Task-Oriented Generation of Visual Sensing Strategies in Assembly Tasks

Jun Miura Katsushi Ikeuchi

February 1995
CMU-CS-95-116

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was sponsored by the Advanced Research Projects Agency under the Department of the Army, Army Research Office under grant number DAAH04-94-G-0006.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARPA or the U.S. government.

Keywords: Task-oriented vision, Sensor planning, Active vision, Visual feedback, CAD-based vision

Abstract

It is generally very difficult, if not impossible, for a robot to perform fine manipulation tasks without the benefit of some form of sensory feedback during actual task execution. As a result, robot sensing strategy planning is an important component in assembly task planning.

This report describes a method of systematically generating visual sensing strategies based on knowledge of the task to be performed. Since visual sensing is usually performed with limited resources, visual sensing strategies should be planned so that only necessary information is obtained efficiently. The generation of the appropriate visual sensing strategy entails knowing *what* information to extract, *where* to get it, and *how* to get it. This is facilitated by the knowledge of the task, which describes what objects are involved in the operation, and how they are assembled.

In the proposed method, using the task analysis based on face contact relations between objects, necessary information for the current operation is first extracted. Then, visual features to be observed are determined using the knowledge of the sensor, which describes the relationship between a visual feature and information to be obtained. Finally, feasible visual sensing strategies are evaluated based on the predicted success probability, and the best strategy is selected.

Our method has been implemented using a laser range finder as the sensor. Experimental results show the feasibility of the method, and point out the importance of task-oriented evaluation of visual sensing strategies.

Contents

1. Introduction **3**

2. Determining What Information is Necessary **5**

2.1. Task Analysis Based on Face Contact State 5

2.2. Representation of Face Contact States 6

2.3. Determining What Visual Information is Necessary 10

2.4. Extension of Analysis of Face Contact State 11

3. Selection of Features to be Observed **12**

3.1. Sensing Primitive 12

3.2. Feature Selection Process 16

3.3. Generation of Candidate Sensor Position 17

4. Task-Oriented Evaluation of Visual Sensing Strategies **19**

4.1. Uncertainty-Based Evaluation 19

4.2. Evaluation Based on Predicted Success Probability 19

4.3. Derivation of Inequalities Representing Success Region 20

5. Implementation of the Method using Line Laser Range Finder **23**

5.1. Laser Range Finder and General Sensing Strategy 23

5.2. Assembly Operation with Visual Feedback 24

6. Experimental Results **24**

6.1. Uncertainty Model of Laser Range Finder 25

6.2. A Peg-in-Hole Operation 26

6.2.1. Face Contact Analysis and Actual Operation 26

6.2.2. Comparison of Predicted Success Probability with Actual Success Ratio 26

6.3. Putting Screwdriver on Bolt 28

6.4. Gear Mating 29

7. Task-Oriented Sensing Strategy Generation System **32**

7.1. Task Model and Sensor Model 32

7.2. Sensing Task Model 34

7.3. Process of Sensing Task Model Generation 35

7.4. Automatic Sensing Strategy Generation using Sensing Task Model 35

8. Concluding Remarks **36**

8.1. Summary 36

8.2. Extension to General Sensing Strategy Generation 36

1. Introduction

In vision-guided robotic operations, vision is used for extracting necessary information for proper task execution. Since visual sensing is usually performed with limited resources, visual sensing strategies should be planned so that only necessary information is obtained efficiently. To determine an efficient visual sensing strategy, knowledge of the task is necessary. Without knowledge of the task, it is often difficult to select the appropriate visual features to be observed. In addition, resources may be wasted in tracking uninformative features.

From this standpoint, research on *task-oriented vision*, *active vision*, or *purposive vision* has been actively investigated [1] [2] [4] [5] [11]. By using knowledge of the task, the vision system can be designed to be fast and robust. However, the designing process tends to be task-specific and requires a significant amount of effort. Thus, it is desirable to develop a systematic method which can generate *task-oriented* visual sensing strategies automatically, namely a method that optimizes each visual sensing strategy according to a given task.

The generation of task-oriented sensing strategy is decomposed into the following three subproblems to be solved successively:

- determine what visual information is necessary for the current task;
- determine which visual features carry such necessary information; and
- determine how to get necessary information with the sensors used.

The first two subproblems are concerned with focusing the attention to informative visual features; the last problem is concerned with evaluation of sensing strategies.

The ability of focusing attention is important to realize efficient visual sensing strategies. There have been several approaches to this problem.

Hutchinson and Kak [10] dealt with the problem of resolving the ambiguity of sensor information. They used Dempster-Shafer theory to represent uncertainties of hypotheses in object identification. An entropy of a set of hypotheses was used as a utility function; a sensor placement was selected which minimizes the entropy.

Rimey [17] presented a framework of task-oriented vision which can solve high-level vision problems such as determining which object to search for next to answer a query. The knowledge of the task is represented by a Bayesian network, and the sensing action is selected which has the highest expected utility. The utility function is defined as the combination of the predicted information value and the sensing cost.

Birnbaum et al. [6] presented a vision system which can explain a scene of blocks world in terms of stability of block structures. Using the rules derived from causal knowledge of naive physics, the focus of attention is moved to look for evidence that explains the situation.

These works are concerned with exploratory visual sensing tasks under uncertainty of the knowledge of the scene. Visual feature set, from which the observed features are selected, is given in advance; it is not automatically derived from the task description.

Kuniyoshi and Inoue [15] proposed a framework of qualitatively recognizing ongoing human action. Using a hierarchical action model, which is given in advance, possible upcoming events are predicted, and visual features to be paid attention to are selected based on that prediction.

Horswill [8] proposed a concept of *specialization* for constructing task-specific robot vision systems. By analyzing the property of the task including the environment in which the robot operates, a simple but robust vision algorithm is organized from a set of given small vision processes.

Ikeuchi and Suehiro [13] have developed a system called the Assembly Plan from Observation (APO) system, which observes a human performing an assembly task, recognizes object relations and relation transitions, maps relation transitions to assembly tasks to cause such relations, and then generate a program which instructs a robot to reproduce the series of movements originally performed by the human. Using the abstract models of assembly operations, necessary information for task recognition is efficiently extracted from the range data.

These works are concerned with the usage of the task, including the constraints on the environment, for concentrating the visual processing to only necessary portion of image. This allows visual recognition to be fast and robust. These approaches are, however, highly task-specific and are based on the careful *a priori* examination of the task.

For sensor planning in inspection tasks, several methods have been proposed which generate a set of features to be observed. Features are indicated directly in the inspection specification [20] or are selected from the specification of entities to be measured through given knowledge of mapping from measurable entities of an object to features to be observed [25]. In sensor planning for inspection, derivation of feature set to be observe is relatively easy because the purpose of the task itself is visual recognition.

The third subproblem (i.e., how to get necessary information) is decomposed further into two more specific problems of determining a set of feasible sensing strategies and subsequently selecting the most appropriate one among them. The goal of the former is to determine sensing condition which satisfies several requirements on imaging such as resolution, field of view, focus and visibility [7] [18] [23] [22]. The ability of solving such a problem would be necessary for any sensor planners as a subroutine of automatically determining feasible sensing conditions.

As mentioned above, the second more specific problem is to determine the best sensing strategy which maximizes some "goodness" function. The minimum uncertainty criterion has often been used [14] [24] [26]; some measure of uncertainty, such as the determinant of the covariance matrix of the parameter vector to be estimated, is used for selecting the best strategy. In certain types of tasks, however, this criterion may not be appropriate; some part of information may need to be more accurate than the rest for a specific task, for example. A weighted sum of uncertainty parameters is one way to deal with such a case. It is, however, difficult to determine appropriate weights for a given task. Thus, some appropriate function should be automatically designed for each task which can measure how each sensing strategy

contributes to the proper execution of the current task.

This report proposes a novel method of systematically generating visual sensing strategies based on knowledge of the task. We deal with visual sensing strategy generation in assembly tasks, in which the environment is *known*, that is, the shape, the size, and the approximate location of every object is known to the system. In this situation, the role of visual sensors is to determine the position of the currently assembled object with sufficient accuracy so that the object can be, with a high degree of certainty, mated with other objects.

In assembly operations, degrees of freedom of assembled objects are gradually constrained. Thus, specific degrees of freedom of the currently manipulated object need to be observed in each assembly operation. In this context, the above-mentioned three subproblems are solved as follows. The description of the current operation indicates the degrees of freedom that should be measured by vision (*what information to extract*). The description also provides candidates of features to be observed. Among those features, by using the knowledge of the relationships between observed features and degrees of freedom to be measured, visual features to be observed are selected (*which features to observe*). Then, candidates of feasible sensing strategies which allow the selected features to be observed are generated. Subsequently, the best sensing strategy is selected based on its contribution to proper execution of the current operation (*how to observe the features*).

The rest of this report is organized as follows. Section 2 describes a theory to determine what visual information is necessary in each assembly operation. The theory is based on the face contact analysis of the operations. Section 3 describes a method to select visual features to be observed using the task description. Section 4 explains a task-oriented evaluation of visual sensing strategies. Section 5 describes an implementation of the proposed methods using a line laser range finder. Section 6 describes the experimental results, which show the feasibility of the mothods and point out the importance of the task-oriented evaluation of visual sensing strategies. Section 7 describes a configuration of task-oriented sensing strategy generation system. Section 8 summarizes the report, and discusses an extension of our approach to more general visual sensing strategy generation.

2. Determining What Information is Necessary

Visual information can be effectively used in certain types of assembly operations, while other types can be performed without visual information if the robot is capable of compliant motion. In this section, we first describe the task analysis based on face contact relations between objects. Then, using the result of the analysis, we explain how to determine what visual information is necessary for each assembly operation.

2.1. Task Analysis Based on Face Contact State

We analyze a state of the environment in terms of face contacts between object surfaces [13]. The analysis first deals with the case where polyhedral objects perform only translational

motions, and is extended later (see Section 2.4). We assume that each assembly operation involves one manipulated object, manipulated by a robot for the current operation, and several stationary environmental objects which have face contacts with the manipulated object. We also assume that the goal of each assembly operation is to establish the required face contact state.

2.2. Representation of Face Contact States

Let us suppose a surface patch of the manipulated object have a face contact to a surface patch of an environmental object. This surface contact pair constrains the manipulated object's possible translation motion by:

$$\boldsymbol{N} \cdot \Delta \boldsymbol{T} \geq 0,$$

where $\Delta \boldsymbol{T}$ denotes possible translational motion vectors of the manipulated object and $\boldsymbol{N}$ denotes the normal direction of an environmental surface patch.

We use points on the Gaussian sphere to specify both a constraint vector and all possible translation vectors. Each vector is translated so that its start point is located at the center of the Gaussian sphere and its end point exists at some point on the surface of the Gaussian sphere. We use this point to denote the vector.

The constraint from a patch pair defines several regions in the Gaussian sphere (see Fig. 1). We refer to the plane perpendicular to the normal, $\boldsymbol{N}$, as the *constraint plane*; this plane divides the Gaussian sphere into two hemispheres. Assuming that the normal points to the north pole of the Gaussian sphere without loss of generality, the northern hemisphere corresponds to possible motion directions; the southern hemisphere corresponds to prohibited motion directions.

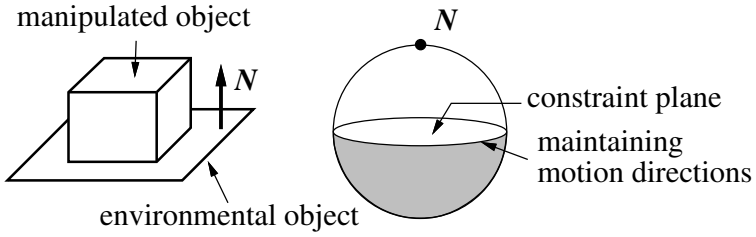


Fig. 1: Constraint inequality depicted on the Gaussian sphere.

When several surface patches of different orientations make contact, possible motion directions are constrained through simultaneous linear inequalities. These constraints are represented as a combined region in the Gaussian sphere.

In Fig. 1, motions of the directions corresponding to the boundary of the southern hemisphere (the equator) maintain the current face contact state. The degrees of freedom

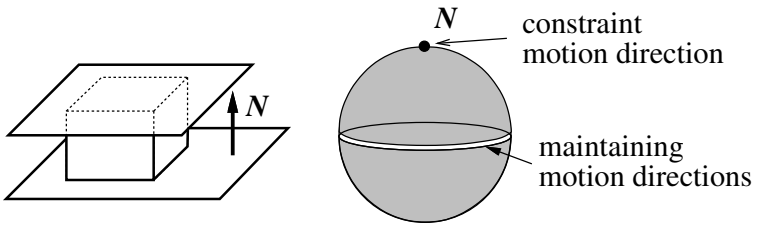


Fig. 2: A bidirectional constraint.

of the maintaining the contact state (*maintaining DOF*) is two. Motions of the directions corresponding to the inside of the detaching hemisphere break the contact state, and is referred to as the *detaching* motion. A pure detaching motion is the detaching motion which does not contain any maintaining motion component. The pure detaching motion in Fig. 1 is along the constraint normal $\boldsymbol{N}$; its degrees of freedom (*detaching DOF*) is one.

Fig. 2 shows the case where two normal vectors of environmental objects have the opposite directions. The possible motion directions of the manipulated object can be represented as *the entire great circle* perpendicular to the axis connecting the two poles. There are no detaching motions; the detaching DOF is zero. One direction along the surface normals is completely constrained; the degrees of freedom of the constraint directions (*constraining DOF*) is one.

We can specify a face contact state by using a triplet of maintaining, detaching, and constraining DOFs. Using this triplet, for example, the states of Figs. 1 and 2 are represented as $(2,1,0)$ and $(2,0,1)$, respectively.

In general, possible contact states are classified into ten contact states as shown in Fig. 3 [13]. The triplet of DOFs for each state is indicated in the figure. Note that the sum of the maintaining DOF (D_m), the detaching DOF (D_d) and the constraining DOF (D_c) is three: $D_m + D_d + D_c = 3$. (Recall that only three-dimensional translational motion is considered here.)

Each assembly operation is considered as a *transition* from one face contact state to another. We extract possible transitions between the ten contact states based on the following criteria [13]:

1. If a *direct detach* motion (a motion that immediately breaks a face-contact) exists, choose it.
2. If a *lateral* motion (a motion that maintain the same contact state) that would break face-contacts by crossing a certain boundary exists, choose it.
3. If several candidate motions satisfy criterion 1 or criterion 2, choose the motion that least reduces the number of face contacts.

The application of these criteria to the analysis of face contact relations results in the contact state transition graph as shown in Fig. 4.

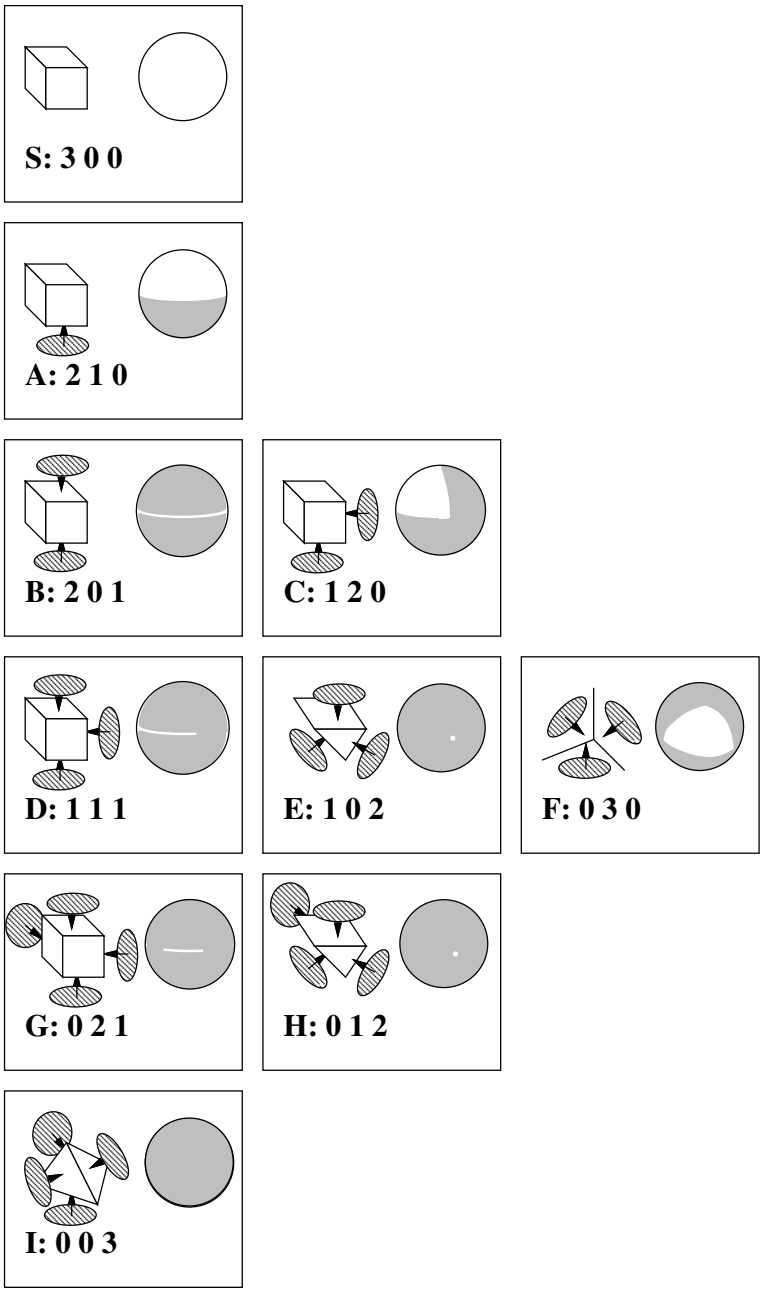


Fig. 3: Ten contact states; the white areas in the Gaussian sphere denote possible motion vectors. Each state has a label. The three digits denote maintaining DOF, detaching DOF, and constraining DOF, respectively [13].

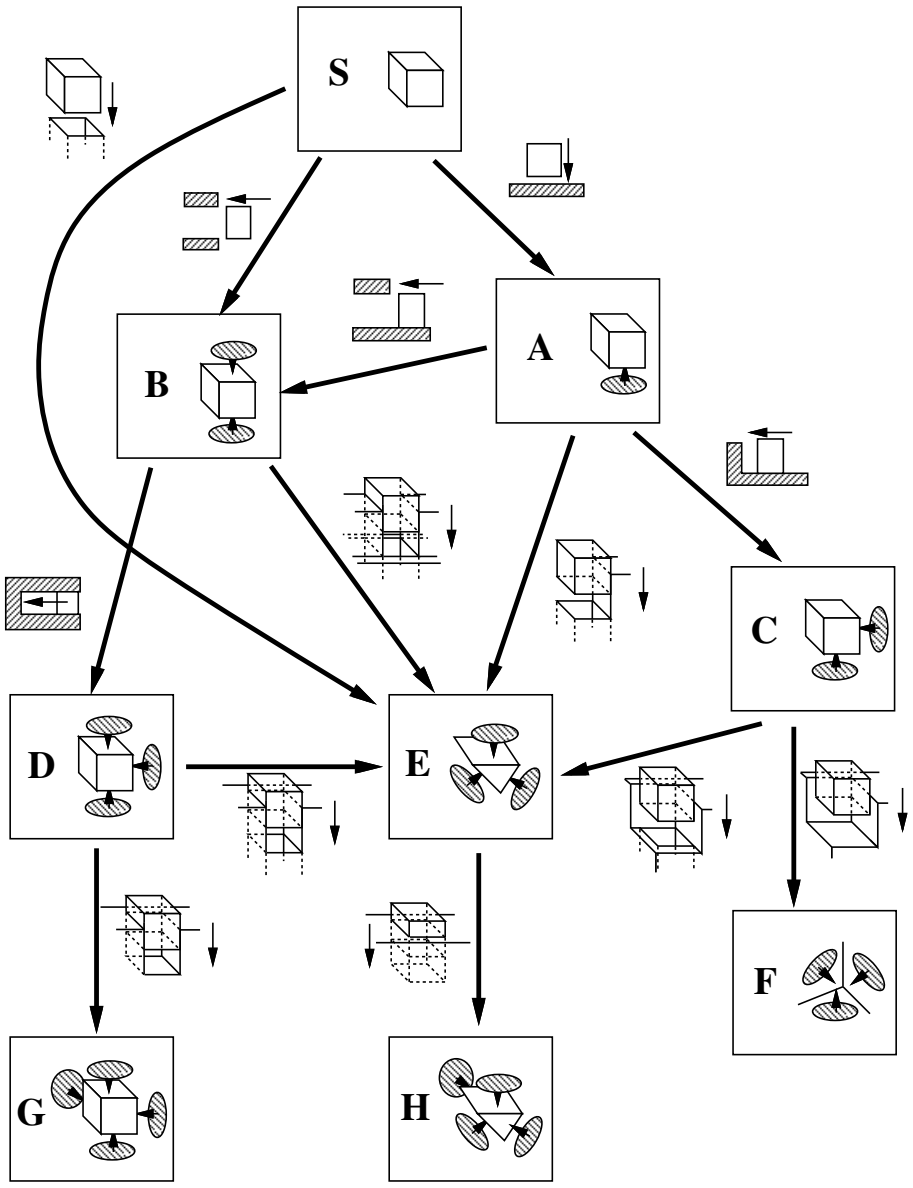


Fig. 4: Contact state transitions represented as a directional graph [13].

2.3. Determining What Visual Information is Necessary

An assembly operation (i.e., a transition of face contact state) always increases constraints on some degrees of freedom of the manipulated object. This increase of constraint is classified into three cases: from maintaining DOF to detaching DOF, from detaching DOF to constraining DOF, and from maintaining DOF to constraining DOF. Fig. 5 shows typical situations corresponding to the three cases.

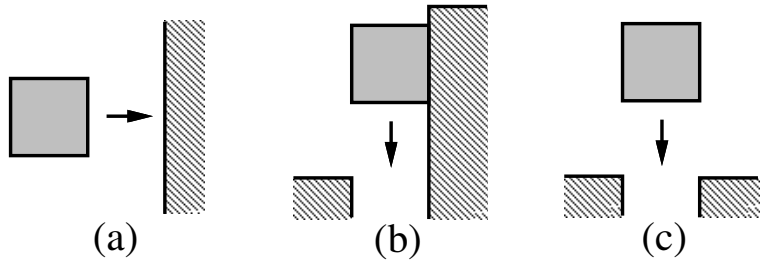


Fig. 5: Three typical cases of increase of constraint on a degree of freedom. Type and transition of the triplet is as follows:
(a): maintaining $\rightarrow$ detaching $((3,0,0) \rightarrow (2,1,0))$.
(b): detaching $\rightarrow$ constraining $((2,1,0) \rightarrow (2,0,1))$.
(c): maintaining $\rightarrow$ constraining $((3,0,0) \rightarrow (2,0,1))$.

Let us examine how the type of the degree of freedom for horizontal motion changes in these cases, and how that change is realized. In case (a), the degree of freedom changes from maintaining DOF to detaching DOF. Since the approaching direction of the block is parallel to the direction of the pure detaching motion at the final state (i.e., the normal vector of the wall), this operation is realized by moving the block until the face contact occurs. Thus, this operation can be performed by compliant motion without visual information. In case (b), the degree of freedom changes from detaching DOF to constraining DOF. Although the horizontal degree of freedom is constrained at the final state, visual information is also unnecessary because the desired horizontal position can be kept by maintaining the contact between the block and the right wall. In case (c), the degree of freedom changes from maintaining DOF to constraining DOF. The horizontal position of the block needs to be adjusted with visual information before mating so that both the left and the right face contact are achieved simultaneously.

If a degree of freedom is maintaining DOF, there is no physical contact regarding that degree of freedom; the positioning of that degree of freedom cannot receive the benefit from force information. Thus, if that degree of freedom is to be constraining DOF by the current operation, that degree of freedom should be observed by vision.[1]

[1]A sophisticated force control-based manipulation strategy may be employed to perform this kind of assembly operation without visual feedback [21]. Even in such a case, reducing errors by visual information would be useful. For example, with the help of visual information, the number of motion steps could be reduced [9].

To summarize, if a degree of freedom becomes constraining DOF from maintaining DOF, that degree of freedom should be observed. If such a transition of degrees of freedom does not exist in the current operation, that operation does not require visual information.[2] By applying this theory to thirteen possible state transitions enumerated in Fig. 4, four transitions (S→B, S→E, A→E and B→E) were found to require visual information (see Fig. 6).

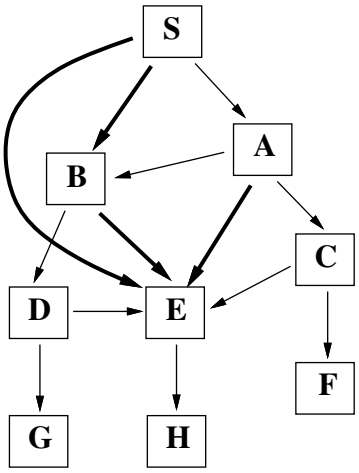


Fig. 6: Classification of state transitions. Bold lines indicate the transitions that require visual information. Thin lines indicate the transitions that do not require visual information.

2.4. Extension of Analysis of Face Contact State

In this subsection, we extend the previous contact state analysis to the case where an object can be composed of planar or cylindrical surfaces, and where, in addition to three translational degrees of freedom, one rotational degree of freedom is allowed. Since most of assembly operations in practical situations are realized by these four degrees of freedom, this extended analysis is considered to be reasonable for practical use.

Fig. 7 shows a typical object, which has both planar and cylindrical surfaces, used in the extended face contact analysis. We limit the contact states of a cylindrical surface to the three cases where (from left to right) no contact occurs, contact occurs on half of its surface, and contact occurs on all of its surface; these cases correspond to maintaining DOF, detaching DOF, and constraining DOF for the translational motion perpendicular to the principal axis of the cylindrical surface. In this extended analysis, we represent a face contact state by a sextuplet of DOF, which is composed of two triplets of translational DOFs and of rotational DOFs.

[2]Note that this theory is based on the above-mentioned assumptions that the manipulated object is assembled through several face contact states, and that the goal of each assembly operation is to establish the required face contact state.

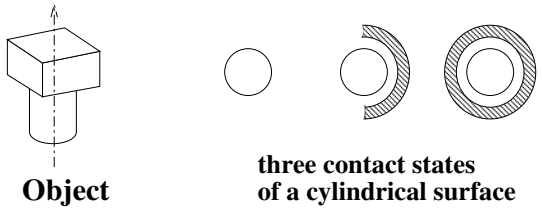


Fig. 7: The object considered in the contact state analysis.

Fig. 8 summarizes the analysis result. In the figure, the possible face contact states are arranged according to their sextuplets of DOFs. The number of states is 54, which is rather large as compared with the number of states in case that only polyhedral objects and translational motions are considered, namely 10.

We then enumerate possible transitions between contact states. By applying the above-mentioned criteria on feasible transitions to the states, we extracted 85 feasible transitions shown in Fig. 9.

The above theory of determining necessary visual information for assembly operations, which is derived for translational motions, is also applicable to rotational motions because an infinitesimal rotational motion just before the state transition can be considered as a translational motion. By examining the change of sextuplet in each state transition, 19 out of 85 transitions are found to require visual information. These transitions are indicated by bold (solid or dotted) lines in Fig. 9. Further examination in terms of the degrees of freedom to be observed by vision classifies these 19 transitions into six groups. Typical situations for the groups are depicted in Fig. 10.

3. Selection of Features to be Observed

In each vision-guided assembly operation, a relevant set of features needs to be selected so that necessary degrees of freedom of the assembled objects are observed. This section describe how to select this set of visual features using knowledge of the task.

3.1. Sensing Primitive

To solve the feature selection problem, we introduce a concept of *sensing primitive*. Sensing primitive is an abstract sensing procedure, which describes the relationship between an observable feature and degrees of freedom to be measured. For each primitive visual feature, such as a straight edge of a polyhedron, one sensing primitive is prepared. The repertoire of sensing primitives is generated in advance by enumerating possible geometric features that could appear in the assembly task under consideration, and that are observable with the sensor used. Examples of sensor primitives are shown below.

ROTATIONAL DOFs

TRANSRATIONAL DOFs

(m, d, c)	*(0, 0, 1)*	*(0, 1, 0)*	*(1, 0, 0)*
(3, 0, 0)			S
(2, 1, 0)	A1	A2	A3
(2, 0, 1)	B1	B2	B3
(1, 2, 0)	C1–1 C1–2	C2–1 C2–2	
(1, 1, 1)	D1–1 D1–2 D1–3 D1–4	D2–1 D2–2 D2–3 D2–4	D3
(1, 0, 2)	E1–1 E1–2 E1–3 E1–4	E2–1 E2–2 E2–3 E2–4	E3
(0, 3, 0)	F1	F2	
(0, 2, 1)	G1–1 G1–2 G1–3	G2–1 G2–2 G2–3	G3
(0, 1, 2)	H1–1 H1–2 H1–3 H1–4 H1–5	H2–1 H2–2 H2–3 H2–4 H2–5	H3–1 H3–2
(0, 0, 3)	I1–1 I1–2	I2	I3

Fig. 8: Summary of extended contact state analysis.

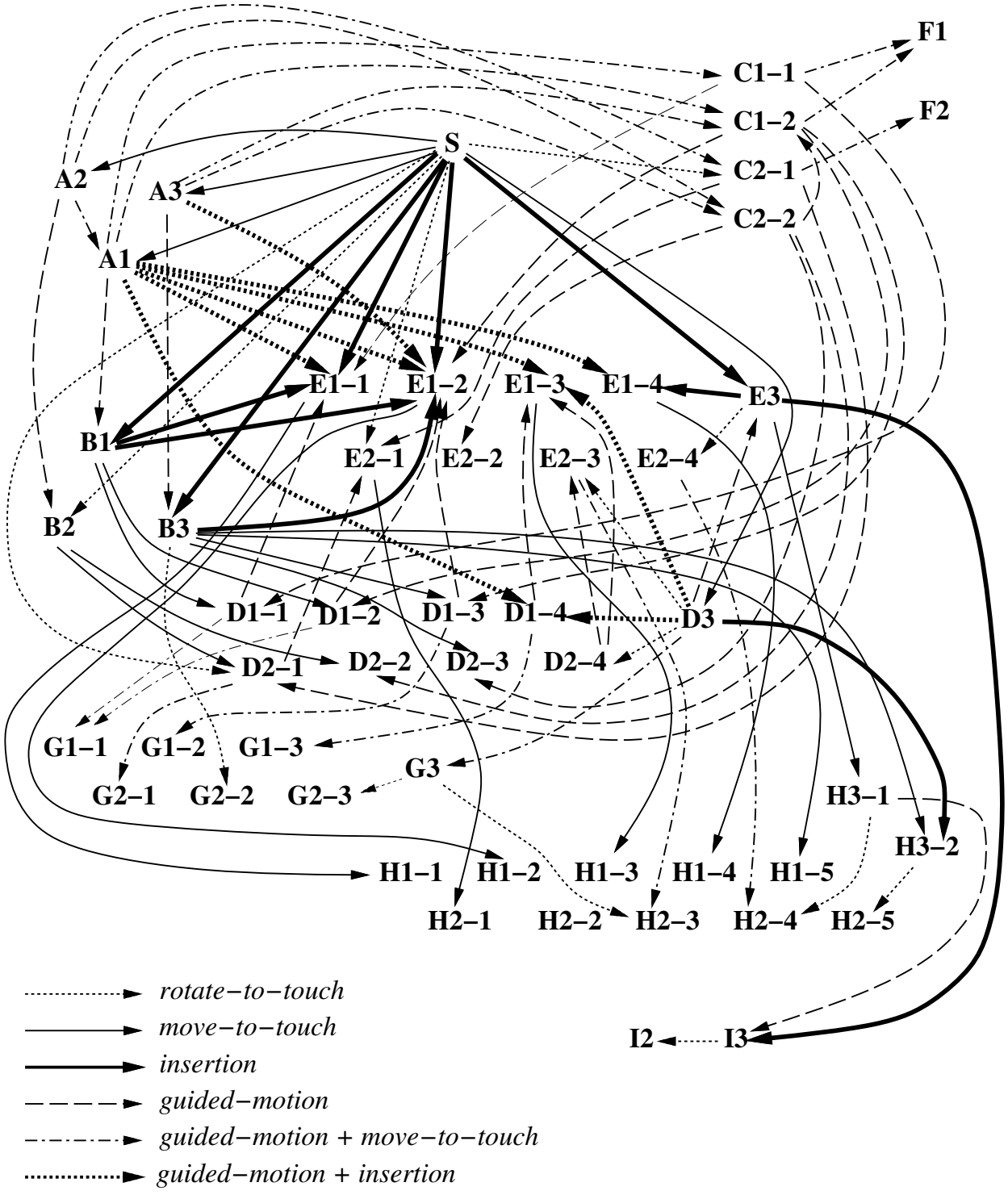


Fig. 9: Transition graph for extended analysis.

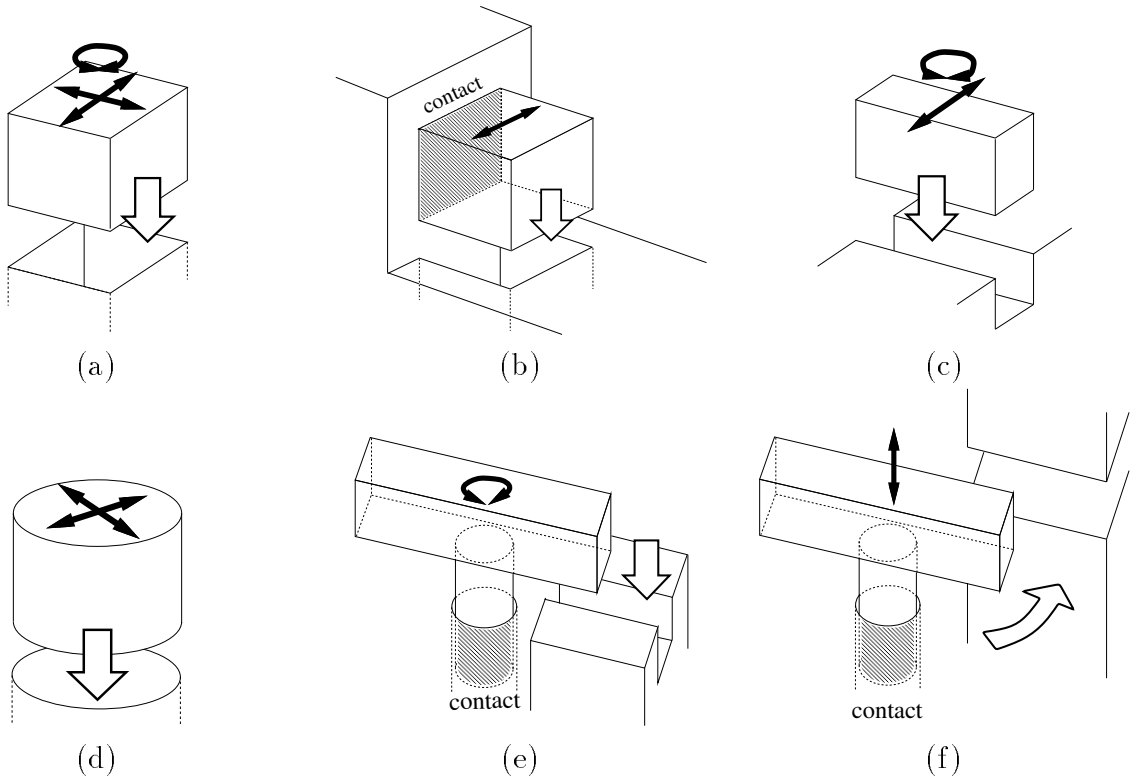


Fig. 10: Transition groups which need visual information. Thick arrows indicate the direction of movement. Thin arrows indicate degrees of freedom to be adjusted by use of visual information. The transition of sextuplet for each case is as follows:

(a): $(3,0,0;1,0,0) \rightarrow (1,0,2;0,0,1)$.

(b): $(2,1,0;0,0,1) \rightarrow (1,0,2;0,0,1)/(2,0,1;0,0,1) \rightarrow (1,0,2;0,0,1)$.

(c): $(3,0,0;1,0,0) \rightarrow (2,0,1;0,0,1)$.

(d): $(3,0,0;1,0,0) \rightarrow (1,0,2;1,0,0)$.

(e): $(1,0,2;1,0,0) \rightarrow (1,0,2;0,0,1)/(1,1,1;1,0,0) \rightarrow (1,0,2;0,0,1)$.

(f): $(1,0,2;1,0,0) \rightarrow (0,0,3;1,0,0)/(1,1,1;1,0,0) \rightarrow (0,1,2;1,0,0)$.

3.2. Feature Selection Process

Fig. 11 illustrates the process of the feature selection. From the task description, the degrees of freedom to be constrained by the current assembly operation is obtained by the method described in the previous section. On the other hand, a set of observable features comes from the face contact information in the task description. By consulting prepared sensing primitives, a feasible set of features is selected. Necessary geometric information in this selection is retrieved from a CAD-based world model.[3]

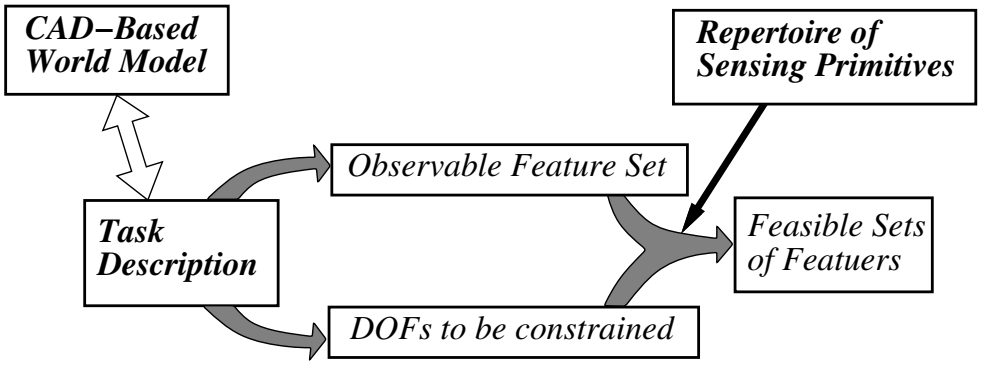


Fig. 11: Selection of features to be observed using task description and sensing primitives.

The selection problem is formalized as follows. Let us consider a general case where an assembled object has six degrees of freedom. Since constraints for translational ones and that for rotational ones can be considered separately, we take two Gaussian spheres, G^T and G^R, and use points on each sphere to represent *movable directions* (on G^T) or *possible rotation axes* (on G^R) under a certain set of constraints. Let us use the following notation:

- A^T_{goal} (A^R_{goal}): A set of points on G^T (G^R) which represents desired constraints which need to be achieved after the next sensing is completed.
- A^T_{curr} (A^R_{curr}): A set of points on G^T (G^R) under the current set of constraints. This includes constraints realized by a motion control during the current assembly operation (e.g., contact-maintaining operation in Fig. 5(b)) as well as geometric constraints achieved so far.
- A^T_i (A^R_i): A set of points on G^T (G^R) which represents constraints to be obtained by measuring the 3D position of the ith feature; this information is described in each sensing primitive.

By observing n different features $\{feature_i | i = 1, \ldots, n\}$, the following two resultant point sets, A^T_{obsd} and A^R_{obsd}, are obtained:

[3]The object recognizer determines each object configuration in the real world and generates a CAD-based world model.

$$A^T_{obsd} = A^T_{curr} \bigcap \left(\bigcap_{i=1}^{n} A^T_i \right),$$
$$A^G_{obsd} = A^G_{curr} \bigcap \left(\bigcap_{i=1}^{n} A^G_i \right).$$

In order that this set of features is sufficient for providing enough constraints, the following condition must be satisfied:

$$A^T_{goal} \supseteq A^T_{obsd} \ \wedge A^R_{goal} \supseteq A^R_{obsd}. \tag{1}$$

A pair of A^T_i and A^R_i is a representation of a *sensing primitive*. For exampe, let us consider the case where we have four primitive features: a straight edge, a planar surface, a circular edge, and a cylindrical surface. The Gaussian spheres representing constraints obtained by measuring the 3D position of each feature are shown in Fig. 12. For example, by observing a straight line of an object, we can determine the position of the object except the displacement along the direction of the line; we can determine the orientation of the object except the rotation about the direction of the line.

As an example of feature selection, let us consider two cases (a) and (b) of peg-in-hole operation shown in Fig. 13. Suppose we are localizing the hole by observing the position of its edges. Since the degrees of freedom to be adjusted are limited on a plane perpendicular to the insertion direction, we use the Gaussian circle instead of the Gaussian sphere. Also, we consider it only for translation because the rotational degree of freedom will be constrained by observing at least one edge.

First, as shown in Fig. 13, A^T_{curr} for case (a) is represented by the full circle, while that for case (b) is represented by two points on the x axis. A^T_{goal} is the Gaussian circle with no points for both cases. Then, as shown in Fig. 14, A^T_i for two edges, e_1 and e_2, is represented by two points corresponding the direction of the edge. By applying equation (1) to these data, we can decide that observing only e_1 is sufficient for case (b), and that observing both e_1 and e_2 is necessary for case (a).

3.3. Generation of Candidate Sensor Position

Once a set of features to be observed is selected, a set of feasible sensor positions is enumerated from which all of the selected features are observable. There has been work done on generating a set of feasible viewpoints which satisfy various observation condition such as visibility and detectability of visual features [7] [18]. Similarly, the candidates for sensor positions are enumerated by considering visibility of features and the possibility of collision between sensors and other objects, as well as the configuration of sensors. The selection of the final sensing strategy from the candidates is described in the next section.

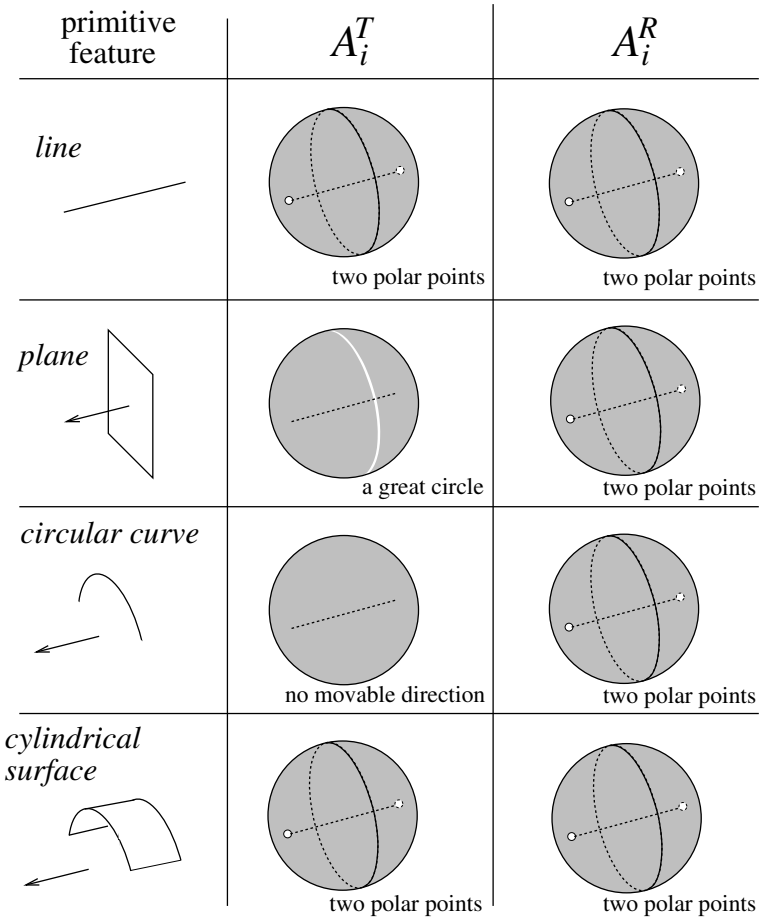


Fig. 12: Example sensing primitives represented by the Gaussian spheres.

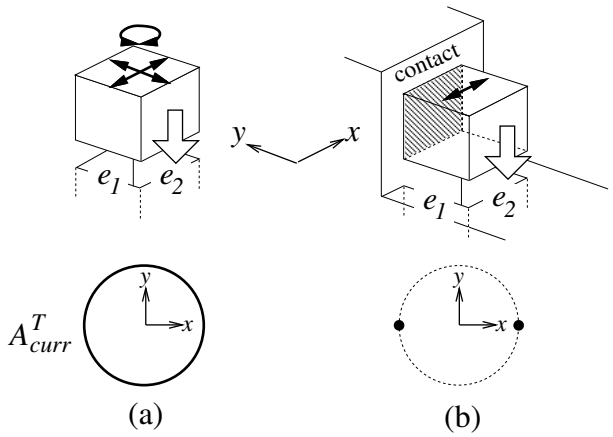


Fig. 13: Two insertion operations and movable directions.

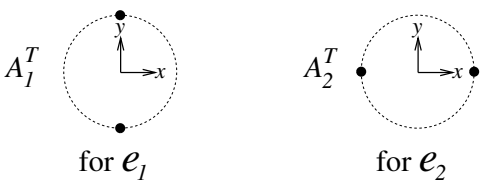


Fig. 14: Constraints obtained by observing two edges: A_1 for edge e_1 and A_2 for edge e_2.

4. Task-Oriented Evaluation of Visual Sensing Strategies

4.1. Uncertainty-Based Evaluation

In the assembly tasks treated in this report, the purpose of sensing is to acquire positional information of the assembled object with respect to the static environment, thereby feeding it back to the robot to adjust the position of the currently manipulated object. Thus, we evaluate sensory information in terms of accuracy of the estimate of the object position.

We assume that sensor data consist of position measurements, and that each measurement has the uncertainty, which can be calculated using the uncertainty model of the sensor. The problem of estimating the object position is described as follows. Let $\boldsymbol{a}$ be the parameter vector representing the object position. Suppose that the parameter vector is estimated from a set of 3D measurements $\{\boldsymbol{x}_i | i = 1, \ldots, N\}$, and that the following equation $\boldsymbol{f}_i$, which represents constraints on measurements (shape of the measured feature) and comes from the CAD-based world model, should ideally be satisfied for each measurements:

$$\boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{a}) = \mathbf{o} \tag{2}$$

The initial estimate of the parameters are also obtained from the CAD-based world model. Since this equation is non-linear in general, by applying the Extended Kalman Filter theory [3] to this estimation problem, the covariance matrix S of $\boldsymbol{a}$ is given by

$$\begin{aligned} S^{-1} &= \sum_{i=1}^{N} M_i^T W^{-1} M_i, \\ M_i &= \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{a}}, \\ W_i &= \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{x}_i} \Lambda_i \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{x}_i}^T, \end{aligned} \tag{3}$$

where Λ_i is the covariance matrix representing the uncertainty in the ith measurement $\boldsymbol{x}_i$.

Accuracy of an estimate can be evaluated based on the covariance matrix S. Note that the uncertainty of each element of $\boldsymbol{a}$ has a different effect to the success of the current operation, and that this relative effects of the elements depends on the assembly operation currently being carried out. A weighted sum of elements in the covariance matrix may be one way to handle such relative effects. It is, however, difficult to determine appropriate weights for a given task, especially when the shape of the assembled is complex.

The next subsection describes a general method to define an appropriate evaluation function for each assembly operation to be performed.

4.2. Evaluation Based on Predicted Success Probability

If the position estimation of the assembled object is poor, the current operation will most likely fail; if position estimation is accurate enough, the operation will succeed. Thus, one

of the most direct criteria that measures the "goodness" of sensing strategy is whether the current operation will succeed with a selected sensing strategy. Therefore, we use the *predicted success probability* of the operation as the criterion; the sensing strategy that is selected is the one most likely to result in successful task execution.

To calculate the success probability, we first calculate a region in the space of the error of the position parameter vector such that if the error is inside the region, the current operation succeeds. We call this region a *success region*. Then, the predicted success probability is given by calculating how much portion of the uncertainty distribution of the error, which is predicted from the sensor model, is inside the success region of the current operation (see Fig.15). This success probability is numerically calculated by quantizing the space of the error vector. After calculating the probabilities for all feasible sensing strategies, the one with the highest probability is selected as the final sensing strategy.

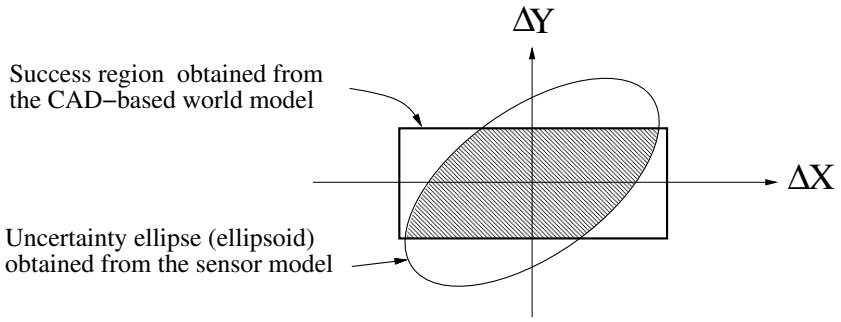


Fig. 15: Calculation of the predicted success probability. This figure shows the case where the position parameter of the object is two-dimensional, (X, Y).

4.3. Derivation of Inequalities Representing Success Region

A success region is formed in a space of unconstrained degrees of freedom in each operation. In case of operation (a) in Fig. 10, for example, the success region is formed in a three-dimensional space composed of two translational and one rotational degrees of freedom on the plane perpendicular to the insertion direction.

A success region is a representation of the clearance of the operation, and is automatically calculated using the CAD models of objects. Since we are now dealing with assembly of objects which are composed of planar or cylindrical surfaces, considering the following three cases is sufficient:

Case (1): insertion of a peg with a polygonal cross-section into a hole (cases (a) and (b) in Fig. 10). In this case, for each pair of vertices of a peg and a hole, two inequalities are generated; if the peg vertex is convex, inequalities are generated between the peg vertex and the two edges crossing at the hole vertex; if the peg vertex is concave, inequalities are generated between the hole vertex and the two edges crossing at the peg vertex. Fig.

16 shows the two possibilities; V, L_1 and L_2 indicate the vertex and the edges used for generating inequalities.

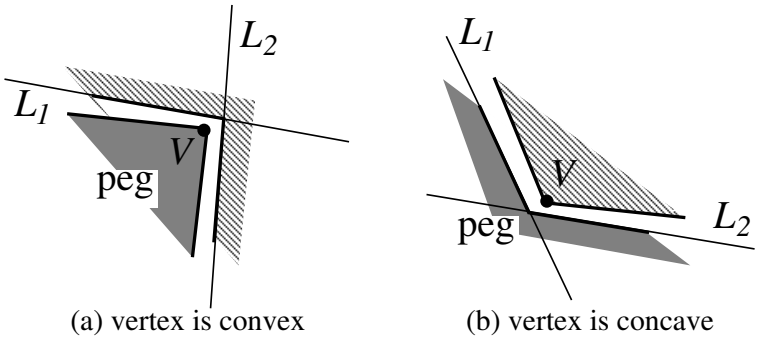


Fig. 16: Vertex (V) and edges (L_1 and L_2) used for generating inequalities in the case of insertion of a peg with a polygonal cross-section into a hole.

Case (2): insertion of a peg with a polygonal cross-section into a parallel gap (cases (c), (e) and (f) in Fig. 10). In this case, for each pair of a vertex and an edge, one inequality is generated. Fig. 17 shows two possibilities; V and L indicate the vertex and the edge for generating an inequality.

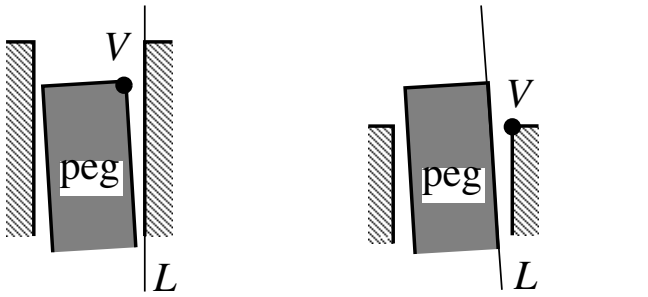


Fig. 17: Vertex (V) and edge (L) used for generating an inequality in the case of insertion of a peg with a polygonal cross-section into a parallel gap.

Case (3): insertion of a peg with a circular cross-section into a hole (case (d) in Fig. 10). In this case, the success region is represented as a circle in a two-dimensional space composed of two translational DOFs; the radius of the circle is the difference between the radius of the hole and that of the peg.

The above calculations of success region deal with a general case, that is, the case where there is no constraints on the peg before the operation. If there is some constraints before the operation (e.g., case (b) in Fig. 10), the actual success region is given as a cross-section of the general success region cut by fixed parameter values under the constraints.

Here, as an example, we derive the success region for the case where the shape of the cross-section is rectangle. Fig. 18 shows a top view of the operation. Edges of the hole are aligned to the X and the Y axes. Let W_X and W_Y be the widths of the peg in the X and Y axes, respectively. Also, let k denote the clearance ratio of the hole. These values come from the CAD model. We need to adjust the position and the orientation of the peg, (X, Y, θ).

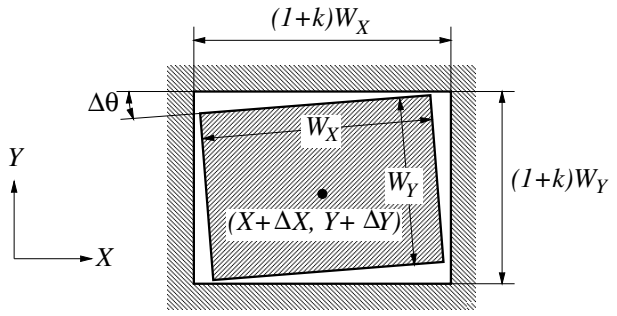
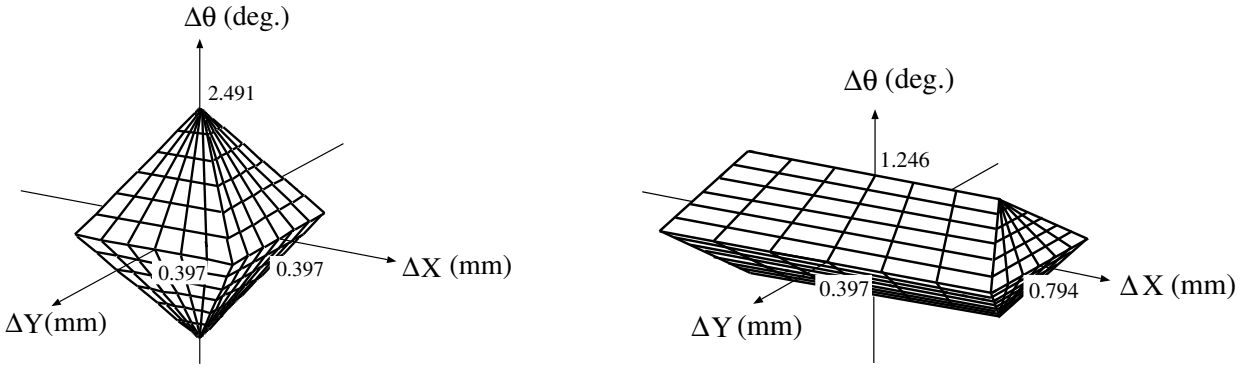


Fig. 18: Rectangular peg-in-hole operation.

Let $(\Delta X, \Delta Y, \Delta\theta)$ denote the error of (X, Y, θ). Based on the above method of success region calculation (case 1), the following set of eight inequalities is derived:

$$
\begin{aligned}
\Delta X + \frac{W_X}{2}\cos\Delta\theta - \frac{W_Y}{2}\sin\Delta\theta &\leq \frac{k+1}{2}W_X \\
\Delta Y + \frac{W_X}{2}\sin\Delta\theta + \frac{W_Y}{2}\cos\Delta\theta &\leq \frac{k+1}{2}W_Y \qquad \text{(upper right vertex)} \\
\Delta X + \frac{W_X}{2}\cos\Delta\theta + \frac{W_Y}{2}\sin\Delta\theta &\leq \frac{k+1}{2}W_X \\
\Delta Y + \frac{W_X}{2}\sin\Delta\theta - \frac{W_Y}{2}\cos\Delta\theta &\geq -\frac{k+1}{2}W_Y \qquad \text{(lower right vertex)} \\
\Delta X - \frac{W_X}{2}\cos\Delta\theta - \frac{W_Y}{2}\sin\Delta\theta &\geq -\frac{k+1}{2}W_X \\
\Delta Y - \frac{W_X}{2}\sin\Delta\theta + \frac{W_Y}{2}\cos\Delta\theta &\leq \frac{k+1}{2}W_Y \qquad \text{(upper left vertex)} \\
\Delta X - \frac{W_X}{2}\cos\Delta\theta + \frac{W_Y}{2}\sin\Delta\theta &\geq -\frac{k+1}{2}W_X \\
\Delta Y - \frac{W_X}{2}\sin\Delta\theta - \frac{W_Y}{2}\cos\Delta\theta &\geq -\frac{k+1}{2}W_Y \qquad \text{(lower left vertex)}
\end{aligned}
\tag{4}
$$

We calculated the actual success regions for two sets of geometric values. Fig. 19 shows the resultant success regions. As shown in the figure, the tolerance in ΔX in case (b) is larger than that in case (a), while the tolerance in $\Delta\theta$ is smaller. If the uncertainty distribution of the position parameter is the same to both cases, the resultant success probabilities should differ from each other. Thus, the effect of the uncertainty in the parameter vector to the task execution needs to be evaluated by considering the success region.



(a) $W_X = 19.05$ (mm), $W_Y = 19.05$ (mm), $k = 0.043$.

(b) $W_X = 38.10$ (mm), $W_Y = 19.05$ (mm), $k = 0.043$.

Fig. 19: The shape of an success region depends on the task.

5. Implementation of the Method using Line Laser Range Finder

5.1. Laser Range Finder and General Sensing Strategy

The proposed method has been implemented using a Toyota line laser range finder (LRF) [16] as the sensor. The LRF emits slit laser, detects highlighted portion of the object by a TV camera, and obtains a line of 3D measurements (see Fig. 20). The LRF is attached to an overhead platform of the RobotWorld [19]. This platform has four degrees of freedom: three degrees of freedom for translation and one for rotation about the vertical axis.

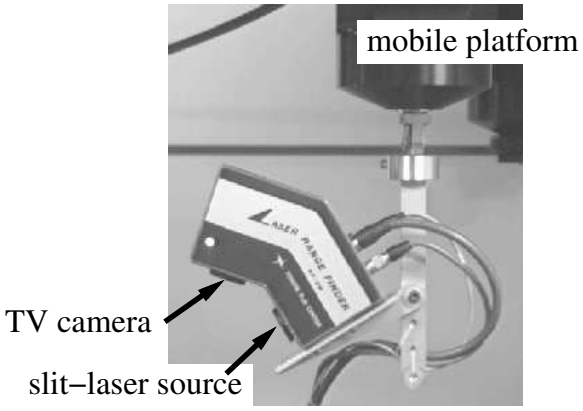


Fig. 20: A line laser range finder.

Every assembly operation that requires visual information is a kind of "peg-in-hole" operation. The location of a peg is measured by observing its side faces; the location of

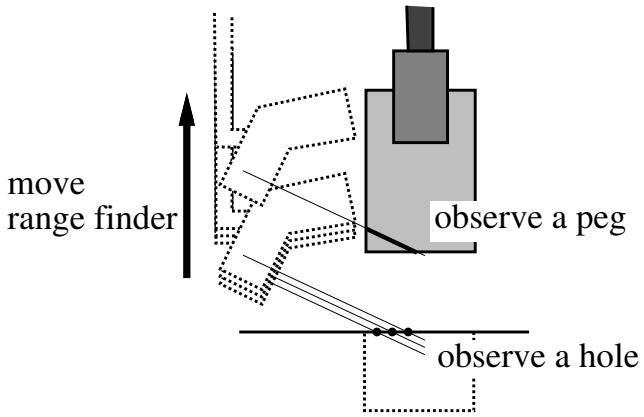


Fig. 21: A strategy for observing a peg and a hole.

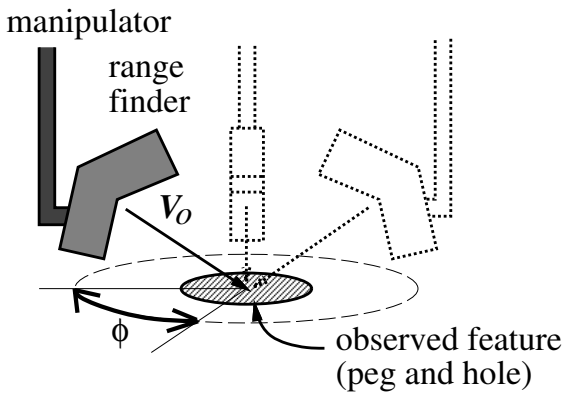


Fig. 22: Candidate positions.

the hole is measured by observing several points (currently, five points) on its edges. Thus, we prepare sensing primitives for the following four geometric features: a straight edge, a circular edge, a planar face and a cylindrical surface. We use a sensing strategy as shown in Fig. 21; data for one assembly operation are collected at several position by moving the LRF in parallel with the insertion direction because the relative displacement on the plane perpendicular to the insertion direction is important for the operation. We also control the position of the range finger so that each measured point is kept within a certain area of the slit laser; the uncertainty of the measurement with the LRF is considered to be almost constant in this area. Thus, the only parameter that specifies the position of the range finder is the angle ϕ between the direction of the laser and some axis of the plane perpendicular to the insertion direction (see Fig. 22).

5.2. Assembly Operation with Visual Feedback

The actual vision-guided assembly operation is performed in a "stop and sense" mode. First, a peg is moved by a manipulator to the position just before a hole. Then, the LRF is placed to the planned position, and measures the position of the hole and the peg. If the error in the relative position between the peg and the hole is within the success region (see Section 4.2), the peg is inserted. Otherwise, the peg position is adjusted and the peg is observed again. This final step is repeated until the relative position becomes satisfactory, and then the peg is inserted.

6. Experimental Results

This section describes the experimental results. We conducted three kinds of operations which are often involved in ordinary assembly tasks: a peg-in-hole operation, an operation of putting the tip of a screwdriver into the slot of a bolt head, and a gear-mating operation.

To validate the selection of visual sensing strategy based on predicted success probability (see Section 4.2), we compared the predicted success probability, which is predicted from the object models and sensor models, with the actual success ratio, which is statistically obtained through a number of actual trials of the same operation by the actual robot and the sensor.

6.1. Uncertainty Model of Laser Range Finder

The laser range finder we used provides quite accurate measurement: less than $0.1mm$ in depth and less than $0.3mm$ in the horizontal position. In order to stress the effect of uncertainty to the success probability, we artificially added a relatively large Gaussian noise to the measurement; we added Gaussian of standard deviation $0.12mm$ to the depth measurement and that of standard deviation $0.30mm$ to the horizontal position measurement; uncertainties of these two measurements are set to be independent of each other. Fig. 23 shows the distribution of 500 measurements of the same point. The resultant uncertainties are reasonably Gaussian with almost the desired standard deviations.

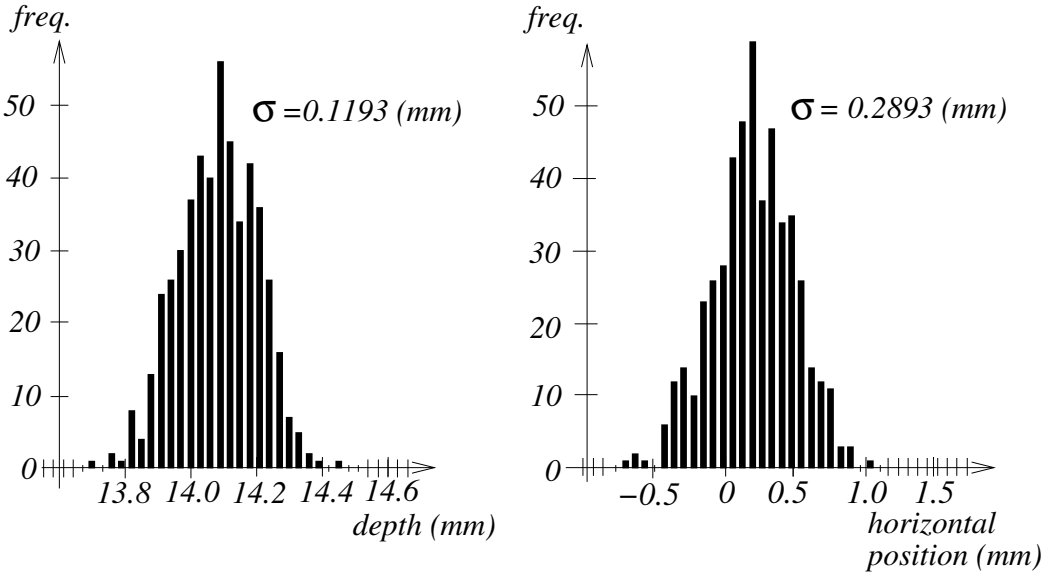


Fig. 23: Distribution of measurements and estimated standard deviation.

The reason why we used this uncertainty model is that the purpose of this report is not to construct an uncertainty model of our laser range finder but to demonstrate that our method can generate the optimal sensing strategy if the uncertainty model of the sensor is given.

6.2. A Peg-in-Hole Operation

6.2.1. Face Contact Analysis and Actual Operation

The first operation is the operation of inserting a peg with a rectangular cross-section into a hole. Fig. 24 shows the face contact analysis of the operation. This operation belongs to group (a) in Fig. 10, and results in two translational and one rotational degrees of freedom being constrained. The face contacts to be achieved are (f_1-f_1'), (f_2-f_2'), (f_3-f_3') and (f_4-f_4'). The candidates for observed features are f_1, f_2, f_3 and f_4 for the peg, and e_1', e_2', e_3' and e_4' for the hole. To obtain sufficient information for localization, two neighboring faces and edges were observed. Considering the conditions that five points are completely observed on an edge, and that the LRF does not collide with the robot manipulating the peg, the range of the feasible viewing direction (ϕ) were determined as shown in Fig. 25. The center of the circular trajectory of the LRF was placed on the vertex at the intersection of the two neighboring edges. Fig. 26 shows a successful peg-in-hole operation.

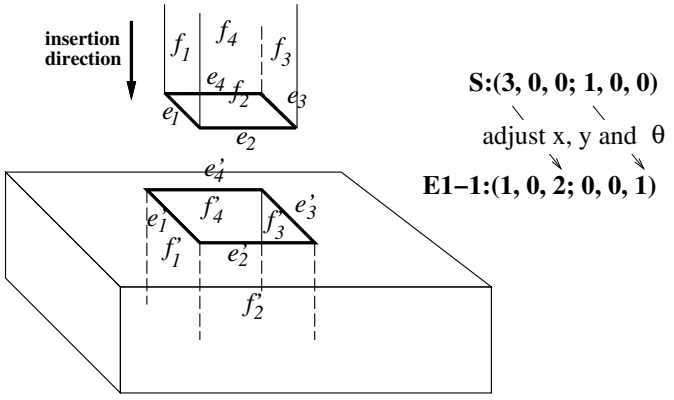


Fig. 24: Face contact analysis of the rectangular peg-in-hole operation. The triplet of DOFs (see Section 2.2) changes from $(3,0,0)$ to $(1,0,2)$.

6.2.2. Comparison of Predicted Success Probability with Actual Success Ratio

We compared the predicted success probability with the actual success ratio in the following two sets of the objects:

Case (a): The cross-section of the peg is a square of $19.05(mm) \times 19.05(mm)$. The clearance ratio of the hole is 0.043. The success region for this case is shown in Fig. 19(a).

Case (b): The cross-section of the peg is a rectangle of $38.1(mm) \times 19.05(mm)$. The clearance ratio of the hole is 0.043. The success region for this case is shown in Fig. 19(b).

In each case, several viewing angle (ϕ in Fig. 25) were selected; for each angle, the same operation was performed 50 times and the numbers of success and of failure were

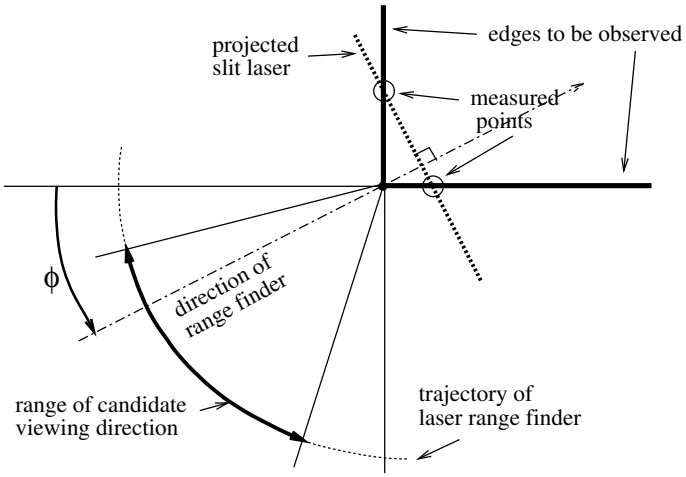


Fig. 25: Top view of candidate viewing directions for observing two edges and faces.

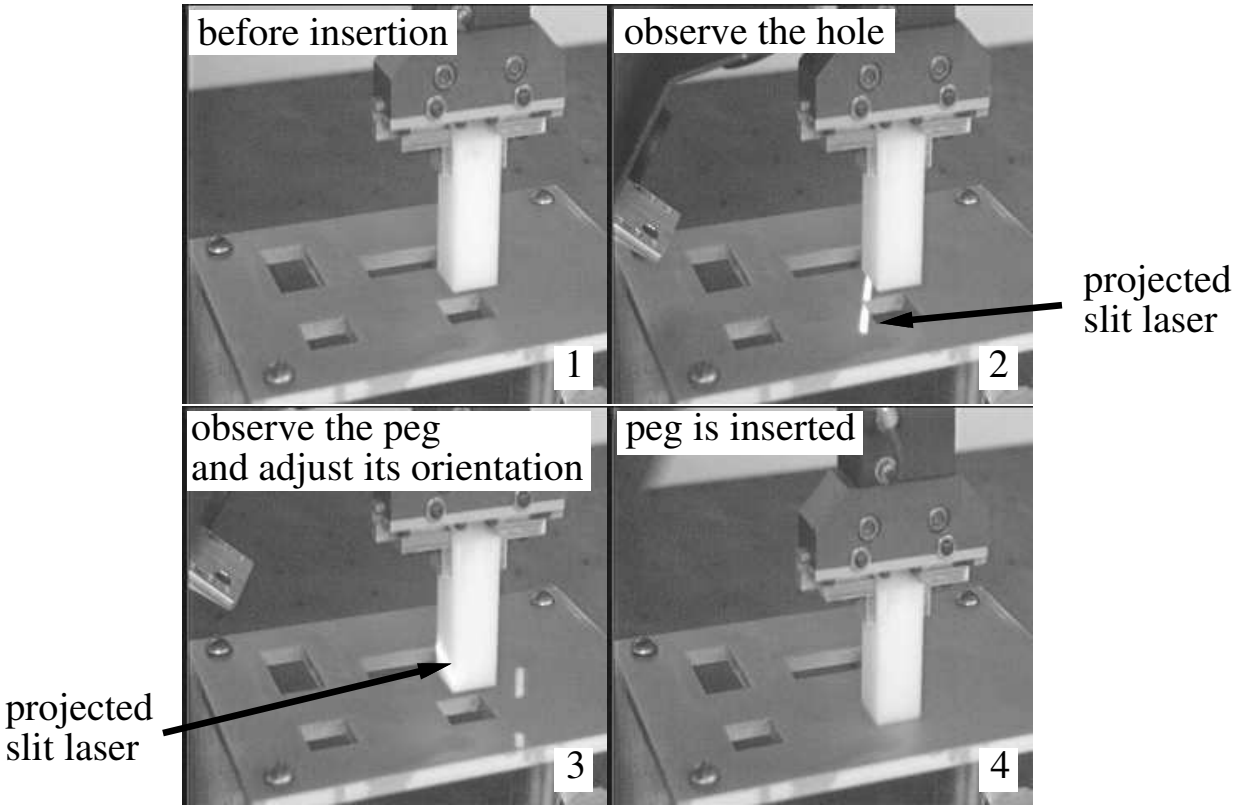
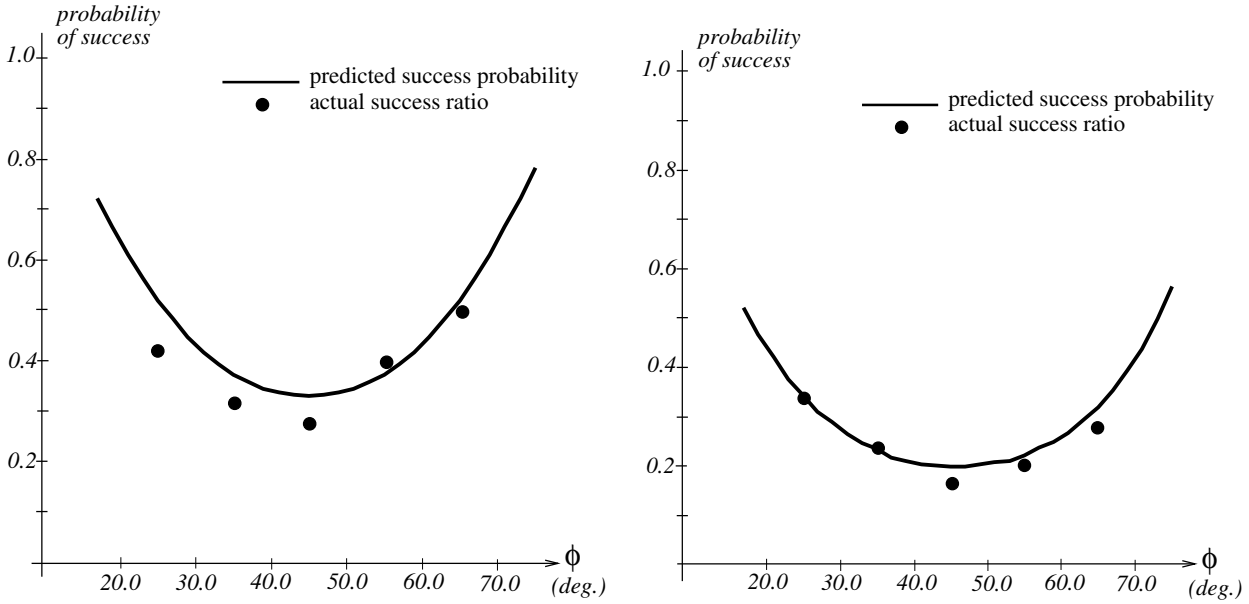


Fig. 26: A successful peg-in-hole operation.

accumulated to calculate actual success ratio.

Since the number of the data points for the peg is much larger than that for the hole, the uncertainty of the peg position is small (about 100 times more accurate) compared to that of the hole position. Thus, we neglected the uncertainty of the peg position and considered only the uncertainty of the hole position in calculation of predicted success probabilities.

Fig. 27 shows the comparison results; the results of actual experiments coincide with the predicted results quite well. We think this result shows the importance of task-oriented evaluation of sensing strategies, i.e., the appropriate sensing strategy should be selected by considering the task to be performed.



Case (a) Case (b)

Fig. 27: Comparison of the predicted success probability with the actual success ratio. The success ratio for each viewing angle ϕ was obtained by repeating the same operation 50 times by the RobotWorld, and by counting the numbers of success and failture of the operation.

6.3. Putting Screwdriver on Bolt

Fig. 28 shows an operation of inserting the tip of a screwdriver into the slot on a bolt head. This operation belongs to group (c) in Fig. 10. By this operation, two degrees of freedom are constrained. The face contacts to be achieved are $(f_1\text{-}f_1')$ and $(f_2\text{-}f_2')$. The candidates for observed features are f_1, f_2, f_3 and f_4 for the screwdriver, and e_1' and e_2' for the hole.

In this case, because of the geometric constraints between manipulators for object handling and for the LRF, the screwdriver and the bolt could not be observed at once. Thus,

the LRF observed only the bolt because the positional uncertainty of the bolt is much larger than that of the screwdriver. Thus, edges e_1' and e_2' of the bolt are observed. Fig. 29 shows a successful operation of putting a screwdriver on a bolt.

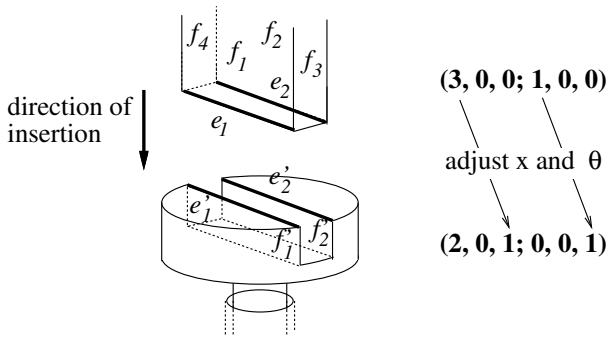


Fig. 28: Contact state analysis of putting a screwdriver on a bolt. Transition in terms of three kinds of DOFs (see section 2) for both translational and rotational ones is also indicated.

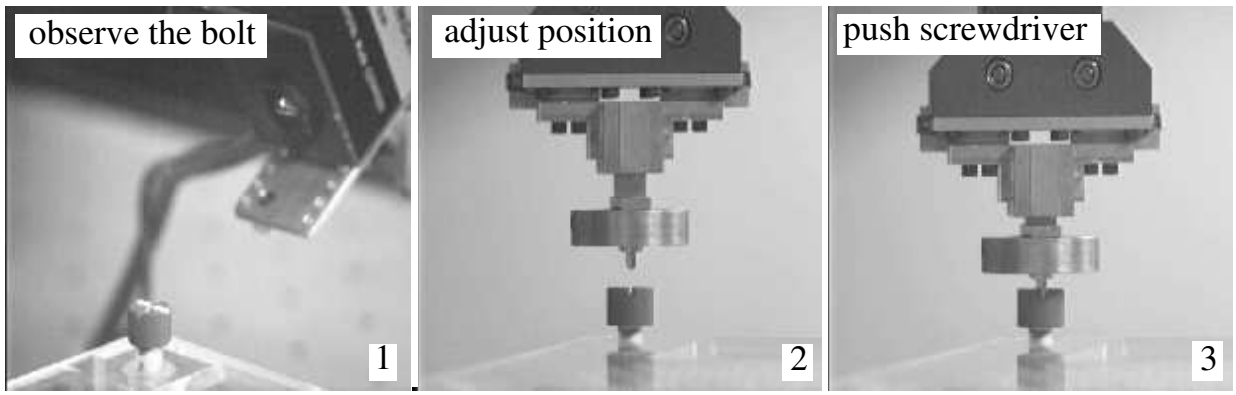


Fig. 29: The screwdriver was successfully inserted into the slot of the bolt head.

Fig. 30 shows the comparison of the predicted success probability with the actual success ratio. Again, the experimental results coincide with each the predicted results quite well.

6.4. Gear Mating

Figure 31 shows a gear-mating operation. This operation belongs to group (e) in Fig. 10. In this operation, *a priori* knowledge about how gears are mated is necessary because there are many potential matches between teeth of gears. First, two virtual edges e_2 and e_2' are generated; one edge is placed on the center of the nearest tooth (or gap) to the line connecting two gear centers; another edge is placed on the center of the nearest gap (or tooth) to the

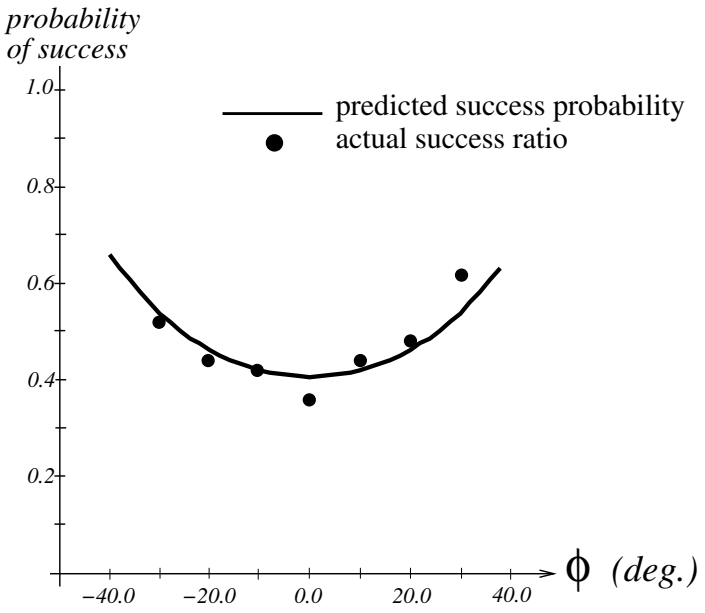


Fig. 30: Comparison of the two success probabilities in the screwdriver-bolt operation. The angle ϕ indicates the relative angle between the direction of the slot and the viewing direction of the range finder. In each viewing angle, the same operation was repeated 50 times to obtain the sucess ratio.

line. Then, the orientation of the inserted gear is adjusted so that these two virtual edges are aligned.

The position of a virtual edge is calculated from the position of the edges on the tooth (or gap) on which the virtual edge is set (see Fig. 32). Assuming that the shape of the tooth is almost rectangular, the virtual edge is obtained by fitting a line of the center points of pairs of edge points.

Since the position of a virtual edge is calculated by measuring several points on the edge, it is safest to place the LRF so that the direction of the slit laser is perpendicular to the edge. Thus, to observe the two virtual edges from one position, the LRF is placed so that the direction of the slit laser is aligned to the line connecting the two gear centers. Fig. 33 shows a successful gear-mating operation.

Fig. 34 shows the comparison of the predicted success probability with the actual success ratio. Also in this case, the experimental results coincide with the predicted results quite well.

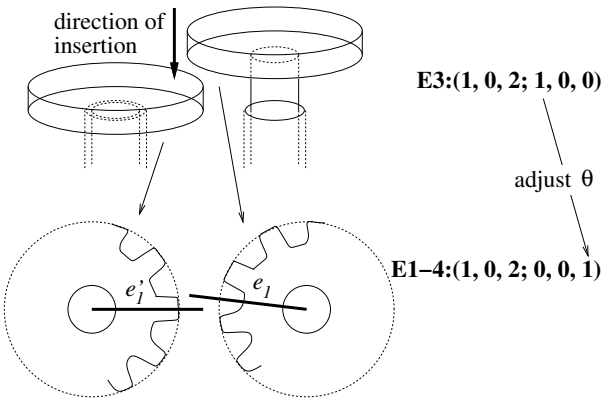


Fig. 31: Contact state analysis of gear mating.

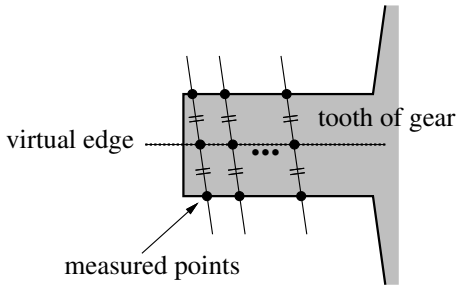


Fig. 32: Measuing the tooth position from edge positions.

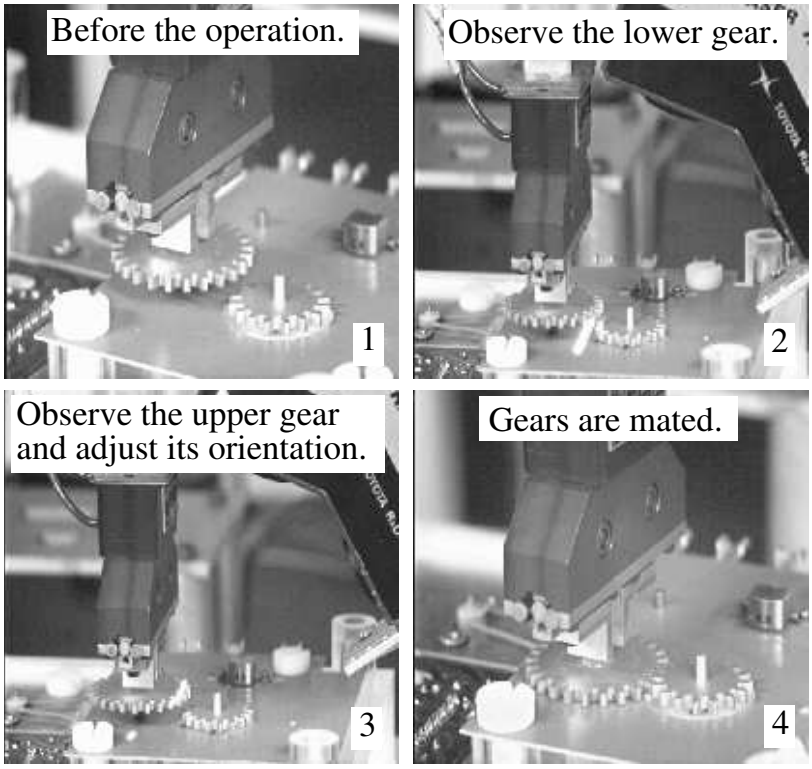


Fig. 33: The gears were successfully mated.

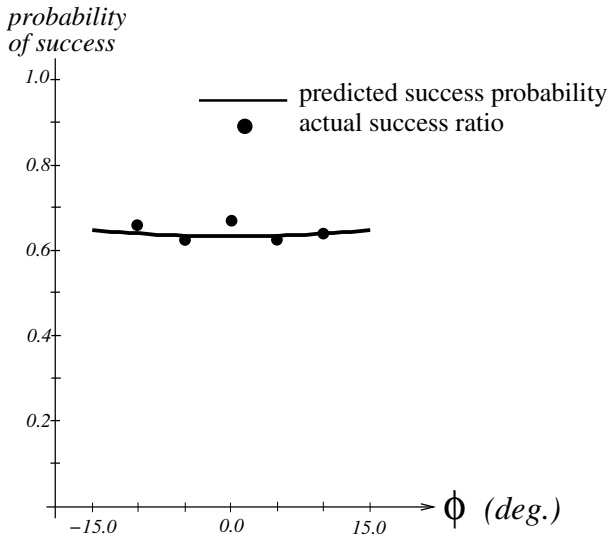


Fig. 34: Comparison of the predicted success probability and the actual success ratio in the gear-mating operation. The angle ϕ indicates the relative anlge between the line connecting two gear centers and the viewing direction of the range finder. In each viewing angle, the same operation was repeated 50 times to obtain the sucess ratio.

7. Task-Oriented Sensing Strategy Generation System

This section describes the structure of task-oriented sensing strategy generation system, which incorporates with various other modules of automatic assembly system such as a task planner and a task executor. Fig. 35 illustrates the outline of task-oriented sensing strategy generation. Before planning, abstract sensing task models, which are abstract templates of sensing strategies, are generated from the task description and the sensor models. At planning time, by instantiating the appropriate sensing task model, the sensing strategy is efficiently generated.

7.1. Task Model and Sensor Model

The description of a task is represented by a structure called an *abstract task model*. An abstract task model associates a state transition with an assembly operation which causes the transition. Each task model has slots for necessary information for performing the operation by a robot, such as assembled objects and geometric relations to be achieved. In addition, the task model contains a robot motion macro and parameters to expand the macro, such as grasping position, departing position and approaching position. An *instantiated task model*, whose slots have actual values, is either generated by observing an assembly operation performed by a human [13], or generated from the task specification by a task planner.

The *sensor model* [12] describes knowledge about a sensor such as features which are

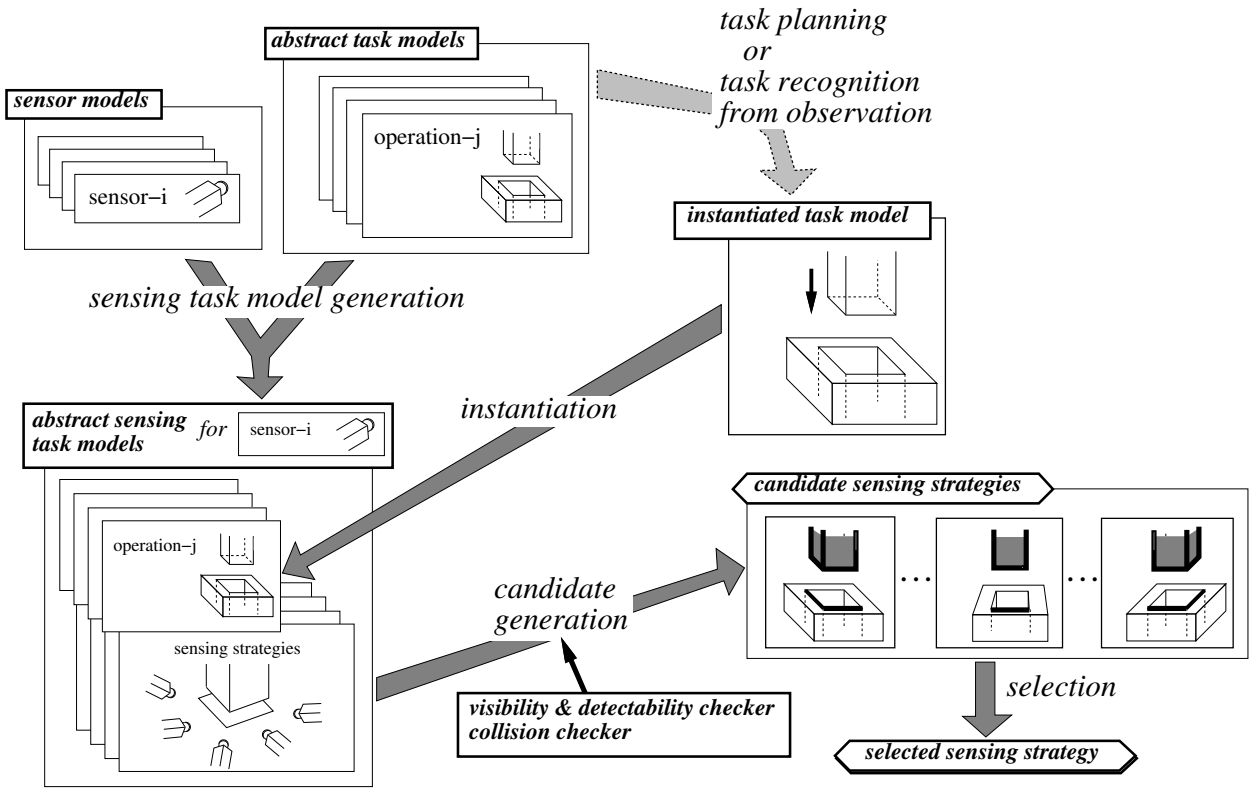


Fig. 35: Framework of task-oriented sensing strategy generation.

observable with the sensor, range of the sensor (distance and field of view), and sensor data uncertainty.

7.2. Sensing Task Model

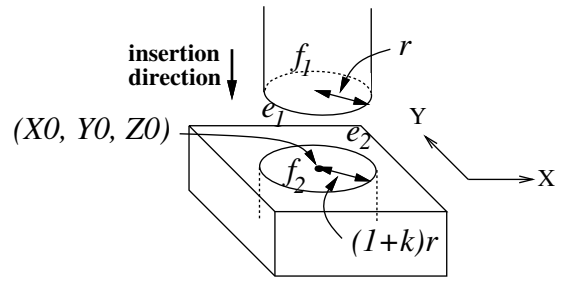
Sensing strategies are dependent upon the sensors used and assembly operations to be performed. An *abstract sensing task model* is generated for a sensor and an assembly operation (or a group of assembly operations) from the sensor model and abstract task models. An abstract sensing task model contains the following information:

- *Transition of the sextuplet*
- *Which degrees of freedom to observe.*
- *Feature set to observe*
- *feasible sensor position set* to observe it.
- *A function for estimating uncertainty* of the relative location of the manipulated object with respect to the stationary environment. Uncertainty estimation is described by a covariance matrix (see Section 4.1).
- *a success region for evaluating sensing strategies* (see Section 4.2).

Some values of the above information are dependent upon geometric values (shape, size and position) of the objects involved in each operation.

In order to efficiently generate visual sensing strategies, we enumerate in advance operations which involve only objects of typical shapes such as rectangular parallelepiped or cylinder, and describe the above infomation in a parameterized form for those operations. Parameters (i.e., the size and the position of objects involved in each operation) are instantiated at planning time by referring to the instantiated task model. Fig. 36 shows a parameterized abstract sensing task model for the operation of inserting a peg with a circular cross-section into a hole.

For those operations that do not have corresponding parameterized information, only the transition of the sextuplet and the degrees of freedom to be observed are stored. The rest of the necessary information is generated from scratch, that is, feasible feature sets are selected by the method described in Section 3.2; uncertainty estimate is calculated by combinining uncertainty estimate for each visual feature into a covariance matrix through matrix transformation based on actual geometric values; a success region for evaluating sensing strategies is derived by the method described in Section 4.3. It would be a useful extension to add a capability to the system which expands the repertoire of "typical" shapes through the interaction with a human expert whenever the system encounters an object for which a parameterized abstract sensing task is not prepared.



transition of sextuplet:	(3,0,0; 1,0,0) → (1,0,2; 1,0,0)
DOFs to be observed:	X and Y
Feature set:	f_1 and e_2
Feasible sensor position:	(on circular trajectory of radius $k+\alpha$ around the hole)
Function to estimate uncertainty:	(obtained from the equation of the circle $(X-X0)^2+(Y-Y0)^2=R$, $Z=Z0$ $(R=r, (1+k)r)$)
Success region:	circle of radius kr

Fig. 36: A parameterized abstract sensing task model for the operation of inserting a peg with a circular cross-section into a hole. f_1 and f_2 indicate faces; e_1 and e_2 indicate edges. The radius of the cross-section of the peg is r; clearance ratio is k.

7.3. Process of Sensing Task Model Generation

Before planning, abstract sensing task models are generated for a sensor model and a set of abstract task models using the following steps:

1. Collect and classify state transitions in which visual information should be used. This step is based on the analysis of transitions of contact states (see section 2).
2. For each state transition groups, enumerate possible shapes of objects, for which parameterized abstract sensing models are prepared. A parameterized abstract sensing model is generated in the following steps:
   (a) Collect feasible sets of visual features to observe.
   (b) Determine feasible sensor positions for each of the above sets, by considering the visibility of the visual features and the collision possibility with them.
   (c) Define a function to estimate the positional uncertainty of the manipulated object.
   (d) Calculate a success region.

7.4. Automatic Sensing Strategy Generation using Sensing Task Model

At planning time, given an instantiated task model and the sensing task models, sensing strategy for the operation is automatically generated by the following steps (see Fig. 35):

1. *Instantiation.* Select the appropriate abstract sensing task model for the current assembly operation, and instantiate it using actual geometric values.
2. *Candidate generation.* From the instantiated sensing task model, possible sensor positions are generated. In this step, considering all objects in the workspace, the visibility and the detectability of visual features are examined. Also, the possibility of collision between the sensor and the environment including other robots is examined.
3. *Selection.* Evaluate each sensing strategy using the evaluation function, and finally select the optimal one.

8. Concluding Remarks

8.1. Summary

We have described a method of systematically generating visual sensing strategies in assembly tasks using the knowledge of the task to be performed. Based on the result of the analysis of the assembly tasks in terms of the transition of face contacts between object surfaces, we can determine which degrees of freedom of the assembled objects should be measured. A set of visual features to be observed are then selected by which the necessary degrees of freedom are measured. Finally, among feasible visual sensing strategies, the one with the highest predicted success probability is selected as the final sensing strategy. The proposed method has been implemented using a line laser range finder as the sensor. The experimental results show the feasibility of the method, and point out the importance of task-oriented evaluation of visual sensing strategies. We also have described a structure of task-oriented visual sensing startegy generation system based on the proposed method.

8.2. Extension to General Sensing Strategy Generation

The proposed method can be applied to other sensing strategy generation problems provided that the description of the task and the model of sensors are given. In order to apply the method to other problems, we would need the following:

- *Primitives* for describing the problem, i.e., for describing the change of the state (or the purpose of the task). We call this kind of primitive the *information primitive*. In this report, we used "degrees of freedom" as an information primitive.
- *Relationships* between observable visual features and obtained information; this information is described with information primitives. This relationship is described in an abstract sensing procedure called a *sensing primitive* (see Section 3.1).
- *Evaluation function.* We used *predicted success probability* as the evaluation function.

Using the above primitives and the function, the outline of task-oriented sensing strategy generation is illustrated as follows (see Fig. 37):

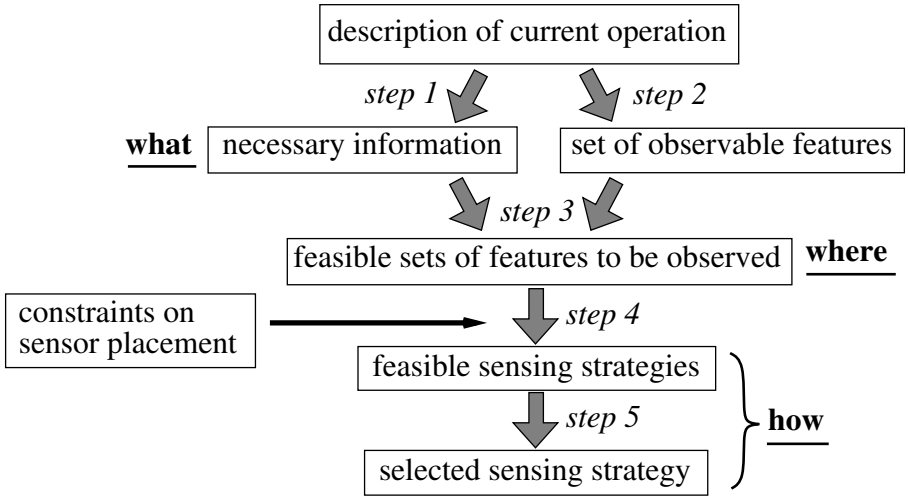


Fig. 37: Outline of task-oriented sensing strategy generation.

Step 1: From the description of the current robotic operation, necessary visual information is determined. This step determines *what* information to extract. If no visual information is required in the current operations, visual sensors are not used.

Step 2: A set of observable features is generated from both the description of the operation and the model of objects.

Step 3: From both the necessary information and the set of observable features, by referring to sensing primitives, sets of visual features to be observed are selected. By observing each set of the features, necessary information will be obtained. This step determines *where* to get necessary information.

Step 4: By considering constraints on observation conditions such as visibility and possibility of collision between sensors and objects, feasible sensing strategies are generated. Each strategy includes not only the sensor placement but also sensor data processing to extract necessary information for the current operation.

Step 5: Among candidates for sensing strategies, the best one is selected based on the task-oriented evaluation. Steps 4 and 5 determine *how* to get necessary information.

Acknowledgement

The authors would like to thank Sing Bing Kang for helpful comments on the draft of this report and Masato Kawade for the experimental setup. This research was conducted at the Task-Oriented Vision Laboratory, the Robotics Institute, Carnegie Mellon University.

Jun Miura was on leave of absence from Osaka University, Osaka, Japan. His stay at Carnegie Mellon University was supported by the Telecommunications Advancement Foundation, Tokyo, Japan.

References

[1] J. Aloimonos. Purposive and Qualitative Active Vision. In *Proc. of 1990 Image Understanding Workshop*, pages 816–828, 1990.

[2] Y. Aloimonos, editor. *Active Perception.* Lawrence Erlbaum Associates, Inc., New Jersey, 1993.

[3] N. Ayache and O. D. Faugeras. Maintaining Representations of the Environment of a Mobile Robot. *IEEE Trans. on Robotics and Automat.*, RA-5(6):804–819, 1989.

[4] R. Bajcsy. Active Perception. *Proceedings of IEEE*, 76(8), 1988.

[5] D. H. Ballard. Reference Frames for Animate Vision. In *Proceedings of IJCAI-89*, pages 1635–1641, 1989.

[6] L. Birnbaum, M. Brand, and P. Cooper. Looking for Trouble: Using Causal Semantics to Direct Focus of Attention. In *Proceedings of the 4th Int. Conf. on Computer Vision*, pages 49–56, 1993.

[7] C. K. Cowan. Automatic Camera and Light-Source Placement Using CAD Models. In *IEEE Workshop on Directions in Automated CAD-Based Vision*, pages 22–31, 1991.

[8] I. D. Horswill. *Specialization for Perceptual Processes.* PhD thesis, Massachusetts Institute of Technology, 1993.

[9] S. Hutchinson. Exploiting Visual Constraints in Robot Motion Planning. In *Proceedings of 1991 IEEE Int. Conf. Robotics and Automat.*, 1991.

[10] S. A. Hutchinson and A. C. Kak. Planning Sensing Strategies in a Robot Work Cell with Multi-Sensor Capabilities. *IEEE Trans. on Robotics and Automat.*, RA-5(6):765–783, 1989.

[11] K. Ikeuchi and M. Hebert. Task Oriented Vision. Technical Report CMU-CS-91-163, School of Computer Science, Carnegie Mellon University, July 1991.

[12] K. Ikeuchi and T. Kanade. Modeling Sensors: Toward Automatic Generation of Object Recognition Program. *Computer Vision, Graphics, and Image Processing*, 48:50–79, 1989.

[13] K. Ikeuchi and T. Suehiro. Toward an Assembly Plan from Observation Part I: Task Recognition With Polyhedral Objects. *IEEE Trans. on Robotics and Automat.*, 10(3):368–385, 1994.

[14] K. Kemmotsu and T. Kanade. Sensor Placement Design for Object Pose Determination with Three Light-Strip Range Finders. In *Proceedings of 1994 IEEE Int. Conf. on Robotics and Automat.*, pages 1357–1364, 1994.

[15] Y. Kuniyoshi and H. Inoue. Qualitative Understanding of Ongoing Human Action Sequences. In *Proceedings of the 13th Int. J. Conf. on Artificial Intelligence*, pages 1600–1609, 1993.

[16] O. Ozeki, K. Higuchi, and S. Yamamoto. Automated Dimension Inspection System for Automotive Plastic Parts with a Laser Probe. In *Proc. of ROBOTS 12 and Vision '88 Conference*, pages 5–51–5–60, Detroit, MI, 1988.

[17] R. D. Rimey. Control of Selective Perception Using Bayes Nets and Decision Theory. Technical Report Technical Report 468, Computer Science Department, The University of Rochester, December 1993.

[18] S. Sakane and T. Sato. Automatic Planning of Light Source and Camera Placement for an Active Photometric Stereo System. In *1991 IEEE Int. Conf. on Robotics and Automat.*, pages 1080–1087, 1991.

[19] V. Scheinman. A Multiple Robot Vision Guided Assembly System. In R. Bolles and B. Roth, editors, *Robotics Research 4*. The MIT Press, 1987.

[20] A.J. Spyridi and A.A.G. Requicha. Automatic Programming of Coordinate Measuring Machines. In *1995 IEEE Int. Conf. on Robotics and Automat.*, pages 1107–1112, 1994.

[21] T. Suehiro and K. Takase. Representation and Control of Motion in Contact and Its Application to Assembly Tasks. In *Proceedings of Int. Symp. on Robotics Res.*, pages 367–374, 1989.

[22] K.A. Tarabanis, P.K. Allen, and R.Y. Tsai. A Survey of Sensor Planning in Computer Vision. *IEEE Trans. on Robotics and Automat.*, 11(1):86–103, 1995.

[23] K.A. Tarabanis, R.Y. Tsai, and P.K. Allen. The MVP Sensor Planning System for Robotic Vision Tasks. *IEEE Trans. on Robotics and Automat.*, 11(1):72–85, 1995.

[24] P. Whaite and F. P. Ferrie. From Uncertainty to Visual Exploration. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 13(10):1038–1049, 1991.

[25] C.C. Yang, M.M. Marefat, and R.L Kashyap. Active Visual Inspection Based on CAD Models. In *1995 IEEE Int. Conf. on Robotics and Automat.*, pages 1120–1125, 1994.

[26] H. Zhang. Optimal Sensor Placement. In *Proceedings of 1992 IEEE Int. Conf. on Robotics and Automat.*, pages 1825–1830, 1992.