3D-vision-based robot navigation: first steps

Luc Robert, Régis Vaillant and Michel Schmitt
INRIA
Sophia-Antipolis 2004 Route des Lucioles
06565 Valbonne Cedex France

Abstract

This article shows a way of using a stereo vision system as a logical sensor to perform mobile robot navigation tasks such as obstacle avoidance. We describe our system, from which the implementation of a task described by an automaton can be done very easily. Then we show an example of a navigation task.

1 Introduction

Many approaches have been proposed to the problem of mobile robot navigation. Some aim at following trajectories by servoing on feature that might be detected and tracked by sensors (the edges of a road [7], predefined beacons [1]). Our main goal is to obtain a passive-vision-based system that can *today* navigate in an unknown environment, avoid obstacles and collect some information without being constantly looked after. To achieve this, we use a trinocular stereovision system, which is a powerful and reliable vision sensor device. This system avoids developing sophisticated algorithms to detect obstacles in 2D images.

2 Hardware and software tools

2.1 Description

Our mobile robot is a *robuter* (fig 1), with two driving wheels and a front turret that supports the set of cameras and allows rotation around a vertical axis.

The primitives we use to describe a robot move are directly correlated to the amount of rotation of each driving wheel to achieve this move. Actually, the low-level implementation invokes a P.I.D. supplied by an odometric sensor on each driving wheel.

Computation is processed on the *Capitan* parallel machine and its host machine, a *Sun* workstation. The communication between these computers and the robot is established through a video and radio links.

2.2 The vision sensor

The processing of a triplet of stereoscopic images goes through the following steps: computation of the intensity gradient using recursive filtering [4], non-maxima suppression, thresholding by hysteresis, edge linking [5], polygonal approximation [2], trinocular stereovision [6].

This sensor device provides us with 3D segments in a frame associated to the system of cameras. Its implementation on *Capitan* has been performed by Régis Vaillant [8], and today it takes about 15s to process a triplet of 256×256 images.

3 Navigation

Before moving the robot, we check whether an object lies on the planned trajectory by sensing in the direction of this trajectory and analysing the set of reconstructed 3D edges.

3.1 Controlling moves

Knowledge of the robuter's geometry and of the orders for executing a robot move provides an estimate of the displacement involved within this operation. However, due to hardware reasons, this estimate is not very accurate in comparison to the degree of accuracy of the odometry on each wheel. Indeed, the final position of the robot highly relies on the synchronization of the two driving wheels.

Since successive executions of the same order lead to a relatively invariant displacement, we decided to allow a finite set of moves, and for each move to rely on a value of the displacement which has been previously determined during a calibration phase.

To each possible move we attach one orientation of the turret, so that the camera system faces the motion space, i.e. the portion of space to be occupied by the robot while performing this move (cf fig 1). To check whether the motion space is free, we apply the vision process and build a local map of the 3D space in the view direction. In practice, the view angle is not wide enough to encompass the entire motion space associated to one move, and the way we circumvent this problem is described in paragraph 3.3. Three cases may occur:

- There are 3D edges in the motion space or close to it.
- There is no such edge, and there are 3D edges further in the view direction.
- Very few edges have been reconstructed.

Only in the second case is the move authorized. The first case corresponds to the presence of an object in the target position. The third case usually occurs when the robot is so close to an object that there is no edge in the view angle or that disparity is too high for the edges of this object to be matched.

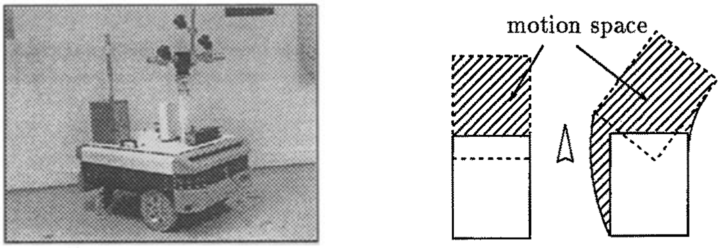

Figure 1: The mobile robot and its motion space

3.2 Behaviour automation

The behaviour of the robot is described by an automaton, each state of which corresponds to one move, and involves a complete cycle:

- determine if the move can be performed
  - move the turret in the adequate direction
  - activate the vision sensor
  - find out whether the motion space is free
- move the robot if possible
- go into the following automaton state, according to the fact that the move was or was not authorized.

The advantages of modelizing behaviours with finite state automata are described by Brooks [3]. Using this structure, we implemented the following task:

move straight as long as possible, otherwise turn right

Because of the geometry of the robuter, it is sometimes impossible to make the robot turn right while avoiding obstacles, for instance when the left side of the robuter is very close to a wall. Then, moving away from the wall requires maneuvers. The automaton which deals with the whole task is described in fig 2. For each possible move, the orientation of the turret to perform sensing is represented on the right side.

3.3 Building a 3D map

Since we know the displacement generated by each robot or turret move, we can project several 3D local maps in a global frame to get a larger 3D map of the environment.

We used those 3D maps to solve the view angle problem raised in paragraph 3.1. A solution to that problem would have been to turn the turret and sense several times, then merge the 3D results *before each robot move.* To enlarge the view angle while avoiding spending too much time in the checking step, we decided to allow only one sense per move (in a direction that allows to see an important portion of the motion space) and to merge the results provided by the last previous sensing processes.

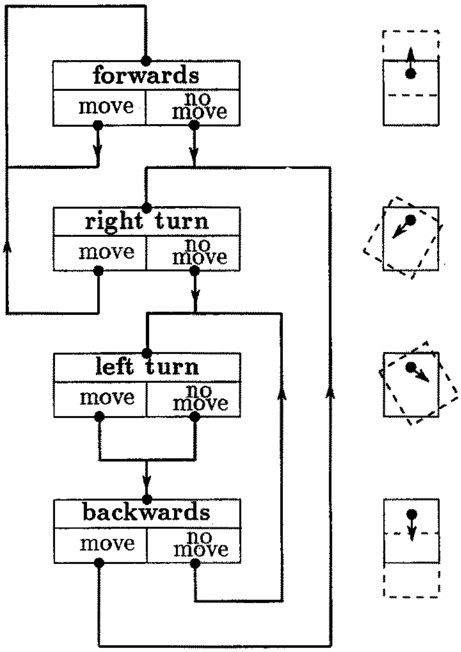

Figure 2: The behaviour automaton

4 Results

Moving every 20s, the robot is able to travel in a room for a few hours while avoiding obstacles. In figure 3, we show one local 3D map, and two different views of a merge of several local 3D maps.

Our system still has two main drawbacks:

- The turret can only move around a vertical axis, so there may always be objects out of the view area, for instance objects of small height located close to the robot.
- Our system is not fast enough to deal with rapid moving objects. Anyway, the use of a real-time hardware architecture for vision will soon help solve that problem.

5 Conclusion

The system we developed allows to implement behaviour automata very easily, and to run them directly on a mobile robotics system. An example of automaton was given that modelizes a simple navigation task. Its execution in different environments led to long and exciting excursions, which tended to prove that much more complex navigation tasks can be implemented within the same structure.

Future work will involve the insertion of a fusion algorithm to build more precise 3D maps and a better model of the environment, then deal with recognition of primitives such as corners, walls, or doors.

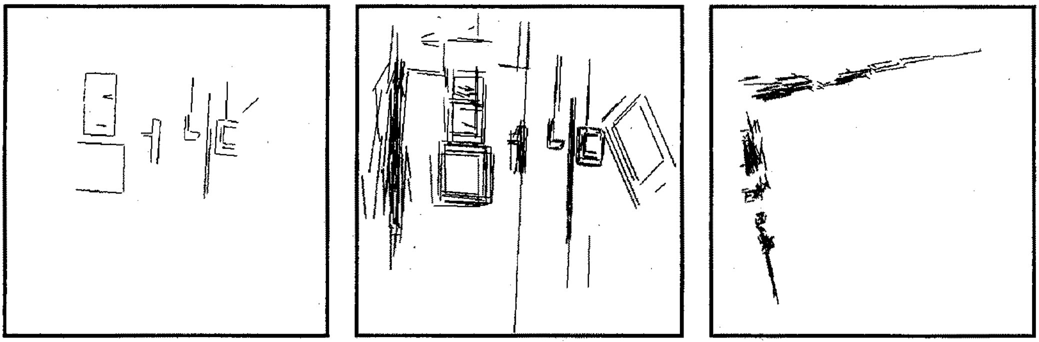

Figure 3: Several views of a corner

Acknowledgements

We would like to thank Pr. Olivier Faugeras for his helpful comments, and Hervé Mathieu for taking care of our robot.

References

[1] J. Amat and A. Casal. Environment Recognition for Automatic Guidance of Agricultural Machines. In *First Workshop on Multi-sensor Fusion and Environment Modelling*, October 1989.

[2] M. Berthod. Approximation polygonale de chaînes de contours. Programmes C, 1986. INRIA.

[3] R.A. Brooks and A.N. Flynn. Robot Beings. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 2–10, 1989.

[4] R. Deriche. Using Canny's Criteria to Derive an Optimal Edge Detector Recursively Implemented. In *The International Journal of Computer Vision*, pages 15–20, April 1987.

[5] G. Giraudon. *Chaînage efficace de contour.* Rapport de Recherche 605, INRIA, February 1987.

[6] F. Lustman. *Vision Stéréoscopique Et Perception Du Mouvement En Vision Artificielle.* PhD thesis, Université de Paris-Sud Centre d'Orsay, December 1988.

[7] C.N. Thorpe, S. Shafer, and T. Kanade. Vision and Navigation for the Carnegie Mellon Navlab. *IEEE Transactions on PAMI*, 362–373, 1988.

[8] Régis Vaillant, Rachid Deriche, and Olivier Faugeras. 3D Vision on the Parallel Machine CAPITAN. In *International Workshop on Industrial Application of Machine Intelligence and Vision*, April 1989.