Recovering 3D Metric Structure and Motion from Multiple Uncalibrated Cameras

Miguel Sainz and Nader Bagherzadeh
Dept. Electrical and Computer Engineering,
University of California Irvine.
msainz,nader@ece.uci.edu

Antonio Susin
Dept. Matematica Aplicada I,
Universitat Politecnica de Catalunya
susin@ma1.upc.es

Abstract

An optimized linear factorization method for recovering both the 3D geometry of a scene and the camera parameters from multiple uncalibrated images is presented. In a first step, we recover a projective approximation using a well known iterative approach. Then, we are able to upgrade from projective to Euclidean structure by computing the projective distortion matrix in a way that is analogous to estimating the absolute quadric. Using the Singular Value Decomposition (SVD) as a main tool, and from the study of the ranks of the matrices involved in the process, we are able to enforce an accurate Euclidean reconstruction. Moreover, in contrast to other approaches our process is essentially a linear one and does not require an initial estimation of the solution. Examples of synthetic and real data reconstructions are presented.

1. Introduction

We are facing the problem of extracting the shape of objects and the way they have been recorded using a single uncalibrated camera. This is known as the *structure from motion problem* (SfM). More specifically, the goal of the problem is to recover the 3D geometry of a scene from the 2D projections obtained from multiple view images, taking into account the motion of the camera. But, neither the camera calibration (intrinsic parameters and pose) nor the geometry of the scene are known.

The correspondence process between points of different frames is assumed to be known in the general SfM. This paper follows the approach of ([11],[6],[7],[3],[8]) where a small set of corresponding points is known. Since we are working with uncalibrated cameras, we choose an stratification approach to recover both camera parameters and structure of the scene. The idea is to upgrade from projective to Euclidean structure. In [5] a good review of different kind of methods is presented. The method presented in this paper allows to recover an Euclidean reconstruction of the model without any initial guess which is one of the drawbacks of most of the existing methods. Another important feature is that the whole process is based on solving linear systems with SVD decomposition. The knowledge of the geometric properties of different transformations represented by the process matrices provide us a valid solution in terms of the rank of these matrices.

Our results demonstrate that with synthetic data a great accuracy can be obtained, and when noise is added the precision is still acceptable. These results are maintained when real data examples are used. We also present different number of views ranging from from 5 to 50, only the computation time is affected maintaining similar error rates.

2. Projective Reconstruction

Next we will address the SfM problem using a small set of points or features and we assume the 2D trajectories of these features along the image sequence are known. We assume no prior knowledge of their coordinates in the 3D space, the relative motion between the camera and the scene (extrinsic parameters), and the camera's internal geometry (intrinsic parameters) and we wish to recover this information only from the 2D measurements corresponding to the set of features we are considering. Following [12], we use standard image coordinates, that is, we scale image pixels to lie in $[-1, 1] \times [-1, 1]$ which guarantees good numerical conditioning.

2.1. Factorization method

The projective factorization method ([11],[7], [3],[8],[6]) is a generalization of the factorization method which was first developed in [10] and [9], for the ortographic projection and the paraperspective models respectively. If no information is known about the camera intrinsic parameters, the motion and the object, only a reconstruction up to an unknown projective transformation is possible to compute.

Our goal is to recover 3D structure and motion from m uncalibrated perspective images of a scene and n 3D ob-

ject points. Let $\mathbf{X}_j = (X_j, Y_j, Z_j, 1)^T, \quad j = 1, \ldots, n$, be the unknown homogeneous 3D point vectors, $P_i, \quad i = 1, \ldots, m$ the unknown 3×4 image projections, and $\mathbf{x}_{ij} = (u_{ij}, v_{ij}, 1)$ the measured homogeneous image point vectors.

We call **projective depths** the non-zero scale factors λ_{ij} relating the world points and its projections

$$\lambda_{ij}\mathbf{x}_{ij} = P_i\mathbf{X}_j \quad i = 1, \ldots, m \quad j = 1, \ldots, n. \quad (1)$$

Each object is defined only up to rescaling. With correctly normalized points and projections the λ's become true optical depths (see [11]).

We can state the problem in matrix form as $\mathbf{W} = \mathbf{P}\mathbf{X}$, where $\mathbf{W}$ is the $3m \times n$ *scaled measurement matrix*, $\mathbf{P}$ is the $3m \times 4$ *perspective matrix* and $\mathbf{X}$ is the $4 \times n$ *shape matrix*. As is stated in [11], the projective depths depend on the 3D structure, which in turn derives from the depths. To recover the values of λ_{ij} an iterative projective algorithm is proposed, based on the Singular Value Decomposition (SVD). Matrix $\mathbf{W}$ has to be a rank-4 matrix if it is the matrix associated to a projection of a set of real points. Consequently, for points in general positions, a rank-4 factorization of the scaled matrix produces a projective reconstruction of the points.

We follow [11] and [2] in building a convergent iterative algorithm for approximating $\mathbf{X}$ and λ_{ij} successively. The algorithm can be stated as:

1. First set $\lambda_{ij}^{(0)} = 1$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ as initial conditions. This can be assumed because the depth values (essentially for the first image) can not be determined uniquely. In fact they can be chosen arbitrarily in the linear subspace generated by the rows of $\mathbf{X}$. As is stated in [8] the final algorithm is robust w.r.t. initial conditions.

2. A first SVD factorization of $\mathbf{W}^{(k)}$, with $\mathbf{W}^{(0)} = \mathbf{W}$, is computed. We use the standard notation (see [4]) $\mathbf{W}^{(k)} = UDV^T$, where U is a $3m \times n$ matrix which their columns are an orthogonal basis of the output (range) subspace of $\mathbf{W}^{(k)}$. D is a $n \times n$ diagonal matrix, their elements σ_i, are known as the *singular values* of $\mathbf{W}^{(k)}$, and finally, V is a $n \times n$ matrix containing an orthonormal basis corresponding to the input (co-kernel) of $\mathbf{W}^{(k)}$.

A first approximation $\mathbf{P}^{(k)} = U_4$ is computed, where U_4 means the submatrix obtained from U using only the 4 first columns (the ones associated to the 4 larger singular values). Analogously, $\mathbf{X}^{(k)} = D_4 V_4^T$ and from that we compute the following estimate $\widetilde{\mathbf{W}}^{(k)} = \mathbf{P}^{(k)}\mathbf{X}^{(k)}$. This choice guarantees (see [4], pp. 72) that we get the best rank4 approximation of $\mathbf{W}^{(k)}$, and the spectral distance (using $\| \quad \|_2$) from the subspace of the rank 4 is exactly σ_5.

3. Once the rank4 approximation $\widetilde{\mathbf{W}}^{(k)}$ is computed, we get an estimate of the 3D coordinates, $\mathbf{X}_j^{(k+1)}$ of the points The new depth $\lambda_{ij}^{(k+1)}$ is chosen to coincide with the projection of $\mathbf{X}_j^{(k+1)}$ into the visual line, that is,

$$\lambda_{ij}^{(k+1)} = \lambda_{ij}^{(k)} \frac{\widetilde{W}_{ij}^{(k)T} W_{ij}^{(k)}}{W_{ij}^{(k)T} W_{ij}^{(k)}} \quad (2)$$

4. After computing the new value of the depth matrix we get an update measurement matrix $W_i^{(k+1)} = \mathbf{x}_{ij}\lambda_{ij}^{(k+1)}$, $j = 1, \ldots, n$ and the process is repeated until the value of the corresponding $\sigma_5^{(k+1)}$ is either small enough or it is stabilized. As we will show in the results only with synthetic data we can obtain $\sigma_5^{(k+1)}$ as small as we want. When noise is added to the projections, the values of $\sigma_5^{(k+1)}$ can reach an small stable value but not zero.

Dataset	# V	# P	# iter	σ_5/σ_4	Max Err
B.array	9	22	814	1.1996e-9	5.4250e-8
B.array	9	22	200	1.2121e-2	0.6221
B.array	9	22	185	2.4143e-2	1.2492
B.fly	50	23	2864	3.0226e-9	7.0317e-8
B.fly	50	23	396	2.6318e-2	0.6405
B.fly	50	23	317	5.2370e-2	1.3786
house	5	38	174	3.3670e-2	2.8772
monitor	8	18	251	2.8810e-2	3.1301

Table 1. Projective recovery data

2.2. Projective reconstruction results

In Table (1) we show some of the results we have obtained with different set of data. The values, # V, # P and # iter, stands for number of views, number of points and number of iterations respectively. The value σ_5/σ_4 corresponds to the final ratio obtained when iterations stop. This value gives an idea of how far from rank4 is the reconstruction using matrix norm. On the other hand, Max Err, is the *reprojection* error in pixels. The set of data we use corresponds to a synthetic building model with different camera locations (see Figure 3) (B.array,B.fly) and different rows corresponds to the noise added which is zero, one and two random pixels. The final rows corresponds to real images *house* and *monitor* (see Figure 2 and 4). One can observe that the number of iterations is much larger in the case of ideal data because the method can reach a high accuracy level. The process is always stopped when the value of σ_5 becomes stable. The point stability is more important than its final value, therefore we stop the iterative process when the relative error of σ_5 becomes less than 10^{-6}. When noise is present the final value can be far from zero but it is always convergent.

3. Metric reconstruction

The factorization of Equation (1) recovers the motion and the shape up to a linear projective transformation H known as the *Projective Distortion Matrix* (PDM)

$$W = \hat{P}\hat{X} = \hat{P}HH^{-1}\hat{X} = PX \quad (3)$$

with $P = \hat{P}H$ and $X = H^{-1}\hat{X}$. We need to impose metric constraints to recover the correct Euclidean motion and shape. This process is called *normalization*. Although different cases can be considered according to the unknown intrinsic parameters of the camera (see Equation (7)), we assume zero skews and center our study to the consider the focal length f as the only unknown parameter. This means that we consider the case where the aspect ratio is 1 and the principal point is at the origin (see [7] for the other possibilities).

3.1. Metric reconstruction and the absolute quadric

Before discussing the details of the proposed method for recovering the metric structure from the projective approximation, we would like to relate our approach with the methods based on epipolar geometry for autocalibration [12], [2].

Most of the existing methods of autocalibration relies on computing the intrinsic parameters of the camera from the relations

$$\omega = KK^T, \qquad \omega = P\Omega P^T. \tag{4}$$

where the 3×3 matrix ω is known as the *dual absolute image conic* and its projection the *absolute quadric* Ω. If ω is known, then K can be easily obtained from a Choleski decomposition of this matrix.

It turns out that estimating the absolute quadric is equivalent to an autocalibration process, because from Ω (with the projective matrix) one can compute ω and finally K. As we will show, recovering the PDM, is equivalent to estimating the absolute quadric.

Let us express the PDM as the following 4×4 matrix

$$H = \begin{pmatrix} H_1 & b_1 \\ h_1^T & 1 \end{pmatrix}. \tag{5}$$

The point corresponding to the Euclidean origin is computed by $H(0,0,0,1)^T$, which is $(b_1, 1)^T$, then b_1 are the coordinates corresponding to the origin. Without loss of generality, we can assume $b_1 = (0,0,0)^T$ (projective and Euclidean space share the same origin).

Now, for each frame i, the projective 3×4 matrix P_i can be decomposed into

$$P_i H = \mu_i K_i \left(R_i | \mathbf{T}_i\right) \quad i = 1, \ldots, m. \tag{6}$$

where

$$K_i = \begin{pmatrix} f_i & \beta_i & u_{0i} \\ 0 & \alpha_i f_i & v_{0i} \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_i = \begin{pmatrix} \mathbf{i}_i^T \\ \mathbf{j}_i^T \\ \mathbf{k}_i^T \end{pmatrix}, \quad \mathbf{T}_i = \begin{pmatrix} \mathbf{T}_{xi} \\ \mathbf{T}_{yi} \\ \mathbf{T}_{zi} \end{pmatrix}, \quad i = 1, \ldots, m. \tag{7}$$

Where μ_i is a scale factor, the calibration matrix K_i encodes the intrinsic parameters of the camera, (u_{0i}, v_{0i}) is the principal point, α_i is the aspect ratio, β_i is the skew and f_i is the focal length. R_i and $\mathbf{T}_i$ are the ith rotation 3×3 matrix and translation vector of the camera for each frame.

If we consider the first three columns of $P_i = \left(\tilde{P}_i, p_i\right)$, the product $P_i H = \mu_i K_i \left(R_i | \mathbf{T}_i\right)$ can be written with two column equations

$$\left(\tilde{P}_i, p_i\right) \begin{pmatrix} H_1 \\ h_1^T \end{pmatrix} = \mu_i K_i R_i, \tag{8}$$

$$\left(\tilde{P}_i, p_i\right) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \mu_i K_i \mathbf{T}_i. \tag{9}$$

3.2. Normalization algorithm

We will give now a description of the normalization algorithm as is stated in [7] and also the improvements we add in order to obtain a more robust one. The main problem is that we recover the absolute quadric as a 4×4 matrix and from that we need to recover the PDM. Indeed, only the part corresponding to $\left(H_1^T | h_1\right)$ is recovered. As we will show below, the method presented in [7] does not guarantee to obtain good results by just applying a rank 3 decomposition.

Combining Equation (6) for all the frames, we can get the global matrix $P = (M|T)$, where $M_i = K_i R_i$, and $T_i = K_i \mathbf{T}_i$.. We express the 4×4 matrix PDM H as $H = (A|B)$ where $A = \begin{pmatrix} H_1 \\ h_1^T \end{pmatrix}$ is 4×3 and B is 4×1. Since from (3) we have $P = \hat{P}H$, then

$$(M|T) = \hat{P}\,(A|B)\,. \tag{10}$$

At this moment, we decouple the computation of the translation from the rotation one. This way, we will be able to compute the Euclidean reconstruction using essentially linear algorithms, instead of the nonlinear ones related with Kruppa's equations [5].

Taken into account that the shape matrix $\mathbf{X}$ is related to the geometry of the object and therefore, independent of the frame, we can express each point in local object coordinates

$$\mathbf{X}_j^T = \left(\tau_j s_{xj}, \tau_j s_{yj}, \tau_j s_{zj}, \tau_j\right), \quad j = 1, \ldots, n. \tag{11}$$

where $s_j = (s_{xj}, s_{yj}, s_{zj})$ are the local coordinates. We can also consider the origin of the world coordinate system placed at the center of mass of the scaled object points.

Now, if we look at the sum of the first coordinates of the projected points, using the center of mass we can obtain

$$\sum_{j=1}^{n} \lambda_{ij} u_{ij} = T_{xi} \sum_{j=1}^{n} \tau_j. \tag{12}$$

Analogously, a similar expression can be obtained for the other coordinates. Next, we can use the translation terms to

compute B solving a linear least square problem. For that, we consider (10) to get

$$T_{xi} = \hat{P}_{xi}B, \qquad T_{yi} = \hat{P}_{yi}B, \qquad T_{zi} = \hat{P}_{zi}B. \quad (13)$$

From (12) we obtain the quotient

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^{n}\lambda_{ij}u_{ij}}{\sum_{j=1}^{n}\lambda_{ij}}, \qquad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^{n}\lambda_{ij}v_{ij}}{\sum_{j=1}^{n}\lambda_{ij}} \quad (14)$$

Finally, from (13) and (14) we can set up an homogeneous system of $2n$ linear equations for the 4 unknowns elements of B. The kernel of the system gives us the elements of B.

On the other hand, we have to compute the matrix A to complete the desired distortion matrix. The information embedded in A is the orientation of the PDM. To express that in a compressible way, first from (6) and (10) we obtain

$$M_{xi} = \mu_i f_i \mathbf{i}_i + \mu_i u_{0i}\mathbf{k}_i,$$

$$M_{yi} = \mu_i \alpha_i f_i \mathbf{j}_i + \mu_i v_{0i}\mathbf{k}_i, \quad M_{zi} = \mu_i \mathbf{k}_i. \quad (15)$$

In the case we are considering $\alpha_i = 1$, $u_{0i} = v_{0i} = 0$, and using that the rotation axis are orthogonal we get the metric relations

$$|M_{xi}|^2 = |M_{yi}|^2, \quad |M_{zi}|^2 = \mu_i^2,$$

$$M_{xi} \cdot M_{yi} = M_{xi} \cdot M_{zi} = M_{yi} \cdot M_{zi} = 0 \quad (16)$$

From (16), the metric constraints can be written as linear constraints using (10)

$$MM^T = \hat{P}AA^T\hat{P}^T = \hat{P}Q\hat{P}^T. \quad (17)$$

obtaining a set of $4m$ linear equations for the 10 unknowns of Q. As we will show in the next section the solution Q that we can obtain is related to the rank of the system (17) and we find that this is essential for getting acceptable results in general cases.

3.3. Recovering the absolute quadric

As we explain above, the distortion matrix is closely related with the absolute quadric. The matrix Q is essentially Ω and from the homogeneous overdetermined system (17) we can get a (non-unique) solution. In [7] they solved the problem adding a new non-homogeneous metric equation (based on the scale factors on (6)) fixing the first factor to one, $\mu_1 = 1$, and adding the equation $|M_{z1}|^2 = 1$ to (17). After obtaining the least square solution they make a rank 3 decomposition of Q to get the matrix A and this way completing the projective distortion matrix. As we have observed this is not in general a robust method of solution because the essential condition (see [12]) rank(Ω)=3 is not imposed to the solution Q obtained from (17).

In the general case the homogeneous system (17) turns to be of rank 8. This is because the unknowns involved in the system (17) are essentially the components of Ω and ω and there is an additional constraint (see [12]) between them. Therefore, we have one extra degree of freedom for the solution which can be used to force the final matrix Q to be rank 3. Indeed, we consider a linear combination of the vectors associated to the zero singular values and we impose $\det(Q) = 0$. This gives us an four order polynomial, using the linear combination coefficients quotient as unique variable, and we choose the best root that gives us a rank 3 matrix Q. After that, we obtain matrix A as a rank 3 approximation of Q using again the SVD decomposition.

	AME	AME1	nAME
σ_8	3.4595e-1	3.4595e-1	3.4595e-1
σ_9	2.0818e-1	2.0818e-1	9.7575e-3
σ_{10}	2.3501e-3	2.3501e-3	2.2559e-3
Q rank	4	3	3
A error	3.5798e-1	1.6120e-9	2.2567e-7
2D Error	19.7051	3.7062442	3.7062632

Table 2. Metric recovery results for real datasets

4. Experiments and Results

Several tests have been performed with both real and synthetic datasets. For the real imagery we have used previously tracked sequences of a computer monitor (8 images and 18 points) and of a model house (5 images and 38 points). The synthetic datasets and their tracking information have been generated using an in-house tool. The first and second synthetic sets (Figure 3 and 6) have 9 cameras and 22 tracked points. The third set is a sequence (Figure 5) of 50 frames and 23 tracked points.

We have applied three different methods, mainly to compare the results obtained using the method in [7] with our improvement based on the rank of the matrix Q.

The first one uses an additional metric equation (AME) as it is proposed in [7], the second one uses the same equation plus the rank3 condition discussed in section 3.3 (AME1), and the third one does not use the additional metric equation (nAME) but enforces the rank3 condition. These three methods have been tested with the initial datasets and also with four levels of uniform distributed noise (0.5, 1.0, 1.5 and 2.0 pixels).

The results of the three methods are shown in Figure 1 where we represent, for one of the datasets, the relative mean error (RME) of the 3D reconstructed points and the recovered camera positions for the different noise levels.

We have found that the AME method works fine for some of the sets but when the amount of noise increases, the RME increases too. One of the reasons for this is the arbitrariness of the additional metric condition. We have 3*m* different equations to choose from, and due to noise in the data, is difficult to decide which is the best candidate to generate

a rank3 Q matrix. Our first solution, the AME1 method enforces this condition for Q and it improves considerably the accuracy of the reconstruction under noisy data. With the nAME method no additional metric condition is necessary, and by enforcing the proper rank to the system we are able to extract the right solution.

Due to the lack of ground true data for the real imagery, we have not been able to perform a numerical verification of the reconstructed set. A 2D reprojection error analysis (Table 2) plus a visual quality check show that the proposed method works well under real data.

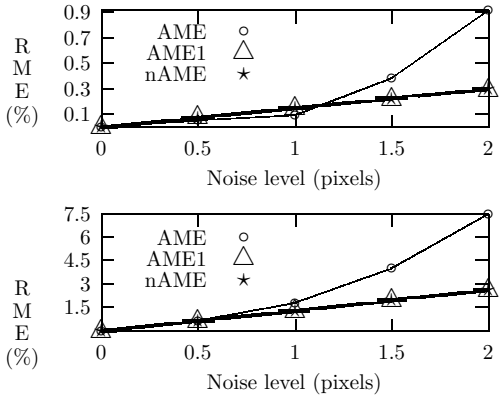

Figure 1. Error of the recovered scene

Finally, some snapshots of the reconstructions are shown in the figures, where a wireframe model of the reconstructed points has been built. The small camera icons have been located at their recovered location with appropriate orientation.

Most of the computation time is used in the iterative projective depth recovery. This becomes very noticeable in the synthetic datasets, specially for the long sequences, where due to the high accuracy of the data, the iterative method takes a long time to converge to a very accurate solution. However with the presence of noise or with the real data, the convergence is very fast (but less accurate). On the other hand the metric reconstruction takes an nonsignificant amount of time compared to the iterative part. We run our tests in a PC P4 at 1.4GHz and the real datasets take around 0.5s to 1s to complete and for the synthetic data the time ranges between 2s up to 78s for the longest sequence (50 frames) with zero noise.

5. Conclusions

A new improvement for solving the SfM problem has been presented. The approach is based on an iterative projective reconstruction and a linear solution for the metric normalization process. The proper analysis of the ranks of the matrices involved in the process plus a rank enforcement step leads to very accurate solutions. One of the advantages of the proposed method is that it does not need any initial solution or arbitrary additional constraints to compute the final result.

We have tested the method under noisy conditions and both multiple views and long image sequences. In all cases, excellent reconstructions have been obtained. Moreover, the proposed method is computationally fast in standard PC computers, making it a very attractive solution.

As a future work, we plan to extend this method to handle the more general case of the intrinsic parameter unknowns. This will allow us to improve the accuracy in the reconstructions of some specific camera configurations were these extra parameter could be significative.

Acknowledgments: We would like to thank Qian Chen for given us the data for the real images examples, and to Heesung Jun for his fruitful discussions that helped us to fix some bugs in our implementation. This rsearch was partially funded by grants from NSF (CCR-9877171 and CCR-0083080) and Gaspar de Portolà grant C99–00.

References

[1] Berthilsson R., Heyden A. and Sparr G., Recursive structure and motion from image sequences using shape and depth spaces. *Proc. IEEE Computer Vision and Pattern Recognition*, 444–449, 1999.

[2] Chen Q., Multi-view Image-Based Rendering and Modeling. *PhD. Dissertation, Computer Science USC*, 1–141, 2000.

[3] Chen Q. and Medioni G., Efficient, iterative solution to M-view projective reconstruction problem. *Proc. IEEE Computer Vision and Pattern Recognition* **1**, 55–61, 1999.

[4] Golub G.H., Van Loan C.F. *Matrix Computations. Third Edition*, Ed. Johns Hopkins, 1996.

[5] Fusiello A., Uncalibrated Euclidean reconstruction: a review. *Image and Vision Computing*, **18**, 555–563, 2000.

[6] Heyden A., Berthilsson R. and Sparr G., An iterative factorization method for projective structure and motion from image sequences. *Image Vision and Computing*, **17**, 981–991, 1999.

[7] Han M. and Kanade T., Creating 3D Models with Uncalibrated Cameras *IEEE Computer Society Workshop on the Application of Computer Vision (WACV2000)*, **9**(2), 137–154, 2000.

[8] Mahamud S. and Hebert M., Iterative projective reconstruction from multiple views motion *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, **2**, 430–437, 2000.

[9] Poelman C.J. and Kanade T., A paraperspective factorization method for shape and motion recovery *Technical Report* CMU-CS 93-219, School of Computer Science, Carnegie Mellon University , December 1993.

[10] Tomasi C. and Kanade T., Shape and Motion from Image Streams Under Orthography: a factorization method *International Journal of Computer Vision*, **9**(2), 137–154, kjh hkjh 1992.

[11] Triggs B., Factorization methods for projective structure and motion *IEEE CVPR96*, 845–851, 1996.

[12] Triggs B., Autocalibration and the Absolute Quadric *IEEE CVPR97*, 609–614, 1997.

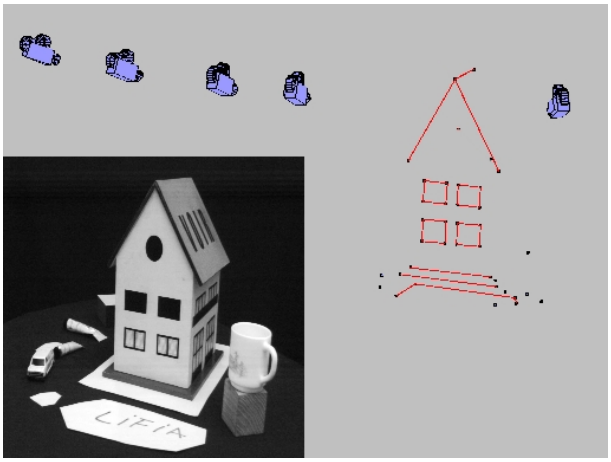

Figure 2. The house example with 5 images.

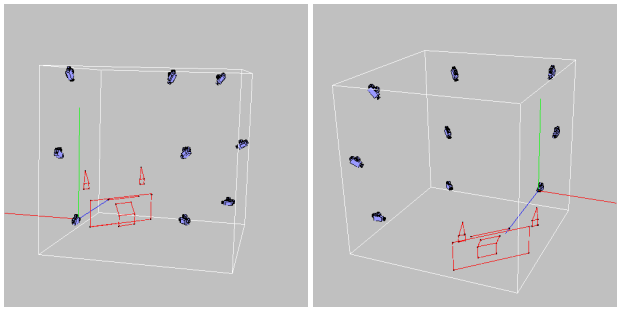

Figure 3. 9 images of a synthetic building and two snapshots of the recovered data.

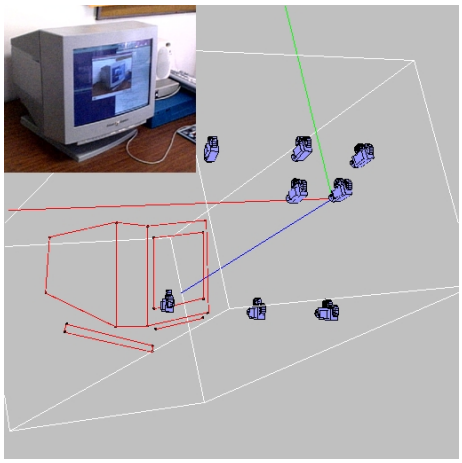

Figure 4. A real set of 8 views of a monitor

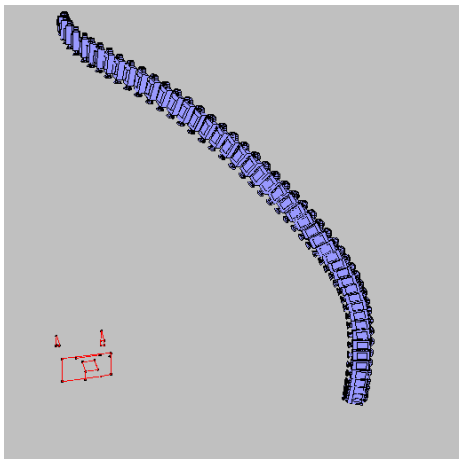

Figure 5. A 50 frames recovered spiral fly

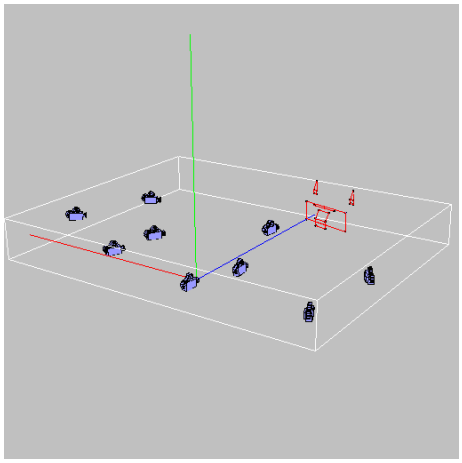

Figure 6. A 9 frames recovered sequence of a planar motion around our building model