

## Modelling Human Assembly Actions from Observation

George V Paul, Yunde Jiar, Mark D Wheeler, and Katsushi Ikeuchi  
The Robotics Institute,  
Carnegie Mellon University,  
Pittsburgh, PA 15213.

### Abstract

*This paper describes a system which can model an assembly task performed by a human. The actions are recorded in real-time using a stereo system. The assembled objects and the fingers of the hand are tracked through the image sequence. We use the spatial relations between the fingers and the objects to temporally segment the task into approach, pre-manipulate, manipulate and depart phases. We interpret the actions in each segment broadly into grasp, push, fine-motion etc. We then analyze the contact relations between objects during the manipulate phase to reconstruct the fine motion path of the manipulated object. The fine motion in configuration space is a series of connected path segments lying on the features (c-surfaces) of the configuration space obstacle. We project the observed configurations onto these c-surfaces and reconstruct the path segments. The connected path segments form the fine motion path. We demonstrate the system using the peg in hole task.*

### 1 Introduction

Automatic planning of robots for assembly tasks has been attempted by a number of researchers [10][11]. This approach is computationally expensive. Another approach is to program robots for assembly tasks by demonstrating the task [4][14][16][9]. Our assembly plan from observation (APO) system is an example of the latter approach. The advantage of our system compared to automatic programming is that it has to only analyze the demonstrated feasible solution for accomplishing an assembly task, instead of computing it.

We use a real-time vision system to record the assembly task performed by the human operator. Models of the hand and the objects are tracked through the image sequence using a localization algorithm. The pose and geometry of the fingers of the hand and of the objects are then obtained in each scene of the task. We analyze this information in two levels. At the coarse level, we use the spatial relations between the hand and the objects to segment the sequence and isolate the manipulate phase. At the fine level, we then analyze the contacts between objects in the manipulate segment to obtain the fine motion of the assembly.

The assembly subtask which involves picking up a part and assembling it into another can be coarsely divided into four phases, approach, pre-manipulate, manipulate and depart. We identify these phases using the spatial relations between the fingers of the hand and the objects in each scene. After segmentation, we interpret the actions in each segment

broadly into common assembly actions such as grasping, fine-motion, screwing etc.

The most significant information in the manipulate segment of the subtask is the fine motion path of the manipulated object. This fine motion path in configuration space is a series of connected path segments lying on the c-space obstacle. We identify the distinct contact configurations between the manipulated object and the environment during the manipulate phase. We then analytically compute the c-surfaces. We use the observed configurations to reconstruct the exact path segments on each c-surface. These connected path segments constitute the fine motion path of the manipulated object.

This work demonstrates a system which can track multiple objects and the fingers of the hand in a recorded assembly sequence. The system can segment the actions of the human based on hand-object relations. The manipulate segment can then be analyzed further to reconstruct the fine motion of the subtask.

The following section explains the related work. Section 3 describes the tracking system which provides the data which is primary to this work. Section 4 explains the segmentation of the subtask based on hand-object relations. Section 5 explains the extraction of the exact fine motion path in the subtask.

### 2 Related Work

The analysis of human assembly actions covers three broad areas. The finger-finger relations of the hand for understanding the grasp, the finger-object relations for segmenting the task and understanding the grasp, and finally the object-object relations for obtaining the fine motion.

Work such as by Kang [8], Tung [16], Ogata [13] and Hashimoto [6] use the dataglove as the primary sensor to analyze finger-finger and finger-object relations while performing assembly tasks. The dataglove is an invasive sensor and has limited accuracy. This has limited such works to mostly pick and place tasks. Inaba [5], Kuniyoshi [9] and others have also used stereo vision systems to observe the task. But these systems are also limited to simple pick and place tasks. Our work uses more precise range data using a stereo system and a robust localization system, which allows us to interpret actions such as fine motion.

The object-object contacts is key to obtaining fine motion used in the assembly. Previous work on the APO system [4][14] used sparse observation of the assembly (end of each subtask). The motion plan was then derived from the

contact states. Our new approach extracts the fine motion by observing the **task** continuously. We accomplish this by adapting some of the results of the configuration space approach to automatic robot planning which was proposed by Lozano-Perez[10][11], Mason[12] and others. Work such as by Bajaj et al [2], Avnaim et al [1], and Brost [3] deal with computing the c-space obstacles analytically. We used Brost's results for our work.

### 3 The Observation System

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 fine motion, consists of relatively small but significant motion. We capture this motion by recording the human assembly using a real-time stereo system [7]. The stereo images give us a sequence of dense and accurate 3D images of the scene. The fingers of the hand and the assembled parts are then identified in the initial scene and tracked continuously through the whole sequence using a localization algorithm.

#### 3.1 Localization and Tracking

We use a geometrical modeler called Vantage to build the models of the fingers and 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) [17]. 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 modelling, 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 and previous velocity of each object to compute the starting location for the next localization.

The pose of each object in a scene is given by  $(x,y,z,\theta,\phi,\psi)$  in a world frame  $W$  as shown in Figure 1. 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 displacement of the objects being tracked as follows:

$$T = \left\{ \begin{array}{l} T_H^u(k), T_X^v(k) \end{array} \left| \begin{array}{l} k = 0, 1, 2, \dots, k_e \\ u = 0, 1, 2, \dots, U_H \\ v = 0, 1, 2, \dots, V_X \end{array} \right. \right\} \quad (1)$$

Where  $k$  is the frame number,  $u$  the finger index and  $v$  the object index.

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.

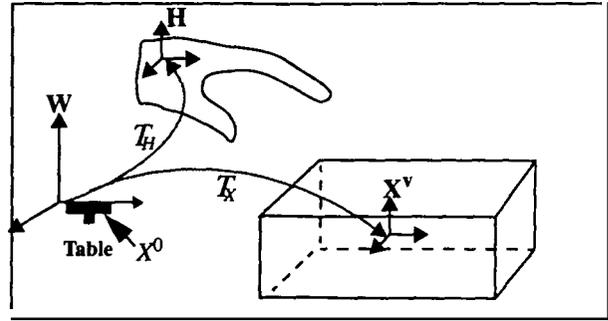


Figure 1 Assembly scene coordinates

The sequence of observed poses of the peg during the peg in hole task is shown in Figure 2. The localized models of the thumb and index finger are superimposed as lightly colored triangulated meshes on the intensity images of each scene. The models of the peg and hole are superimposed as dark triangulated meshes.

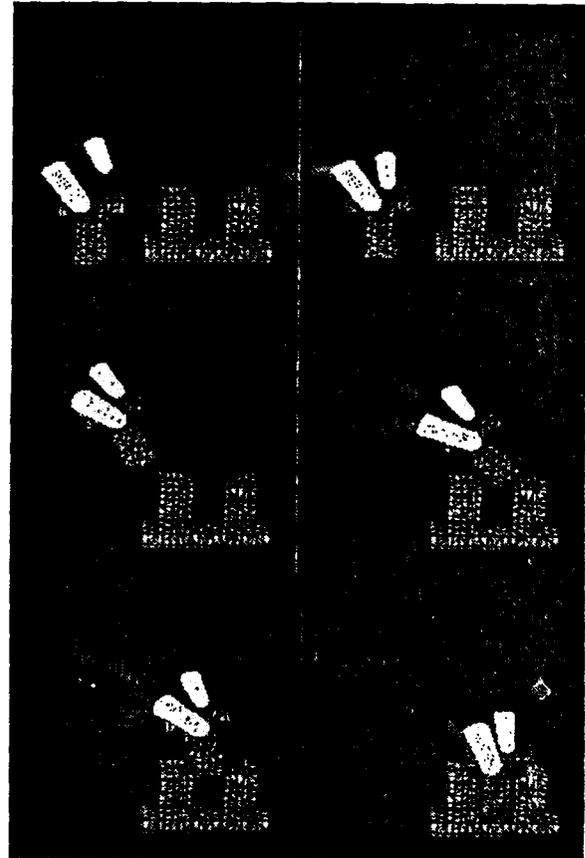


Figure 2 Object tracking in the observed scenes (only 6 of the 70 scenes are shown here)

### 4 Hand Action Interpretation

The action of the human hand while performing an assembly subtask can be segmented into four distinct phases[18]. These are, approach, pre-manipulate, manipulate

and depart as shown in the Figure 3. We identify the instants in the image sequence which delineate these phases. We then analyze each segment separately to interpret the actions performed in each phase. After task segmentation and action interpretation, the system generates a task sequence:

$$Task_{Subta} = \{ \langle Start, Subtask_w, End \rangle | (w = 1, 2, \dots, n) \}$$

$$Subtask_w = \langle Approach_w, Pre-Manipulate_w, Manipulate_w, Depart_w \rangle$$

The task sequence contains  $n$  subtasks, each subtask consisting of four actions, approach, pre-manipulate, manipulate, and depart.

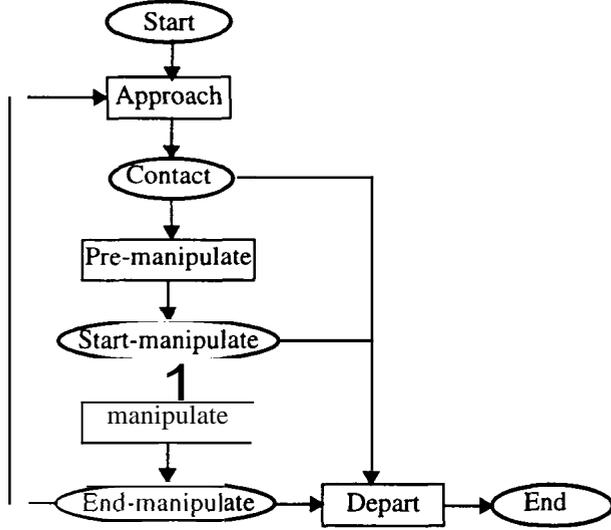


Figure 3 Assembly Subtask State Diagram

#### 4.1 Hand-Object Relations

We base the segmentation of each subtask on the spatial relationship between the fingers of the hand and the objects. We use *Vantage* to model each finger of the hand  $H^u$  and each object  $X^v$  used in the assembly. Thus the geometry of the features of both the hand and the objects are available to us as shown in Figure 4.

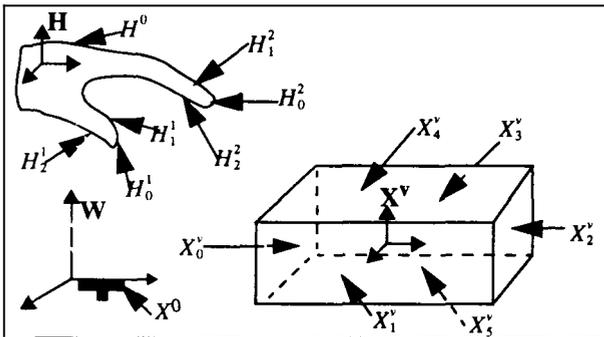


Figure 4 An example of an assembly task

The observation system provides us with the poses of the fingers of the hand and the objects in each scene. Using the geometry of the features, we define the distance between

a finger  $H^u$  and an object  $X^v$  in frame  $k$  as the nearest Euclidean distance.

$$R_{HX^v}^{distance}(k) = \min \left\{ \left\| \mathcal{H}_{ij}^u(k) - X_{st}^v(k) \right\| \right\} \quad (2)$$

where  $\mathcal{H}_{ij}^u$  and  $X_{st}^v$  are the surface point coordinates of the hand and the object with respect to the world frame W.

If  $R_{HX^v}^{distance}(k)$  is equal to zero, there is contact. The contact is completely defined by the set of features in contact as given by (3).

$$C_{H^u X^v} = \left\{ \left\| \mathcal{H}_{ij}^u(k) \right| R_{H^u X^v}^{distance}(k) = 0 \right\} \quad (3)$$

#### 4.2 Assembly Subtask Segmentation

The four phases of an assembly subtask can be obtained by analyzing the contact relations in each frame of the observed task. The three instants separating the phases are, contact, start-manipulate and end-manipulate.

##### Contact

The instant  $k_c$  is the frame at which at least one finger of the hand makes contact with an assembly object. The hand-object relation at this instant will be defined by (4). The contacted object  $X^v$  is now referred to as the manipulated object.

$$Contact_w = \left\{ \left( T_H(k_c^w) T_{X^v}(k_c^w) \right) \left( C_{H^u X^v}(k_c^w) \neq 0 \right) \wedge \left( C_{H^u X^v}(k) = 0 \right) \right\}_{k = k_c^w - 1, k_c^w - 2, \dots, k_{em}^w - 1} \quad (4)$$

##### Start-Manipulate

The frame  $k_{sm}$  at which manipulation starts, occurs when the manipulated object is about to be moved from its previous stable state. The hand-object relation at this instant is given by (5).

$$Start - Manipulate, = \left\{ \left( T_H(k_{sm}^w) T_{X^v}(k_{sm}^w) \right) \left( \dot{X}^u(k_{sm}^w) = \dot{X}^v(k_{sm}^w) \neq 0 \right) \wedge \left( \dot{X}^v(k) = 0 \right) \wedge \left( C_{H^u X^v}(k) \neq 0 \right) \right\}_{k = k_{sm}^w - 1, k_{sm}^w - 2, \dots, k_c^w} \quad (5)$$

##### End-Manipulate

The frame  $k_{em}$  at which the hand stops manipulating the object occurs when the hand is departing from the manipulated object. Theoretically there is no contact point between the hand and the object at and after this moment. The hand-object relation at this instant will be given by (6).

$$End - Manipulate_w = \left\{ \left( T_H(k_{em}^w) T_{X^v}(k_{em}^w) \right) \left( C_{H^u X^v}(k_{em}^w) = 0 \right) \wedge \left( C_{H^u X^v}(k) \neq 0 \right) \right\}_{k = k_{em}^w - 1, k_{em}^w - 2, \dots, k_{sm}^w} \quad (6)$$

### 4.3 Action interpretation

Once the object relations are used to compute the instants  $k_w$ ,  $k_c$  and  $k_{em}$ , the four segments, approach, pre-manipulate, manipulate and depart can be analyzed separately. The actions of the human in each segment can be broadly classified into common assembly actions such as grasp, push, pickup, screw etc., as explained below.

#### 4.3.1 Approach

The approach phase is defined by the actions between the starting frame to the frame  $k_c$  where there is contact between at least one finger  $H^u$  of the hand and the object.

$$Approach_w = \left\{ T_{H^u}(k) T_{X^v}(k) \mid k = k_{em}^{w-1}, k_{em}^{w-1} + 1, \dots, k_c^w - 1 \right\} \quad (7)$$

#### 4.3.2 Pre-manipulate

After making contact with the manipulated object at the end of the approach phase, we differentiate two types of relations between the hand and the manipulated object called touch (Figure 5a and b) and grasp (Figure 5c). The actions during the manipulate phase can then be analyzed using this differentiation.

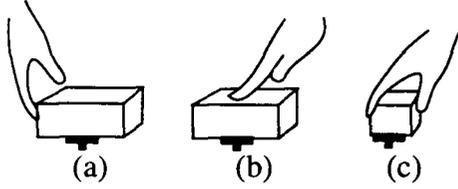


Figure 5 Touch and Grasp

##### (1) Touch

If only one finger or more fingers are aligned together touch an object, this can be a touch for a pushing action. In general, assume that there are  $n$  parts of the hand contacting the object with one contact unit normal  $n_{H^i}$ , the touch condition is defined by (8)

$$Touch_w = \left\{ \left( T_{H^u}(k), T_{X^v}(k) \right) \mid \forall A^i \geq 0, \sum A^i > 0 \Rightarrow \left( \sum A^i \cdot \vec{n}_{H^i} \neq 0 \right) \right. \quad (8)$$

$$\left. k = k_c^w, k_c^w + 1, \dots, k_{em}^w \quad i = v_1, v_2, \dots, v_n \right\}$$

##### (2) Grasp

On the other hand, when there are multiple contacts between the fingers and the object and the forces along the contact normal are at equilibrium, we define the action to be a grasp. The condition is defined by (9).

$$Touch_w = \left\{ \left( T_{H^u}(k), T_{X^v}(k) \right) \mid \exists A^i \geq 0, \sum A^i > 0 \wedge \left( \sum A^i \cdot \vec{n}_{H^i} = 0 \right) \right. \quad (9)$$

$$\left. k = k_c^w, k_c^w + 1, \dots, k_{em}^w \quad i = v_1, v_2, \dots, v_n \right\}$$

### 4.3.3 Manipulate

We interpret the actions in the manipulate phase (from frame  $k_{sm}$  to frame  $k_{em}$ ) into distinct assembly actions: push, push-pull, pickup and fine motion, screw and drop.

#### (1) Push

When the fingers of the hand touch the manipulated object while it maintains contact with the table, we define the manipulate action as a push. The formal definition is given in (10).

$$Push_w = \left\{ \left( T_{H^u}(k), T_{X^v}(k) \right) \mid \left( C_{X^0 X^1}(k) \neq 0 \right) \wedge Touch(k_{sm}^w) \right\} \quad (10)$$

#### (2) Push-Pull

When the fingers grasp the manipulate object while it slides on the table, we define the action to be push-pull.

$$Push - Pull_w = \left\{ \left( T_{H^u}(k), T_{X^v}(k) \right) \mid \left( C_{X^0 X^1}(k) \neq 0 \right) \wedge Grasp(k_{sm}^w) \quad \left( k = k_{sm}^w, k_{sm}^w + 1, \dots, k_{em}^w \right) \right\} \quad (11)$$

#### (3) Pick-Up and Fine Motion

When the hand grasps the manipulated object and moves it such that it makes contact with other assembled objects, we define the action to be pick-up and fine motion.

$$Pick - up = \left\{ \left( T_{H^u}(k), T_{X^v}(k) \right) \mid \left( C_{X^0 X^1}(k) = 0 \right) \wedge Grasp(k_{sm}^w) \quad k = k_{sm}^w, k_{sm}^w + 1, \dots, k_{em}^w \right\} \quad (12)$$

The detailed definition of push-pull, screwing and drop is described in [18].

## 4.4 Experimental Results

The result of action interpretation for a peg in hole task shown in Figure 1 is shown in Figure 6. The graph Figure 6(a) shows the minimum relative distance between the fingers of the hand and the peg as given by (1). The frame  $k$ , is 14. The approach phase is from frame 0-14. The graph Figure 6(c) shows the speed of the manipulated peg. This gives us start-manipulate instant  $k_{sm}$  as 24. The pre-manipulate phase is from frame 15-24. The end-manipulate instant  $k_{em}$

is 66. The manipulate phase is from frame 25-66. Finally the depart phase is from frame 67-70.

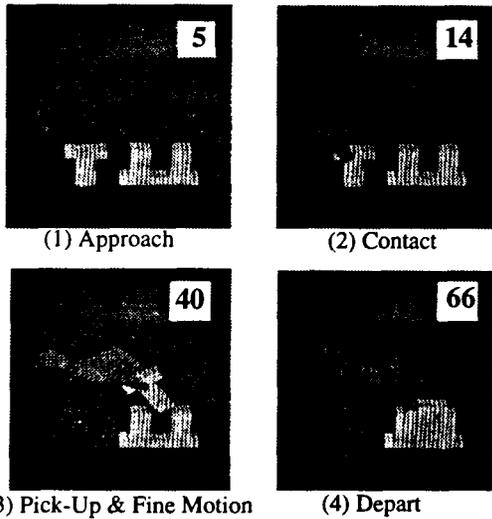
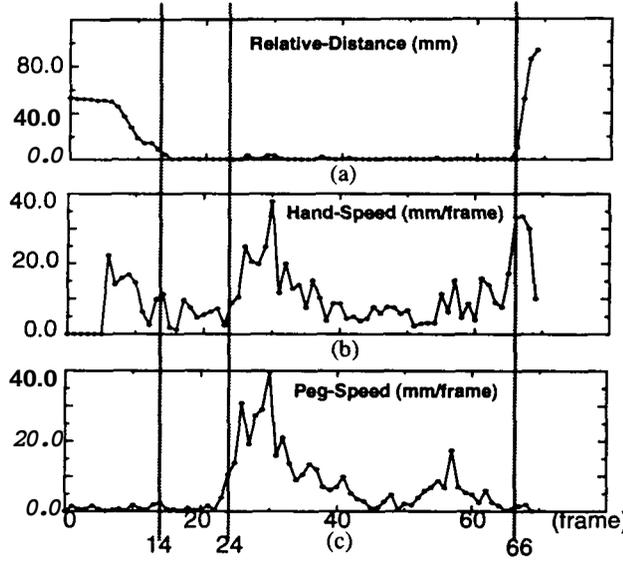


Figure 6 Segmentation of the Peg-in-Hole subtask

The action during the pre-manipulate phase was interpreted as a grasp. The action during the manipulate phase was interpreted as a pick-up and fine motion.

## 5 Fine Motion Modelling

The manipulate segment is the most significant part of an assembly task. The fine motion is the motion of the manipulated object in contact with the environment. We show how this fine motion path can be computed from the discrete observations of the scene.

Figure 7 illustrates the theory behind this analysis. Consider a series of observed 2D scenes of the dark rectangle coming into contact with a large rectangle as shown in Figure 7(a). If we consider only 2D translation, the c-space obstacle will be the shaded rectangle as shown in Figure 7(b). Instead of computing this complete c-space obstacle, we

find the contacts made in each observed scene and compute only the necessary features of the c-space obstacle,  $C_1$  and  $C_2$ , which correspond to the contacts. The observations can then be corrected by projecting onto their corresponding c-obstacle feature. We can also find the intersections between adjacent path segments explicitly. The fine motion path is then a series of path segments  $s_0, s_1$ , and  $s_2$  as shown in Figure 7(c).

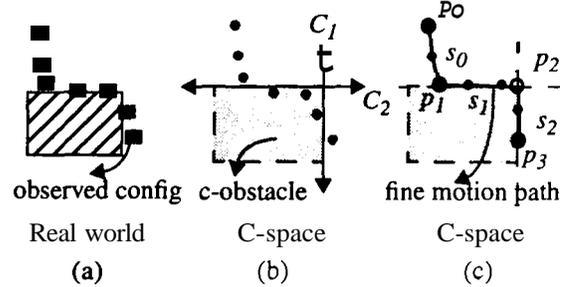


Figure 7 Path in c-space for 2D (translation only).

Work such as by Brost[3] analytically computes the c-surfaces for polygonal objects in a plane. We use these results by projecting the observed 3D scenes onto the Z plane.

## 5.1 Assembly Contacts

The key information in an observed scene in the manipulate phase of an assembly subtask is the contacts that are made between the manipulated object and the environment. From the geometry of the contacts, we can compute the c-surface. For polygonal objects in a plane, there are four types of feature pairs in contact as shown in Figure 8.

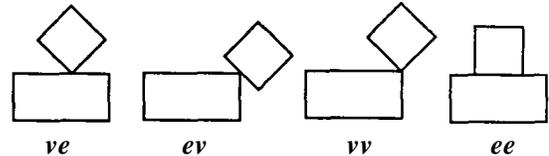


Figure 8 Types of contact pairs

Given the pose of the manipulated object and the objects in the world, we compute the feature pairs in contact using the following contact conditions.

For vertex-on-edge ( $ve$ ) and edge-on-vertex ( $ev$ ) contacts, if the vertex coordinates are  $(v_x, v_y)$  and the edge equation is  $e_a x + e_b y + e_c = 0$ , then the primary condition for the pair to be in contact is given by (13).

$$e_a v_x + e_b v_y + e_c < \delta_{ve} \quad (13)$$

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

For vertex-on-vertex ( $vv$ ) contacts if the two vertex coordinates are  $(v_x^1, v_y^1)$  and  $(v_x^2, v_y^2)$ , then the condition for the pair to be in contact will be given by (14).

$$\sqrt{(v_x^1 - v_x^2)^2 + (v_y^1 - v_y^2)^2} < \delta_{vv} \quad (14)$$

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

For edge-on-edge (ee) contacts if the edge equations are  $e_a^1 x + e_b^1 y + e_c^1 = 0$  and  $e_a^2 x + e_b^2 y + e_c^2 = 0$ , then the primary condition for the pair to be in contact will be given by (15).

$$1 - (e_a^1 e_a^2 + e_b^1 e_b^2) < \delta_{ee1} \wedge |e_c^1 - e_c^2| < \delta_{ee2} \quad (15)$$

In addition, for ee contacts we check if the edge segments overlap.

## 5.2 C-surface

By definition, the set of configurations of the manipulated object which maintains a set of feature pairs in contact is called the c-surface corresponding to the contacts. The order of this feature in c-space depends on the degree of contact. For example, a single **ve** contact results in a two dimensional c-surface in  $(x,y,\theta)$  space, whereas two simultaneous **ve** contacts correspond to a one dimensional edge in c-space.

For polygonal objects in a plane, there can be two types of two dimensional c-surfaces called facets, four types of one dimensional c-surfaces called edges and one zero dimensional c-surface called vertex. We briefly introduce the representation of a two dimensional c-surface corresponding to a single vertex-on-edge (**ve**) contact here. The representations of the all other possible contact configurations can be found in Brost's work [3].

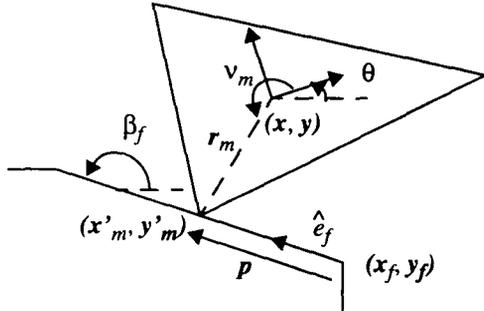


Figure 9 A single ve contact

Consider two polygons in contact in a plane as shown in Figure 9. The vertex of the moving object is in contact with the edge of the fixed object. The configuration of the moving triangle is  $(x,y,\theta)$ . The configuration obstacle feature corresponding to this contact will be a subset of an infinite two dimensional surface embedded in  $(x,y,\theta)$  space. This surface (c-surface) can be parameterized by two variables  $(p, \theta)$ . Brost defines  $p$  as a perimeter variable which is the distance of the contact vertex from the clockwise vertex of the contact edge as shown in Figure 9. The parameter  $\theta$  is identical to the angle of the moving object.

The infinite c-surface can then be described in parametric form by (16), (17) and (18). An example of the 2D c-surface corresponding to a **ve** contact is shown in Figure 10.

$$x = f_x(p, \theta) = k_{1x} + k_{2x}p + k_{3x} \cos(\theta + \kappa_{3x}) \quad (16)$$

$$k_{1x} = x_f \quad k_{2x} = \cos(\beta_f) \quad k_{3x} = -r_m \quad \kappa_{3x} = v_m$$

$$y = f_y(p, \theta) = k_{1y} + k_{2y}p + k_{3y} \sin(\theta + \kappa_{3y}) \quad (17)$$

$$k_{1y} = y_f \quad k_{2y} = \sin(\beta_f) \quad k_{3y} = -r_m \quad \kappa_{3y} = v_m$$

$$\theta = f_\theta(p, \theta) = \theta \quad (18)$$

There exists an inverse mapping of a configuration space point in  $(x,y,\theta)$  space to a point on the c-surface defined by the  $(p, \theta)$  variables. The form of these equations are given by (19) and (20).

$$p = f_p(x, y, \theta) = \overrightarrow{e_{cw}} \cdot \hat{e}_f$$

$$\overrightarrow{e_{cw}} = \begin{bmatrix} (x + r_m \cos(\theta + v_m) - x_f) \\ (y + r_m \sin(\theta + v_m) - y_f) \end{bmatrix} \quad (19)$$

$$\theta = f_\theta(x, y, \theta) = \theta \quad (20)$$

Given any configuration space point  $(x,y,\theta)$  the equations (19) and (20) will find the  $(p, \theta)$  of the closest point which lies on the c-surface. We use this property to correct the raw configurations obtained from the observation system.

The counterparts of the equations (16)-(18) and (19)-(20) exist for the **ev** facets and the four classes of edges. These are derived in [3].

## 5.3 Path Segments

If the observed configurations  $S_j = \{o_{j_1}, o_{j_2}, \dots, o_{j_k}\}$  correspond to the particular contact configuration, we can compute the c-surface  $C_j$  corresponding to it. We project the observation onto the surface and interpolate a continuous path segment on the surface in  $(p,\theta)$  space as shown in Figure 10.

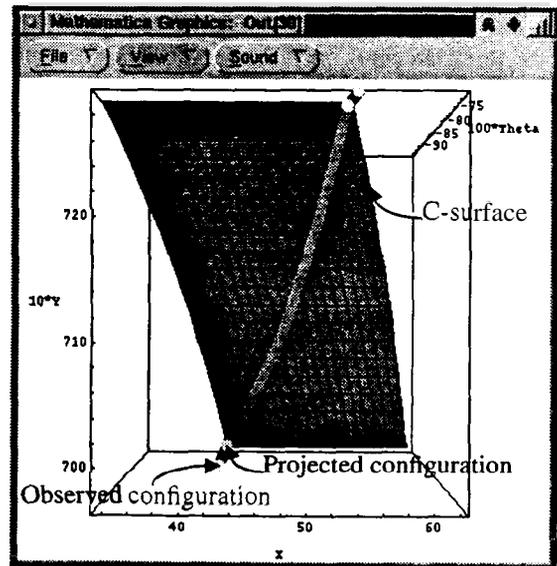


Figure 10 C-surface corresponding to a ve contact

The complete fine motion path is a series of connected path segments lying on adjacent c-surfaces. The start point of a path segment on  $C_i$  lies on the intersection of  $C_i$  and  $C_{i-1}$ . The end point of the same path segment will lie on the intersection of  $C_i$  and  $C_{i+1}$ . These points are critical points in the path. We compute these start and end points of each path segment explicitly using a novel numerical technique, which works for all combinations intersecting c-surfaces.

The technique can be illustrated by considering a simple 2D case as shown in Figure 11. Let the last observation corresponding to the c-surface  $C_i$  be  $o_i$ , and let  $C_{i+1}$  be the c-surface corresponding to the next contact configuration. We want to compute the intersection  $q$  of the two c-surfaces  $C_i$  and  $C_{i+1}$  closest to  $o_i$ .

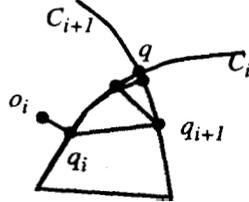


Figure 11 Intersection of two path segments

We find the intersection by successively projecting the observed point onto  $C_i$  and the  $C_{i+1}$  until the point converges. The procedure is as follows.

1. Project  $o_i$  onto  $C_i$  using equations (19)-(20) and obtain  $(p_i, \theta_i)$ . Convert  $(p_i, \theta_i)$  back to  $(x_i, y_i, \theta_i)$  using equations (16)-(18). This is the projected point  $q_i$ .
2. Project  $q_i$  onto  $C_{i+1}$  using equations (19)-(20) and obtain  $(p_{i+1}, \theta_{i+1})$ . Convert  $(p_{i+1}, \theta_{i+1})$  back to  $(x_{i+1}, y_{i+1}, \theta_{i+1})$  using equations (16)-(18). This is the projected point  $q_{i+1}$ .
3. If  $|q_i - q_{i+1}| < \delta$  (threshold), return  $q_{i+1}$ ; else set  $o_i = q_{i+1}$  and go to step 1.

The procedure converges because the initial configuration  $o_i$  is close to the intersection point of two c-surfaces whose intersection exists. An advantage of this technique is that it can find intersections of path segments lying on different classes of c-surfaces.

## 5.4 Implementation

There are three main steps of the algorithm which converts the observed poses of the assembled object to the fine motion path. The Algorithm is as follows:

- Instantiate the models of the assembled objects at each observation,  $o_i$ . Find the feature pairs which are in contact for each  $o_i$  of the assembled object with the environment as explained in Section 5.1. Let the contact feature pair set for the observed configuration  $o_i$  be  $FP_i$ .

- Compute the unique c-space obstacle feature  $C_i$  using equations (16), (17) and (18) corresponding to each observation  $o_i$  using the feature pair set  $FP$ , as explained in section 5.2.

- Segment the observed configurations  $\{o_1, o_2, \dots, o_k\}$  into contiguous segments  $\{S_1, S_2, \dots, S_n\}$  such that all observations in a segment  $S_j$  maintain the same contact pair set  $c$  and have the same c-surface.

$$S_j = \{o_{j_1}, o_{j_2}, \dots, o_{j_k}\} \quad (21)$$

$$FP_{j_1} \cap FP_{j_2} \cap \dots \cap FP_{j_k} = c$$

- Compute the path segment  $PS_j$  lying on the c-space obstacle feature  $C_j$  through the observed configurations in the segment  $S_j = \{o_{j_1}, o_{j_2}, \dots, o_{j_k}\}$  as explained in section 5.3.

The complete assembly motion path is then the concatenation of the path segments  $PS_i$ . The reconstructed path in configuration space of the peg in the task observed in Figure 2 is shown in Figure 12.

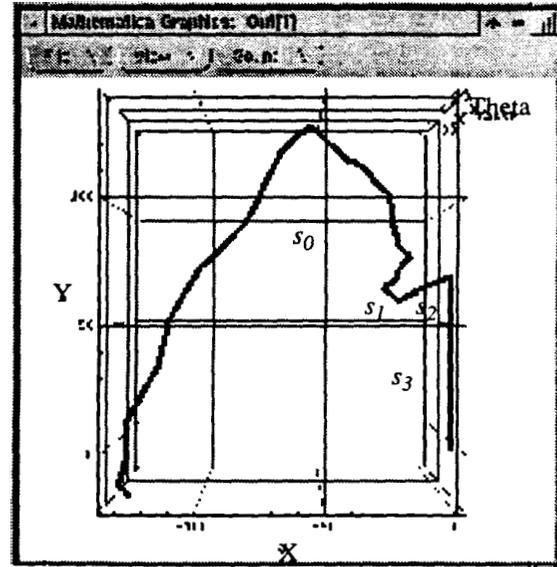


Figure 12 Path of the peg in configuration-space

The path segment  $s_0$  in shows the path the peg in free space. This is considered special because no contacts exist and the feature corresponding to this state is the whole 3D space  $(x, y, \theta)$ . 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  $ev$  contact that is made initially.  $s_2$  corresponds to the  $ev-ev$  contact. Finally,  $s_3$  corresponds to the multiple  $ee$  contacts when the peg is in the hole. The

modelled path in c-space will correspond to the motion of the assembled peg as shown in Figure 13.

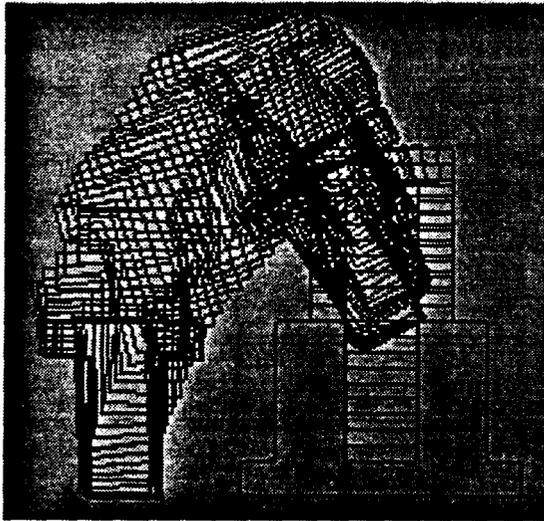


Figure 13 Path of the peg in the assembly task

## 6 Conclusions

We have demonstrated a system which can observe the a human performing an assembly and reconstruct the assembly actions. The reconstruction is based on analyzing the observations at two levels. At the coarse level, we temporally segment the assembly based on hand-object relations. The actions in each segment can be interpreted into categories such as grasp, push, manipulate etc. At the fine level, the object-object contact relations in the manipulate segment can be further analyzed to reconstruct the fine motion path.

We plan to extend the system to model fine assembly paths of 3D polyhedral objects in the 3D space.

## Acknowledgments

This research was funded in part by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), US Air Force, Wright Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465 AEPA Order No 7597 and in part by NSF under Contract CDC 31211797.

## 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] Bajaj, C. and Kim, M. Generation of Configuration Space Obstacles: Moving Algebraic Surfaces. *Int. Journal of Robotics Research*, 1990, vol.9, No 1, pages 92-112.
- [3] Brost, R. C. Analysis and Planning of Planar Manipulation Tasks. *Ph.D. Thesis*, CMU-CS-91-149, 1991, pages 96-103, 264-268.
- [4] Ikeuchi, K. and Suehiro, T. Towards an Assembly Plan from Observation, part I: Assembly Task Recognition with Polyhedral Objects. *IEEE Trans on R&A*, vol 10, no 3, 1994, pages 368-385.

- [5] Inaba, M. and Inoue, H. Visual-based Robot Programming. *Int Sym on Robotics Research*. 1989, pages 129-136.
- [6] Hashimoto, H and Buss, M, Skill Acquisition for the Intelligent Assisting System using Virtual Reality Simulator. *Annual Conf of IEEE Industrial Electronics Society*, 1983, pages 12-58.
- [7] 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.
- [8] Kang, S.B, Robot Instruction by Human Demonstration. Ph.D. Thesis, Tech. Rep. CMU-RI-94-167, 1994.
- [9] Kuniyoshi, Y, Inaba, M, and Inoue, H, Learning by Watching: Extracting Reusable Task Knowledge from Visual Observation of Human Performance. *IEEE Trans on R&A*, Vol 10, No 6, 1994, pages 799-822.
- [10] Lozano-Perez, T. Automatic Planning of Manipulator Transfer Movements. *IEEE Trans. System Man and Cybernetics*, 1981, SMC-11(10): pages 681-689.
- [11] 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.
- [12] 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.
- [13] Ogata, H. and Takahashi, T. "Robotic Assembly Operation Teaching in a Virtual Environment" *IEEE Trans. on Robotics and Automation*, vol. 10, no. 3, 1994, pages 391-399.
- [14] Paul, G.V. and Ikeuchi, K. Modelling Planar Assembly Tasks: Representation and Recognition. *IEEE Int Conf on Intelligent Robots & Systems*, Aug 1995. Vol 1, pages 17-22.
- [15] 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.
- [16] Tung, C.P. and Kak, A. Automatic Learning of Assembly Tasks Using a DataGlove System. *IEEE Int Conf on Intelligent Robots & Systems*, Aug 1995, Vol 1, pages 1-8.
- [17] Wheeler, M.D and Ikeuchi, K. Sensor Modelling, 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.
- [18] Jiar, Y, Wheeler, M and Ikeuchi, K. Hand Action Perception for Automatic Robot Programming by Demonstration. CMU Tech Report. CMU-CS-96-116., Mar, 1996.