Chapter 18

An Embedded Vision System for RoboCup

Pedro Cavestany-Olivares
University of Murcia, Spain

David Herrero-Pérez
University of Murcia, Spain

Juan José Alcaraz-Jiménez
University of Murcia, Spain

Humberto Martínez Barberá
University of Murcia, Spain

ABSTRACT

In this chapter, the authors describe their vision system used in the Standard Platform League (SPL), one of the official leagues in RoboCup competition. The characteristics of SPL are very demanding, as all the processing must be done on board, and the changeable environment requires powerful methods for extracting information and robust filters. The purpose is to show a vision system that meets these goals. The chapter describes the architecture of the authors’ system as well as the flowchart of the image process, which is designed in such a manner that allows a rapid and reliable calibration. The authors deal with field features detection by finding intersections between field lines at frame rate, using a fuzzy-Markov localisation technique. Also, the methods implemented to recognise the ball and goals are explained.

INTRODUCTION

In the past few years we have witnessed a significant development within the field of robotics. In particular, there has been a great research on mobile platforms. This research has had to deal with many challenges, as computing resources on board usually are limited and most algorithms require outputs to be in real time. In addition to this, the main exteroceptive sensor of average mobile robots is a low-quality camera, which has led researchers to seek robust and simple filters so that noise is ruled out and imprecisions dimmed. Our work aims at contributing on this research.

In this chapter we describe our vision system used in the Standard Platform League (SPL) which is one of the official leagues in RoboCup competition. All the code, examples and implementations that we present in this work have been developed for Los Hidalgos team, which has participated in the 2010 and 2011 editions of the RoboCup in association with L3M team, and in several international competitions.

DOI: 10.4018/978-1-4666-2672-0.ch018
In the SPL, platform and environment are standard. The standard platform is the Nao robot, a humanoid robot provided by Aldebaran Robotics. Nao is a light, compact, humanoid robot, fully programmable and easy to operate. The Nao version in which our software has been implemented has a biped configuration based in 21 degrees of freedom, which enables it to develop a great mobility. It is 57 cm tall and weighs 4.5 Kg. Concerning the computational architecture, the robot is equipped with an x86 AMID Geode 500MHz processor, 256 MB of SD-RAM and 2GB of flash memory. There are two communication interfaces in the robot: Ethernet and Wi-Fi 802.11g. The operative system is Open, an open Linux based distribution. The main exteroceptive sensors of the Nao are two non-stereo cameras, placed on the forehead and on the mouth of the robot, provided with VGA resolution. The first point to undertake is whether it is worthwhile using both cameras of the Nao, alternatively, or only one, and which one. Switching the camera takes at least 2 sec, and in every switch it is necessary to recalibrate the parameters of each one. We decided to use only one camera. The mouth camera is able to see both the feet of the robot and far away objects, thanks to the head joints. Since the robot has to see the ball when is near, so that it may get ready to kick the ball, and the goals, that usually are relatively far, this is the camera that we work with.

The standard environment is given by the soccer field, and the colour of the involved objects. Thus, player uniforms and goals are coloured: the blue goal belongs to the blue team, and the yellow goal to the red team. The ball is orange. Therefore, colour segmentation and its recognition is critical. For example, due to lighting conditions, orange could easily be mistaken for yellow. Our segmentation approach has been developed to avoid these confusions and to permit a quick and good calibration. Our recognition algorithms are designed to face as many situations of sighting the different objects as possible, especially in the case of goals. However, goals are not frequently perceived in game conditions, because robots are constantly looking at the floor when seeking the ball. Moreover, natural landmarks over the field are constantly perceived, i.e. field lines, which can be used to update robot localisation more frequently. For these field lines to be successfully used there are two issues that must be addressed: a robust field features detection in real time, and a robust localisation able to manage such information. We are using a fuzzy-Markov self-localisation technique, in which the robot location is modelled as a belief distribution on a 2½ D possibility grid (Buschka et al., 2000). This formalism allows us to represent and track multiple possible positions where the robot might be. Furthermore, it only requires an approximate model of the sensor system and a qualitative estimate of the robot displacement.

According to the league rules, all the processing must be done on board. For practical reasons, it has to be performed in real time, which prevents us from using time consuming algorithms. Nao camera provides an image every 33 ms, and our algorithms should be able to process every image within this time. As SPL allows any change in the software scope, we will try to avoid the middleware provided by Aldebaran, which slows down the image process.

We follow the ThinkingCap architecture (Martínez-Barberá & Saffiotti, 2000), a two-layer architecture which clearly reflects a cognitive separation of modules. From the conceptual point of view, modules are arranged by the nature of their processing tasks. From a software point of view, interfaces are clear and well defined, so that replacing or improving modules is not a difficult task.

Specifically, the objectives of this chapter are:

- To describe a fully functional vision system integrated in ThinkingCap architecture on a limited mobile robotic platform.