Triangulation without Correspondences *

Yong-Qing Cheng, Robert T. Collins, Allen R. Hanson, Edward M. Riseman

Department of Computer Science
University of Massachusetts
Amherst, MA. 01003

Abstract

This paper presents two different algorithms for reconstructing $3D$ points from two sets of noisy $2D$ image points without knowing point correspondences given the corresponding poses from the two images. We first present a new way to form a $2D$ similarity function between two points from two images via $3D$ pseudo-intersection. Based on principles of proximity and exclusion, the first algorithm uses a new affinity measure between $2D$ image points from two different images and a competition scheme to establish image point correspondences and recover their corresponding $3D$ points simultaneously. Based on an optimal graph theoretic approach, the second algorithm uses the similarity function to construct a bipartite graph, builds a corresponding flow network, and finally finds a maximum network flow that determines the correspondences between two images. The two proposed algorithms have been applied to aerial images from the ARPA RADIUS project. Experimental results have shown that the proposed algorithms are robust.

1 Introduction

A fundamental and important problem in computer vision is to build $3D$ models of objects and scenes from a sequence of images. So far, extensive research has been done to develop robust algorithms in this area [1-16], including monocular motion sequences, stereo pairs, and a set of distinctive views. The basic principle to deal with this problem is a triangulation process. For a general triangulation process, it is assumed that the intrinsic (lens) parameters and extrinsic (pose) parameters of each camera are known, or that the 3×4 projective transformation matrix which represents a relationship between a $3D$ point and its corresponding $2D$ point is known (as in the RADIUS project). Usually, $2D$ features are extracted first such as corners, curvature points, and lines from each frame of an image sequence. Then, the correspondence of these features is established between any two successive frames, i.e., the correspondence problem. Finally, the $3D$ information is recovered from these $2D$ correspondences in the image sequence. The two most extensively used triangulation algorithms are point-based triangulation and line-based triangulation. The reason to use image lines as an alternative to image points is that lines provide a more stable image feature.

Unfortunately, this basic triangulation process assumes the correspondence problem has been resolved. This has caused criticism and doubts about feature-based methods because the process of finding $2D$ image feature correspondences is time consuming and is difficult to implement reliably. This paper addresses this problem.

In recent years, much work has been done on a variety of correspondence problems [5-16]. Many researchers have worked on the problem of motion estimation without correspondences [5-9,12-16]. Aggarwal et al [8] gave an excellent review of the correspondence problem. Aloimonos, et al. [11], presented an algorithm to estimate $3D$ motion without correspondences by combining motion and stereo matching. Recently, Huang and his research group [7,15-16] presented a series of algorithms to estimate rigid-body motion from $3D$ data without matching point correspondences. Goldgof et al. [7] presented moment-based algo-

*This work was funded by the RADIUS project under DARPA/Army contract number TEC DACA76-92-C-0041 and also by the National Science Foundation grant No. CDA-8922572.

rithms for matching and motion estimation of $3D$ points or lines sets without correspondences and applied these algorithms to object tracking over the image sequences. Lee et al. [9] proposed an algorithm to deal with the correspondence problem in image sequence analysis.

Objects in the world can be nonrigid, and an object's appearance can deform as the viewing geometry changes. Consequently, much work has also been done that addresses the problem of correspondence and description by using deformable models[10-11,17-19]. Scott and Longuet-Higgins [10] developed an algorithm to determine the possible correspondences of $2D$ point features across a pair of images without any other information (in particular, they had no information about the poses of the cameras). They first incorporated a proximity matrix description which describes Gaussian-weighted distances between features (based on inter-element distances) and a competition scheme allowing candidate features to contest for best matches. Then they used the eigenvectors of this matrix to determine correspondences between two sets of feature points. Shapiro and Brady [11] also proposed an eigenvector approach to determining point-feature correspondence based on a modal shape description. Recently, Sclaroff and Pentland [19] described a modal framework for correspondence and description.

In this paper, we first investigate the problem of determining image point correspondences given the poses of two images while simultaneously computing the corresponding $3D$ points. Here, camera pose consists of an orientation R_l and a $3D$ position τ_l which map the world coordinate system to the camera coordinate system. The problem can be formulated as follows:

Given two sets L and R of $2D$ image points from two images I_l and I_r: $L = \{p_i^l(u_i, v_i) \mid p_i^l \in I_l, i = 1, 2, ..., n_l\}$ and $R = \{p_j^r(u_j, v_j) \mid p_j^r \in I_r,\ j = 1, 2, ..., n_r\}$, and two corresponding poses (R_l, τ_l) and (R_r, τ_r) for the two images I_l and I_r, the goal is to compute a set of $3D$ points $P_q(x_q, y_q, z_q)$ $(q = 1, 2, ..., n,\ n \leq min\{n_l, n_r\})$ representing n correspondences between L and R without knowing in advance the image point correspondences.

First we present a new way to form a $2D$ similarity function between two points from two images

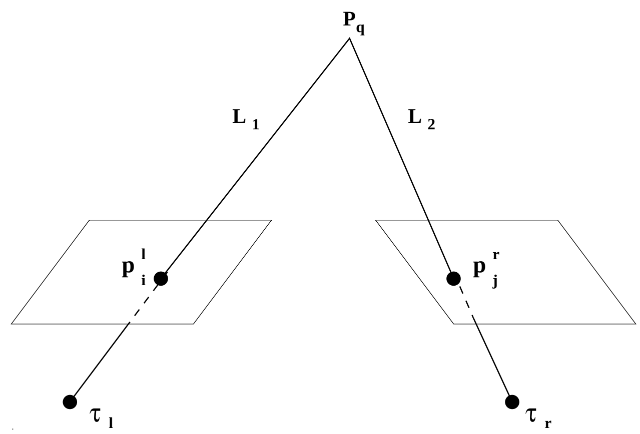

Figure 1: A triangulation process for a pair of images.

via $3D$ pseudo-intersection. Then we present two different algorithms for reconstructing $3D$ points from noisy $2D$ image points without knowing the point correspondences. The first algorithm uses a new version of an affinity measure to extend the work in [10] via the principles of proximity and exclusion. Our second algorithm is based on an optimal graph theoretic approach using the similarity function to construct a bipartite graph, build a corresponding flow network, and finally find a maximum network flow that determines the correspondences between two images. The two proposed algorithms have been applied to aerial images from the ARPA RADIUS project. Experimental results have shown that the proposed algorithms are robust.

2 2D similarity function via 3D pseudo-intersection

Given two poses (R_l, τ_l) and (R_r, τ_r) from two images I_l and I_r, for any pair of $2D$ points p_i^l and p_j^r $(i = 1, 2, \ldots, n_l;\ j = 1, 2, \ldots, n_r)$ from I_l and I_r, there exist two $3D$ lines L_1 and L_2 such that L_1 passes through points p_i^l and τ_l, and L_2 passes through points p_j^r and τ_r, as shown in Figure 1. L_1 and L_2 are the *projection lines* of points p_i^l and p_j^r, respectively.

Suppose each projection line L_k $(k = 1, 2)$ is defined as

$$\frac{x - x_k}{u_{xk}} = \frac{y - y_k}{u_{yk}} = \frac{z - z_k}{u_{zk}} \tag{1}$$

with unit direction vector $\vec{u_k} = (u_{xk}, u_{yk}, u_{zk})^T$.

Consider first how to compute an optimal $3D$ *pseudo-intersection point* $P_q(x_q, y_q, z_q)$ with the smallest sum of distances from $P_q(x_q, y_q, z_q)$ to two lines L_1 and L_2. The error function can be defined [7] as

$$
\begin{aligned}
E = \ & [(x_q - x_k)u_{yk} - (y_q - y_k)u_{xk}]^2 \\
& +[(x_q - x_k)u_{zk} - (z_q - z_k)u_{xk}]^2 \\
& +[(y_q - y_k)u_{zk} - (z_q - z_k)u_{yk}]^2
\end{aligned}
\tag{2}
$$

After setting $\frac{\partial E}{\partial x_q} = \frac{\partial E}{\partial y_q} = \frac{\partial E}{\partial z_q} = 0$, we obtain the optimal $3D$ pseudo-intersection point $P_q(x_q, y_q, z_q)$[7]

$$
\begin{bmatrix} x_q \\ y_q \\ z_q \end{bmatrix} = \left[\sum_{k=1}^{2} A_k \right]^{-1} \left[\sum_{k=1}^{2} \left(A_k \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} \right) \right] \tag{3}
$$

where

$$
A_k = \begin{bmatrix} u_{yk}^2 + u_{zk}^2 & -u_{xk}u_{yk} & -u_{xk}u_{zk} \\ -u_{xk}u_{yk} & u_{xk}^2 + u_{zk}^2 & -u_{yk}u_{zk} \\ -u_{xk}u_{zk} & -u_{yk}u_{zk} & u_{xk}^2 + u_{yk}^2 \end{bmatrix}
$$

If p_i^l and p_j^r are the corresponding image points from two successive images I_l and I_r, then P_q is the real $3D$ point recovered by the traditional triangulation algorithm. However, there are three cases that are exceptions: (1) no $3D$ point could be obtained for p_i^l and p_j^r, because the two $3D$ lines L_1 and L_2 are parallel; (2) an incorrect "negative" $3D$ point could be obtained for p_i^l and p_j^r, due to the two $3D$ lines L_1 and L_2 intersecting behind one or both cameras, as shown in Figure 2; (3) a wrong "epipolar" $3D$ point P_w is obtained corresponding to p_i^l and p_j^r, due to incorrect correspondences, e.g. p_i^l could appear to correspond to either p_j^r or p_w^r as shown in Figure 3. This case shows that a point p_i^l in image I_l could intersect with the projection line of more than one image point in image I_r.

The first case, with parallel projection lines, is exceedingly rare, but is easily be detected by examining whether there exists a solution for Equation (3). It also can be detected by examining whether the directions of the projection lines L_1 and L_2 are the same.

For the second case, as all pairs of image points from two images are considered as possible correspondences, some of those will intersect in their negative directions and satisfy the minimal distance condition to lines L_1 and L_2, but are incorrect. Fortunately, it is easy to detect this kind of

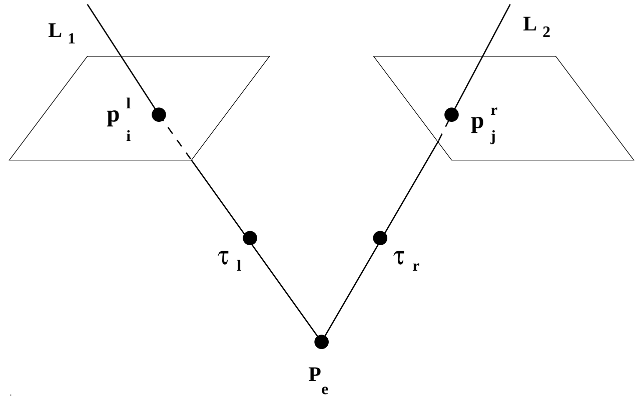

Figure 2: A wrong "negative" 3D point corresponding to a pair of image points.

"negative" $3D$ point by examining the directions of rays $\vec{\tau_l p_i^l}$ and $\vec{\tau_l P_q}$ or $\vec{\tau_r p_j^r}$ and $\vec{\tau_r P_q}$ to make sure that they are the same.

The third case is caused by an incorrect correspondence, often due to ambiguity. For example, as shown in Figure 3, suppose p_i^l corresponds to p_j^r with P_q as the correct $3D$ point. However, the known poses specify epipolar lines, and since p_w^r lies on the known epipolar line of p_i^l in image I_r, then p_i^l and p_w^r could intersect at a $3D$ point P_w. However, this kind of ambiguity might be detected because p_w^l might correspond to point P_w appearing in image I_l. For this case, the maximum correspondences could be detected for two sets of points, i.e., p_i^l corresponds to p_j^r and p_w^l corresponds to p_w^r. Unfortunately, if the point P_w^l doesn't appear in the first image I_l, it is difficult to resolve this inherent ambiguity. In such situations, a third image would greatly reduce such ambiguities.

For any pair of image points (p_i^l, p_j^r), we project the "pseudo-intersection" point P_q into the two images I_l and I_r, then get the two projected image points $p_i^{l'}(u_i', v_i')$ and $p_j^{r'}(u_j', v_j')$:

$$
\begin{aligned}
u_i' &= S_{xi} \frac{(R_l(P_q)+\tau_l)_x}{(R_l(P_q)+\tau_l)_z} \\
v_i' &= S_{yi} \frac{(R_l(P_q)+\tau_l)_y}{(R_l(P_q)+\tau_l)_z} \\
u_j' &= S_{xj} \frac{(R_r(P_q)+\tau_r)_x}{(R_r(P_q)+\tau_r)_z} \\
v_j' &= S_{yj} \frac{(R_r(P_q)+\tau_r)_y}{(R_r(P_q)+\tau_r)_z}
\end{aligned}
\tag{4}
$$

where S_{ui} and S_{vi} are the intrinsic camera scale factors along the "u" and "v" directions on the image plane I_l, and S_{uj} and S_{vj} are the intrinsic

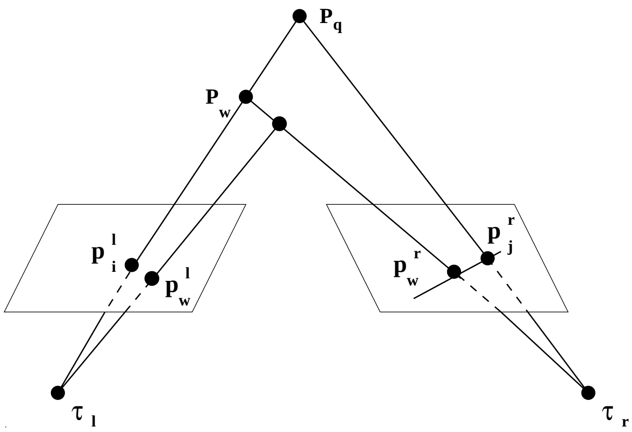

Figure 3: A wrong "epipolar" 3D point corresponding to a pair of image points.

camera scale factors along the "u" and "v" directions on the image plane I_r.

Finally, we compute the error functions E^l_{ij} and E^r_{ij}:

$$E^l_{ij} = \|p^l_i - p^{l'}_i\|_2, E^r_{ij} = \|p^r_j - p^{r'}_j\|_2$$

A $2D$ *similarity function* $sf(p^l_i, p^r_j)$ is defined as

$$sf(p^l_i, p^r_j) = E^l_{ij} + E^r_{ij} = \|p^l_i - p^{l'}_i\|_2 + \|p^r_j - p^{r'}_j\|_2$$

The criterion underlying $sf(p^l_i, p^r_j)$ is that the best estimate for any $3D$ pseudo-intersection point is the point that minimizes the sum of the least-squares distances between the predicted image location of the computed $3D$ point and its actual image locations in the first and second images. if $sf(p^l_i, p^r_j) = \infty$, it means that p^l_i is not similar at all to p^r_j; if $sf(p^l_i, p^r_j) = 0$, it means that p^l_i is perfectly similar to p^r_j.

The next two sections present two algorithms for determining point correspondences using the similarity function $sf(p^l_i, p^r_j)$.

3 Algorithm based on principles of proximity and exclusion

Following the basic idea in Scott and Longuet-Higgins' work[10], we uses pose information to achieve a new powerful version of proximity matrix. The first step is to detect any "negative" $3D$ point P_e and construct a $n_l \times n_r$ proximity matrix H of a Gaussian-weighted error function H_{ij} ($i = 1, 2, ..., n_l$; $j = 1, 2, ..., n_r$) using the similarity function

$$H_{ij} = e^{-sf(p^l_i, p^r_j)/2\sigma^2}$$

where σ is the control parameter for the degree of spatial interaction between the two sets of image points.

The second step is to perform a singular value decomposition (SVD) of H, i.e.

$$H = UDV^T$$

where U and V are orthogonal matrices, and D is a diagonal matrix in which the nonnegative singular values appear along its diagonal in descending numerical order.

The final step is to compute the correlation between U's rows and V's columns and obtain an association matrix A:

$$A = UIV^T = UV^T$$

where superscript T denotes the transpose of a matrix. I was obtained by replacing each diagonal element in D by a 1, i.e. I is an identity matrix. Each element A_{ij} indicates the strength of attraction between p^l_i and p^r_j. If A_{ij}=1, there is a perfect correspondence between p^l_i and p^r_j; if A_{ij}=0, there isn't any affinity between p^l_i and p^r_j at all. The affinity between p^l_i and p^r_j is strong only if A_{ij} is largest in both its row and its column.

4 Algorithm based on maximum network flow

The problem of triangulation without correspondences seems on the surface to have little to do with flow networks, but it can in fact be reduced to a maximum-flow problem. In this section, we show how the problem of triangulation without correspondences is formulated as a maximum flow problem on a flow network.

Given the two sets of points L $= \{p^l_i \mid i = 1, 2, ..., n_l\}$ from I_l and $R = \{p^r_j \mid j = 1, 2, ..., n_r\}$ from I_r, then an undirected bipartite graph $G = (V, E)$ can be constructed as follows: $V = L \cup R, E = \{e_{ij}\}$ in which each edge e_{ij} ($i = 1, 2, ..., n_l$; $j = 1, 2, ..., n_r$) with a unit weight corresponds to a weighted link between p^l_i in I_l and p^r_j

in I_r if the "distance" between them, defined as $sf(p_i^l, p_j^r)$ is less than threshold T_d, and the corresponding optimal $3D$ pseudo-intersection point P_q computed by equation (3) is not "negative". Here, the threshold T_d is chosen empirically. Obviously, the graph arising in such a case is a bipartite graph by construction, since two points in the same image cannot be linked.

Furthermore, from graph theory, we know that given an undirected graph $G = (V, E)$, a *matching* is a subset of edges $M \subseteq E$ such that for all vertices $v \in V$, at most one edge of M is incident on v. A vertex $v \in V$ is matched by M if some edge in M is incident on v; otherwise, v is unmatched. A *maximum matching* is a matching of maximum cardinality, that is, a matching M such that for any matching M', we have $\mid M \mid \geq \mid M' \mid$. Therefore, the problem of triangulation without correspondences can be considered as the problem of finding a maximum matching in a bipartite graph G.

In order to reduce the problem of a maximum matching in the bipartite graph G to a maximum flow problem in the flow network G', the trick is to construct a flow network in which flows correspond to correspondences. We build a corresponding flow network $G' = (V', E')$ for the bipartite graph G as follows: Let the source s and sink t be new vertices not in V, let $V' = V \cup \{s, t\}$, and let the directed edges of G' be given by

$$\begin{aligned} E' = \quad & \{(s,u) : u \in L\} \cup \{(u,v) : u \in L, v \in R, \\ & (u,v) \in E\} \cup \{(v,t) : v \in R\} \end{aligned}$$

and finally, assign unit flow capacity to each edge in E'.

The following theorem [21] shows that a matching in G corresponds directly to a flow in the corresponding flow network G'.

Theorem 1 *Let $G = (V, E)$ be a bipartite graph with vertex partition $V = L \cup R$, and let $G' = (V', E')$ be its corresponding flow network. If M is a matching in G, then there is an integer-valued flow f in G' with value $\mid f \mid = \mid M \mid$. Conversely, if f is an integer-valued flow in G', then there is a matching M in G with cardinality $\mid M \mid = \mid f \mid$.*

Intuitively, a maximum matching in a bipartite graph G corresponds to a maximum flow in its corresponding flow network G'. If so, the correspondence problem is equivalent to finding the maximum flow in $G' = (V', E')$, and we can compute a maximum matching in G by finding a maximum flow in G'. It has been shown [21] that if we use the Ford-Fulkerson method, the maximum flow f computed by it can ensure that $\mid f \mid$ is integer-valued, and the cardinality of a maximum matching in a bipartite graph G is the value of a maximum flow in its corresponding flow network G'. Therefore, the correspondence problem can be exactly reduced to finding the maximum flow in G'. Specifically, our algorithm has the following steps:

The first step which is the same as in the first algorithm, is to compute $sf(p_i^l, p_j^r)$ for all possible pairwise matches between any pair of image points (p_i^l, p_j^r). Then, the second step is to generate a bipartite graph $G = (V, E), V = L \cup R$, where L and R are disjoint and all edges in E go between L and R, such that if $sf(p_i^l, p_j^r) \leq T_d$, then there exists an edge in E from p_i^l to p_j^r. The third step is to build the corresponding flow network $G' = (V', E')$ using the above process. The final step is to use the Ford-Fulkerson method to efficiently obtain a maximum matching M from the integer-valued maximum flow. A complete description of their algorithm can be found in [21] and will not be given here.

Since any matching in a bipartite graph G has cardinality at most $min(\mid L \mid, \mid R \mid) = O(\mid V \mid)$, the value of the maximum flow in G' is $O(\mid V \mid) = O(n_l + n_r)$. We can therefore find a maximum matching M in a bipartite graph G in time $O(\mid VE \mid)$. For each vertex, the number of edges which is incident on the vertex can be considered as a constant. Thus, the time complexity is approximately $O(\mid V \mid^2) = O((n_l + n_r)^2)$.

5 Experimental Results

In this section, we will illustrate the two algorithms and demonstrate their performance on images $J1$ and $J2$ (shown in Figure 4) from the RADIUS "Model Board 1" set.

In order to demonstrate the robustness of the two algorithms, they were compared in the presence of noise. In general, there are two sources of error which contribute to the error of 3D points recovered from two images: (1) localization errors of the 2D image points and (2) errors in the estimate of the intrinsic and extrinsic camera parameters.

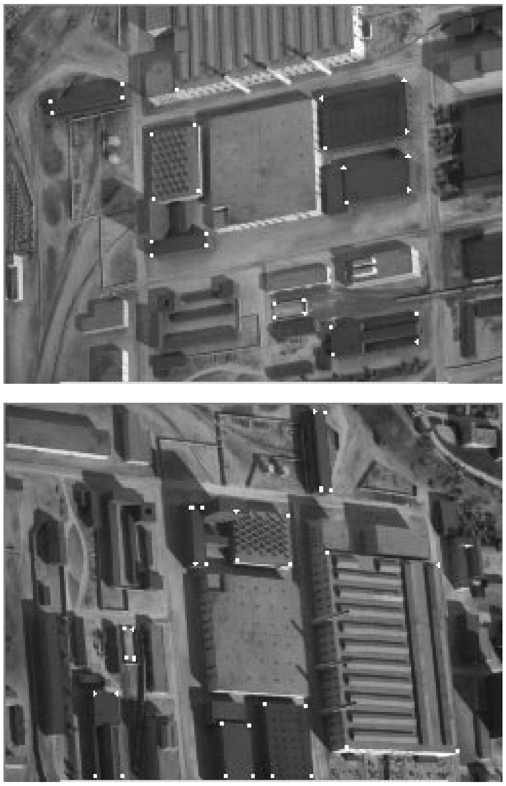

Figure 4: A pair of RADIUS images.

The contribution of the error of the 2D image points from two images is complicated. To aid the error analysis in determining the correspondences between the two images shown in Figure 4, 32 ground truth corner points were selected and projected into each image as shown by the white dots. The image point locations from each image were corrupted by Gaussian noise. Noise for each image point location was assumed to be zero-mean, identically distributed, and independent. The standard deviation ranges from 0.5 pixel to 4.0 pixels. For each level of noise, 100 noisy sample sets were created and the two algorithms were run on each of the samples. From each sample run, the number of incorrect correspondences and the squared distance error between the triangulated point position and its ground truth position were computed. The average number of incorrect correspondences and average triangulation error for each noise level are shown in Figures 5-6, respectively. The experiments have shown that the two algorithms work very well if there is no difference in sizes of image points from two images, i.e. for two images, each 3D point to be recovered

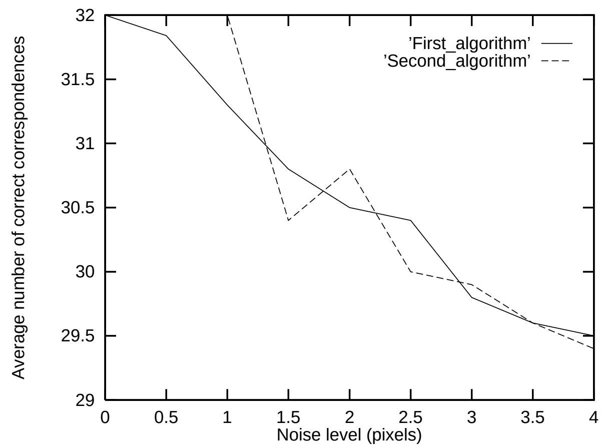

Figure 5: Comparison of number of correct correspondences for the two algorithms as a function of noise.

appears in each image and only their correspondences are unknown.

In order to show how the number of incorrect correspondences is affected by the number of missing points (i.e. no correct correspondence in the other image), new data sets were created by randomly deleting some percentage of image points from the same two sets of 32 image points used above. Also, four levels of Gaussian noise from 1 pixel to 4 pixels were added to these new data sets. The average number of incorrect correspondences over 100 sets of samples for each noise level were computed. Figures 7 and 8 show how the performance of the two algorithms was affected by the number of missing points and the noise. Our experiments have shown that the two algorithms can tolerate a difference in the number of image points from two images and are robust against a reasonable level of noise.

The experimental results [20] also show that it is very difficult to choose an appropriate value for the parameter σ in the first algorithm and a σ that is too large can not be chosen. For the first algorithm to work well, a sufficiently large σ must be chosen, yet it must not be too large since thsi would drive the smaller singular values towards zero. If this occurs, the association matrix becomes unstable. This conforms the conclusion reached in [11].

The second algorithm utilizes a threshold T_d on the error function to build the initial bipartite graph. This threshold must be chosen empiri-

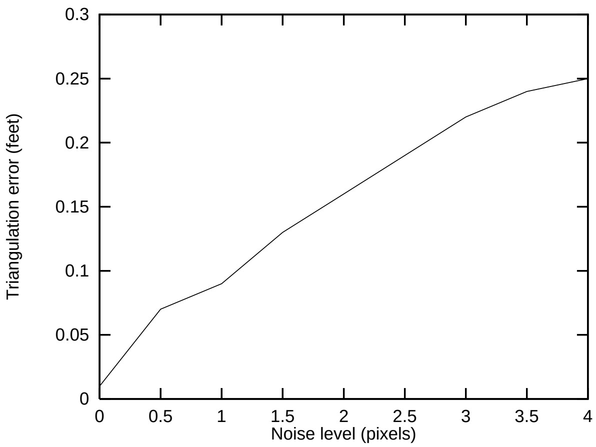

Figure 6: Average triangulation error for the two algorithms.

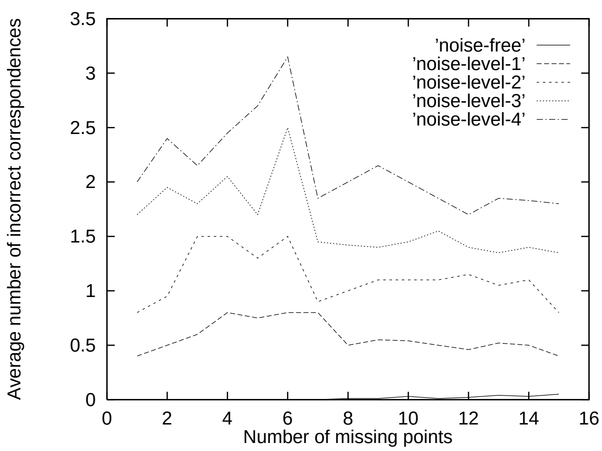

Figure 7: Comparison of number of incorrect correspondences for the first algorithm.

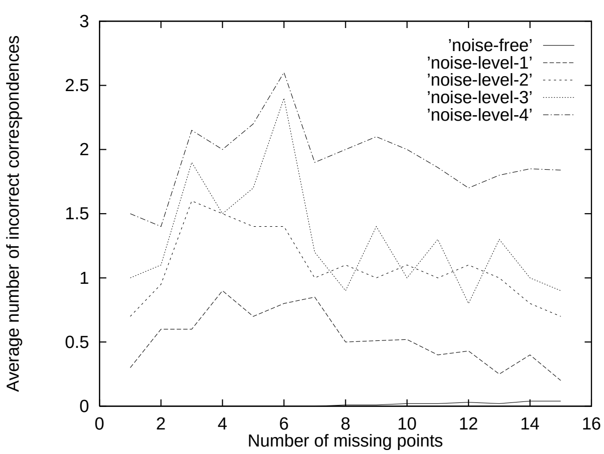

Figure 8: Comparison of number of incorrect correspondences for the second algorithm.

cally. Our experiments showed that it is crucial to choose an approriate value for the threshold T_d.

6 Conclusions and Future Work

Based on a similarity function between two points from two images via $3D$ pseudo-intersection, this paper present two different algorithms to reconstruct 3D points from noisy 2D image points without knowing the point correspondences. The first algorithm is based on a principle of proximity and a principle of exclusion. It first builds a proximity matrix to represent the affinities, then does a SVD decomposition of the proximity matrix to get an association matrix, and finally obtains the correspondences from the association matrix. The second algorithm first reduces the problem of triangulation without correspondences to that of a maximum matching problem of the bipartite graph, then reduces the maximum matching problem to a maximum flow problem of the flow network, and finally determines the correspondences by finding a maximum network flow from the flow network.

The work presented in this paper compared the two algorithms, in terms of robustness with respect to noise and the difference between the set sizes of image points from two images. The experiments showed that the two algorithms are robust. The two algorithms do have several advantages: (1) they can automatically detect the outliers from the $2D$ image points from a pair of images by thresholding the error function for left and right pseudo-intersection projections; (2) they are not too sensitive to noise or the difference between the set sizes of image points from two images; (3) their computational complexity is low; (4) they can be expanded to reconstruct $3D$ lines from noisy $2D$ image lines. From the preliminary experiments conducted thus far, it appears that there is only a small difference between the two algorithms relative to the difference in the size of the point sets. However, there are several reasons to choose the network flow algorithm over the first one: (1) potential instabilities in the association matrix (as described earlier); (2) computational consideration in the association matrix; (3) the network flow is easily extended to multiple images.

For the two algorithms, however, there is some inherent ambiguity which cannot be distinguished.

For such cases, additional images are needed. Currently, reconstruction from $2D$ image points and lines without correspondences is a component of the image understanding system being developed at our computer vision lab under the RADIUS project.

In the future, we are examining ways of modifying the second algorithm so that the threshold T_d is not needed. This will involve introducing mechanisms for weighting in the matching problem and the maximum flow formulation. Furthermore, this algorithm will be extended to perform triangulation from multiple images without knowing the correspondences.

References

[1] Steven D. Blostein and Thomas S. Huang, "Quantization errors in stereo triangulation", Proc. of First International Conf. on Computer Vsion, pp. 325-334, 1987.

[2] Z. Zhang and O. D. Faugeras, "Tracking and grouping 3D line segemnts", Proc. of 3rd International Conf. on Computer Vision, pp. 577-580, Osaka, Japan, 1990.

[3] R. Kumar, "Model dependent inference of 3D information from a sequence of 2D images", Ph.D dissertation, Dept. of Computer Science, University of Massachusetts, Amherst, 1992.

[4] R. Deriche, R. Vaillant and O. D. Faugeras, "From noisy edge points to 3D reconstruction of a scene: a robust approach and its uncertainty analysis", Proc. of the Second Europeian Conf. on Computer Vision, pp. 71-79, 1992.

[5] T.-C. Chou and K. Kanatani, "Recovering 3D rigid motions without correspondences, Proc. ICCV, pp. 534-538, June 1987.

[6] R. Szeliski, "Estimating motion from sparse range data without correspondences", Second Int. Conf. on Computer Vision, pp. 207-216, December 1988.

[7] D. B. Goldgof, H. Lee and T. S. Huang, "Matching and motion estimation of three-dimensional point and line sets using eigenstructure without correspondences", Pattern Recognition, Vol.25, No. 3, pp. 271-286, 1992.

[8] J. K. Aggarwal, L. S. Davis and W. N. Martin, "Correspondence processes in dynamic scene analysis", Proc. of IEEE 69, pp. 562-572, 1981.

[9] Chia-Hoang Lee and Anupam Joshi, "Correspondence problem in image sequence analysis", Pattern Recognition, Vol. 26, No. 1, pp. 47-61, 1993.

[10] G. L. Scott and H. C. Longuet-Higgins, "An algorithm for associating the features of two patterns", Proc. Roy Soc Lond, Vol. B244, pp. 21-26, 1991.

[11] L. S. Shapiro and J. M. Brady, "Feature-based correspondence: an eigenvector approach", Image and Vision Computing, Vol. 10, No. 5, pp. 283-288, June, 1992.

[12] J. Aloimonos and I. Rigoutsos, "Determining 3-D motion of a rigid surface patch without correspondences under perspective projection", Proc. of AAAI, pp. 681-688, August 1986.

[13] A. Basu and J. Aloimonos, "A robust algorithm for determining the translation of a rigidly moving surface without correspondence for robotics applications", Proc. of IJCAI, pp. 815-818, August 1987.

[14] E. Ito and J. Aloimonos, "Is correspondence necessary for the perception pf structure from motion?", Image Understanding Workshop, pp. 921-929, April 1988.

[15] H. Lee, Z. Lin and T. S. Huang, "Finding 3-D point correspondences in motion estimation", Proc. of Eighth Int. Conf. on Pattern Recognition, pp. 303-305, 1986.

[16] H. Lee, Z. Lin and T. S. Huang, "Estimating rigid-body motion from three-dimentional data without matching point correspondences", Int. J. Imaging Systems Technol. 2, pp. 55-62, 1990.

[17] A. Pentland and S. Sclaroff, "Closed-form solutions for physically-based shape modeling and recognition", IEEE PAMI, 13(7), pp. 715-729, July 1991.

[18] A. Pentland and B. Horowitz, "Recovery of non-rigid motion and structure", IEEE PAMI, 13(7), pp. 730-742, July 1991.

[19] S. Sclaroff and A. Pentland, "A modal framework for correspondence and description", Proc. of IEEE, pp. 308-313, 1993.

[20] Y. Q. Cheng, R. Collins, A. Hanson, and E. Riseman, "Triangulation without correspondences (I)", Technical Report (to appear).

[21] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to algorithms", The MIT Press, 1990.