Fine Motion Planning Using Skill-Based Backprojection with Uncertainties in Control, Visual Sensing and Model

Akira Nakamura*, Takashi Suehiro*, Kazuo Machida**

*Intelligent Systems Institute **Energy Electronics Institute
National Institute of Advanced Industrial Science and Technology
AIST Tsukuba Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan
E-mail: a-nakamura@aist.go.jp, t.suehiro@aist.go.jp, machida-k@aist.go.jp

Abstract

Manipulator tasks such as assembly can be generally divided into several motion primitives called "skills." Skill-based motion planning is an effective way to execute complicated tasks. When planning an assembly process, fine motion planning such as backprojection in the configuration space is often used. At IROS '96, we introduced the concept of skill into the backprojection method. General and skillful planning can be derived by skill-based backprojection, and manipulation tasks that approximate those performed by humans can be achieved. At that time, we considered only manipulator control errors to be planning uncertainties. In practice, however, visual sensing errors and model errors cannot be ignored. This paper describes our skill-based backprojection method for handling control, visual sensing and model errors by fine motion planning.

Key words: manipulation skill, fine motion planning, backprojection, skill library

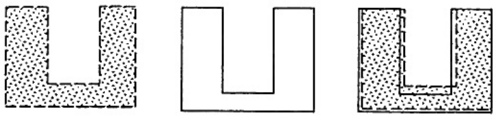
1. Introduction

In recent years, robots have been rapidly introduced into several fields. In order to play a part in extensive fields, manipulation robots need to perform various tasks using special techniques. By analyzing human task motions such as assembly and disassembly, we have shown that movements consist of several significant motion primitives. We called these "skills" and have shown that most manipulator tasks can be segmented into sequences of skills [1]-[5].

Various fine motion planning techniques have been studied for such manipulation tasks as assembly and disassembly. Fine motion planning in a configuration space has been studied as a method of artificial intelligence. In a configuration space planning is simplified since an object is represented as a point within the space. Lozano-Perez et al. proposed the concept of pre-image, and their planning technique used generalized damping while accounting for sensor and control uncertainties [6]. Erdmann proposed the backprojection method in which the goal region is projected in reverse using an error cone, and this made it easier to obtain the reachable region to the goal than by using the pre-image [7]. Donald proposed motion planning with uncertainty, not only in sensing and control, but also in geometric models [8]-[9]. Latombe and Shekhar et al. extended motion planning to a multi-step approach, using pre-image back-chaining [10], and then analyzed goal recognition capability in motion planning with uncertainty [11]. Christiansen proposed an empirical backprojection method based on a data-intensive approach that differed from traditional analytical techniques [12]. More recently, probabilistic backprojections have been proposed [13]-[16].

We proposed a fine motion planning method based on backprojection for various tasks that were composed of several manipulation skills at IROS '96 [17]. We showed that general and skillful planning can be derived with ease. However, we dealt then with manipulator control errors only as an uncertainty, though sensing and modeling errors may also impact the reliability of task achievement.

Uncertainties in manipulation can be classified into three kinds of errors: control errors, sensing errors and model errors. We showed a fine motion planning method using skill-based backprojection to handle control and visual sensing errors at ICAR '99 [18]. However, model errors cannot be ignored if a rough geometric model is used or exact planning is needed. In general, planning with model errors is complex since a large number of model errors must be taken into account. For example, the method of motion planning with model uncertainty that was proposed by Donald is calculated in a



(a) Real Object (b) Model (c) Model matching

Fig. 1 Model matching with model errors

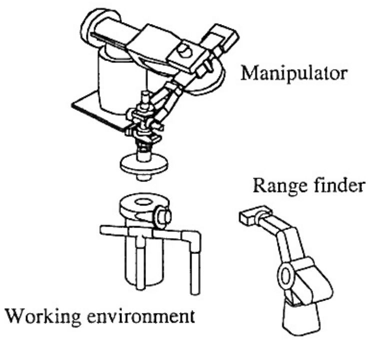
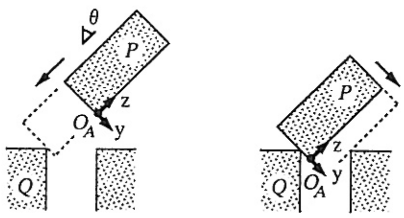


Fig. 2 Manipulation system with a hand-eye

generalized configuration space by increasing dimensions [8]. However, it is possible that planning can be simplified by considering the process of model matching and the adaptability of manipulation skill. We will consider a real object as shown in Fig. 1(a) which is in the shape of a rectangle with a hole. The model is shown in Fig. 1(b) and is a little larger than the real object. Fig. 1(c) shows model matching using data on the upper-side and right-side edges of the object. Planning of a peg-in-hole task is then simple since errors on these two edges are not taken into account. Moreover, it can be simplified by taking into account the tolerance brought by manipulation skills in the task.

This paper describes our proposal for a fine motion planning method using skill-based backprojection to handle not only control errors but also visual sensing errors and model errors. Our method assumes that a hand-eye range finder set on a manipulator (Fig. 2) is used since it is necessary to obtain range data from a position that assures highly reliable task achievement.

The next section explains our concept of manipulation skills, skill in the configuration space, skill-based backprojection to handle only control errors, and composition of the skill sequence. The backprojection method with visual sensing errors and model errors is explained in section 3. The processes of visual sensing, geometric modeling and execution of a task are explained for task and skill levels in section 4. Our proposed method is demonstrated using an example of a peg-in-hole task in section 5.



(a) Move-to-touch$_f$ skill (b) Move-to-touch$_c$ skill

Fig. 3 Move-to-touch skills

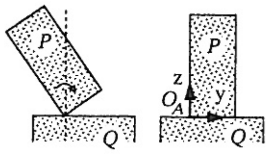


Fig. 4 Rotate-to-level skill

2. Manipulation Skills

This section explains our concept of skills. See References 1 - 5 for more details. In this section, we will take into account only the control errors as uncertainties.

2.1. Skill Primitives

In assembly and disassembly tasks, skills in which the contact states vary are particularly significant. We will consider three skills which play an important part in such tasks: move-to-touch, rotate-to-level and rotate-to-insert. Most assembly manipulation tasks are comprised of these three skills. In this paper, we consider skill motions as occurring in two-dimensional environments.

(1) Move-to-touch Skill

The move-to-touch skill is the transition from a free to a vertex-to-face contact between a grasped object P and another object Q in velocity control mode (Fig. 3(a)). A similar transition of maintaining contact in a different direction of motion is also part of this skill (Fig. 3(b)). These two transitions are represented respectively by the move-to-touch$_f$ skill and the move-to-touch$_c$ skill.

(2) Rotate-to-level Skill

The rotate-to-level skill is the transition from a vertex-to-face contact to an edge-to-face contact (Fig. 4). This skill is performed with pushing force.

(3) Rotate-to-insert Skill

When the clearance is small in an insertion task, it is generally difficult to achieve the state in Fig. 5(b)

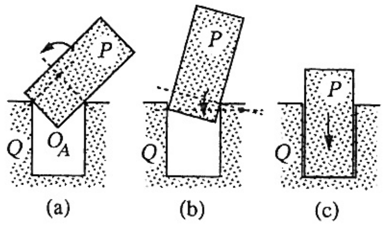
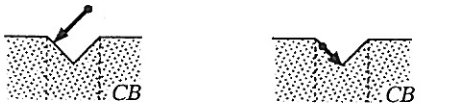


Fig. 5 Rotate-to-insert skill



(a) Move-to-touch$_f$ skill (b) Move-to-touch$_c$ skill

Fig. 6 Move-to-touch skills in C-space

Fig. 7 Move-to-touch$_f$ skill with control uncertainty

directly. The state in Fig. 5(a) is achieved first using other skills, such as the skill sequence of Fig. 3(a) and (b). The state in Fig. 5(b) is then accomplished by gradually raising the object while maintaining contact as in Fig. 5(a). The rotate-to-insert skill is the motion of rotating the object P obliquely into the hole in another object Q to insert it accurately. In our study, we assume that the rotate-to-insert skill also includes the pressing motion required to achieve the goal of the insertion task (Fig. 5(c)).

2.2. Skill Commands

We conceptually represent these skill primitives as operation commands.

(touch_v w a): move-to-touch$_f$ skill of transition of velocity a along the w-axis.

(touch_p w a v b): move-to-touch$_c$ skill of transition of velocity a along the w-axis with pressing force b on the v-axis to maintain constant contact. There is also the possibility that the actual direction of transition is inclined due to the constraints of contact. In such a case, we regard the w-element of the actual direction of transition as velocity a.

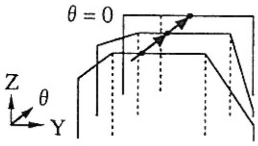


Fig. 8 Rotate-to-level skill in C-space

(rotate w c v b): rotate-to-level skill of rotation of angular velocity c around the w-axis with pushing force b on the v-axis.

(insert w c v b): rotate-to-insert skill of rotation of angular velocity c around the w-axis and pressing motion with pushing force b on the v-axis.

In these commands, w and v correspond to x, y or z. As described in this paper, we assume that a, b and c correspond to either a "+" or "-", not a value. For example, the commands of motion in Fig. 3(a), (b) are represented respectively by *(touch_v z -)* and *(touch_c y + z -)*.

2.3. Skills in Configuration Space and Backprojection

Next, we will discuss the trajectory of skill motions in the configuration space.

(1) Move-to-touch Skill

In the configuration space the trajectories of the object P being manipulated with the move-to-touch$_f$ skill in Fig. 3(a) and the move-to-touch$_c$ skill in Fig. 3(b) are drawn in Fig. 6(a) and (b), respectively. CB is a C-obstacle which represents an object Q in C-space based on the reference point O_A of object P. To take into account the uncertainty of control when using the move-to-touch$_f$ skill, we have drawn the trajectory using the control uncertainty cone shown in Fig. 7.

(2) Rotate-to-level Skill

Assuming that reference point O_A is a vertex in prior contact with the surface (Fig. 4), the position of O_A on the YZ-plane in the configuration space stays constant (Fig. 8).

(3) Rotate-to-insert Skill

The trajectory of the object in the configuration space when manipulated with the rotate-to-insert skill is shown in Fig. 9. The transfer motion of the vertex from Fig. 5(a) to Fig. 5(b) happens at orientation $\theta = \theta_1$ in Fig. 9, where the phase of C-obstacle CB changes.

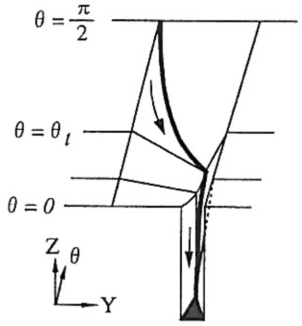
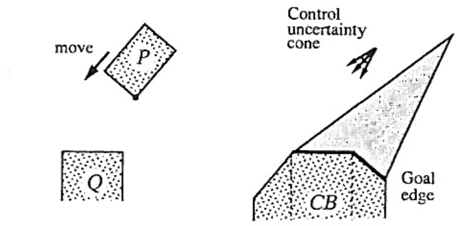


Fig. 9 Rotate-to-insert skill in C-space



(a) Move-to-touch$_f$ skill (b) Backprojection

Fig. 10 Backprojection of move-to-touch$_f$ skill with uncertainty of control

The backprojection of each skill is derived by the reverse trajectory from each goal [7]. For example, the backprojection of the move-to-touch$_f$ skill (Fig. 10(a)) is drawn by projecting a similar velocity cone from the goal edge (Fig. 10(b)).

2.4. Construction of Skill Library and Composition of Skill Command Sequence

First, the skill library is constructed in advance. The trajectories drawn in the configuration space are derived for several skill primitives such as the above-mentioned skills. A skill library consisting of skill primitives expressed by trajectories in the configuration space is thus constructed.

Next, a command sequence to perform a specific task is created using skill primitives from the skill library. In this paper, we will discuss assembly as a task, and assume that the command sequence is arranged in increasing number of contact points for each skill primitive. Therefore, multi-step backprojections are made up as the number decreases at each step, since we take the reverse direction of time into consideration. Furthermore, we assume that the objects in the workspace are modeled as polygons. We exclude cases of low probability such as a direct transition from one-point contact to three-point contact, and omit parameters in skill commands. Then, variations in the number of contact points in the skill commands are shown as follows.

Table 1 Variations in the number of contact points in backprojection for each skill command

(touch_v)	1→0
(touch_p)	2→1 3→2
(rotate)	2→1
(insert)	3→2

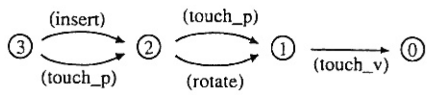


Fig.11 State diagram for the number of contact points in backprojection

(1) *(touch_v)*

The move-to-touch$_f$ skill, that is, a transition from the free state to a one-point contact (Fig. 3(a)) occurs by this command.

(2) *(touch_p)*

A transition occurs from a one-point contact to a two-point contact (Fig. 2(b)), or from a two-point contact to a three-point contact, based on the move-to-touch$_e$ skill.

(3) *(rotate)*

A transition from a one-point contact to a two-point contact occurs (Fig. 3).

(4) *(insert)*

We consider three-point contact as a goal state of insertion by taking actual manipulation into account.

For backprojection, these variations are revised in reverse sequence as shown in Table 1. By putting the number of contact points on the nodes, a state diagram describing the variations in the backprojection (Table 1) is derived as shown in Fig. 11. Candidate command sequences can be obtained by applying the number of contact points at the goal and at start of the task to Fig. 11.

The method to decide the most suitable command sequence and most suitable initial state is shown in [17]. The most suitable sequence by which the largest region of backprojection can be obtained and the instructed task can actually be performed is chosen. The most suitable initial state is chosen from the viewpoint of task achievement reliability as shown in [17].

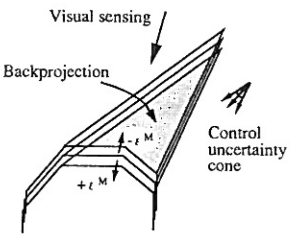
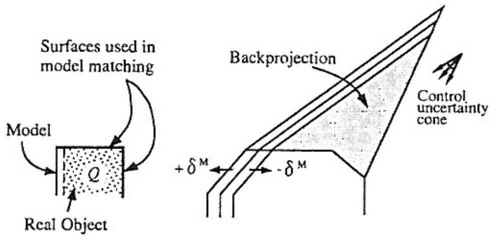


Fig. 12 Backprojection with visual sensing errors



(a) Model Matching (b) Backprojection

Fig. 13 Backprojection with model errors

3. Backprojection with Three Kinds of Uncertainties

In this section we will explain backprojection of manipulation skills that takes into account three kinds of uncertainties. In section 2 we explained backprojection using only control errors and we did not deal with the two other uncertainties: sensing errors and model errors. However, if planning must have high reliability, these two uncertainties cannot be ignored.

First, we will consider backprojection that takes into account visual sensing errors. In general, position data obtained by a range finder will have large errors in depth data. Therefore, for simplicity, we consider only the sensing errors in depth from a range finder. Uncertainty of a point in C-space can thus be expressed by a line segment. On the other hand, if a grasped object P is in contact with a static object Q, relative position errors between these two objects P and Q are not likely to occur. Therefore, we will consider a skill in which both objects P and Q are not in contact at the start, that is, the move-to-touch$_F$ skill (Fig. 10). Assuming that the maximum value of the visual sensing error ε in the depth is ε^M, the backprojection is derived by moving the object Q containing the (sub)goal edges within $\pm\varepsilon^M$ in the direction of the sensing and then extracting the overlapping regions of all backprojections (Fig. 12). When the directions of the visual sensing and manipulation are the same, the size of the backprojection is largest. However, it is difficult to achieve visual sensing from this viewpoint because of collision avoidance between the vision system and manipulator. Therefore, visual sensing should be performed as closely as possible from the direction of manipulation. While we considered errors in the depth direction, if it is necessary to take into account errors in other directions, backprojection can be similarly derived by changing the uncertainty of a point to an elliptical region.

Next, we will consider backprojection that takes into account model errors. In general, a geometric model does not perfectly express the real object. When model matching is done, gaps occur between the real object and the model. By considering surfaces, edges and vertices used in model matching, there will be parts in which it is not necessary to consider gaps, so the number of parts in which it is necessary to consider gaps can be decreased. Moreover, it is possible that the number can be decreased by taking into account tolerance in the manipulation skills. For simplicity, we will explain backprojection with model errors in the move-to-touch$_F$ skill (Fig. 10). Assuming that the upper-side and right-side edges are used in model matching (Fig. 13(a)) and that the maximum value of the model error δ in the direction of a gap on the upper-side edge is δ^M, the backprojection is derived by varying the gap within $\pm\delta^M$ in the direction of the edge and then extracting the overlapping regions of all backprojections (Fig. 13(b)). While we assumed that the angle between the upper-side and right-side edges had no error, if the inconsistency of the angle cannot be ignored, the backprojection can be derived by model matching in which the upper-side edge is given priority. Moreover, while we considered only the model errors of the static object Q, if the model errors of the grasped object P cannot be ignored, the backprojection can be derived by varying a C-obstacle CB with the condition that a reference point of the object P has no error.

We will next consider backprojection that takes into account three uncertainties: control errors, visual sensing errors and model errors. This backprojection is derived by moving all parameters in uncertainties and then extracting the overlapping regions of all backprojections.

4. Process of Sensing, Modeling, Planning and Execution

We will now describe the procedure of sensing, modeling, planning and execution for a manipulation task. If the models of each object are precise and range data is obtained correctly, the environment model of the working region of the robot can be constructed

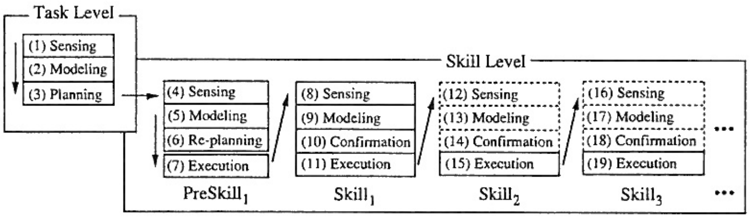


Fig. 14 Process flow

accurately. Thus, overall manipulation planning should be able to be carried out only by initial visual sensing. In reality, however, visual sensing errors and model errors cannot be ignored. Therefore, visual sensing and modeling should be carried out at each step just before planning the task and skills. Figure 14 shows the procedures of sensing, modeling, planning and execution. In this scheme, planning of the task level is first performed, and then executions of skill level are carried out according to sequences derived from the task planning.

<Step 1> Task Level

At the task level, the skill sequence comprising the task is decided and backprojection is derived. First, visual sensing of the working environment of the robot is carried out using a range finder and modeling is performed. Next, planning is done, and skill command sequences and backprojection are derived by the method shown in 2.4.

Measurement and modeling in this level decide the global arrangement of the objects in the working environment of the robot. Since global and rough data is used, the environment model often has some uncertainty. The backprojection derived in this level may likewise have some uncertainty.

<Step 2> Skill Level

At the skill level, each skill in the command sequence $\{Skill_1, Skill_2, \cdots\}$ is executed in order. Before the sequence is performed, transition of the grasped object P to the initial state is carried out. We represent the transition as $PreSkill_1$.

(2.1) $PreSkill_1$

First, visual sensing of the object Q including the goal or subgoal is carried out as closely as possible from the direction to the backprojection derived at the task level. Second, geometric modeling of the object Q is carried out. Third, planning occurs. Since local and precise data is used, more precise planning can be performed. During this re-planning, the exact backprojection taking into account visual sensing errors and model errors is derived

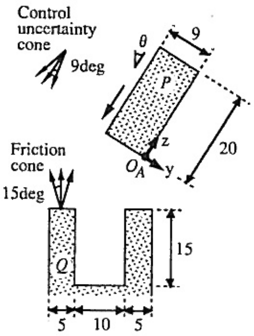


Fig. 15 Peg-in-hole

by the method shown in section 3. Next, transition to the initial state is executed.

(2.2) $Skill_i$ $\{i = 1, 2, \cdots\}$

In this step, sensing, modeling and execution of each skill in the command sequence are carried out. Sensing and modeling are performed both for the object Q including the (sub)goal and the grasped object P to make sure that the grasped object P actually exists in each start region. To be exact, we confirm by transforming the grasped object P to a point in C-space.

First, sensing and modeling for $Skill_1$ are carried out. Then, visual sensing is performed from the same direction as $PreSkill_1$. Next, the position of the grasped object P moved by $PreSkill_1$ is checked. Since visual sensing errors for the grasped object P also have to be taken into account, the backprojection becomes smaller. This reduction of the backprojection will be explained with an example in the next section. After the position is confirmed, $Skill_1$ is executed.

Next, the same process is carried out for each $Skill_i$ $\{i = 2, 3, \cdots\}$. However, sensing, modeling and confirmation with respect to $Skill_i$ can be omitted if the subgoal of $Skill_{i-1}$ is included in the backprojection of $Skill_i$.

5. Example

We will explain our method of planning using a peg-in-hole task (Fig. 15) as an example. All parameters of both objects P and Q are the same as those in [17].

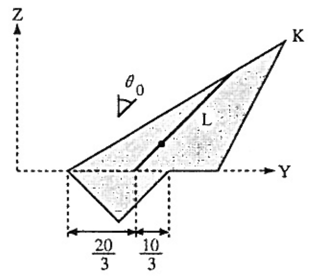


Fig. 16 Most suitable initial state

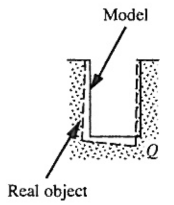


Fig. 17 Holes of model and real object

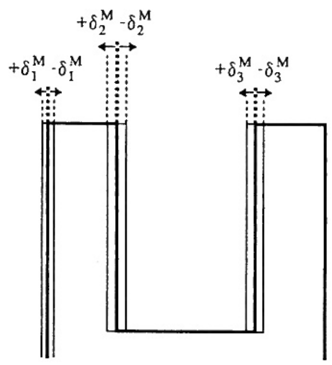


Fig. 18 Model errors

<Step 1> Task level

First, visual sensing for the working environment of the manipulation robot is performed from an arbitrary viewpoint using a range finder and modeling is performed. Next, the sequence of skill commands is derived. The results of the example in [17] show that the most suitable reverse command sequence is *{ (insert) → (touch_p) → (touch_v) }*. Therefore, each step comprising the task is as follows.

$PreSkill_1$: transition to initial state by position control

$Skill_1$: move-to-touch skill in $-z$-direction

$Skill_2$: move-to-touch skill in y-direction

$Skill_3$: rotate-to-insert skill

Furthermore, the same example [17] shows that the most suitable initial state is an orientation $\theta_0 \fallingdotseq 45.8$ *deg* and a position (Y_0, Z_0) on the line segment L in Fig. 16.

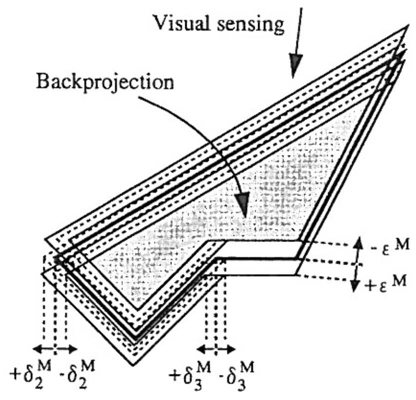


Fig. 19 Backprojection with model errors and visual sensing errors at Q

<Step 2> Skill level

(2.1) *$PreSkill_1$*

First, visual sensing of the object Q is carried out using a range finder and model matching is performed. In general, some errors exist between a real object and the model. We will assume that the upper-side and right-side edges are used in model matching, as we described in section 3. Then, the upper-side edge has no error, but the hole has errors as shown in Fig. 17. By taking into account tolerance of skill primitives in a peg-in-hole task, however, only three errors δ_1, δ_2, δ_3 at the corners as shown in Fig. 18 will influence the backprojection. Then, the backprojection taking into account these errors is derived as shown in Fig. 19. Next, the grasped object P is transferred into the region of the backprojection by the manipulation.

(2.2) *$Skill_i$ {i = 1, 2, ⋯}*

(i) *$Skill_1$*

Sensing and modeling is carried out from the same viewpoint as *$PreSkill_1$*. Since visual sensing errors for the grasped object P also have to be taken into account, the backprojection becomes smaller by ε^M in the sensing direction as shown in Fig. 20. If the grasped object exists within this region, *$Skill_1$* is executed.

(ii) *$Skill_2$*

Since the subgoal of *$Skill_1$* is included in the backprojection of *$Skill_2$*, sensing, modeling and confirmation of *$Skill_2$* can be omitted. *$Skill_2$* is therefore executed following the execution of *$Skill_1$*.

(iii) *$Skill_3$*

This is done similarly to the procedure in *$Skill_2$*.

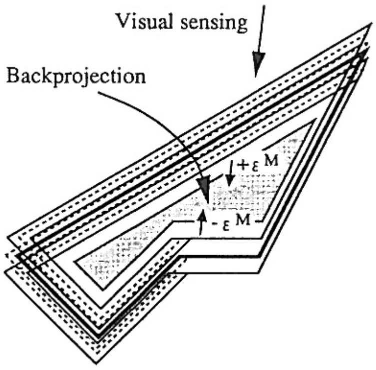


Fig. 20 Backprojection with model errors at Q and visual sensing errors at P and Q

6. Conclusion

We have shown fine motion planning using skill-based backprojection that takes into account uncertainties in control, visual sensing and model. Since we extended the skill-based backprojection method we proposed at IROS '96 to handle visual sensing errors and model errors, our technique can now be applied to a real-world system in which these uncertainties cannot be ignored. The reliability of task achievement can clearly be increased.

In the future we will study an approach to derive more appropriate position and orientation for the visual sensing and more suitable model matching. We must also take into consideration a skill-based backprojection method to handle these three kinds of uncertainties in a three-dimensional environment for application to real-world tasks.

References

[1] T. Hasegawa, T. Suehiro, T. Ogasawara, T. Matsui, K. Kitagaki and K. Takase, "An Integrated Tele-Robotics System with a Geometric Environment Model and Manipulation Skills," '90 Int. Workshop on Intelligent Robots and Systems, pp. 335-341, Tsuchiura, Japan, July 1990.

[2] T. Suehiro and K. Takase, "Skill based manipulation system," (in Japanese). Journal of the Robotics Society Japan, Vol. 8, No. 5, pp. 551-562, Oct. 1990.

[3] T. Hasegawa, T. Suehiro and K. Takase, "A Robot System for Unstructured Environments Based on an Environment Model and Manipulation Skills," '91 Int. Conf. on Robotics and Automation, pp. 916-923, Sacramento, USA, April 1991.

[4] T. Hasegawa, T. Suehiro and K. Takase, "A Model-Based Manipulation System with Skill-Based Execution in Unstructured Environment," '91 Int. Conf. on Advanced Robotics, pp. 970-975, Pisa, Italy, June 1991.

[5] T. Hasegawa, T. Suehiro and K. Takase, "A Model-Based Manipulation System with Skill-Based Execution," IEEE Trans. on Robotics and Automation, Vol. 8, No. 5, pp. 535-544, Oct. 1992.

[6] T. Lozano-Perez, M. T. Mason and R. H. Taylor, "Automatic Synthesis of Fine-Motion Strategies for Robots," Int. J. Robotics Research, Vol. 3, No. 1, pp. 3-24, Spring 1984.

[7] M. Erdmann, "Using Backprojections for Fine Motion Planning with Uncertainty," Int. J. Robotics Research, Vol. 5, No. 1, pp. 19-45, Spring 1986.

[8] B. R. Donald, "Robot Motion Planning with Uncertainty in the Geometric Models of the Robot and Environment: A Formal Framework for Error Detection and Recovery," '86 Int. Conf. on Robotics and Automation, pp. 1588-1593, San Francisco, USA, April 1986.

[9] B. R. Donald, "Planning Multi-Step Error Detection and Recovery Strategies," '88 Int. Conf. on Robotics and Automation, pp. 892-897, Philadelphia, USA, April 1988.

[10] J. C. Latombe, A. Lazanas and S. Shekhar, "Robot Motion Planning with Uncertainty in Control and Sensing," Artificial Intelligence, Vol. 52, pp. 1-47, Nov. 1991.

[11] S. Shekhar and J. C. Latombe, "On Goal Recognizability in Motion Planning with Uncertainty," '91 Int. Conf. on Robotics and Automation, pp. 1728-1733, Sacramento, USA, April 1991.

[12] A. D. Christiansen, "Manipulation Planning from Empirical Backprojections," '91 Int. Conf. on Robotics and Automation, pp. 762-768, Sacramento, USA, April 1991.

[13] R. C. Brost and A. D. Christiansen, "Probabilistic Analysis of Manipulation Tasks: A Research Agenda," '93 Int. Conf. on Robotics and Automation, pp. 549-556, Atlanta, USA, May 1993.

[14] R. C. Brost and A. D. Christiansen, "Probabilistic Analysis of Manipulation Tasks: A Conceptual Framework," Int. J. Robotics Research, Vol. 15, No. 1, pp. 1-23, Feb. 1996.

[15] S. M. LaValle and S. A. Hutchinson, "An Objective-Based Stochastic Framework for Manipulation Planning," '94 Int. Conf. on Intelligent Robots and Systems, pp. 1772-1779, Munich, Germany, Sept. 1994.

[16] S. M. LaValle and S. A. Hutchinson, "An Objective-Based Framework for Motion Planning under Sensing and Control Uncertainties," Int. J. Robotics Research, Vol. 17, No. 1, pp. 19-42, Jan. 1998.

[17] A. Nakamura, T. Ogasawara, T. Suehiro and H. Tsukune, "Skill-Based Backprojection for Fine Motion Planning," '96 Int. Conf. on Intelligent Robots and Systems, pp. 526-533, Osaka, Japan, Nov. 1996.

[18] A. Nakamura, T. Suehiro, Kazuo Machida and H. Tsukune, "Fine Motion Planning Using Skill-Based Backprojection to Handle Control and Visual Errors," '99 Int. Conf. on Advanced Robotics, pp. 31-38, Tokyo, Japan, Oct. 1999.