Random Sampling for Pose Determination and Refinement

Gerhard Roth

Laboratory for Intelligent Systems
National Research Council of Canada
Ottawa, Canada

Martin D. Levine

McGill Research Center for
Intelligent Machines
McGill University, Montreal, Canada

Abstract

Pose determination is the process of finding the pose (position and orientation) of a part with a known geometry in a scene using sensor data. No prior estimate of the pose is assumed to be available. If an estimate of the pose is available then pose determination becomes pose refinement. It is shown that both can be modelled as optimizations of a cost function, which in the worst case has many local minima. This optimization model makes it clear that the most important issue is how to efficiently find the global minimum of the cost function, since this will produce the best pose. The random sampling concept states that a random subset of correspondences between model and sensor data can be used to hypothesize a pose, and that if enough random subsets are chosen then the chance of finding a good pose is high. This principle is used to produce a general algorithm for pose determination and refinement which is capable of dealing with many different kinds of part models. Elimination theory makes it possible to create a hypothetical pose from a subset of correspondences whenever the model primitives are described by a polynomial equation.

1 Introduction

To estimate the pose of an object is to use sensor data to compute the position and orientation of that object with respect to a known reference frame. The pose is information which is essential for manipulation and planning purposes in robot vision. There have been many approaches to pose determination. It will be shown that finding the pose is equivalent to solving a particular kind of optimization problem, and that all the previous approaches can be modelled in this fashion. This optimization model highlights the fact that the most important issue is how to quickly find a good pose, which means finding the global minimum of a particular cost function. The model is also applied to the pose refinement problem, which is simply a restricted version of pose determination.

Random sampling is a process in which a randomly chosen subset of a set is used to represent the characteristics of that set. For pose determination this means selecting a subset of all possible correspondences between model and data primitives and using this subset to produce a hypothetical pose. Random sampling has been previously applied to certain pose determination problems [10, 1], but its generality and relation to optimization have not been noted or understood. It will be shown that any pose determination problem can be solved with random sampling, as long as the ability to compute a hypothetical pose from a minimal subset of correspondences exists. That this ability exists is demonstrated for a wide variety of geometric primitives. The resulting algorithm has a simplicity and generality which has not heretofore been obtained in a single approach for solving the pose determination problem. Some concrete examples are used to illustrate the concepts.

2 Previous Approaches

There are many previous approaches to pose determination and refinement. One of the earliest techniques for pose determination uses pose clustering [27, 28]. Each possible pairing of image and model features votes in a parameter space of transformations. The best cluster in this parameter space is the computed pose. This approach is closely related to the methods based on the Hough Transform [30, 26].

A completely different approach is based on the use of interpretation trees [12]. Here a tree of assignments of image to model primitives is pruned using simple rules based on local geometric properties. This process continues until the largest consistent set of assignments is found, and this set is used to compute the pose. The difficulty with this approach is that the execution time is an exponential function of the length of the assignment tree.

One way to have an execution time which is not exponential in the number of features is the hypothesize and test paradigm. This is the basis of a number of approaches to pose determination [2, 1], of which the most recent is the alignment method [17]. The idea is to use a small number of assignments of model to image features to hypothesize a pose, and then to verify whether this pose is correct. This is also the same principle used in geometric hashing in which the verification process is made faster by a hash table which is built off-line. This hash table is used online to find all the model primitives which match a hypothesized pose, from which the best pose is then determined [31].

In pose determination no estimate of the pose is available beforehand, while in pose refinement the assumption is made that such an estimate already exists, and that this estimate is a good one. This initial estimate is used to select some subset of correspondences between model and image features, and once made, these correspondences are assumed to be correct. Given such correspondences, pose refinement has been shown to be an optimization problem, and nonlinear optimization algorithms have been applied to obtain the solutions [18]. The assumption is that the initial pose estimate is close enough to the final estimate to guarantee convergence of the optimization algorithm. There has been some work in pose refinement using robust statistics [15].

The use of robust methods in the optimization algorithm makes it possible to ignore some percentage of incorrect correspondences between model and image features. We will discuss the differences between pose determination and refinement in more detail in the next section. Random sampling can be applied to both problems, since both can be described as an optimization process.

3 Optimization Model

There are so many pose determination and refinement algorithms that it seems impossible to find any commonality between them. In this section we will show that all such algorithms can be modelled in a simple way. This model shows that the process of finding the best pose is equivalent to finding the global minimum of a cost function. This cost function measures the conformity of the model to the sensor data, and the best pose maximizes this conformity.

To explain these concepts more clearly, a number of formal definitions are necessary. The assumption is made that the raw sensor data consist of an unordered set of points in two-or three-dimensional Euclidian space. Such sensor data are called geometric sensor data [3], and can be obtained directly from active sensors such as laser rangefinders [22], or by processing passive sensor data using methods such as stereo vision. A geometric primitive is a curve or surface that can be described by a number of free parameters. The object whose pose is to be determined is modelled as a union of k such geometric primitives, labelled $m_1, m_2, \ldots m_k$, and it is assumed to be a rigid object. An individual geometric primitive can be whatever is appropriate to the task at hand. Examples of geometric primitives are points, lines, planes, or more complex shapes such as cylinders or cones. For now it is assumed that all the primitives are of the same type, and later in the paper we will discuss how this restriction can be lifted. Similarly, a set of l geometric primitives labelled $s_1, s_2, \ldots s_l$ are extracted from the sensor data. The type of all the sensor primitives must be the same, but the type and number need not be the same as the model primitives. However, the sensor primitives must be no more complex (have the same number or fewer free parameters) than the model primitives. This means for example, that the model primitives could be lines, and the sensor primitives points, but not vice-versa. Only a subset of the sensor primitives need belong to the object whose pose is to be determined.

The pose is defined by a transformation T, which when applied to the sensor primitives, should align them with the model primitives. The type of transformation depends on the particular geometric primitives, on the dimension of the geometric sensor data and on the task at hand. When applied to the sensor primitives, the transformation T produces a set of transformed primitives labelled $s'_1, s'_2, \ldots, s'_l$ where $s'_j = T(s_j)$. Each transformed sensor primitive s'_j can be placed in correspondence with one of the members of a set q_j of model primitives. For pose determination, the set consists of all the model primitives, while for pose refinement it is usually assumed that the set consists of a single model primitive. This is because a prior pose estimate of T is used to associate each transformed sensor primitive with a single model primitive. Therefore the cardinality of this set ($||q_j||$) is at least one, and at most k. For every such correspondence there must be some way of measuring the conformity of a sensor to a model primitive. This is done by a distance function d, which measures the similarity between two primitives. Similarity in this case is not necessarily the Euclidian distance but can be any appropriate measure. Assume the transformed sensor primitive s'_j is placed in correspondence with model primitive m_i. Then $d(m_i, s'_j)$ is a function which returns a scalar, and this scalar indicates the degree of match between the transformed sensor primitive and the model primitive. The particular form of d is not important, but it must return zero when the two primitives are identical. It is clear that d is an indirect function of T, the transformation between the model and the sensor data, since $s'_j = T(s_j)$.

The final component of our model is the cost function h, which is defined as $h(d_1, d_2, \ldots, d_r)$ where r is the total number of possible correspondences. From the preceding definition it is clear that $r = \sum_{j=1}^{l} ||q_j||$ since each of the l primitives can correspond to any model primitive in the associated correspondence set. The cost function h returns a scalar which is the degree of match between the entire model and the sensor data. As the transformation T changes, the value of h changes since it is a function of the distance measurements, which are themselves a function of T. The objective is to find the value of T which produces the minimum h since this is the pose which maximizes the match between the model and sensor data. This model is quite general, and we will show that by using different h and d functions it is possible to account for all known pose determination and pose refinement algorithms.

The first use of this model is to understand the difference between pose determination and refinement. For pose refinement a prior pose estimate of T is used to associate each transformed sensor primitive with a single model primitive. This is done by applying the initial estimate of T to each sensor primitive and then making a correspondence between this primitive and the closest model primitive. In this case the total number of correspondences r is equal to l, since $||q_j|| = 1$ for $j = 1, \ldots, l$. If each member of the set of initial correspondences is correct, a nonlinear optimization algorithm can be used to find the best pose T [18]. In our model this means that the h function has only $O(1)$ (a fixed number) of local minima. Each local minimum indicates a different solution to the underlying system of nonlinear equations that are used in the optimization process [9]. The number of such local minima is small, since the total number of solutions is limited. Thus if the initial correspondences are correct, pose refinement is equivalent to minimizing a cost function which has only a small number of local minima.

If no prior estimate of the pose is available, then it must be determined which model primitive corresponds to each sensor primitive. With no prior estimate, each transformed sensor primitive can be put in correspondence which every model primitive. Thus r is equal to lk since $||q_j|| = k$ for $j = 1, \ldots, l$. Note that only a small subset of the initial correspondences is correct since each sensor primitive can only correspond to a single model primitive. Somehow the h function must select the correct correspondence for each sensor primitive and ignore the rest. (We will use the robust statistics term "inlier", for a correct correspondence, and "outlier" for an incorrect correspondence [13].) It can be shown that this ability implies that h has many local minima, since each local minimum implicitly encodes a partition of correspondences into inliers and outliers, and there are many such partitions. The best of all the local minima is the global minimum, and this is the one which produces the best pose transformation T. Since it cannot be assumed that all the initial correspondences are correct, pose determination is equivalent to the optimization of a cost function which may have many local minima.

A simple example will clarify these concepts. Consider the case where the model and sensor primitives consist of a finite set of points in the plane. Define $d(m_i, s'_j) = |m_i - s'_j|^2$ so that d is simply the square of the distance between the model and sensor point. The transformed sensor point s'_j is defined as $s'_j = Rs_j + M$ where R is a 2D rotation matrix, M is a 2D translation vector and s_j is the initial sensor point. The simplest possible h function would sum all the d values; that is $h(T) = \sum_{j=1}^{l} d(m_i, s'_j)$ where model point m_i has been placed in correspondence with sensor point s'_j. This h function assumes that every one of the l sensor points is placed in correct correspondence with a single model point. This is the classical pose refinement algorithm, since to correctly make such a correspondence there must be some prior estimate of the pose which is reasonably accurate. The global minimum can be found by a standard nonlinear optimization algorithm, and this is clear if the definition for d is inserted directly into the h function.

A single incorrect correspondence can bias the pose arbitrarily if a non-robust optimization method is used to compute the pose transformation T. An improvement would be to limit the effect of an incorrect correspondence by modifying the distance function d. If the value of this function did not increase as the distance between the transformed sensor and model primitive increased, then the influence of a single bad correspondence would be limited. This idea of limiting the influence of a single outlier is appropriately called the influence function approach in the field of robust statistics [13]. In this case, $h(T) = \sum_{j=1}^{l} \rho(d(m_i, s'_j))$ where ρ is the influence function, and this approach has been used in the pose refinement literature [15]. This modification makes the pose refinement more robust since the h function is now capable of discarding a certain percentage of bad correspondences. However, as noted previously, this means that the h function has potentially many more local minima since this number increases dramatically as a function of the percentage of bad correspondences. For pose refinement, an initial estimate of the pose exists, so the total number of bad correspondences is a much smaller fraction of all the correspondences than for a pose determination algorithm, and therefore the number of local minima in h is accordingly smaller. According to the robust statistics literature, the percentage of bad correspondences that can be tolerated is 50% of the total. We will discuss this limitation later in the paper, and will show that in a practical sense it does not exist.

The clustering and Hough transform approaches to pose determination can be shown to be equivalent to template matching, where a template is a band of fixed size around the model. If the template is matched with the transformed sensor data, the result is the number of sensor points within the model template. The template size is defined by the cell (or cluster) size in parameter space, while the template shape is defined by the part model and the hypothesized pose T. For the 2D point example, a good template would be a circle of radius r around each model point, and the template match would count the number of sensor points which under the transformation T are within a distance r of a model point. The appropriate h function would be $h(T) = \sum_{i=1}^{k} \sum_{j=1}^{l} c(m_i, s'_j)$ where $c(m_i, s'_j)$ is zero if $d(m_i, s'_j) < r$ and is one otherwise. Note that the input to the h function consists of each of the kl possible correspondences, so it is clear that while the previous example dealt with pose refinement, this example deals with pose determination.

Primitive Extraction and Fitting
Input = Points
Goal = Find Optimum Parameter Vector A
Minimize $h(A) = h(e_1, \ldots, e_r)$
where e_i is Residual Error for point i
Inlier = point on primitive
Outlier = point not on primitive
Pose Determination and Refinement
Input = Correspondences
Goal = Find Optimum Pose T
Minimize $h(T) = h(d_1, \ldots, d_r)$
where d_{ij} is Distance Measure for correspondence ij
Inlier = correct correspondence
Outlier = incorrect correspondence

Table 1: Relation between Primitive Extraction and Pose Determination

In a companion paper [24] we described a similar optimization model for primitive extraction, which is the problem of finding geometric primitives in sensor data. This ability is assumed to exist in this paper, since the sensor primitives $s_1, \ldots, s_l$ are assumed to have been extracted from the raw sensor data. The link between the two approaches is shown in Table 1.

This table makes an analogy between concepts from both papers. In both cases the ability to tolerate outliers means that the associated h function may have a large number of local minima. This ability is a necessity for both pose determination and in practice, it is also important for pose refinement since a single bad correspondence can arbitrarily bias the computed pose.

4 Random Sampling

The optimization model shows that in some sense pose determination and refinement is mathematically a solved problem. Thus the main issue is how to efficiently find a good solution. This means determining the global minimum of an h function which may have many local minima. There is a number of known methods for solving such global optimization problems [29]. The difficulty with these methods is their computational cost; they are very slow and are impractical for robot vision without massively parallel hardware such as a connection machine [14]. The random sampling concept is based on the observation that it is often possible to compute the characteristics of an entire set from a randomly chosen subset. For pose determination this means that computing the pose from a subset of correspondences often produces a pose nearly as good as the one computed using all the correspondences. A key question is what is the smallest possible subset that can be used for this purpose. This is equivalent to asking what is the minimal number (which we label P) of correspondences between model and sensor primitives that produces a unique pose transformation T. The answer depends on the particular geometric primitives that have been chosen. For example, if both sensor and model primitives are 2D points, then two correspondences are necessary and sufficient to define a unique T.

The relationship of P to different types of model primitives will be discussed in detail later in the paper.

In computer vision applications random sampling can be used to produce an hypothesis in the hypothesize and test paradigm [1]. In our case, the hypothesis is a particular pose T, and a random subset of correspondences is used to generate this value of T. The h function tests this hypothesis by measuring the goodness of the pose T. Therefore, this optimization model completely encompasses the hypothesize and test paradigm. The idea of using a minimal random subset of correspondences to hypothesize a pose was first introduced in a number of papers describing random sampling [10, 5], but it was not generalized to deal with a wide variety of geometric primitives. The random sampling concept is much more general than has been previously realized. The following pseudo-code shows how random sampling can be applied to any pose determination or refinement problem. The input is the model defined by model primitives $m_1, \ldots, m_k$, the sensor primitives $s_1, \ldots, s_l$, and the correspondence subsets $q_1, \ldots, q_l$:

For Z randomly chosen sets of P correspondences from the correspondence subsets

DO

1. *Compute the pose T that aligns the P model and sensor primitives.*
2. *Compute the distances $d_1, \ldots, d_r$ between model and sensor primitive of every correspondence with respect to pose T.*
3. *Rank the goodness of the pose T by evaluating the cost function $h(d_1, \ldots, d_r)$.*
4. *Save the pose T with the smallest cost as the best pose.*

ENDDO

The output consists of the best pose T and the associated value of the cost function h. This information, along with the initial definition of the h function, enables each of the possible correspondences in the correspondence sets to be marked as an inlier or an outlier.

4.1 Computing a Hypothetical Pose

We have previously discussed step 2 above, how to compute the distance measure between primitives. In this section, we will discuss step 1, how to compute the hypothesized pose. First consider how to choose Z, the number of sets of random samples. It is assumed that each of the P correspondences uses a different sensor primitive. Since there are at most k model primitives that could correspond to a single sensor primitive, the number of possible sets of random samples of size P is k^P. In practice this number is too large to use as a value for Z for any reasonably sized k; however, a much smaller value of Z can be used. Let the probability of a single correspondence of a sensor primitive to a model primitive in the correspondence subset being correct be ϵ. Then the probability of all P correspondences in a single one of the Z draws being correct is ϵ^P. Therefore the probability of at least one of the Z draws producing a correct set of P correspondences is $1-(1-\epsilon^P)^Z$. This formula is a simple application of combinatorial analysis and the same result is presented in [10]. Using this formula the value of Z can be computed for any given ϵ and P. Usually the probability of success is set to .95, and Z is set to the value necessary to achieve this probability. While the value of Z computed in this way is less than k^P, it is still quite large if ϵ is small. This analysis shows that pose determination is inherently a very difficult problem because $\epsilon = 1/k$, since there are k elements in each correspondence set. Therefore any prior estimate of the pose is potentially very useful since it increases ϵ by decreasing the number of possible correspondences in each correspondence subset.

Consider the case where the sensor primitives are points, and the model primitives are defined by a polynomial equation. This is one of the most natural ways to perform pose determination and refinement because using only points as sensor primitives means no time is taken for primitive extraction. The model primitives are defined by a polynomial equation, which in 2D is $\sum_{i=0}^{m}\sum_{j=0}^{n} a_{ij}x^i y^j = 0$ and in 3D is $\sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} a_{ijk}x^i y^j z^k = 0$. These polynomial equations define a curve in 2D and a surface in 3D. Using only polynomials for geometric primitives may appear to be restrictive, but it has been shown that all of the basic primitives in CAD systems can be converted to such implicit polynomial equations [25]. Since the parts whose pose we wish to find usually have an associated CAD definition, such a representation is not restrictive in practice.

Next we consider how to produce a hypothetical pose from P correspondences. Application of the pose transformation T should align the sensor primitives with the model primitives to which they correspond. If we have P correspondences, then substituting the transformed sensor primitive into the appropriate model equation will produce a system of polynomial equations. The unknown in the system is the pose transformation T, and solving the system will produce the hypothetical pose. The maximum number of such equations is six for 3D primitives and three for 2D primitives, since these are the degrees of freedom of objects in space. However, the number of equations may be less because of object symmetries.

The following example in which the model is a 2D polygon defined by a number of line segments [11] and the sensor data is a set of points in the plane illustrates this technique. In this case the pose transformation T consists of a rotation and translation in the plane. The pose has three degrees of freedom, so the number of correspondences from sensor points to polygon edges necessary to define a unique transformation T is three. The pose transformation T applied to a point (x, y) in the plane is defined by the following equation.

$$
\begin{aligned}
x' &= (x+h)\cos(\theta) - (y+k)\sin(\theta) \\
y' &= (x+h)\sin(\theta) + (y+k)\cos(\theta)
\end{aligned}
$$

The pose of the model is represented by the transformation $T = (\theta, h, k)$ which defines the rotation and translation in the plane. If T is applied to a sensor point, then the modified point must be on their corresponding line segments. If there are three sensor points $(x0, y0), (x1, y1)$ and $(x2, y2)$, then substituting these transformed values into the equations defining each model line, and setting $u1 = \cos(\theta)$ and $u2 = \sin(\theta)$ produces the following polynomial system.

$$
\begin{aligned}
&a0 + a1((x0+h)u1 - (y0+k)u2) + a2((x0+h)u2 + (y0+k)u1) \\
&b0 + b1((x1+h)u1 - (y1+k)u2) + b2((x1+h)u2 + (y1+k)u1) \\
&c0 + c1((x2+h)u1 - (y2+k)u2) + c2((x2+h)u2 + (y2+k)u1)
\end{aligned}
$$

Model Primitive	Degrees of Freedom - P
Line	2
Circle	3
Ellipse	5
Plane	3
Cylinder	4
Cone	5
Sphere	3
Torus	5

Table 2: The number of degrees of freedom P for different model primitives with a point as sensor primitive.

If each of these polynomials is set equal to zero and the equation $u1^2 + u2^2 - 1 = 0$ is added to encode the constraint that $\cos(\theta)^2 + \sin(\theta)^2 = 1$ the result is a nonlinear system of polynomial equations. Since the number of equations (four) equals the number of unknowns $(u1, u2, h, k)$ the implication is that the system has a fixed number of solutions.

This method for producing the set of equations whose solution is the hypothesized pose T can be used whenever the sensor and model primitives are defined by polynomial equations, and not just for the special case where the sensor primitives are points. As has just been shown, however, when the primitives are points the equations are particularly simple. Table 2 shows the degrees of freedom (and thus the number of correspondences P) when the model is a single geometric primitive ($k = 1$) of the given type and all the points correspond to this single primitive.

In order to create a hypothetical pose T, it is necessary to solve the nonlinear system of polynomial equations created by substituting the transformed sensor primitives into the equations of the model primitives. The ability to automatically solve for any nonlinear polynomial system is the key to the first step of the general random sampling algorithm for pose determination, i.e, generating a hypothetical pose from P correspondences.

5 Solving Nonlinear Polynomial Systems

There are two approaches to solving such systems. These are numerical methods and symbolic methods. Traditionally numerical methods, such a Newton-Rhapson, have been used, but these require an initial estimate which must be close to the actual solution. Even though more modern numerically based methods that use homotopies [19, 21] find all the solutions, they are computationaly very costly. Symbolic methods, on the other hand, are very useful since the final result is a polynomial equation in a single unknown which can be efficiently solved. From this solution, simple back-substitution can be used to solve for the other variables. The difficulty is how to eliminate all the free variables except one.

There are two symbolic methods which can be used for this purpose. They are the Gröbner basis technique [6] and multivariable resultants [7], both of which belong to the field of elimination theory. The end result of these methods is the elimination of all but one of the free variables, to create a polynomial in a single unknown. The solution to a polynomial equation with a single unknown variable exists in a closed form if it is of low order, or can be found numerically using fast algorithms. The solution for this unknown is then used to solve for another unknown, and so on in a recursive fashion until all the unknowns have been found [16]. Even though it has been proven mathematically that both of these symbolic methods are always able to solve a nonlinear polynomial system, the problem is that the time required to find the solutions is an exponential function of the number of equations and unknowns. Fortunately, the pose T has at most six free parameters, so it is necessary to solve no more than six simultaneous equations in order to find the hypothetical pose T.

The Gröbner basis method is used because it produces equations of the least possible degree. The degree of each equation in the system is that of the term with the highest degree. The degree of a term is the sum of the degrees of the individual powers of the variables in that term. For a system of equations the total degree of the system is the product of the degrees of each equation. In our example in the previous section, the total degree of the system of equations with unknowns $(u1, u2, h, k)$ is sixteen. The theorem of Bezout [16] says that the number of solutions of a polynomial system is less than or equal to its total degree. Unfortunately, this means that to solve the system it may be necessary to find the roots of a polynomial in a single unknown of very large degree. However, it is very often the case that there are multiple roots, in which case the number of distinct solutions (the actual degree) is significantly less than the total degree. In our example in the previous section the actual number of solutions is two, not sixteen. The Gröbner basis method produces only two solutions, while the multivariable resultant will produce sixteen solutions.

Using the Gröbner basis method, solutions have been produced symbolically for the about half the cases listed in Table 2. As modern symbolic algebra packages, such as MAPLE, are running on larger and faster computers, it should ultimately be possible to solve for T for all the cases in this table.

6 Examples

In this section a number of examples are described to illustrate the concepts discussed above. The first example will be the one discussed previously, where the model is a polygon in the plane and the sensor data is a set of points in the plane. The objective is to find the best pose, and it is not assumed that all the sensor points belong to the polygon. This is the classic pose determination problem with no known constraints on the pose.

To evaluate a hypothetical pose, the pose transformation T is applied to the sensor points and the number of sensor points within a small distance from the model is counted. In practice this distance must be determined from a careful study of the sensor and its noise characteristics. In this example, it is arbitrarily chosen. In Figure 1, two instances of the pose determination algorithm in action are shown. Fig. 1(a) and (c) show some simulated sensor data, and Fig. 1(b) and (d) show the model with the best pose aligning the model and sensor data. The pose was determined using a random sampling algorithm in which three sensor points were randomly selected, a hypothetical pose determined and the pose evaluated by counting the number of transformed sensor points that are within a certain distance of the model. The hypothetical pose T was determined by solving a nonlinear system of equations defined by these correspondences. The hypothesize and test strategy using random sampling makes it possible to avoid searching the entire interpretation tree to find the correct pose, as was done in [11]. This means that as the

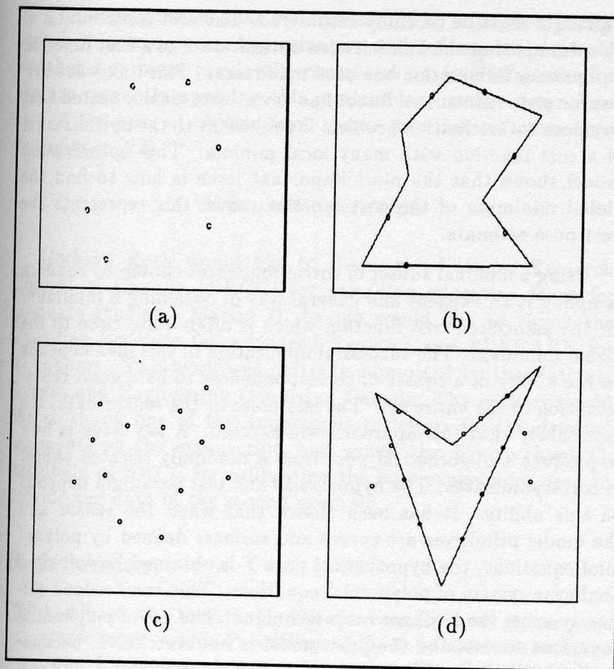

Figure 1: Two Dimensional Pose Determination (a) (c) Initial sensor points (b) (d) Computed pose with model superimposed on sensor points

number of model primitives increases, this approach will become faster than a method which must search the entire interpretation tree [17].

The following example applies random sampling to the pose refinement problem. The assumption in pose refinement is that an estimate of the pose is available, along with some indication of the accuracy of this estimate. Using the same model primitives as the previous example this situation would occur when the polygon describes the walls of a room, and the points consist of sensor data taken from a laser rangefinder [4]. If the rangefinder is mounted on an autonomous vehicle then the vehicle itself can provide the initial pose estimate from its internal navigation system [8]. However, the accuracy of this estimate decreases as the vehicle travels, so its must be periodically corrected by a pose refinement process. This correction is done by adjusting the vehicle pose till the sensor data has the best alignment with the 2D polygon environment model. This has been accomplished previously using a pose refinement algorithm based on numerical optimization [8]. While effective, the difficulty with this approach, and in fact with any local optimization procedure, is that there is no way of predicting the range of convergence, even for noise free data. In other words, as the error in the initial estimate of the vehicle position increases the optimization procedure will at some point fail to converge to the correct answer. In terms of the optimization model presented in Section 3 the process is converging to a local minimum instead of the global minimum. As we have discussed previously, this is due to the fact that the optimization approach to pose refinement assumes that the correspondences between the sensor and model primitives are correct.

By contrast, the random sampling algorithm does not make

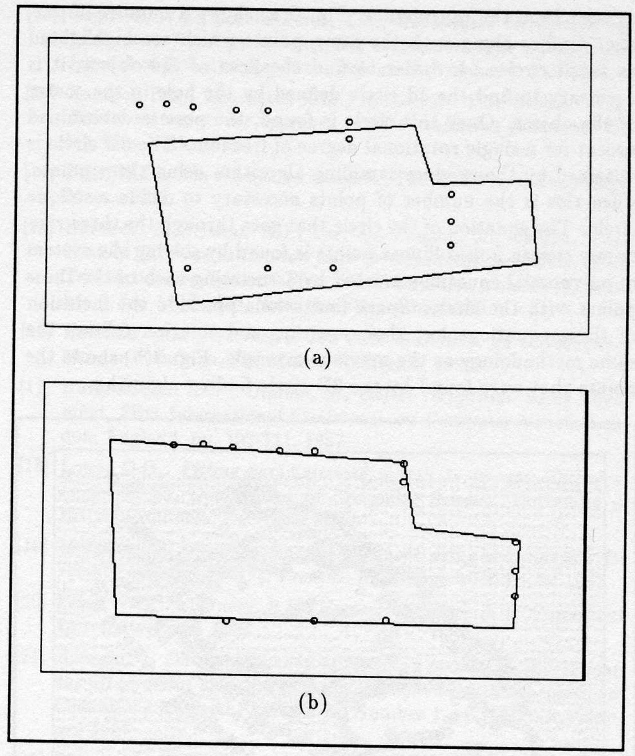

Figure 2: Two Dimensional Pose Refinement superimposed on sensor points (a) Initial sensor points and initial pose superimposed on sensor points (b) Final pose with model superimposed on sensor points

this assumption, and is guaranteed to convergence (at least probabilisticly) given enough random samples. It is not difficult to adjust the number of random samples according to the accuracy of the initial estimate. If the initial pose estimate is very accurate, few random samples are needed, and as this estimate worsens more random samples are required. In the worst case when the initial estimate is arbitrarily bad then pose refinement problem becomes pose determination. Fig. 2(a) show some simulated sensor data, along with the initial pose estimate that would be provided by the autonomous vehicle superimposed on the sensor data. Fig. 2(b) shows the refined pose obtained using the random sampling algorithm. Even though the same basic algorithm that was used in pose determination is used in pose refinement, the result is obtained much faster because the initial vehicle position is used to limit the number of possible correspondences during the random sampling process. We plan to apply this approach to adjusting the position of a mobile vehicle using actual range data.

The final example has as input a number of three dimensional points produced by a laser rangefinder mounted on a robot wrist [23]. The objective is to find the pose of an object called an "H-Fixture", which is the approximate shape and size of one of the grapple fixtures for the space station. This rangefinder collects parallel profiles of the object, and from these profiles is extracted a subset of the input called jump points. The latter are points which have a significant difference in depth from their neighbours. These jump points define the 3D outline of the object as seen from the rangefinder. Fig. 3(a) shows a number of parallel profiles along with the jump points which are highlighted as small circles. In order to find the pose of the object, it is necessary to find the 3d circle defined by the hole in the center of the object. Once this circle is found, the pose is determined except for a single rotational degree of freedom. The 3D circle is obtained by the random sampling algorithm using three points, since this is the number of points necessary to define a unique circle. The equation of the circle that goes through the three randomly chosen non-collinear points is found by solving the system of polynomial equations created by associating each of the these points with the circle. Space limitations preclude the inclusion of these equations but their creation and solution follows the same methodology as the previous example. Fig. 3(b) shows the points that were found by the 3D circle finding algorithm.

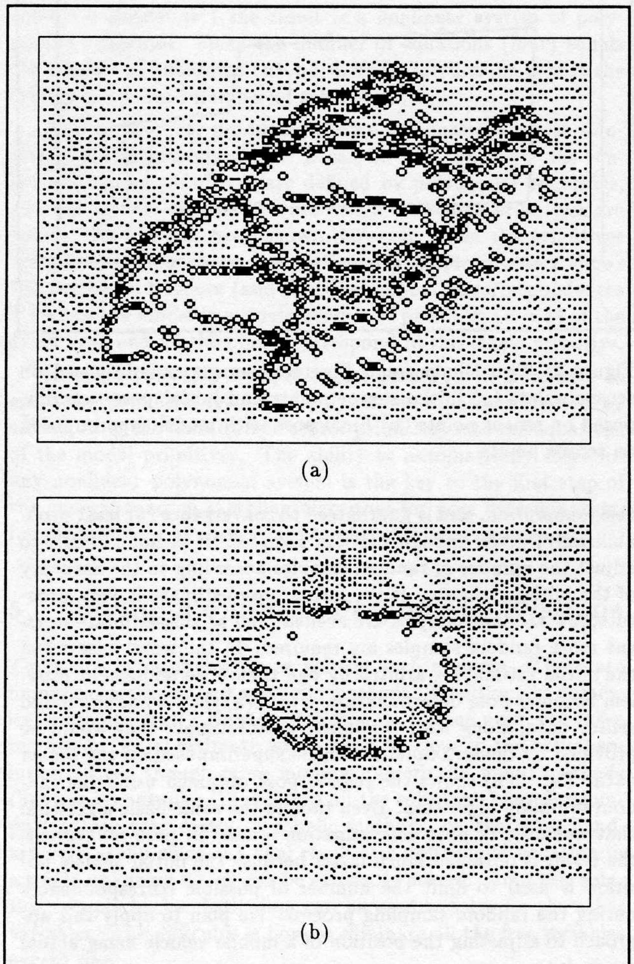

Figure 3: The extracted circle points obtained from the initial jump points (a) Initial jump points (b) The extracted circle points.

7 Discussion and Conclusions

Since there has already been so much work done in pose determination and refinement the contributions of the random sampling paradigm must be carefully considered. The first contribution of this work is that the link between optimization of a cost function and pose determination has been made clear. This link was obvious for pose refinement, but it has never been clearly stated that any pose determination problem is equivalent to the optimization of a cost function with many local minima. This optimization model shows that the most important issue is how to find the global minimum of the cost function, since this represents the best pose estimate.

Using a minimal subset of correspondences chosen by random sampling is an efficient and general way of obtaining a minimum for the associated cost function which is often quite close to the global minimum. The successful application of this idea depends on the ability of a subset of correspondences to be a good representation of the entire set. The less noise in the sensor data, the more likely that this approach will succeed. A key issue is how to produce a hypothetical pose from a randomly selected subset of correspondences. The hypothesize and test paradigm depends on this ability. It has been shown that when the sensor and the model primitives are curves and surfaces defined by polynomial equations, the hypothetical pose T is obtained by solving a nonlinear system of polynomial equations. This can be done efficiently using the Gröbner basis technique. The use of polynomial equations to describe the part model is not restrictive, because it is possible to translate the parts created by most CAD systems into this representation. We have not addressed the issue of parts which are defined by a set of primitives which are not all of the same type. One approach to this problem is to associate each sensor primitive to different kinds of model primitives, and then solve the resulting polynomial system of equations. This is an ongoing area of research.

One contribution of random sampling is its generality. Even though the idea of using a minimal set of correspondences to hypothesize a pose has been described previously [9], no mention has been previously made of how to extend this idea to any geometric primitive defined by an implicit equation. Random sampling can be used whenever a hypothetical pose can be produced from a minimal set of correspondences, which can be done whenever the models are described by polynomial equations. Other techniques such as geometric hashing [31] have only been applied to simple models such as lines and planes. Random sampling can also be used to make pose refinement algorithms more robust. If a previous estimate of the pose is available, this estimate can be used to limit the number of possible correspondences. However, it still may be the case that some correspondences are incorrect, which may cause significant errors in the final result. Pose refinement algorithms can be made more robust using the influence function approach [15] from robust statistics, but even for noise free data convergence depends on a good initial starting estimate. This is not the case for random sampling since for noise free data convergence is guaranteed if enough samples are taken. It is also that using random sampling pose refinement algorithms can deal with more than 50% bad correspondences. The difficulties that previous robust pose refinement methods using influence functions have had with a large percentage of bad correspondences are due to the heuristic way in which the starting position of the associated optimization algorithm was chosen [15]. With random sampling, the global minimum can be found even when there are more than 50% incorrect correspondences. In our first and third examples the percentage of correct correspondences (inliers) is considerably less than 50%.

The conclusion is that random sampling is a powerful approach to solving pose determination and refinement problems, and has not been previously considered in its full generality. We are currently applying this methodology to the determination of the pose of objects from sparse data taken from a laser rangefinder [22].

Acknowledgements

Gerhard Roth would like to thank the National Research Council of Canada (NRCC) for providing education leave for graduate studies. Martin D. Levine would like to thank the Canadian Institute for Advanced Research and PRECARN for its support. This work was partially supported by the National Science and Engineering Council of Canada. The authors would like to thank one of the anonymous reviewers for the detailed comments on the manuscript, and would also like to thank C. Archibald of NRCC for calibrating the laser rangefinder used to provide the data in the second example.

References

[1] Archibald, C., Merritt, C., *Pose determination of objects from sparse range images*, Proceedings of the 2nd Conference on Intelligent Autonomous Systems, Amsterdam, The Netherlands, December, 1989.

[2] Ayache, N., Faugeras, O.D., *Hyper: A new approach to the recognition and positioning of two dimensional objects*, IEEE Transactions On Pattern Analysis and Machine Intelligence, Volume 8, Number 1, pp. 44-54, 1986.

[3] Besl, P.J., *Describing geometric signals*, Workshop on range image understanding, Vision Interface, London, Ontario, pp. 52-85, June 19, 1989.

[4] Blais, F., Rioux, M., Domey, J., Beraldin, J.A., *A very compact real time 3-d range sensor for mobile robot applications*, Mobile Robotics 3, SPIE Volume 1007, pp. 330-338, Cambridge, Mass., 1988.

[5] Bolles, R. C., Fischler, M. A., *A ransac-based approach to model fitting and its application to finding cylinders in range data*, Seventh International Joint Conference on Artificial Intelligence, pp. 637-643, Vancouver, Canada, 1981.

[6] Buchberger, B., *Gröbner basis: an algorithmic method in polynomial ideal theory*, Recent trends in multidimensional system theory, N.K. Bose, Ed. Reidel, Hingham, Mass., pp. 184-232, 1985.

[7] Canny, J.F., *The complexity of robot motion planning*, The MIT Press, Cambridge, Mass., 1987.

[8] Cox, I.J., Kruskal, J.B., *On the congruence of noisy images to line segment models*, Second International Conference on Computer Vision, Tampa, Florida, pp. 252-258, 1988.

[9] Dhome, M., Richetin, M. Lapreste, J.T., Rives, G., *Determination of the attitude of 3-d objects from a single perspective view*, IEEE Transactions On Pattern Analysis and Machine Intelligence, Volume 11, Number 12, pp. 1265-1278, December 1989.

[10] Fischler, M.A., Bolles, R.C., *Random sample consensus*, Communications of the ACM, Volume 24, Number 6, pp. 381-395, June 1981.

[11] Gaston, P.C., Lozano-Perez, T., *Tactile recognition and localization using object models: the case of polyhedra in the plane*, IEEE Transactions On Pattern Analysis and Machine Intelligence, Volume 6, Number 3, pp. 257-266, 1984.

[12] Grimson, W.L., Lozano-Perez, T.L., *Model-based recognition and localization from sparse range and tactile data*, The International Journal of Robotics Research, Volume 3, Number 3, Fall 1984.

[13] Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., Stahel, W.A., *Robust statistics: the approach based on influence functions*, Wiley, New York, 1986.

[14] Hills, W.D., *The connection machine*, The MIT Press, Cambridge, Mass., 1985.

[15] Haralick, M.H., Joo, H.J., Lee, C., Zhuang, X., Vaidya, V.G., Kim, M.B., *Pose estimation from corresponding point data*, IEEE Transactions on Systems, Man and Cybernetics, Volume 19, Number 6, pp. 1426-1439, November/December 1989.

[16] Hoffman, C.M., *Geometric and solid modelling, an introduction*, Morgan Kaufman, San Mateo, California, 1989.

[17] Huttenlocher, D.P., Ullman, S., *Object recognition using alignment*, First International Conference on Computer Vision, London, England, pp. 102-111, 1987.

[18] Lowe, D.G., *Fitting parameterized models to images*, Technical report 89-26, Department of Computer Science, University of British Columbia, December 1989.

[19] Morgan, A., *Solving polynomial methods using continuation for science and engineering*, Prentice-Hall, Englewood Cliffs, 1987.

[20] Press, W.H., Flannery, B.P., *Numerical recipes in C*, Cambridge University Press, 1988.

[21] Renegar, J, *On the efficiency of Newton's method in approximating all zeroes of a system of complex polynomials*, Mathematics of Operations Research, Volume 12, Number 1, pp. 121-148, February 1987.

[22] Rioux, M., *Laser rangefinders based on synchronized scanning*, Applied Optics, Volume 23, pp. 3837-3844, 1984.

[23] Rioux, M., Blais, F., *Compact three dimensional camera for robotic applications*, Journal of the Optical Society of America, Volume 3, pp. 1518-1520, 1986.

[24] Roth, G., Levine, M.D., *Random sampling for primitive extraction*, International Workshop on Robust Computer Vision, Seattle Washingtion, October 1-3, 1990.

[25] Sederberg, T.W., Anderson, D.C., *Implicit representation of parametric curves and surfaces*, Computer Vision, Graphics and Image Processing, Volume 28, pp. 72-84, 1984.

[26] Silverberg, T.M., Davis, L., Harwood, D., *An iterative Hough procedure for three-dimensional object recognition*, Pattern Recognition, Volume 17, Number 6, pp. 621-629, 1984.

[27] Stockman, G.C., *Object recognition and localisation via pose clustering*, Computer Vision, Graphics and Image Processing, Volume 40, pp. 361-387, 1987.

[28] Stockman, G.C., *Matching images to models for registration and object detection via clustering*, IEEE Transactions On Pattern Analysis and Machine Intelligence, Volume 4, Number 3, pp. 229-241, 1982.

[29] Torn, A., *Global optimization*, Springer-Verlag, Berlin, 1989.

[30] Thompson, D.W., Mundy, J.L., *Three dimensional model matching from an unconstrained viewpoint*, IEEE International Conference on Robotics and Automation, Raliegh, North Carolina, March 31-April 3, pp. 208-220, 1987.

[31] Wolfson, H., *Model based object recognition by geometric hashing*, European Conference on Computer Vision 90, pp. 526-536, April 1990.