Assembly Plan from Observation*

Katsushi Ikeuchi Sing Bing Kang

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 overviews our effort toward the realization of this method.

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 [8]. 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.* [7] and Takahashi *et al.* [12].

In our APO approach, a human operator performs assembly tasks in front of a video camera. The system obtains a continuous sequence of images from the camera which records 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.

Section 2 designs the abstract task models used in the task recognition process while section 3 discusses how to use the models inthe task recognition system. Our recent work on the temporal segmentation of the task sequence and the subsequent grasp recognition is detailed in sections 4 and 5.

2 Defining Abstract Task Models

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.

*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.

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

2.2 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).

We can classify any nth directional assembly relation into one of the ten representative assembly relations [5].

2.3 Abstract task model

An abstract task model associates an assembly relation transition with an action which causes such a transition. We will extract what kind of transition occurs within the assembly relation taxonomy. We conduct this analysis by considering possible disassembly operations [5]. By assigning an appropriate motion template to each arc of the graph, we have developed abstract task models as shown in Figure 1. Note that the abstract task models also able to handle bolt-and-nut mechanical relations. See [6] for more details.

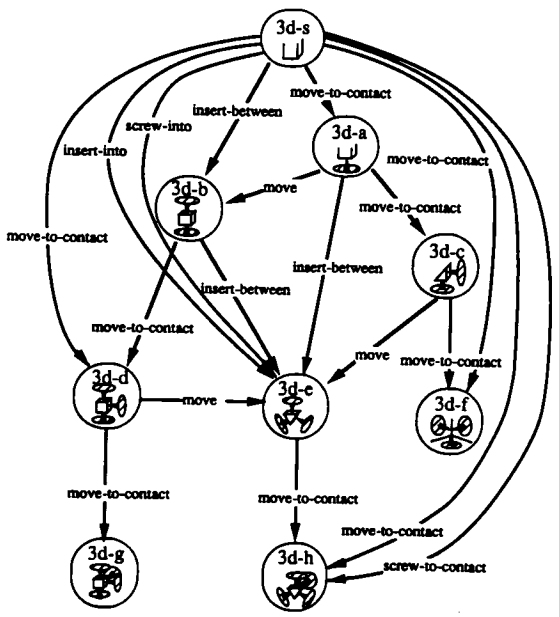

Figure 1: Abstract task models

3 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. 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.

3.1 Object model

Object models are described using our geometric modeler Vantage [2]. Each part is modeled 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 such as face equations and vertex coordinates are stored at slots of face frames and vertex frames.

3.2 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 two range images at two different periods of assembly: before the task and after the task. Figure 2 shows the two 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. 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.

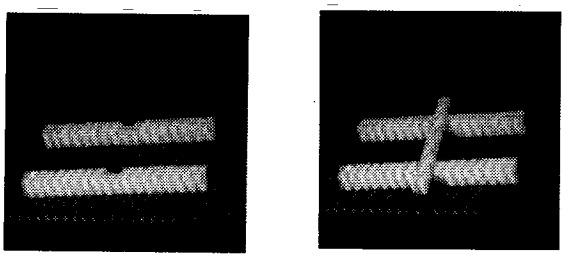

Figure 2: Two images

3.3 Object Recognition

The system creates 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 the "difference" regions which correspond to surfaces of the manipulated object. The object recognition program recognizes the manipulated object, and determines its current pose 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 3, a cylindrical

bar is the manipulated object. It has just been placed across the two bodies.

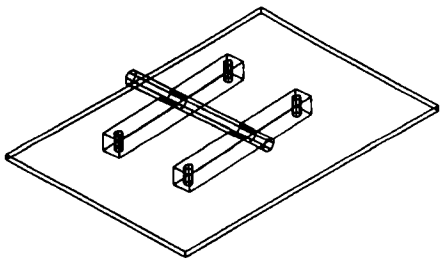

Figure 3: Model of the world

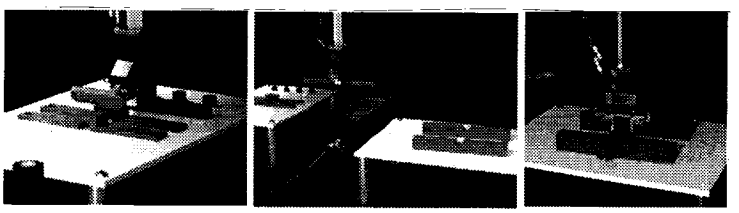

Figure 4: Robot performance

3.4 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 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.

The Vantage geometric modeler represents curved surfaces, such as cylindrical or spherical surfaces, 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 curved mating relation. Thus, the system retrieves the s-to-d curved surface task model.

3.5 Task instantiation

The s-to-d curved surface task model contains the motion parameters. 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 and performs the task as shown in Figure 4.

Thus far, we have described work on deducing the task based on two different snapshots of the task, namely the before- and after-the-task images. This would not be sufficient to extract direct and detailed information on the human grasping strategy, the type of motions involved in the task, as well as the hand global motion that may be of use in planning the robot execution of the task. We address this deficiency by temporally segmenting the task sequence into meaningful segments for further analyses, one of which is human grasp recognition.

4 Temporal segmentation of tasks from human hand motion

We propose to analyze image sequences obtained during the human assembly task to identify the grasping strategy as well as the task actions from observation. The first step in this direction is to determine the motion breakpoints based on human hand configuration and pose throughout the task sequence [16].

We can segment the entire task into meaningful subparts (pregrasp, grasp, and manipulation phases) by analyzing both the fingertip polygon area and the speed of the hand. The fingertip polygon is the polygon formed by the fingertips as its vertices. A very useful measure that can be used to segment the task more effectively is called the *volume sweep rate* [16], which is the product of the fingertip polygon area and the hand speed. The volume sweep rate measures the rate of change in both the fingertip polygon area and the speed of the hand.

The algorithm to segment a task sequence into meaningful subsections starts with a list of breakpoints comprising local minima in the speed profile. The global segmentation procedure basically makes use of the goodness of fit of the volume sweep rate profiles in the pregrasp phases to parabolas (inverted U-shapes). The desired breakpoints are obtained by minimizing the mean fit error of the parabolas subject to the first three conditions.

Our experiments are conducted using a hand-tracking system which comprises the CyberGlove [17] and Polhemus [18] devices. The CyberGlove measures 18 hand joint angles while the Polhemus device measures the pose (i.e., translation and orientation) of the human hand in 3D space. The Ogis light-stripe rangefinder and a CCD camera provide the range and intensity images, respectively.

An example task whose breakpoints have been correctly identified are shown in Figure 5. The breakpoints have been correctly identified despite the different types of manipulative actions (pick-and-place, insertion, and screwing actions). As can be seen in Figure 5, the volume sweep rate profiles have highly accentuated peaks during the pregrasp phases, thus facilitating the determination of the motion breakpoints.

Once the motion breakpoints have been determined, recognition of the grasp employed can then be carried out on the temporally located grasp frame/s.

5 Grasp classification and recognition

Grasp identification is central to the recognition of grasping tasks. In order to identify grasps, we need a suitable grasp taxonomy. To this end, we use a grasp representation

call the *contact web* [14]. The contact web spatially represents effective contact between segments of the hand and the object; its notation is shown in Figure 6. Each effective contact point is associated with contact position and force vector (approximated by the object normal at the point of contact).

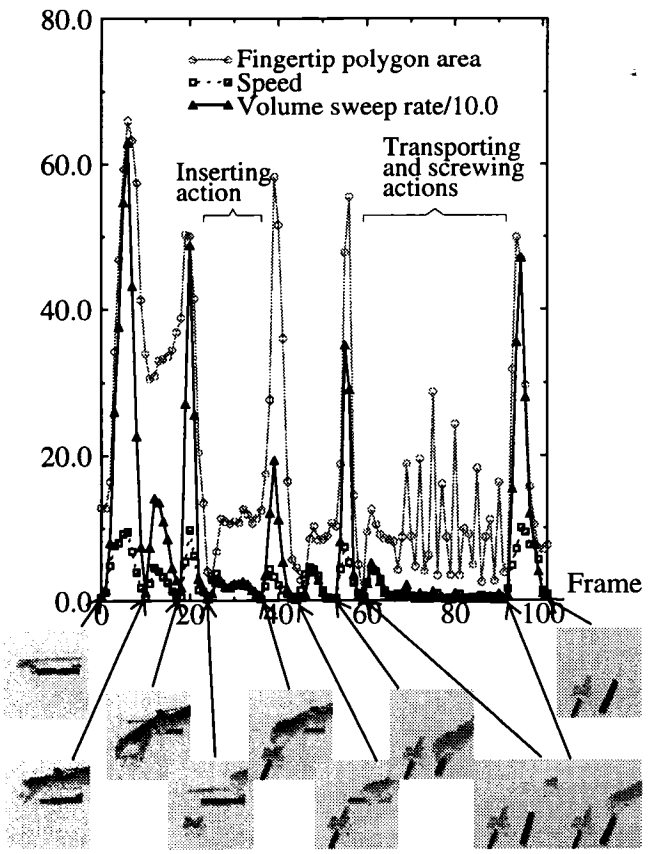

Figure 5: Identified breakpoints in task sequence

5.1 The proposed grasp taxonomy

The proposed grasp taxonomy based on the contact web. The first level dichotomy is the volar/non-volar grasp branch; a grasp is initially classified according to whether there is direct object-palmar surface interaction or not (respectively). The non-volar grasps are further classified as *fingertip grasps* and *composite non-volar grasps*.

Intermediate-level grasp concepts, namely the *virtual finger* [20] and *opposition space* [21] are also used to complement the contact web. This enables the grasp to be hierarchically represented [15]. We have described a mapping function that groups real fingers into virtual fingers, i.e., collections of "functionally" equivalent (in terms of similarity of action against the object surface) fingers [14]. A result of this mapping is an index called the *grasp cohesive index*, which indicates the degree to which the fingers that are grouped into virtual fingers act in a similar manner against the grasped object.

5.2 Procedure for grasp recognition

Using the results of a series of experiments conducted [22], a grasp can be identified from the following general steps:

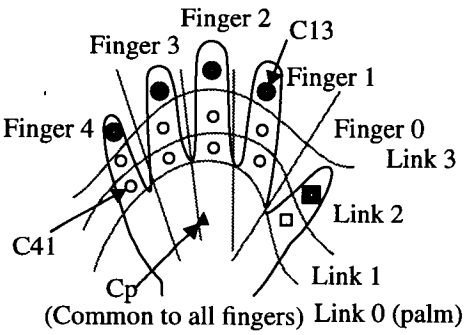

Figure 6: Contact notation on the right hand (palmar side)

1. Compute the real finger-to-virtual finger mapping which yields the virtual finger compositions and the grasp cohesive index.
2. If the palm surface is not involved in the grasp, classify it as a non-volar grasp.
3. Otherwise, by checking the grasp cohesive index and, if necessary, the degree of thumb abduction, classify it either as a spherical, cylindrical or coal-hammer (type 1 or 2) power grasp.

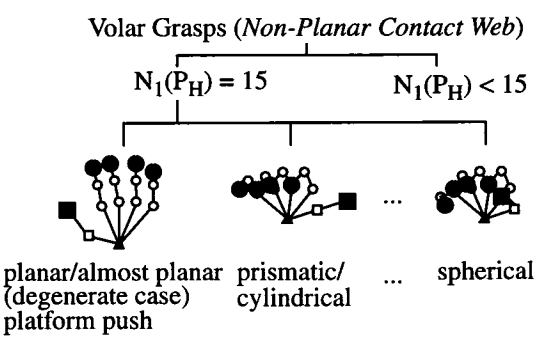

Figure 7: Discrimination graph for volar grasps

A grasp is initially classified as a volar grasp or a non-volar grasp. If the the grasp is non-volar, its detailed identification follows the non-volar taxonomy, i.e., according to the number of finger and finger segments touching the object, and the shape of the contact points [14]. However, if it is a volar grasp, further identification follows the discrimination procedure shown in Figure 7. The "coal-hammer" grasp is a special case of the cylindrical power grasp, and is identified by the high degree of thumb abduction. We define the type 1 "coal-hammer" grasp to be one in which the thumb does not touch the held object, while the type 2 "coal-hammer" grasp refers to one in which the thumb touches the object. The type 2 "coal-hammer" grasp is differentiated from the cylindrical power grasp by its high degree of thumb abduction.

5.3 Experimental results

The results of two of the experiments using the CyberGlove are shown in Figure 8. The grasps in Figure 8(a) and (b) have been correctly identified as a type 2 "coal-hammer"

cylindrical power grasp and a five-fingered disc precision grasp respectively.

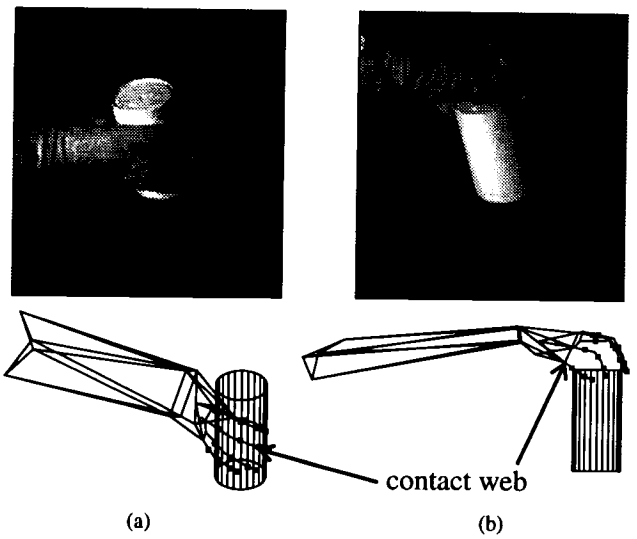

Figure 8: Experiments using the CyberGlove: (a) a power grasp, and (b) a precision grasp

6 Conclusion

We have described a method that enables a robotic system to observe a human perform an assembly task, recognize object relations and relation transitions, and map relation transitions to assembly tasks to cause such transitions. This system subsequently generates a program which instructs a robot to reproduce the series of movements originally performed by the human. In short, this method enables a robotic system to recognize an assembly task performed by a human and produce the corresponding operational sequences for a robot. We have also reported on-going work on the temporal segmentation of the entire task sequence and the recognition of the human hand grasp.

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 *Proc. of IEEE Intern. Conf. on Comp. Vision and Patt. Recog.*, pages 226–233, San Francisco, 1985. IEEE Computer Society.

[4] S. Hirai and T. Sato. 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] K. Ikeuchi, M. Kawade and T. Suehiro. Assembly Task Recognition with Planar, Curved, and Mechanical Contacts In *Proc. of IEEE Intern. Conf. on Robotics and Automation*, vol. 2, pages 688–674, Atlanta, GA, May 1993.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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/RSJ Intern. Conf. on Intelligent Robots and Systems*, Raleigh, NC, July 1992.

[12] 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.

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

[14] S.B. Kang and K. Ikeuchi. Grasp recognition using the contact web. In *Proc. IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*, pages 194–201, Raleigh, NC, July 1992.

[15] S.B. Kang and K. Ikeuchi. A grasp abstraction hierarchy for recognition of grasping tasks from observation. In *Proc. IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*, pages 621–628, Yokohama, Japan, July 1993.

[16] S.B. Kang and K. Ikeuchi. Temporal segmentation of tasks from human hand motion. Technical Report CMU-CS-93-150, Carnegie Mellon University, April 1993.

[17] CyberGlove™ System Documentation. Virtual Technologies, June 1992.

[18] 3Space Isotrak User's Manual. Polhemus, Inc., Jan. 1992.

[19] Knowledge Craft Manual - Vol. 1: CRL Technical Manual. Carnegie Group, Inc., 1989.

[20] M.A. Arbib, T. Iberall, and D.M. Lyons. Coordinated control programs for movements of the hand. In *Experimental Brain Research Series 15 - Generation and Modulation of Action Patterns*, eds. H. Heuer, and C. Fromm, Springer-Verlag, pages 111–129, 1985.

[21] T. Iberall, G. Bingham, and M.A. Arbib. Opposition space as a structuring concept for the analysis of skilled hand movements. In *Hand Function and the Neocortex*, eds. A.W. Goodwin, and I. Darian-Smith, Springer-Verlag, pages 158-173, 1986.

[22] S.B. Kang and K. Ikeuchi. A framework for recognizing grasps. Technical Report CMU-RI-TR-91-24, Carnegie Mellon University, Nov. 1991.