Multiple Robot Boundary Tracking

Victor A. Mejia
Faculty Advisor: Todd Wittman
Math 199 Independent Study
December 2007

Multiple Robot Boundary Tracking

- We want to track boundary of a region in an image
- Algorithm proposed by graduate student Zhong Hu
- inspired by the robotic vehicle tracking algorithm developed by (Kemp-Bertozzi-Marthaler 2004).
- We want to (1) track multiple shapes, (2) more robustness to noise, (3) allow for parallel computing
- Want to decrease running time and automate starting points for “robot trackers”
- We propose a multiple-robot boundary tracking algorithm
### Stopping Criterion

- A robot stops when it crosses the path of any other robot.
- We stop when it is within $R$ pixels of the other robot’s path.
- $R$ is the maximum possible radius of circle circumscribing the polygon that robot$_1$ is sweeping out.
- Either $d_1$ or $d_2$ must be less than $R$.

---

### Stopping Criterion

- We need an upper bound on $R$.
- Max trajectory range at $\theta_0 = 45^\circ$.
- Run algorithm starting at $(0, 0)$.
- Stop when $\theta = -45^\circ$.
- $R = D/2$.
- We use $R$ to detect when robots have crossed paths.
Cooperative Boundary Tracking Algorithm

**Input:** tracking velocity $v$, angular velocity $\omega$, number of robots $n$

1. Automate $n$ initial starting points for robots
2. Determine value of $d$ using CUSUM filters ($d = 1$ or $d = -1$)
3. $\theta = 0$;
4. Calculate $R$ based on $v$ and $\omega$

while robots have not intersected paths

(a) $\theta_i = \theta_i + d_i \cdot \omega$
(b) $x_i = x_i + v_i \cdot \cos(\theta_i)$, $y_i = y_i + v_i \cdot \sin(\theta_i)$
(c) increase $v$ if we have made a full circle
(d) determine value of $d$ using CUSUM filters
(e) if we have crossed the boundary
   store boundary point and set $d = -d$
   update $\theta$ using angle correction (Jin-Bertozzi 2007)
(f) if robot intersects boundary path of any other robot (including itself),
   then stop robot

Demonstration of Features

Without angle correction

With angle correction and increase of $v$
Multiple Robot Tracking (user-defined points)

11 robots, $v = 2$

7 robots, $v = 2$
Gaussian noise variance = 0.05
Multiple Robot Tracking (user-defined points)

11 robots, Gaussian noise variance 0.05

How do we automate starting points?

- Randomly sample the image until you find $n$ starting points that are inside the segment you want to track
Automated Starting Points

5 robots

7 robots

Automated Starting Points

7 robots, Gaussian noise variance = 0.05
Automated Starting Points

Possible Enhancements

- Better automation of choosing the starting points
- Communication between robots
- Extend to real data