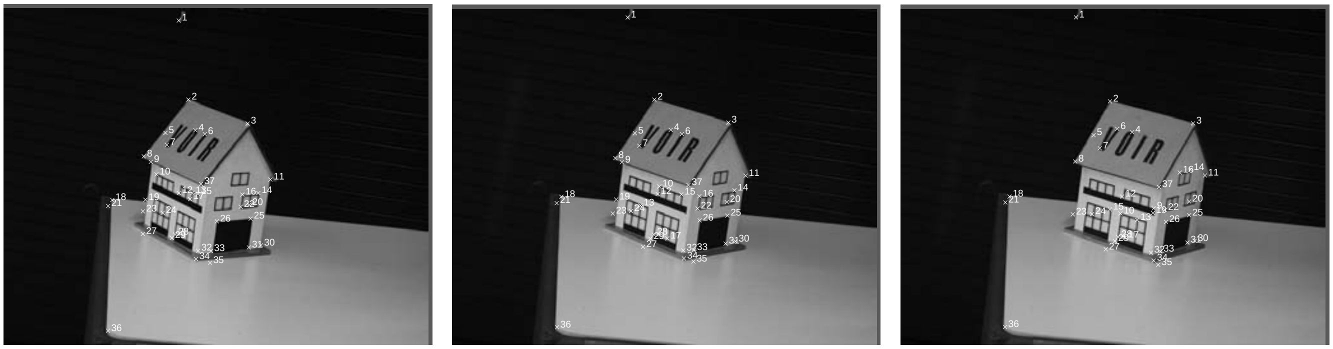

Figure 6. The house with matched points.

adjustement takes at least 50 times more, to optimize over the motion, and 1000 times more to optimize over motion and structure.

References

[1] K. Astrom, A. Heyden, F. Kahl, and M. Oskarsson. Structure and motion from lines under affine projections. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 285–292, September 1999.

[2] D. Cox, J.Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer, 1996.

[3] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, 1981.

[4] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. In J.O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, pages 97–108. Springer-Verlag, May 1994.

[5] L. Quan. Self-calibration of an affine camera from multiple views. *International Journal of Computer Vision*, 19(1):93–105, May 1996.

[6] L.S. Shapiro, A. Zisserman, and M. Brady. 3D motion recovery via affine epipolar geometry. *International Journal of Computer Vision*, 16(2):147–182, 1995.

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

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