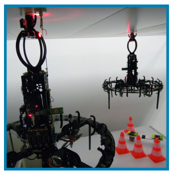


Distributed motion planning for ground objects using a network of robotic ceiling cameras

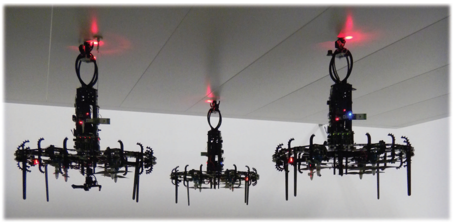
Andreagiovanni Reina, Gianni A. Di Caro, Frederick Ducatelle and Luca M. Gambardella

Dalle Molle Institute for Artificial Intelligence Studies (IDSIA) — Lugano, Switzerland

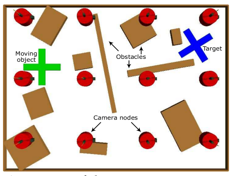
areina@iridia.ulb.ac.be, {gianni, frederick, luca}@idsia.ch

1 Problem description

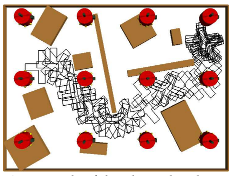
The problem consists in defining the precise sequence of roto-translations of a rigid holonomic object of arbitrary shape that has to be transported from an initial to a final location through a large, cluttered environment. We propose a *distributed planning system* modeled as a swarm of flying robots, each equipped with a camera and wireless communications, that are initially deployed in the environment and take static positions at the ceiling, forming a distributed camera network.



As reference models for the robots we consider the *eye-bot* robots, that have been developed in the Swarmanoid project (*http://www.swarmanoid.org*). Each robot is equipped with a video camera pointing to the floor, and has an infrared system for measuring the relative bearing and distance between two robots and for wireless communications.



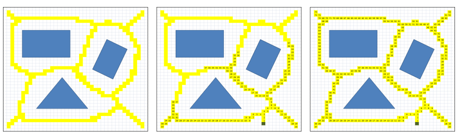
A scenario of the experiment.



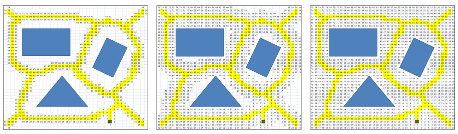
An example of the planned path.

2 Path planning

Our distributed planner is derived from the classical *numerical potential field* technique for a single camera planner (Latombe et al., 1991; Choset et al., 2005) which is computed using the *wave front expansion with skeleton* on a bidimensional uniform cell partitioning. This solution first spreads the potential over a subset of the free space, called *skeleton*, which corresponds to the *Voronoi diagram*; then the potential is computed in the rest of the map. The potential descent is performed using A*.



Diffusion of the potential field over the skeleton.

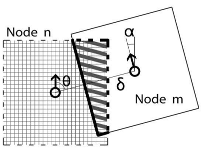


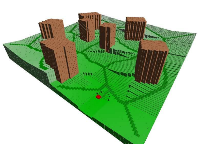
Diffusion of the potential field over the remaining free space.

3 The three phases of the process

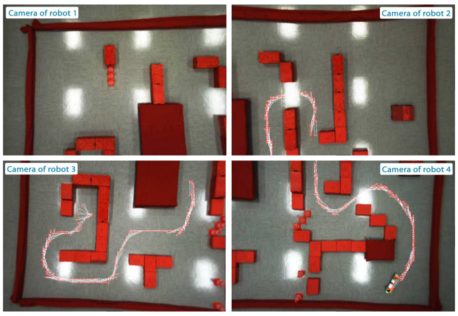
The distributed algorithm has three phases:

1. ***Neighbor detection***. Each robot builds a neighbor table, in which the relative positions of nearby nodes are stored with some estimation error.
2. ***Potential field diffusion***. Each robot expands the potential field on its part of the map, and sends to its neighbors the frontier values.



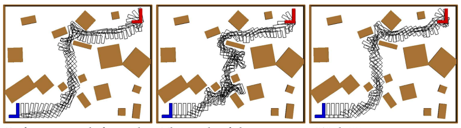


3. ***Path calculation***. The robot above the start position begins path calculation. When the trajectory exits from its area of view, the robot sends the object coordinates to a selected neighbor. Then the process iterates from robot to robot until the target position is reached.



3.1 Heuristics to reduce local minima attraction

Given the distributed nature of the approach, the same local minimum in the potential field can negatively affect the path calculation phase of multiple robots. We propose two sets of heuristics: a first set for the *detection and removal of local minima* from the map, and a second set for *reducing the negative effects of the local minima attraction* (computational time and quality of the trajectories).



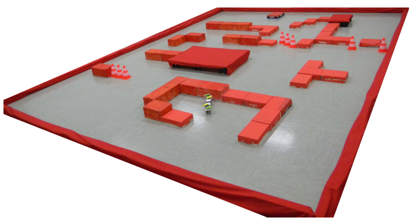
Reference path from the global planner | The path of the system without heuristics | With Heuristics.

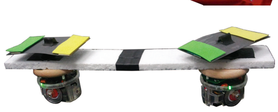
3.2 Local Adaptivity

An important advantage of our distributed system is that it can locally and quickly *detect and adapt to a change in the environment* (e.g., dynamic obstacles). A centralized system would correct the Voronoi skeleton, repeat the potential field diffusion and restart the path planning. In our distributed architecture, the system can reduce the re-initialization costs by locally replanning only a limited part of the path. A node that detects a change in its local map informs the other nodes only if an alternative local partial path cannot be found.

4 Implementation with real robots

The proposed approach was validated on a set of experiments in a real setup. The holonomic object moving on the ground is implemented through a set of 2 non-holonomic robots, the *e-pucks*, interconnected by a rigid structure. In this way, they form an object with a relatively large shape, which is able to rotate and move in any direction. The size of the moving area is 33 m^2. The multi-robot system on the ceiling is implemented with a set of 4 cameras connected to different computers. Each camera is controlled by an independent process, which co-operates and communicates with the other processes, locally plans the path, and then directs the navigation of the e-puck system through the ground area under its local field of view.



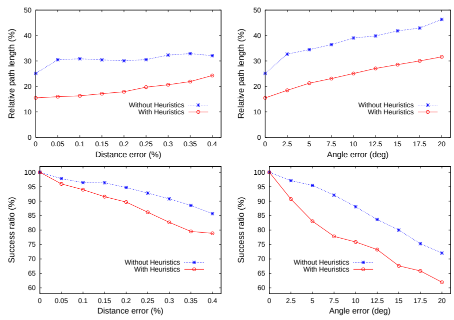


5 Experimental results

We studied in simulation the performance of the proposed planning solutions in terms of: effectiveness, efficiency, scalability, and robustness to alignment errors. As performance metrics, we selected *success ratio*, the percentage of successful runs, and *path quality*, the relative length of the path compared to the path calculated by a *centralized algorithm with complete and perfect knowledge*. We considered a set of 25 sample scenarios with varying area dimensions, position and number of eye-bots, shapes of moving object, and obstacle positions.

5.1 Effect of heuristics vs. position errors

We assume that the partial view of neighbor robots overlaps, in order to permit the sharing of the rigid object position. However, the overlapping is subject to errors deriving from errors in camera calibrations and in the measure of robots' relative positioning. We studied the effect of the *relative positioning error* between nodes in terms of *angle* and *distance* error for the algorithm with and without the heuristics.



Performance vs. errors. The error values on the x-axis indicate the standard deviation of a zero mean Gaussian distribution used to sample the distance/angle error between the robot pairs. For each scenario we ran 40 trials. Each data point represents the average of 25x40 experiments.

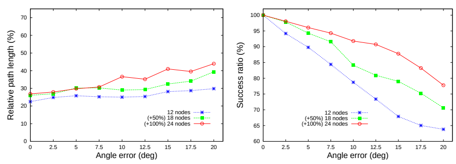
For relatively low errors the performance is always very close to that of the centralized algorithm, while for increasing errors:

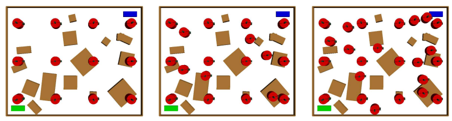
- The algorithm with heuristics degrades rapidly in terms of success rate but slowly for path quality.
- The algorithm without heuristics behaves in opposite way.

In both cases, the system is relatively sensitive to errors on the angle, while it is quite robust to distance errors.

5.2 Scalability performances

We study the performance of the distributed system with respect to an increase of system's resources. In the plots below we show the effect of increasing the *density of the nodes* over a fixed area while varying angle errors. Increasing node redundancy allows the system to deliver a higher success ratio. This effect is more marked in the experiments with larger errors, in which the presence of a larger number of robots can balance the effect of these errors to find alternative valid paths. On the other hand, the increase in the density leads to longer paths.





12 robots | 18 robots [+50%] | 24 robots [+100%]