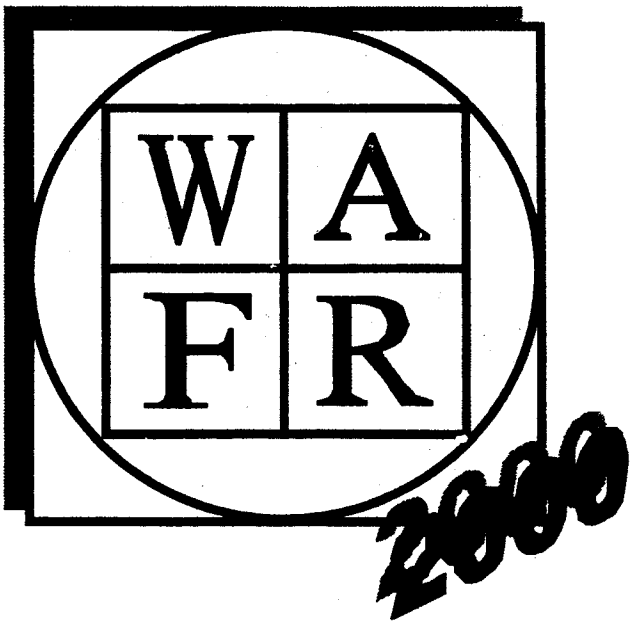


# Fourth International Workshop on Algorithmic Foundations of Robotics

## Pre-Prints of the Proceedings

March 16-18, 2000
Dartmouth College
Hanover, New Hampshire

# Pulling Motion Based Tactile Sensing

Makoto Kaneko and Toshio Tsuji, *Faculty of Engineering, Hiroshima University, Higashi-Hiroshima, 739-8527, Japan*

## Abstract

An algorithm for detecting the shape of 2D concave surface by utilizing a tactile probe is discussed. Pulling a tactile probe whose tip lies on an object's surface can be easily achieved, while pushing it is more difficult due to stick-slip or blocking up with irregular surface. To cope with the difficulty of pushing motion on a frictional surface, the proposed sensing algorithm makes use of the pulling motion of tactile probe from a local concave point to an outer direction. The algorithm is composed of three phases, local concave point search, tracing motion planning, and infinite loop escape. The proposed algorithm runs until the tactile probe detects every surface which it can reach and touch. We show some computer simulations obtained along the proposed algorithm.

## 1 Introduction

This paper focuses on the shape detection of 2D surface by utilizing a tactile probe which can detect any contact point between it and environment. One emphasis of our research is to study how the probe motion should be planned for the object including concave surface. Suppose that human moves his (or her) finger tip on the table as shown in Figure.1, where (a) and (b) denote the pushing and the pulling modes, respectively. It should be noted that pushing motion is not achieved smoothly due to appearance of the well-known stick-slip and often blocked by a small irregularity on the surface, while pulling motion can be achieved even under significant friction. Considering this fact, we discuss the pulling motion based algorithm where no pushing motion appears. If the environment's surface is unknown, however, whether the resulting motion becomes pulling or pushing strongly depends on the surface geometry and on the direction of the motion

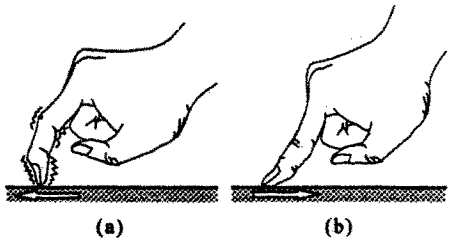


**Figure 1:** *Pulling and pushing motions by human hand.*

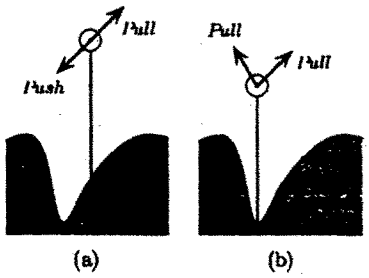


**Figure 2:** *Pulling and pushing motions by a sensor probe.*

imparted to the tactile probe. In case that the sensing motion starts from an arbitrary point on the surface, pulling and pushing motions are expected for outer and inner directions, respectively, as shown in Figure.2(a). If we can choose the local concave point as a starting one, however, we can expect pulling motions for both directions, as shown in Figure.2(b). In order to utilize this advantage, the algorithm first searches the local concave point by applying the bisection method (*local concave point search*). The tactile probe is then pulled from a local concave point to outer directions while keeping the tip of the probe in contact with the environment (*tracing motion planning*). During the tracing motion, however, the tip of the probe may get out of the surface and touch again at a new contact point due to the existence of another local concave area, or the contact point may jump from the tip to an arbitrary

point of the probe due to the existence of a local convex area. When the tactile probe recognizes such a particular situation, the sensing system stores each contact point before and after jump. For such a non-traced area, the sensing motion is repeated recursively. There might be a failure mode in which the sensing motion results in repeating mode without finding any new contact point. To emerge from such an infinite loop, we prepare *infinite loop escape* by which the tactile probe can always find a new contact point if it exists. We show that the proposed algorithm can continue to run till the tactile probe detects every surface which it can reach and touch. Also, we show some computer simulation obtained along the proposed algorithm.

## 2 Related Work

There are a number of works discussing tactile and haptic perception linked with multi-fingered dexterous hands [1]–[10]. Caselli et. al. [11] proposed an efficient technique for recognizing convex object from tactile sensing. They developed internal and external volumetric approximation of the unknown objects and exploited an effective feature selection strategy along with early pruning of incompatible objects to improve recognition performance. On the other hand, algorithms for tactile sensing have also been reported. Gaston and Lozano-Perez [12] and Grimson and Lozano-Perez [13] discussed object recognition and localization through tactile information under the assumption that the robot possesses the object models. Cole and Yap [14] have addressed "Shape from probing" problem, where they discussed how many probes are necessary and sufficient for determining the shape and position of a polygon. They showed $3n-1$ probes are necessary and $3n$ are sufficient for any $n$-gon, where $n$ is the number of probes. Most of these works [1]–[9], [11]–[14], however, deal with convex objects only and never discuss concave ones.

As far as we know, there are only a few papers [15]–[20] addressing tactile sensing for concave objects. Russell [15] designed a whisker sensor composed of an insensitive flexible beam, a potentiometer, a return spring and counterweight. Assuming that the whisker tip always makes contact with an environment, he showed the output of the sensor when scanning the concave bowl of a teaspoon. Roberts [17] discussed the strategy for determining the active sensing motion for the given set of

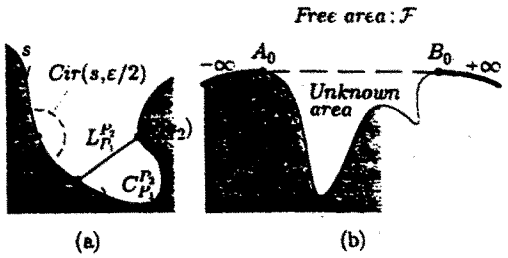


**Figure 3:** *Definition of symbols and problem notation.*

convex and concave polyhedral model objects. Chen, Rink and Zhang [18] introduced an active tactile sensing strategy to obtain local object shape, in which they showed how to find the contact frame and how to find the local surface parameters in the contact frame. This work might be applied to a concave surface that can be described by the second order polynomial equation, although their experiment is limited to a convex object only. In these works, they picked up extremely simple concave objects as test examples but included no precise discussion on the inherent sensing algorithm for concave objects. This paper is an extended version of our former work [19].

## 3 Problem Formulation

### 3.1 Preliminary definitions

Let $P(s)$ (or simply $P$) be a point on the environment's surface, where $s$ is the coordinate along the surface as shown in Figure.3(a). We define $Dist(P(s_1), P(s_2))$, $C_{P_1}^{P_2}$ and $L_{P_1}^{P_2}$ as the distance, the environment's contour, and the line segment between $P(s_1)$ and $P(s_2)$, respectively. When $C_{P_1}^{P_2}$ becomes known by a probe tracing, we define $C_{P_1}^{P_2} \in W_1$, where $W_1$ is the assembly of the area traced by the probe. If $Dist(P(s_1), P(s_2)) < \varepsilon$ exists for a $\forall$ small $\varepsilon > 0$, we can regard $L_{P_1}^{P_2}$ as an approximate surface of $C_{P_1}^{P_2}$, and define $C_{P_1}^{P_2} \in W_2$, where $W_2$ is the assembly of approximately detected area by straight-line approximation. When the tactile probe recognizes the particular area between $P(s_1)$ and $P(s_2)$ where it can not reach and touch physically, we define the area as the non-reachable area and describe by $C_{P_1}^{P_2} \in W_3$, where $W_3$ is the assembly of the non-reachable area verified by a probing motion. For every concave object (or environment), we can make an equivalent convex shape by connecting common tangential lines. The outside

of the equivalent convex is defined as the free area $\mathcal{F}$ as shown in Figure.3(b) where it is guaranteed that there is no object (or environment). We also define $Area(\mathcal{G})$ as the area of $\mathcal{G}$, where $\mathcal{G}$ denotes the region constructed by two probe postures, while the precise definition of $\mathcal{G}$ will be given later.

### 3.2 Main assumptions

Distributed sensing elements cover all over the tactile probe. The probe has negligible thickness and connected with the end-joint of a robot arm having sufficient degrees of freedom so that the probe can take arbitrary position and posture in 2D plane. The arm is assumed to have a joint torque and a joint position sensors in each joint. Both sensors are indispensable for achieving a compliant motion and determining the probe position. The torque sensor is further utilized for confirming whether a clockwise (or counter clockwise) rotation of the probe is allowed or not. The probe is long enough to ensure that the end-joint of the arm always exists in the free area $\mathcal{F}$, which enables us to neglect any geometrical constraint coming from the robot arm and to focus on the probe motion. The probe is sufficiently stiff to avoid bending. In order to avoid complicated discussion, we give the following assumption on the shape of environment. Consider a small circle whose center and radius are given by $P(s)$ and $\varepsilon/2$, respectively, as shown in Figure.3(a). We also assume that there always are only two intersection points between the circle and the environment for an arbitrary $s \in [-\infty, +\infty]$ which means that the environment is smooth enough in $\varepsilon$ level and does not include too much irregular surface. This assumption provides a valid reason for approximating the environment's contour as the straight-lined one, namely, $C_s^{s+\varepsilon} \cong L_s^{s+\varepsilon}$. We also assume that there exists only one object (or environment). The last assumption means that the objects (or environment) could be complicated but no island like object in addition to the main object.

### 3.3 Problem formulation and Nomenclature

**Problem formulation :** *Given* $C^{A_0}_{P(-\infty)} \in W$ *and* $C^{P(+\infty)}_{B_0} \in W$, *construct an algorithm such that* $C^{B_0}_{A_0} \in W$ *is achieved under the assumption in 3.2, where* $W = \cup_{i=1}^{3} W_i$.

$\mathcal{F}$ : Free area
$A_iB_i$ : Area between $A_i$ and $B_i$
$\mathcal{G}_i$ : Region constructed by two probe postures when the unknown area $A_iB_i$ is found
$L^{B_i}_{A_i}$ : Line segment between $A_i$ and $B_i$
$\mathcal{V}_i$ : Half plane that we can see in the right hand side when moving from $B_i$ to $A_i$
$\mathcal{T}_i$ : Semi-infinite region sandwiched by $P_iA_i$ and $P_iB_i$, where $P_i$ is the joint position

## 4 Pulling Motion Based Sensing

### 4.1 Outline of the algorithm

Figure.4 shows an example explaining the sensing algorithm. The probe is first inserted from an arbitrary point in the free area $\mathcal{F}$ toward the unknown area until the tip makes contact with the environment. By monitoring the torque sensor output, it is checked whether a clockwise (or counter clockwise) rotation of the probe is possible or not. After checking such a geometrical condition, the probe is rotated in the direction of rotation free till it again makes contact as shown by the dotted line in Figure.4(a), where $\varphi$ is the rotational angle. Choosing the equally divided direction $\varphi/2$, we again insert the probe till it makes contact with the environment. By repeating this procedure, the tip can finally reach the local concave point, where the probe loses any rotational degree of freedom (*local concave point search*). Then, the probe is moved from the local concave point to the outer direction, while maintaining constant torque control for the last joint, where a clockwise torque is applied during the probe motion from $D_0$ to $A_0$ and a counter clockwise torque is imparted during the probe motion from $D_0$ to $B_0$. Figure.4(b) and (c) show two examples of tracing motion. By imparting the torque depending on the direction of tracing motion, it is ensured that the probe tip makes contact with the environment if the surface is smooth enough as shown in Figure.4(b) (*tracing motion planning*). Thus, the tracing motion is executed by a pulling motion alone. This is the reason why we call the algorithm the pulling motion based sensing algorithm. If the environment includes another local concavity as shown in Figure.4(c), however, the tip will be once away from the surface and then make contact with another part of the environment due to the constant torque command imparted to the joint. When the probe recognizes such

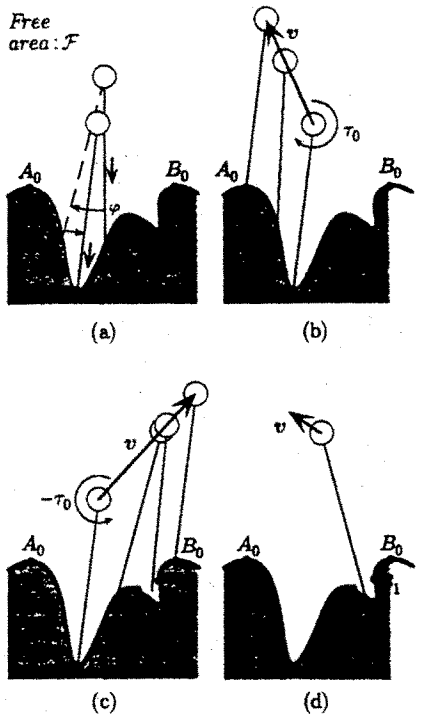


**Figure 4:** *Outline of the algorithm.*

an additional concave area, we register $A_1$ and $B_1$ as a new pair indicating the unknown area. The sensing motion is repeated recursively for $A_1B_1$. During both *local concave point search* and *tracing motion planning*, the sensing motion may result in an infinite loop depending upon the environment's geometry. In order to emerge from such an infinite loop, we prepare the *infinite loop escape*, in which the probe is inserted so that it can reach a new contact point between the designated area. The *infinite loop escape* is not always necessary but called upon request when the same contact point is detected repeatedly.

## 4.2 Local concave point search

### 4.2.1 Initial pass planning

**Definition 1** *Consider two probe postures when the unknown area $A_iB_i$ is found (see Figure.5(a)). $\mathcal{G}_i$ is defined as the region constructed by connecting $A_i$, $B_i$ and each joint of the probe. If both $A_i$ and $B_i$ are detected by one probe posture, $\mathcal{G}_i$ is generated by rotating the probe around the tip till it makes contact with the environment as shown in Figure.6(a).*

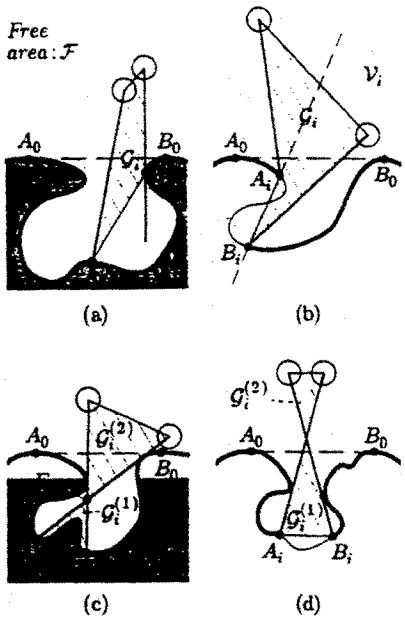


**Figure 5:** *Two probe postures when $A_iB_i$ is detected.*

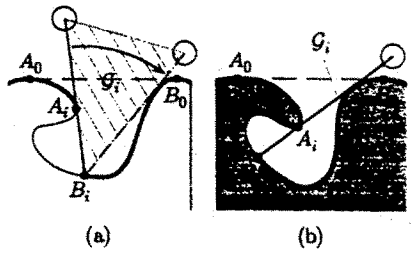


**Figure 6:** *An example of single probe detection.*

All possible shapes for $\mathcal{G}_i$ can be classified into three groups, one polygon (triangle or quadrangle), two triangles (Figure.5(c) and (d)), and one line segment (Figure.6(a)), where the difference between (c) and (d) depends upon whether $L_{A_0}^{B_0}$ passes through $\mathcal{G}_i^{(1)}$ or $\mathcal{G}_i^{(2)}$. The quadrangle in Figure.5(a) or (b) results in a triangle when the line segment $A_iB_i$ lies on the extended line of one of two probe postures. For a single probe detection as shown in Figure.6(a), $\mathcal{G}_i$ forms either a triangle or a line segment ($Area(\mathcal{G}_i) = 0$). $Area(\mathcal{G}_i) = 0$ means no more rotational degree of freedom around the tip. In other words, there exists no other pass except the current one resulting in the single probe posture as shown in Figure.6(b). Once $Area(\mathcal{G}_i) = 0$ is detected, the algorithm categorizes the $A_iB_i$ into never touching

area $W_3$ and leaves from the initial pass planning. In this subsection, we temporarily assume $Area(\mathcal{G}_i) \neq 0$. Under such an assumption, there exist only two patterns for $\mathcal{G}_i$ namely, one polygon and two triangles. Since $\mathcal{G}_i$ always links with the free area $\mathcal{F}$ under assumption in 3.2 (the probe is long enough to ensure that the joint never enters within the unknown area), it is guaranteed that if there is an environment within $\mathcal{G}_i$, its root must be connected with $A_iB_i$. If this does not hold true, the environment in $\mathcal{G}_i$ must be an island-like-object. Under the assumption of single object (or environment), however, such a situation never appears. In case of Figure.5(c), the probe trajectory should be strictly bounded, because it must be planned to pass through the common point for both triangles to absolutely avoid contacting with the environment whose root exists except $A_iB_i$. As for the environment classification, we give the following definitions.

**Definition 2** *For the area $A_iB_i$, internal and external environments are defined as follows:*
*$Int(A_iB_i)$: Internal environment is defined as the one whose root comes from the area between $A_i$ and $B_i$.*
*$Ext(A_iB_i)$: External environment is defined by the area $A_iB_i$ except $Int(A_iB_i)$.*

Let us now define the area $\mathcal{V}_i$ which also restricts the initial pass to $Int(A_iB_i)$.

**Definition 3** *Suppose an arbitrary point $Q$ denoted by the vector $q$ with respect to $\Sigma_0$. Let $a_i$ and $b_i$ be two vectors expressing the positions $A_i$ and $B_i$ with respect to $\Sigma_0$. Define $\mathcal{V}_i$ as an assemble of $q$ satisfying $V_i(q) = sgn\{(a_i - q), (b_i - q)\} > 0$, where $sgn(x, y)$ is given by*

$$sgn(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \otimes \boldsymbol{y}}{\|\boldsymbol{x} \otimes \boldsymbol{y}\|} \tag{1}$$

*where, $\otimes$ denotes a vector product for two vectors $x = (x_1, x_2)^{\mathrm{T}}$ and $y = (y_1, y_2)^{\mathrm{T}}$.*

Consider two half planes whose boundary line includes the line segment $A_iB_i$. By definition 3, $\mathcal{V}_i$ denotes the half plane that we can see in the right hand side when moving from $B_i$ to $A_i$. Note that $\mathcal{G}_i \cap \mathcal{V}_i \neq \mathcal{G}_i$ for Figure.5(b), while $\mathcal{G}_i \cap \mathcal{V}_i = \mathcal{G}_i$ for Figure.5(a), (c), (d) and Figure.6(b). Another remark is that $\mathcal{G}_i$ forming a concave quadrangle (Figure.5(b)) is converted into convex one by $\mathcal{G}_i \cap \mathcal{V}_i$, while $\mathcal{G}_i$ having two triangles (Figure.5(c)) keeps the concave shape even under $\mathcal{G}_i \cap \mathcal{V}_i$. Supposing that the probe is inserted along its longitudinal direction, we now describe a sufficient condition for making the tip reach $Int(A_iB_i)$.

**Theorem 1** *A sufficient condition for making the tip reach $Int(A_iB_i)$ is to move the probe along the line passing through one point on $L_{A_0}^{B_0} \cap \mathcal{G}_i \cap \mathcal{V}_i$ and one point on $L_{A_i}^{B_i}$, where if $\mathcal{G}_i \cap \mathcal{V}_i$ is composed of two triangles, one point on $L_{A_i}^{B_i}$ is replaced by the common point for both triangles.*

**Proof :** $L_{A_0}^{B_0} \cap \mathcal{G}_i$ expresses the line segment of $L_{A_0}^{B_0}$ within $\mathcal{G}_i$. Similarly, $L_{A_0}^{B_0} \cap \mathcal{G}_i \cap \mathcal{V}_i$ denotes the line segment of $L_{A_0}^{B_0}$ within $\mathcal{G}_i \cap \mathcal{V}_i$. Note that there are two possible shapes constructed by $\mathcal{G}_i \cap \mathcal{V}_i \cap \overline{\mathcal{F}}$, one is convex (Figure.5(a), (b), (d) and Figure.6(b)) and the other is concave (Figure.5(c)).

(i) $\mathcal{G}_i \cap \mathcal{V}_i \cap \overline{\mathcal{F}}$ is a convex polygon: Assume a half-straight line starting from an arbitrary point on $L_{A_0}^{B_0} \cap \mathcal{G}_i \cap \mathcal{V}_i$ toward $L_{A_i}^{B_i}$. Since $\mathcal{G}_i \cap \mathcal{V}_i \cap \overline{\mathcal{F}}$ is convex polygon, the half-straight line comes out only from the line segment $A_iB_i$ without passing through the other line segments. Therefore, the tip comes out from $L_{A_i}^{B_i}$ or stops due to the existence of $Int(A_iB_i)$ before reaching $L_{A_i}^{B_i}$. Once the tip comes out from $L_{A_i}^{B_i}$, the only feasible case is that the tip makes contact with $Int(A_iB_i)$. In any case, the tip finally reaches on $Int(A_iB_i)$.

(ii) $\mathcal{G}_i \cap \mathcal{V}_i \cap \overline{\mathcal{F}}$ is a concave polygon: Figure.5(c) is the only example of this case. Since the polygon is composed of two triangles having the common top angle, the half-straight line starting from an arbitrary point on $L_{A_0}^{B_0} \cap \mathcal{G}_i \cap \mathcal{V}_i$ toward the common point always reaches $Int(A_iB_i)$ without coming out from the other line segment forming the polygon. Thus, the theorem holds true. ∎

The approaching strategy based on theorem 1 guarantees to find a pass for reaching a new contact point within the designated environment's surface. In the algorithm, we simply move the probe along the line connecting two central points on $L_{A_0}^{B_0} \cap \mathcal{G}_i \cap \mathcal{V}_i$ and $L_{A_i}^{B_i}$.

#### 4.2.2 Bisection method

Once the probe detects an initial contact point, the bisection method is continuously applied for finding a local concave point, while it is not always obtained. In this subsection, after a couple of definitions, we provide

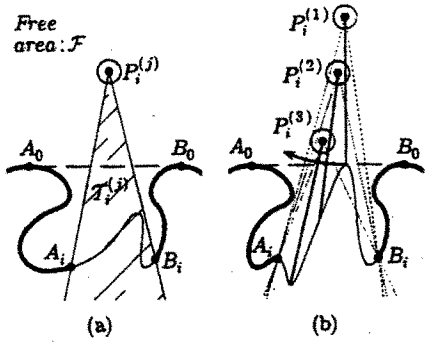


**Figure 7:** *Definition of $\mathcal{T}_i^{(j)}$.*

a theorem ensuring for always making the tip converge on $Int(A_iB_i)$. Suppose that the tip of the probe is already in contact with $Int(A_iB_i)$ by initial pass planning. Then, bisection method has the following procedure:

*(i) Swing motion: The probe is rotated around the joint until either it makes contact with an environment or it exceeds a prescribed rotational angle.*

*(ii) Dividing: The angular displacement obtained during the swing motion is equally divided. Then, the probe is swung back by the equally divided angle.*

*(iii) Inserting motion: The probe is inserted along the longitudinal direction till the tip makes contact with an environment.*

*(iv) (i) through (iii) are repeated until the probe results in one of the following states, (a) The probe loses any rotational degree of freedom around the joint, or (b) The tip converges the intersection between the environment's surface and the boundary imparted as a constraint condition.*

**Definition 4** *Let the joint position be $P_i$. Define $\mathcal{T}_i$ as a semi-infinite region sandwiched by two lines $P_iA_i$ and $P_iB_i$.*

$\mathcal{T}_i^{(j)}$ is shown by the hatched line in Figure.7(a), where the upperscript $(j)$ denotes the value after $j$-th insertion. Now let us consider an extreme case, where the initial contact is achieved at the top of the hill as shown in Figure.7(b). The bisection method starts by swinging the probe in the left direction (or right direction). Since the probe always goes into the safety area $\mathcal{F}$ in this particular case, the probe will not make contact with the environment any more. Therefore, we need

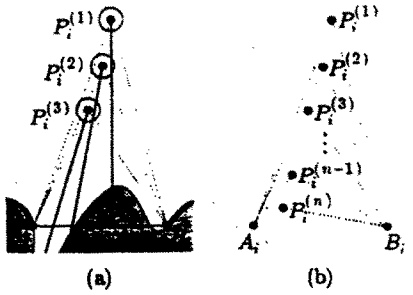


**Figure 8:** *Relationship between $P_i^{(j)}$ and $\mathcal{T}_i^{(j)} \cap \mathcal{V}_i$.*

a boundary, such that we can stop the swing motion. $\mathcal{T}_i^{(j)}$ provides a reasonable boundary, although it is still not a sufficient boundary for finally making the tip converge on $Int(A_iB_i)$.

**Theorem 2** *Suppose that an initial contact with $Int(A_iB_i)$ is already completed. Also suppose that the maximum swing angle is determined such that the probe never comes out from the line segments $P_i^{(j)}A_i$ and $P_i^{(j)}B_i$. If there is no $Ext(A_iB_i)$ within $\mathcal{T}_i^{(1)} \cap \mathcal{V}_i$, the tip always reaches $Int(A_iB_i)$ through the bisection method.*
m

**Proof :** By the assumption imparted to the maximum swing angle, the probe never comes out from the line segments $P_i^{(j)}A_i$ and $P_i^{(j)}B_i$ as shown in Figure.8(a). From Figure.8(b), we can easily show $(\mathcal{T}_i^{(1)} \cap \mathcal{V}_i) \supset (\mathcal{T}_i^{(2)} \cap \mathcal{V}_i) \supset \cdots \supset (\mathcal{T}_i^{(n)} \cap \mathcal{V}_i)$, which means if there is no $Ext(A_iB_i)$ within $\mathcal{T}_i^{(1)} \cap \mathcal{V}_i$, $\mathcal{T}_i^{(j)} \cap \mathcal{V}_i (j = 1, 2, \ldots, n)$ cannot include $Ext(A_iB_i)$ either. Therefore, the tip comes out from $L_{A_i}^{B_i}$ or stops due to the existence of $Int(A_iB_i)$ before reaching $L_{A_i}^{B_i}$. Once the tip comes out from $L_{A_i}^{B_i}$, the only feasible case is that it makes contact with $Int(A_iB_i)$. In any case, the tip finally reaches on $Int(A_iB_i)$. ∎

In order to utilize theorem 2, we have to ensure that there is no $Ext(A_iB_i)$ within $\mathcal{T}_i^{(1)} \cap \mathcal{V}_i$. In case that there exists an external environment within $\mathcal{T}_i^{(1)} \cap \mathcal{V}_i$, however, the tip does not always reach $Int(A_iB_i)$ even though the bisection method is executed within $\mathcal{T}_i^{(j)} \cap \mathcal{V}_i$. Figure.9(a) may provide a good example. Suppose that the initial contact is achieved at the top of the hill and the bisection method is executed as shown in Figure.9(a). At the end of the bisection method, the tip will converge to the local concave point $D_i$ be-

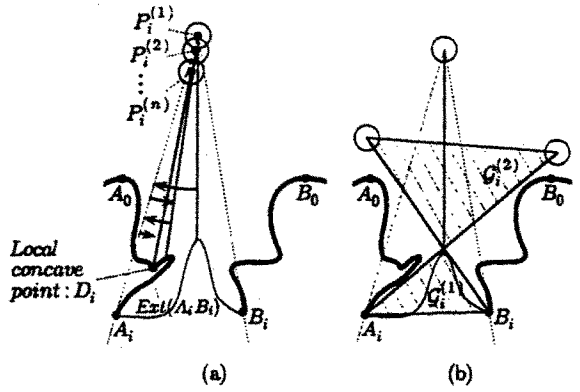


**Figure 9:** *An example of environment where there exists $Ext(A_iB_i)$ within $\mathcal{T}_i^{(1)} \cap \mathcal{V}_i$.*

longing to $Ext(A_iB_i)$. To exclude the convergence to such an external environment, we again recall $\mathcal{G}_i$ where it is guaranteed that there is no $Ext(A_iB_i)$ in it. For example, if the bisection method is planned, such that the tip may move within $\mathcal{T}_i^{(j)} \cap \mathcal{V}_i \cap \mathcal{G}_i$ as shown in Figure.9(b), the convergence on $Int(A_iB_i)$ is always guaranteed. Theorem 3 provides a sufficient condition for making the tip always converge on $Int(A_iB_i)$.

**Theorem 3** *Suppose that an initial contact with $Int(A_iB_i)$ is already completed. A sufficient condition for making the tip finally converge on $Int(A_iB_i)$ is to execute bisection method such that the tip may not come out from the boundary of $\mathcal{T}_i^{(j)} \cap \mathcal{V}_i \cap \mathcal{G}_i$ except the line segment $A_iB_i$.*

**Proof :** Since there is no $Ext(A_iB_i)$ within $\mathcal{T}_i^{(j)} \cap \mathcal{V}_i \cap \mathcal{G}_i$, the tip comes out from $L_{A_i}^{B_i}$ or stops due to the existence of $Int(A_iB_i)$ before reaching $L_{A_i}^{B_i}$. Therefore, by the similar logic that explained in the proof in theorem 2, we can show that the tip finally reaches on $Int(A_iB_i)$ in any case. ∎

## 4.3 Tracing motion planning

### 4.3.1 Realization of the pulling motion

When the *local concave search* comes to the end, there are basically two possible cases as shown in Figure.10, namely, the tip finally reaches a local concave point (a), and it stops at a non-local concave point ((b) and (c)). In Figure.10(b) and (c), the tip stops due to the boundary constraint (b) and due to multiple contacts (c), respectively.

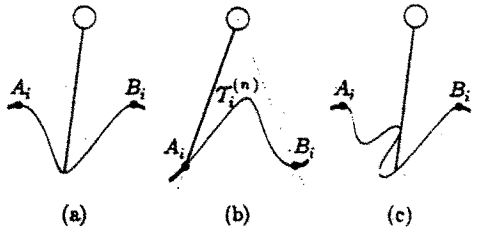


**Figure 10:** *Final states after bisection method: (a) Local concave point; (b) Intersection between the environment's surface and $\mathcal{T}_i^{(n)}$; (c) Multiple contacts.*

**Definition 5** *$Y_1$ denotes the final contact mode where $Y_1 = 1$ means that only the tip makes contact with the environment, and $Y_1 = 2$ means multiple contacts at the tip and other points. $Y_2$ denotes the rotational constraint around the joint, where $Y_2 = 0$, 1 and $-1$ are full constraint (Figure.10(a)(c)), single constraint for the clockwise direction, and single constraint for the counter clockwise direction (Figure.10(b)), respectively.*

**Definition 6** *Define $\mathcal{K}_1$ as an assemble of $v$ satisfying $v^{\mathrm{T}}t^* < 0$, where $v$ and $t$ are vectors expressing the moving direction of the joint, and the longitudinal direction of the probe, respectively, and $*$ denotes the value just after the bisection method is completed. Also, define $\mathcal{K}_2$ as an assemble of $v$ satisfying $sgn(v, t^*) < 0$.*

More simply, $\mathcal{K}_1$ denotes the half plane excluding $t^*$ when the whole plane is divided at the joint by the line perpendicular to $t^*$.

**Definition 7** *$Face(right) = ON$ (or $Face(left) = ON$ or $Face(tip) = ON$) means that a part of right side (or left side or tip) of the probe makes contact with an environment.*

Based on the definition 5, the three cases (a), (b), and (c) in Figure.10 can be classified by $(Y_1, Y_2) = (1, 0)$, $(Y_1, Y_2) = (1, -1 \text{ or } 1)$, $(Y_1, Y_2) = (2, 0)$, respectively. For each case, let us now consider a sufficient condition for realizing the pulling motion based tracing motion. Let $\mathcal{K}$ be the region where the joint can be moved without generating any pushing motion. In case (a), $\mathcal{K} = \mathcal{K}_1$ (the hatched area in Figure.11(a)). In case (b), if $Y_2 = -1$, $\mathcal{K} = \mathcal{K}_1 \cap \mathcal{K}_2$ (the double hatched area in Figure.11(b)) and if $Y_2 = 1$, $\mathcal{K} = \mathcal{K}_1 \cap \overline{\mathcal{K}_2}$. In case (c), if $Face(left) = ON$, $\mathcal{K} = \mathcal{K}_1 \cap \mathcal{K}_2$ (the double hatched area in Figure.11(c)) and if $Face(right) = ON$, $Y_2 = 1$, $\mathcal{K} = \mathcal{K}_1 \cap \overline{\mathcal{K}_2}$.

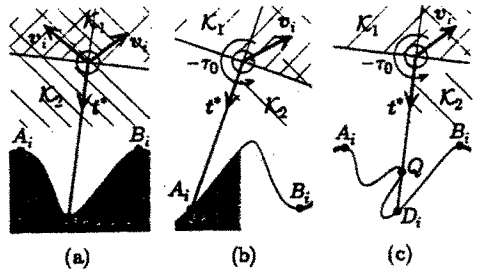


**Figure 11:** *Tracing motion planning.*

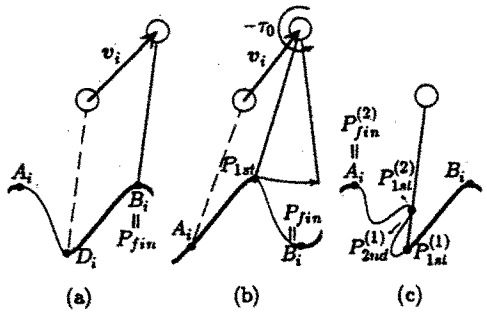


**Figure 12:** *Realization of tracing motion.*

**Theorem 4** *A sufficient condition for achieving a pulling motion based tracing is to determine the moving direction of the joint and the joint torque, such that* $v_i \in \mathcal{K}$ *and*

$$\tau = \tau_0 sgn(v_i, t^*) \tag{2}$$

*where* $\tau_0 (> 0)$ *is the reference torque and the positive direction of* $\tau$ *is chosen in the clockwise direction, and* $\mathcal{K}$ *is given below.*

(A) : *If* $(Y_1, Y_2) = (1, 0)$, $\mathcal{K} = \mathcal{K}_1$.

(B) : *If* $(Y_1, Y_2) = (1, -1)$, $\mathcal{K} = \mathcal{K}_1 \cap \mathcal{K}_2$.

(C) : *If* $(Y_1, Y_2) = (1, 1)$, $\mathcal{K} = \mathcal{K}_1 \cap \overline{\mathcal{K}_2}$.

(D) : *If* $(Y_1, Y_2) = (2, 0)$ *and* $Face(left) = ON$, $\mathcal{K} = \mathcal{K}_1 \cap \mathcal{K}_2$.

(E) : *If* $(Y_1, Y_2) = (2, 0)$ *and* $Face(right) = ON$, $\mathcal{K} = \mathcal{K}_1 \cap \overline{\mathcal{K}_2}$.

**Proof :** (Omitted) ∎

In case (A), when $v_i$ satisfying $\mathcal{K} = \mathcal{K}_1$ is chosen, only the right or the left tracing motion will be achieved. For completing the tracing motion for both directions, we further separate $\mathcal{K} = \mathcal{K}_1$ into $\mathcal{K} = \mathcal{K}_1 \cap \mathcal{K}_2$ and $\mathcal{K} = \mathcal{K}_1 \cap \overline{\mathcal{K}_2}$. Then, we execute the right and the left tracing motions by choosing $v_i$ satisfying $v_i \in \mathcal{K}_1 \cap \mathcal{K}_2$ and $v_i \in \mathcal{K}_1 \cap \overline{\mathcal{K}_2}$, respectively.

#### 4.3.2 All possible cases during tracing motion

Tracing motion is continued until the tip successfully traces from the initial point $D_i$ to $A_i$ (or $B_i$) as shown in Figure.12(a). However, the contact between the probe and the environment is not always guaranteed during the tracing motion. For example, the tip may be away from the environment at the top of the hill during the tracing motion from $A_i$ to $B_i$, as shown in Figure.12(b). Once the probe is away from the surface, the joint torque sensor can not detect any reaction torque from the environment, by which the sensing system can recognize the case. Now let us classify what sort of cases appear during a tracing motion and consider how to deal with each case.

**Definition 8** *Let* $P_{fin}$ *be the destination point for a tracing motion. We choose* $P_{fin} = A_i$ *for* $sgn(v_i, t^*) > 0$ *or* $P_{fin} = B_i$ *for* $sgn(v_i, t^*) < 0$.

We stop the tracing motion when a part of the probe reaches $P_{fin}$. During the tracing motion, however, there might appear a contact point jump due to the surface geometry. For such a contact point jump, we define two points $P_{1st}$ and $P_{2nd}$ as follows.

**Definition 9** *Let* $P_{1st}$ *and* $P_{2nd}$ *be the contact points just before and after a contact point jump, respectively. In case of a single probe detection, the point closer to the tip is chosen as* $P_{1st}$.

All possible cases during a tracing motion can be classified as follows:

⟨Case 1⟩ All contact points are continuously detected by the probe tip for the designated area.

⟨Case 2⟩ At least, one contact point jump appears.

⟨Case 3⟩ $v_i{}^T t > 0$ is detected.

⟨Case 4⟩ The joint loses the moving degree of freedom in the direction given by $v_i$.

All cases can be detected by utilizing the sensors implemented in the robot-probe system. For example, ⟨Case 1⟩ and ⟨Case 2⟩ are confirmed by the probe output. ⟨Case 3⟩ is certified by the sensors mounted in each joint of the robot arm. ⟨Case 4⟩ can be confirmed by the torque sensor. Since ⟨Case 1⟩ means that the designated unknown area $Int(A_iB_i)$ becomes known, we can simply categorize such an area into $W$. For one of the three other cases, however, we have to memorize a part of $Int(A_iB_i)$ as a further non-detected area.

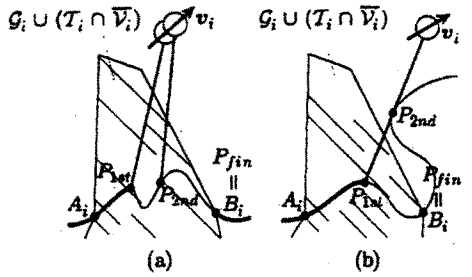


**Figure 13:** *$P_{2nd}$ detected on $Int(A_iB_i)$ or $Ext(A_iB_i)$.*

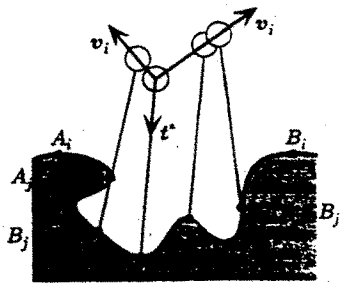


**Figure 14:** *How to store non-detected area.*

Let us now consider which points we should register for ⟨Case 2⟩ , ⟨Case 3⟩ and ⟨Case 4⟩ . Just for our convenience, ⟨Case 2⟩ is further classified into the following three cases.

⟨Case 2-1⟩ $P_{2nd}$ exists on $Int(A_iB_i)$.
⟨Case 2-2⟩ $P_{2nd}$ exists on $Ext(A_iB_i)$.
⟨Case 2-3⟩ $P_{2nd}$ does not exist within $\overline{\mathcal{F}}$.

⟨Case 2-1⟩ or ⟨Case 2-2⟩ is distinguished by checking whether $P_{2nd} \in \mathcal{G}_i \cup (T_i \cap \overline{\mathcal{V}_i})$ or $P_{2nd} \in \overline{\mathcal{G}_i} \cap (\overline{T_i} \cup \mathcal{V}_i)$, respectively, as shown in Figure.13(a) and (b). ⟨Case 2-3⟩ happens when the probe enter $\mathcal{F}$ without any contact. In ⟨Case 2-1⟩ , the area between $P_{1st}$ and $P_{2nd}$ is registered as a non-detected area and then the tracing motion is continued for the remaining non-traced area. In both ⟨Case 2-2⟩ and ⟨Case 2-3⟩ , we stop any further tracing motion and register the area between $P_{1st}$ and $P_{fin}$ as a non-detected area. ⟨Case 3⟩ may occur because the condition realizing a pulling motion is satisfied only at the starting position. Once ${v_i}^{\mathrm{T}}t > 0$ is detected, we stop the tracing motion and register the area between $P_{1st}$ and $P_{fin}$ as a non-detected one, where $P_{1st}$ is replaced by the point in which ${v_i}^{\mathrm{T}}t > 0$ is detected. ⟨Case 4⟩ may also occur depending on the surface geometry, the probe posture and the moving direction of the probe. Once the probe loses the degree of freedom in the direction $v_i$, we stop the tracing motion and register the area between $P_{1st}$ and $P_{fin}$ as a non-detected one, where $P_{1st}$ is replaced by the point in which the probe loses the degree of freedom. In case of Figure.11(c), according to the theorem 4, the tracing motion is not executed in the direction from $D_i$ to $A_i$. In such a situation, we register $(P^{(1)}_{1st}, P^{(1)}_{2nd})$ and $(P^{(2)}_{1st}, P^{(2)}_{fin})$ as non-detected areas after regarding $(P^{(1)}_{1st}, P^{(1)}_{2nd}) = (D_i, Q)$ and $(P^{(2)}_{1st}, P^{(2)}_{fin}) = (Q, A_i)$ as shown in Figure.12(c).

#### 4.3.3 Registering into $A_jB_j$

The non-detected area, such as $(P_{1st}, P_{2nd})$ has to be finally stored into $(A_j, B_j)$ for executing the program recursively. In this subsection, we briefly describe the basic rule to obtain the set of $(A_j, B_j)$.

⟨Rule to determine $(A_j, B_j)$⟩ (see Figure.14)
(i) In case of $P_{fin} = A_i$: $(A_j, B_j) = (P_{2nd} \text{or} P_{fin}, P_{1st})$.
(ii) In case of $P_{fin} = B_i$: $(A_j, B_j) = (P_{1st}, P_{2nd} \text{ or } P_{fin})$.

This rule is for keeping the characteristics of $\mathcal{V}_i$ defined in 4.2.1.

### 4.4 Infinite loop escape

There might be a particular state in which the tip can find the same point repeatedly during the *local concave point search* and the *tracing motion planning*. In order to avoid such an undesirable mode, we prepare the *infinite loop escape*, where the probe temporarily searches a new contact point by utilizing the same way taken in the initial pass planning. Since the initial pass planning ensures to find a new contact point between $A_iB_i$, we can separate the area $A_iB_i$ into two new areas $A_{i+1}B_{i+1}$ and $A_{i+2}B_{i+2}$, as shown in Figure.15(a). After dividing the area, we leave from the *infinite loop escape* and come back to the normal mode given by *local concave point search* and *tracing motion planning*. Now, assume that an infinite mode appears every time after initial pass planning motion. In such an extreme case, $A_{i+1}B_{i+1}$ (or $A_{i+2}B_{i+2}$) is further separated into two unknown areas $A_{i+3}B_{i+3}$ and $A_{i+4}B_{i+4}$, and so forth, as shown in Figure.15(a). For every newly divided area, if $Dist(A_j, B_j) < \varepsilon$ is satisfied, any further separation between $A_j$ and $B_j$ is stopped and $C^{B_j}_{A_j} \in W_2$ is assigned. This implies that the algorithm brings the environment's shape in relief even when an

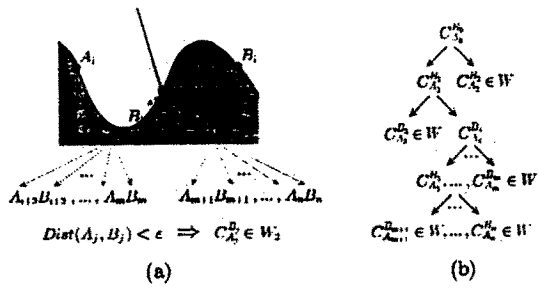


**Figure 15:** *Infinite loop escape.*

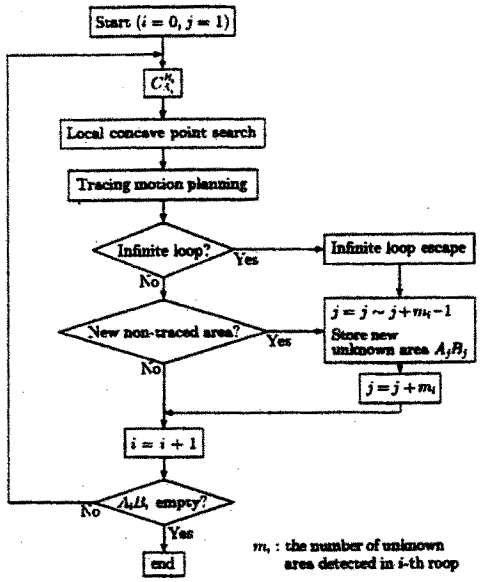


**Figure 16:** *Flowchart.*

infinite loop occurs continuously.Figure.16 shows the overall flowchart of the algorithm.

## 5 Simulation

Figure.17 and Figure.18 show simulation results, where the continuous and the dotted lines denote the known and the unknown areas, respectively, and the line segment passing through the joint expresses the moving direction of the joint, which is determined by the sufficient condition given by theorem 4. Figure.17 is a simple example, where the tip can trace every surface eventually. Also, there is no $Ext(A_iB_i)$ within $T_i^{(1)} \cap \mathcal{V}_i$ and, therefore, the local concave point can be found by utilizing either theorem 2 or theorem 3. Figure.18 shows an example including a couple of never touching

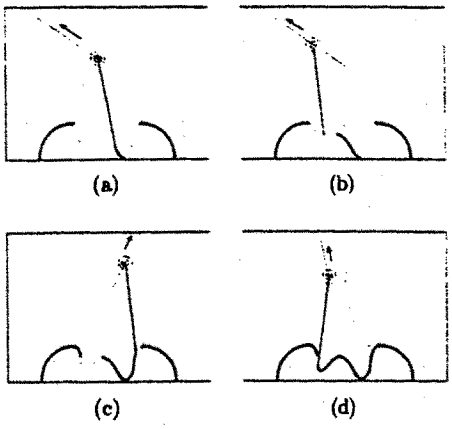


**Figure 17:** *Simulation result (example 1).*

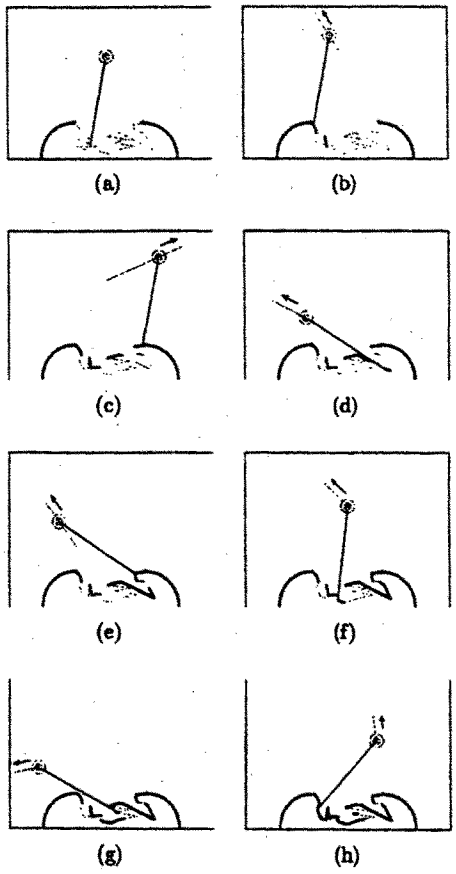


**Figure 18:** *Simulation result (example 2).*

areas, where the tip can not reach and touch. Although this example includes such a complicated surface, it can be seen that the proposed algorithm enables the tip to

reach every reachable area step by step and, finally, it finds the contour of the surface except three never touching areas. From these simulation results, it can be understood that the unknown area gradually decreases and finally disappears except the particular area which the probe cannot reach and touch.

## 6 Conclusions

A tactile sensing algorithm for detecting the concave surface was discussed. In our next step, we will relax this assumption so that the tip can reach much larger area. Also, we will try to extend the pulling-motion-based tactile sensing into a 3D concave environment. Since tracing every area for a 3D environment is not as feasible as that in a 2D one, the parametrized local surface function [18] may be a useful tool for expressing the surface shape with respect to a local contact frame. We believe, however, that the concept of the pulling-motion-based tracing should be still included even in the algorithm for surface sensing of a 3D environment.

This work has been supported by CREST of JST(Japan Science and Technology).

## References

[1] Dario, P. and G. Buttazzo: An anthropomorphic robot finger for investigating artificial tactile perception, Int J. Robotics Research, vol.6, no.3, pp25–48, 1987.

[2] Fearing, R. S. and T. O. Binford: Using a cylindrical tactile sensor for determining curvature, Proc. of the IEEE Int. Conf. on Robotics and Automation, Philadelphia, pp765–771, 1988.

[3] Maekawa, H., K. Tanie, K. Komoriya, M. Kaneko, C. Horiguchi, and T. Sugawara: Development of a finger-shaped tactile sensor and its evaluation by active touch, Proc. of the IEEE Int. Conf. on Robotics and Automation, Nice, p1327, 1992.

[4] Salisbury, J. K.: Interpretation of contact geometries from force measurements, Proc. of the 1st Int. Symp. on Robotics Research, 1984.

[5] Brock, D.L. and S.Chiu: Environment perception of an articulated robot hand using contact sensors, Proc. of the IEEE Int. Conf. on Robotics and Automation, Raleigh, pp89–96, 1987.

[6] Kaneko, M., and K. Honkawa: Compliant motion based active sensing by robotic fingers, Preprints of the 4th IFAC Symp. on Robot Control, Capri, pp137–142, 1994.

[7] Allen, P. and K. S. Roberts: Haptic object recognition using a multi-fingered dexterous hand, Proc. of the IEEE Int. Conf. on Robotics and Automation, pp342–347, 1989.

[8] Bajcsy, R.: "What can we learn from one finger experiments?", Proc. of the 1st Int. Symp on Robotics Research, pp509–527, 1984.

[9] Bicchi, A., Salisbury J. K. and D. J. Brock: Contact sensing from force measurements, Int. J. of Robotics Research, vol.12, no.3, 1993

[10] Bays, J. S. : Tactile shape sensing via single- and multi-fingered hands, Proc. of the IEEE Int. Conf. on Robotics and Automation, PP 290–295, 1989.

[11] Caselli, S., C. Magnanini, F. Zanichelli, and E. Caraffi: Efficient exploration and recognition of convex objects based on haptic perception, Proc. of the IEEE Int. Conf. on Robotics and Automation, PP 3508–3513, 1996.

[12] Gaston, P. C., and T. Lozano-Perez: Tactile recognition and localization using object models, The case of polyhedra on a plane, MIT Artificial Intelligence Lab. Memo, no.705, 1983.

[13] Grimson, W. E. L. and T. Lozano-Perez: Model based recognition and localization from sparse three dimensional sensory data, AI Memo 738, MIT, AI Laboratory, Cambridge, MA, 1983.

[14] Cole, R., and C. K. Yap: Shape from probing, J. of Algorithms 8, pp19–38, 1987.

[15] Russell, R. A. : Using tactile whiskers to measure surface contours, Proc. of the 1992 IEEE Int. Conf. on Robotics and Automation, pp1295–1300, 1992.

[16] Tsujimura, T. and T. Yabuta: Object detection by tactile sensing method employing force/moment information, IEEE Trans. on Robotics and Automation, vol.5, no.4, pp444–450, 1988.

[17] Roberts, K. S. : Robot active touch exploration, Proc. of the IEEE Int. Conf. on Robotics and Automation, PP 980–985, 1990.

[18] Chen, N., R. Rink, and H. Zhang: Local object shape from tactile sensing, Proc. of the IEEE Int. Conf. on Robotics and Automation, PP 3496–3501, 1996.

[19] Kaneko, M., M. Higashimori, and T, Tsuji: Pulling motion based tactile sensing for concave surface, Proc. of the IEEE Int. Conf. on Robotics and Automation, pp2477–2484, 1997.

[20] Boissonnat, J. D. and M. Yvinec: Probing a scene of non-convex polyhedra, Algorithmica, vol.8, pp321–342, 1992.