

A Semi-automatic System to Infer Complex 3-D Shapes from Photographs

Qian Chen and Gérard Medioni

University of Southern California, Los Angeles, CA 90089-0273, USA
{chenqian,medioni}@iris.usc.edu

Abstract

We present a semi-automatic image-based modeling system to make 3-D models from photographs. The human operator intervenes only for simple and straightforward tasks. The design of such a system is motivated by applications where the geometric shapes of the objects are complex, and the requirement on the appearance of the models is high. Ease of use is another consideration. We use binocular stereo as the basic model recovery tool, and propose a new volumetric matching algorithm that fully explores the spatial coherence between the two images. Inaccurate matches are mostly clustered around the border area between the foreground and the background, thus easy to be manually removed. To make the system more practical, we insist on using uncalibrated cameras. Experimental results on different types of real objects are presented.

1. Introduction

Blending different visual elements, such as two video streams or a virtual object within a real scene is an emerging area of research. In *teleconferencing*, several geographically separated participants are brought into one virtual environment where certain interaction happens. The idea presented in [7] is to track, at one end of the network, individual data such as face orientation and expression, and use this information to drive models at the other end of the network. In such a system, it is critical to have high fidelity human face models in order to produce the sense of “being-there”.

As another example, we have recently started to explore the concept of *cyber touch*. The expectation is to enable people who are browsing the web page of a museum not only look, but also “touch” the objects through the use of haptic devices [14].

One way of obtaining models is to use a laser scanner [4]. Frequently, however, practical situation may prevent the objects from being scanned for various reasons including cost, complexity and delays involved. Creating models from photographs thus provides a convenient and cheap alternative.

Automatic shape recovery from images is one of the research goals of computer vision. In particular,

many algorithms have been proposed to solve the *binocular stereo* problem. There are two main difficulties, however. One of them is that, for real images, there exist multiple similarity peaks. Seldom can an algorithm be robust enough to eliminate the wrong matches to the degree that no obvious irregularity appears in the reconstructed model. The other difficulty has to do with camera calibration, a tedious process usually performed off-line. The calibration result is only valid in a certain depth range. The accuracy degrades as the actual depth departs from the original set-up.

Knowing the difficulties of the automatic methods, researchers in the graphics arena have developed interactive techniques. One such effort was [6] where human face models were created from two orthogonal views. The orthographic assumption severely limits both the application, and the quality of the results. In [8], multiple perspective views are accepted, from which both the points’ coordinates and the cameras’ poses are estimated using least square pose estimation. This removes the assumption of orthographic projection. Furthermore, the authors adopt a radial-basis function in the interpolation process which is known to be C^2 continuous.

The drawback of the above and other purely interactive techniques [12], [13] is twofold. First, only a sparse set of points is generated in 3-D. All remaining ones are computed through interpolation whose accuracy is unknown and hard to control. Also, accurate selection of many correspondences in multiple photographs requires a high level of interaction, and is labor intensive.

To resolve the dilemma, we have adopted a semi-automatic approach, that is, relying on computer vision techniques to recover the model automatically, while allowing a human operator to initialize the automatic process and to decide which part of the reconstruction is acceptable. Figure 1 is a flowchart of our system where the dashed boxes indicate that the tasks may require a small amount of human involvement.

The input to the system is a pair of stereo images. The user selects ~ 12 corresponding points whose locations are refined by automatic methods. The system may reject a selected pair of points if they do

not match well. The so-called *fundamental matrix* is computed. Alternatively, in case the images contain many high intensity curvature points (e.g. eye corners of a human face), the approach [11] can establish correspondences automatically, bypassing manual selection. The fundamental matrix is used to align the two images to a common image plane. Once the images are aligned, the automatic matching process begins. The output is a dense *disparity map* which encodes the horizontal displacement between the two images. The shape inference step converts the disparity information into Euclidean coordinates. Some spurious points may appear at the border between foreground and background, due to depth discontinuities, which are difficult to eliminate automatically. This again necessitates human intervention.

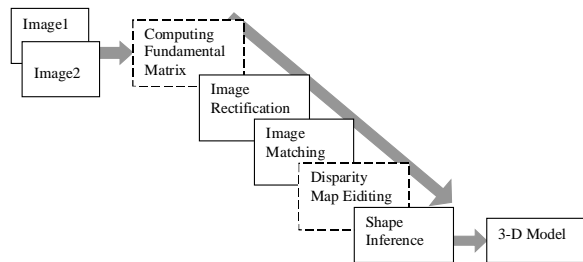


Figure 1. System flowchart

Our paper proceeds as follows. In section 2, we present the image alignment algorithm. Section 3 describes the matching algorithm. Section 4 addresses shape inference. Several examples are presented and explained in section 5. We conclude the paper in section 6.

2. Image Rectification

In general, the two cameras used to generate the stereo images are not parallel. The task of rectification is to align the two image planes so that they are coplanar and their scanlines are horizontally parallel. Our algorithm relies on the so-called *epipolar geometry* between two views of the same scene, which is algebraically described by the fundamental matrix. We treat the image plane as a two-dimensional projective space P^2 which has the following properties.

In P^2 , points and lines are dual entities which satisfy the *Principle of Duality* – for any projective result established using points and lines, a symmetrical result holds in which the roles of lines and points are interchanged: points become lines, and vice versa.

A transformation is represented by a 3×3 matrix

with 8 independent coefficients (there is a scale factor). Thus, it is determined by 4 point correspondences.

In Figure 2, l_1, r_1 and l_2, r_2 (dark) are two pairs of *epipolar lines*. v_1 is the average of the y coordinates of the four end points of l_1 and r_1 , and v_2 is that of l_2 and r_2 . The rectification transformation, one for each image, maps the dark trapeze to the grey rectangle.

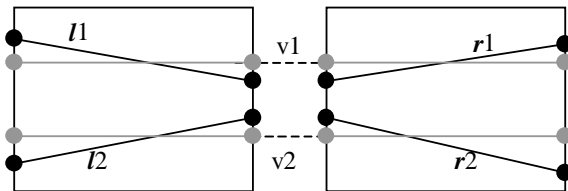
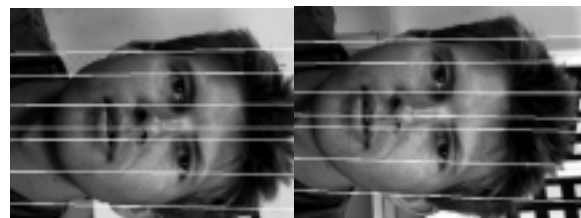
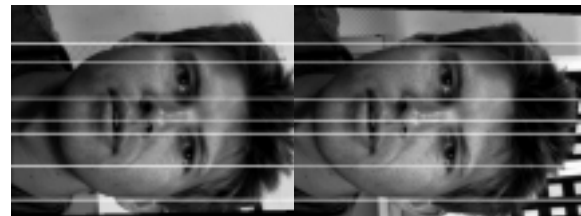


Figure 2. Illustration of the rectification algorithm

Figure 3 depicts the rectification process where the epipolar lines are superimposed. Clearly the goal has been achieved, that is, the scanlines are parallel in each image and the corresponding ones are collinear across the images. We would like to point out that the only information needed in our formulation is the fundamental matrix which can be recovered from 8 pairs of point correspondences. Performing full camera calibration is thus avoided.



(a) before rectification



(b) after rectification

Figure 3. Image rectification

3. Image Matching As Extremal Surface Extraction

As summarized in [3], image matching is typically formulated as a search optimization problem based on some local similarity measurement. At the same time, global constraints are enforced to resolve ambiguities (multiple matches). Traditionally, the correspondence

information is represented by *disparity* – the difference between the \mathbf{x} coordinates of the two matching pixels. If we treat the disparity d as a function of the pixel coordinates (u,v) in one of the images, then $d(u,v)$ defines a piecewise continuous surface over the domain of the image plane. This surface is often referred to as the *disparity surface*. From this point of view, what is actually achieved by the search is to locate, in an abstract 3-D space, the disparity surface. The output – *disparity map*, records the disparity value d for each pixel (u,v) .

The key idea of our approach is to embed the disparity surface into a volume, and assign to each voxel a value proportional to the probability of the corresponding voxel being on the disparity surface. This way the image matching problem is converted into an extremal surface extraction problem. Compared to scanline-based approaches [2], spatial coherence is better utilized in ours. Compared to [5], rather than fit algebraic surface patches, we find discrete surface patches using volume rendering techniques.

3.1 The “u-v-d volume”

We use the normalized cross-correlation over a window as the similarity measurement of two potential matches:

$$\Phi(u,v,d) = \frac{\text{Cov}(W_l(u,v), W_r(u+d,v))}{\text{Std}(W_l(u,v)) \cdot \text{Std}(W_r(u+d,v))} \quad (1)$$

where W_l and W_r are the intensity vectors of the left and right windows of size ω centered at (u,v) and $(u+d,v)$ respectively, d is the disparity, “Cov” stands for covariance and “Std” for standard deviation. The width and height of the (left) image together with the range of d form the *u-v-d volume*. The range of the Φ function is $[0,1]$. When Φ is close to 1, the two pixels are well correlated, hence have high probability of being a match. When Φ is close to zero, that probability is low. In implementation, a threshold needs to be set. We discuss how to choose its values in later sections.

3.2 Disparity surface extraction

The fact that Φ is a local maximum when (u,v,d) is a correct match means that the disparity surface is composed of voxels with peak correlation values. Matching two images is therefore equivalent to extracting the maximal surface from the volume. Simply applying the Marching Cubes algorithm will easily fall into the trap of local maxima because the *u-v-d volume* is noisy. We thus implemented a propagation type of algorithm. In addition, we make

use of the Disparity Gradient Limit [1] which states that $|\Delta d|/|\Delta u| < 1$.

3.2.1 Algorithm description

The algorithm undergoes two steps.

Step 1. Seed voxel selection

A voxel (u,v,d) is a seed voxel if, (i) it is unique – meaning for the pixel (u,v) , there is only one maximum at d along the scanline v , and (ii) $\Phi(u,v,d)$ is greater than a threshold $t1$. A seed necessarily resides on the disparity surface. Otherwise, the true surface point (u,v,d') for which $d' \neq d$ would be a second local maximum. To find seeds, the image is divided into a number of buckets. Inside each bucket, pixels are checked randomly until either one seed is found or all pixels have been searched with failure. During the search, the voxel values are cached to save computation time for the next step.

The value of $t1$ indicates the confidence on the seed points and is set close to 1, and is determined so that at least 10 seeds are found. In all the examples that follow, the range of $t1$ is between 0.993 and 0.996.

Step 2. Surface tracing

Simultaneously from all seed voxels, the disparity surface is traced by following the local maximal voxels whose correlation values are greater than a second threshold $t2$. The $|\Delta d|/|\Delta u| < 1$ constraint determines that when moving to a neighboring voxel, only those at $d, d-1, d+1$ need to be checked.

Initially, we store the seed voxels in a FIFO queue. After tracing starts, we pop the head of the queue every time, and check the 4-neighbors of the corresponding pixel. When two surface fronts meet, the one with the greater correlation value prevails. If any new voxels are generated, they are pushed to the end of the queue. This process continues until the queue becomes empty. To enforce smoothness, we assign (u',v',d) higher priority than $(u',v',d-1)$ and $(u',v',d+1)$. If $\Phi(u',v',d)$ already surpasses $t2$, we will not check the other two voxels. To obtain sub-pixel accuracy, a quadratic function is fitted at the newly generated voxel.

$t2$ determines the probability that the current voxel is on the same surface that is being traced. It turns out that the value of $t2$ is not critical. The default value (0.6) is used in all of our examples.

3.2.2 Time complexity

The complexity of the seed selection part is bounded by $O(WHD\omega)$ where W and H are respectively the width and height of the image, D is the range of the disparity, and ω is the size of the

correlation window. This is the major bottleneck, although the actual execution seldom reaches this limit. That of the tracing part is bounded by $\mathbf{O}(WH\omega)$. Since some voxels have already been computed during the first stage, this limit is never reached. Actual running times on examples are given later.

3.3 Example – the Renault auto part

Figure 4(a) and (b) show the *Renault auto part* pair whose sizes are 256x256. The u - v - d volume with disparity range from -20 to 5 is shown in (c). (d) shows the volume after non-maxima suppression which we see is still quite noisy. This indicates that simply using local similarity measurement for correspondence search could be disastrous. (e) is the volume after tracing where the disparity surface appears clearly. Shown in (f), (g), (h) are vertical cross-sections of the volume above each of them respectively. In this example where $W=H=256$, $D=25$, $\omega=9$, the actual running time on a SGI O2 are 20 seconds for the first step, which results in 42 seed points, and 4 seconds for the second step.

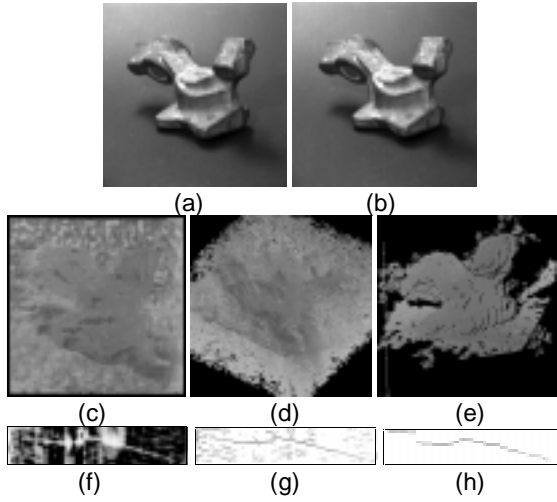


Figure 4. Demonstration of the matching algorithm

4. Shape Inference

The task of shape inference is to convert the dense disparity map into a cluster of Euclidean points. Since our interest is in the shape appearance of the object, a similitude transformation to the final reconstruction is allowed. Assume that the world coordinate system is that of the first camera, and let (x,y,z) be the world coordinates of a point whose images are (u,v) and $(u+d,v)$ respectively. In the first camera $u=fx/z$, $v=fy/z$ where f is the common focal length of both cameras. Let the distance between the two cameras' projection

centers, or the *baseline*, be B , then in the second camera $u+d=f(x+B)/z$. Thus $z=fB/d$. Combining results from both cameras, the coordinates of the point are $(uB/d, vB/d, fB/d)$. This highlights the traditional way of reconstruction, which requires both interior and exterior camera calibration in order to obtain the values for f and B . In the following, we present our reconstruction algorithm which uses *uncalibrated* cameras. In a nutshell, the algorithm runs like this: First, a projective reconstruction is obtained directly from the image correspondences. Then it is upgraded into an Euclidean one by utilizing some inherent properties of the pinhole camera model, which are usually used implicitly.

4.1 Projective reconstruction

Let $X_j, j=1, \dots, n$ be n points whose images are $(u_{ij}, v_{ij}), i=1,2, j=1, \dots, n$. Camera projections are

$$\begin{bmatrix} u_{1j} \\ v_{1j} \\ 1 \end{bmatrix} = \frac{1}{z_i} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \frac{1}{z_i} [A | 0] \begin{bmatrix} X_j \\ 1 \end{bmatrix} \quad (2a)$$

for the first camera, and

$$\begin{bmatrix} u_{2j} \\ v_{2j} \\ 1 \end{bmatrix} = \frac{1}{z_i} \begin{bmatrix} f & 0 & 0 & fB \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \frac{1}{z_i} [A | AT] \begin{bmatrix} X_j \\ 1 \end{bmatrix} \quad (2b)$$

for the second camera, where A is a 3×3 diagonal matrix with elements $(f, f, 1)$, and $T=[B,0,0]^T$.

We can go one step further by putting all image data into one **measurement matrix** W :

$$W = \begin{bmatrix} \begin{bmatrix} u_{11} \\ v_{11} \\ 1 \\ u_{21} \\ v_{21} \\ 1 \end{bmatrix} & \dots & \begin{bmatrix} u_{1n} \\ v_{1n} \\ 1 \\ u_{2n} \\ v_{2n} \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} A & 0 \\ A & AT \end{bmatrix} \begin{bmatrix} X_1 & \dots & X_n \\ z_1 & \dots & z_n \\ \frac{1}{z_1} & \dots & \frac{1}{z_n} \end{bmatrix} \\ = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}_{6 \times 4} [R_1 \dots R_n]_{4 \times n} \quad (3)$$

Hence the rank of W is 4 and a **rank-4-factorization** of W generates simultaneously the camera projection matrices and the points' homogeneous coordinates. However, such a factorization is not unique. For any non-singular 4×4 matrix H , $W=(PH^1)(HR)$. Any such obtained reconstruction is different from the "true" one by a projective transformation, hence projective reconstruction. To compute the factorization, we perform Singular-Value-Decomposition (SVD) on W

and zero out all singular values except the four largest ones. The difference between an earlier algorithm [9] and ours is that, in the former, an affine camera model was assumed.

4.2 Euclidean reconstruction

Given an arbitrary projective reconstruction P and R , Euclidean reconstruction amounts to finding the matrix H such that $P_1 H^1 = [A \mid 0]$ and $P_2 H^1 = [A \mid AT]$ where A and T have the previously specified forms. Accordingly, R is replaced by HR .

Let us separate H into a 3×4 part H' and a 1×4 part h^T , and rewrite the equations as:

$$P_1 = \begin{bmatrix} A & 0 \\ & H' \\ & h^T \end{bmatrix} = AH', \text{ and}$$

$$P_2 = \begin{bmatrix} A & AT \\ & H' \\ & h^T \end{bmatrix} = AH' + Ath^T. \quad (4)$$

As a result, $H' = A^{-1}P_1$ and $Th^T = A^{-1}(P_2 - P_1)$. Th^T is the tensor product of T and h . Since only the first component of T is non-zero, h should be parallel to the vector coming from the first row of $A^{-1}(P_2 - P_1)$. The length of h determines the overall scale of the Euclidean reconstruction and is not critical. Consequently, we simply let h equal to that vector and set t to 1. The implication is that we have chosen the baseline as the unit distance, and have a reconstruction with some arbitrary scale.

We have in fact converted the projective reconstruction into an affine one with an unknown scale in the z -axis which is controlled by the focal length f . We can interactively adjust f until the reconstruction looks satisfying to us. Further non-linear refinement on f can be carried out using methods such as [10].

5. Results

Example 1 – the Renault auto part

The original image pair and the computed disparity surface have been displayed in Figure 4 already. To reconstruct an object like this using interactive techniques only would be difficult because its shape is very generic: there are no obvious geometrical properties such as parallelism, orthogonality, symmetry, etc.; neither are there any distinguished feature points that can be easily picked. It becomes possible, however, with our semi-automatic approach. As mentioned in the introduction, the only places where human assistance is used are, (i) initially choosing several pairs of points to compute the fundamental matrix, and (ii) culling from the disparity map the portion corresponding to the part

itself. Two texture-mapped views of the reconstruction are shown in Figure 5.

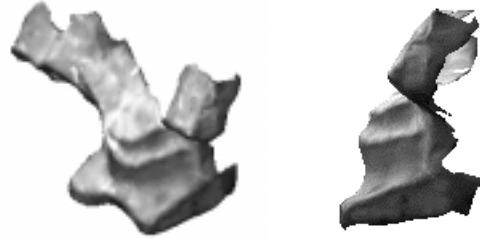


Figure 5. Reconstruction of the *Renault auto part*

Example 2 – human faces

The first example is *Hervé's face* which we have seen in Figure 3. Several textured views of the reconstructed frontal face are shown in Figure 6(a), which is culled from the disparity map in (b). The initial correspondences for computing the fundamental matrix were automatically recovered using the software from [11].



(a) reconstruction with texture (b) disparity map

Figure 6. Results on *Hervé's face*

Figure 7 is another face example where the person is smiling. Both shaded and textured views are shown.



Figure 7. *Doug's face* – original and reconstructed

It is clear that major face features such as the eyebrows, nose, mouth have been faithfully recovered. We would like to point out that face is the portion of a head that is the most difficult to recover. The skull, in contrast, can be estimated from profiles using interactive techniques [6] because it is mostly covered by hair, and its exact shape in most applications is not critical.

Example 3 – a complete head

To obtain a complete head model, a Styrofoam head was placed on a rotary table. Six pairs of stereo images were taken. Table rotation was arranged so

that the frontal face has more coverage than the back. One of the input images is shown in Figure 8(a). To find out the transformation between the default world coordinate system – that of the first camera – and the rotary table’s coordinate system, a cube was placed on the rotary table. Care was taken so that the lower corner of the cube was at the center of the platform. Figure 8(b) depicts the reconstructed head after fusing points from all partial reconstruction. The top of the head was cut off because this portion was unobservable from the cameras. From the point cloud, a surface was constructed. The difference between (c) and (d) was due to the size of the convolution kernel.

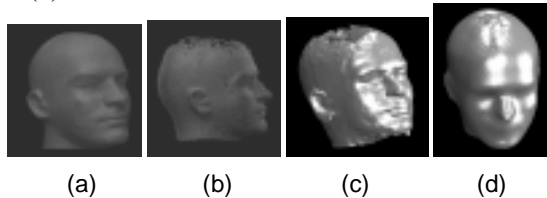


Figure 8. Original and reconstruction of a complete head

Example 4 – a complete teacup

To understand what our technique can do for the *cyber touch* project, the museum staff has chosen a teacup as the testing object whose shape is even more complex than a human head. Six stereo images around the teacup were taken at 60° apart. The one at the 0° is shown in Figure 9(a). (b) shows the reconstruction as a colored point cluster. (c) and (d) show the surface reconstruction. Compared to the original teacup, the reconstruction (body, lid, handle and mouth) appears plausible.

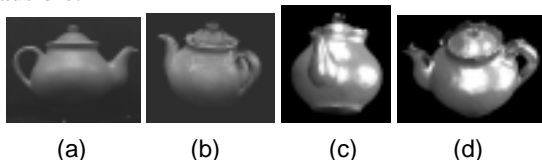


Figure 9. Original and reconstruction of a teacup

6. Conclusions

In traditional computer vision applications such as aerial image analysis and robot navigation, human intervention is considered a half-failure because the real-time requirement is not fulfilled. In multimedia and virtual reality applications, the situation is just reversed – it is unwise not to take advantage of the human operator although it is still desirable to minimize the human involvement. In this sense, our semi-automatic system provides the right trade-off. This is possible mainly because of the volumetric matching method that appears to handle the ambiguity problem very well, leaving only inaccurate matches at

the border. In addition, to make the system more practical, we have proposed algorithms that use uncalibrated cameras, bypassing the tedious calibration procedure.

References

- [1] P. Burt and B. Julesz, “A disparity gradient limit for binocular fusion”, *Perception*, 9:671-682 (1980).
- [2] Cox, S. Hingorani, S. Rao, B. Maggs, “A Maximum-Likelihood Stereo Algorithm”, *Computer Vision and Image Understanding*, 63(3):542-567 (1996).
- [3] U. Dhond and J. Aggarwal, “Structure from stereo - a review”, *IEEE T. Systems, Man and Cybernetics*, 19(6):1489-1510 (1989).
- [4] B. Guenter, C. Grimm, D. Wood, H. Malvar, F. Pighin, “Making Faces”, *SIGGRAPH 98*, 55-66, (1998).
- [5] W. Hoff and N. Ahuja, “Surfaces from stereo: integrating feature matching, disparity estimation, and contour detection”, *IEEE T. Pattern Analysis and Machine Intelligence*, 11(2):121-136, 1989.
- [6] Horace H.S.Ip, L. Yin, “Constructing a 3D individualized head model from two orthogonal views”, *Visual Computer*, 12:254-266, (1996).
- [7] U. Neumann, R. Enciso, T.-Y. Kim, D. Fidaleo, J.-Y. Noh and J. Li, “A Real-Time Tele-Immersion Prototype”, submission to ACM Symposium on Interactive 3D Computer Graphics 99.
- [8] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, D. Salesin, “Synthesizing Realistic Facial Expressions From Photographs”, *SIGGRAPH 98*, 75-84, (1998).
- [9] C.J.Poelman and T. Kanade, “A Paraperspective Factorization Method for Shape and Motion Recovery”, *IEEE T. Pattern Analysis and Machine Intelligence*, 19(3):206-218, 1997.
- [10] M. Pollefeys, R. Koch and L. V. Gool, “Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters”, *Proc. Int’l Conf. Computer Vision 98*, 90-95, 1998.
- [11] Z. Zhang, R. Deriche, Q.-T. Luong and O. Faugeras, “A Rotust Approach to Image Matching: Recovery of the Epipolar Geometry”, *Proc. Int’l Symposium of Young Investigators on Information-Computer-Control*, 7-28, Beijing, China, 1994.
- [12] Web site, *3D Construction Company*, <http://www.3dconstruction.com>.
- [13] Web site, *Eos Systems Inc.* <http://www.photomodeler.com>
- [14] Web site, *IMSC Interactive Art Museum* <http://imsc.usc.edu/Research/ci/art.html>