Assembly Task Recognition with Planar, Curved and Mechanical Contacts*

Katsushi Ikeuchi Masato Kawade Takashi Suehiro

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Currently, most robot programming is done either by manual programming or by the "teach-by-showing" method using a teach pendant. Both of these methods have been found to have several drawbacks.

We have been developing a novel method for programming a robot: the assembly-plan-from-observation (APO) method. The APO method aims to build a system that has threefold capabilities. It observes a human performing an assembly task, it understands the task based on this observation, and it generates a robot program to achieve the same task.

This paper concentrates on the APO's main loop, which is task recognition. Using object recognition results, the task recognition module determines what kind of assembly task is performed. The previous system, on which we reported previously, recognizes assembly tasks which only handles polyhedral objects. The system reported in this paper extends the task recognition module to handling curved objects and other mechanical contacts. We will define task models for these cases and then verify two concepts: such task models are useful in recognizing assembly tasks, and it is possible to generate robot motion commands for repeating the same assembly task.

1 Introduction

Several method for programming a robot have been proposed. Such methods include the following: teach-by-showing, teleoperation [4], textual programming, and automatic programming [7]. Among these four representative methods, teleoperation and automatic programming are the most promising. Yet, these methods are often inconvenient and impractical.

We have been developing a novel method which combines automatic programming and teleoperation. We intend to add a vision capability, which can observe human operations, to an automatic programming system. We will refer to this paradigm as Assembly-Plan-from-Observation (APO). Several other researchers have also been developing systems towards similar goals, such as those by Kuniyoshi et.al. [6] and Takahashi et.al. [11].

In our APO paradigm, a human operator performs assembly tasks in front of a video camera. The system obtains a continuous sequence of images from the camera which is recording the assembly tasks. In order for the system to recognize assembly tasks from the sequence of images, the system has to perform the following six operations:

- *Temporal Segmentation* - dividing the continuous sequence of images into meaningful segments which correspond to separate human assembly tasks,
- *Object Recognition* - recognizing the objects and determining the object configurations in a given image segment.
- *Task Recognition* - recognizing assembly tasks by using the results of an object recognition system.
- *Grasp Recognition* - recognizing where and how the human operator grasps an object for achieving the assembly task.
- *Global Path Recognition* - recognizing the path along which the human operator moves an object while avoiding collision.
- *Task Instantiation* - collecting necessary parameters from the object recognition operation, grasp recognition operation, and global path recognition operation allows us to develop assembly plans to perform the same task using a robot manipulator.

In this paper, we will concentrate on the task recognition and task instantiation modules, because these two modules form the main loop of the assembly plan from observation method.

The outline of the modules are as follows: Using the object models from a given image segment, our object recognition module identifies each object. The module represents the recognition results in a world model by using the geometric modeler, Vantage [2].

Our task recognition module recognizes object relations in two world models (pre and post assembly task) and extracts the transition between two object relations. The task recognition module has *abstract task models* in a data base. Each abstract task in the data base associates a transition between two object relations with an assembly task which causes this transition. Among the task models in the data base, the system identifies a task model that describes the

---

*This research was sponsored in part by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, Ohio 45433-6543 under Contract F33615-90-C-1465 Order No. 7597, and in part by National Science Foundation, under Contract CDC-9121797.

1050-4729/93 $3.00

current transition and, thus, determines the current assembly task.

Our task instantiation module represents the recognition result as an instantiated task model. An instantiated task model includes finding the appropriate motion parameters for the given scenes. Such motion parameters include object locations and grasping locations for the action. These parameters are converted from the current object configurations in the current world model. The instantiated task model also includes the global path along which to move an object. The system, then, inserts these motion parameters in the command sequence. Finally, the command sequence is sent to the robot.

The central issue in task recognition is the type of representations which will be used for describing an assembly task. The main purpose of an assembly task is to put together two separate parts into one subassembled part and to establish one particular class of surface contact relation. Thus, we have decided to use surface contact relations as the basic representation in the system.

In a previous paper, we constructed abstract task models based on planar face contacts. The main limitation was its inability to handle anything other than polyhedral objects. However, in order to apply our APO paradigm to general assembly operations, the system should be able to handle other types of objects which consist not only of planar faces but also of curved surfaces. This paper will extend our abstract task models to include curved surfaces.

Section 2 will define ten assembly relations. Section 3 will extend our task models to handling curved surfaces based on the assembly relations. Section 4 will discuss how to use task models in the task recognition loop. Section 5 will conclude this paper.

2 Defining Assembly Relations

2.1 Assembly relations

In order to develop abstract task models, we have to define representations to describe assembly tasks. This section will define assembly relations for such representations.

The primal goal of an assembly task is to establish a new surface contact relationship among objects. For example, the goal of peg-insertion is to achieve surface contacts between the side and bottom surfaces of the peg and the side and bottom surfaces of the hole. Thus, it is effective to use surface contact relations as the central representation for defining assembly task models.

In each assembly task, at least one object is manipulated. We will refer to that object as the *manipulated* object. The manipulated object is attached to other stationary objects, which we refer to as the *environmental* objects, so that the manipulated object achieves a particular relationship with the environmental objects.

We will define *assembly relations* as surface contact relations between a manipulated object and its stationary environmental objects. Note that we do not exhaustively consider all of the possible surface contact relations between all of the objects; this would result in a combinatorial explosion of possibilities. We can avoid the exponential complexity by concentrating on a select group of surface contacts, namely, those that occur between the manipulated object and the environmental objects.

2.2 Constraints given by an infinitesimal patch pair

When considering possible contact relations, we mainly take into account the kinds of translation operations that are necessary for achieving these relations.

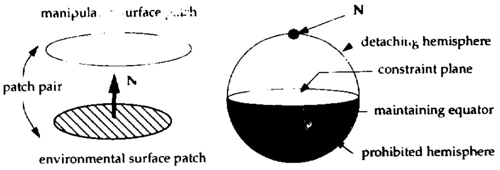

Figure 1: Constraint inequality depicted on the Gaussian sphere

Let us suppose that an infinitesimal surface patch of the manipulated object has a surface contact with an infinitesimal surface patch of an environmental object. This surface contact pair constrains the manipulated object's possible translation motion by:

$$\mathbf{N} \cdot \Delta T \geq 0, \tag{1}$$

where **N** denotes the normal direction of an environmental surface patch and ΔT denotes possible translational motion vectors of the manipulated object.

We use points on the Gaussian sphere to specify both a constraint vector and all possible translation vectors. Each vector is translated so that its starting point is located at the center of the Gaussian sphere and its end point exists at some point on the surface of the Gaussian sphere. This point is unique to the vector. We use this point to denote the vector. We can assume that the constraint normal, **N**, points to the north pole of the Gaussian sphere without loss of generality; the normal is represented as the north pole of the Gaussian sphere.

The plane perpendicular to the normal, **N**, divides the Gaussian sphere into two hemispheres. The points on the northern hemisphere satisfy the strict inequality, $\mathbf{N} \cdot \Delta T > 0$ and denote the motion vectors which break the face contact of the surface pair. The points on the southern hemisphere satisfy $\mathbf{N} \cdot \Delta T < 0$ and correspond to prohibited motions which make the manipulated object run into an environmental object. The points on the equator satisfy $\mathbf{N} \cdot \Delta T = 0$; the corresponding motions of the manipulated surface patch maintain the face contact. See Figure 1 for the definitions.

2.3 Taxonomy of assembly relation

Each assembly relation consists of several surface patches of different orientations. Since each different orientation provides a linear inequality, the resulting possible motion directions of an assembly relation are constrained through simultaneous linear inequalities. The possible motion directions are depicted as a region on the Gaussian sphere, which we refer to as an admissible region of the assembly relation.

Admissible regions have various shapes on the Gaussian sphere. By grouping admissible regions based on their shapes, we can establish ten distinct patterns of admissible regions, and thus ten representative assembly relations. These ten relations consists of: entire sphere (3d-s), hemisphere region (3d-a), crescent region (3d-c), m convex polygonal region (3d-f), a whole arc of a great circle (3d-b), a half arc of a great circle (3d-d), a partial arc of a great circle (3d-g), a pair of polar points (3d-e), one point (3d-h), and null region (3d-i). Figure 2 shows these ten patterns.

We can classify any nth directional assembly relation into one of the ten representative assembly relations. Note that since we consider inequalities between pairs of infinitesimal surface patches, we can utilize this taxonomy not only for the analysis of polyhedral objects but also for the analysis of curved objects.

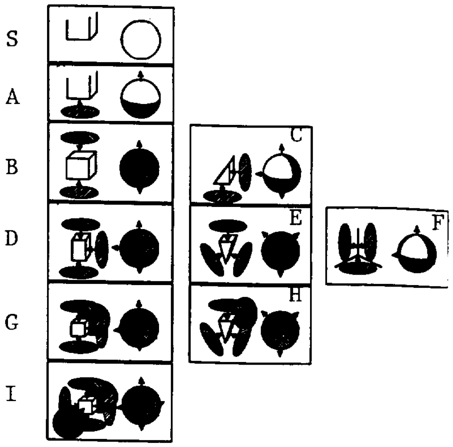

Figure 2: Ten assembly relations

3 Abstract task model

Basically, an abstract task model associates an assembly relation transition with an action which causes such a transition. In this section, we will extract what kind of transition occurs within the assembly relation taxonomy in Figure 2. We will conduct this analysis by considering possible disassembly operations [5].

3.1 Planar face contact

By considering disassembly operations between manipulated and environmental polyhedra, we have obtained the assembly relation transition directional graph shown in Figure 3. See [5]. By assigning an appropriate motion template to each arc of the graph, we have developed abstract task models for polyhedra.

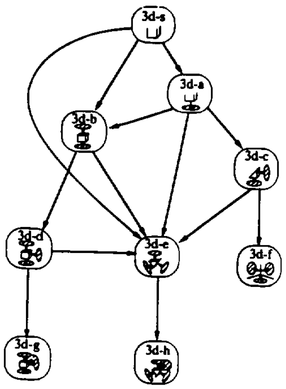

Figure 3: Assembly relation transition graph for planar faces

3.2 Curved surface contact

The assembly relation taxonomy is defined between infinitesimal surface patches. We can apply the same taxonomy to the analysis of curved surface contacts. There is no difference between planar surface and curved surface contacts in terms of the classification of assembly relations.

On the other hand, a difference occurs in applicable control strategies to establish such assembly relations. For example, in the planar faces, we can achieve 3d-c assembly relation via 3d-a assembly relation; we achieve a 3d-a assembly relation by a move-to-contact operation, and then 3d-c by another move-to-contact operation as shown in Figure 3. However, for example, to achieve a 3d-c assembly relation between a negative and positive cylindrical surface, we cannot use such a two-step strategy; we have to execute a direct move-to-contact operation from a given approach direction.

In order to extract such specific operations for curved surfaces, we will analyze contact relations between curved surfaces, and then consider possible transitions among them.

By using a Gaussian curvature (K) and a mean curvature (M), each surface patch can be characterized and categorized into the following six classes [3]:

* Planar surface (PL) ($K = 0, M = 0$)
* Positive Cylindrical surface (PC) ($K = 0, M > 0$)
* Negative Cylindrical surface (NC) ($K = 0, M < 0$)
* Positive Elliptic surface (PE) ($K > 0, M > 0$)
* Negative Elliptic surface (NE) ($K > 0, M < 0$)
* Hyperbolic surface (HY) ($K < 0$)

Between these six surface types, assembly relations as shown in Figure 4 occur. For an explanation of the table, let us consider the case that a manipulated object has a positive cylindrical surface (the second column). If the mating environmental surface is either a planar or a positive cylindrical surface, all the contact surface patches have the same direction; we can classify this relation as the 3d-a assembly relation. Let us suppose the environmental surface is a negative cylindrical surface. If the curvature of the environmental surface is different from the curvature of the manipulated surface, this is trivial case; 3d-a. We do not need to consider this case. If the environmental curvature is same as the manipulated curvature, either 3d-c, 3d-d or 3d-e assembly relation occur depending on the distribution of the contact orientations. The 3d-d assembly relation is depicted as the representative case in the table. If the environmental surface is a positive elliptic surface, the contact occurs at a point; the 3d-a assembly relation occurs. Neither negative elliptic surfaces nor hyperbolic surfaces can make a contact with a positive cylindrical surfaces. Other cases can be understood in the similar manner. From this table, we can summarize that the following four assembly relations occur between a pair of continuous curved surfaces: 3d-a, 3d-c, 3d-d, 3d-e, 3d-h, and 3d-f.

We will analyze assembly relation transitions of curved contacts using disassembly operations as we did in the planar contact cases [5]. All the assembly relations, except 3d-e, in the table disappear by a single detach motion resulting in the 3d-s relation.

Mating surface pairs between PL-PL, PL-PC, PL-PE, PC-PL, PC-PC, PC-PE, PE-PL, PE-PC and PE-PE form 3d-a assembly relations. A detaching motion along the contact normal direction breaks the contact; the 3d-a relation disappears resulting in the 3d-s relation; an a-to-s transition occurs. We can use the same move-to-contact motion template to achieve an s-to-a curved transition; we do not need a separate motion template.

Some mating surface pairs formed by NC-PC, PC-NC, NC-PE, PE-NC, HY-PC, HY-PE, PC-HY, PE-HY, have a 3d-d assembly relation. For example, let us consider the case

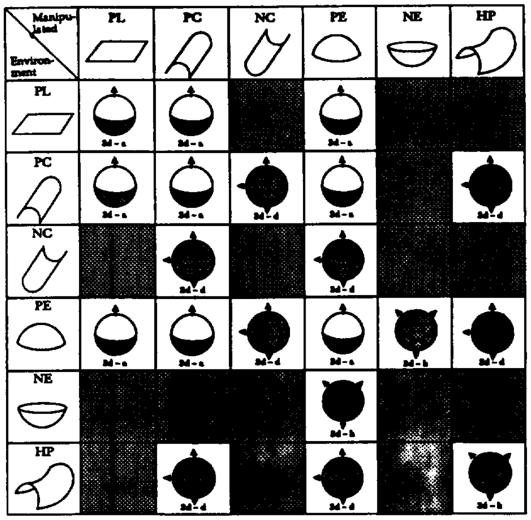

Figure 4: Assembly relations between a pair of curved surfaces

that a positive cylindrical manipulated surface has a 3d-d relation with a negative cylindrical environmental surface. The admissible region of 3d-d is a great hemicircle perpendicular to the axis of the cylinder. (See 3d-d in Figure 2.) By applying a detaching motion toward the center of the admissible hemicircle, we can break the 3d-d assembly relation and generate the 3d-s relation.

We construct an s-to-d curved surface task model and augment the abstract task models. The approach direction of this template is along the center direction of the admissible hemicircle. This task model is invoked only when the manipulated and environmental surfaces have 1) the 3d-d assembly relation, 2) the curved surface characteristics, given by the geometric models, 3) the same curvature, and 4) correspond to one of the mating pairs. Some of the mating pairs also have a 3d-c assembly relations depending on the distribution of contact normals. For this case, we construct an s-to-c curved surface task model. This task model has the similar invoking condition as the s-to-d curved surface task model.

PC-NC and NC-PC may have a 3d-e assembly relation. For example, a cylindrical bar inserted into a cylindrical hole provide this 3d-e curved surface assembly relation. By applying disassembly operation along the axis of the cylinder, we may reach either 3d-s, 3d-a, 3d-b, 3d-c, 3d-d depending on the shape of contact surfaces as is in the planar case [5]. However, there is no significant difference in the motion template for the curved surface and planar face contact. We do not construct any new curved surface task models for these transitions: s-to-e, a-to-e, a-to-b, a-to-c, and a-to-d. We will use the same planar task models.

In case that the contact pair is formed by PE-NE, NE-PE, HY-HY, the assembly relation is either 3d-h or 3d-f. This relation disappears with a single detaching motion; either a h-to-s or a f-to-s transition occurs. Thus, we construct and add the following two task models to the abstract task models: a s-to-h curved surface task model and a s-to-f curved surface task model. Similar criteria is used to invoke these task models as is used for the s-to-d curved surface task model.

3.3 Mechanical contacts

Besides these geometric contacts, industrial parts may have mechanical contacts such as bolt-nut, snaps, and gears. These mechanical contacts can be achieved by using different motion templates from ordinal motion templates for planar contacts. Thus, we will construct separate specific task models for handling these mechanical contacts.

We augment geometric models of these parts by adding mechanical properties. For example, a bolt is represented as a positive cylindrical surface. On top of that geometric representation, we add a bolt property as well as more detailed mechanical properties such as pitch and depth. A nut surface is represented as a negative cylindrical surface with nut properties. Thus, geometrically a bolt-nut relation is classified into either 3d-e or 3d-h assembly relation.

We construct s-to-e bolt-nut and s-to-h bolt-nut task models and add them to the abstract task models. The motion template in the task model contains the slots to indicate 1) approach direction of the driver for screwing the bolt and 2) the position to hold the nut. These slots are filled by observing human hand positions.

Retrieval of s-to-h bolt-nut task model occurs in the following order. 1) The system recovers the 3d-h assembly relation from the distribution of contact surface orientations: 2) it determines that the manipulated and environmental surfaces have positive and negative cylindrical properties, respectively (PC-NC mating relation): 3) it determines the property slots of the manipulated and environmental objects and recovers bolt and nut properties with the same pitch (bold-nut mating relation).

Let us consider that a human inserts a bolt into an unthreaded hole. The first step determines the 3d-h assembly relation from the geometric information. The second step recovers the PC-NC mating relation. The third step retrieves the property slots. It determines the bolt property from the manipulated object and no property from the environmental object (the hole). The bolt-nut mating relation is not established. The system invokes the usual s-to-h task model and inserts a bolt into a hole without screwing.

By using a similar method, we can also implement other mechanical relations which require special motion templates, such as snaps and gears.

Figure 5 shows the new abstract task models.

4 Task recognition system

In order to illustrate how the system works, we will demonstrate assembly operations using the following four kinds of parts: head, body, bar, and nut. Figure 11 shows an assembly, which consists of two bodies, two heads, one bar, and four nuts, assembled successfully by the system. The system has the geometric models of these objects. However, the system has to decide in what order and how to assemble these parts into a mechanical object from the observation.

4.1 Object model

Object models are described using our geometric modeler Vantage. A user represents each part using a CSG representation. Vantage converts CSG trees into boundary representations. Each boundary representation of a part consists of faces, edges, and vertices. Vantage is a frame-based geometric modeler; each geometric primitive such as a face, an edge, and a vertices - as well as the object itself - is implemented using frames. Topological relations among them are represented using winged-edge representations and are stored at appropriate slots of edge frames. Geometric information

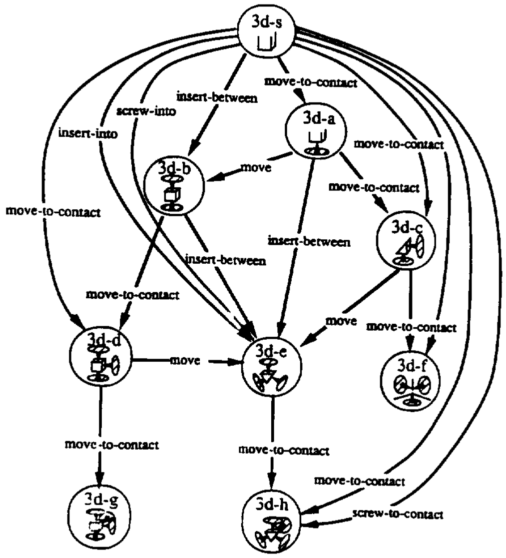

Figure 5: Abstract task models for general case

such as face equations and vertex coordinates are stored at slots of face frames and vertex frames.

Each object has its own body coordinate system. The frame of an object maintains the current transformation from the world coordinate system to the body coordinate system. We have also stored several different grasping positions in each object frame; these are represented with respect to the body coordinate system.

Vantage represents curved surfaces in two levels: approximate and global level. At the approximate level, a curved surface is represented as a collection of its polygonal approximations, while at the global level it is represented as a single frame which contains global properties such as surface equations and global connectivities to adjacent faces. It also maintains pointers to its approximate level faces.

In the current system, we have implemented two mechanical properties: bolt and nut surfaces. Geometrically, bolt and nut surfaces are represented either as a negative or positive cylindrical surfaces. Mechanical properties, such as the nut and bolt thread pitch, are stored in the mechanical property slots of the global surface frames. This method can be applied to other mechanical properties such as snaps and gears.

4.2 Image acquisition

An operator presents each assembly task one step at a time to the system. Each assembly task is observed by two different image acquisition systems: a B/W image acquisition system and a range image acquisition system. The B/W images are used to detect meaningful actions of the human operator, while the range images are used to recognize objects and hands in the scene. The system continuously observes the scene using the B/W camera and monitors brightness changes. If there is a brightness difference between two consecutive images, the system invokes a range finder to obtain a range image of the scene.

The system needs three range images at three different periods of assembly: before the task, during the task and after the task. Figure 6 shows the three images taken in one of the assembly steps. During this assembly task, the bar is put across the two bodies. The body on the table is the before-the-task image for this step. Putting the bar across the two bodies is the during-the-task image for this step. The bar lying on the two bodies is the after-the-task image for this step. The previous after-the-task image is used as the current before-the-task image. Currently, in order to take the during-the-task image, the operator has to stop his hand after he has finished the assembly task but before he has brought his hand out of the field of view. This range image will be used to choose an appropriate grasp position from among the available ones. After his hand disappears, the system takes an after-the-task image.

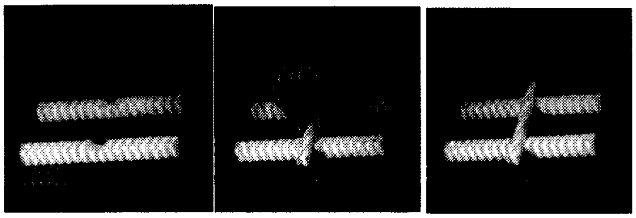

Figure 6: Three images

4.3 Object Recognition

The system makes a difference image by subtracting a before-the-task image from an after-the-task image. By applying a segmentation program to this difference image, the system extracts difference regions which correspond to surfaces of the manipulated object. The object recognition program recognizes the manipulated object, and determines its current configuration from the difference regions. The system only analyzes the difference regions; it ignores the other regions which correspond to stationary environmental objects. Thus, even in a very cluttered scene, it is efficient and robust. Based on the recognition result of the manipulated object, the system generates the current world model. In Figure 7, a cylindrical bar is the manipulated object. It has just been placed across the two bodies.

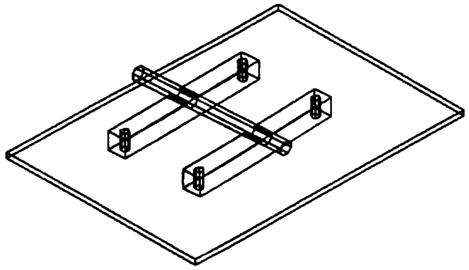

Figure 7: world-model

4.4 Task recognition

The system obtains two different kinds of information from the current world model:

* surface contact relations (task recognition)
* motion parameters (task instantiation).

By comparing the surfaces of the manipulated objects with those of the environmental objects in the updated world-model, the system determines contact surface pairs. When a pair of surfaces share common face equations, and the

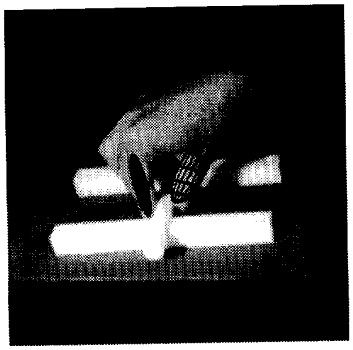

Figure 8: Hand position

vertices of the manipulated object project onto the surface of the environmental object, the system decides that the pair must contact each other. Note that since the system only examines the surfaces of manipulated objects against those of environmental objects, a combinatory explosion does not occur in this pairing operation. This remains true even when the system is handling a relatively large number of objects.

Currently, our Vantage geometric modeler represents curved surfaces, such as cylindrical or spherical surfaces by using two levels of representation: approximate and global. Using the approximate level representation, the system determines that the assembly relation is 3d-d: the normal direction of the approximate environmental face contacts are coplanar and exist at the great hemicircle of the Gaussian sphere. By considering the global representation of the environmental surface, the system determines that this pair is a PC-NC mating relation. Thus, the system retrieves the s-to-d curved surface task model.

4.5 Grasp recognition

The system makes a difference range image between the after-the-task and during-the-task image. The difference region corresponds to the hand of the operator. By fitting a superquadric representation to this region, the system recognizes the rough position of the hand. Then, the system determines the intersection between the obtained superquadric and the geometric representation of the manipulated object.

Each object model has several grasping points. The system sorts the available grasping points by the distance from the intersection point of the superquadric and the geometric representation. See Figure 8.

For each grasp point, the system checks for a collision. It generates the sweep volume of the gripper from the grasp point to the approach point, and checks whether this volume intersects any environmental objects. Once it finds a collision free grasp point, it will fill the grasp point on the instantiated task model.

4.6 Task instantiation

The s-to-d curved surface task model contains the motion parameters as shown in Figure 9. Each slot of the motion parameters contains a symbolic formula for obtaining the corresponding motion parameter from the object's current configuration. By retrieving the current configuration of the manipulated object in the world model, the system fills the motion parameters.

```
{ s-to-d
 {object-position:      NP    from observation}
 {approach-position:  AP   NP + AD * A N
 {approach-direction:AN   the center direction of the admissible directions}
 {approach-distance: AD   projection of the bounding box of contact
                               environmental patches to AN}
 {grasp-position:       GP   from observation}
  {command: "MOVE TO CONTACT  (NP+GP)  ALONG ( -AN)"}}
```

Figure 9: s-to-d curved surface task model

It often occurs that although the current relation is obtained correctly, the motion parameters contain errors, thereby causing the subsequent failure of the manipulator in performing the assembly task. The current system extracts two different kinds of information from an observation: surface contact relations and motion parameters. Our method for obtaining surface contact relations is robust against noise. We have classified all possible surface contact relations into ten representative classes. Since we classify the current relation into one of the ten distinct patterns, this discretization leads to a robust process. On the other hand, the motion parameters are obtained by simply converting object configurations. Thus, observation errors are propagated to the final motion parameters. We have developed a method to clean up the motion parameters based on the correctly obtained assembly relations [10].

In the current implementation, for PL-PL mating relations, the system adjusts motion parameters so that all the vertices of one planar surface exist on the other planar surface. A cylindrical surface is represented using its axis equations and radius. For NC-PC mating relations, the system adjusts motion parameters so that each NC-PC mating pair have parallel axes.

4.7 Program generation

The assembly task analyzed above is comprised of the following three actions: 1) picking up the bar from the part table 2) bringing it to the approach point 3) attaching the bar to the bodies. In the current implementation, all the objects are stored on the part table; the system knows their positions a priori. From recognizing that the bar is put across the bodies, the system infers that the bar is picked up from the part table. After instantiating the s-to-d task model, the system also instantiates the s-to-s task model and grasp (bringing to the approach point), and the a-to-s task model (picking up the bar from the part table).

Since the bar has a PC-PL 3d-a contact on the part table, the grasp task model brings the gripper to the grasp point of the bar and the a-to-s task model brings the bar from the original position to the depart position. The grasp point is determined from 1) grasp point chosen by s-to-a task model and 2) the current bar position on the part table. The depart point is given by the current bar position and the environmental constraint direction.

The s-to-s task model brings the bar to the safe position of the depart position. Here the safe position is the highest point which our robot can reach. Then, it brings the bar in a horizontal direction to the safe position of the approach point, which is given by the s-to-d task model. The bar descends from the safe point to the approach point. Then, the s-to-d task model forces the bar to have 3d-d contact with the bodies. Figure 10 shows the generated program corresponding to these movements and the robot performance of the program.

4.8 Another example (bolt-nut)

Figure 11 shows an assembly task which screws a bolt to the body. The object recognition program recognizes the bolt on the body's top surface. By examining the surface contact pairs, the task recognition module detects that the approximate faces of the bolt and the approximate faces of the body contact each other. From the distribution of the normals and the global level representations, the system determines that these two surface have a NC-PC 3d-e relation. The global surface of the bolt has the bolt property, and the global surface of the hole of the body has the nut property; thus, these objects are matching mates. The system retrieves the 3d-e bolt-nut task model from the abstract task models. From the 3d-e bolt-nut task model, the system finds that it is necessary to use a screwdriver while holding the objects. By extracting hand positions, the system determines the hold position and the screwdriver position.

5 Conclusion

We have described a method that can observe a human performing an assembly task, recognize object relations and relation transitions, map relation transitions to assembly tasks to cause such transitions, and then generate a program which instructs a robot to reproduce the series of movements originally performed by the human. In short, this method can recognize an assembly task performed by a human and can produce the corresponding operational sequences for a robot.

This paper describes a component of a greater effort to develop a complete APO system. The main limitation of our current system is that it analyzes only a small number of discrete images. We are planning to analyze image sequences to determine a more precise grasping strategy, a global path plan, and a fine motion strategy.

References

[1] A.P. Ambler and R.J. Popplestone. Inferring the positions of bodies from specified spatial relationships. *Artificial Intelligence*, 6(1):157–174, 1975.

[2] P. Balakumar, J.C. Robert, R. Hoffman, K. Ikeuchi, and T. Kanade. Vantage: A frame-based geometric/sensor modeling system – programmer/user's manual v1.0. Technical Report CMU-RI-TR-91-31, Carnegie Mellon University, Robotics Institute, 1991.

[3] P.J. Besl and R.J. Jain. Intrinsic and extrinsic surface characteristics. In *CVPR*, pages 226–233, San Francisco, 1985. IEEE computer society.

[4] S. Hirai and Sato. T. Motion understanding for world model management of telerobot. In *Proc. IEEE/RSJ Intern. Workshop on Intelligent Robots and Systems*, pages 124–131, 1989.

[5] K. Ikeuchi and T. Suehiro. Towards an assembly plan from observation, part i: Assembly task recognition using face-contact relations (polyhedral objects). In *Proc. of IEEE Intern. Conf. on Robotics and Automation*, Nice, France, May 1992. a longer version is avaiable as CMU-CS-91-167.

[6] Y. Kuniyoshi, H. Inoue, and M. Inaba. Design and implementation of a system that generates assembly programs from visual recognition of human action sequences. In *Proc. IEEE/RSJ Intern. Workshop on Intelligent Robots and Systems*, pages 567–574, August 1990.

[7] T. Lozano-Perez. Automatic planning of manipulator transfer movements. *IEEE Trans. System Man and Cybernetics*, SMC-11(10):681–689, 1981.

[8] T. Lozano-Perez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine-motion strategies for robots. In M. Brady and R. Paul, editors, *Robotics Research 1*, pages 65–96. MIT Press, Cambridge, MA, 1984.

[9] L.S.H. Mello and A.C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. In *Proc. of IEEE Intern. Conf. on Robotics and Automation*, pages 56–61, 1989.

[10] T. Suehiro and K. Ikeuchi. Towards an assembly plan from observation, part II: Correction of motion parameters based on face contact constraints. In *Proc. of IEEE Intern. Conf. on Intelligent Robots and Systems*, Raleigh, NC, July 1992.

[11] T. Takahashi and H. Ogata. Robotic assembly operation based on task-level teaching in virtual reality. In *Proc. of IEEE Intern. Conf. on Robotics and Automation*, May 1992.

[12] R.H. Wilson and T. Matsui. Partioning an assembly for infinitesimal motions in translation and rotation. In *Proc. of Intern. Conf. Intelligent Robots and Systems*, pages 1311–1318, Raleigh, NC, July 1992.

```
DEPARTS(1,[0])
MOVES(1,[  5.949,  16.461])
ROTATES(1,[  5.949,  16.461,   2.586, -90.000])
APPROACHS(1,[  5.949,  16.461,   2.586, -90.000])      Grasping
MOVES(1,[  5.949,  16.461])
ROTATES(1,[  5.949,  16.461,   1.011, -90.000])
APPROACHS(1,[  5.949,  16.461,   1.011, -90.000])
CLOSE(1)
ROTATES(1,[  5.949,  16.461,   1.405, -90.000])
MOVES(1,[  5.949,  16.461])                            3d-a to 3d-s
APPROACHS(1,[  5.949,  16.461,   1.405, -90.000])
DEPARTS(1,[0])
MOVES(1,[ 26.467,   7.300])                            3d-s to 3d-s
ROTATES(1,[ 26.467,   7.300,   2.207,   3.011])
APPROACHS(1,[ 26.467,   7.300,   2.207,   3.011])
MOVES(1,[ 26.467,   7.300])
ROTATES(1,[ 26.467,   7.300,   1.700,   3.011])
APPROACHS(1,[ 26.467,   7.300,   1.700,   3.011])      3d-s to 3d-d
MOVES(1,[ 26.467,   7.300])
ROTATES(1,[ 26.467,   7.300,   1.503,   3.011])
APPROACHS(1,[ 26.467,   7.300,   1.503,   3.011])
OPEN(1)                                                Ungrasping
```

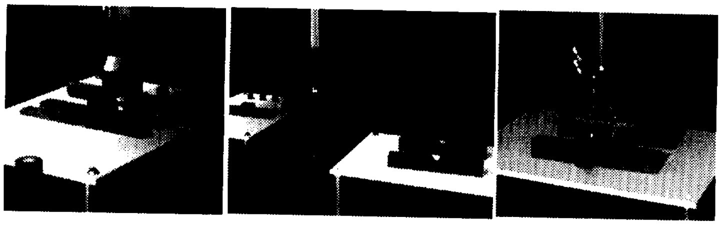

Figure 10: Generated program and the robot performance

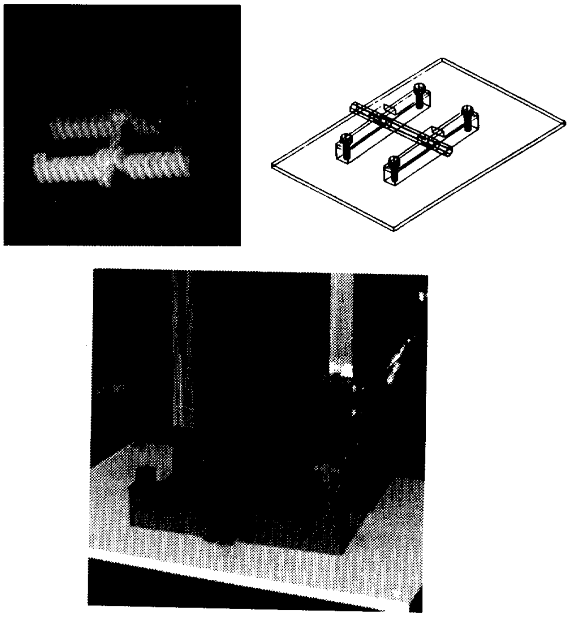

Figure 11: Screwing a bolt with a screwdriver