

**Figure 6 Three views of the recovered structure of the Renault part**

## 5 Conclusion and Discussion

We have described an algorithm that takes as input a few images that show various aspects of an object and is able to estimate the location of the feature points of the object with respect to a single reference frame. Despite the use of nonlinear least squares fitting in structure and motion estimation, we assume no knowledge of either the shape of the object or the motion involved to obtain good initial guesses. While our algorithm uses algebraic technique in initialization, experimental results show that the quality of our solution does not degrade with the accuracy of the initial guess.

While our algorithm is iterative, the number of iterations is linear in the number of pairs of close-by views. In this case, the total number of close-by view pairs is linear in the number of input images,  $m$ . Since solving for structure and motion parameters by Levenberg-Marquardt algorithm takes  $O(N^3)$  time, where  $N$  is the total number of parameters to be estimated, which is  $3n+6m$  in our case, our algorithm takes  $O(m(n+m)^3)$  time.

In the experiments, we found that our iterative algorithm indeed converges over a broad range of initial values. Using the result of Weinshall, Werman and Gdalyanhu [13] on the stability and likelihood of views, we can infer that the “flattest” view of the object is very likely to exist as an input image to our problem. Thus, simple initialization may be sufficient for structure parameter. However, it also means that the motion required to obtain an image that shows a different aspect would be relatively large.

Obtaining the 3D location of feature points of an object is the very first step towards building complete 3D model for the object. Surface information are much more useful and thus need to be inferred. While one can use var-

ious existing methods that reconstruct surface from unorganized 3D points to obtain a surface description, some surface information are actually available in the image of the object. Information such as junctions, symmetries, and visibility are useful in determining the surface of the object, and thus should not be ignored. We are investigating methods that make use of this information to derive surfaces.

## Acknowledgment

Special thanks to Dr. Zhengyou Zhang of INRIA-Sophia Antipolis for stimulating discussions about obtaining initial guesses for motion estimation using analytical methods and providing software to conduct experiments.

## References

- [1] A. Azarbayejani, T. Galyean, B. Horowitz, and A. Pentland, “Recursive Estimation for CAD Model Recovery”, *Proc. IEEE 2nd CAD-based vision workshop*, 1994.
- [2] R. Deriche, Z. Zhang, Q. T. Luong, and O. Faugeras, “Robust recovery of the epipolar geometry for an uncalibrated stereo rig”, *ECCV*, pp. 567-576, 1994.
- [3] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, Cambridge, Mass., 1993.
- [4] T. Joshi, N. Ahuja, and J. Ponce, “Silhouette-based structure and motion estimation of a smooth object”, *IJW*, 1994.
- [5] H.C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections”, *Nature*, 193:133-135, 1981.
- [6] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, England, 1988.
- [7] L. Rosenthaler, F. Heitger, O. Kubler and R. von der Heydt, “Detection of General Edges and Keypoints”, *ECCV*, pp.78-86, 1992.
- [8] R. Szeliski and S. B. Kang, “Recovering 3D Shape and Motion from Image Streams using Non-Linear Least Squares”, *CVPR*, pp. 752-753, 1993.
- [9] C.J. Taylor, D.J. Kriegman, and P. Anandan, “Structure and motion in two dimensions from multiple images: A least squares approach”, *IEEE Workshop on Visual Motion* (Princeton, New Jersey), pp. 242-248, Oct 1991.
- [10] C. Tomasi and T. Kanade, “Factoring image sequences into shape and motion”, *IEEE Workshop on Visual Motion* (Princeton, New Jersey), pp. 21-28, Oct. 1991.
- [11] R.Y. Tsai and T.S. Huang, “Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces”, *IEEE Trans. Patt. Anal. Machine Intell.*, PAMI-6(1):13-27, Jan. 1984.
- [12] J. Weng, N. Ahuja, and T. S. Huang, “Optimal Motion and Structure Estimation”, *IEEE Trans. Patt. Anal. Machine Intell.*, vol 15, no. 9, pp. 864-884, 1993.
- [13] D. Weinshall, M. Werman and Y. Gdalyanhu, “Canonical Views, or the Stability and Likelihood of Images of 3D Objects”, *Image Understanding Workshop* (Monterey, CA), pp. 967-971, 1994.
- [14] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong, “A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry”, to appear in *Artificial Intelligence*.

Fr		Rotation			Translation				
		angle	axis		dist.	direction			
(b)	actual	90.00	0.0008	0.7435	-0.6686	.6765	-0.0364	-0.7857	-0.6174
	result	89.99	0.0010	0.7417	-0.6707	.6742	-0.0398	-0.7872	-0.6153
(c)	actual	51.20	-0.9805	0.1730	-0.0935	.3812	-0.4661	-0.8165	-0.3405
	result	51.19	-0.9804	0.1726	-0.0947	.3816	-0.4645	-0.8182	-0.3387
(d)	actual	90.00	0.0008	-0.7435	0.6686	.9365	0.9674	-0.1374	0.2123
	result	89.97	-0.0010	-0.7420	0.6703	.9350	0.9673	-0.1396	0.2114
(e)	actual	145.50	0.0500	-0.9786	0.1823	1.0	0.8826	-0.1193	-0.4546
	result	145.51	0.0496	-0.9815	0.1846	1.0	0.8798	-0.1198	-0.4599

Table 1 Motion results for synthetic data

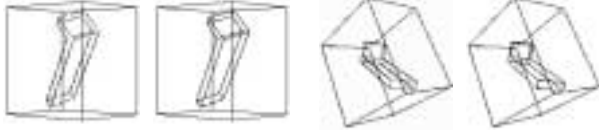


Figure 3 Two views of the computed data (dots) and the actual data (lines).

Renault part is rotated about the vertical axis and translated away from the camera. Figure 4 shows some of the input views. While most of the point correspondences can be established by applying the process described in section 3, the aspect change between some of the close-by views is too large such that correlation cannot give enough correct matches for verification. Frame 7 and frame 8 is such a pair. In that case, we have to input some of the matches manually so that the epipolar geometry can be estimated and then be used to find more correct matches. Clustering is not computed as in section 4.1, but is input since the images are sequenced. We tested several initialization schemes for local estimation. As a typical case, Figure 5 shows the change of image error during the estimation of motion and structure for the cluster with frames 0 to 6. The matches obtained are so noisy that the linear solutions are always wrong, and sometimes unrealizable. If we use these estimations to initialize both the motion and structure parameters, the performance of the algorithm degrades significantly. If we initialize the structure parameters as on a plane parallel to the image plane instead, the algorithm performs like using simple initialization. As the robust algorithm we adopted always produces good motion estimation, the algorithm usually converges to the solution in fewer iterations. While different initialization gives different convergency rate, all

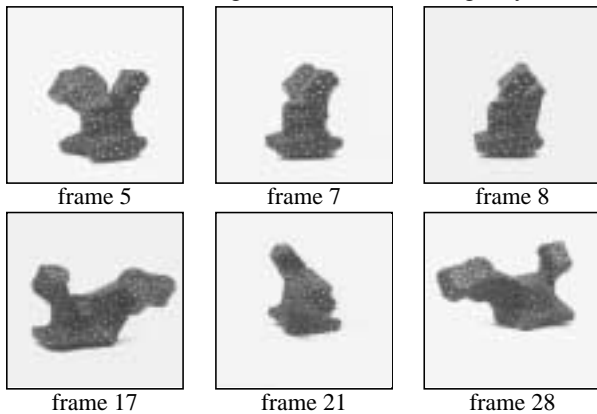


Figure 4 Real images

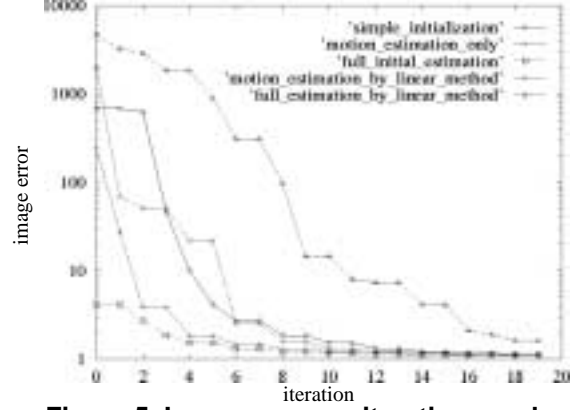


Figure 5 Image error vs. iteration number of them usually lead to the same solution.

After the final estimation, the location of 576 points are recovered with respect to frame 17. The recovered motion parameters are listed in Table 2. While we do not have the complete set of ground truth, the exact values of angle of rotation and distance of translation are known. We also know that rotation is roughly about the y axis and translation is along the z axis. Since the reference point  $p_0$  we chose may not be on the actual axis of rotation, we cannot measure the accuracy of the recovered translation. Generally speaking, we can conclude that the object motion is accurately recovered. Figure 6 shows the recovered structure. Instead of showing the point cloud, we fit surfaces to some of the points and then rendered a few views. Among the set of input images, we triangulates those that show the flattest views of the object and then backproject the triangulation to the 3D space. Despite the poor quality of the fitted surface, we can see that the structure of the object is recovered.

Fr	true rot.	Rotation			Translation				
		angle	axis		dist.	direction			
0	150	150.34	-0.0131	-0.9642	-0.2646	19.21	-0.4058	-0.0861	-0.9098
1	165	165.23	-0.0207	-0.9649	-0.2616	17.49	-0.4538	-0.0992	-0.8855
2	180	179.93	0.0305	0.9632	0.2670	15.49	-0.5025	-0.1172	-0.8565
3	165	164.93	0.0272	0.9693	0.2440	15.86	-0.4675	-0.1213	-0.8756
4	150	153.41	0.0288	0.9711	0.2369	15.00	-0.4603	-0.1313	-0.8779
5	141	141.36	0.0313	0.9720	0.2325	12.39	-0.5078	-0.1612	-0.8462
6	132	132.54	0.0315	0.9737	0.2256	11.90	-0.4824	-0.1684	-0.8595
7	123	119.78	0.0356	0.9758	0.2153	12.23	-0.4293	-0.1634	-0.8882
8	114	114.45	0.0372	0.9768	0.2105	8.43	-0.5427	-0.2336	-0.8067
9	105	106.11	0.0364	0.9782	0.2042	6.86	-0.5699	-0.2763	-0.7738
10	96	96.77	0.0429	0.9783	0.2025	5.17	-0.6277	-0.3492	-0.6956
11	90	90.52	0.0439	0.9787	0.2004	5.34	-0.5302	-0.3227	-0.7840
12	75	74.18	0.0538	0.9797	0.1929	3.37	-0.5422	-0.4472	-0.7112
13	60	58.86	0.0698	0.9802	0.1851	2.57	-0.3912	-0.4833	-0.7831
14	45	42.77	0.0704	0.9807	0.1820	2.70	-0.1509	-0.3514	-0.9239
15	30	29.06	0.0770	0.9821	0.1716	2.90	-0.0186	-0.2219	-0.9748
16	15	15.61	0.1413	0.9827	0.1192	0.58	0.1396	-0.5479	-0.8247
18	15	15.77	0.1401	-0.9516	-0.2733	1.68	-0.2316	0.1816	0.9556
19	24	25.17	0.0850	-0.9675	-0.2381	2.79	-0.2630	0.1774	0.9483
20	33	34.52	0.0617	-0.9719	-0.2270	4.61	-0.2445	0.1399	0.9594
21	42	44.20	0.0448	-0.9752	-0.2163	3.93	-0.4169	0.2018	0.8862
22	51	53.71	0.0361	-0.9776	-0.2073	5.82	-0.3761	0.1533	0.9138
23	60	63.62	0.0300	-0.9788	-0.2023	5.76	-0.4955	0.1741	0.8509
24	69	73.16	0.0260	-0.9789	-0.2023	6.86	-0.5139	0.1563	0.8434
25	75	79.01	0.0223	-0.9799	-0.1980	6.21	-0.6322	0.1796	0.7536
26	90	89.59	0.0458	-0.9725	-0.2281	6.79	-0.7142	0.1793	0.6764
27	105	104.64	0.0332	-0.9745	-0.2215	7.51	-0.7712	0.1703	0.6132
28	120	119.70	0.0231	-0.9745	-0.2227	8.48	-0.7852	0.1545	0.5996
29	135	134.90	0.0121	-0.9733	-0.2288	9.97	-0.7367	0.1335	0.6628
30	150	149.90	-0.0001	-0.9716	-0.2363	11.16	-0.6994	0.1187	0.7047

Table 2 Motion results for real images

hind the image plane. To improve the estimation, we explored various methods that uses different parameterization of the essential matrix, and concluded that better estimation can be obtained by imposing that the rank of the essential matrix to be two. By applying the algorithm in [14], we thus obtain a more robust estimation of the essential matrix from which the camera motion can be computed linearly. Structure and motion of the object can then be estimated linearly too. When there are more than two images in a cluster, more than one reconstruction of the scene can be computed. Merging of structure parameters is done by adjusting the scale of the reconstructed scenes. If a point appears in more than one reconstructed scene, the mean value of all the estimates is used as the location of the point.

In our formulation of the image error function,  $f_0$  and  $p_0$  are the two parameters that need to be specified. We select  $f_0$  as the view that has the largest number of matched points and  $p_0$  as the point that produces projections in most images in the cluster.

### 3.3 Propagation of structure estimation

The correctness of the solution to the motion and structure estimation can be measured by the image error associated with it. If the error is too large, better initial guesses are needed to obtain a good result. New initial guesses are obtained from close-by cluster that share an image with the cluster and have obtained a good solution. Only shape information is propagated because adjacent clusters only share one image.

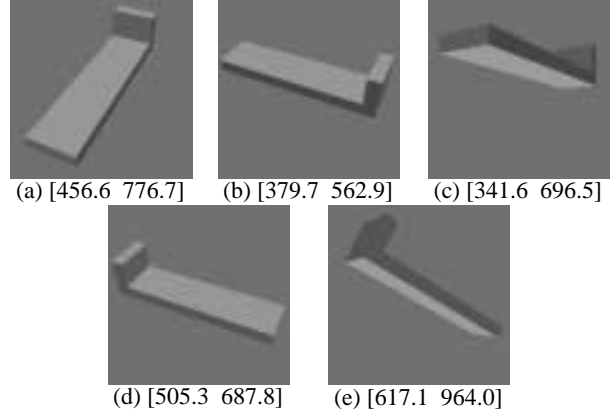
### 3.4 Merging local parameter estimates into a global framework

After the structure and motion parameters are estimated locally, this information is integrated into the global framework. The motion equation for images in adjacent clusters is :

$$\begin{aligned}
p_i^t &= T_{ts} \left( p_i^s \right) \\
&= R_{tr} \left( \left( R_{rs} \left( p_i^s - p_{0_s}^s \right) + t_{rs} + p_{0_s}^s \right) - \left( R_{rs} \left( p_{0_r}^s - p_{0_s}^s \right) + t_{rs} + p_{0_s}^s \right) \right) \\
&\quad + \left( R_{rs} \left( p_{0_r}^s - p_{0_s}^s \right) + t_{rs} + p_{0_s}^s \right) + t_{tr} \\
&= \left( R_{tr} R_{rs} \left( p_i^s - p_{0_s}^s \right) + R_{rs} \left( p_{0_r}^s - p_{0_s}^s \right) + t_{tr} + t_{rs} + p_{0_s}^s \right) \quad (4)
\end{aligned}$$

where frame  $f_r$  corresponds to the image that belongs to both clusters, and  $p_{0_r}$  is the point chosen as the center of rotation in frame  $f_r$ . The relation between images in non-adjacent clusters can then be derived similarly as :

$$\begin{aligned}
p_i^t &= T_{ts} \left( p_i^s \right) \\
&= R_{tr_1} \dots R_{r_2 r_1} R_{r_1 s} \left( p_i^s - p_{0_{r_1}}^s \right) + R_{tr_1} \dots R_{r_2 r_1} R_{r_1 s} \left( p_{0_{r_1}}^s - p_{0_{r_2}}^s \right) + \dots \\
&\quad + t_{tr_1} + \dots + t_{r_2 r_1} + t_{r_1 s} + p_{0_s}^s \quad (5)
\end{aligned}$$



**Figure 2 Synthetic images.  $[z_{near} z_{far}]$  gives the actual range of depth.**

where frames  $f_{r1}, \dots, f_{rI}$  correspond to images belonging to more than one cluster, and  $p_{0_{ri}}$ 's are the centers of rotation. Therefore, given the estimation of the structure and motion parameters for all the image clusters, we can derive the estimation of the motion parameters between any two frames. Structure parameter estimates are obtained by using the global motion parameters to transform the structure parameters from a local framework to the reference frame. These parameter estimates are then refined by minimizing the image error function again.

## 4 Experimental Results

In order to show the advantage of using analytical solution as initial guesses, we apply our algorithm to recover structure and motion from the five synthetic image shown in Figure 2, which involve large motion and deep structure.

254 feature points are chosen randomly on edges and are obtained from the image generator with pixel accuracy. Since the change in aspect of the object is very large, it is difficult to establish correspondences by the method described in section 3. Thus, correspondences are obtained from the image generator too. Given the correspondences, the images are clustered as  $\{[a,b],[a,c],[a,d],[d,e]\}$ . Since the data are relatively noise free, we can obtain very good initial estimation of the camera motion using the linear analytical method. We also try to obtain local structure and motion estimation by using trivial initial guesses as in [8]. Remarkably, although only [a,b] converges to a solution that gives a small image error, by propagating parameter estimates to [a,c] and [a,d], and then to [d,e], the local structure and motion are recovered for all clusters of images. As shown in Table 1 and Figure 3, the final estimation with merged data from either set of local estimation as initial guess gives very accurate result.

We also tested our algorithm with real images. 31 images are chosen from a sequence in which a textured

as the candidate match for the point. The candidate matches are then verified in the relaxation process where continuity constraint is propagated through the neighborhood.

Surviving matches are tested against the epipolar constraint. These matches are used to estimate the underlining epipolar geometry by applying the least-median-of-squares method by Deriche *et al.*[2] to compute the so-called fundamental matrix [3]. Successful estimation indicates that the image pair shows the same aspect of the object. Outliers are thus removed from the list of matches. The estimated epipolar geometry is then used to guide the search for more point correspondence.

### 3 Structure and motion estimation

Given  $n$  point correspondences in  $m$  image frames, we estimate the motion parameters  $T_{j0}$ , for  $j = 1, \dots, m$ , and structure parameters  $p_i^0$ , for  $i = 1, \dots, n$ , with respect to a chosen frame  $f_0$ , by minimizing the nonlinear image error function :

$$\sum_{i=1}^n \sum_{j=1}^m \delta_i^j \left( \|u_i^j - h_u(p_i^j)\|^2 + \|v_i^j - h_v(p_i^j)\|^2 \right) \quad (3)$$

where  $\delta_i^j = \begin{cases} 0 & \text{if } \begin{pmatrix} u_i^j \\ v_i^j \end{pmatrix} \text{ cannot be detected} \\ 1 & \text{otherwise} \end{cases}$

and  $p_i^j$  and  $(h_u, h_v)^T$  are as expressed in (1) and (2) resp.

We apply the Levenberg-Marquardt algorithm to solve for  $T_{j0}$ 's and  $p_i^0$ 's. The major advantages of this algorithm are its simplicity, the descent property, the excellent convergence rate near the solution, and the absence of a line search. However, like all other nonlinear methods, the correctness of the solution is sensitive to how close the initial guess is to the real solution. In our case, since the parameters cover a wide range of values, it is hard to come up with any reasonable simple initial guess. And since the aspect change of the object can be quite large, it is impossible to obtain analytical solutions for all parameters.

However, motion of the camera between close-by views can be computed by analytical methods such as the linear algorithms that use epipolar constraint to determine camera rotation and translation[3][5]. Once the camera motion is determined, structure of the scene and the object motion between the two frames can also be obtained linearly.

We thus proceed by grouping images that are taken from close-by viewpoints into clusters and then estimate the parameters locally for each image cluster. In order to merge these local structure together, we require overlapping of at least one image between close-by clusters. Since the input images cover most aspects of the object, such a clustering should exist. Initial guesses are obtained by applying the analytical method discuss in section 4.2 to solve

for motion between a chosen frame  $f_0$  and the other images in the cluster. Initialization for structure parameters are obtained by merging all reconstructed scenes together through adjusting the scale of each scene. As the initial guesses may not be close to the real solution, the Levenberg-Marquardt algorithm may not converge to the correct solution for some of the image cluster. But we can measure the correctness of the solution by the image error defined in (3). Since image clusters are overlapping, as long as there is a good solution for one of the image clusters, we can propagate the structure information to other close-by clusters. We thus have a better initial guess for the structure parameters for some of the image clusters which in turn can produce good solutions that can be propagate to some other clusters. The local structure and motion parameters are then merged together to give an initial guess for minimizing all the image errors of all views.

#### 3.1 Image clustering

Since images taken from close-by viewpoints contains the projection of the same feature of the object, we can measure the ‘‘closeness’’ of views as the ratio of the number of points that show up in all views to the total number of points in all the images. The clustering thus proceed as follows : we randomly pick an image as the seed of a cluster. The image that is ‘‘closest’’ to the views in the cluster is added. Images are being added to the cluster until the ‘‘closeness’’ ratio of the views in the cluster become low. The seed of a new cluster is chosen from the existing clusters among the images that are least ‘‘close’’ to the seed of its cluster. The process is repeated until all the images are included in some cluster.

#### 3.2 Local structure and motion estimation

For each cluster of images, we compute the structure and motion parameters with respect to a chosen frame  $f_0$  by minimizing the image error function in (3). As mentioned above, initial guesses are obtained analytically by computing the camera motion between frame  $f_0$  and every frame in the cluster using the epipolar constraint. Various algorithms have been published to solve this problem by exploiting the desirable properties of the epipolar constraint, namely, linearity and independency of structure of the scene. We use the well-known linear algorithm by Faugeras *et al.*[3] in our implementation. As to be shown in section 5, the linear algorithm gives very good initial guesses when the images are not corrupted by noise. However, it fails to produce reasonable results when being applied to real images where noisy and wrong feature matches usually occur. In some case, the scene reconstructed by using the camera motion so obtained has some points in front and some points be-

tained analytically from pairs of images. Our experiments showed that while the linear motion recovery algorithm sometimes gives good initial guesses that improve convergence, its sensitivity to noise makes it even secondary to simple initialization. We thus use a more robust motion estimation algorithm by Zhang *et al.* [14] in our initialization stage. Since the correctness of the initial guesses varies, the correctness of the result for each image cluster, as measured by the image error function, may vary. Good estimates are propagated among clusters to modify the initial guess for local structure and motion estimation. This process iterates until all local estimates are reasonably good. Results from different clusters are then merged together to form a global structure which is then refined by the least squares fitting again. Figure 1 shows an overview of our approach.

The paper is organized as follows. Section 2 defines the problem to be solved. Section 3 describes the processes of extracting feature points and establishing correspondences. Section 4 gives the details of our structure and motion estimation algorithm. Experimental results are shown in section 5. Section 6 concludes this paper with a discussion.

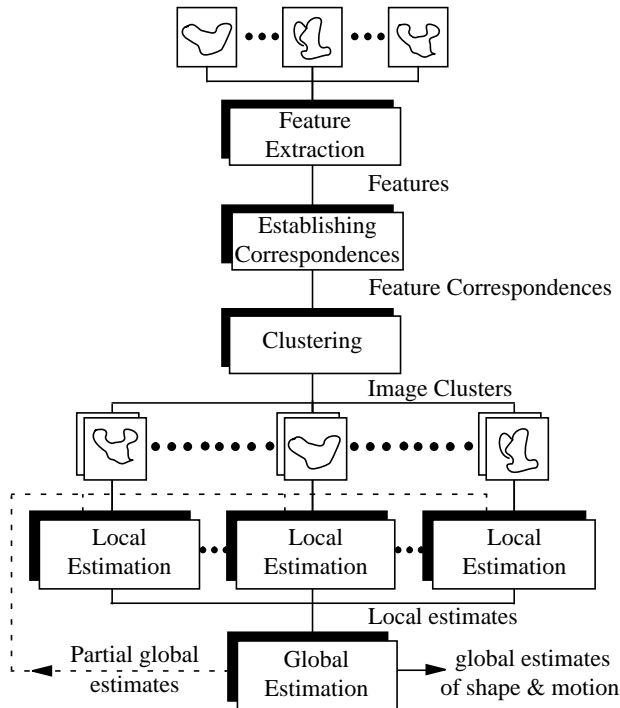


Figure 1 Overview of our approach

## 1 Problem Formulation

We deal with the problem of recovering structure of an object from a set of unregistered images that show all the different aspects of the object. We consider the situation

where the images are taken by a fixed camera at different time instants in a dynamic scene where the object is seen. The world coordinate system is defined by the camera position and orientation, while the rigid transformation  $T_{kj}$  that relates the position of any point  $p_i^j = (x_i^j \ y_i^j \ z_i^j)^T$  in frame  $f_j$  to its position  $p_i^k$  in frame  $f_k$  is formulated as:

$$p_i^k = T_{kj}(p_i^j) = R_{kj}(p_i^j - p_0^j) + t_{kj} + p_0^j \quad (1)$$

where  $p_0^j$  is a point chosen on the object,  $R_{kj}$  is a rotation matrix and  $t_{kj}$  is a vector that represents translation. We represent each rotation by a unit quaternion:

$$q = \left( q_0, q_1, q_2, \sqrt{1 - (q_0^2 + q_1^2 + q_2^2)} \right)$$

The translation vectors  $t_{kj}$ 's, the three independent parameters of the rotation quaternion  $q_{kj}$ 's, and the shape parameters  $p_i^{0j}$ 's for a chosen frame  $f_0$  are the parameters to be estimated.

We model the camera as pinhole, and we assume the camera is strongly calibrated, that is, the intrinsic parameters that specify the mapping of the normalized image coordinates to the retinal image coordinates are known. By choosing the center of the image plane as the origin of the world coordinate system, the projection equation that relates the 3D location of a point  $p_i^j$  to its image location  $(u_i^j \ v_i^j)^T$  becomes:

$$\begin{pmatrix} u_i^j \\ v_i^j \end{pmatrix} = \begin{pmatrix} h_u(p_i^j) \\ h_v(p_i^j) \end{pmatrix} = \begin{pmatrix} \frac{x_i^j}{1 + \beta z_i^j} \\ \frac{y_i^j}{1 + \beta z_i^j} \end{pmatrix} \quad (2)$$

where  $\beta = 1/f$  is the inverse focal length. This formulation decouples the camera representation from the structure representation. The advantages of choosing this equation over the conventional projection equation can be found in [1].

## 2 Feature extraction and matching

We use the keypoint detector developed by Rosenthaler *et al.*[7] to extract feature points. These points can correspond to high curvature points on the object, 2D corners in the texture pattern on the object, or junctions due to occlusion.

We need to establish the correspondences of these points in the images. Since the displacement of corresponding point in two images can be quite large, we perform matching within the multiresolution (pyramid) structure in a coarse-to-fine fashion. A Gaussian pyramid is constructed for each image. For all pairs of images, correspondences are established at the lowest level of resolution by applying correlation operation on a small window. Several potential matches are saved and then propagate up to the highest level of resolution. The match with the highest score is chosen

# Structure and Motion from a Sparse Set of Views

Mi-Suen Lee\*

Gerard Medioni

Institute for Robotics and Intelligent Systems  
University of Southern California  
Los Angeles, California 90089-0273

Rachid Deriche

INRIA-Sophia Antipolis  
BP 93, 2004 route des Lucioles  
06902 SOPHIA-ANTIPOLIS Cedex, France

## Abstract

*We address the problem of acquiring 3D information of an object from multiple images. While long image sequence contains more clues about the motion of the object in the scene, it provides no more information about the object than a few images that show various aspect of the object. We propose an algorithm that uses nonlinear least squares fitting to compute structure and motion from a small number of images in which various aspect of an object is shown. The location of features that show up in different aspect of the object are computed with respect to a single reference frame. As with all other nonlinear problems, our algorithm requires initial guesses. While we adopted an analytical method in the initialization stage, experimental results on synthetic data and real images show that the quality of our solution does not degrade with the accuracy of the initial guesses.*

## 1 Introduction

The problem of extracting structure and motion from intensity images has been studied extensively in the computer vision community. Numerous approaches have been developed to tackle various form of the problem, ranging from classical methods that use only two frames and/or a few points [5][11][12] to methods that use long sequence of images [1][8][10] and higher order features such as lines and curves [4][9]. Most of these methods either only produce structure information for a single aspect of the object and/or require a long image sequence.

In this paper, we address the problem of building complete 3D models for objects from a few intensity images taken at various unknown viewpoints. We aim at recovering the location of all the points visible in the input images with respect to a single reference frame. In contrast to most of the methods that use multiple images, we do not need dense, ordered, image sequences. The input images are taken from viewpoints that show many different aspects of the

object. While the burden of processing a large number of images is alleviated, we have to deal with larger change in aspect of the object and larger motion between views. In this case, most features only show up in a couple of views. Moreover, due to large motion between views, it is harder to establish feature correspondences. In this paper, we propose an algorithm that uses nonlinear least squares fitting to solve for structure and motion parameters without *a priori* knowledge of the shape or motion, while handling partial or wrong feature matches.

The least squares formulation allows us to deal with various camera models, partial or uncertain feature matches across images, and different kind of features, such as points and lines, simultaneously. Weng *et al.* [12] showed that optimal estimation of structure and motion from two images in the presence of noise amounts to minimizing the nonlinear image error function. However, as with all other nonlinear problems, the solution to this minimization involves an iterative process that requires, and is sensitive to, initial guesses. In [12], initial guesses for motion parameters were obtained linearly using the epipolar constraint. Structure and motion were then solved for alternately. Despite the observation that linear algorithms exhibit high sensitivities to noise, the robustness of this approach was unaccounted for in [12]. More recently, Szeliski and Kang [8] reformulated the image error function so as to avoid an initialization stage based on algebraic or linear reconstruction algorithms. By minimizing the reformulated nonlinear image error function with simple initial guesses, their algorithm can recover 3D shape and motion simultaneously from image streams. While our approach applies a similar nonlinear least squares technique to solve the problem, we deal with larger change in aspect of the object and larger motion between views, where a single trivial initial guess cannot give reasonable result.

We approach the problem as follows. Images that are taken from close-by viewpoints are clustered together. For each cluster of images, we use the Levenberg-Marquardt algorithm [6], a general purpose optimization technique, to solve for structure and motion parameters locally by minimizing the image error function. Initial guesses are ob-

---

\* This work was done in part at INRIA-Sophia Antipolis, supported by the NSF-INRIA US-France Cooperative Research (INRIA) grant (INT-9214760) : "Geometric Reasoning about 3-D objects".