

# Automating Microassembly with Ortho-tweezers and Force Sensing

J.A. Thompson, R.S. Fearing  
Department of EE&CS

University of California, Berkeley, CA 94720  
ronf@robotics.eecs.berkeley.edu

## Abstract

*We describe a microassembly system based on two one degree of freedom probes ("ortho-tweezers") and a three degree of freedom translating stage, controlled by a Java application with Python scripting support. We describe how basic manipulation primitives can be combined in a Python script to perform automated, force-feedback controlled assembly operations. We demonstrate a fully automated pick and place task using 200×200×100 micron blocks. Using a temporary handling block, the system can manipulate long, thin, fragile parts, such as semiconductor strain gauges.*

## 1 Introduction

The problem of robotic microassembly has been explored using high precision actuators and vision feedback in work by Codourey et al [1995], Feddema and Simon [1998], Kasaya et al [1998], Nelson et al [1998], and Sulzmann [1998]. Vision-based approaches are limited by poor depth of field of high power microscopes, cluttered views, and lack determination of contact or contact forces. In addition, it is difficult to perform several distinct operations in parallel as microscopes are quite bulky and expensive (although parallel operations can be performed with rigid pallets and fixtures [Feddema and Christensen 1999]). Alternatively, force sensor based approaches, such as described in the present paper, can be local and provide exact information about contact between surfaces (Zesch and Fearing [1998], Sitti and Hashimoto [1999], Zhou and Nelson [1998]).

Previous micromanipulation work has used single probes or parallel jaw grippers to manipulate parts. The parallel jaw gripper approach follows from macro-robotics where a simple gripper is used with a 6 degree of freedom (DOF) arm to reorient and position parts. As sub-centimeter 6 DOF micro-robot arms are not yet available, we show how macro-scale dextrous manipulation techniques can be used with much simpler mechanisms to reorient and position parts. (Bicchi [2000] provides a historical survey of work toward simplified robust grasping.) By using gripping forces which exceed adhesion forces, we can use Coulomb friction to control part sticking and sliding.

This paper reports development of such a microassembly system based on dextrous micromanipulation primitives described in Shimada et al [2000]. In that paper, an assembly process was shown which was created by open-loop commands of the probe and stage positions, without the use of strain-gauge feedback, and relied on exact positioning of all the parts. Micromanipulation primitives were developed to reorient a part in the plane, regrasp a part in the plane, pivot a part about a fixture, and align a part with an edge. In the current paper, to make the micromanipulation primitives more robust, we developed several manipulation primitives which utilize force feedback from the strain gauges on the probe tips. For example, the Sweep Probe X To Contact primitive will search for a part on the surface. Therefore, the system is much more robust to uncertainty in part location.

Allen et al [1990] describe manipulation primitives to control a multi-sensor robotic Utah-MIT hand. These primitives were combined in a programming language called DIAL which was originally developed for scripting graphical animations. DIAL was chosen because it is convenient for controlling many tasks in parallel. A drawback, however, is that DIAL runs the script in "open-loop" without the ability to branch on different conditions found through feedback from the robot. To allow for this feedback control, we chose the Python language (see <http://www.python.org>), which is a full scripting language that allows conditionals, loops and subroutines. We describe our system's programming environment in more detail in section 1.4.

### 1.1 Ortho-tweezers mechanism

The system uses a new type of gripper based on a mechanism ("ortho-tweezers") which can be more easily implemented at the micron scale than an anthropomorphic hand. The mechanism is shown graphically in figure 1.

The part is grasped by two 1 DOF compliant probes, called probe X and probe Y, mounted orthogonally to each other. (Hence, "ortho-tweezers"). Probe X is so called because it can only move in the +/- X direction, and probe Y only moves in the +/- Y direction. The part is resting on a 3 DOF translating stage which prepositions the part where the probes can grip it, shown in figure 2.

The probes can rotate the part by  $\theta$  around the Z axis. Figure 3 shows a single grip rotating the part by 90°.

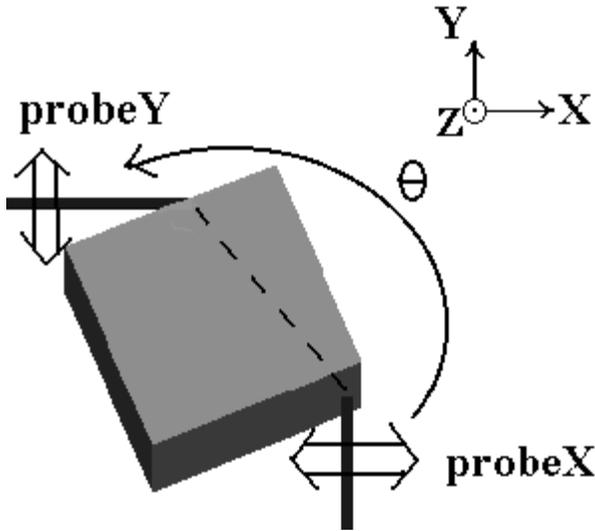


Fig. 1: Ortho-tweezers mechanism.

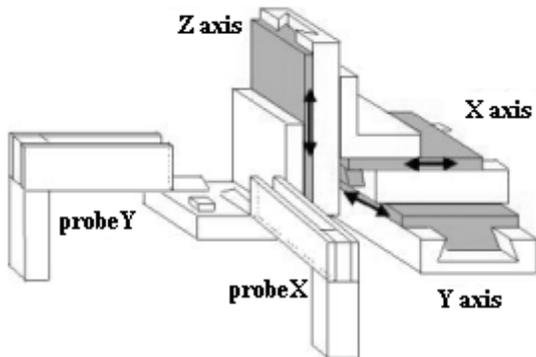


Fig. 2: Probe X and probe Y with the three axes that move the three DOF translating stage. A small square part is shown on the stage.

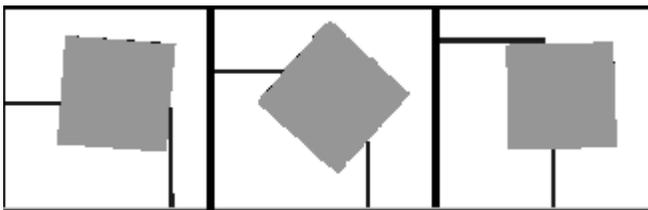


Fig. 3: Probes rotating the part by 90°

## 1.2 Strain Gauges

Each probe has semiconductor strain gauges on the base of the arm and on the tip (Entran ESB-020-350, <http://www.entran.com>). The gauges on the base of the arm sense the overall deflection of the probe as it is driven by a piezoelectric actuator (Thunder TH8-R <http://www.face-int.com/thunder>). The gauges on the stainless steel tip (figure 4) sense the gripping force with sub-micron accuracy. Note that, for example, the strain gauges on probe X tip only sense left/right forces. They do not

sense forces pressing "into" the tip orthogonal to this.

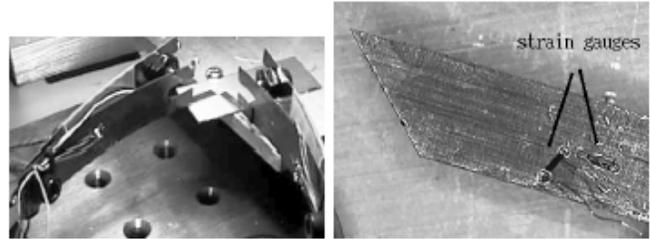


Fig. 4: a) Ortho-tweezers hardware with perpendicular piezo-driven probes. b) 1 mm semiconductor strain gauges mounted on stainless steel probe tips

We estimate the stiffness  $k$  of the probe tip to be about 100 N/m, so that a deflection of  $10^{-6}$  meters corresponds to about  $10^{-4}$  Newtons, or 0.1 milli-Newton. This small force accuracy allows us to reliably "tap" a block with 0.4 milli-Newtons as it sits on Gel-Pak without altering its position. (See section 1.3 for details on Gel-Pak.) This "tapping" operation is used for the Sweep Probe To Contact and Grasp manipulation primitives (described in section 3) in order to locate a part. To fully grasp the part, the probes apply a force of about 4 milli-Newtons.

Figure 5 shows a plot of force versus time on one of the probe tips for a grasping operation. The beginning of the plot shows ten small "tapping" forces as the probe contacts the part in order to locate it. The the force plot increases sharply as the gripping force is applied.

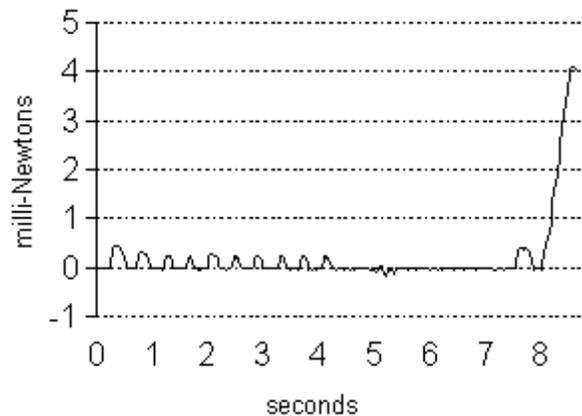


Fig. 5: Plot of force in milli-Newtons versus time on probe tip for grasping operation. Ten locating taps are clearly seen, and then a large force during the grip.

Note that the strain gauges are sensitive enough to resolve the small tapping force of about 0.4 milli-Newtons versus the gripping force of about 4 milli-Newtons, yielding an order of magnitude difference in the forces used to sense versus manipulate.

### 1.3 Sticking Effects

As described in Fearing [1995], a major obstacle to overcome in microassembly is the strong adhesion effects which make it difficult to release a part from the gripper. In the previous work described in Shimada et al [2000], ungrasping was achieved by placing the block against a barrier on the work surface which would allow the grippers to be pulled away. We have recently found, however, that low taction Gel-Pak as the work surface can meet most of our needs. (See <http://www.gelpak.com>, GEL-Film PF-80-X0.) A block can be placed on the Gel-Pak and released reliably. Furthermore, the block will stay in place as it is repeatedly contacted from the side as described below in the Grasp manipulation primitive.

### 1.4 Ortho-Tweezers Control Application

We have developed a Java application to control the ortho-tweezers system. A screen shot of the main window is shown in Figure 6.

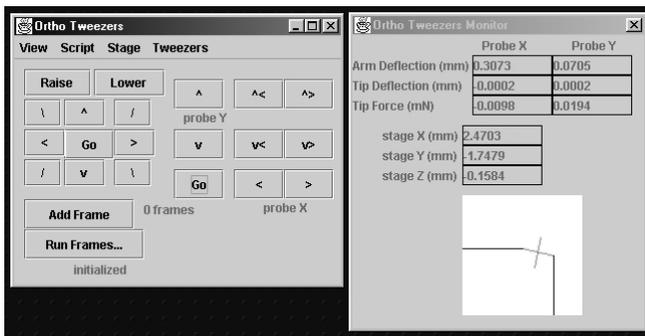


Fig. 6: Ortho-tweezers Java application

We chose Java for two main reasons. First, it is a standardized programming environment for writing an application with a graphical user interface. And second, its integrated multi-threading and exception handling make it easy to write control procedures which can be interrupted when the user clicks the "Cancel" button.

The manipulation primitives described in section 3 are all part of the Java application. However, if a user wishes to extend the system to control a larger assembly operation, as described in the demonstration pick and place in section 4, it is inconvenient to have to write, compile and execute new Java code. We would prefer to define new procedures with a script. Fortunately, the Jython software package provides an easy way to allow a Python script to call Java procedures. (See <http://www.jython.org>.) The appendix shows the Python script used for the pick and place example.

## 2 Use of Handling Blocks

Many thin microparts, such as a strain gauge, cannot be

directly gripped by the ortho-tweezers probes. However, the ortho-tweezers can reliably grip and manipulate a small block, as shown by the demonstration pick and place in section 4. Therefore, we can attach a handling block to the micropart, manipulate the part to its assembled location, and remove the handling block.

As a means of temporarily attaching the handling block to a strain gauge, we use low melting-point wax (Edmund Scientific Flexwax 120.  $T_{melt} = 49^{\circ} \text{C}$ .) The strain gauge (as with most parts in our system) sits on a sheet of Gel-Pak which in turn is on a surface with heating coils. (Since the strain gauge in the figure 7 is only  $1.0 \times 0.15 \times 0.012 \text{ mm}$ , the heating coils are too large to see.) In figure 7a, the handling block (200 microns wide) has been dipped in a shallow pool of low melting-point wax, elsewhere on the heated surface. As the handling block is moved to the strain gauge (figure 7b), the wax cools and solidifies. But as soon as it contacts the warm strain gauge, it melts again.

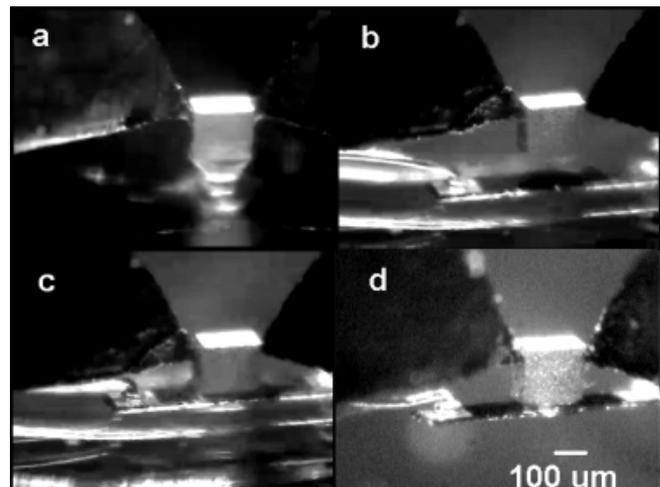


Fig. 7: Attaching handling block to strain gauge with molten wax. The probe tips are viewed from the side as in figure 2. a) Withdrawing from molten wax reservoir. b) Before placing on gauge. c) Holding as wax cools. d) Attached gauge lifted free of surface. The strain gauge is 1 mm in length and sits on Gel-Pak which is on a heating surface. The block is  $200 \times 200 \times 100$  microns.

Next, as the ortho-tweezers continue to hold the handling block in place (figure 7c), the heat is turned off and the wax cools, forming a bond with the strain gauge. The tweezers can now lift the strain gauge and fully manipulate it to the place of assembly (figure 7d).

If we need to attach a handling block to another micropart, we must first relocate this one to a different section of the working surface away from the heating element. If we do not, then when we turn the heater back on in order to melt the wax for attaching the next handling block, the wax bond on the first handling block

will be melted again. This problem can be solved with multiple heating elements, so that only the micropart having the handling block attached is heated.

### 3 Manipulation Primitives

Each of the manipulation primitives described here is a call to a Java method. There are many other lower-level methods not shown here, such as to move the stage, deflect the probes, or measure the strain gauges. We only describe the higher-level primitives because they demonstrate our micromanipulation approach which is based on force sensor data from the probe tips. The theme behind these primitives is to minimize sensing and actuation degrees of freedom. For example, there are no Z axis sensors, but we need to know Z height, so the Seek Z Surface primitive accomplishes this with the same X axis probe tip sensor used in other primitives.

#### 3.1 Seek Z Surface

To grasp a block from the surface, it is first necessary to know the location of the surface in the Z (vertical) direction. The Seek Z Surface primitive slowly moves the probes closer to the stage while making small sweeping motions with probe X. (The choice of probe X instead of probe Y is arbitrary.) The algorithm continues until it detects a force on the probe X tip, or until it reaches a given limit in Z displacement. If the force is detected, the algorithm deflects the probe a little more, which increases the felt force, and "backs off" until the force is no longer detected. This is assumed to be the Z surface. Note that this works because the surface is Gel-Pak and so, although the probe X is sweeping parallel to the surface, it will "grab" the surface and detect a force on the tip.

#### 3.2 Sweep Probe X or Y To Contact

Once the Z level of the surface is found, it is necessary to locate a block in the X or Y direction. For example, to find the edge of a block in the Y direction, the Sweep Probe X To Contact algorithm sweeps with probe X to the left looking for a contact force on the tip, as shown in figure 8a.

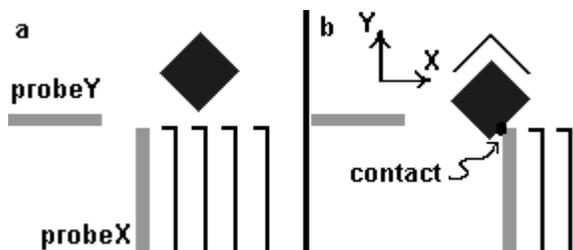


Fig. 8: Sweep Probe X To Contact. a) Probe X sweeps to left without contact. b) The block has been moved an increment in Y, so probe X now makes contact.

If no contact is found, it restores the probe X position to the right and increments the Y position of the probes (which is actually accomplished by decreasing the Y position of the translating stage on which the block rests). The algorithm repeats until probe X detects a force on the probe tip, or until the algorithm reaches a limit in overall Y displacement without contact. If the force is detected (figure 8b), it is assumed that the tip is in contact with the right side of the block near the lower part of the block (as measured in the Y direction.) The Sweep Probe X To Contact algorithm can also sweep probe X from left to right in order to find the left side of a block near the lower Y part of the block.

To find the location of the block in the X direction, Sweep Probe Y To Contact is used in a similar manner.

#### 3.3 Grasp

Now that the block has been located in the X, Y and Z directions, it is ready to grasp it. The Grasp algorithm assumes that the block is a rectangular object positioned at a non-zero angle to the X-Y axes. Since parts are usually picked from a pre-arranged pallet, this is a reasonable assumption. The Sweep Probe X To Contact and Sweep Probe Y To Contact primitives have ensured that moving probe X to the left will contact the lower-right side of the block and that moving probe Y downwards will contact the upper-left side of the block. (See figure 9.)

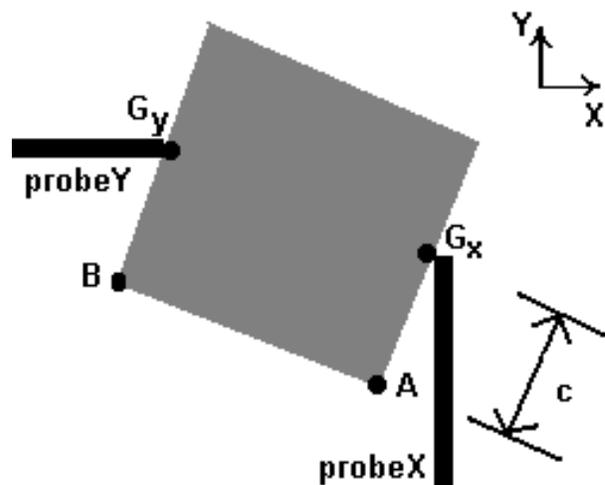


Fig. 9: Grasp algorithm. Lower corner of block at point A, left corner of block at point B. Offset distance c from corners for the grip points  $G_x$  and  $G_y$ .

But the Grasp algorithm must find the precise locations of the lower and left corners of the block. To find the lower corner of the block at point A, the Grasp algorithm sweeps probe X to the left until it contacts the block, then "backs off" and the stage is moved slightly in the Y direction so that probe X will contact the part closer to point A. The

algorithm repeats this until it no longer contacts the block when it sweeps probe X to the left. The point at which it did contact the block on the previous iteration is assumed to be point  $A$ . Sweeping with probe Y in a similar manner is used to locate the left corner of the block at point  $B$ .

The Grasp algorithm next computes a normalized vector  $\hat{e}_1$  along the edge from point  $A$  to point  $B$  :

$$\hat{e}_1 = \frac{\mathbf{B} - \mathbf{A}}{|\mathbf{B} - \mathbf{A}|}$$

and rotates it  $90^\circ$  to obtain a normalized vector  $\hat{e}_2$  so that it points along the lower-right edge of the block:

$$\hat{e}_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \hat{e}_1$$

It then uses the offset distance  $c$  along the edge to find the contact points  $\mathbf{G}_x$  and  $\mathbf{G}_y$ , at which probe X and probe Y will grasp the block:

$$\begin{aligned} \mathbf{G}_x &= \mathbf{A} + c \hat{e}_2 \\ \mathbf{G}_y &= \mathbf{B} + c \hat{e}_2 \end{aligned}$$

The distance  $c$  is determined by knowing the approximate geometry of the part. For example, for a part 200 microns on a side, we may pick  $c=100$  microns.

To apply the gripping force, the Grasp algorithm uses the Seek Z Axis Angle primitive, described next, to seek a grip at the present angle between the probe tips and at the desired gripping force.

### 3.4 Seek Z Axis Angle

The Seek Z Axis Angle manipulation primitive attempts to rotate the block around the Z axis to a given angle while maintaining a desired force between the probe tips and the block. (Rotation around the Z axis is shown in figure 3.) We typically choose a desired gripping force near the maximum that our strain gauges can resolve. As was shown in figure 5, this is about 4 milli-Newtons.

The points  $\mathbf{G}_x$  and  $\mathbf{G}_y$  (shown in figure 9) are the grip points for probe X and probe Y, respectively. As explained in section 1.2, strain gauges on the probe arm measure deflection, so we always know the deflection  $\Delta x$  of probe X from where it would touch probe Y and the deflection  $\Delta y$  of probe Y from where it would touch probe X. In terms of  $\mathbf{G}_x$  and  $\mathbf{G}_y$ , these are:

$$\begin{aligned} \Delta x &= (\mathbf{G}_y - \mathbf{G}_x) \cdot \hat{i}_x \\ \Delta y &= (\mathbf{G}_x - \mathbf{G}_y) \cdot \hat{i}_y \end{aligned}$$

The angle  $\theta$  between the probe tips is:

$$\theta = \tan^{-1} \left( \frac{\Delta x}{\Delta y} \right)$$

Thus,  $\theta$  increases for positive rotation around the Z axis.

At any one time, to rotate the block in a certain direction we have to decide whether to move one probe "inward" toward the position where both probes would touch, or to move the other probe "outward." For example, to rotate the block slightly from the position shown in the middle of figure 3 to the position shown on the right, we can move probe X left ("inward") or probe Y up ("outward"). The decision is made as follows.

Case 1: The tip force on both probes is less than the desired gripping force. In this case, the algorithm moves the appropriate probe slightly inward to cause rotation in the desired direction. This also has the effect of increasing the force.

Case 2: The tip force on either of the probes is greater than the desired gripping force. In this case, the algorithm moves the appropriate probe slightly outward to cause rotation in the desired direction. This also has the effect of decreasing the force.

The algorithm repeats this procedure until the block is rotated to the desired angle.

The stability of this algorithm during the rotation relies on the passive stability of a two-finger grasp as described in [Fearing 1986], and as applied to the ortho-tweezers configuration in [Shimada et al 2000].

### 3.5 Seek Grip Axis Angle

It is possible to rotate a block around the grip axis by pushing it against the horizontal edge of a static barrier as illustrated in figure 10. As shown by [Fearing 1986] and [Gopalswamy and Fearing 1989] two-finger grasps of polygons and polyhedra (respectively) will automatically slide to a stable configuration if the angle between the included faces is less than twice the friction angle. Additionally, a tangential force at one finger will cause the grasped part to roll about the opposite finger. As these grasping methods do not require feedback, and are robust to initial conditions, they are well-suited to the micro-domain. Grasping methods and automatic planners using slip have been discussed further by Brost [1986], Carlisle et al [1994], Erdmann et al [1993], Goldberg [1993], Rus [1993], Yoshikawa et al [1993], and Wiegly et al [1997].

Rao et al [1996] have shown that the combination of a rotation around the Z axis and a rotation around a grip axis can achieve any orientation of a part. Thus, by combining the Seek Z Axis Angle primitive and Seek Grip Axis Angle primitive (described in this section), the ortho-tweezers mechanism can fully rotate the part.

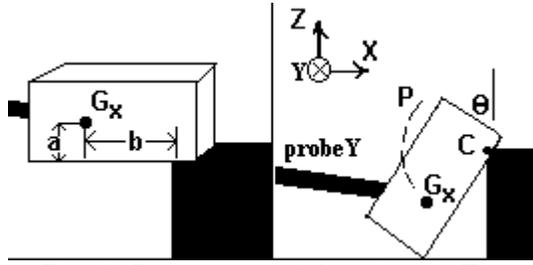


Fig. 10: Rotation around the grip axis. The block (white) is rotated against the barrier (black). The probes grip the block with distances  $a$  and  $b$  from the point of contact  $C$  with the barrier, rotating along circular path  $P$  around  $C$  to an angle  $\theta$  from the horizontal. Probe X is not shown because we are viewing along it.

In figure 10, the Y axis goes into the page, so we are viewing along probe X to where it contacts the block at point  $G_x$ , and hence probe X is not shown.

The Seek Grip Axis Angle algorithm moves the grip points along a path  $P$  until the block is rotated an angle  $\theta$  from the horizontal.  $P$  is a circular path centered around the contact point  $C$  between the block and the barrier. In order to compute the contact point  $C$ , the algorithm must be given the distances  $a$  and  $b$ , which are, respectively, the vertical distance along the Z axis from the probe grip point to the bottom of the block, and the horizontal distance along the X axis from the grip point to where the barrier contacts the bottom of the block. The distance  $a$  is known because Seek Z Surface was used to determine the vertical position of the surface which contacts the bottom of the block, and the probes were raised the distance  $a$  before using the Grasp primitive. The distance  $b$  is known because the position of the static barrier is determined in advance, and the positions of the probe tips are known.

The point  $G_x$  where probe X contacts the block is known at each position. We choose the initial angle  $\theta_{init}$  between  $G_x$  and  $C$  as:

$$\theta_{init} = \text{atan2}(a, -b)$$

where  $\text{atan2}(z,x)$  computes  $\tan^{-1}(z/x)$  with the resulting angle in the correct quadrant. Our final angle  $\theta_{final}$  of  $G_x$  relative to  $C$  is:

$$\theta_{final} = \theta_{init} + \theta$$

so that  $G_x$  traverses a path along the circle of increasing angle until it finishes when:

$$\mathbf{G}_x = \mathbf{C} + \sqrt{a^2 + b^2} \left( \cos \theta_{final} \hat{\mathbf{i}}_x + \sin \theta_{final} \hat{\mathbf{i}}_z \right)$$

### 3.6 Ungrasp

The Ungrasp algorithm makes use of Seek Z Axis Angle (as did Grasp) by seeking the present Z axis angle but at zero force. Since the Seek Z Axis algorithm makes small changes continually on both probes, the grip is released smoothly, whereas simply removing one probe, then the other, would cause the block to rotate away from its present angle. Once the probe tips are touching the block, but at zero force, the algorithm moves the probes slightly toward the surface in the Z direction. This is done to "unstick" the probes from the surface. Now, the algorithm can open the probes to separate from the block.

### 3.7 Heater On/Off

As described in section 2, we use a heating element in the surface to melt and cool wax in order to attach a handling block to a micropart. The Heater On primitive takes a number  $n$  and activates heating element  $n$ . Similarly, the Heater Off primitive deactivates heating element  $n$ .

At present we have only built one heating element into the assembly surface, but in the future we plan to have multiple elements. This will improve the situation described in section 2, and allow a micropart with its attached handling block to remain cool while the next micropart on another heating element can be warmed up in order to receive its handling block.

## 4 Demonstration Pick and Place

As a demonstration of the microassembly system, we created a Python script to pick 12 blocks from the pallet (figure 11) and place them in the circular positions of a clock face (figure 12). Each of the blocks is  $200 \times 200 \times 100$  microns.

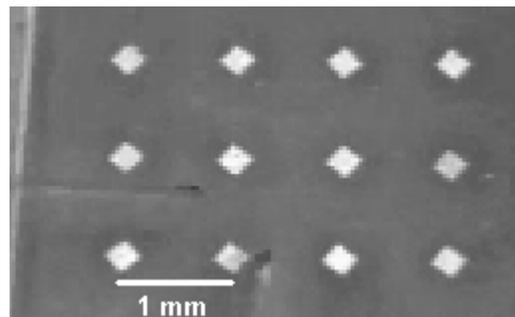


Fig. 11: Twelve blocks arranged 4 by 3 on a pallet with 1 mm spacing. The blocks are at  $45^\circ$  for easier grasping. Each block is  $200 \times 200 \times 100$  microns.

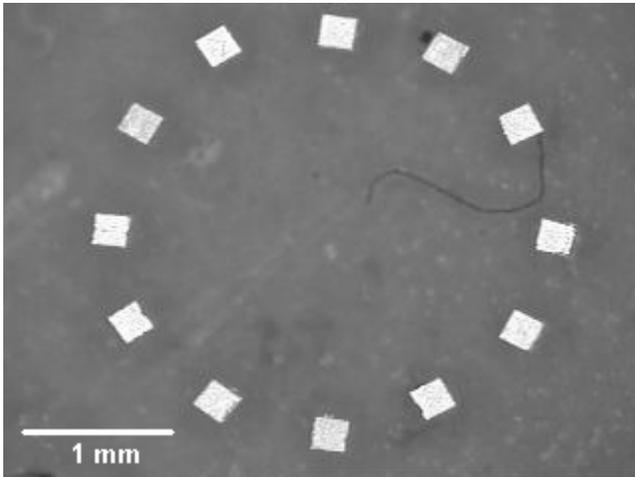


Fig. 12: Result of pick and place. Blocks on a clock face. The circle diameter is 3 mm. Each block is  $200 \times 200 \times 100$  microns.

Notice that the blocks have been rotated appropriately around the Z axis as well as placed in the correct position.

The pallet from which the blocks are picked is pre-arranged with blocks in a 4 by 3 rectangle, each separated by one millimeter. Each block is at  $45^\circ$  for the most reliable grasp. The location of a block on the pallet only need be approximately known because the Grasp algorithm can locate it precisely.

The Python script is shown in the appendix. It defines a few useful subroutines, including `relocate(i)` which relocates block `i` from the pallet to its position on the clock face. Finally, it commands:

```
for i in range(12):
    relocate(i)
```

which calls `relocate(i)` for each of the 12 blocks.

For long-term reliability testing, we created another script which locates the 12 blocks and then "unassembles" them by returning each to the pallet. One such cycle thus has 24 pick and place operations. Our best to date is that the system will complete almost 5 cycles before failure, for a total of 116 pick and place operations. The limiting factor in assembly time is the speed of the translating stage, which is 0.166 mm/sec .

## 5 Conclusion and Future Work

In conclusion, the ortho-tweezers microassembly system can robustly locate and manipulate microparts using force feedback. By combining rotations around the Z axis and grip axis, and by using the three degree of freedom translating stage, parts can be manipulated with full six degrees of freedom. By combining manipulation

primitives in a Python script, new manipulation routines can be defined and combined to perform complex assembly.

Our next task for the system is to place and glue strain gauges to make a new pair of ortho-tweezers probes using the ortho-tweezers system. Each gauge is only a millimeter in length, so assembly by hand under a microscope is tedious and error-prone. Using the system to assemble some of its own parts will further demonstrate its flexibility and reduce costs.

## 6 Acknowledgments

The authors thank Eiji Shimada, a visiting industrial fellow from NEC Corporation, Japan, for his work in developing the ortho-tweezers system.

## 7 Appendix: Pick and Place Script

Following is the Python script used to pick and place the blocks shown in figure 5. While we don't have the space to explain the script in detail, we include it to show its small size and how the manipulation primitives are used.

```
def palletLocation(n):
    result = Point3d (location(0))
    # Pallet is in rows of 4
    result.x += n%4
    result.y += n/4
    return result

def clockLocation(n):
    result = Point3d (location(1))
    result.x += 1.5*sin((n/12.) * 2.*pi)
    result.y += 1.5*cos((n/12.) * 2.*pi)
    return result

def clockAngle(n):
    # There are three different angles.
    return 3.14/2 - (n%3)*(pi/6)

def setTargetZ(target):
    seekXYArmDeflection(.3,.3)
    moveToZ (.5)
    moveToPoint(target)
    seekZSurface(1)
    target.z = stageZ() + .04

def seekAndGrasp(point):
    seekXYArmDeflection(.5,.5)
    moveToPoint(point)
    seekZSurface(1)
    moveDeltaZ(.04)
    grasp(.1,4)
    moveToZ (.5)

def place(target, angle):
    moveToZ(target.z + .5)
    moveToXY(target.x, target.y)
    seekZAxisAngle(angle,4)
```

```

moveToZ(target.z - .02)
ungrasp()
def relocate (n):
    target = clockLocation(n)
    setTargetZ(target)
    seekAndGrasp(palletLocation(n))
    place (target, clockAngle())
for i in range(12):
    relocate(i)

```

## 8 References

P.K. Allen, P. Michelman, K.S. Roberts, "A System for Programming and Controlling a Multisensor Robotic Hand," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 6, November/December 1990

A. Bicchi, "Hands for Dexterous Manipulation and Robust Grasping: A Difficult Road Toward Simplicity," *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 6, December 2000

R.C. Brost, "Automatic Grasp Planning in the Presence of Uncertainty," *IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, April 1986

B. Carlisle, K. Goldberg, A. Rao and J. Wiegly, "A Pivoting Gripper for Feeding Industrial Parts," *IEEE Conf. on Robotics and Automation*, San Diego, May 1994

A. Codourey, W. Zesch, R. Buechi and R. Siegwart, "A robot System for Automated Handling in Micro-World," *Proc. IEEE/RSJ Intelligent Robots and Systems*, pp. 185-190, Pittsburg, PA, August 1995

M. Erdmann, M.T. Mason and G. Vanecek Jr., "Mechanical Parts Orienting: The Case of a Polyhedron on a Table," *Algorithmica, Special Issue on Computational Robotics*, Vol. 10, No. 2, August 1993

R.S. Fearing, "Simplified Grasping and Manipulation with Dexterous Robot Hands," *IEEE Journal of Robotics and Automation* Vol RA-2, No. 4, December 1986

R.S. Fearing, "Survey of Sticking Effects for Micro-Parts," *IEEE Int. Conf. on Robotics and Intelligent Systems IROS*, Pittsburg, PA 1995

J.T. Feddema and R.W. Simon, "Visual Servoing and CAD-Driven Microassembly," *IEEE Robotics and Automation Magazine*, Vol. 5, (no. 4):18-24, December 1998

J.T. Feddema and T.R. Christensen, "Parallel Assembly of LIGA Components," *Tutorial on Modeling and Control of Micro and Nano Manipulation, IEEE Int. Conf. on Robotics and Automation*, Detroit, Michigan, May 1999

K. Goldberg, "Orienting Polygonal Parts Without Sensors," *Algorithmica, Special Issue on Computational*

*Robotics*, Vol. 10, No. 3, pp. 201-225, August 1993

S. Gopalswamy and R.S. Fearing, "Grasping of Polyhedral Objects with Slip," *IEEE Int. Conf. on Robotics and Automation*, Scottsdale AZ, May 1989

T. Kasaya, H. Miyazaki, S. Saito and T. Sato, "Micro Object Handling Under SEM by Vision-Based Automatic Control," *SPIE Conf. on Microrobotics and Micromanipulation*, Vol. 3519, pp. 181-188, Boston, MA, Nov. 1998

B.J Nelson, Y. Zhou and B. Vikramaditya, "Sensor-Based Microassembly of Hybrid MEMS Devices," *IEEE Control Systems*, pp. 35-45, Dec. 1998

A. Rao, D. Kriegman, and K. Goldberg, "Complete Algorithms for Feeding Polyhedral Parts Using Pivot Grasps," in *IEEE Transactions of Robotics and Automation*, (12)2, April 1996

D. Rus, "Coordinated Manipulation of Polygonal Objects," *IEEE/RSJ Int. Conf. on Intelligent Robot Systems*, pp. 106-112, Yokohama, Japan, July 1993

E. Shimada, J.A. Thompson, J. Yan, R.J. Wood and R.S. Fearing, "Prototyping Millirobots using Dexterous Microassembly and Folding," *Proc. ASME IMECE/DSCD*, Orlando Florida, November 5-10, 2000

M. Sitti and H. Hashimoto, "Two-Dimensional Fine Particle Positioning Using a Piezoresistive Cantilever as a Micro/Nano-Manipulator," *IEEE Int. Conf. on Robotics and Automation*, pp. 2729-2735, Detroit, MI, May 1999

A. Sulzmann, P. Boillat and J. Jacot, "New Developments in 3D Computer Vision for Microassembly," *SPIE Conf. on Microrobotics and Micromanipulation*, SPIE Vol. 3519, pp. 36-42, Boston, MA, Nov. 1998

J. Wiegley, K. Goldberg, M. Peshkin and M. Brokowski, "A Complete Algorithm for Designing Passive Fences to Orient Parts," *Assembly Automation*, Vol. 17, No. 2, August 1997

T. Yoshikawa, Y. Yokokohji and A. Nagayama, "Object Handling by 3 Finger Hands Using Slip Motion," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 99-105, Yokohama, Japan, July 1993

W. Zesch and R.S. Fearing, "Alignment of Microparts Using Force Controlled Pushing," *SPIE Conf. on Microrobotics and Micromanipulation*, Boston, MA, Nov. 1998

Y. Zhou and B.J. Nelson, "Adhesion Force Modeling and Measurement for Micromanipulation," *SPIE Conf. on Microrobotics and Micromanipulation*, Vol. 3519, pp. 169-180, Boston, MA, Nov. 1998