Proceedings of the 1991 IEEE
International Conference on Robotics and Automation
Sacramento, California - April 1991

Exploiting Visual Constraints in Robot Motion Planning

Seth Hutchinson
The Beckman Institute
University of Illinois
Urbana, IL 61801

Abstract

This paper addresses a number of issues concerning the integration of visual and physical constraints for the synthesis and execution of error-tolerant motion strategies. Object features and their projections onto the image plane of a supervisory camera are used to define visual constraint surfaces. These visual constraint surfaces can be directly used to enforce the following types of constrained motion: motion terminated on contact with a visual constraint surface, motion maintaining constant contact with a visual constraint surface, and motion that is simultaneously constrained by both visual and physical constraint surfaces. Existing preimage planning techniques are extended for synthesis of motion strategies that exploit these types of motion.

1 Introduction

One of the primary reasons that robots have not yet seen widespread use in manufacturing is their inability to operate successfully in the presence of uncertainty. This limitation is particularly severe given the many sources of uncertainty in a typical robotic work cell. These include: limited sensing accuracy, errors in robot control, and discrepancies between the geometric object models and the physical objects (including the parts to be manipulated and the robot itself). A great deal of robotics research has been devoted to discovering ways to cope with these uncertainties.

A number of planning systems have been developed that characterize uncertainty by systems of constrained variables [1, 9, 20]. These planners must have an a priori knowledge of the effects of manipulations and sensing on uncertainty. An alternative is to use feedback from sensors to adapt execution to the actual state of the world that the robot encounters. To date, the most successful planning systems that use this approach comprise the preimage planners [5, 4, 6, 14]. Using guarded motions and compliance [15], assembly actions are made more robust by taking advantage of physical constraints imposed on motion by the geometry of the objects in the work cell.

One of the limitations of preimage planners derives from the local nature of force sensing. Since a force sensor only reports the forces and torques exerted on the gripper, the position of the manipulator is often ambiguous. This is the problem of goal recognizability [6]. The robot might know that it has made contact with some physical constraint surface, and yet not know which surface.

Visual feedback presents a natural mechanism for resolving such ambiguities. Some work has been done in the area of integrating vision with assembly robots (for example [16, 18]), but so far, none of this work has produced a unified formalism for creating uncertainty-tolerant motion plans that use position, force and visual sensing. The main reason for this is a lack of any general formalism for applying visual sensing to motion control problems. The work reported to date has focused on highly constrained problems (e.g. tracking welding seams [2, 10]), or has imposed artificial limits on the number of degrees of freedom of the controlled motions [8, 19]. As a consequence of this, no system has yet been developed that is capable of planning motions using visual feedback to compensate for uncertainties. That is, there are no visual analogues for the preimgae force-based planners.

This paper introduces a formalism for using visual feedback in motion synthesis and control. The basic tool that is used to exploit visual constraints is the *visual constraint surface*. Visual constraint surfaces, which will be introduced in Section 2, can be used to control motion in the following ways:

1. Motion of the manipulator can be constrained to maintain contact with a visual constraint surface.
2. Motion can be terminated on contact with a visual constraint surface.
3. Visual constraint surfaces can be used in conjunction with physical constraint surfaces to force motion toward a goal configuration when this would not be possible using physical constraint surfaces alone.

2 Visual Constraint Surfaces

As described above, preimage planners rely on compliant motions that are guided by physical constraints.

This section introduces visual constraint surfaces. These are virtual surfaces that do not correspond to any physical surface in the work cell. For clarity, the following two simplifying assumptions are made. Image curves used to form visual constraint surfaces will correspond to object features (e.g. edges) and not merely to occluding contours of curved surfaces. Visual information is obtained from a single camera, positioned so that it can observe the work cell without occlusion from the robot arm. These assumption will be relaxed shortly.

Consider an edge of a particular object in the field of view. Let the geometry of the edge be defined by the parameterized curve $\mathbf{X}(\tau)$. Assume that the image plane is defined with origin at $\mathbf{r_0}$ and local coordinate system specified by the orthonormal vectors $\mathbf{I}$, $\mathbf{J}$. The normal to the image plane is specified by $\mathbf{K} = \mathbf{I} \times \mathbf{J}$. Let $\mathbf{X}'(\tau)$ represent the projection of $\mathbf{X}(\tau)$ onto the image plane. Then, for perspective projection:

$$\begin{cases} X'_I(\tau) = \dfrac{-d_f(\mathbf{X}(\tau) - \mathbf{r_0}) \cdot \mathbf{I}}{(\mathbf{X}(\tau) - \mathbf{r_0}) \cdot \mathbf{K} - d_f} \\ X'_J(\tau) = \dfrac{-\,d_f(\mathbf{X}(\tau) - \mathbf{r_0}) \cdot \mathbf{J}}{(\mathbf{X}(\tau) - \mathbf{r_0}) \cdot \mathbf{K} - d_f} \\ X'_K(\tau) = \dfrac{-\,d_f}{(\mathbf{X}(\tau) - \mathbf{r_0}) \cdot \mathbf{K} - d_f} \end{cases} \tag{1}$$

where d_f is the distance from the image plane to the center of projection of the camera (i.e. the focal length of the lens). Note that $X'_K(\tau)$ is the ratio of the focal length d_f to the perpendicular distance from the point $\mathbf{X}(\tau)$ to the center of projection. While $X'_K(\tau)$ is not required to express the image coordinates of a projected point, it can be used to construct the inverse mapping (i.e. the mapping from image coordinates to world coordinates).

The curve $\mathbf{X}(\tau)$ and its projection, $\mathbf{X}'(\tau)$, can be used to define a ruled surface [7]:

$$\mathbf{S}(\tau, \lambda) = \mathbf{X}(\tau) + \lambda(\mathbf{X}'(\tau) - \mathbf{X}(\tau)) \tag{2}$$

Recall that a ruled surface is generated by a family of lines. For $\mathbf{S}(\tau, \lambda)$, a particular generating line is obtained for each valid value of τ. Such a line will be referred to as a generating line of $\mathbf{S}(\tau, \lambda)$, or simply a generating line. For perspective projection, each generating line is a projection ray through the center of projection and the image plane point $\mathbf{X}'(\tau)$. Each ruled surface, $\mathbf{S_i}(\tau, \lambda)$, that can be constructed from an object edge and its corresponding image projection is a *visual constraint surface.* Such a surface is a virtual rather than a physical surface. Fig. 1 illustrates such a surface.

When an image curve does not correspond to an object edge, it is typically the projection of the occluding contour of the surface of a curved object (see, for example, [12]). The equation for this curve can be determined off-line, prior to execution of planned motions. Therefore this does not impose any extra computational burden on the visual processing system.

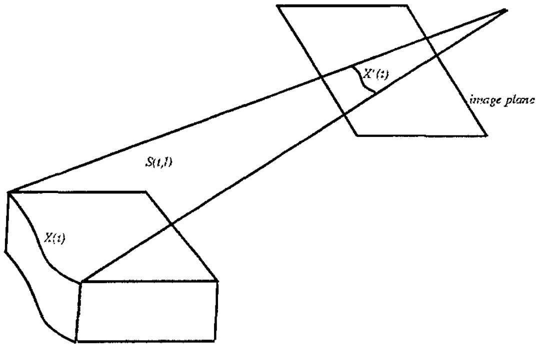

Figure 1: Ruled Surface $\mathbf{S}(\tau, \lambda)$ formed using $\mathbf{X}(\tau)$ and $\mathbf{X}'(\tau)$

3 Motion Tangent to a Visual Constraint Surface

The first way that visual constraint surfaces can be used to control robot motion is analogous to compliant motions that rely on force feedback for servo control. In this case the motion "complies" to the visual constraint surface in one of two ways.

- **Type I Motion:** The motion is constrained so that some fixed spatial relationship between the manipulator and $\mathbf{S}(\tau, \lambda)$ is maintained.
- **Type II Motion:** The motion is constrained so that the manipulator maintains some fixed spatial relationship to a specific generating line of $\mathbf{S}(\tau, \lambda)$.

While it is possible to imagine many types of spatial relationship between the manipulator and the constraint surface, in order to reduce the complexity of both the visual processing and the visual servo control, only relationships that maintain contact between some distinguished point, $\hat{P}$, and the visual constraint surface are considered. $\hat{P}$ can be a point on the manipulator, a point on the object in its grasp, or any point in space that has a fixed relationship to the manipulator (for example, the center point between the two manipulator jaws). Therefore, $\hat{P}$ maintains contact with $\mathbf{S}(\tau, \lambda)$ for Type I motion, and with a specific generating line of $\mathbf{S}(\tau, \lambda)$ for type II motion. Type I motion constrains the manipulator to move so that $\hat{P}$ projects onto $\mathbf{X}'(\tau)$, while Type II motion constrains the manipulator to move so that $\hat{P}$ projects onto $\mathbf{X}'(\tau_1)$, for some specific τ_1.

As an example, suppose that a peg held in the robot's gripper is positioned in Cartesian space so that some $\hat{P}$ lies on the surface $\mathbf{S}(\tau, \lambda)$. $\hat{P}$ could be chosen as the mid-point on the visible portion of the curved edge at the bottom of the peg. Since $\hat{P}$ lies on $\mathbf{S}(\tau, \lambda)$, the projection of $\hat{P}$ must be a point on the image plane curve $\mathbf{X}'(\tau)$.

There have been a number of efforts to integrate visual data directly into robot servo control. Khosla et al.

use visual input to generate a path in Cartesian space [10]. This path (which corresponds to a weld seam) is used to modify the trajectory of the manipulator in real time. Skaar, Brockman and Hanson report a system that updates the relationship between visual cues and manipulator joint coordinates in real time while tracking an object in the scene [19]. In [21], a Jacobian matrix that relates differential changes in image features to differential changes in Cartesian space position is used in conjunction with an adaptive controller. In [8], a system is described that generates a smooth trajectory using a feature-based approach. Again, a Jacobian matrix relating image features to positions in Cartesian space is used, but in this system visual data arrives at random times, and the trajectory is modified asynchronously.

When using visual constraint surfaces, the observed image provides only one set of constraints on the manipulator's motion. The remaining constraints on motion come from a commanded velocity (supplied by the planner). This is similar to motion at a commanded velocity that allows sliding on a physical constraint surface. Therefore, control of the manipulator's motion must be derived from the robot's position sensors in conjunction with the camera image. The visual servo aspect is viewed as providing correction to the commanded manipulator velocity.

To incorporate this correction into the robot control system, a Jacobian matrix is used to derive an incremental correction velocity. The nominal command velocity of the manipulator will be determined using standard methods (see [3] for example). The velocity correction will be applied in a direction perpendicular to **K**. Therefore, there are two degrees of freedom in the correction vector. These can be mapped to the two degrees of freedom of a point moving in the image plane. Thus, there is a generally nonsingular Jacobian matrix, J_v, which relates differential changes in the image plane projection of a point to differential changes in its Cartesian position. For a point p in Cartesian space, and its projection p', the relationship between $\dot{p}$ (the Cartesian velocity) and $\dot{p}'$ (the image plane velocity) is given by:

$$\dot{p}' = J_v \dot{p} \tag{3}$$

The choice of velocity correction is illustrated in Fig. 2. When deriving the velocity correction, $\dot{p}'$ is the image plane displacement between the projection of point $\hat{P}$ and the desired projection of $\hat{P}$. In the figure, the projection of $\hat{P}$ is represented by p', and the desired projection of $\hat{P}$ is represented by $\hat{p}'$. The image plane correction displacement is represented by the vector $\dot{p}'$ in the figure. The Cartesian velocity correction is chosen to be parallel to the image plane. This correction is represented by the vector $\dot{p}$ in the figure.

Once $\dot{p}$ has been determined (using $\dot{p} = {J_v}^{-1}\dot{p}'$), the manipulator Jacobian, J_m, (that relates joint velocity to Cartesian velocity) can be used to calculate the correction to the nominal trajectory in joint space. This is discussed in [8, 10].

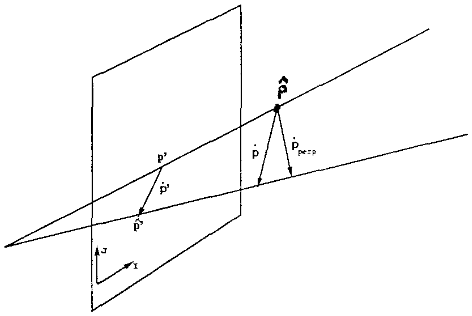

Figure 2: This figure illustrates how the displacement between the ideal and actual projections of $\hat{P}$ onto the image plane map to a correction displacement in the world frame.

4 Visual Termination Conditions for Motions

Guarded motions are motions that are executed at some chosen velocity until a terminating condition is satisfied. Termination conditions have been expressed in terms of force exerted on the manipulator, position sensing, and elapsed time [5, 4, 6, 14].

To terminate motions based on visual cues, these cues must be extracted from visual data in real time. This prohibits the use of quantitative visual data derived using numerical iterative processing techniques, and therefore rules out precise estimation of positions and orientations of the manipulator as terminating conditions. An alternative is to define qualitative visual cues that can be readily extracted from the visual data.

For most configurations of the manipulator, small motions will not change the qualitative structure of the extracted edge image of the scene (which includes the manipulator and the objects in the work cell). Configurations for which this is not true define event surfaces. These event surfaces are exactly the visual constraint surfaces defined in Section 2. In other words, the qualitative structure of edge image of the scene only changes if the manipulator makes contact with a visual constraint surface. It should be noted that this is in many ways analogous to the aspect graph used in computer vision [11].

The utility of visual event surfaces for motion termination is due to two factors: the visual data (unlike force and position sensing) provides global information, and the visual information is not required to be precise (since only qualitative information is used), thus sparing the computational burden associated with many computer vision tasks.

5 Goal Driven Servo Control Using Combined Visual and Force Feedback

Vision and force feedback can be used simultaneously to constrain motion to progress toward some goal, when this would not be possible using either visual or force feedback individually. Consider the task of inserting a peg into a hole in a block. If the system relies on force controlled compliant motion, once contact has been established between the peg and the block's surface, the motion of the manipulator can be constrained to maintain that contact. Unfortunately, force feedback alone cannot provide any information about the trajectory required to move the peg closer to the hole. This section briefly describes how visual constraint surfaces can be used to force the compliant motion to make progress toward the goal.

Consider the slightly simpler problem of moving the manipulator so that $\hat{P}$ (some distinguished point on the manipulator) comes into contact with some point $\hat{G}$ on the surface of an object. Further, assume that $\hat{P}$ is currently in contact with the object's surface. Let the projections of $\hat{P}$ and $\hat{G}$ onto the image plane be denoted by $\hat{P}'$ and $\hat{G}'$. Except for degenerate cases, the following is true: when contact is maintained between $\hat{P}$ and the object surface, compliant motions that cause the image plane distance between $\hat{P}'$ and $\hat{G}'$ to decrease will cause $\hat{P}$ to converge to $\hat{G}$ in Cartesian space.

This can be seen using the following argument. Consider two concentric circles, C_1 and C_2 in the image plane, centered at $\hat{G}'$, with C_1 having the smaller radius. For perspective projections, these circles and the vector from $\hat{G}'$ to the center of proje tion define two concentric, conical visual constraint s rfaces, $\mathbf{S}_1(\tau, \lambda)$ and $\mathbf{S}_2(\tau, \lambda)$. The intersections of $\mathbf{S}_1(\tau, \lambda)$ and $\mathbf{S}_2(\tau, \lambda)$ with the planar top surface of the object define two concentric ellipses. When $\hat{P}'$ moves from C_2 to C_1 in the image plane, it must be the case that $\hat{P}$ has moved from $\mathbf{S}_2(\tau, \lambda)$ to $\mathbf{S}_1(\tau, \lambda)$, and, due to the physical constraint of the compliant motion, to the inner ellipse on the object surface. If this process is extended for image plane circles with successively smaller radii, $\hat{P}$ will eventually move onto a degenerate $\mathbf{S}(\tau, \lambda)$, which will be a single line through the points $\hat{G}$ and $\hat{G}'$. Note that it is not necessarily the case that the distance between $\hat{P}$ and $\hat{G}$ will monotonically decrease as the distance between $\hat{P}'$ and $\hat{G}'$ decreases, only that the position of $\hat{P}$ will converge to $\hat{G}$ as the distance between $\hat{P}'$ and $\hat{G}'$ is decreased.

6 Motion Planning Using Visual Constraints

This section considers the synthesis of motion plans that use both visual and physical constraints. This is done by extending the techniques used by the preimage motion planning formalism to exploit visual constraints.

Since both motion tangent to a visual constraint surface and motion terminated at a visual constraint surface are direct analogues to force controlled motions (i.e. compliant and guarded motions), the preimage style of motion planning can be directly applied to these two types of visually constrained motions, provided visual constraint surfaces can be transformed into configuration space constraint surfaces. Since an algebraic description of each visual constraint surface is known (this is simply $\mathbf{S}(\tau, \lambda)$), this can be done using existing techniques for constructing a configuration space from algebraic descriptions of the obstacles in the work space (see [17, 13] for example).

The general approach used by preimage planners to construct a motion strategy is to use recursive back chaining to compute preimages from the goal back to the initial configuration. A preimage of a goal is the set of points from which a commanded velocity is guaranteed to terminate recognizably in the goal configuration. Recognizing that the goal configuration has been achieved is delegated to a termination predicate. As described in [6], goal recognizability is often an elusive quality, due in part to the local nature of force sensing coupled with uncertainties in the exact position of the manipulator.

For motions that terminate on contact with a visual constraint surface, goal recognizability is fairly straightforward, since, as discussed in Section 4, the visual surface is defined in terms of observable visual events. Therefore, by definition of a visual constraint surface, contact between the manipulator and a visual constraint surface will always be a recognizable contact (i.e. the system will be able to determine both that the contact has been made, and with which visual constraint surface the contact has been made).

Given that any contact with a visual constraint surface is recognizable, the remaining problem is to find the set of points from which a commanded velocity is guaranteed to come into contact with a particular surface (i.e. a preimage of the surface). This can be accomplished by using the negative velocity cone to back project constraints as discussed in [6]. Recognizing that a visual constraint surface is bounded by four curves, the preimage of the surface is bounded by the backprojections of the negative velocity cone from the set of points in these curves. (The four curves that bound a visual constraint surface are: the physical edge that generates the surface, the projection of the edge in the image plane, and two projection rays that are defined by the end points of the physical edge and their projections onto the image plane.) The technique is the same as for physical constraint surfaces, an explanation of which can be found in [6].

Since there is no friction generated by a visual constraint surface, the motion planner does not need to consider sticking during a commanded motion along a visual constraint surface. This greatly reduces the effect of uncertainty in velocity control for motion planning using visual constraint surfaces. Further, since a visual constraint surface is not a physical surface, the planner also has the option of ignoring the surface. This is done by generating a commanded velocity that will intersect the surface and instructing the termination predicate to ignore the intersection when it occurs.

Figure 3 shows a two-dimensional example of a three step plan that uses visual constraints to move the configuration from some uncertain initial configuration (represented by an error ball) to the indicated goal region in a two-dimensional configuration space. This example assumes frictionless contacts. The point c^*_{init} is the measured initial configuration of the object to be moved to the goal. Note that the actual initial configuration of the object could correspond to any point in the error ball centered at c^*_{init}. The goal surface, marked G in the figure, is the bottom surface of a hole in a block resting on the work table. Also shown in the figure are the image plane of the camera, and the error cone associated with a command velocity.

The plan shown in this figure relies on the visual constraint surface generated by point **A** and its projection onto the image plane (note that in the two-dimensional case, visual constraint surfaces are generated by points rather than by curves). The command velocity v_1 moves the object into contact with the surface using a visual *termination condition*, as described in Section 4. Following this contact, a Type II motion is used to bring the object into the hole by executing command velocity v_2. The motion of v_2 is terminated upon physical contact with the side of the hole (using the force sensor). Finally, v_3 moves the object into the goal region using physical compliance. This example illustrates how, by using both vis: al and physical constraints, fairly simple plans can be ısed to achieve goals even in the presence of significant uncertainty. Although this example problem could have been solved using only physical constraints, it would have required a number of additional steps. In particular, given the uncertainties in position and control illustrated in the figure, a force-based strategy would require at least five steps, e.g. move to the table; slide across the table to make contact with the side of the block; slide upward along the side of the block; slide across the top the block and into the hole using a velocity directed to the left and slightly down; move down the inside wall of the hole to the goal.

In addition to interleaving motions guided by physical and visual constraints, it is possible to use physical and visual constraints simultaneously, so that motion can be forced toward a goal configuration when physical constraints alone would not be sufficient for this purpose. Formally, the combination of visual and physical constraints has the effect of increasing the size of the preimage of the associated goal (that is, the set of points from which the commanded velocity is guaranteed to terminate recognizably in the goal). Consider Fig. 4. The preimage of the goal (again, the goal G is the bottom surface of the hole), assuming the shown commanded velocity, is the region labeled P (again, frictionless contacts are assumed). When the planner considers visual constraints, the preimage is expanded to include all points on the surfaces S_1 and S_2, such that the points are within the field of view of the observing camera. The figure shows the two projection rays that bound the field of view of the camera, and the included portions of surfaces S_1 and S_2. It is not immediately obvious that S_1 should b- included in this preimage, since a preimage is compute l relative to the commanded velocity, and for this example, the commanded velocity is away from the goal for points in region S_1. However, as was described in Section 5, by using visual servo control to reduce the image plane distance between the object and the goal, the velocity will be altered to force progress to the goal.

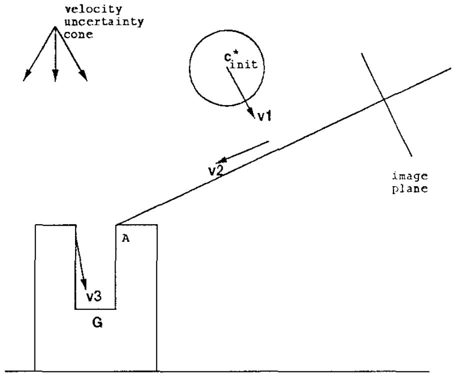

Figure 3: This figure illustrates a three step motion strategy to move an object from any initial configuration within the error ball centered at c^*_{init} to the goal region (indicated by G) using a single visual constraint surface that is generated by the object vertex A and its projection onto the camera image plane.

7 Conclusions

This paper has addressed issues related to the integration of visual and physical constraints for the purpose of creating robust motion strategies that will succeed even when there is significant uncertainty in the world. The general problem has been divided into three subproblems:

1. Defining a formal representation for the types of visual constraints that can be used in motion control.
2. Defining a control mechanism that can be used to execute motions subject to visual constraints.
3. Describing how the existing preimage approach to motion planning can be extended to exploit visual constraints.

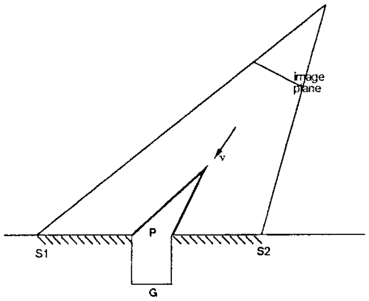

Figure 4: This figure illustrates how the preimage of the goal, G, is increased by combining visual constraints with the physical constraints.

The research presented in this paper is still in its early stages. The mathematical preliminaries have been worked out, and an implementation is now in progress. The implementation includes control software to generate trajectory corrections in real time based on visual feedback, and a configuration space preimage planner that uses visual constraints as well as physical constraints. The control software is being developed for use with a Unimation PUMA 560 robot.

References

[1] R. A. Brooks. Symbolic error analysis and robot planning. *International Journal of Robotics Research*, 1(4), Winter 1982.

[2] W. F. Clocksin, J. S. E. Bromley, P. G. Davey, A. R. Vidler, and C. G. Morgan. An implementation of model-based visual feedback for robot arc welding of thin sheet steel. *International Journal of Robotics Research*, 4(1):13–26, Spring 1985.

[3] J. J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley Pub. Co., Inc., Reading, Mass., 1989.

[4] B. R. Donald. A geometric approach to error detection and recovery for robot motion planning with uncertainty. *Artificial Intelligence*, 37(1-3):223–271, December 1988.

[5] B. R. Donald. Planning multi-step error detection and recovery strategies. *International Journal of Robotics Research*, 9(1):3–60, Feb 1990.

[6] M. Erdmann. Using backprojections for fine motion planning with uncertainty. *International Journal of Robotics Research*, 5(1):19–45, Spring 1986.

[7] I.D. Faux and M. J. Pratt. *Computational Geometry for Design and Manufacture.* Ellis Horwood Ltd., Chichester, 1985.

[8] J. T. Feddema and O. R. Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5):691–700, October 1989.

[9] S. A. Hutchinson and A. C. Kak. SPAR: A planner that satisfies operational and geometric goals in uncertain environments. *AI Magazine*, 2(1):30–61, Spring 1990.

[10] P. K. Khosla, C. P. Neuman, and F. B. Prinz. An algorithm for seam tracking applications. *International Journal of Robotics Research*, 4(1):27–41, Spring 1985.

[11] J.J. Koenderink and A.J. Van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[12] D.J. Kriegman and J. Ponce. On recognizing and positioning curved 3D objects from image contours. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(12):1127–1137, 1990.

[13] J. C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[14] T. Lozano-Perez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1):3–24, Spring 1984.

[15] M. T. Mason. Compliance and force control for computer controlled manipulators. In B. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, editors, *Robot Motion: Planning and Control*, pages 373–404. MIT Press, Cambridge, Mass., 1982.

[16] P. Saraga and B. M. Jones. Simple assembly under visual control. In Alan Pugh, editor, *Robot Vision*, pages 209–223. IFS Pub. Ltd., U. K., 1983.

[17] J. Schwartz and C. K. Yap. *Advances in Robotics.* Lawrence Erlbaum Associates, Hillsdale N. J., 1986.

[18] Y. Shirai and H. Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5:99–108, 1973.

[19] S. B. Skaar, W. H. Brockman, and R. Hanson. Camera-space manipulation. *International Journal of Robotics Research*, 6(4):20–32, Winter 1987.

[20] R. H. Taylor. The synthesis of manipulator control programs from task-level specifications. AIM-282, Standford Artificial Intelligence Laboratory, 1976.

[21] L. E. Weiss, A. C. Sanderson, and C. P Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, RA-3(5):404–417, october 1987.