A Quasi-Linear Method for Computing and Projecting onto C-Surfaces: Planar Case

George V Paul and Katsushi Ikeuchi

Dec 1 1996
CMU-RI-TR-96-37

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890

This work was sponsored in part by National Science Foundation under IRI-9224521 and in part by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson, AFB, OH 45433-6543 under F33615-90-C-1465 and F33615-93-1-1282.

A Quasi-Linear Method for Computing and Projecting onto C-Surfaces: Planar Case

George V Paul and Katsushi Ikeuchi

Dec 1 1996
CMU-RI-TR-96-37

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890

This work was sponsored in part by National Science Foundation under IRI-9224521 and in part by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson, AFB, OH 45433-6543 under F33615-90-C-1465 and F33615-93-1-1282.

Keywords: assembly planning, robot programming, task level programming, task model, c-surface, configuration space, planar quaternion.

Abstract

This report presents a general method to compute configuration space (c-space) obstacle surfaces (c-surfaces) in planar quaternion space and for projecting points onto them. We parameterize the general c-surface using the rotation angle and the vector of translation parameters of the individual contacts. Once we compute the domain of the rotation parameter, we can setup the translation parameters in a linear equation. The singular value decomposition of this equation gives us with the exact parameters of translation. We extend the theory to project a point in c-space onto the c-surface.

We implement our theory on the Assembly Plan from Observation (APO) system. The APO observes discrete instants of an assembly task and reconstructs the compliant motion plan employed in the task. We compute the contacts at each observed instant and the corresponding c-surface. We then interpolate the path on each c-surface to obtain segments of the path. The complete motion plan will be the concatenation of the connected path segments.

Table of Contents

1 Introduction

Consider three cases of an object in contact in a plane as shown in Figure 1. We are given an approximate configuration of the object with respect to its environmental objects. We can deduce the contacts made in each case as indicated by the ellipses in the figure. Using these contacts we want to describe the freedom of the object as shown by the parameters in bottom half of the figure. We would also like to correct the configuration in each case as shown in bottom of Figure 1.

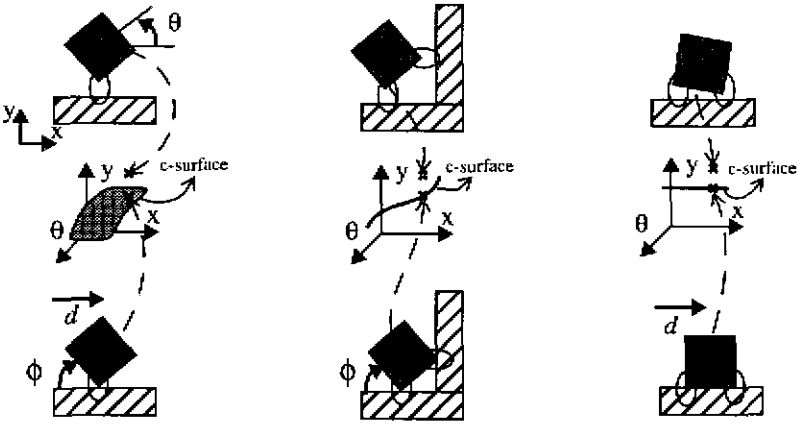

Figure 1 Examples of Contact Configurations

When an polygonal object maintains a set of contacts with another, its configuration in c-space will be constrained to a surface called the c-surface as shown in the center of Figure 1. This c-surface can be described in terms of a few independent parameters. The c-surface representation of the basic vertex-on-edge (*ve*) and edge-on-vertex (*ev*) contacts can be defined by a rotation parameter ϕ and a translation parameter d. For multiple contacts the c-surface will be the intersection of the c-surfaces of the individual contacts. Our work shows that the parameters of the intersecting surface can be expressed in terms of the parameters of the individual surfaces in a partly linear manner

Much work has been done in computing the c-surface in the field of motion planning. The representation of the c-surface is directly dependent on the representation of the c-space. The simplest c-space representing planar displacements is the (x,y,θ) space. Brost[2] and others[1] have studied c-surfaces in this space and derived the equations for them. The disadvantage of the (x,y,θ) space is that the space is not homogenous which makes the representation difficult. Another approach is to use planar quaternions to represent the c-space as was done by Ge and McCarthy[3]. The advantage of using planar quaternions in representing rigid body displacements is that all the group properties of displacements are preserved in it unlike the (x,y,θ) representation[8]. Thus the algebra of planar quaternions can be used to simplify the representation of the c-surfaces.

Our c-surface formulation is based on planar quaternion approach proposed by Ge and McCarthy. Unlike the algebraic method proposed by them to compute c-surfaces, we use the structure of the planar quaternion equation of the basic *ve* and *ev* contacts to make the prob-

lem quasi linear. The structure of the c-surface equations for the basic contacts is such that the rotation part of the quaternion equation is separate from the rest of the equation. We use of this characteristic to make the representation of the general c-surface quasi-linear. We choose the rotation parameter to be the same as the rotation angle of the object and the translation parameter to be the vector of the translation parameters of the individual contacts. This makes the translation part of the c-surface representation a linear matrix equation. The singular value decomposition (SVD) of this equation then provides us with the exact parametrization of the translation of the c-surface.

Once we have the representation of a c-surface corresponding to a set of contacts, we can compute the projection of a c-space point onto it. Again we exploit the separability of rotation to compute the projected point in a quasi-linear manner. First we project the c-space point onto the rotation space and obtain the values of the rotation parameters. Then we setup the translation parameters as a linear equation. The values of these translation parameters of the projection can then be obtained using least square minimization.

A major advantage of this quasi-linear approach is that it is possible to handle any number of contacts without having to enumerate all possible configurations. Another key benefit of the planar quaternion approach is that the theory is scalable to 3D space from the planar case by replacing planar quaternions with dual quaternions.

We implement our theory on the APO system. The system observes discrete instants of a human assembling an object and reconstructs the path used during the assembly. We compute the contacts at each observed scene using the geometrical models. Given the geometry of the features in contact we can find the c-surface corresponding to it and also the projection of the observed configuration on it. Given a series of discrete points lying on the same c-surface, we can interpolate a continuous path through them by using the parameters of the c-surface. The complete motion plan is then a series of connected path segments lying on adjacent c-surfaces of the c-space obstacle.

2 Related Work

Most of the related work is in the field of motion planning for robots. The main idea in all such work is the analytic computation of the configuration space obstacles given the geometry of the features in contact. One essential difference in these works is the type of representation of the configuration space.

Computing c-surfaces for polygonal objects in a plane in (x,y,θ) space was done by Brost[2]. This work elucidates all possible contact configurations and individually derives the equations for the c-surfaces corresponding to them. We used this work to implement the first version of the APO system. Unfortunately the work has no counterpart in 3D.

Ge and McCarthy[3][8] introduced four dimensional planar quaternions (z_1,z_2,z_3,z_4) to represent c-surfaces. Their work specifically deals with c-surfaces for a planar manipulator in contact with objects in the plane. The configuration space in their work involved the joint angles of the planar robot.

Our work uses the essentials of the contact representation of Ge and McCarthy. We use it to build the c-space obstacle surface corresponding to multiple contacts. While their work uses algebra to compute the c-surfaces, we use the characteristic of the contact equations to make the computation quasi linear.

Work by Suehiro[12] et al uses an iterative method to compute the correct configuration given a set of face-on-face contacts of an object with the environment. This work does not use the concept of c-surfaces.

3 Planar Quaternions

The displacement of an object in a plane can be represented by its translation parameters (x,y) and the rotation parameter θ. The (x,y,θ) space is a three dimensional projective space in the four dimensional planar quaternion space.

3.1 Representation of Displacement

The displacement of an object in a plane can be represented by its translation parameters (x,y) and the rotation parameter θ. The displacement can also be represented by the rotation angle θ about the pole (p_x, p_y) both of which are intrinsic properties of the displacement. The pole for the displacement (x,y,θ) is given by (1).

$$(p_x, p_y) = \left(\frac{\frac{x}{2}\cos\left(\frac{\theta}{2}\right) + \frac{y}{2}\sin\left(\frac{\theta}{2}\right)}{\sin\left(\frac{\theta}{2}\right)}, \frac{\frac{-x}{2}\sin\left(\frac{\theta}{2}\right) + \frac{y}{2}\cos\left(\frac{\theta}{2}\right)}{\sin\left(\frac{\theta}{2}\right)} \right) \tag{1}$$

The planar quaternion representing a displacement (x,y,θ) is a vector of four numbers, $Z = (z_1,z_2,z_3,z_4)$. It is assembled from the pole and rotation angle as given by (2)[8].

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} \frac{x}{2}\cos\left(\frac{\theta}{2}\right) + \frac{y}{2}\sin\left(\frac{\theta}{2}\right) \\ \frac{-x}{2}\sin\left(\frac{\theta}{2}\right) + \frac{y}{2}\cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) \\ \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{2}$$

Given a planar quaternion Z we can compute (x,y,θ) using (3).

$$\begin{aligned} \theta &= 2\,\mathrm{atan}(z_3, z_4) \\ x &= 2(z_1 z_4 - z_2 z_3) \\ y &= 2(z_1 z_3 + z_2 z_4) \end{aligned} \tag{3}$$

One of the advantages of using dual quaternions is the ease of composing transformations. The composition of two displacements represented by quaternions X and Y will be given by the quaternion product $Z = XY = [X^+]Y = [Y^-]X$, which is defined by a matrix product as shown in (4).

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} x_4 & -x_3 & x_2 & x_1 \\ x_3 & x_4 & -x_1 & x_2 \\ 0 & 0 & x_4 & x_3 \\ 0 & 0 & -x_3 & x_4 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} y_4 & y_3 & -y_2 & y_1 \\ -y_3 & y_4 & y_1 & y_2 \\ 0 & 0 & y_4 & y_3 \\ 0 & 0 & -y_3 & y_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (4)$$

We can see from (2), (3) and (4) that the rotation part of the planar quaternion is isolated from the rest of the quaternion. We will show how this characteristic can be taken advantage of when representing c-surfaces later.

3.2 Representation of Basic Contacts

For planar objects in contact there are two basic contacts, a vertex-on-edge (*ve*) contact and a edge-on-vertex (*ev*) contact as shown in Figure 2. All other contact configurations can be expressed as a combination of these basic contacts. The c-surface corresponding to basic contacts can be represented by a constraint equation in the planar quaternion space[3].

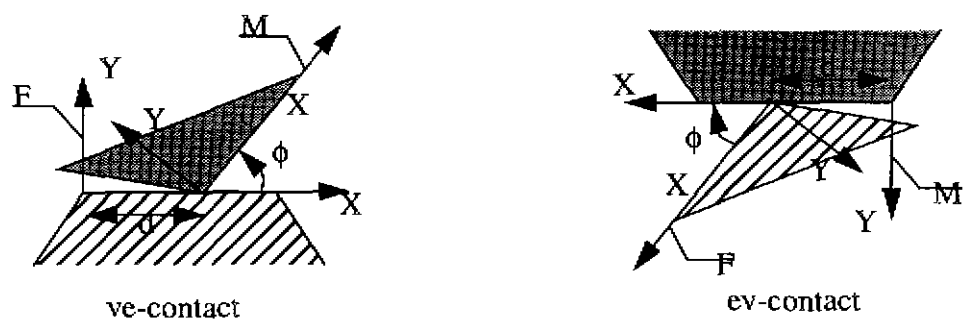

Figure 2 Basic contacts

Let the coordinate systems on the vertex and edge are defined as shown in Figure 2. The relative configurations of the two coordinate systems can be expressed in terms of two parameters ϕ and *d*. as given by (5)[3]. This is the c-surface of the contact. The equation is the same for both *ve* and *ev* contacts except for the variable ε which is *+1* for *ve* contact and *-1* for *ev* contacts

$$Y(\phi, d) = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} \frac{d}{2}\cos\left(\frac{\phi}{2}\right)\varepsilon \\ \frac{-d}{2}\sin\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix} \tag{5}$$

We can write the translation part of the planar quaternion in terms of the rotation part and the d parameter as a matrix equation as in (6).

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \left[D(y_3, y_4)\right] d = \begin{bmatrix} y_4 \\ -y_3 \end{bmatrix} [d] \tag{6}$$

3.2.1 Representation in World Coordinates

In real life the configuration of the objects are defined in terms of a world coordinate system as shown in Figure 3. $T_{bf} = (f_1,f_2,f_3,f_4)$ is the quaternion representing the transformation of the fixed feature with respect to the world coordinate system. $T_{ml} = (m_1,m_2,m_3,m_4)$ is the quaternion representing the transformation of the moving feature with respect to the object coordinates. $Y(\phi,d) = (y_1,y_2,y_3,y_4)$ is the quaternion representing the contact. The configuration of the moving object with respect to the world is $Z_{bl} = (z_1,z_2,z_3,z_4)$. The relation between the transformation of the objects are as shown in Figure 3,

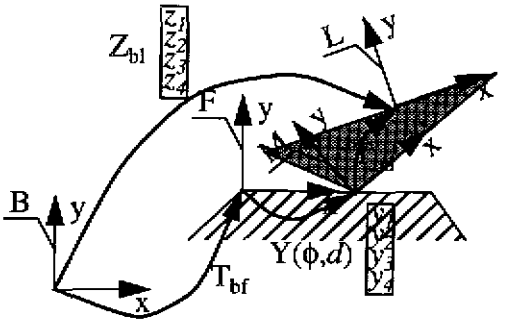

Figure 3 World Coordinates

We can express the configuration of the object in terms of the contact quaternion as shown in (7). The eight terms in the matrix $(t_1, t_2,...t_8)$ can be computed from the quaternions representing the transformations T_{bf} and T_{ml}, both of which are known from the geometry of the objects in contact.

$$Z_{bl}(\phi, d) = T_{bf} Y(\phi, d) T_{ml} = [T^{+}_{bf}][T^{-}_{ml}] Y(\phi, d)$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} t_1 & -t_2 & t_3 & t_4 \\ t_2 & t_1 & t_5 & t_6 \\ 0 & 0 & t_7 & t_8 \\ 0 & 0 & -t_8 & t_7 \end{bmatrix} \begin{bmatrix} \frac{d}{2}\cos\left(\frac{\phi}{2}\right)\epsilon \\ -\frac{d}{2}\cos\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix} = \begin{bmatrix} [C_{11}] & [B] \\ 0 & [C_{22}] \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \tag{7}$$

Once again we see that the rotation is separate from the rest of the quaternion. This makes it possible to split the constraint equation into two parts as shown in (8) and (9).

$$\begin{bmatrix} z_3 \\ z_4 \end{bmatrix} = [C_{22}] \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = [C_{22}] \begin{bmatrix} \sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = [C_{11}] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + [B] \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = [C_{11}] [D(y_3, y_4)] d + [B] \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} \tag{9}$$

3.2.2 Limits on Parameters

In order for the contact to be legal the two variables ϕ and d can vary only within limits. The limits for the variable ϕ are from $(0, \alpha_v)$ where α_v is the exterior angle of the vertex. The limits for the variable d are from $(0, l_e)$ where l_e is the length of the edge.

4 C-Surface Representation

We now formulate the general representation of c-surface for a set of *ve* of *ev* contacts. The c-surface corresponding to the set of contacts will be the intersection of the c-surfaces of the individual contacts. Generally, the number of parameters needed to define an intersection of c-surfaces is lesser than that of the component c-surfaces. We parameterize the general c-surface as follows. We use the rotation angle θ of the object as one parameter. We then make a vector of the parameter *d* of the individual c-surface, $(d_1, d_2,...)$. We will show how the constraints from the individual contacts can be used to constrain the elements of the *d* vector. We illustrate the idea by generalizing from the set of one and two contacts to multiple contacts.

4.1 One contact

When there is just a single contact, the c-surface is the same as the basic contact it is made of. The parameters are θ and *(d)*. The limits of θ can be obtained by mapping the limits of ϕ into θ using (3). The parameter *d* is the identical to and has the same limits as the *d* of the basic contact.

4.2 Two contacts

When there are two simultaneous contacts in a plane there can only be two possibilities as shown in Figure 4 and Figure 5.

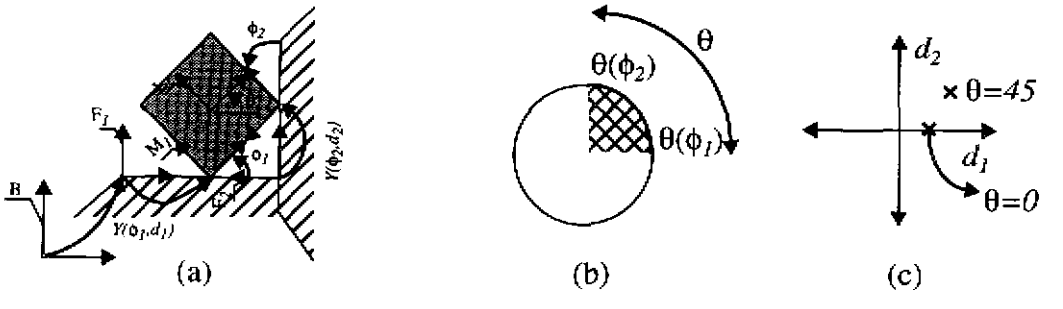

Figure 4 Example 1 of the two contact case

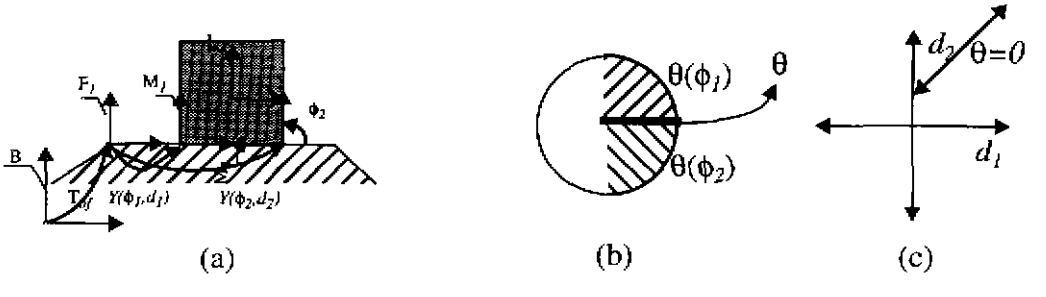

Figure 5 Example 2 of the two contact case

4.2.1 The θ Parameter

Using (8) and (3), we can compute the values of the rotation parameter θ for the two contacts ϕ_1 and ϕ_2 for any rotation in the configuration Z of the object as shown in (10) and (11).

$$\begin{bmatrix} z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} C^1{}_{22} \end{bmatrix} \begin{bmatrix} y_3^1 \\ y_4^1 \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} C^2{}_{22} \end{bmatrix} \begin{bmatrix} y_3^2 \\ y_4^2 \end{bmatrix} \tag{11}$$

But all values of θ may not be legal. The legal range of θ can then be computed by mapping the limits of ϕ_1 and ϕ_2 into a unit circle representing θ as shown in Figure 5(b). The legal range is the intersection of the two ranges of ϕ_1 and ϕ_2. In the case of the contacts shown in Figure 4 the range is 0 to 90. On the other hand the range for the contacts shown in Figure 5 is single valued at 0. In the former case the rotation is a variable parameter defining the c-surface. In the latter case the rotation parameter is constant.

4.2.2 The *d* Parameters

The *d* parameters d_1 *and* d_2 can be expressed in terms of the angle parameters as given by (12) and (13).

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} C_{11}^1 \end{bmatrix} \begin{bmatrix} D(y_3^1, y_4^1) \end{bmatrix} d^1 + \begin{bmatrix} B^1 \end{bmatrix} \begin{bmatrix} y_3^1 \\ y_4^1 \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} C_{11}^2 \end{bmatrix} \begin{bmatrix} D(y_3^2, y_4^2) \end{bmatrix} d^2 + \begin{bmatrix} B^2 \end{bmatrix} \begin{bmatrix} y_3^2 \\ y_4^2 \end{bmatrix} \tag{13}$$

Equating (12) and (13) we get the constraint equation involving the *d* parameters as given in (14). If we are given the geometry of the contacts and the rotation θ, The *d* parameters can be obtained by solving the linear equation of the form *AX=B*.

$$\left[\begin{bmatrix} C_{11}^1 \end{bmatrix} \begin{bmatrix} D(y_3^1, y_4^1) \end{bmatrix} - \begin{bmatrix} C_{11}^2 \end{bmatrix} \begin{bmatrix} D(y_3^2, y_4^2) \end{bmatrix} \right] \begin{bmatrix} d^1 \\ d^2 \end{bmatrix} = \begin{bmatrix} B^1 \end{bmatrix} \begin{bmatrix} y_3^1 \\ y_4^1 \end{bmatrix} - \begin{bmatrix} B^2 \end{bmatrix} \begin{bmatrix} y_3^2 \\ y_4^2 \end{bmatrix} \tag{14}$$

The solution *X* of this equation can be obtained using the singular value decomposition technique as given by (15).

$$\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} B \end{bmatrix} \qquad \begin{bmatrix} U \end{bmatrix} \begin{bmatrix} W \end{bmatrix} \begin{bmatrix} V^T \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} B \end{bmatrix}$$

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} V \end{bmatrix} \begin{bmatrix} W^{-1} \end{bmatrix} \begin{bmatrix} U^T \end{bmatrix} B + \sum_{(\forall j)(w_j = 0)} k_j \begin{bmatrix} v_{1j} \\ v_{2j} \end{bmatrix} \tag{15}$$

If A is non-singular as in the case of contacts in Figure 4 then there will be a unique X for every θ. In this case the ds are not independent parameters defining the c-surface. The more interesting case is when A is singular as in the case of Figure 5. In that case the columns of V corresponding to the singular values will correspond to a degree of freedom of the c-surface. The number of parameters will be the same as the number of singular values and the parameters defining the c-surface will be the variables k_j.

4.3 Multiple contacts.

We can easily generalize the two contact case to the case of N contacts. The parameters in this case will be θ and $(d_1, d_2, \ldots d_N)$.

4.3.1 The θ Parameter

As for the two contact case we compute the range of θ for every contact using the limits of ϕ_i for all N contacts using (16). We then intersects the ranges of θ to get the range of θ. If the range of θ is a single value, then the θ is not a parameter defining the c-surface. If it is not single valued then θ will be a parameter defining the c-surface.

$$\begin{bmatrix} z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} C^i_{22} \end{bmatrix} \begin{bmatrix} y^i_3 \\ y^i_4 \end{bmatrix} \qquad i = 1 \rightarrow N \tag{16}$$

4.3.2 The d Parameters

The individual d parameters, d_i can be expressed in terms of the angle parameters as shown in (17).

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} C^i_{11} \end{bmatrix} \begin{bmatrix} D(y^i_3, y^i_4) \end{bmatrix} d^i + \begin{bmatrix} B^i \end{bmatrix} \begin{bmatrix} y^i_3 \\ y^i_4 \end{bmatrix} = c^i d^i + b^i \qquad i = 1 \rightarrow N$$

$$c^i = \begin{bmatrix} C^i_{11} \end{bmatrix} \begin{bmatrix} D(y^i_3, y^i_4) \end{bmatrix} \qquad b^i = \begin{bmatrix} B^i \end{bmatrix} \begin{bmatrix} y^i_3 \\ y^i_4 \end{bmatrix} \tag{17}$$

Equating N pairs of contact equations (17) we can construct the constraint equation involv-

ing the d parameters as given in (18). If we are given the geometry of the contacts and the rotation θ, the d vector can be obtained by solving the linear equation of the form $AX=B$ as shown in (19).

$$\begin{bmatrix} c^1 & -c^2 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & c^i & -c^{i+1} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -c^1 & 0 & 0 & 0 & \dots & c^N \end{bmatrix} \begin{bmatrix} d^1 \\ \dots \\ d^i \\ \dots \\ d^N \end{bmatrix} = \begin{bmatrix} b^1 - b^2 \\ \dots \\ b^i - b^{i+1} \\ \dots \\ b^N - b^1 \end{bmatrix} \tag{18}$$

$$[A] \begin{bmatrix} d_1 \\ \dots \\ d_i \\ \dots \\ d_N \end{bmatrix} = [B] \qquad [U][W]\left[V^T\right] \begin{bmatrix} d_1 \\ \dots \\ d_i \\ \dots \\ d_N \end{bmatrix} = [B] \tag{19}$$

The complete solution X of this equation can be obtained using the singular value decomposition *(SVD)* technique[11] as given by (20). According to the definition of the *SVD*, the diagonal matrix W contains the singular values. The matrix $[W']^{-1}$ is obtained by replacing the non-zero diagonal elements with their reciprocals. The columns of the matrix U corresponding to the non-singular values spans the range of A. Whereas the columns of the matrix V corresponding to the singular values will span the null-space of A.

$$\begin{bmatrix} d_1 \\ \dots \\ d_i \\ \dots \\ d_N \end{bmatrix} = [V]\left[W'\right]^{-1}\left[U^T\right][B] + \sum_{(\forall j)(w_j = 0)} k_j \begin{bmatrix} v_{1j} \\ \dots \\ v_{ij} \\ \dots \\ v_{Nj} \end{bmatrix} = \begin{bmatrix} d'_1 \\ \dots \\ d'_i \\ \dots \\ d'_N \end{bmatrix} + \sum_{(\forall j)(w_j = 0)} k_j \begin{bmatrix} v_{1j} \\ \dots \\ v_{ij} \\ \dots \\ v_{Nj} \end{bmatrix} \tag{20}$$

If A is non-singular, then there will be a unique X for every θ. In this case the ds are not independent parameters. The more interesting case is when A is singular when the columns of V corresponding to the singular values will correspond to degrees of freedom of the c-surface. The number of parameters will be the same as the number of singular values and the parameters of the c-surface will be the variables k_j.

5 Projection onto C-Surface

Given any point in configuration space, its projection onto a c-surface will be the closest point on the c-surface. If the c-surface is represented using the parameters as explained in the previous section, the projection can be completely determined by the values of the parameters defining the c-surface. The projection can be done in two steps as was done with the representation. First we compute the value of θ for the point, then we project the point onto the d space. We illustrate the projection for the one contact and then for the multiple contacts.

5.1 One contact

5.1.1 Projection onto θ space

Given the configuration in c-space $Z = (z_1, z_2, z_3, z_4)$ we can use equation (8) to compute the parameter θ for the c-surface. We first map the limits of the contact angle ϕ to θ using equations (8) and (3). The closest value to this legal range is the angle of the projection.

5.1.2 Projection onto d space

Once we have θ computed, we substitute the corresponding (y_3, y_4) into (21).

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} C_{11} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} C_{11} \end{bmatrix} \begin{bmatrix} D(y_3, y_4) \end{bmatrix} d + \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} \tag{21}$$

We can set up the linear equation with d as the unknown as shown in (22).

$$\begin{bmatrix} C_{11} \end{bmatrix} \begin{bmatrix} D(y_3, y_4) \end{bmatrix} d = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} y_3 \\ y_4 \end{bmatrix} \tag{22}$$

Note that this is an over constrained equation in d. The value of d in the least square sense can be obtained by using *SVD*.

5.2 Multiple Contacts

An example of a projection of a given approximate configuration for a two contact case is shown in Figure 6. The projection is done by first projecting onto the θ space followed by the projection onto the d space. We directly proceed to the method of projection for the mul-

tiple contacts case.

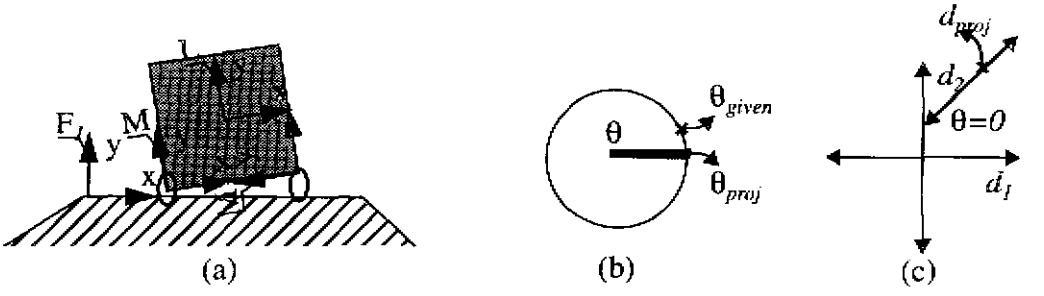

Figure 6 Projection for a two contact case

5.2.1 Projection onto θ space

We compute the legal range of the parameter θ of the c-surface as explained in the previous section for multiple contacts. We then find the closest value of the range to the given object rotation angle as shown in Figure 6(b).

5.2.2 Projection onto *d* space

We use the computed θ to obtain the (y_3^i, y_4^i) and substitute them into (23). The N equations are then summed into a single matrix equation as shown in (24).

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \left[C_{11}^i\right]\left[D(y_3^i, y_4^i)\right]d^i + \left[B^i\right]\begin{bmatrix} y_3^i \\ y_4^i \end{bmatrix} = c^i d^i + b^i \qquad i = 1 \rightarrow N$$

$$c^i = \left[C_{11}^i\right]\left[D(y_3^i, y_4^i)\right] \qquad b^i = \left[B^i\right]\begin{bmatrix} y_3^i \\ y_4^i \end{bmatrix} \tag{23}$$

$$\underbrace{\begin{bmatrix} c^1 & \cdots & c^i & \cdots & c^N \end{bmatrix}}_{C}\begin{bmatrix} d^1 \\ \cdots \\ d^i \\ \cdots \\ d^N \end{bmatrix} = \sum_{i=1}^{N}\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - b^i\right) \tag{24}$$

The d vector is already constrained by the c-surface as given by (20). We substitute this constrained d vector into (24) to obtain (25).

$$[C]\left(\begin{bmatrix} d_1' \\ \cdots \\ d_i' \\ \cdots \\ d_N' \end{bmatrix} + \sum_{(\forall j)(w_j = 0)} \begin{bmatrix} v_{1j} \\ \cdots \\ v_{ij} \\ \cdots \\ v_{Nj} \end{bmatrix}\right) = \sum_{i=1}^{N} \left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - \begin{bmatrix} B^i \end{bmatrix} \begin{bmatrix} y_3^i \\ y_4^i \end{bmatrix} \right) \tag{25}$$

Equation (25) can be written as a linear equation in k_j, as shown in (26) which can be solved in the least square sense (using *SVD*).

$$[C]\left[\cdots \begin{bmatrix} v_{1j} \\ \cdots \\ v_{ij} \\ \cdots \\ v_{Nj} \end{bmatrix} \cdots\right]_{(\forall j)(w_j = 0)} \begin{bmatrix} \cdots \\ k_j \\ \cdots \end{bmatrix} = \sum_{i=1}^{N} \left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} - \begin{bmatrix} B^i \end{bmatrix} \begin{bmatrix} y_3^i \\ y_4^i \end{bmatrix} \right) - [C] \begin{bmatrix} d_1' \\ \cdots \\ d_i' \\ \cdots \\ d_N' \end{bmatrix} \tag{26}$$

Once we get the values of θ and k_j, we can substitute them into (20) to obtain the values of d vector. This will completely define the projected point in configuration space.

6 Implementation

We implemented the theory developed in the previous sections on our Assembly Plan from Observation system which programs robots for assembly tasks by demonstrating the task. Automatic planning of robots for assembly typically involves the combinatorics of computing all possible contacts to build the configuration space (c-space) obstacle, and then searching for a feasible path in c-space. Unlike that approach, our system computes only the features of the c-space obstacle which are relevant to the observed assembly task. In addition, our system reconstructs a feasible path, instead of searching for one.

We use a real time vision system to record the assembly task. We then recognize and track the parts in each observed scene. The position and geometry of the objects are used to find the contacts between the objects in each observed scene. Using this, we identify the distinct contact configurations involved in the assembly task. Next, we analytically compute the feature on the c-space obstacle corresponding to each observed configuration. We then use the observed configurations to reconstruct the path segments lying on each computed c-space obstacle feature. These path segments constitute the model of the assembly task. Finally, we can use the modeled path to program a robot to repeat the assembly task.

6.1 Observation of the Assembly Task

The analysis of human assembly actions is based on the relations between the hand and the assembled parts at every instant of the assembly. Parts of the assembly task, such as during the compliant motion, consists of relatively small but significant motion. We capture this motion by recording the human assembly using a real-time stereo system [5]. The stereo images give us a sequence of dense and accurate 3D images of the scene. The assembled parts are then identified in the initial scene and tracked continuously through the whole sequence using a robust localization algorithm.

6.1.1 Localization and Tracking

We use a geometrical modeler called Vantage to build the models of the assembled parts. These models are then used to localize the parts and fingers in the scene using a 3D template matching algorithm called (3DTM) [13]. The 3DTM has features which make it well suited for localization in assembly scenes where occlusion and noise are inevitable.

The 3DTM can localize an object in a 3D scene, given a rough estimate of the location of the object in the scene. The algorithm uses sensor modeling, probabilistic hypothesis generation and robust localization techniques to make localization fast and accurate. We extend the localization algorithm from localizing one object in one image to tracking multiple objects in a series of images. We achieve tracking by using the previous locations of each object to compute the starting location for the next localization.

The pose of each object in a scene is given by (x,y,z,θ,ϕ,ψ) in a global reference frame as shown in Figure 7. The output of the localization is a list of poses of the objects in the scene.

The output of the tracking system is a list of list of poses of the objects being tracked.

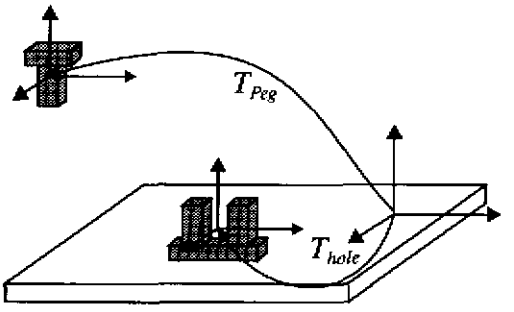

Figure 7 Assembly scene coordinates

The APO system uses the geometrical modeler called Vantage to create scenes of the task by instantiating models of the assembly objects at their observed poses. The sequence of observed poses of the peg during the peg in hole task is shown in Figure 8. The localized models of the peg and hole are superimposed as dark triangulated meshes on each of the observed intensity images.

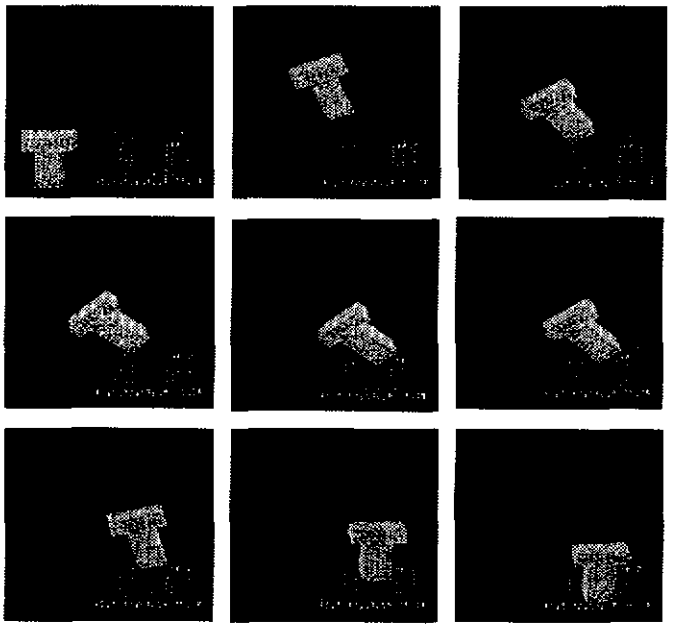

Figure 8 Object tracking in the observed scenes (only of the 70 scenes are shown here)

It should be noted that the raw pose obtained from the observation system is error prone. Despite this, we can extract the essential contact information in each observation, and use it to reconstruct the path.

6.2 Assembly Contacts

The key information in an observed scene of an assembly task is the contacts that are made between the assembled object and the environment. This involves identifying the feature pairs which are in contact.

Each feature pair in contact involves a feature of the assembled object (vertex and edge for polygonal objects) and a feature of the objects in the world as shown in Figure 9.

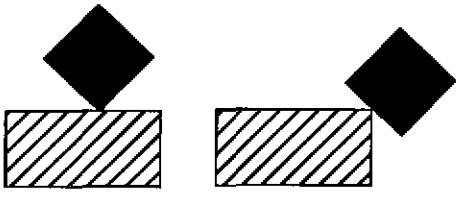

Figure 9 Types of contact pairs

Given the pose of the assembled object and the objects in the world, we can obtain the geometry of each feature of the assembled object and that of the other objects in the environment from the geometrical modeler, Vantage. We then check each feature pair for contact using the contact condition. For vertex-on-edge (*ve*) and edge-on-vertex (*ev*) contacts, if the vertex coordinates are (v_x, v_y) and the edge equation[1] is $e_a x + e_b y + e_c = 0$, then the primary condition for the pair to be in contact is given by (27).

$$e_a v_x + e_b v_y + e_c < \delta_{ve} \tag{27}$$

In addition, we check if the projection of the vertex lies within the end points of the edge.

All possible contact configurations can be expressed as a combination of these *ve* and *ev* contacts. Thus, for every observed configuration o_i, of the assembled object we end up with a set of feature pair sets in contact $\{c_{i1}, c_{i2}, \ldots\}$.

6.3 Observations to Motion Plan

The output of the observation system is list of observed configurations of the assembled object, $\{o_1, o_2, \ldots o_k\}$. The system instantiates the models of the assembled objects at each observed configuration, o_i. The next step is to find the feature pairs which are in contact for each o_i of the assembled object using the contact condition (27). Given the approximate configurations of the assembled objects and the set of contacts we can compute the c-surface C_i as explained in section 4.

We then segment the observed configurations $\{o_1, o_2, \ldots o_k\}$ into contiguous segments $\{S_1, S_2, \ldots S_n\}$ such that the c-surfaces for all observations in a segment S_j have the same c-surface

1. (e_a, e_b) is a 2d unit vector.

C_j.

Next we interpolate a continuous path through the projections of the observed points on each c-surface. The continuous path segment is obtained by interpolating in the parameter space. Any point on this curve will maintain the same contacts c_j. This will be the path segment PS_j. An example of this interpolation for a single ve contact in (θ,d) space is shown in Figure 10.

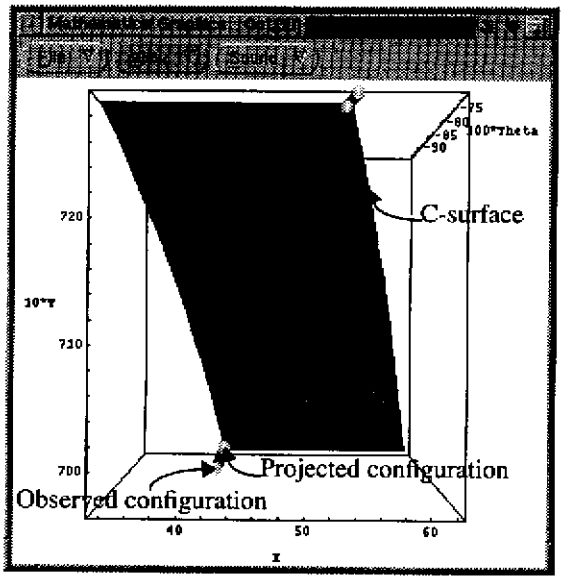

Figure 10 C-surface corresponding to a *ve* contact

The complete compliant motion path for the observed assembly task is then the concatenation of the path segments PS_i. The reconstructed path in c-space of the peg in the task observed in Figure 8 is shown in Figure 11.

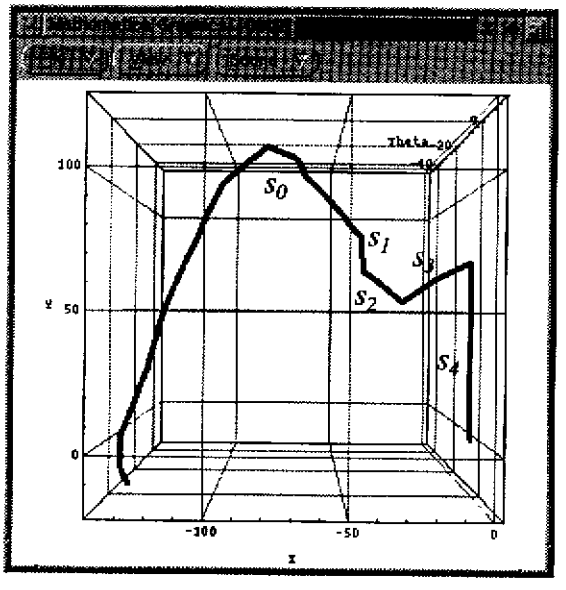

Figure 11 Path of the peg in configuration-space

The path segment s_0 shows the path the peg in free space. This is considered special because no contacts exist. This path usually avoids obstacles. Hence unlike other path segments, this path is interpolated between all observed configuration points. The path segment s_1 corresponds to the single *ve* contact that is made initially. The segment s_2 corresponds to the single *ev* contact following s_1. s_3 corresponds to two *ev* contact. Finally, s_4 corresponds to the multiple *ev* and *ve* contacts when the peg is in the hole. The modeled path in c-space will correspond to the motion of the assembled peg as shown in Figure 12.

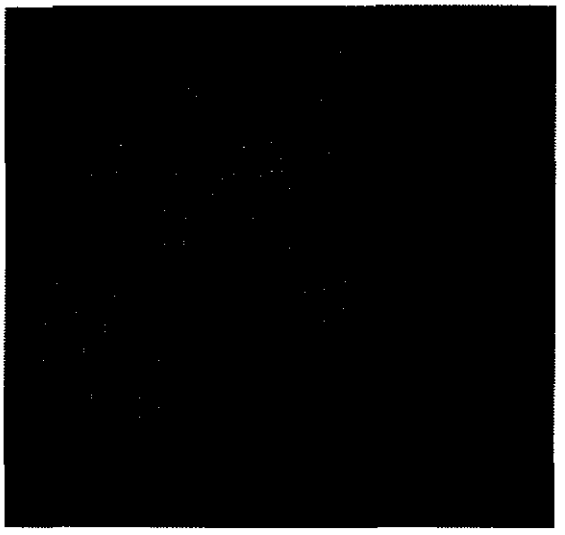

Figure 12 Path of the peg in Vantage

7 Conclusions

We have developed a general quasi-linear method for computing the c-surfaces for polygonal object in contact in planar quaternion space. We extended the theory to compute the projection of a point in c-space onto the c-surface.

We are able to implement the theory to model the compliant motion path of planar assembly tasks from observation. We reconstructed the path by first computing the c-surface and projection onto it. Then we computed the path segments lying on these c-surfaces. The connected path segments constitute the model of the observed assembly task.

We can scale the theory from the planar case to 3D space by replacing planar quaternions with dual quaternions. We plan to extend our APO system to model the assembly path of 3D polyhedral objects in 3D space.

Acknowledgments

We wish to thank Santiago E Conant for his help in converting VANTAGE to C++, Mark D Wheeler for his help with 3DTM, and Yunde Jiar for his help with the stereo system.

References

[1] Avnaim, F. Boissonnat, J.D, and Faverjon, B. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. *IEEE Int. Conf. on R & A* 1988, pages 1656-1661.

[2] Brost, R C. Analysis and Planning of Planar Manipulation Tasks. *Ph.D. Thesis*, CMU-CS-91-149, 1991, pages 96-103, 264-268.

[3] Ge, Q. and McCarthy, J. M. Equations for Boundaries of Joint Obstacles for Planar Robots. *IEEE Int Conf on R&A*, vol 1, 1989, pages 164-169.

[4] Inaba, M. and Inoue, H. Visual-based Robot Programming. *Int. Sym on Robotics Research*, 1989, pages 129-136.

[5] Kang, S.B, Webb, J.A, Zitnick, C.L and Kanade, T, An active multibaseline stereo system with real-time image acquisition, Tech. Rep. CMU-CS-94-167, 1994.

[6] Lozano-Perez, T. Automatic Planning of Manipulator Transfer Movements. *IEEE Trans. System Man and Cybernetics,* 1981, SMC-11(10): pages 681-689.

[7] Lozano-Perez, T. Mason, M. T. and Taylor, R. H. Automatic Synthesis of Fine Motion Strategies for Robots, *Int. Journal of Robotics Research*, 1984, vol.3, No 1, pages 3-24.

[8] McCarthy, J. M. An Introduction to Theoretical Kinematics, *The MIT Press*, 1990, Chapter 4, pages 53-65.

[9] Mason, M.T. "Compliance and force control for computer controlled manipulators" *IEEE Trans. Systems, Man and Cybernetics,* vol. 11, no. 6, 1981, pages 418-432.

[10] Paul, G.V. and Ikeuchi, K. Modeling Planar Assembly Paths from Observation. *IEEE Int Conf on Intelligent Robots & Systems,* Nov 1996, Vol 2, pages 520-525.

[11] Press, W.H. Flannery, B.P. Teukolsky, S.A. and Vetterling, W.T. Numerical Recipes in C. *Cambridge University Press,* 1990, pages 60-71.

[12] Suehiro, T. and Ikeuchi, K. Towards an Assembly Plan from Observation, part II. Correction of Motion Parameters Based on Face Contact Constraints. *IEEE Int Conf on Intelligent Robots & Systems,* Jul 1992.

[13] Wheeler, M.D and Ikeuchi, K. Sensor Modeling, Probabilistic Hypothesis Generation, and Robust Localization for Object Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence.* Vol 17, no 3, pages 252-265, Mar, 1995.