Vision-based Navigation and Reinforcement Learning Path Finding for Social Robots

Xavier Pérez*, Cecilio Angulo*, Sergio Escalera+ and Diego Pardo*

* *CETpD. Technical Research Centre for Dependency Care and Autonomous Living, UPC,*
Neàpolis, Rambla de l'Exposició, 59-69, 08800 Vilanova i la Geltrú, Spain
xavips@gmail.com, cecilio.angulo@upc.edu, diego.pardo@upc.edu
+ *Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via 585, 08007 Barcelona, Spain*
+ *Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra*
sergio@maia.ub.es

October 1, 2010

Abstract

It is proposed an exportable and robust system for automatic Robot Navigation in unknown environments. The system is composed by three main modules: the Artificial Vision, the Reinforcement Learning, and the reactive anti-collision module. The aim of the system is to allow a robot to automatically find a path that leads to a given goal, avoiding obstacles, only using vision and the least number of sensors.

Keywords: Robot Vision, SURF, BoVW, Motion Field, Robot Navigation, Reinforcement Learning, Policy Gradient.

1 Introduction

Path finding for mobile robots is a complex task composed by required and challenging subgoals. In order to follow the best route between two points in the environment, it is usually needed a map to optimize the route and follow it. However, it would be more useful a solution for which it was not necessary to know the world's map.

This work presents two important challenges, *path finding* and *navigation*. Path finding is considered as the high level robot guidance from place to place, whereas term navigation is used through the document as the set of subprocesses needed to fulfill path finding decisions.

2 Overview

The presented problem can be structured in three layers. First of all, path finding layer is the high level. It looks for the robot finding the exit of an unknown maze in the real world. To achieve this goal, the robot has to be able to learn a route and follow it avoiding collisions. Therefore, it is needed to perform reliable actions and a reliable state representation. These constraints pack the second layer: the navigation layer. Finally, the third layer, named framework layer, is the lowest level layer. To fulfill second layer goals in a remote way, it is needed a complete working environment, a stable robot framework, and a reliable communications system.

2.1 Reinforcement Learning

According to the Reinforcement Learning (RL) paradigm, robot should take actions within its universe, looking for maximizing some notion of cumulative reward. RL algorithms attempt to find a policy that maps its current state to the actions the robot should take in those states, where the world is typically formulated as a finite-state Markov decision process (MDP). Formally, the basic RL model, as applied to MDPs, consists of:

- set of world states X.
- set of actions U.
- set of scalar rewards $\in \mathbb{R}$.

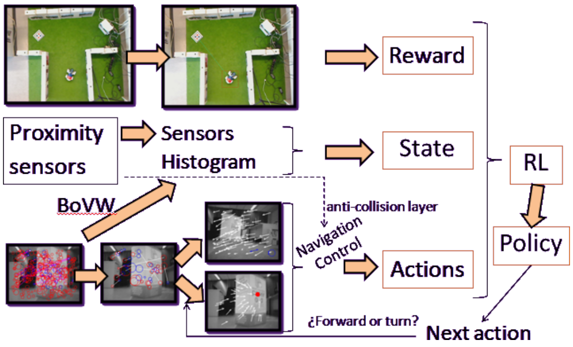

Figure 1: System overview

At each time t, the agent (i.e. the robot) perceives its state $x_t \in X$ and the set of possible actions $U(x_t)$. It chooses an action $u_t \in U(x_t)$ and receives from the environment a reward r_t and the new state x_{t+1}. Based on these interactions, the agent must develop a policy π. Our approach, shown in Figure 1, is defined by:

- set of world states $x \in \mathbb{R}^n$, where $n =$ dictionary size $+$ 3 sensors.
- set of actions $U = [$*forward, backward, left, right*$]$.
- set of scalar rewards $r \in \mathbb{R}$.

Where, first of all, *turn left* and *turn right* actions are referred to turn 90 °, since 360 possible angle would generate a large ramification problem. Then, reward will take heuristic values depending on distance between Sony Aibo and goal, and their relative distance (subsection 2.2). Finally, state will be represented by histograms extracted from images from Sony Aibo camera.

As is explained in subsection 2.3, the method used to compute histograms is named Bag of Visual Words (BoVW) [5]. It depends on a given dictionary of visual words, in our case fixed to 50. In addition, edge detector sensor and two infrared sensor values are used, with represents $n = 53$. This value implies a high state space dimensionality, overnight to grid the state space supposing all states will be visited. In this case, the problem can not be addressed using traditional RL algorithms like Sarsa or other algorithms based on Temporal Difference (TD) [4]. It is necessary to look for a continuous RL algorithm which supports high state dimensionality, therefore, a Policy Gradient Reinforcement Learning method (PGRL) [3] is needed. It is chosen Natural Actor-Critic Algorithm described in [2] because it supports a high *state dimensionality*.

2.2 Reward

In order to guide the learning process of the robot, the Reinforcement Learning (RL) algorithm needs some kind of reward, depending on the goodness of the current state and the last action applied $r_t = \Upsilon(x_t, u_t)$. In our case, the goodness depends on the proximity of robot to the goal, and its relative orientation, computed using a zenith camera. It is important to say that **in any case** positions will be used directly on RL module, for example to define the state. This will only be used through computing the reward.

Environment for this module could be considered as an industrial environment: Camera height is static, illumination ranges are fixed and there are not terrain variations; therefore, marker color parts always will have similar areas, and distance between them always will be much the same. This simplification give us the option to put artificial landmarks on the robot's body to track it and on the goal to locate it, using a fast and robust color filter.

It is important to remark that artificial landmarks are not used **for anything else** along the work.

2.3 State definition

It is needed to describe robot position and orientation with a high level of certainty, i.e. similar states on the map should have similar state representations and very different state representation is due to distant or very different states. Only using robot sensors, without a given map, without the possibility to build our own map, and without the use of artificial landmarks, it is not possible to determine the global position of the robot. Our approach uses proximity sensors and "Bag of Visual Words" (BoVW) [5] on images from robot's camera. BoVW is the computer vision version for "Bag of Words" (BoW), a method for representing documents frequently used on Information Retrieval. BoVW follows the same idea for image representation, where each image is treated as a document, and features extracted from the image are considered the words. Process to create the dictionary consist on take a huge number of pictures of the maze and cluster their features, as is shown in Figure 2.

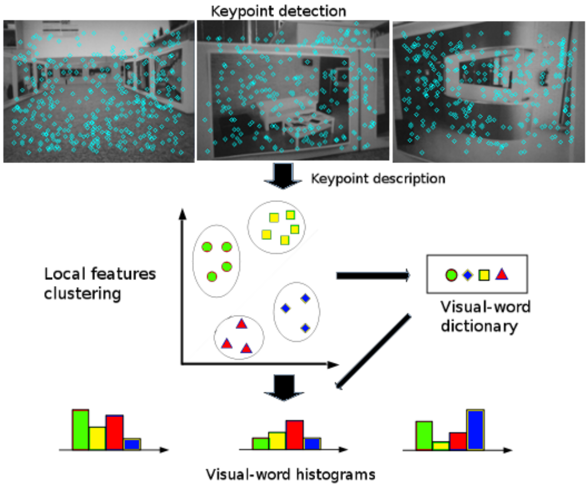

Figure 2: Bag of Visual Words (BoVW)

Once dictionary is built, the process to describe images as vectors of "words" is described in Figure 2. First of all, features must be extracted from each image, it is used SURF (Speeded Up Robust Features [1]) descriptor; then, it is determined in which cluster belongs each feature, getting a BoVW for each image; finally, frequency of occurrence of each word is computed to fill histograms.

2.4 Actions

Actions ordered by RL module must always carry out in the same way. Therefore, it is needed to avoid unexpected behaviors implementing reliable actions: *controlled forward* and *controlled turn*. Both controls have their reasoning particularities, but the first steps of image processing are shared by both modules and *State definition* (subsection 2.3). Common first step is feature extraction, applying SURF [1] on every image received from Aibo camera. Second one, only shared by navigation controls, is to find correspondences between features from consecutive images; obtaining a set of *motion vectors* describing robot motion information from Aibo's head point of view in 2D. Moreover, simple odometry information is included.

2.4.1 Forward

Our approach to solve forward control navigation could not be to walk toward something, but to use consecutive images to get motion information through calculate the Vanishing Point (VP). VP is the appearance of a point on the horizon at which parallel lines converge, i.e. given a stereovision system like human vision, VP is a point in a perspective where real-world parallel lines intersect between them. We do not have a stereovision system, because Sony Aibo only has one camera. However, *Motion Vectors* could be used as our particular real-world "parallel lines". As a consequence, VP could be achieved looking for *Motion vector* intersections, as shows Figure 3. Intuitively, VP is the focus of the movement i.e. VP shows the direction of the movement of the robot. Therefore, control will consist on maintain VP in the center of the image.

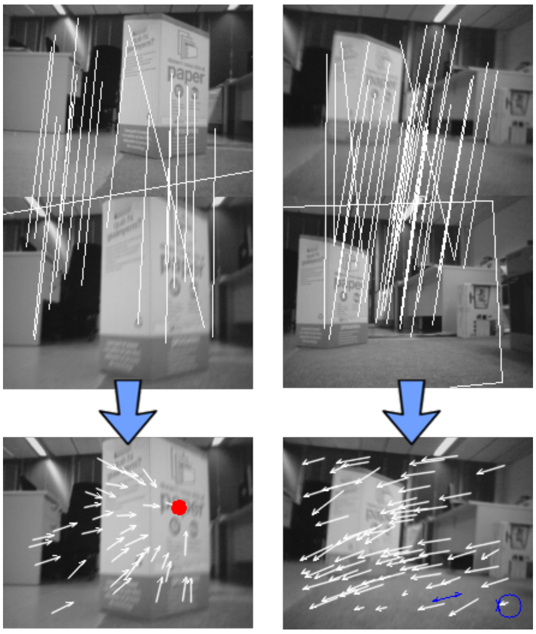

Figure 3: Top: Correspondences between consecutive images; Bottom left: motion vectors and vanishing point; Bottom right: motion vectors and mean motion vector

2.4.2 Turn

The main idea is to turn the head in an specific angle, using neck encoders; then, start turning the body in the direction the head is pointing while robot keeps its head still; finally, turn is completed when head and body are aligned. To maintain its head still, Sony Aibo point of view is not fixed in any object or point. By contrast, it tries, during all the process, to continue watching the same image, avoiding image modifications. When robot is turning, *Motion vectors* are parallel lines in the image indicating the turn sense and its direction and magnitude (shown in Figure 3): *steering angle*. This angle describes the distortion suffered by the image and it is used as the error signal on turn Navigation Control to correct pan and tilt angles.

3 Conclusions

In this work we presented a new approach for navigation control of mobile robots. Designed Vision-based navigation works really well on Sony Aibo, and we hope that it could work even better on wheeled robots. The proposed system only uses the robot camera to achieve a controlled loop to go forward and other one to turn a desired angle. In addition, the robot uses proximity infrared sensors in order to avoid obstacles. Moreover, reliable state representation is obtained using proximity sensors and a 50 length histogram resulting from BoVW. Furthermore, zenith camera was used to compute reliably the reward needed by the Reinforcement Learning algorithm (RL). Finally, RL algorithm is able to work with high dimensionality data were implemented and tested. Robot looks for the goal, producing behavior changes based on experience, but without finding the optimal route that reaches the goal. However, it seems a reasonable useful approach despite of the needing of a better configuration for learning optimal parameters in order to achieve the desired results.

References

[1] Bay H. Ess A. Tuytelaars T. Van Gool L., *Surf: Speeded up robust features*, Computer Vision and Image Understanding (CVIU), 2008

[2] Peters, J. Vijayakumar, S. Schaal, S., *Policy Gradient Methods for robotics*, In International Conference on Intelligent Robots and Systems (IROS), 2006

[3] Peters, J., *Machine Learning for Robotics: Learning Methods for Robot Motor Skills*, VDM-Verlag, 2008

[4] Sutton, R. S. Barto A. G., *Reinforcement learning: an introduction*, MIT Press, 1998

[5] Yang, J. Jiang Y.G. Hauptmann, A. Ngo C.W., *Evaluating bag-of-visual-word representation in scene classification*, MIR07 ACMMM, 2007