Pusher-watcher: An approach to fault-tolerant tightly-coupled robot coordination

Brian P. Gerkey
Maja J Matarić

USC Robotics Research Lab

http://robotics.usc.edu

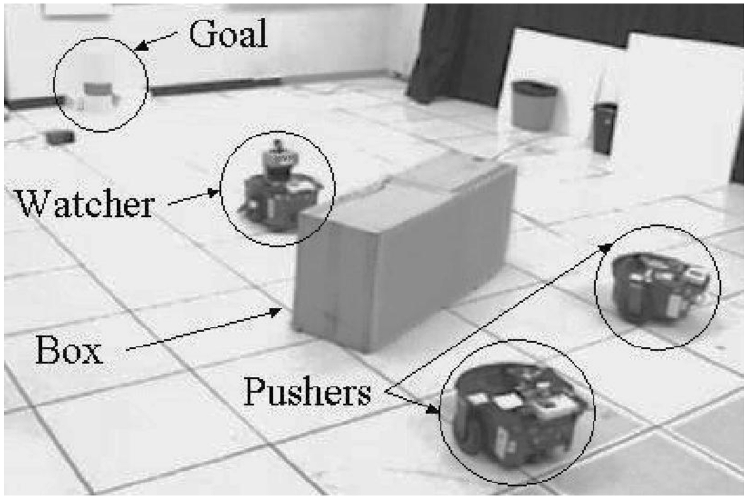
ICRA 2002, Washington, DC

Introduction

- How can groups of heterogeneous robots coordinate their behavior so as to execute tightly-coupled tasks?
- We propose a partial answer, in the form of the task allocation system MURDOCH.
- Using MURDOCH, we have built `pusher-watcher`, a distributed control system for multi-robot cooperative box-pushing.
- We have validated `pusher-watcher` with a team of physical robots.

Cooperative box-pushing

- The task is to move a large box from an initial to a goal location.
- We assume:
  - The pushing robots cannot perceive the goal due to occlusion by the box.
  - The box can be sensed by the robots that are to push it and the goal can be sensed by the robot that is to act as watcher.
  - The watcher robot can sense the position and orientation of the box relative to its own pose.
  - There is an obstacle-free path between the box's initial location and the goal that is wide enough for the box and robots to pass.



pusher-watcher algorithm

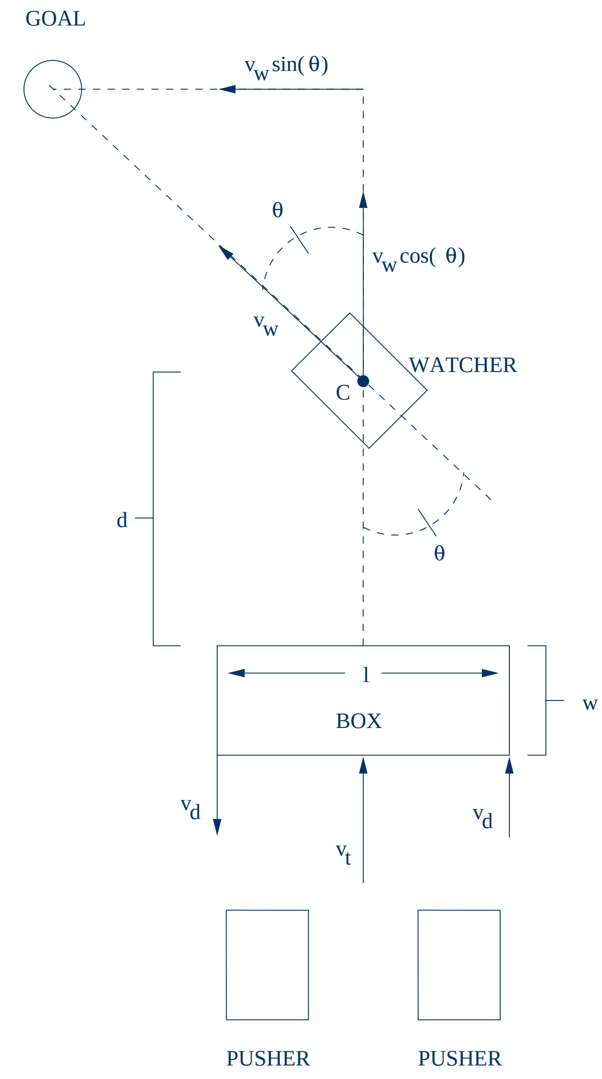
- Two robots acts as `pushers`, a third as `watcher`.
- The goal is to rotate the box orthogonal to the path to the goal, while translating it toward the goal.
- `pusher` velocities are determined analytically:

$$v_t = v_w \cos\theta$$

$$(d+w)\dot{\theta} = v_w \sin\theta$$

$$v_d = \frac{l}{2}\frac{v_w}{(d+w)}\sin\theta$$

$$v_p = v_t \pm v_d$$



MURDOCH: auctioning tasks

- We are interested in dynamic environments in which robots may come and go at any time.
- We treat the entire collective as an anonymous pool of resources that can be applied to tasks that we want accomplished.
- We solve the resultant resource allocation problem through the use of a variant of the Contract Net Protocol (R. G. Smith 1980).
- Essentially, when a task is to be assigned, the task is put up for *auction*, and capable robots *bid* for it by stating their current fitness.

Auction details

- To support anonymity, robots communicate via a broadcast-based *publish/subscribe* messaging system.
- When a task is available a robot, acting as auctioneer, sends out a descriptive *task announcement*.
- Each capable (and available) robot then estimates its fitness for the task using a *metric* and responds with a *bid*.
- The auctioneer closes the auction and awards the winner a *time-limited* contract to execute the task.
- The auctioneer subsequently monitors task progress.

Tasks and metrics in `pusher-watcher`

- The robots' task vocabulary consists of two tasks: `push-box` and `watch-box`, both parameterizable.
- For the `watch-box` task:
  - Required resources are a laser and a camera.
  - Goal color is parameterizable.
  - The metric measures perceived proximity to the box.
- For the `push-box` task:
  - Required resource is a camera.
  - Box color, visual offset, and velocity are parameterizable.
  - The metric measures the error in visual offset.

Putting it all together

- We start with some robots with cameras, and some others with both cameras and lasers.
- The user (who wants the box pushed) poses a `relocate-box` task to MURDOCH; this hierarchical task is composed of one `watch-box` task, which has two child `push-box` tasks.
- The `watch-box` task is auctioned off to the most fit capable robot, which becomes the `watcher`.

(still) Putting it all together

- The `watcher` then executes its task, which consists of:
  - finding the goal
  - determining the angular error of the box
  - calculating new pushing velocities
  - auctioning pairs of `push-box` tasks
  - monitoring the pushers' progress
- `push-box` task contracts are 3 seconds in length.
- When the situation does not change, old contracts are simply renewed.
- Non-performing robots are blacklisted.

Pusher failure

Pusher failure and recovery

Partial pusher failure

Curved trajectories

Conclusions

- We have implemented and experimentally validated `pusher-watcher`, a novel distributed algorithm for box-pushing by teams of mobile robots.
- Using the auction-based task-allocation facilities provided by MURDOCH, `pusher-watcher` is tolerant to robot failures, and efficient in its use of resources.

gerkey@robotics.usc.edu
http://robotics.usc.edu/~gerkey

http://playerstage.sourceforge.net