Using Weighted Bipartite Matching to Solve the General 3-D Rigid Motion Problem [*]

Xiaoguang Wang, Yong-Qing Cheng, Robert T. Collins,
Allen R. Hanson, Richard Weiss, and Robert Moll

Department of Computer Science
Box 34610, University of Massachusetts
Amherst, MA. 01003-4610

Abstract

This paper presents a solution to recovering the rigid transformation (rotation and translation) that brings two 3-D point sets into alignment, when the correspondences between the points are not known and there exist missing data. This is called the general 3-D rigid body motion problem (also known as absolute orientation problem). For ideal cases with no missing points, a closed-form solution based on eigenstructure decomposition is proposed to recover the correspondences of the two point sets and their motion. For general cases where missing points occur, this paper proves that the correspondence problem could be reduced to a weighted bipartite matching problem. Using a heuristic measure of point pair affinity derived from the eigenstructure decomposition, an efficient weighted bipartite matching algorithm is used to determine the correspondences. Outlier issue is explored. A fast outlier removal algorithm is combined with the weighted bipartite matching algorithm to improve the the correspondence recovery. The robustness of the algorithm to both noise and outliers is shown in simulations and experiments on data from real images.

Keywords: 3-D rigid body motion, 3-D point correspondence, absolute orientation

[*] This work was funded by the RADIUS project under ARPA/Army TEC contract number DACA76-92-C-0041.

1 Introduction

Motion estimation is an important problem in many aspects of computer vision and has been extensively studied [1, 5, 7, 8, 13, 15, 16]. The problem is to recover the 3-D motion parameters and/or the structure of the objects undergoing motion from images taken at two or more time instances. We restrict our attention to *rigid body motion*, in which the structure of the object stays the same before and after the motion, and *3-D based motion*, meaning the motion analysis is based on 3-D feature points that are obtained from low-level processing. 3-D based motion estimation is important for two reasons. First, 3-D devices such as range sensors are becoming more readily available. Second, 3-D based motion is useful in those cases where 3-D data can be reconstructed from 2-D image data. In addition, 3-D based methods can be used in solving 2-D based motion problems with certain constraints [8, 14]. In this paper, we do not consider either the derivation of the initial 3-D feature points from low-level processing or how to make use of the methods in 2-D problems.

The difficulty of motion estimation is due to a cyclic dilemma existing inherently in the problem: it would be easy to solve for the motion parameters if we knew the correspondences of the 3-D points at two time instances, and the correspondences would not be hard to obtain if the motion parameters were known; but when neither the correspondences nor the motion parameters are known, simultaneously solving for both becomes difficult. Given a set of 3-D points on a rigid body in one Cartesian coordinate system and another set of points from the same body in a rotated and translated coordinate system, to estimate the rotation and translation parameters is called a *motion estimation problem* (also known as *absolute orientation problem* [7]), to recover the correspondences of the points in the two sets is called a *correspondence problem*.

Previous motion research has been divided roughly into two approaches, depending on whether or not the correspondences are recovered. Some researchers [2, 3] tried to recover the point correspondences prior to estimating the motion parameters. They used the structure invariance property of rigid bodies to make local structural comparison between the points in the two sets. That is a combinatorial search problem, which is considerably time-consuming. The other research trend is the estimation of motion parameters without knowing correspondences. Notable results are shown in Aloimonos and coworkers' work [1, 8], Huang and his colleagues' work [5, 15] , and others [13, 16]. These researchers all developed a similar approach to motion estimation without correspondence. The approach rests on a property that the eigenstructure of a *scatter matrix* is invariant to rotation, so that the rotation can be solved linearly in a closed form. The advantage of this approach is its mathematical elegance and algorithmic efficiency. As pointed out by the authors, the philosophy behind this approach is to avoid the difficulty of the correspondence problem and to use the invariance to access the motion parameters directly.

In this paper, we adopt a different philosophy. We emphasize that correspondences reveal much information about a motion, and thus are important in motion estimation. Roughly speaking, errors in motion estimations based on feature points come from two causes: noise and outliers. *Noise* refers to the case where the same point on the rigid body appears in both sets before and after the motion, but their positions are not precise due to corruption of the spatial coordinates by noise. *Outliers* in a set are those points whose corresponding points in the other set are missing. To deal with noise, knowing correspondences is useful, since there is evidence [20] showing that more stable motion parameters can be computed when correspondences are given. To deal with outliers, correspondence analysis is even more crucial. With the knowledge of correspondence, the motion can be estimated on a point-by-point basis, whereas the correspondenceless algorithms can only work in a globally approximated way. The existence of outliers would make the global approximations very unreliable. This makes the correspondenceless algorithms work in a very narrow domain, called the *ideal motion problem*, where the point numbers of the two sets are equal and each point in a set must have a corresponding point in the other set. Correspondenceless algorithms for *general motion problems* have not been reported in the literature.

With this point of view, the methodological question follows: without knowing the motion parameters, is it possible to recover the correspondences efficiently? In this paper we propose a new method for solving the correspondence problem, and give an affirmitive answer to the question. The method is based on a discovery that the correlation matrix of the observed 3-D point set has the property of eigenstructure invariance with respect to point permutation. With this property, a closed-form solution can be obtained for ideal motion and correspondence problems. As oppposed to correspondenceless methods that recover the rotation from the scatter matrix, the immediate goal of the new method is to recover a permutation matrix representing the correspondences of the points. In general motion problems, where outliers exist, the permutation matrix recovery is replaced by a well-defined weighted bipartite matching problem, which has solid foundation in graph theory. Using weighted bipartite matching, badly-matched points can be detected and removed as outliers. As a result, the method is able to deal with general motion problems. In summary, we break up the chicken-and-egg dilemma of resolving motion and correspondences by recovering the correspondences first, from which we try to separate the "good" ones and use them to estimate the motion. This is different from the global approximation philosophy of correspondenceless methods.

The rest of the paper is organized as follows. Section 2 describes the theory and method of the closed-form solution to correspondences for ideal cases. Section 3 discusses general correspondence problems using weighted bipartite matching. Simulation results are shown in Section 4, and results from real data are given in Section 5. In Section 6 we discuss future work.

2 A Closed-Form Solution to 3-D Motion Estimation

In this paper, *motion* is defined in terms of a fixed 3-D point $\mathcal{S}_i$ represented in two $\mathbf{R}^3$ Cartesian coordinate systems $\mathcal{O}_A$ and $\mathcal{O}_B$. Suppose $\mathcal{S}_i$'s coordinates are observed to be a_i and a_i' in $\mathcal{O}_A$ and $\mathcal{O}_B$, respectively. a_i and a_i' are 3×1 vectors related by equation

$$a_i' = Ra_i + T, \tag{1}$$

in which R is a 3×3 matrix that reflects the rotation between $\mathcal{O}_A$ and $\mathcal{O}_B$, and T a 3×1 vector reflecting the translation between $\mathcal{O}_A$ and $\mathcal{O}_B$. R is an *orthogonal matrix*, satisfying

$$R^T R = I. \tag{2}$$

We are given two sets, $\mathbf{A}$ and $\mathbf{B}$, of observed points in $\mathcal{O}_A$ and $\mathcal{O}_B$: $\mathbf{A} = \{a_i\}_{i=1}^m$ and $\mathbf{B} = \{b_j\}_{j=1}^n$, but we don't know the correspondence of any particular $a_i \in \mathbf{A}$ and $b_j \in \mathbf{B}$. The motion estimation problem is to estimate the R and T from these observed points. The correspondence problem is to recover the correspondences of the points in $\mathbf{A}$ and $\mathbf{B}$. Juxtaposing the observed vectors, we get two matrices with the sizes of $3 \times m$ and $3 \times n$:

$$A = \begin{bmatrix} a_1, & ..., & a_m \end{bmatrix}, \tag{3}$$

$$B = \begin{bmatrix} b_1, & ..., & b_n \end{bmatrix}. \tag{4}$$

We call A and B the *observation matrices*. Note: the columns a_i in A and b_i in B are not corresponding points. a_i may have its corresponding points residing at b_j in B. That is, $b_j = a_i' \in \mathbf{B}$.

A general motion problem may have data in $\mathbf{A}$ and $\mathbf{B}$ with $m \neq n$, signifying non-corresponding points (due to occlusion or feature mis-detection). For convenience, in this section we first investigate the ideal case where $m = n$ and assume that, for each $a_i \in \mathbf{A}$, there always exists an $a_i' \in \mathbf{B}$ satisfying (1), and vice versa.

2.1 Centralization of Observed Vectors

A good property [1, 15, 7] of the ideal case is that the geometric centers of $\mathbf{A}$ and $\mathbf{B}$ satisfy equation (1) as well. This property enables the elimination of T's influence when we estimate R.

Formally, define

$$c_A = \frac{1}{m}\sum_{i=1}^{m} a_i, \tag{5}$$

$$c_B = \frac{1}{n}\sum_{j=1}^{n} b_j, \tag{6}$$

and

$$\alpha_i = a_i - c_A, \text{ for } i = 1, ..., m, \tag{7}$$

$$\beta_j = b_j - c_B, \text{ for } j = 1, ..., n. \tag{8}$$

It holds that, if $a_i \in \mathbf{A}$ and $a'_i \in \mathbf{B}$ satisfy (1), then

$$\alpha'_i = R\alpha_i, \text{ for } i = 1, ..., m, \tag{9}$$

in which $\alpha'_i = a'_i - c_B$ is α_i's corresponding point in $\mathbf{B}$. We call α_i and β_j *centralized vectors.* Obviously these centralized vectors maintain the same correspondences of the observed points.

In a manner parallel to the observation matrices, we define the *centralized observation matrices* by

$$A_C = \begin{bmatrix} \alpha_1, & ..., & \alpha_m \end{bmatrix}, \tag{10}$$

$$B_C = \begin{bmatrix} \beta_1, & ..., & \beta_n \end{bmatrix}. \tag{11}$$

2.2 Permutation Matrix

Now we introduce a *permutation matrix*, P, as a tool for solving the correspondence problem. In this paper, permutation matrices are defined as those that swap columns when they are multiplied with a matrix, Z, from the right hand side. That is, the matrix ZP is just a re-ordering of the columns of Z. A *strict form* of permutation matrices is an identity matrix with its rows re-ordered [6], e.g.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{12}$$

It is easy to see that, given a particular matrix Z as well as its column re-ordered version Z_P, there always exists a permutation matrix P, in strict form, such that $Z_P = ZP$. With this property and equation (9), the following claim holds: in the ideal motion problem, there always exists an $m \times m$ permutation matrix P such that

$$B_C = RA_C P; \tag{13}$$

particularly, there is such a P in strict form. Equation (13) shows that the correspondence problem of the centralized observation matrices can be described by a permutation matrix.

The existence of permutation matrix does not guarantee the uniqueness. In fact, given a particular matrix Z, there may exist an infinite number of matrices that swap the columns of Z when they are multiplied from the right hand side. Furthermore, these matrices may or may not be in the strict form as in (12). Throughout this paper, we call them "permutation matrices in *general form*". Similarly, a permutation matrix may or may not be an orthogonal matrix; however, permutation matrices in strict form must be orthogonal, i.e.

$$P^T P = I. \tag{14}$$

2.3 Eigenstructure Decomposition

Our correspondence-based method is rooted in the observation that the correlation matrices of the centralized observation matrices are insensitive to rotations, and that the eigenstructure of the correlation matrices are invariant to permutations. *Correlation matrices* of the centralized observation matrices are defined by

$$C_A = A_C^T A_C, \tag{15}$$

$$C_B = B_C^T B_C. \tag{16}$$

From equation (13)(15)(16), we directly get

$$B_C^T B_C = P^T A_C^T R^T R A_C P = P^T A_C^T A_C P \tag{17}$$

or

$$C_B \;\; = \;\; P^T C_A P, \tag{18}$$

in which R is eliminated due to its orthogonality. When P is orthogonal (there is at least one orthogonal P as describe above), equation (18) shows a *similar transformation*[6]. It reflects the eigenstructure invariance of the correlation matrices C_A and C_B. With this discovery, we have eliminated R from the problem, and converted the motion estimation problem to the computation of the permutation matrix P. Because C_A and C_B are symmetric matrices, the computation of P can be done in a very simple way. From the eigenstructure decomposition theory, there exist orthogonal matrices U_A and U_B and a diagonal matrix D, such that

$$C_A \;\; = \;\; U_A D U_A^T, \tag{19}$$

$$C_B \;\; = \;\; U_B D U_B^T. \tag{20}$$

C_A and C_B share the same *eigenvalues* on D's diagonal. The columns of U_A and U_B are known as *eigenvectors* of C_A and C_B. It follows from (18) that

$$U_B D U_B^T = (U_A^T P)^T D (U_A^T P). \tag{21}$$

One solution of P is given by

$$U_B^T = U_A^T P. \tag{22}$$

Equation (22) holds only if the signs of the eigenvectors in U_A and U_B are consistent. We will discuss this issue later. At the moment, we assume that U_A and U_B are consistent. Immediately from (22),

$$P = (U_A^T)^{-1} U_B^T = U_A U_B^T. \tag{23}$$

This tells us that the correspondence permutation matrix can be obtained in closed form from a very simple computation. The matrices D, U_A and U_B can be calculated using the Jacobi Transformation [17].

2.4 Solving for R and T

Recall equation (13). After we have solved for P, we can get $A_C P$ by permutating the centralized observation matrix A_C. Each column of $A_C P$ is a corresponding vector to the same column in B_C. Adding back the center vectors c_B and c_A to the columns of B_C and $A_C P$, we will get the the original observation matrix B and a permutated matrix A' of A. B and A' are correponding matrices, reflecting the correspondences in **A** and **B**.

So far we solved the 3-D rigid body correspondence problem in a closed form, and the problem is converted to a rotation and translation recovery problem with correspondences known. This is a subject that has been extensively studied. Recent work on the subject is done by Wang and Jepson [20]. They proposed a novel closed-form algorithm solving for the translation and rotation, which is claimed to be very robust in the presence of noise. Although closed-form solutions are also presented for ideal motion problems by Aloimonos's group [1, 8], Huang's group [5, 15], and others [13, 16], they do not explicitly know the correspondences of the two data sets and thus are not able to take the advantages of noise resistant algorithms that have been developed. That is, from the viewpoint of noise resistance, our correspondences-recovering method is potentially advantageous.

It is worth noting here that eigenstructure decomposition methods have long been used in solving motion problems in that eigenvectors are an important structural representation of a rigid body. Equation (13) is a sufficient description of the ideal rigid body motion problem: two centralized observation matrices are related by the unknown R and P. Correspondenceless methods [1, 5, 8, 13, 15, 16] were based on the analysis of the eigenstructures of the 3×3 *scatter matrices* $A_C A_C^T$ and $B_C B_C^T$. In essence, they used the property of the orthogonality of P to solve for R:

$$B_C B_C^T = R A_C P P^T A_C^T R^T = R A_C A_C^T R. \tag{24}$$

To the contrary, our method uses the the orthogonality of R to solve for P, as seen in equation (17). It is easy to see the perfect symmetries between equation (17) and (24), the scatter matrix and the correlation matrix, and the rotation matrix and the permutation matrix. These symmetries reflected the inherent cyclic dilemma of motion problems. The traditional correspondenceless methods and the new method we present exploit different philosophies and use symmetric ways to avoid the dilemma. This is the clearer and more complete view of the ideal rigid body motion problem. Moreover, correlation matrices preserve more structural information than scatter matrices. So our method based on correlation matrices is easy to extend to more general motion problems, which is discussed Section 3.

2.5 Implementation Issues

Equations (19)-(23) are only theoretical derivations based on general symmetric matrices. A further observation of the correlation matrices reveals that each of C_A and C_B has at most three nonzero eigenvalues. This is because A_C is a $3 \times m$ matrix, containing at most three independent row vectors. The rows of $C_A = A_C^T A_C$ are merely linear combinations of the rows of A_C; consequently, it has at most three linearly independent vectors. This fact brings a number of interesting results with regard to implementations.

First, an immediate corollary is that there are only three eigenvectors, the three that correspond to the nonzero eigenvalues, that represent the eigenstructure of a correlation matrix. $(m-3)$ of the eigenvectors in U_A have nothing to do with the eigenstructure of C_A. They have only a mathematical meaning to expand the three dimensional space of the eigenstructure to $\mathbf{R}^m$ space.

Second, as we shall see, the $(m-3)$ eigenvectors with zero eigenvalues can be neglected when we try to solve for the correspondences. So the eigenstructure decomposition problem reduces to solving for the three nonzero eigenvalues and the associated eigenvectors, rather than the whole matrices U_A and U_B. It is known [6] that eigenvalues and eigenvectors can be solved for efficiently provided only a small number of them are desired.

Third, let

$$W = \mathrm{diag}(\, d_1, \quad d_2, \quad d_3\,), \tag{25}$$

$$Q_A = \begin{bmatrix} u_1, & u_2, & u_3 \end{bmatrix}, \tag{26}$$

$$Q_B = \begin{bmatrix} v_1, & v_2, & v_3 \end{bmatrix}, \tag{27}$$

in which d_1, d_2, d_3 are the three nonzero eigenvalues of the correlation matrices, u_1, u_2, u_3 the associated eigenvectors of C_A, and v_1, v_2, v_3 the associated eigenvectors of C_B. With the fact that other eigenvalues are zero, it is easy to derive a set of equations parallel to equations (19)-(23):

$$C_A = Q_A W Q_A^T, \tag{28}$$

$$C_B = Q_B W Q_B^T, \tag{29}$$

$$Q_B W Q_B^T = (Q_A^T P)^T W (Q_A^T P), \tag{30}$$

$$Q_B^T = Q_A^T P. \tag{31}$$

We call Q_A^T and Q_B^T *eigenvector matrices.* Since Q_A^T and Q_B^T are $3 \times m$ matrices, it is obvious from equation (31) that P has infinitely many solutions, one of which is

$$P = Q_A Q_B^T. \tag{32}$$

The P obtained from (32) is generally not in strict form, but it is easily provable that it is a permutation matrix that makes the column vectors in B_C and $A_C P$ correspond to each other.

Fourth, the eigenvectors' sign issue must be treated carefully when using the eigenstructure decomposition method. Only consistent signs of the eigenvectors in U_A and U_B, or in Q_A and Q_B, can derive a correct P in formulas (23) and (32). In the event that no a priori knowledge is given, usually we have to examine all the sign combination cases. The analysis above ensures that there are only eight cases to be examined when P is computed, since only three eigenvectors are needed. In other words, the decomposition of correlation matrices has no more complexity than that of scatter matrices in terms of the sign issue. (See [1, 13, 15] for details of the approach to solving the eigenvector sign problem as well as a complexity analysis.)

3 Correspondence via Weighted Bipartite Matching

The previous section ends up with a closed-form solution to motion parameter estimations. A shortcoming of this solution is that it is limited to ideal motion problems. In case that the numbers of observed points in the two sets are different ($m \neq n$), equation (30) cannot be established, and thus P cannot be determined. In addition, in general cases the two sets of observed points are obtained separately from different observing environments, such that there will usually be some points in $\mathbf{B}$ that have no corresponding points in $\mathbf{A}$, and vice versa. We cannot detect these outliers solely by using the closed-form solution (32). Traditional methods based on scatter matrix decomposition cannot do it, either.

In this section, we extend the correlation matrix decomposition method to deal with general motion problems, that is, we try to determine the corresponding point pairs from the two observed point sets while removing the non-corresponding points. The existence of non-corresponding points may make two observed point sets no longer appear as a rigid body. In particular, eigenstructure-based algorithms depend heavily on the correctness of the eigenstructure of the observed data.

Outliers can ruin the eigenstructure, especially when they severely bias the geometrical centers of the observed data sets. In this paper, we deal with the case where the outliers are not so dominant and the observed point sets still "look like" they come from a rigid body.

3.1 Affinity

The term *affinity* is defined by Ullman [19] as a measurement of the pairing likelihood of two objects in two sets. One reason why the closed-form method fails to eliminate the non-corresponding points is that it does not consider the affinity of the correpondences — although matrices B_C and $A_C P$ are column-to-column corresponding, we don't know the degree of affinity of each column pair. Our task is to find out a pairing affinity of the observed points in **A** and **B**. Pairs with high affinity values are to be considered the corresponding points; others to be outliers.

From Section 2 we know that the correspondences of **A** and **B** are represented by the correspondences of the columns in the centralized observation matrices A_C and B_C. Furthermore, equation (31) indicates that the column correspondences of A_C and B_C are equivalent to column correspondences of the eigenvector matrices Q_A^T and Q_B^T. In fact, in ideal cases where there is no noise, for each column vector in Q_B^T, there should be an identical column vector in Q_A^T. For this reason, the column vectors in Q_A^T and Q_B^T are called *feature vectors* [18].

We define the pairing affinity of the observed points as the negative of the Euclidean norms of the feature vectors' distances in their eigenvector matrices, weighted by eigenvalues. Formally, if Q_A and Q_B are defined as (26)(27) and

$$u_k = \begin{bmatrix} p_{1k}, & p_{2k}, & \dots, & p_{mk} \end{bmatrix}^T, \text{ for } k = 1, 2, 3, \tag{33}$$

$$v_k = \begin{bmatrix} q_{1k}, & q_{2k}, & \dots, & q_{nk} \end{bmatrix}^T, \text{ for } k = 1, 2, 3, \tag{34}$$

then the affinity can be represented by a $m \times n$ matrix $H[h_{ij}]$, whose elements are *likelihood values*

$$h_{ij} \;=\; -(d_1^2 \parallel p_{i1} - q_{j1} \parallel^2 + d_2^2 \parallel p_{i2} - q_{j2} \parallel^2 + d_3^2 \parallel p_{i3} - q_{j3} \parallel^2), \tag{35}$$

where $\begin{bmatrix} p_{i1}, p_{i2}, p_{i3} \end{bmatrix}^T$ and $\begin{bmatrix} q_{j1}, q_{j2}, q_{j3} \end{bmatrix}^T$ are feature vectors. The purpose of the negative is to make h_{ij} an increasing function with respect to the similarity of the two feature vectors. The multiplications by eigenvalues make the eigenvectors that correspond to bigger eigenvalues have more influence on h_{ij}. Practically, the eigenvalues of the two correlation matrices will not be exactly the same due to noise and outliers. So the formula to calculate h_{ij} should be replaced by

$$h_{ij} \;=\; -(d_{A1}d_{B1} \parallel p_{i1} - q_{j1} \parallel^2 + d_{A2}d_{B2} \parallel p_{i2} - q_{j2} \parallel^2 + d_{A3}d_{B3} \parallel p_{i3} - q_{j3} \parallel^2), \tag{36}$$

where d_{Ak} and d_{Bk}, $k = 1, 2, 3$, denote the nonzero eigenvalues of C_A and C_B, respectively.

It is worth noting that Shapiro and Brady [18] have presented an eigenstructure method similar to ours in solving correspondence problems. They defined affinity in a similar way to (36). However, the motion problem was not their main focus, and they did not give a systematic, computational method for determining the correspondences from affinity measurements.

3.2 Weighted Bipartite Matching

Weighted bipartite matching [12] is a mature mathematical tool which can be used to solve motion correspondence problems with affinity. Let $G(V_A, V_B, E)$ be an undirected complete *bipartite graph*, whose nodes are partitioned into two disjoint sets V_A and V_B. The edge set E consists of all the edges between nodes in V_A and V_B. A subset $M \subseteq E$ is said to be a *matching* if no two edges in M are incident on the same node. Given an edge-weighted bipartite graph, a *weighted bipartite matching problem* is to find a matching for which the sum of the weights of the edges is maximum.

The correspondence problem with the affinity measure defined above can be modeled by a weighted bipartite matching problem. The partitioned node sets V_A and V_B are composed of the feature vectors in the eigenstructure matrices Q_A and Q_B. The weight associated with each edge is the h_{ij} defined in affinity matrix H. A matching algorithm will find a maximum weighted matching such that the feature vectors have the most affinity to each other between Q_A and Q_B.

Weighted bipartite matching problems, also known as *assignment problems*, have long been investigated in graph theory. From a theoretical point of view, matching problems are tightly related to *network flow* theories. Particularly, weighted matching problems can be reduced to a set of flow problems called *minimum cost flow*. A classic algorithm, called the *Hungarian method*, for solving the assignment problem was given by Kuhn [9]. The time complexity of the Hungarian method is $O(m^2n)$, where $m = |V_A|$ and $n = |V_B|$. In 1980, Karp [10] gave an algorithm to solve the assignment problem in expected time $O(mn \log(\min(m,n)))$, provided that the weights of the edges from any fixed node in V_A are identically distributed random variables. We adopt Karp's method in our problem to solve for the correspondences of the feature vectors. In the problem, the weight distribution of edges from a fixed node in the bipartite graph are determined by the relationship of the eigenstructure and the spatial position of the particular point. Since we don't confine our rigid body to certain sizes and shapes, the weights of edges from a fixed node (i.e. a 3-D point) have an statistically identical distribution.

3.3 Outlier Removal

The weighted bipartite matching algorithm results in a matching that describes the point correspondences in set **A** and **B**. From (35)(36), we know that the correct correspondences dominate the matching. Consequently, a LMS (Least Median Square) method can be used to filter the matching, thus solving the correspondence problem. As the LMS method is time-consuming, here we propose a fast algorithm for outlier removal. The algorithm makes further use of the affinity heuristic.

The ultimate goal of correspondence computation is the estimation of motion parameters. To get a good estimate, attempts can be made in two directions: to get as many correct correspondences as possible, and to remove as many incorrect correspondences as possible. Usually the second direction is more important, since outliers tend to cause big errors in the estimated motion parameters. With this idea, the principle of our algorithm is to work in the second direction.

First, the matching algorithm always outputs a matching with cardinality $\min(m, n)$. If $m < n$ then there are $(n - m)$ nodes in V_B that are not matched with any nodes in V_A, because they failed in competing with other m nodes in V_B. That means these $(n - m)$ feature vectors have low affinity with the other set. We remove these vectors as outliers.

Second, for each of the edges in the matching, the associated weight shows the affinity of the two nodes. Those edges with big affinity value are considered to be true correspondences; those with small affinity value tend to be outliers. A partitioning operation is applied to the matching $M = M' + M''$, in which any edge in M' has a higher weight than any edge in M''. The set M'' is then removed as outliers. The heuristic criterion we use for partitioning M is to minimize the function

$$t(M', M'') = |s(M') - \gamma s(M'')|, \tag{37}$$

in which $s(M')$ and $s(M'')$ are the sums of the weights of the edges in M' and M'', and γ is a parameter that affects the size of M'' to be removed.

Due to errors in eigenstructures, the affinity obtained from Q_A and Q_B is erroneous. Some true correspondences may be associated with low affinity values, and thus removed as a result of the above two steps. This results from our algorithm's principle of outlier removal. We sacrifice some true correpondences for removing outliers faster. In criterion (37), γ can be set empirically. A smaller *gamma* tends to cause a bigger set of M'', removing more outliers in the cost of sacrificing more true correspondences.

The ability to remove outliers is the most distinguishing property of the eigenstructure decomposition method based on correlation matrices. Correspondenceless algorithms cannot detect outliers from the observed data because they never perform a correspondence analysis. When outliers exist, correspodenceless algorithms have a hard time determining accurate motion estimations. In the new method, the motion parameters are estimated from the "purified" set M'. Therefore, the new method has a much wider application domain.

3.4 Implementation Issues

An outline of an outlier-removing motion estimation algorithm is as follows. The input is the observation matrices A and B. The output is the motion parameters R and T.

1. Centralize A and B to get A_C and B_C.

2. Compute the correlation matrices C_A and C_B using formula (15)(16).

3. Decompose the eigenstructure of C_A and C_B using the Jacobi transformation to get eigenvalue matrix W and eigenvector matrices Q_A and Q_B.

4. Compute the affinity matrix H using formula (36).

5. Get the maximum weighted matching using Karp's algorithm.

6. Remove the low affinity correspondences from the matching.

7. Use the high affinity correspondences to compute R and T.

We now discuss some implementation issues regarding the algorithm. The first issue concerns the signs of the eigenvectors. In Section 2 we mentioned that there are eight possible combinations. Traditional methods usually compute the R's in all the eight cases (sometimes four cases with some constraints) and use each R to determine the correct case. In our method, the correct sign case can be determined without computing R. The weighted bipartite matching algorithm outputs a matching associated with the maximum sum of the affinity (weights). Obviously, if the sign setting is incorrect, the affinity of the matching will be worse than the correct case. Therefore, we can test all the eight sign cases using the matching algorithm, and choose the one with the biggest weight sum as the correct setting for computing R and T.

Second, as mentioned in Section 2, noise resistant approaches exist for solving for R and T when correspondences are known. This is applicable to Step 7 in the above algorithm. We have proved that the correspondences of the feature vectors in Q_A and Q_B precisely reflect the correspondences of the observed vectors in A and B. After the outliers are eliminated, the stable approach to computation of R and T can be applied to the subsets of $\mathbf{A}$ and $\mathbf{B}$ containing "good" correspondences.

Third, we point out that better estimation of the motion parameters can be obtained by iteratively running the algorithm. Due to noise and outliers, the centers of $\mathbf{A}$ and $\mathbf{B}$ do not reflect the centers of the well-corresponding points in $\mathbf{A}$ and $\mathbf{B}$. Errors are thus introduced from the centralization of the observed vectors. After the first round of running our algorithm, some outliers are removed, and thus we can have a better estimation of the centers, which we use to do the second round, and so on. When all outliers are removed, we can get an accurate translation and rotation estimation.

4 Simulations

Simulations are performed on a set of 3-D points representing a rigid body. 20 points are randomly generated with a uniform distribution in a $(100 \times 100 \times 100)$ volume. The points are then transformed to a rotated and translated coordinate system as the second observation data set of the rigid body. The translation vector is set to $(10, 20, 30)$. The rotation is set to $(40°, 50°, 60°)$ in Euler Angles. The points in the two sets are reordered randomly for our algorithm to recover the correspondences. The experiments are designed to test the effects of both noise and outliers on algorithm performance. Gaussian noise, with zero mean and variance σ ranging from 0 to 9, is added to each point in the two sets. To simulate outliers, we use a random dropping technique to drop off l_a points from one set and l_b points from the other. No corresponding point pair is dropped from the two sets. That is, the input sets of the algorithm are $\mathbf{A}$, with $(20 - l_a)$ points, and $\mathbf{B}$, with $(20 - l_b)$ points, and there are $(20 - l_a - l_b)$ pairs of corresponding points and l_a and l_b outliers in $\mathbf{A}$ and $\mathbf{B}$, respectively.

We propose a *hit rate*, ρ, as the evaluation for the performance of the algorithm:

$$\rho = \frac{|M_{hit}|}{|M|}, \tag{38}$$

in which M is the output matching of the algorithm, and $M_{hit} \subseteq M$ is the subset of correct correspondences in M. As mentioned in Section 3, the most important thing is to get rid of the outliers in the matching, i.e. to improve the hit rate. In the experiments, the algorithm is run for two iterations. The first iteration is to get the initial matching M_0 of **A** and **B**. The second iteration is to get a new matching M_1 out of M_0', which is the purified matching using criterion (37). We set $\gamma = 1$ in the criterion. Both of the hit rates of M_0 and M_1 are calculated for evaluation. For each configuration of (σ, l_a, l_b), the algorithm is performed one thousand times, from which we get $\bar{\rho}_0$ and $\bar{\rho}_1$ (the average hit rates of M_0 and M_1), and sd_0 and sd_1 (their standard deviations).

Our first experiment is on perfectly corresponding sets: there are no missing points in **A** and **B**. The curves of average hit rates as a function of noise variance are shown in Figure 1. The second experiment is to test the algorithm's performance on outliers without noise. We keep set **B** with the 20 points (i.e. $l_b = 0$) and drop points from **A**. So there are l_a points in **B** missing their correspondences in **A**. Results are shown in Figure 2. The third experiment is on testing noisy data with outliers. The hit rate response curves for this condition are shown in Figures 3 and 4 for the two particular cases: 1) $l_a = l_b = 2$, and 2) $l_a = 4, l_b = 0$.

Some conclusions can be drawn from the simulation results. Firstly, the algorithm performs well in the presence of noise. For ideal motion problems where there are no ouliers (Figure 1), the correspondences are perfectly recovered when the noise is small. When noise increases, the stability goes down only slowly. For general cases (Figures 3 and 4), the average hit rate curves show the same flatness. Secondly, when outliers exist, the iteration of low affinity correspondence removal makes a significant improvement in performance, with or without noise. From Figures 2, 3 and 4 we can see that the average hit rate rises about 10 percentage points consistently at any number of outliers and any noise levels. Thirdly, when the number of outliers is small, i.e. the motion keeps good rigidity, the algorithm can recover the correspondences that are almost all correct. Fourthly, when the number of outliers increases, the hit rate drops. However, averagely speaking, correct correspondences account for the majority in the resulting matchings in a wide range of input conditions. In Figure 2, when nearly half of **B** miss their corresponding points in **A**, the algorithm still results in matchings with about 50% correct correspondences on average. In those cases, the LMS method could be applied to the resulting matchings for more refinement.

5 Experiments on Real Images

The 3-D correspondence algorithm is applied to two sets of real images known as the BOX sequences and ROOM sequences. The image sequences were captured with a SONY B/W AVC-D1 camera, with an effective field of view of approximately $(23^\circ \times 23^\circ)$ and $(40^\circ \times 40^\circ)$ for the BOX and ROOM sequences respectively [11]. The intrinsic parameters of the camera were calculated from the manufacturer's specification sheets. The images were digitized to 256×256 pixels. Within each sequence, the relative poses between frames were recorded when capturing the pictures.

There are two sources of 3-D point data in the experiments. One source, refered to as *model*, are points on the rigid body (box or objects in the room) that were measured in advance with respect to a world coordinate system. The other source is *triangulated data*, which are acquired by a triangulation procedure using corresponding 2-D data extracted from images in the sequence, together with the relative pose information between the frames in the sequence [4].

Both model and triangulated data are noisy. The noise in the model is due to the error in measurement and was estimated during measurement. The noise in the triangulated data is due to many factors such as camera distortion and errors in low-level image processing (feature point extraction). It is well-known that these 2-D errors have a big effect on the triangulated 3-D data, especially when the frames for triangulation are very close to each other. To estimate the noise between two 3-D point sets of a rigid body, we transform one set of points to the other's coordinate system using the relative R and T, and then calculate the deviation (DEV) of each corresponding point. The average DEV and the biggest DEV are of most interest. It is worth noting that the DEV's in our experiment are only used to show the reliability of the algorithm. They are typically not available beforehand in 3-D correspondence problems, because the R and T and the correspondences of the points are not known. DEV is only an absolute value. To show the effects of the noise, we must also consider how far the points in a set are apart from each other. The *Distance to the Nearest Point* (DNP) of each point is used for this consideration. The average DNP and the minimal DNP in a set are of most interest.

5.1 BOX sequences

The experiment performed on the BOX sequences addressed this problem: given two independent image sequences of a rigid body, with only relative poses known within each sequence, can we recover the motion between the two sequences? Suppose we have two viewers of the same object, each having an image sequence of the object captured independently. Can we compute the positional relationship between these two viewers from their image sequences?

We have two sequences of the same box. 30 points were picked for the experiment. Their depths to the camera ranged from 550mm to 700mm in both sequences. We only know the relative poses between frames within each sequence. With this knowledge, 3-D points are triangulated in each sequence independently and represented with respect to the first frame's coordinate system of each sequence. So the problem is reduced to a 3-D correspondence problem for these two sets. Figure 5 shows the 30 points with one of the images in the sequences. The average DNP of the 30 points was about 27mm and the minimal DNP was 15mm. The average DEV between the two triangulated sets was 1.1mm and the maximal DEV was 1.5mm. Because all the points appear in both sets, it is an ideal correspondence problem with fairly low-level noise. The eigenvalues of the correlation matrices of the two triangulated 3-D point sets are (103859, 51418, 17248) and (103587, 51639, 17123). Our algorithm perfectly recovered all the correspondences in one iteration, and thus solved the problem of the motion between the two independent sequences. The actual and recovered rotation (in Euler Angles) and translation parameters between the first frames of the two sequences are shown in Table 1.

Table 1: The actual and recovered motion parameters between first frames of the two box sequences

	Actual Parameters	Recovered Parameters
θ	$-1.59°$	$-1.22°$
ϕ	$-21.45°$	$-21.17°$
ψ	$-13.45°$	$-13.50°$
T_x	-240.7 mm	-241.1 mm
T_y	-12.9 mm	-9.4 mm
T_z	55.6 mm	54.6 mm

5.2 The ROOM sequence

The ROOM sequence was generated by fixing a camera to a moving PUMA arm. The location of 30 points (marked in Figure 6) in the world coordinate system was measured to an accuracy of approximately 0.2 feet along each axis. The depth of these model points varied from 13 feet to 33 feet in the sequence.

Model matching is the experiment we designed for the ROOM sequence. From the image sequence we triangulate the points that appear in the model and try to match the triangulated data with the model. Once the correspondences are correctly recovered, the transformation from camera coordinate system to the world coordinate system is easily solved. However, as seen in Figure 5, some points were not captured in the image sequence due to the limited view of the camera. Only 26 points were successfully triangulated with respect to the camera coordinate system in the sequence. So it is a general 3-D point correspondence problem. Moreover, the triangulated data turned out to be very noisy in the ROOM sequence. The average and minimal DNP's of the 30-point model are 1.46 feet and 0.65 feet respectively, but the average and maximal DEV of the triangulated data from the model are 0.30 feet and 1.25 feet. That is, the maximal error in the triangulated data are even bigger than the distance of a pair of points in the model. The eigenvalues of the correlation matrices of the model and the triangulated data are (1351, 322, 77) and (1084, 254, 63).

The algorithm starts with the 30 and 26 points in the data sets, and gets an initial matching of 26 points in both sets, among which there are 16 correct corresondences. As mentioned above, the hit rate is the most important thing in the matching analysis. Although the LMS method is applicable to this initial matching, we run the fast outlier removal algorithm iteratively, with $\gamma = 1$, for better hit rates. The algorithm ends up with a fully correct matching after 7 iterations. The variation of the hit rates in these iterations are shown in Figure 8. At the last iteration, the cardinality of the matching is 11. That is, we find a correct matching of 11 points in both sets without the help of the LMS method. Using this correct matching, the motion between the camera coordinate system and the world coordinate system is recovered. Table 2 shows the actual and recovered rotation (in Euler Angles) and translation parameters between the two coordinate systems.

Table 2: The actual and recovered motion parameters between the model and the triangulated data in the room sequence

	Actual Parameters	Recovered Parameters
θ	$-163.37°$	$-163.52°$
ϕ	$17.40°$	$17.73°$
ψ	$-131.80°$	$-131.74°$
T_x	0.05 ft	0.11 ft
T_y	-3.11 ft	-3.07 ft
T_z	32.57 ft	32.57 ft

6 Conclusions and Discussions

In this paper we first reveal a theoretical symmetry in 3-D motion problems: the symmetry between the rotation matrix R and the permutation matrix P. Unlike the correspondenceless philosophy in 3-D point motion estimation, we emphasize instead the importance of the correspondences in outlier elimination and solution stability. A closed-form method based on correlation matrix eigenstructure decomposition is proposed, parallel to the traditional scatter matrix eigenstructure decomposition. In this method, correspondences are determined before estimating motion parameters. Using a heuristic measure of point pair affinity computed from the eigenstructure, a weighted bipartite matching algorithm is adopted to determine the correspondences. We also emphasize the importance of hit rate in matchings, because a high hit rate ensures that LMS methods can work out an outlier-free matching. By using the affinity heuristic again, a fast outlier removal algorithm is proposed to improve the hit rate in bipartite matchings.

Simulations and experiments on real images were done to test the proposed algorithms. The experiments show that the algorithms are noise resistant, and are feasible in general motion problems where outliers exist. Thus our method potentially has a wider application domain than traditional eigenstructure-based methods. Also, it is easy to see that the method is not confined to 3-D rigid body motion problems. It could be extended to 2-D motion and even deformable body motion problems.

Some remaining issues are subject to future study. One of them is the criteria for purifying the matchings in the outlier removal algorithm. Currently we have no theoretical proof or direct evidence that the criterion in (37) is the best. The criterion to remove low affinity correspondences determines both the quality and the speed of outlier removal, and hence is an important problem. A related issue is the stopping criterion for the algorithm's iterations. It is seen from the simulations that the outlier removal iterations sometimes increase the standard deviation of the hit rate, although they consistently improve the average hit rate. This indicates that the the algorithm iterations sometimes lead to a local minimum. How to improve these liabilities of the algorithm is under current study.

Further issues to consider include studying the limitations of the eigenstructures-based methods. High speed is one of the advantages of eigenstructure methods. However, when there are too many outliers in the point sets, the eigenstructure is totally destroyed and such methods are problematic. The method proposed in this paper has a wider application domain because it has the ability to deal with ill-eigenstructured data by iteratively improving the eigenstructure of the two observed data sets. An interesting research direction for many applications is to pre-process the observed 3-D data so that they have similar eigenstructures, then input the pre-processed data to our algorithm to recover the correspondences and motion parameters. The combination of clever pre-processing with the algorithms proposed in this paper might lead to methods that solve even more general motion problems.

References

[1] J. (Y.) Aloimonos and I. Rigoutsos, "Determining the 3-D motion of a rigid surface patch without correspondence under perspective projection," *Proc. AAAI*, pp. 681-688, August 1986.

[2] H. H. Chen and T. S. Huang, "Maximal matching of two three-dimensional point sets," *Proc. Eight Int. Conf. on Pattern Recognition*, pp. 1048-1050, October 1986.

[3] J.-C. Cheng and H.-S. Don, "A structural approach to finding the point correspondences between two frames," *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 1810-1815, April 1988.

[4] Y. Cheng, R. Collins, A. Hanson and E. Riseman, "Triangulation without correspondences," *Proc. Arpa Image Understanding Workshop*, pp. 993-1000, Monterey, CA, 1994.

[5] 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.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations (second edition)*, The Johns Hopkins University Press, 1989.

[7] B. K. P. Horn, *Robot Vision*, The MIT Press, 1986.

[8] E. Ito and J. (Y.) Aloimonos, "Is correspondence necessary for the perception of structure from motion?" *Image Understanding Workshop*, pp. 921-929, 1988.

[9] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist. Quart.*, vol. 2, pp. 83-97, 1955.

[10] R. M. Karp, "An algorithm to solve the $m \times n$ assignment problem in expected time $O(mn \log n)$," *Networks*, vol. 10, pp. 143-152, 1980.

[11] R. Kumar, *Model Dependent Inference of 3D Information from a Sequence of 2D Images*, Ph.D. Dissertation, Dept. of Computer Science, Univ. of Massachusetts at Amherst, February 1992.

[12] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.

[13] C.-H. Lee and A. Joshi, "Correspondence problem in image sequence analysis," *Pattern Recognition*, vol. 26, pp. 47-61, 1993.

[14] C.-T. D. Lin, D. B. Goldgof and W.-C. Huang, "Motion estimation from scaled orthographic projections without correspondences," *Image and Vision Computing*, vol. 12, no. 2, pp. 95-108, March 1994.

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

[16] S.-C. Pei and L. G. Liou, "Using moments to acquire the motion parameters of a deformable object without correspondences," *Image and Vision Computing*, vol. 12, no. 8, pp. 475-485, October 1994.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C, The Art of Scientific Computing (Second Edition)*, Cambridge University Press, 1992.

[18] 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.

[19] S. Ullman, *The interpretation of visual motion*, The MIT Press, 1979.

[20] Z. Wang and A. Jepson, "A new closed-form solution for absolute orientation," *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 129-134, 1994.

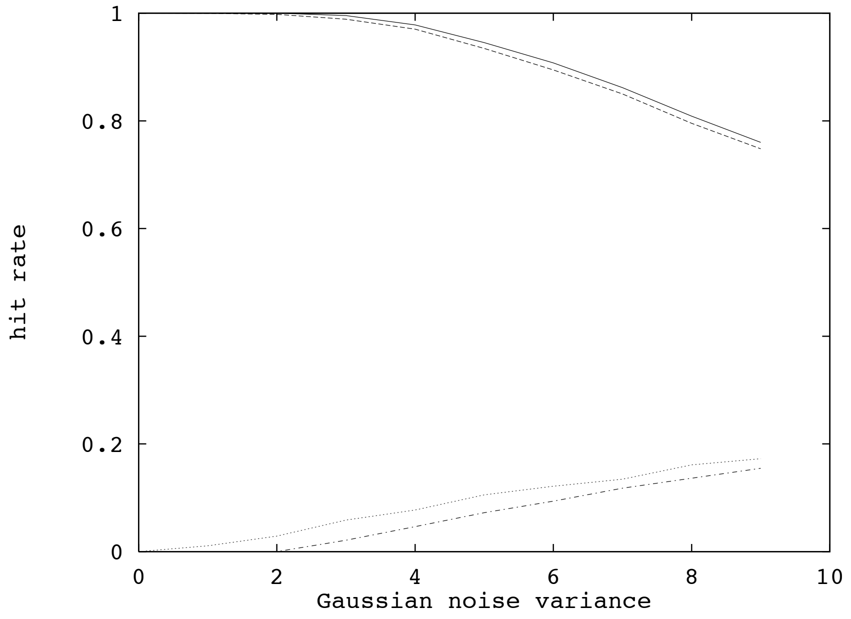

Figure 1: Performance against noise without outliers: $l_a = l_b = 0$ (solid line: $\bar{\rho}_0$; dash line: $\bar{\rho}_1$; dot-dash line: sd_0; dot line: sd_1)

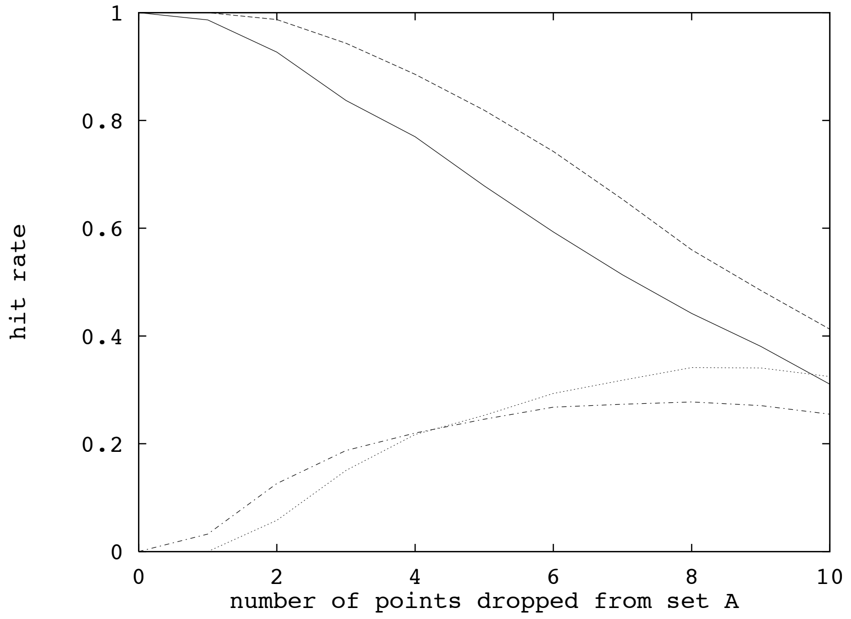

Figure 2: Performance against outliers without noise: $\sigma = 0, l_b = 0$ (solid line: $\bar{\rho}_0$; dash line: $\bar{\rho}_1$; dot-dash line: sd_0; dot line: sd_1)

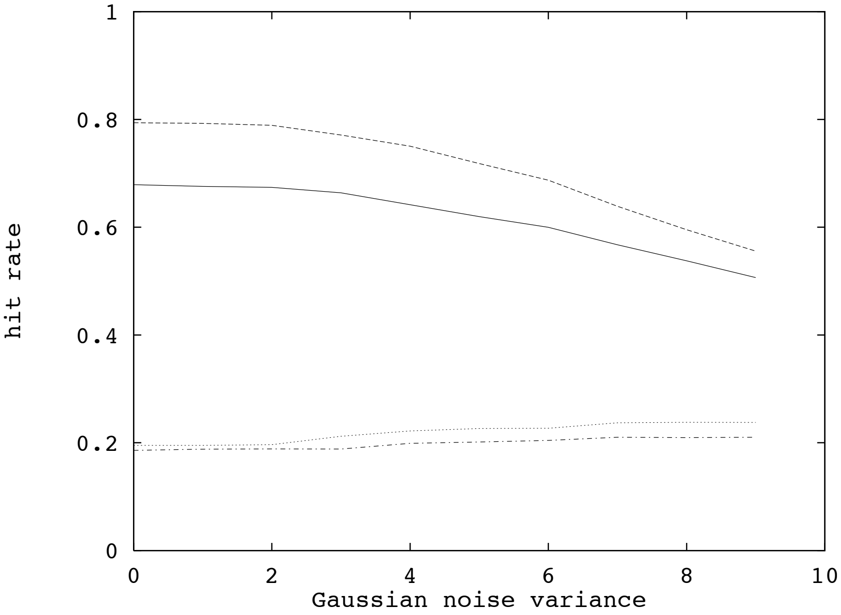

Figure 3: Performance against noise with outliers: $l_a = l_b = 2$ (solid line: $\bar{\rho}_0$; dash line: $\bar{\rho}_1$; dot-dash line: sd_0; dot line: sd_1)

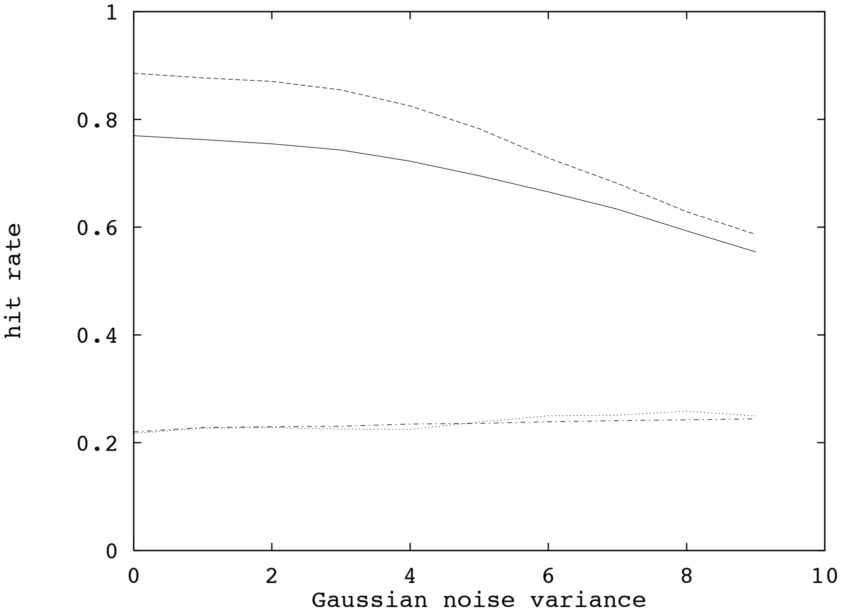

Figure 4: Performance against noise with outliers: $l_a = 4, l_b = 0$ (solid line: $\bar{\rho}_0$; dash line: $\bar{\rho}_1$; dot-dash line: sd_0; dot line: sd_1)

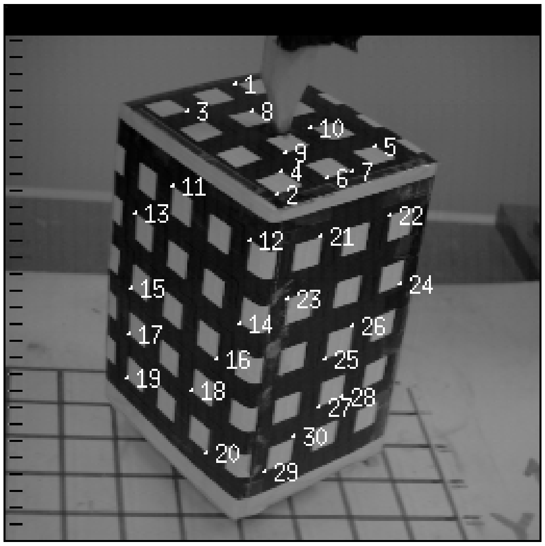

Figure 5: Triangulated 3-D points with respect to a local camera coordinate in the box sequence

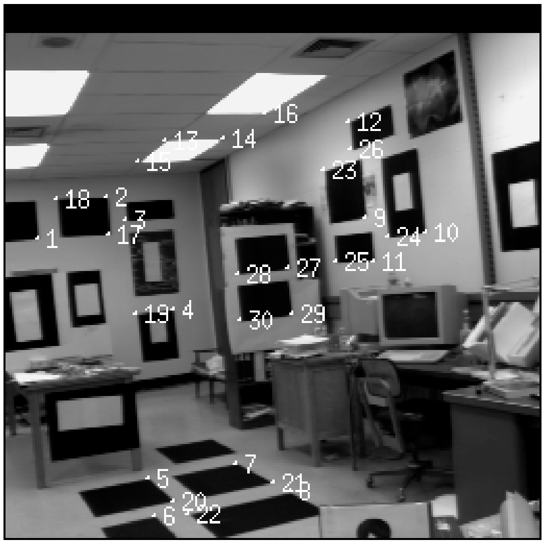

Figure 6: A measured 3-D model displayed with one of the image in the room sequence

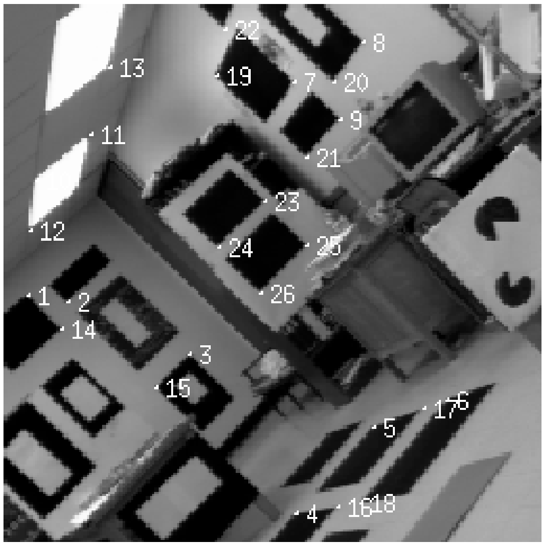

Figure 7: Triangulated 3-D points with respect to a local camera coordinate in the room sequence

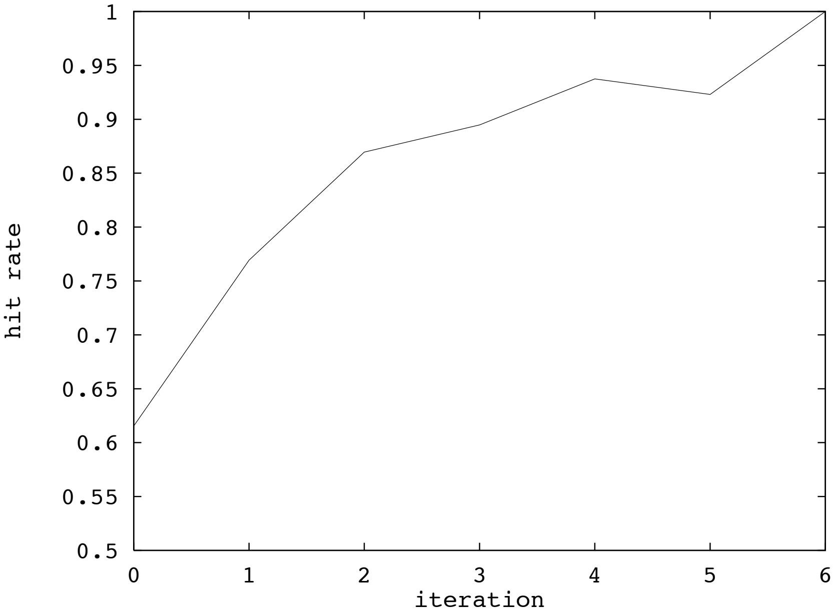

Figure 8: The hit rate in each iteration of the outlier removal algorithm