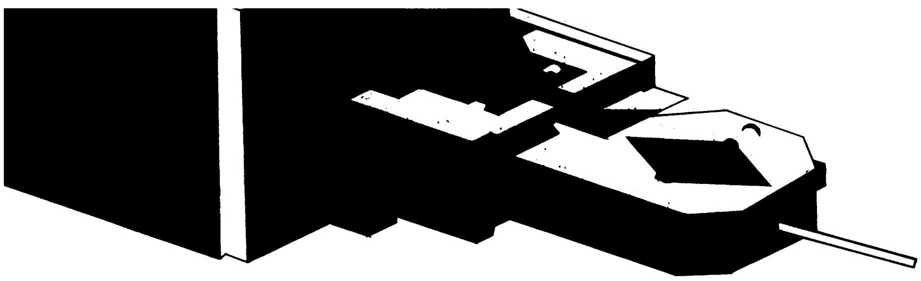


CAD Model-Based Localization of Parts in Manufacturing

Kristjan T. Gunnarsson and Friedrich B. Prinz
Carnegie Mellon University

We need automatic part localization in flexible manufacturing. One method of achieving it uses a CAD model and sparse data.

We present a solution to the problem of determining the location of a part in a robotic manufacturing situation. The problem is solved by matching a database surface description of the part with discrete points measured on its surface. The matching is accomplished through the calculation of a transformation. The transformation specifies the position and orientation of the part as it appears in front of the robot. The method was developed for use in flexible manufacturing where low precision, general purpose fixtures, and positioners are employed. Knowledge of the approximate location of the part is used. In a typical application the uncertainty in the location of the part is small relative to the dimensions of the part, but more than an order of magnitude greater than the tolerances allowed by the manufacturing operation.

The introduction of a robot into a manufacturing process requires an investment in time and effort to program the robot. Consequently, industrial robots have found use in large-batch manufacturing such as the automotive industry, where a large volume of identical parts justifies the initial outlays for programming. Profitable deployment of robots in small-batch manufacturing demands a high degree of flexibility with respect to the tasks the robot performs and adaptability to changes in the robot's environment. However, with flexibility comes uncertainty. The cost of precision fixtures precludes them from small-batch manufacturing. The robot must be equipped with intelligence that allows it to adjust to the location of the parts in its environment.

A human laborer can easily cope with the randomness of the factory environment. He will either pick up or approach the parts wherever they may be. The parts are generally not placed precisely in the same location every time, but randomly placed within a certain area. A *dumb* welding robot will attempt to weld the parts in precisely the same location every time, i.e. the programmed location. If the part is not precisely at that location, the robot will not weld properly; if the part is missing altogether, the robot will still attempt to weld, and will weld in thin air.

A robot able to determine the location of production parts in its working envelope significantly expands the scope of tasks and manufacturing situations in which it works. Set-up time and costs for a production run are reduced because the design and construction of precision fixtures is eliminated. In general, the need to tailor the environment precisely to the

0018-9162/87/0800-0066$01.00©1987IEEE

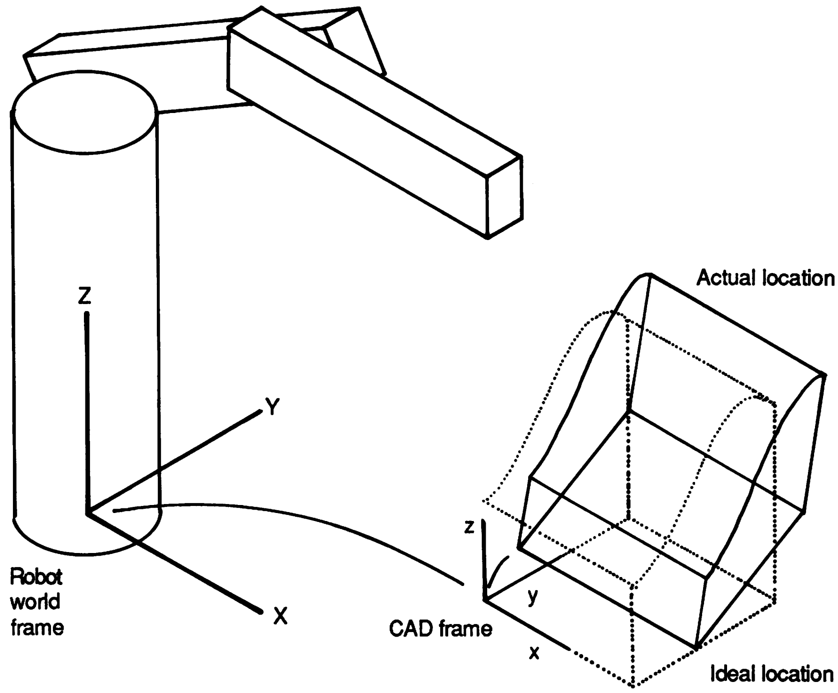


Figure 1. Definition of the world and CAD database (part) frames of reference, indicating an uncertainty in the location of a part. The ideal or planned location of the part is drawn in dotted lines, but an actual location is drawn in solid lines.

robot is relaxed because the robot can adapt to its environment.

Part localization

Part localization refers to the problem of determining the position and orientation of an object. Referring to Figure 1, localizing a part means being able to answer the following question: What are the Cartesian coordinates of any specified point on the part in the robot's world frame? In robotic manufacturing, the part location must be determined or assumed before an operation can be accomplished. In drilling, for example, holes cannot be placed accurately unless the part's position is known; long holes may be incorrectly angled if the part's orientation is not known.

To answer this question, all points on the part must be known. A CAD database description of the surface of the part supplies this information. Associated with the CAD database description is a coordinate frame referred to as the CAD or part frame; this is the frame in which the part has been designed and viewed on the screen of a CAD station (see Figure 1). However, the location of the part, and the CAD frame, in the robot's world frame depends on where the part is placed in front of the robot. The location of the CAD frame in the world frame is given by the transformation, $\mathbf{T}_l$, which is represented by a 4 × 4 homogeneous matrix of the form

$$\mathbf{T}_l = \begin{bmatrix} \mathbf{n}_1 & \mathbf{o}_1 & \mathbf{a}_1 & \mathbf{t}_1 \\ \mathbf{n}_2 & \mathbf{o}_2 & \mathbf{a}_2 & \mathbf{t}_2 \\ \mathbf{n}_3 & \mathbf{o}_3 & \mathbf{a}_3 & \mathbf{t}_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The column vectors of $\mathbf{T}_l$, $\mathbf{n}$, $\mathbf{o}$, and $\mathbf{a}$, specify the orientations of the CAD frame's unit vectors in the world frame, and the column vector $\mathbf{t}$ specifies the origin of the CAD frame in the world frame. If $\mathbf{x}$ is some point on the part expressed in the CAD frame, then $\mathbf{T}_l\mathbf{x}$ is the same point in the world frame.

If the location of the CAD frame is uncertain, points must be measured on the part's surface as it appears in front of the robot. By mathematically matching the CAD database description of the part with the measured points we can find the location of the part. The objective of the matching is to transform the CAD frame into the robot's world frame and thereby determine $\mathbf{T}_l$. The only information needed from the database are the equations of the surface of the part. A CAD database may contain information about the edges, vertices, and volumes of a part, but we don't need this information because no attempt is made to measure edges, vertices, etc. In summary, localization requires a CAD database surface description of the part, measurement of surface points on the actual part, and matching the two.

The database description is assumed complete in the sense that every point on the part's surface is accounted for. However, a part can be localized with only six measured points. The measured data is considered *sparse* when 6 to, say, 60 points are measured. This is frequently the case if tactile position sensing is used. However, when machine vision and laser ranging systems gather hundreds of data points, the data is called *dense*. Many optical systems are highly suitable for part localization applications because of their speed and accuracy, and dense data can be reduced to sparse data. The method described in this article was developed for sparse data, but is equally suitable for tactile and nontactile (optical and ultrasonic) sensing. For example, a simple tactile sensor used for part localization uses electrical contact between the electrode of an arc welding torch and the workpiece.

Part localization has primarily been studied by researchers working in the areas of machine vision and scene analysis.[1] Although these technologies are not necessarily intended for application in manufacturing, they have nevertheless found use there. The two principal objectives in scene analysis are recognition of the objects making up the scene and composition of a general description of the scene.[2] Accurate localization of the objects in the scene is a secondary consideration in many instances. In manufacturing applications of machine vision, part localization is important and thus has received considerable attention.[3] Part localization methods developed by machine vision scientists are based primarily on the use of dense data, because the nature of vision sensing is to gather large amounts of data.[4,5] Sparse data methods invented by vision scientists[6] usually include the reduction of the original dense data as a first stage.

The method described in this article differs in many respects from machine vision methods. A primary function of scene analysis algorithms is identification and recognition of the constituent objects of the scene. Here, we assume that the part to be localized is known. This assumption is justified in most manufacturing situations. The emphasis in our method is on accurate localization and sparse data. The method does not require estimation of surface normals nor calculation of the surface of the part.

Applications of part localization

Part localization is needed in various robotic manufacturing situations. In mechanical assembly, tight-fitting parts cannot be assembled unless their relative positions and orientations are known. Successful grasping depends on knowing the accurate orientation of a part prior to attempting the grasp. Mechanical compliance only reduces the needed localization accuracy. In robotic arc welding, for example, the welding torch must be positioned accurately in the weld groove. In gas-metal-arc welding (GMAW), the cross-seam positioning uncertainty must not be worse than the diameter of the electrode. Normal to the seam, the accuracy must be even greater. Failure to accurately position the torch produces a defective weldment and may possibly damage or destroy the welding torch. Gauge-free inspection directly measures key points on the surface of a manufactured part, to check dimensional integrity as specified by the design. The part is matched with a prototype or a database by measuring corresponding points, both on the part and in the reference description. However, the part location must be known before a prescribed point on the part can be measured.

The general solution to the localization problem is to avoid it by use of dedicated fixtures or specialized machinery. This solution is not economically justified in small-batch and flexible manufacturing, where general-purpose fixtures are used. The greater flexibility of such fixtures is obtained at the expense of accuracy; parts of various shapes and sizes can be fixed in them, but none can be positioned accurately.

Part-localization methods have an important application in connection with *model-based off-line programming* of robots. The alternative to off-line programming is *teach-by-doing*, where the robot arm is physically led through the motions of the task and points along its trajectory are stored in the controller's memory. Robots are used to their best advantage when programmed off-line, irrespective of the manufacturing task, be it part handling, welding, or inspection. Unlike the teach-by-doing mode, model-based off-line programming does not require the physical use of a robot and part, and production is not disrupted. With off-line programming, a CAD database description of the part is used to plan the motion of the robot. For instance, in the case of flame cutting or deburring, the path over which the robot must carry the tool and its orientation are calculated from the database description of the part's surface. A kinematic model is then used to calculate the joint motions of the robot, or the robot is programmed directly in world coordinates.

Part localization is essential for off-line programming to be effective. The complete tool path and orientation relative to the part are calculated off-line. However, the position and orientation of the path as a whole depend on the actual location of the part as it appears on the factory floor in front of the robot. This situation is demonstrated in Figure 1. The part, and consequently the tool path and tool orientation, is described in the CAD frame, which is independent of the part's actual location, but the robot trajectory depends on the actual location of the part in its world frame.

Robots are used to their best advantage when programmed off-line.

Assume that a task is specified by a sequence of Cartesian points, $\mathbf{x}_i$, along the surface of a part; these are the points the robot tool must follow to carry out the task. If these points have been generated at a CAD station using a model of the part, then the trajectory points are specified in the CAD frame. (Notice that this task description is independent of the part's actual location.) Normally, the parts are expected to appear in front of the robot in a planned location called the ideal location and specified by the transformation $\mathbf{T}_I$ (see Figure 1).

If a part now appears in the ideal location, then what are the robot's trajectory points for the specified task? To answer this question, the points in the CAD frame must be transformed, or projected, into the robot's world frame:

$$\mathbf{x}_{W_i} = \mathbf{T}_I \mathbf{x}_{CAD_i} \quad (2)$$

where $\mathbf{x}_{CAD_i}$ are the task points specified in the CAD frame, and $\mathbf{x}_{W_i}$ are the corresponding robot trajectory points in the robot's world frame.

In manufacturing situations where the location of a part is uncertain, the preceding procedure is likely to be inaccurate because the ideal location transformation $\mathbf{T}_I$ may not refer to the actual part location. Referring to Figure 1, a part is shown in a location differing from the ideal location by the offset transformation $\mathbf{T}_{off}$. In this case, the true location transformation is given by the multiple of the ideal and offset transformations, and the robot trajectory points are

$$\mathbf{x}_{W_i} = \mathbf{T}_I \mathbf{T}_{off} \mathbf{x}_{CAD_i} \quad (3)$$

The part localization method presented in this article determines the true location (transformation) of a part using its CAD database description and discrete measured points on the actual part.

The method just presented gives a robot the ability to adjust to different locations of the parts in its working envelope. Note that this procedure is not limited to model-based off-line programming. In fact, it is equally applicable to the teach-by-doing method. In this case, a robot has been taught a trajectory for a part in a specific location. If the next part is in a different location, its true location is found by a localization algorithm and the original trajectory is modified in a way similar to that shown above.

Localization of polyhedra

This section presents a solution for localization of a part with a surface made up of planar surface facets or patches. Past research has devoted considerable attention to such surfaces, commonly encountered in design and frequently used to model or approximate surfaces of modest curvature.

Previous methods of localization have not only required the measurement of surface points, but also the local estimation of surface normals.[3,6] Local estimation of surface normals on a surface requires that three points (or more) on a planar facet be measured. The local normal is calculated by passing a plane through the points; the normal of that plane becomes the local estimate of the surface normal. Although surface points must be measured on a part to indicate its location, local surface normal estimation is not necessary.

It is desirable to have a method of localization that does not require local surface normal estimation. Local surface normal estimation is sensitive to random surface roughness and measurement uncertainties, which can lead to imprecise localization, but high localization accuracy is of major importance in most manufacturing applications.

Accurate surface normal estimation requires that the points used in calculating the normal be widely spaced over the surface facet to minimize the sensitivity to errors. Many points clustered together, which are readily obtained with optical sensing systems, are unsuitable. When only the approximate location of a part is known, reliable measurement of widely spaced points on a specific planar facet is difficult. Furthermore, measured points are often at a premium in manufacturing situations. This is especially true if tactile sensing is used. It must be possible to localize a part with a minimum number of points.

We show a method for the localization of arbitrarily-shaped surfaces in a later section. This method is based on repeated polyhedral approximations of the curved surface. While the measurement of the surface normal of a plane is elementary *in principle*, the local estimation of the normal of a curved surface is not. Therefore, it is desirable to have a localization method that does not require surface normals.

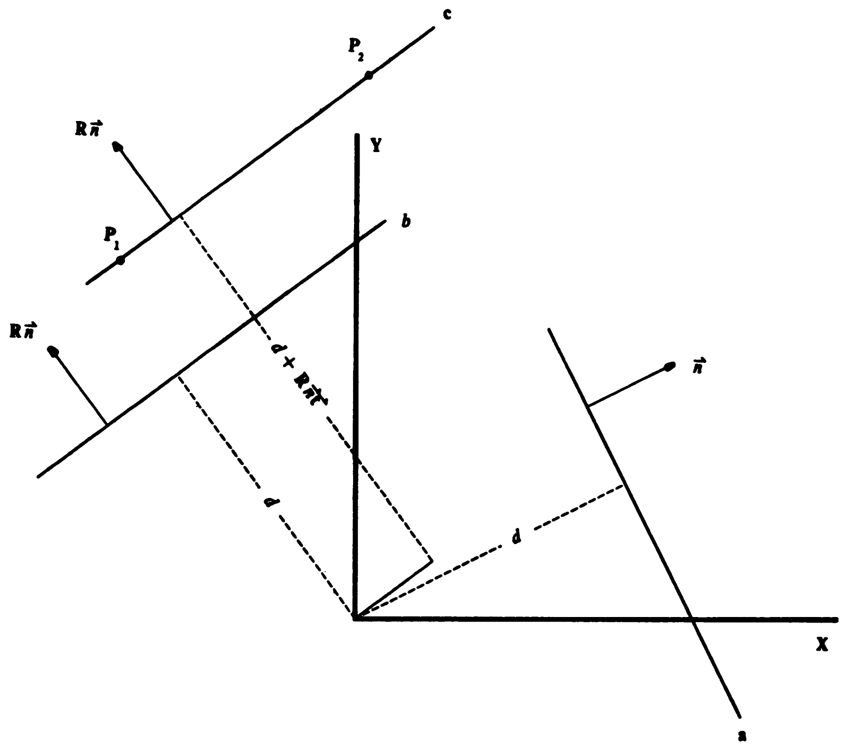


Figure 2. Illustration of the rotation and translation of a plane (shown in two dimensions) and the effect of these operations on the describing equation, where "a" is the untransformed state, "b" is the rotated state, and "c" is the rotated and translated state.

Problem formulation. We will not present the mathematical details of localizing a part by means of a rigid body transformation here.[5] Instead we discuss the basic ideas and formulations of the procedure.

Assume that a polyhedron must be localized. Its CAD database description gives the equations of all of its planar facets, i.e. the set of m planar surface facets, $S_i(\mathbf{n}_{3\times1}, d_{1\times1})$ i $= 1, 2, 3, \ldots, m$. Each facet, S_i, is characterized by its unit surface normal vector, $\mathbf{n}_i$, and the shortest distance from the facet to the origin d_i, which is a scalar. This is the standard vectorial representation of an unbounded plane. The normal vectors and shortest distances ($\mathbf{n}_i$ and d_i) are given in the CAD frame. Points are measured on the surface of an actual part either by a robot-mounted, or a remote, sensor, and a set of discrete points, $\mathbf{x}_{j_i}\, j = 1, 2, \ldots, m_i$, is generated. These points are given in the robot's world frame. The Cartesian point (vector) $\mathbf{x}_{j_i}$ denotes the jth point measured on the ith surface facet, and m_i is the number of points measured on the ith facet. Note that the correspondence of every measured point to a specific facet is assumed known. Localizing the part then means finding a transformation that matches the planar surface facets with their corresponding points.

In general, if a plane and a point are given it is a simple matter to check if the point lies in the plane. This is done by inserting the point into the equation for the plane:

$$\delta = \mathbf{n}\,\mathbf{x} - d \qquad (4)$$

Expression 4 gives the distance, δ, of the point $\mathbf{x}$ from the given plane; if the point is in the plane, its distance to it is zero. A matching between the surface and the measured points can now be defined: Match the surface with the measured points by making the distances between the points and their corresponding surface facets zero, or as small as possible. This is a very natural definition of a matching. The mechanical task of inserting a peg in a hole can be viewed as the task of minimizing the distance between the surfaces of the hole and peg, respectively.

A matching has now been defined, but what instrument is available for affecting this matching? The answer is the localization transformation. The distances between the measured points and the surface will be made as small as possible by appropriately *rotating* and *translating* the CAD database surface. With reference to Figure 1, the localization method must answer the question of where to position the CAD frame in the robot's world frame, and how to orient it.

A rigid body transformation is composed of two parts: rotation and translation. The translation is a vector (the fourth column vector of the matrix in Expression 1), but the rotation operator is a matrix (the upper left-hand 3×3 submatrix of Expression 1). Figure 2 illustrates the mathematical operation of rotating and translating a plane in two dimensions; in two dimensions a planar facet becomes a line segment. Assume that the line segment **a** shown in Figure 2 must pass through the points P_1 and P_2 also shown in the figure; the line segment must be made to pass through the points by rotating and translating it. Denote the rotation matrix by **R** and the translation vector by **t**. The effect of rotating segment **a**, $S(\mathbf{n}, d)$, is the rotated segment **b**, $S(\mathbf{R}\ \mathbf{n}, d)$. The rotation changes the orientation of the line segment (the normal vector), but has no effect on the shortest distance to the origin (as shown in the figure). The effect of translating the rotated line segment is the segment **c**, $S(\mathbf{R}\ \mathbf{n},\ d + (\mathbf{R}\ \mathbf{n})\mathbf{t})$. The translation has no effect on the segment's orientation; it only changes its distance to the origin.

Expression 4 is the distance of a point **x** from its untransformed line segment. The distance after the transformation is

$$\delta_t = (\mathbf{R}\ \mathbf{n})\mathbf{x} - ((\mathbf{R}\ \mathbf{n})\mathbf{t} + d) \qquad (5)$$

In the general three-dimensional case, the distance of a point $\mathbf{x}_{j_i}$ from its corresponding rotated and translated plane, $S(\mathbf{R}\ \mathbf{n}_i,\ d_i + \mathbf{R}\ \mathbf{n}_i\ \mathbf{t})$, is

$$\delta_{j_i} = (\mathbf{R}\ \mathbf{n}_i)\mathbf{x}_{j_i} - ((\mathbf{R}\ \mathbf{n}_i)\mathbf{t} + d_i) \qquad (6)$$

The rotation and translation must be so chosen to make the above expression zero (or as small as possible) for all points (j's) and surface facets (i's). Notice that the entire surface is transformed as a whole, i.e. all of its facets undergo the same rotation and translation. This leads to the least squares minimization problem,

$$\underset{\mathbf{R},\mathbf{t}}{\text{minimize}} \sum_{i=1}^{m} \sum_{j_i=1}^{m_i} ((\mathbf{R}\mathbf{n}_i)\mathbf{x}_{j_i} - (\mathbf{R}\mathbf{n}_i)\mathbf{t} - d_i)^2 \qquad (7)$$

The squared expression is the distance of each point to its respective facet. These distances are summed over all points on all planes, and the total must be minimized by properly calculating the transformation (**R** and **t**). The individual distances are squared so that positive and negative distances do not cancel out.

There are two major reasons for formulating the problem as an optimization problem. First, in this way, one problem formulation covers all possible situations regarding both the total number and distributions of the measured points on the facets. Second, the effect of possible noise in the measured points can be reduced by measuring more than the minimum number of points and overconstraining the problem. In the case of perfect measurements and perfect agreement between the CAD database and the part, all distances and their squared sum will be zero when the transformation has been correctly calculated. If errors exist, an optimal fit between the measured data and the database will be calculated. The numerical solution is calculated by an iterative procedure based on Lagrange multiplier and Newton-Raphson methods.[5]

The above problem formulation places no upper bound on the number of facets, and places no special conditions on the discrete points. However, every mathematical problem must be properly formulated.

The most general CAD database description of a surface or part is in terms of discrete points.

In the present case, the given surface and points must allow for the calculation of a unique transformation. Two basic conditions must always be met:

- A total of at least six points, measured on at least three facets, must be provided.
- Any three facets for which at least one point was measured on each of them must have surface normals that are linearly independent.

At least six points must be measured because a rigid body transformation is characterized by six independent variables: three degrees of translation (x, y, and z), and three angles of rotation (the three euler angles, or the roll, pitch, and yaw angles). If points are measured on only three facets, the only acceptable distributions of exactly six points on those surfaces are either 2:2:2 or 3:2:1. If only one point is measured on each facet, at least six facets are required. If these conditions are not met, a mathematical singularity is encountered and a solution cannot be calculated. Furthermore, notice that the formulation does not assume the possibility of estimating local surface normals from the sensed data.

The correspondence of measured points to specific surface facets is a basic assumption of the algorithm. If a polyhedron has n facets and k arbitrary points are measured on it, the total number of combinations relating the points to the facets is n^i. Assuming six facets and six points, this number is 46,656! Grimson and Lozano-Perez[6] considered all possible combinations, but presented powerful geometric pruning methods for eliminating impossible point-facet combinations. However, their method requires surface normal estimation. The correspondence assumption is justified when an estimate of the part's true location is available. If this estimate is accurate to within a fraction of the part's dimensions, the correspondence of measured points to facets can be guaranteed. This is the case in the manufacturing applications for which this method was developed.

Demonstration. The localization method is demonstrated in Figure 3, which shows the localization of a part made up of six planar facets. In the figure, the part drawn in light lines represents the CAD database surface in the ideal location. Instead of drawing discrete points (six points were used in this demonstration, one on each surface) representing the actual location of the part, another part is drawn in heavy lines for the same purpose. This helps the observer, who otherwise would have great difficulty visualizing the progress of the algorithm. The numerical procedure is both reliable and fast; typical execution time for this procedure on a VAX 11/750 is on the order of a second.

The accuracy of the localization method is primarily governed by the accuracy of the sensing system. As a rule of thumb, we can expect the localization inaccuracy to be on the same order as the combined average surface roughness and sensor uncertainty. The selection of measurement points is also important. Whenever possible, points should be measured on those facets that have the greatest angular separation, i.e. facets whose normals are closest to 90 degrees apart. This maximizes the computational sensitivity and consequently the accuracy. The points should also be spread over the surface to facilitate accurate orientation estimation.

Localization of quadric surfaces

The method described in the preceding section has been extended to allow the localization of surfaces comprised of quadric patches. Quadric surfaces deserve special attention for a number of reasons. These surfaces include planes, cones, cylindrical and elliptical pipes, spheres, etc., and are described by a second-order implicit equation in analytical geometry. They are common in mechanical design and thus frequently encountered in localization problems. Surfaces represented by implicit equations of order greater than two are difficult for a designer to visualize. Consequently, explicit parametric equations such as splines are used when designing a surface of order higher than two.

Figure 4 demonstrates this method, showing the localization of a part made up of two intersecting pipes. The part drawn in light lines represents the CAD database surface in the ideal location, and the part drawn in heavy lines represents the actual location of the part. Six points were used in this demonstration, three on each pipe. Again, the algorithm is seen to accurately converge in few iterations.

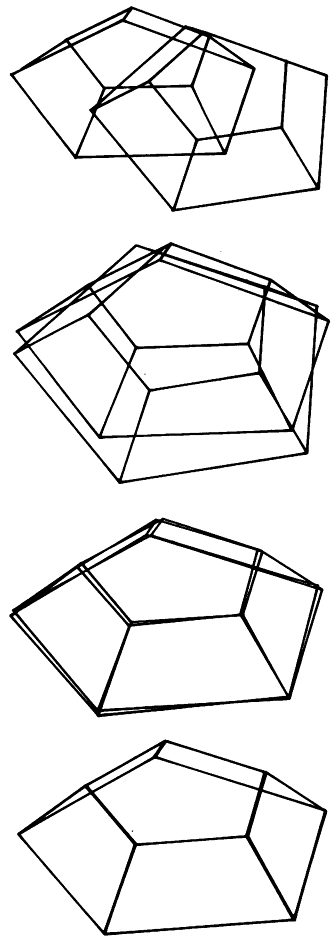


Figure 3. Demonstration of the localization algorithm for polyhedra. The part has six facets, and we assume one data point measured on each facet. The CAD database surface is drawn in light lines in the ideal location; the part is drawn in heavy lines in the actual location.

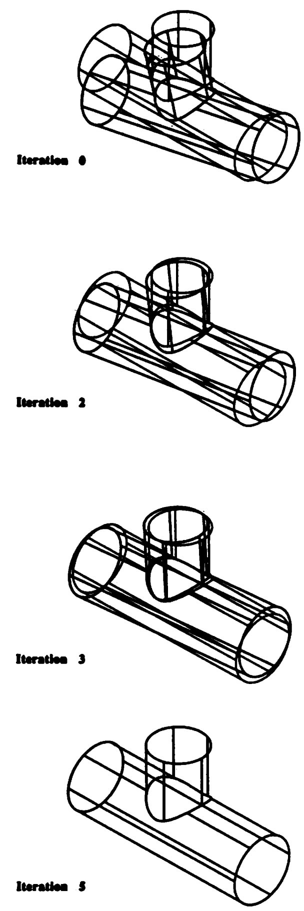


Figure 4. Demonstration of the localization of a part made up of two intersecting pipes. The CAD database surface is drawn in light lines in the ideal location; the part is drawn in heavy lines in the actual location.

Localization of parametric surfaces

The basic problem to solve when localizing a surface is to match points measured on a surface to a mathematical CAD database description of that surface. The matching procedure is to reduce the distances between the points and the surface as much as possible. The most elementary mathematical operation in a matching algorithm calculates the distance between a point and the surface. This makes implicit equations the ideal database representation because they take the form

$$f(x, y, z) = 0 \tag{8}$$

Simply by inserting the Cartesian coordinates of a point, (x, y, z), into the implicit equation we can obtain a measure of the distance between the point and the surface.

The most general CAD database description of a surface or part is in terms of discrete points, because all other surface descriptions can be reduced to that form. However, for localization purposes a database description must be complete in the sense that any point measured on an actual surface must be accounted for in the description. This can be accomplished by use of interpolating functions, such as splines, which are explicit representations of a surface. The Cartesian coordinates of any point on the surface are given by the equations

$$x = f_1(u, v) \tag{9}$$

$$y = f_2(u, v) \tag{10}$$

$$z = f_3(u, v) \tag{11}$$

where u and v are independent parametric variables, generally referred to as surface coordinates. Various kinds of splines are available, but for localization, the simplest ones are bicubic parametric splines:

$$\mathbf{x}(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} \mathbf{a}_{ij} u^i v^j \tag{12}$$

where u and v are the curvilinear surface coordinates, $\mathbf{a}_{ij}$ contains the spline parameters, and $\mathbf{x}$ is any point on the surface.

Explicit surface equations map a point in a two-dimensional surface space to a three-dimensional Cartesian space:

$$\mathbf{P}_2(u, v) \rightarrow \mathbf{P}_3(x, y, z) \tag{13}$$

When attempting to match a CAD database expressed by explicit equations with measured points, a discrepancy in spaces becomes obvious: the database is in a surface space, but the measured points are in a Cartesian (world) space. It is not trivial to calculate the distance of a point to a parametric surface; we must use an iterative numerical procedure. The localization method for polyhedra requires us not only to calculate the distance of a point to the surface, but also requires an analytical expression for the distance. In the case of parametric surface descriptions, no such expression can in general be derived.

The localization method presented below is an iterative scheme based on repeatedly approximating the splined sur-

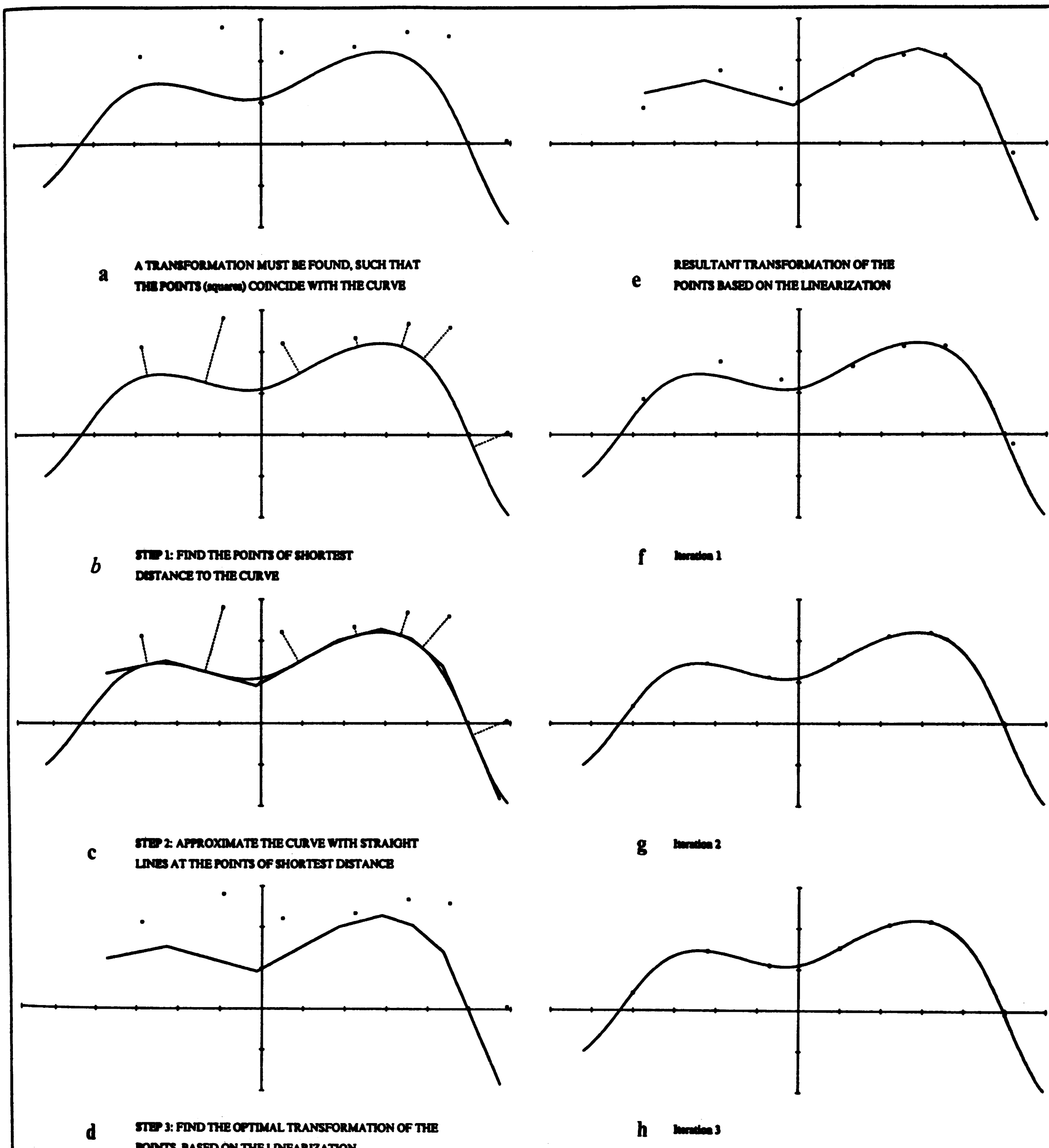


Figure 5. Two-dimensional demonstration of the localization algorithm for parametrically described curves, showing the localization of a cubic spline.

face with a polyhedron. At the start of each iteration, the algorithm finds the points of shortest distance to the splined surface from each measured point. The spline is approximated by planes (or, alternatively, quadrics) at each such point. A transformation is calculated based on these approximations and the measured points using the method for localizing polyhedra. The CAD database surface is then transformed closer to the measured points. This procedure is repeated until a satisfactorily accurate match has been reached. With this approach, the curved surface to be localized is approximated by an open polyhedron; this approximation is dynamically adjusted as the surface and the measured points are moved closer together. This method applies to both concave and convex surfaces.

The method is summarized as follows:

(1) Assume the CAD database surface is in the ideal location.

(2) For each measured point, calculate the point of shortest distance on the surface.

(3) Approximate the CAD database surface with planes at the points of shortest distance.

(4) Calculate an optimal transformation between the approximated surface and the measured points using the method for localizing polyhedra.

(5) Apply the calculated transformation to the CAD database surface.

(6) Test for convergence. If the transformation calculated in Step 4 is sufficiently close to the identity transformation, then quit; else go back to Step 2.

This method is similar to Newton's method for calculating the roots of functions and shares many of its convergence characteristics.[5] Figure 5 demonstrates the method in two dimensions.

Figure 5a shows a curve segment (database description) and seven discrete points (measured points). The problem is to find a transformation to match the curve and points. In this demonstration the transformation is applied to the points instead of the curve (database); this is a minor modification, but computationally simpler. As outlined above, the first step is to find the points on the curve of shortest distance to the measured points; Figure 5b shows the results of this operation. The second step involves linearizing the curve about the points of shortest distance, as illustrated in Figure 5c and Figure 5d, where the approximating polygon is drawn together with the measured points. The third step involves finding the optimum transformation of the points based on the linearization; the results of this transformation are shown in Figures 5e and 5f. Note that this procedure converges in one iteration when the surface is a polyhedron. As seen in Figure 5f, the points have moved very close to the spline despite the large initial offsets. Figures 5g and 5h show how the whole process is repeated twice, with a satisfactory match obtained in only three iterations.

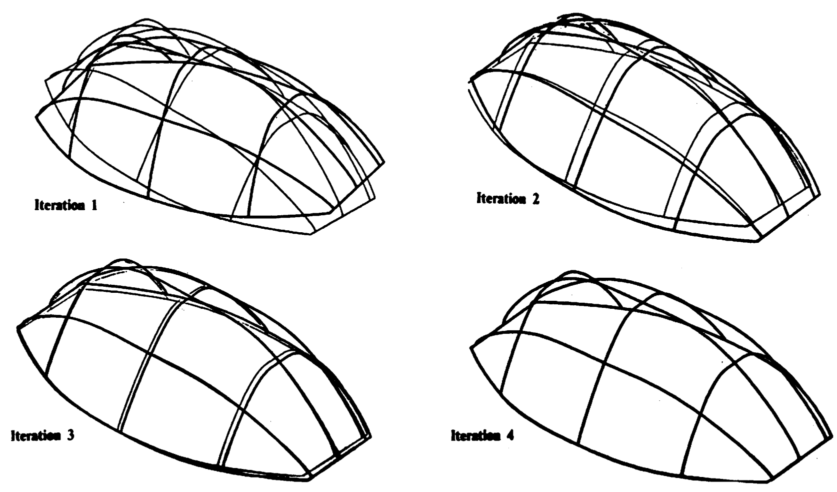


Figure 6. Localization of a surface described by parametric equations (bicubic splines). The CAD database surface is drawn in heavy lines in the ideal location; it is drawn in light lines in the actual location.

Figure 6 shows a three-dimensional demonstration of this method. The hemispheroidal surface is reminiscent of a canopy. This type of surface is difficult to represent with implicit equations; it is highly curved in certain places, but nearly planar in others. Figure 6a shows the surface in light lines rotated by about 10 degrees about all three axes and translated about 1/5 of its scale from the ideal location (heavy lines). Eight measured points, widely distributed over the surface, were used in the simulation, six points around the sides and two on the top. Convergence took place quickly, which is typical of this method.

This localization method is particularly effective against translation offsets. Translations affect the surface uniformly; all points are displaced equally. Rotations, on the other hand, do not displace all surface points equally; the greater the radius of gyration of a point (distance from the axis of rotation), the more it is displaced. The method requires an initial estimate of the actual location of the surface (the ideal, or planned, location), which is possible in most manufacturing situations. Specifically, the method is intended for localization of parts in low-precision general-purpose fixtures, for jigs, and on positioners, for which such information is readily obtainable.

We have demonstrated a method of part localization designed for manufacturing applications. This method is based on matching a CAD database description of the part's surface with data measured on an actual surface. The method is suitable for use with both tactile and nontactile sensing systems, and for both simple surfaces (polyhedra and quadrics) and arbitrarily-shaped surfaces. The method uses knowledge about the approximate location of the part prior to precise localization. It requires neither surface normal data nor the determination of the equation of the surface from the data. Instead, we obtain a direct matching through a rigid body transformation between the data and database.

The major manufacturing advantages of part localization follow:

• Part localization allows the use of low-precision, but general-purpose, fixtures needed in flexible and small-batch manufacturing. It enables a robot to adapt to variations in the location of parts in its working envelope.

• The design and construction of precision jigs and fixtures is time consuming and expensive, and localization methods offer potential savings in time and cost.

• Methods of automatic localization enable gauge-free inspection of parts.

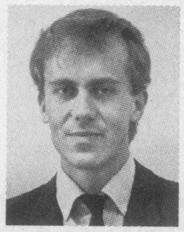
• Part-localization methods provide a critical link in the integration of CAD/CAM with robotics. They bridge the gap between the CAD station and the factory floor in model-based off-line programming of robots.□

Acknowledgment

The research presented in this article was supported by the Welding Consortium, The Robotics Institute, Carnegie Mellon University.

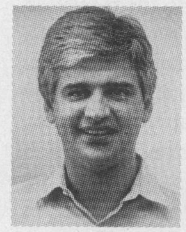
References

1. R.C. Bolles and P. Horaud, "3DPO: A Three-Dimensional Part Orientation System," *Int'l J. Robotics Research*, Fall 1986, pp. 3-26.
2. K. Ikeuchi, "Generating an Interpretation Tree From a CAD Model to Represent Object Configurations for Bin-Picking Tasks," Tech. report CMU-CS-86-144, Dept. of Computer Science, Carnegie Mellon Univ., August 1986.
3. O.D. Faugeras and M. Hebert, "The Representation, Recognition, and Locating of 3D Objects," *Int'l J. Robotics Research*, Fall 1986, pp. 27-52.
4. M. Brady et al., "Describing Surfaces," *Computer Vision, Graphics, and Image Processing*, Vol. 32, 1985, pp. 1-28.
5. K.T. Gunnarsson, *Optimal Part Localization by Database Matching With Sparse and Dense Data*, PhD dissertation, Dept. of Mechanical Engineering, Carnegie Mellon Univ., May 1987.
6. W.E. Grimson and T. Lozano-Perez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data," *Int'l J. Robotics Research*, Fall 1984, pp. 3-35.



Kristjan T. Gunnarsson is a research engineer in the Department of Mechanical Engineering at Carnegie Mellon University. His research activities include automatic tool calibration for robots, automated inspection of aluminum castings, and off-line programming of robots.

Gunnarsson received a bachelor's degree from the University of Iceland in 1979, a master's degree from Pennsylvania State University in 1981, and a PhD degree from Carnegie Mellon University in 1987, all in mechanical engineering.



Friedrich B. Prinz is a professor in the Department of Mechanical Engineering at Carnegie Mellon University. He conducts research in the areas of robotics, CAD/CAM, and the application of artificial intelligence for manufacturing.

Prinz received his PhD in physics from the University of Vienna in 1975.

Readers may write to Prinz at the Department of Mechanical Engineering, Scaife Hall, Rm. 403, Carnegie Mellon University, Pittsburgh, PA 15213.