A RISC approach to Sensing and Manipulation

John F. Canny[*]
University of California, Berkeley
Berkeley, CA 94720
jfc@cs.berkeley.edu

Kenneth Y. Goldberg[†]
University of Southern California
Los Angeles, CA 90089-0273
goldberg@usc.edu

Abstract

This paper is about sensing and manipulation strategies using simple, modular robot hardware. RISC robotics is an attempt to fuse automation and robotic technologies. It uses traditional automation hardware such as parallel-jaw grippers and optical beam sensors, together with geometric planning and sensing algorithms. RISC systems should be cost-effective and reliable, and easy to setup and reconfigure. They should also be flexible enough to support small batch sizes and rapid changes in part design needed in forthcoming flexible/agile manufacturing systems. The RISC acronym, borrowed from computer architecture, suggests the parallels between the two technologies. RISC robots perform complex operations by composing simple elements. The elements may be individual light beam sensors, grouped together to form an array for recognition. Or a complex manipulation task may be performed via a sequence of grasp steps by different grippers specialized for acquisition and placement. This paper emphasizes three areas: (i) RISC sensing, primarily optical beam sensing (ii) RISC manipulation using simple parallel-jaw grippers or minimal configurations of fingers (iii) Computer-aided design of RISC workcells.

1 Introduction to RISC robotics

Borrowing an acronym [PD80], we chose the name RISC for our approach to manufacturing robotics [CG93]. In our case the acronym stands for **R**educed **I**ntricacy in **S**ensing and **C**ontrol. We choose it because it is suggestive of the design goals we are promoting. RISC robots perform complex operations by composing simple elements. The elements may be individual light beam sensors, grouped together to form an array for recognition. Or a complex manipulation task may be performed by a sequence of grasp steps by different grippers specialized for acquisition and placement. For example:

- Flexible part placement and orientation is performed using several acquire/transport/place steps with simple, specialized grippers. For example, a pick-reorient-place sequence may employ a parallel-jaw gripper for the initial grasp and a second three-fingered gripper with RCC collar for the final insertion. The choice of grippers is part of the planning process for the assembly.
- Where more degrees of freedom are needed, they can often be provided by temporarily using several manipulators together to form a single "virtual robot". Pan/tilt tables can also be used to provide rotational motion not otherwise available.
- Sensing is performed by sparse sensors, such as optical beams, and low-resolution tactile. A powerful, accurate sensor can be built using 3 to 6 on-off beam sensors, which can recognize and localize parts to 0.001 inches in milliseconds.
- Sensors and actuators can be combined to yield very flexible active sensors. A reflective beam sensor can be mounted on the robot's end effector and used to build models for later recognition or for inspection or feature localization in tight spaces.

2 RISC Principles

2.1 Assembly: Breaking it down

First we break some abstract assembly tasks into chunks that simple hardware can manage.

2.1.1 Task decomposition in time

The "pick-and-place" operation is the building block of what we think of as robotic assembly. A part is picked

---

[*] Supported by a David and Lucile Packard Foundation Fellowship and by NSF Presidential Young Investigator Grant IRI-8958577 and NSF grant IRI-9114446

[†] Supported by NSF awards IRI-9123747, DDM-9215362, and by a National Young Investigator Award. Also funded by a grant from the State of California's Office of Competitive Technology and by equipment grants from Adept Technology and Silma, Inc.

up off a table, conveyor or pallet, moved to its destination, and mated there somehow as part of the assembly.

The object will almost certainly have some uncertainty in its pose when grasped, even if it is lying on a pallet. This uncertainty can be reduced with local sensing, or by the grasp operation itself. For example, parallel-jaw grippers are good at grasping with small amounts of uncertainty, and leave only one degree of uncertainty after the grasp.

The requirements for placing the part in its destination are quite different. There are geometric constraints that require the grasper remain clear of the assembly while it moves the part home, and it is very likely that a suitable grasp for this second phase cannot be applied when the object is picked up. So a regrasp step is needed [LP87]. To deal with the uncertainty between the grasped part and the rest of the assembly, either a second type of local sensor (see next section) or compliance must be built in to the placer.

Rather than building all these capabilities into a single manipulator, a better solution applies two grippers to the task. Imagine a cylindrical shaft being grasped from a pallet by a parallel-jaw gripper. The grasp action itself will constrain the cylinder pose except in the direction of the cylinder axis, and perhaps some vertical uncertainty.

A second 3-jaw gripper will grasp the now-exposed cylinder end. This grasp operation removes still more uncertainty, although some remains along the cylinder axis. However, in the figure, the gripper is shown with a local sensor, a single light beam sensor, that detects the cylinder end as the gripper fingers pass by it. This allows it to very precisely constrain all the degrees of freedom of the cylinder. The 3-jaw gripper may be attached to an RCC passive compliance, or it may be fitted with another local sensor to precisely locate a destination feature, to facilitate the final placement of the cylinder.

Not all pick-and-place tasks are this complicated. It may be possible to grasp the object directly with the final placement gripper, which is obviously a better solution. But these cases pose no difficulties for robotics or traditional manufacturing. It is precisely the difficult cases which drive us (the robotics community) either toward more complex manipulators, or in the framework propsed here, toward **multi-step solutions using simple manipulators**.

We believe that manipulation with multi-step strategies is very general, and we are attempting some ambitious examples to verify our expectations. We do not yet know how many new types of gripper will be needed in the repertoire of a RISC workcell. We expect that 2 and 3 jaw parallel grippers and various types of fastener drivers (e.g. screwdrivers) will do 90% of the work, and one or two special purpose grippers may be needed for a new assembly. How the standard grippers are selected and how the special purpose grippers might be designed is discussed later in section 2.4.

2.1.2 Sensor/Actuator decomposition: Units of Sensing and Actuation

In most manufacturing (leaving aside food handling, a rapidly-growing but special subset with its own special problems), part shapes and other physical properties are highly constrained. Were they not, assembling the parts at even reasonable tolerance would be impossible. This is an essential property of the manufacturing environment, and the key property that the RISC approach exploits.

The real uncertainty then, rests in the poses of the parts. Even a few millimeters is a huge amount of variation for an insertion at a clearance of 50 microns. For singulated (not contacting others) parts on tables or conveyors, there are 3 degrees of freedom for each stable configuration. If the part is placed on a tilted table so that it slides against one wall, it retains only one degree of freedom, its position along the wall. Parts in bins are subject to a full 6 degrees of freedom. But even there, if the part shape is known, a sensor that returns 6 values can determine the pose down to a few possibilities, or uniquely. If a sensor returns more measurements than are needed for pose determination, the extra values can be used for recognition, because the redundant sensor readings are very unlikely to be consistent with more than one object.

Most of the time then, a sensor needs to provide information about from 1 to 6 degrees of freedom. In section 3.1, we will describe sensors constructed from small arrays of light beam sensors. These sensors can be very easily tailored to applications, because the basic sensor unit is a single light beam, rather than an array. In our experiments, we frequently use these arrays of sensors, but we also frequently use a single beam or reflective sensor to constrain one or two degrees of freedom. An example is the reflective hole sniffer described in [PC93], which is similar to figure 1 . Because the pose information sought from the sensor consists of so few real values, it makes sense to use a description of sensors that makes clear their dimensionality, so that can be related to the information required for each task.

We propose here that a sensor should be viewed as comprising some number of **"units of sensing"**. For the beam sensors, a unit of sensing is most naturally a single beam. For other sensing technologies, a unit of sensing should correspond to a single real value provided by the sensor. So array sensors like cameras and tactile sensors comprise roughly 256k and several hundred units respectively. We propose this point of view for several reasons: (i) It makes explicit the amount of information that the sensor interpretation algorithms must process, in relation to the number needed for pose determination (ii) It indicates the approximate initial and maintenance cost of the sensor. This does not work across technologies, because camera pixels are very much cheaper than beam sensors, but it does serve to compare high-resolution and low-resolution sensors of the same type. (iii) It supports a task-specific design of a sensor that provides enough information for the task at hand without overkill.

For similar reasons it makes sense to break actuators down into **"units of actuation"**. These will normally correspond to the degrees of freedom of the actuators. This allows every device that causes or constrains part motion to be considered. Not just robot arms and grippers, but fixtures, conveyors, AGVs, and various types of feeders. All these devices affect the 3 to 6 degrees of freedom of a part.

The advantages of this point of view are (i) It indicates the approximate complexity of controlling the actuator (ii) It is a good guide to setup and maintenance cost (iii) It allows a measure of the "efficiency" of the actuator. i.e. How many actuator degrees of freedom are used, and how many part degrees of freedom are constrained. (iv) It provides the right uniform vocabulary for CAD tools for an entire assembly or manufacturing system, not just a robot workcell.

2.1.3 Modularity, or What is a Robot?

The RISC approach then, views sensors and actuators in terms of basic units, corresponding to real values sensed or to degrees of freedom controlled. So it resonates with the (often derided) Japanese view of robots as any device with degrees of freedom that affects part motion or shape. As we shall see in the next section, degrees of freedom can be grouped together and regrouped in a task-dependent way. This grouping in general has nothing to do with how the degrees of sensing or actuation freedom are grouped physically.

Contrast this with the classical approach to robotic assembly, which stresses the need for robots with a full 6dof, a general purpose sensor, such as a camera or rangefinder, and a dextrous grasper. Good mechanical design practice mandates the use of vertical assembly steps whenever possible. This leaves two or more of the robots degrees of freedom unused during these operations, and they cannot be used for something else. The grasper will likewise be using a fraction of its capability most of the time, and there is no way to reallocate the unused degrees of freedom. Of course, there are some assembly steps that require motion out of the vertical plane, but as we shall see later, there are ways of adding these degrees of freedom in a more economical way.

In like manner, the camera or rangefinder can observe only one portion of the workspace at a time. Its full power to provide a rich description of the image is always available, but almost never used, since the part types and geometries are known. But there will be some amount of pose uncertainty between each feeder and gripper, and between any pair of parts to be mated. This uncertainty can be largely eliminated by distributing beam sensing units throughout the workspace. Our RobotWorld workcell now has dozens of fixed and moving beam sensors distributed through it. The hardware needed to run all this is the sensors themselves, and one IO board. Cameras and image processing hardware are becoming ever cheaper, smaller and faster, but it is still rare to see more than one in a workspace, let alone the half-dozen that would be needed to provide the same level of sensing as our beam-instrumented workcell. Providing sensing in small chunks allows a much better match to structured environments like manufacturing workcells.

2.2 RISC Assembly: Building it up

2.2.1 Merging Sensors and Actuators: Instrumented Actuators

Positional uncertainty is an inescapable fact of life in assembly. It exists between parts and part handlers, before and after the object is acquired by the handler. And it exists between the handlers, i.e. between the feeders, fixtures and manipulators, so that when a part passes from one to the next, its positional uncertainty will increases unless care is taken to reduce it. In RISC we use *local sensing* to reduce uncertainty between actuators. A local sensor is a sensor mounted on a handler so that it can accurately localize parts that the handler deals with. For example, we use cross-beam sensors mounted on conveyor belts to determine the very uncertain pose of objects coming down the conveyor. And we have both cross-beam and reflective to accurately center over a part to be grasped, assisting in part acquisition. The reflective sensor allows the end-effector to accurately locate a feature for an insertion step, as shown in figure 1, assisting in part placement.

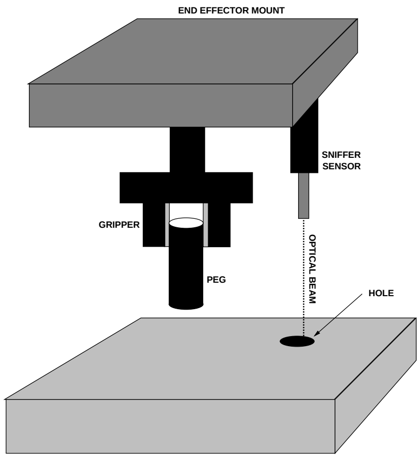

Figure 1: An instrumented gripper

A gripper with cross-beam sensor mounted on it may function as a part manipulator, or purely as a position sensor. Occasionally, this kind of sensor is needed to localize a feature potruding from a partial assembly, e.g. an alignment peg, while no manipulation of this feature is needed. We also plan to use this kind of sensor to perform model acquisition for the cross-beam sensor. By making 2n passes over the object, such a sensor can compute a complete description (from the point of view of what another cross-beam sensor can see) of an object with n sides. So local sensing blurs the distinction between sensors and actuators.

Local sensing deals with uncertainty wherever it arises. If used extensively, workcells with local sensing are much, much simpler to set up. No elaborate calibration between handling devices is needed. Only enough to ensure that parts fed by one device can be seen by the next device's sensors. This is a very important aspect for a modular system. It *should* be a simple process to add new degrees of freedom or sensors to the workspace of a flexible workcell. But the state of the art in most industrial vision systems is that after precisely calibrating the camera frame, the system must be trained by placing *each part to be recognized* at many different poses throughout the camera's field of view to compensate for optical distortion. New actuators require a less elaborate, but still tedious calibration phase to achieve their potential positioning accuracy. With local sensing, calibration can often be avoided completely, because devices measure and compensate for part position every time they are used. Or it can be done by a simple cross-calibration step, where a sensor from one device localizes a sensor of the other, thereby accurately linking their coordinate frames. This was done in our peg-in-hole insertion routine [PC93], which achieves 25 micron tolerance insertions without chamfering at 99% repeatability, without prior calibration.

Thus the most effective grouping of sensor and actuator units in a modular workcell is not into more complex sensors and actuators, but into **instrumented actuators**. An instrumented actuator may still be simple, e.g. have only one degree of freedom (a conveyor), but with a cross-beam sensor added, it becomes a powerful feeding/localizing module.

2.2.2 Merging Manipulators, Feeders and Fixtures: Virtual Robots

Once one steps back from the view of robots as 6-axis universal positioners, a huge variety of possibilities opens up for forming novel liasons between actuators, fixtures and other types of passive elements to effect part pose. Matt Mason and his students have created a science of part pose control through sliding motion. It is difficult to say whether the tilting trays, sliding fences and barriers they use should be called robots, feeders or fixtures, but they clearly have aspects of all three. We have implemented our RobotWorld controller software with this view in mind. See figure 2 for a diagram of RobotWorld. RobotWorld has multiple 4dof cartesian "placement modules" in a single workspace. Every actuator in the workspace, including RobotWorld placement modules, grippers, a conveyor, piggyback degrees of freedom for the modules, and a vice, are broken down as separate degrees of freedom. They may be joined in any combination to form "virtual robots" which are controlled synchronously. Applications of this that we have done or plan in the near future: (i) two modules each with a soft finger attached acting as a two-finger gripper, (ii) module and conveyor moving synchronously (iii) pan-tilt table (we dont have one yet) and 4-axis RobotWorld module together giving a full 6 degrees of freedom relative to the part.

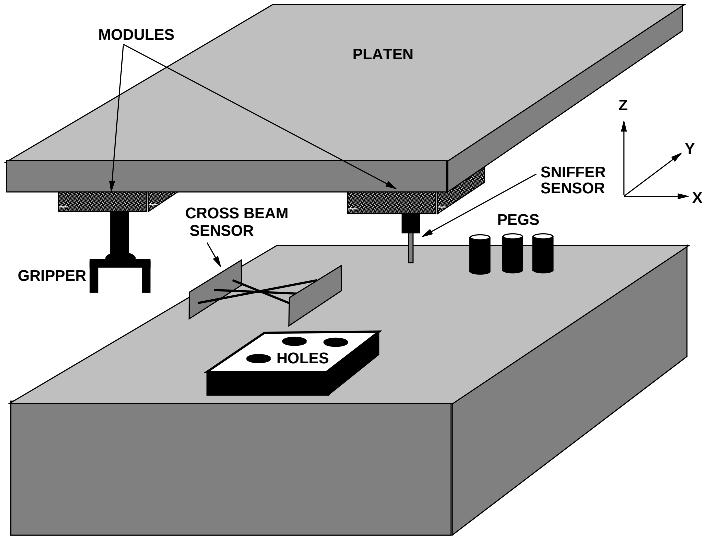

Figure 2: The RobotWorld multi-robot system

2.3 Discussion

In proposing this paradigm we are trying to choose technologies appropriate to the manufacturing environment, and which are known to perform well there. We are seeking a best middle ground between manufacturing and robotic technologies. Rather than applying complex, general purpose technologies to manufacturing, we have found that the sensors and actuators already used there are capable of great flexibility when used in a modular way with appropriate algorithms. They have the advantages of:

- Reliability. RISC sensors and actuators have fewer components so little can go wrong. They are also modular, so failed components can be replaced from standard stock.

- Software Simplicity. A side-effect of simple hardware is that control and sensing algorithms *tend* to be simple. This eases learning, reduces set-up time, and most importantly, facilitates fixes in the worst-case scenario of a bug causing the line to crash.
- Easy Set-up and Reconfigurability. Discussed earlier. As well as hardware and software simplicity, a consequence of using instrumented actuators as the workcell building blocks, which eliminates much or all of the initial calibration.
- Low cost. The hardware costs are certainly low, but initial hardware outlay is often a minor consideration in the overall economics of a manufacturing workcell. But the ongoing savings because of the points above are likely to be much more significant, because they continue to accrue over time.

In the time since the report on RISC robotics [CG93] appeared we have been able to gather some reactions from other researchers. The discussion that resulted led to rapid exchange of ideas and eventually to clarification and better understanding of the approach on both sides. Here are some typical comments:

- **RISC is just another acronym, where are the new ideas?**

  We have certainly been influenced by earlier work stressing the importance of using simple hardware for manufacturing. Especially the work of Whitney [Whi86a] who cogently argued for the use of simple, dedicated systems for manufacturing. And the work of Mason [Mas86] and others on manipulation by pushing certainly has gone far in the direction we are taking up. But there has not been a systematic body of work studying manipulation and sensing algorithms for simple hardware. Like most "paradigms", the core ideas are simple to state, intuitive and perhaps even obvious. If they are obvious, then others should take up the work without the RISC acronym, or with a different one. Whether the acronym stays or goes is not important, as long as the work is done. Right now the schism between academic research on robotics and what industry is asking for and needs is ever-widening. We feel it is very important to address those needs and move the two communities closer. The RISC approach is our best attempt to articulate a direction to do this.

- **The problems are not new, e.g. optical beam recognition: vision researchers have solved 2D model-based recognition long ago**

  Some of the problems we have studied look deceptively simple. The very sparse data provided by the beams is particularly challenging. Imagine trying to tell a hex nut from a similar-size washer from an overhead silhouette, which is the 2D model-based vision problem. Now imagine trying to do the same thing using 6 arbitrary points (not vertices) on the boundary of each, which is the beam recognition problem. The beam data requires 3 point matches before pose can be determined, and grouping is impossible. 2D grey-scale recognition needs only a pair of edges to establish pose, and these may be grouped to share a vertex, so correspondence requires only a single compound feature.

  Manipulation with sliding fences and gripper jaws was already an active area of research, and there are many interesting problems to be solved. The distinction between part feeders and sensorless manipulators is becoming blurred[GME91], and with RISC sensing, there is no longer a reason to avoid sensing because of computational or time considerations.

  One of the most intriguing aspects of RISC is that it blurs even the distinction between planning and workcell design. RISC sensors and grippers have easily characterized behaviour, and this makes possible symbolic description of their constraints and capabilities. At the simplest level, this suggests choosing the best gripper based on grasp quality considerations or the best sensor of those available in the workcell. At a more global level, this may involve laying out the entire workcell so as to minimize the number of degrees of freedom needed to complete the anticipated manipulation steps. Or it may involve designing a sensor to have the best possible positioning accuracy, while avoiding contact with parts and grippers during manipulation. This research direction is rich with possibilities.

- **Manufacturing Systems are already using RISC methods**

  This is true only of the hardware itself. RISC robotics is about adding advanced planning and design algorithms, which have traditionally been applied to complex hardware, to simple manufacturing hardware. Rapid deployment manufacturing is far from a reality. Feeder design and debugging remains a tedious and costly process. What's missing are powerful software tools to allow rapid design and debugging, and versatile online sensing to ease the burden on part feeders to produce near-perfectly oriented parts.

- **The approach should be validated experimentally before claims are made about its usefulness**

  We are in accord with the spirit of this comment. The goal of our experiments with sensors and actuators is to build a library of strategies and a graphical user interface that support rapid development of an assembly plan. We have reported results on peg-in-hole insertion in [PC93]. Our plan is to perform a complete assembly of a mechanical device, namely a model-aircraft engine. The sensing/manipulation hardware has already proven itself in manufacturing environments for simple tasks, so we are not taking a great risk. We have found that we get remarkable accuracy with off-the-shelf hardware (0.001 inch with inexpensive optic-fiber beam sensors). And by building up a repertoire of sensor elements, we can deal with special sensing problems that are difficult or impossible with general-purpose sensors. In the course of the miniature engine assembly, we hope to uncover some of these problems, and expand our collection of sensor building blocks.

2.4 RISC and Design

The RISC approach to manufacturing replaces general purpose sensors and actuators with special-purpose, modular hardware. A dextrous arm/ manipulator/ camera combination can be thought of as an interpreter for assembly plans. The assembly sequence can be changed with minimal effort by changing the plan. A RISC workcell for a particular task can be thought of as a **compiled** version of an assembly plan. Many plan steps become **design choices** instead. e.g. planning finger placements for a stable grasp maps to choosing modular fixture placements on a drilled worksurface. Planning a series of part motions maps to designing actuators with enough degrees of freedom and stroke to perform those motions. And choosing a series of placements of a movable camera maps to placing appropriate beam sensors at all those sites.

Clearly, we do not want to overconstrain the functionality of the workcell in the quest for efficiency. But as we have argued here, and through our experiments, simple hardware is capable of great efficiency *and* flexibility when used in the right way. The key to exploiting it fully is good design. If a workcell is to be truly flexible, that is, if it is to be rapidly configured and re-configured, the design must be supported by powerful CAD tools.

We are now in the course of implementing a very general algebraic constraint satisfaction system to deal with mechanical design problems. This system is the product of 5 years of work on practical algebraic algorithms. A preview of the system was presented in an ESPRIT workshop on motion planning in Rodez, France in March of this year [Can93]. Most of the algorithms that comprise the system had not been implemented before. In early tests, we have found that each contributes one to several orders of magnitude of speedup over other methods. Overall, the system should provide a qualitatively higher level of problem-solving ability for non-linear optimization problems.

While this system is not specialized to design of RISC systems, RISC does provide many well-defined and interesting design problems, and an excellent testbed for the algebraic system. Because of the simplicity of the sensors, there are not too many design parameters. When designing a standard 3-beam cross-beam sensor, the parameters are length of each beam, the angles between them, and the height above the work surface. The constraints of cross-talk, sensitivity and adequate spacing are easy to describe in algebraic form.

A more challenging problem is to design a custom sensor to orient a highly symmetric part. Standard vibratory feeders can orient the vast majority of part geometries reliably, but some parts cannot be fed this way. Instead they require special packaging or handling that increases the cost of the workcell enormously. One solution is to use a standard feeder to reduce the possible orientations to a small subset, and then synthesize a sensor design to discriminate between these. The design tool must choose a placement of a beam so that the beam breakpoints are different for different object poses, and so that the sensor remains clear of the part in all possible poses.

The design of custom grippers for difficult parts is another important subproblem. A starting point is to use our optimal grasp planner to choose some finger placements, and then check various finger geometries for collision with the rest of the object. Ideally, the fingers should be drivable from a standard two or three-jaw base.

In the long term, our goal is to integrate the RISC workcell CAD tool with the CAD tool for the device to be assembled. At the very least we will attempt to characterize the most important ease of assembly considerations (relative to a RISC type workcell) that should be presented to the device designer, and what information he or she should provide to the workcell designer.

2.5 Case studies

The real test of our approach is whether it can handle the full assembly of a mechanical device. We have chosen first to assemble a mid-sized model-aircraft engine. We chose it because (i) it is a good size for RobotWorld's workspace, (ii) it has modest complexity, with about two dozen parts (iii) in spite of this, it has some very challenging subproblems. The cylinder/piston fit is essentially zero tolerance. These engines have no piston rings, so they rely on the tightness of this fit for their compression. Pistons and cylinders are finished to an extremely smooth finish by honing, and then hand matched to get suitable fit. There are non-vertical insertion steps, near but not perfectly cylindrical parts, and non-rigid subassemblies.

Our goal is to make the assembly program parametric so that it can assemble several other sizes of engine that have roughly the same makeup. We will structure the program into subroutines that can be re-used for other assemblies. We have already acquired a number of these from previous assembly demos. We are seeking other good test problems, and we hope to attempt several other realistic assemblies in the future. It should be easier to do this as our assembly routine library grows. We expect to run into unforseen problems, and these should lead us to expand our vocabulary of simple sensors and actuators.

3 Related Work

An overview of the RISC approach and some open problems is given in [CG94]. Many others have argued for simplicity in the factory. As Dan Whitney argued in his 1993 Keynote Address, it is vital to consider the robot in the context of the assembly environment [Whi93]. Nevins and Whitney [NW78] stressed the distinction between structured and unstructured environments, and Whitney [Whi86b] pointed out the what is often an opposition between flexibility and efficiency in manufacturing. In structured environments such as factories, where repetition is the rule, the emphasis is on efficiency. The idea of modular manufacturing systems is gaining in popularity [Rog93, ea93, Hoi93].

There are many examples of research that can be viewed as examples of RISC : One example is the RCC wrist that achieves compliant peg-in-hole assembly using passive mechanical elements rather than active feedback in a general-purpose manipulator [Dra77]. Recently, Goswami and Peshkin showed how to "program" such a wrist to achieve desired behavior by changing the damping constants of its passive components [GP90].

One of the recurring themes in our approach is the role of mechanical compliance in lieu of sensing. Mason [Mas91] and his students have developed a science of part pose control through sliding motion, demonstrating that it is often possible to replace sensors with mechanical solutions. Others, such as Erdmann and Donald are studying ways to reduce the complexity of sensors. Other applications of mechanical compliance are explored in [Bro91, PS88].

An important principle in computer vision is the exploitation of *domain constraints* to simplify algorithms. Rather than representing all conceivable images, using domain constraints one describes the simpler space of images that can occur that satisfy various physical and structural constraints. RISC is the natural extension of domain constraints to hardware. A part on a table has only 3 degrees of freedom, so a sensor that provides 6 numbers is entirely appropriate.

Kanade used the term "KISS" (Keep it Simple) to describe a collection of recent results in machine vision where simple processors at each pixel permit extremely fast update rates. Here, the correspondence problem can be avoided since motion between frames is greatly reduced [Kan92]. While there is some relation to our use of simple elements, the primary difference is that Kanade applies fixed arrays of simple elements to unstructured scenes rather than planning for repetitive operations.

Recently, Jia and Erdmann [JE94b] gave an algorithm for localizing a polygon by *inscription* within two cones of light beams. In [JE94a], the same authors considered the problem of sensing by testing which of a set of points is inside or outside a polygon. They showed that classifying a set of objects based on this data is NP-complete, but gave an approximate algorithm that works in polynomial time.

3.1 Sensing

We have studied two types of optical beam sensor. The first is called a "cross-beam sensor", see figure 3. When an object passes through the apparatus, the cross beams perceive a horizontal cross-section of the object. The times when the beams are broken and unbroken are recorded, as shown in figure 4.

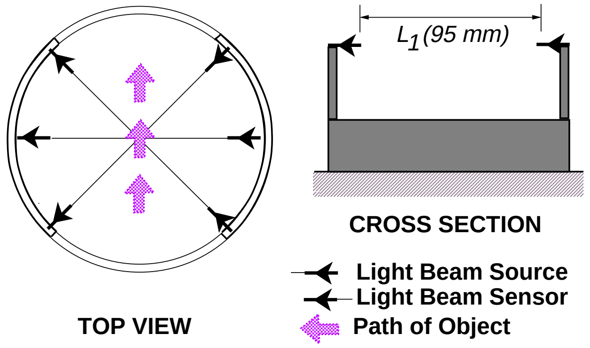

Figure 3: The usual cross-beam sensor configuration

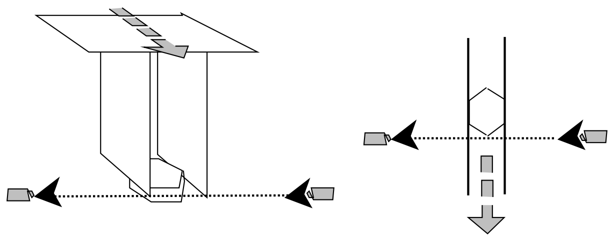

Figure 4: A *critical* point occurs when the object breaks or unbreaks the beam of light

With 3 beams, the breakpoints define a hexagon bounding the object cross-section (6 real values). In spite of the coarseness of this information, because the beam measurements are so precise (~ 25 microns), the pose can almost always be determined unambiguously. Because the measurements are redundant, 6 measurements versus 3 degrees of freedom, the data can actually be used for recognition. A linear-time geometric algorithm to do recognition from beam data is described in [WCM93]. It works by computing the *diameter function* of the object, which is the distance between highest and lowest points on the object as a function of its orientation. The implementation described there takes a few milliseconds to recognize and compute pose. Also described there is a hash table version which takes a few microseconds to accomplish the same thing. The cross-beam data is particularly well-suited to table lookup because the effective table dimension is only one. So with data quantized to 1000 values, the table takes up a few thousand words of memory.

The cross-beam sensor relies on a consistent horizontal cross-section to accomplish its task. It does not work for flat parts. For these we use a parallel-beam sensor, which usually uses reflective elements. A parallel-beam sensor is shown in figure 5.

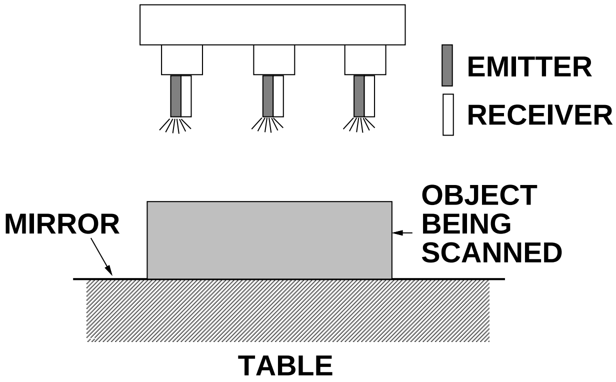

Figure 5: A reflective parallel beam sensor. Relative motion between sensor and part is normal to the page

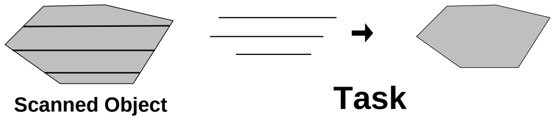

Figure 6: The parallel-beam scan data determined from an object's shadow

The scan data from the parallel-beam is particularly difficult to deal with because even using relative measurements, the data still depend on two of the object's degrees of freedom, unlike the cross-beam sensor which depends only on one. Indexing schemes generate lookup tables whose effective dimension is 2, and are consequently very large.

Our first approach was to use a geometric algorithm for matching, and to beat the $O(n^3)$ bound for the alignment method applied to this problem. In [WC93] we described an $O(n+A)$ correspondence algorithm for objects with convex polygonal silhouettes, and an $O(n^2 \log n + A)$ algorithm for objects with non-convex silhouettes, where n is the object's complexity and A is the total number of feasible matches. Typically for convex objects, A, the total number of matches is $O(n)$. The worst case for convex objects is $O(n^2)$, and the worst case for general objects remains $O(n^3)$ although it is typically much lower.

Our second approach used pre-computed indexing tables [WC94b]. These tables contain all feasible sensor values consistent with a model. At run-time, actual beam or scanning sensor values are used as indices, and the table provides matching information between model and image features. The match info is then used for accurate calculation of object pose. One novelty in [WC94b] is the data structure used to store the data, which exploits its coherence and uses less space than hash tables.

3.2 A RISC Manipulator

Perhaps the least complex manipulator is the parallel-jaw gripper, having one degree of freedom with binary pneumatic control Although widely used in industry, conventional wisdom holds that these grippers lack versatility [McK91]. With a minor modification, however, these grippers can be used to recognize and orient an important class of industrial parts.

3.2.1 The Modified Gripper

The quality of a grasp configuration depends on many factors including the orientation of the part with respect to the gripper. This orientation may not be known precisely or may be disturbed by the act of grasping. For the parallel-jaw gripper grasping polygonal parts, Brost [Bro88] defined a grasp as stable if at least three vertices of the part are in contact with the gripper jaws and any further closing of the gripper would deform the part; see Figure 7.

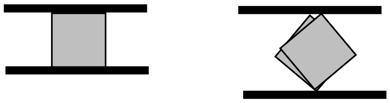

Figure 7: The grasp configuration on the left is stable; those on the right are not.

Unstable grasp configurations result from friction between the part and the jaws. This suggests that it may be desirable to eliminate friction between the object and the jaws. One approach is to coat the jaws with grease, but this has the disadvantage that the object will slip when the gripper is lifted out of the plane. We can achieve low friction in the plane of the object but high friction orthogonal to the object by mounting a sliding plate (linear bearing) on one jaw. The inner surface of both jaws is covered with a high-friction material such as rubber. See Figures 8 and 9.

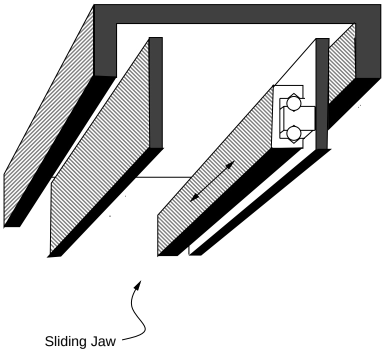

Figure 8: The modified gripper with sliding jaw. (Based on drawing by Ben Brown).

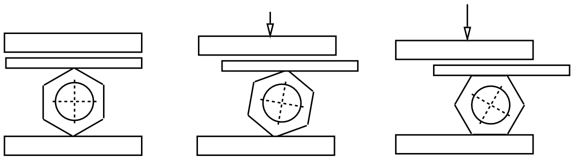

Figure 9: Time-sequence of grasping with the modified gripper. (1) As the two outer jaws close over a typical object (hex nut), horizontal forces cause the sliding jaw to translate to the left (2) until the object is gripped in a stable configuration (3). (Based on drawing by Ben Brown).

To achieve a stable grasp, we close the jaws as far as possible without deforming the object. Gripper forces will cause the sliding bearing to translate until the part rotates into a stable orientation. We built a prototype of this gripper using an off-the-shelf linear bearing with a rubber band to provide spring force and a dab of grease to provide damping. We experimented with several object shapes grasped randomly in two sets of 250 trials. Without the bearing, approximately half the grasps were stable. With the bearing, every grasp was stable. Although any physical bearing experiences some friction, we conjecture that vibration arising from the gripper drive is sufficient to dislodge unstable orientations, so that the mechanism is effectively frictionless.

One of the primary advantages is that this modification requires no additional sensors or actuators; a low-cost and lightweight linear bearing can be easily retro-fit to any parallel-jaw gripper without requiring interface software [Gol91].

3.2.2 Recognizing Parts

The parallel-jaw gripper described above can be used to recognize parts by measuring the distance between the jaws, say with a linear potentiometer. This is similar to using light beams as described in Section ??; the difference is that in this case closing the jaws causes the parts to rotate into a new configuration. For a given set of k parts with constant cross section (2.5D parts), we consider the following two problems: (1) given a set of measurements derived from random grasps of one part, decide which part was grasped. (2) find a sequence of grasp angles for the gripper, conditional on measurements, for efficiently recognizing parts from the given set.

For the first problem, since more than one part may give rise to the same diameter and the diameter sensor may be corrupted by noise due to surface compliance and backlash, we can use a Bayesian decision procedure to estimate the most probable part Since the set of grasps is random, we can assume that prior to each grasp, the part's orientation with respect to the gripper has a uniform probability distribution on the set of planar orientations. Note that each stable orientation of a given part corresponds to a minimum in the part's diameter function (see Figure ??). Thus the prior probability for each measured diameter can be derived in time $O(n)$. This becomes a conditional probability when considering a set of parts. Lacking any information to the contrary, we might assume that initially, each part is equally likely. After each measurement, the posterior probability is computed. After all measurements have been considered, we can decide on the most likely part. This method can also be adapted to allow for sensor noise using a Gaussian error model. For details see [KG92].

Of course we expect to achieve better performance by tailoring the grasp strategy to the geometry of parts in the set, as shown in figure 10. This, the second problem stated above, is the planning problem. As mentioned earlier, [RG94] showed that some parts cannot be distinguished by measuring diameter alone. But if we restrict attention to parts that are distinguishable, we can find optimal strategies by considering cliques in the following graph. Let $G = (V, E)$ be an undirected graph such that each vertex corresponds to a stable orientation from the given set of parts. Let $n = |V|$. We construct an edge between any two vertices with the same diameter. For each edge, let $R(e)$ be the set of gripper orientations that would disambiguate the neighboring vertices in a subsequent grasp. Let G have m edges. We can construct G in time $O(n^3)$ and it can be partitioned into disjoint maximal cliques (connected components) in time $O(n + m)$.

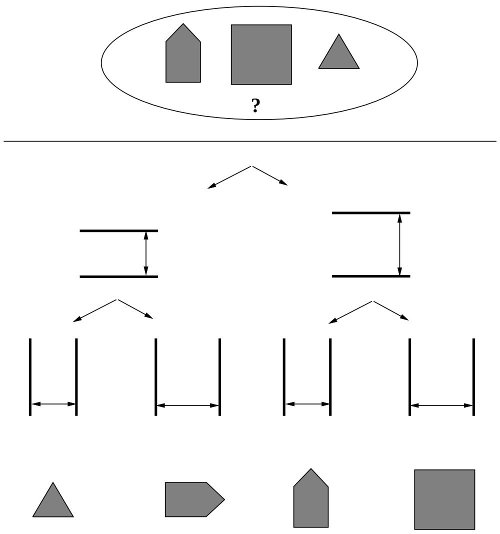

Figure 10: A grasp plan for distinguishing the three parts shown at the top.

Each possible gripper measurement identifies one of the cliques in G. If the clique contains stable orientations from one part, we are done. If it contains stable orientations from exactly two vertices connected by edge e, we pick a gripper angle in $R(e)$, and regrasp to disambiguate between the two associated parts. However, if the clique contains more than two vertices, we look for a gripper angle in the intersection of $R(e)$ for all edges in the clique, $\bigcap_E R(e)$. If this intersection is empty, we require more than one additional grasp to identify the part. In [RG94] we give two planning algorithms. The first runs in time $O(n^2 \log n)$ but may not generate the shortest plan. The second finds the shortest plan but may require time exponential in n in the worst case because it considers all possible partitions of each clique. In either case, the resulting plan will never require more than n grasps. This algorithm can also be adapted to account for measurement noise.

3.2.3 Orienting Parts

Perhaps surprisingly, it is possible to achieve a desired final grasp configuration *without sensors*. Goldberg [Gol93] describes an algorithm for orienting polygonal objects using the parallel-jaw gripper to grasp and ungrasp at a prespecified sequence of angles depending on object geometry. That is, for any additional stability criterion that prefers one of the stable grasp configurations over the others, we can achieve it using the compliant motion algorithm. The planning algorithm finds the shortest such sequence for any n-sided object in time $O(n^2)$. This algorithm has recently been extended to curved parts [RG92].

3.3 Fixture Design

The task of immobilizing a workpiece via mechanical devices, commonly called fixturing or workholding, is an essential problem in manufacturing. Machining fixtures must handle very large forces (20KN), whereas assembly fixtures handle smaller forces (50N). Fixture apparatus design is more a craft than a science. Without geometric analysis, a fixturing expert system is capable only of describing “types” of fixturing components, not the positions of the fixtures and the object.

As a first step towards designing an analytic fixture planning system, we have analyzed a nontrivial task: Automatically placing modular fixture elements on the jaws of a *fixture vice*, a device commonly used in woodworking. Each jaw has a flat surface to which pegs can be attached and the jaws can open or close, as shown in Figure 11. The fixture vice possesses the minimum number of degrees of freedom necessary (one) to deal with workpiece variations. Theoretically, it can immobilize any generic two and a half dimensional object. It could also be used as an *adaptable* gripper. The fixture vice is based on three mechanical devices: pegs, a fixture table, and a vice. It combines simplicity and efficacy. Vice contacts can only occur at vertices, but fixture vice contacts can occur anywhere on an object. Fixture vices can hold objects without crushing corners, and can immobilize objects using internal holes. The algorithm described in [WC94a] will enumerate all peg placements that achieve *force closure* when the jaws are closed. That is, any planar force or torque on the object is resisted by the pegs. Four contacts are necessary for force closure in the plane, and the degree of freedom in the vice jaws is needed to move all four pegs into simultaneous contact with the object.

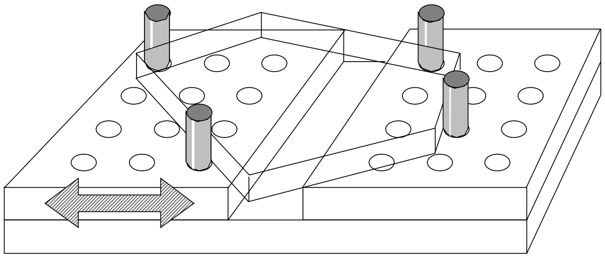

Figure 11: A fixture vice consists of two fixture table jaws capable of translating in x.

Another fixturing device with similar properties was proposed independently in [BG94]. This vice uses a single movable fixture to give 4 simultaneous contacts. Their algorithm enumerates peg placements, this time for the remaining 3 pegs which fit on a modular surface.

3.4 Other RISC Manipulation

In [FC92] we presented some general criteria for optimality of force-closure grasps. The criteria measure the ratio of external forces to the finger forces needed to resist them in the worst case. The criteria have a simple geometric interpretation in the space of generalized forces. Various metrics can be used to define the magnitude of the external force, so that the grasp quality measure can be task dependent.

In [LC91, LC92] we described an algorithm for efficiently computing distance between polyhedral objects. For convex objects, the algorithm works in expected constant time when used incrementally. It maintains the pair of closest features between the two objects, and is especially well adapted to incremental use. We are adding impact and free-body dynamics and to the distance code, so that we can simulate the APOS and bowl feeders.

In [PC93] we described an implementation of a peg-in-hole insertion strategy that used two types of beam sensor to avoid prior calibration. The strategy works for any size of circular peg, in fact the software does not know the peg size, and achieves 25 micron tolerance non-chamfered insertions at 99% reliability. Absolute calibration is avoided by using a cross-beam sensor to locate a movable reflective sensor, as well as the peg. The reflective sensor is then moved to localize the hole, and only a relative displacement from peg to hole is needed.

References

[BG94] R. C. Brost and K. Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. *IEEE Conference on Robotics and Automation*, pages 535–542, 1994.

[Bro88] Randy C. Brost. Automatic grasp planning in the presence of uncertainty. *The International Journal of Robotics Research*, December 1988. Also appeared in Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, April, 1986.

[Bro91] Randy C. Brost. *Analysis and Planning of Planar Manipulation Tasks.* PhD thesis, CMU, January 1991.

[Can93] J. Canny. Some practical tools for algebraic geometry. In *Esprit PROMotion Workshop, Rodez, France*, pages 39–52, 1993.

[CG93] J. F. Canny and K. Goldberg. A “RISC” paradigm for industrial robotics. Technical Report RAMP 93-2, UC Berkeley, 1993.

[CG94] John Canny and Ken Goldberg. RISC robotics: Recent results and open problems. In *IEEE Conference on Robotics and Automation*, pages 1951–1958, San Diego CA., 1994.

[Dra77] S. H. Drake. *Using Compliance in Lieu of Sensory Feedback for Automatic Assembly.* PhD thesis, MIT, 1977.

[ea93] H. Tsukune et al. Modular manufacturing. *Journal of Intelligent Manufacturing*, 4, 1993.

[FC92] C. Ferrari and J. Canny. Planning optimal grasps. In *IEEE Conference on Robotics and Automation*, pages 2290–2295, 1992.

[GME91] Ken Goldberg, Matthew T. Mason, and Michael A. Erdmann. Generating stochastic plans for a programmable parts feeder. In *International Conference on Robotics and Automation.* IEEE, April 1991.

[Gol91] Ken Goldberg. A kinematically-yielding gripper. In *22nd International Symposium on Industrial Automation*, October 1991. See U. S. Patent 5,098,145, granted March 26, 1992.

[Gol93] Ken Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10(2):201–225, August 1993. Special Issue on Computational Robotics.

[GP90] Ambarish Goswami and Mike Peshkin. Mechanical computation for passive force control. In *American Control Conference*, 1990.

[Hoi93] Debra Hoitomt. Paradigm shifts in manufacturing. *IEEE Robotics and Automation Society Newsletter*, 7(3), August 1993.

[JE94a] Y-B. Jia and M. Erdmann. The complexity of sensing by point sampling. In *Workshop on Algorithic Foundations of Robotics*, 1994. San Francisco, CA.

[JE94b] Y-B. Jia and M. Erdmann. Sensing polygon poses by inscription. In *IEEE Conference on Robotics and Automation*, pages 1642–1649, 1994. San Diego, CA.

[Kan92] Takeo Kanade. Keynote talk: The kiss principle in machine vision. In *International Conference on Intelligent Robots and Systems.* IEEE/RSJ, July 1992.

[KG92] Duk Kang and Ken Goldberg. Shape recognition by random grasping. In *International Conference on Intelligent Robots and Systems.* IEEE/RSJ, July 1992. To appear in the *IEEE Transactions on Robotics and Automation.*

[LC91] M. Lin and J. Canny. A fast algorithm for incremental distance calculation. In *IEEE Conference on Robotics and Automation*, pages 1008–1014, 1991.

[LC92] M. Lin and J. Canny. Efficient collision detection for animation. In *Third Eurographics Workshop*, 1992.

[LP87] T. Lozano-Pérez. Handey: A robot system that recognizes, plans and manipulates. In *IEEE Conference on Robotics and Automation*, pages 843–849, 1987. North Carolina.

[Mas86] M. T. Mason. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics Research*, 5(3):53–71, 1986.

[Mas91] Matthew T. Mason. Kicking the sensing habit. In *Asilomar Winter Workshop*. AAAI, November 1991.

[McK91] P. J. McKerrow. *Introduction to Robotics.* Addison-Wesley, 1991.

[NW78] James L. Nevins and Daniel E. Whitney. Computer-controlled assembly. *Scientific American*, 1978.

[PC93] Eric Paulos and John Canny. Informed peg-in-hole insertion using optical sensors. In *SPIE Conference on Sensor Fusion VI*, 1993. Boston Massachusetts.

[PD80] D. Patterson and D. Ditzel. The case for the reduced instruction set computer. *Computer Architecture News*, October 1980.

[PS88] Michael A. Peshkin and Art C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 4(5), October 1988.

[RG92] Anil Rao and Ken Goldberg. Grasping curved planar parts with a parallel-jaw gripper. Technical Report 299, IRIS, August 1992.

[RG94] Anil Rao and Ken Goldberg. Shape from diameter: Recognizing polygonal parts with a parallel-jaw gripper. *International Journal of Robotics Research*, 13(1), February 1994.

[Rog93] G. G. Rogers. Modular production systems: A concurrent manufacturing philosophy. In *IEEE International Conference on Robotics and Automation*, 1993.

[WC93] A. Wallack and J. Canny. A geometric matching algorithm for beam scanning. In *SPIE Symposium on Vision Geometry II*, pages 143–159, 1993. Boston, Massachusetts.

[WC94a] A. Wallack and J. Canny. Planning for modular and hybrid fixtures. In *IEEE Conference on Robotics and Automation*, pages 520–527, 1994.

[WC94b] Aaron Wallack and John Canny. Efficient indexing techniques for model based sensing. In *International Conference on Computer Vision and Pattern Recognition*, pages 259–266. IEEE, june 1994.

[WCM93] A. Wallack, J. Canny, and D. Manocha. Object localization using crossbeam sensing. In *IEEE Conference on Robotics and Automation*, pages 692–699, 1993.

[Whi86a] D. Whitney. Real robots dont need jigs. In *IEEE Conference on Robotics and Automation*, pages 746–752, 1986.

[Whi86b] Daniel E. Whitney. 'real robots don't need jigs'. In *International Conference on Robotics and Automation*. IEEE, May 1986.

[Whi93] Daniel E. Whitney. From robots to design. *IEEE Robotics and Automation Society Newsletter*, 7(3), August 1993.