Localization on Curved Objects Using Tactile Information

Yan-Bin Jia

Department of Computer Science
Iowa State University
Ames, IA 50011-1040

Oct 30, 2001
The Problem

A flat object of known shape on the table

- free to slide
- curved boundary
- unknown orientation
- approximate position

How to grasp the object?
Touch and Feel It!

Use two fingers with

- ability to translate and rotate
- *tactile* capability — detecting contact
- *enough* friction with the object boundary
Finger Localization

While rolling, the two fingers simultaneously record contacts at multiple time instants.

Estimate finger contacts on the object from tactile data.

*The problem becomes geometric under rolling.*
Related Work

• Kinematics of Contact

   Cai & Roth ’87; Montana ’88; Li & Canny ’90

• Tactile Sensing

   Salisbury ’84; Fearing & Binford ’88; Howe & Cutkosky ’92

• Tactile Localization, Recognition & Reconstruction

   Grimson & Lozano-Pérez ’84; Allen & Roberts ’89; Fearing ’90
   Chen et al. ’96; Keren et al. ’00; Jia ’00; Moll & Erdmann ’01

• Multi-fingered Hands & Haptics

   Kerr & Roth ’86; Hollerbach ’87; Cole et al. ’89
   Roberts ’90; Okamura et al. 97; Pai & Reissel ’97; Bicchi ’00

• Grasping & Syntheses

   Mishra et al. ’87; Brost ’88; Nguyen ’89; Goldberg ’90
   Markenscoff et al. ’90; Hong et al. ’90; Chen & Burdick ’92
   Bicchi & Kumar ’00; Ponce et al. ’93; Blake & Taylor ’93
   Rimon & Blake ’96

• Curve Processing

   Goodman ’91; Manocha & Canny ’92; Sakai ’99
Rolling on a Fixed Object

$L$ and $\Theta$ computable from tactile data & finger rotation.

**Finger localization** — find all boundary segments $(a, b)$ with length $L$ and total curvature $\Theta$.

\[
\ell(a, b) = \int_a^b \left\| \beta'(s) \right\| ds = L
\]

\[
\Phi(a, b) = \int_a^b \kappa(s) \left\| \beta'(s) \right\| ds = \Theta
\]

where $\kappa$ is curvature.
The Marching Algorithm

- Most curves are not unit-speed, i.e., $\|\beta'\| \neq 1$.
- Arc length $\ell(a, b)$ often has no closed form.
- Total curvature $\Phi(a, b)$ has closed form only when...

Algorithm  March counterclockwise on the boundary.
- numerically complete — finds all segments.
Convex Boundary

Compute $t_0$ such that $\ell(s_0, t_0) = L$.

**Case 1:** $\Phi(s_0, t_0) < \Theta$

**Case 2:** $\Phi(s_0, t_0) > \Theta$

**Claim** $\lim_{i \to \infty} (s_i, t_i) = (a, b)$ in linear rate.
Non-Convex Boundary

Precompute

- all *inflection points* $z_1, \ldots, z_n$ where $\kappa(z_i) = 0$.
- arc lengths $\ell(z_i, z_{i+1})$, total curvatures $\Theta(z_i, z_{i+1})$.

March $s$ and $t$ while distance $L$. Transitions between

1. Convex-convex mode ($\kappa(s) \geq 0$ and $\kappa(t) \geq 0$)
2. Concave-concave mode
3. Convex-concave mode
4. Concave-convex mode

**Running time:** $\Theta(T/h)$, where $T$ is the curve domain size and $h$ the step size.
An Example
Critical Points of Total Curvature

The marching algorithm can be modified to find all curve segments \((s, t)\) of length \(L\) and satisfying

\[
\int_s^t f(u) \, du = C
\]

Particularly, segments \((s, t)\) of length \(L\) such that

\[
\kappa(t) - \kappa(s) = \int_s^t \kappa'(u) \, du = 0.
\]

with these changes in the marching algorithm:

\[
\Phi(s, t) \longrightarrow \kappa(t) - \kappa(s) \\
\kappa \longrightarrow \kappa' \\
\text{inflections} \longrightarrow \text{vertices} (\kappa' = 0)
\]

Such \(s\) is a critical point of total curvature \(\Phi(s, t(s))\) over a segment starting at \(s\) and of arc length \(L\).
Example of Critical Segments
Two Fingers on a Free Object

In the rolling mode, comparison of contacts on the two fingers eliminates the unknown object rotation.

\[ \Phi(s_0, s_1) - \Phi(t_0, t_1) = \theta_1, \]
\[ \Phi(s_0, s_2) - \Phi(t_0, t_2) = \theta_2. \]
Arc lengths $L_1, L_2, L_3, L_4$ computable from tactile data.

Geometrically, find two points $s_0$ and $t_0$ from which

- the two segments of lengths $L_1$ and $L_3$, resp., differ by $\theta_1$ in total curvature.
- the two segments of lengths $L_2$ and $L_4$, resp., differ by $\theta_2$ in total curvature.
Domain Partitioning

\( \phi_i(s) \): the total curvature of a segment starting at \( s \) and having length \( L_i \).

Critical points of \( \phi_1 \) and \( \phi_2 \) divide the domain into intervals \([a_0, a_1], \ldots, [a_{n-1}, a_n]\).

Critical points of \( \phi_3 \) and \( \phi_4 \) divide the domain into intervals \([b_0, b_1], \ldots, [b_{m-1}, b_m]\).

Over each \([a_i, a_{i+1}]\), \( \phi_1 \) and \( \phi_2 \) are monotone.
Over each \([b_j, b_{j+1}]\), \( \phi_3 \) and \( \phi_4 \) are monotone.
A One-to-one Correspondence

\[ [y, x] \rightarrow [w, z] \]

\( s \mapsto t \) \quad \text{such that} \quad \phi_1(s) - \phi_3(t) = \theta_1
Alternative March

\[
\phi_4(w_0) + \theta_2 > \phi_2(x_0) \quad \rightarrow \quad \phi_4(w_1) + \theta_2 = \phi_2(x_0) \\
\quad \rightarrow \quad \phi_3(w_1) + \theta_1 > \phi_1(x_0) \\
\quad \rightarrow \quad \phi_1(x_1) = \phi_3(w_1) + \theta_1 \\
\quad \rightarrow \quad \phi_4(w_1) + \theta_2 > \phi_2(x_1) \\
\quad \rightarrow \quad \ldots
\]
An Example

real finger locations
Remarks

- Finger localization based on contact data.

- Total curvature on the object boundary “felt” by the fingers plays a key role.

- A scheme for curve computation
  - Dissecting domain into intervals over which the function is monotone.
  - Marching combined with numerical bisection.
  - Linear on the order of the size of discretization.
  - Numerically complete.
Antipodal Points for Grasping

Extending the curve processing paradigm — computation of all antipodal points on a closed curve: