Yue-Fei: Object Orientation and Id without Additional Markers

Jacky Baltes

Centre for Image Technology and Robotics
University of Auckland,Auckland
New Zealand
j.baltes@auckland.ac.nz
http://www.citr.auckland.ac.nz/~jacky

Abstract. This paper describes a novel approach to detecting orientation and identity of robots using a global vision system. Instead of additional markers, the shape of the robot is used to determine an orientation using a general Hough transform. In addition the movement history as well as the command history are used to calculate the quadrant of the orientation. The identity of the robot is determined by correlating the motion of the robot with the command history. An empirical evaluation shows that the performance of the new video server is at least as good as that of a traditional approach using additional coloured markers.

1 Introduction

This paper describes a new approach to image processing in the RoboCup domain, that has been implemented in the video server of the All Botz, the University of Auckland F180 team. The videoserver has been used with good success last year, but problems remained with correctly identifying robots. These shortcomings have been resolved in our new videoserver, called YUE-FEI.

In the F180 league, most teams use a global vision system to control up to five robots per team in a game of robotic soccer. In order to be able to control the robots, coloured markers or bar codes are put on top of the robots to simplify the vision task.

Coloured markers or bar codes are an easy and robust method, but have two big disadvantages. Firstly, the calibration of sets of colours, so that they can be detected over the entire playing field and do not interfere with each other is a very time consuming task. The resulting calibration is not robust. Even small changes in the lighting conditions require a re-calibration. Secondly, these methods do not scale to large teams of robots with eleven players or more.

Therefore, the All Botz developed a new flexible and scalable approach, that uses the original shape of the robot as its sole source of vision information. In other words, the robots are unaltered except for the addition of a coloured marker, which is required by the F180 RoboCup rules. A generalized Hough transform is used to infer the orientation of the robot from a sub-image. The image processing determines an exact orientation of one side of the robot (an angle between 0 and 90 degrees), but there is not sufficient information to determine the quadrant of the angle. Thus, the quadrant is determined by correlating the movement history (e.g., dx, dy) and current command (e.g., move forward) to the motion of the robot.

The most difficult vision problems in the RoboCup domain is to determine the identity of a robot. All other teams use unique features of the robots to determine their id. As the number of robots increases it becomes more difficult to find unique features that can be recognized efficiently and robustly. In our system, the identity of the robot is determined through correlating the command stream from the individual controllers to the observed behavior of the robot.

Section 2 describes the vision problems associated with the F180 league and how these problems were addressed by other teams previously. Section 3 describes the design of YUE-FEI, the new video server of the All Botz. The results of an empirical evaluation comparing the performance of YUE-FEI against that of a traditional video server are shown in section 5. Directions for future research and further improvements are shown in section 6.

2 Global Vision in the RoboCup

Most teams in the RoboCup competition use additional coloured markers to create feature points on the robot. In the simplest case, the two points have a distinct colour, which makes it easy to determine the orientation of the robot by relating it to the orientation of the line between the two points.

One of the most difficult aspects of the vision processing is the visual identification of robots. To be able to control a robot, the control system needs to know the identity of the robot, so that the commands are sent to the correct robot (e.g., Robot 3 move forward).

So far, the only solution to this problem suggested by teams competing in the RoboCup are to use individual colour markers, “bar codes” , manual tagging, or placing robots into predefined positions.

Most teams identify their robots through different colours. Although this approach is conceptually simple, it is non-trivial to find a parameters for a colour model that allows the detection of a colour over the entire playing field.

Another possibility is to identify a robot using some easily distinguishable geometrical pattern. Popular methods are to identify different patterns based on their size or their aspect ratio.

A third possibility is to manually identify (tag) each robot before play starts or play continues after a stoppage. For example, the user may click on robot one through five in turn. The vision server then continues to track the robot.

3 The Yue-Fei Videoserver

The solutions described in the previous section have severe disadvantages since they do not scale up to larger teams and to more flexible camera positions.

It is difficult to distinguish between more than six different colours reliably over the whole playing field. The selection of the additional colours becomes important and often leads to conflicts with other team, since they also try to find a suitable set of colours. At RoboCup-99, the All Botz as well as most other teams spend a lot of time calibrating their colours and there were some discussions about what colours are permissible.

Based on these observations and our experiences at RoboCup-99, we started work on the design of a new video server Yue-Fei. The design goals for the new video server were to provide a scalable, robust, flexible solution to the orientation and identification problem. For the reasons mentioned above, we ruled out additional colours and additional patterns on the robot.

If we do not want to use additional patterns, then what else is there? The only information left is the image of the robot itself. So the goal was to design a videoserver that uses only a single marker and no other patterns on the robot.

3.1 Orientation information using the Generalized Hough Transform

Figure 1 contains three zoomed views of our robots from our video camera. The views correspond to the worst case (i.e., the robot is at the far end of the playing field) for our vision system. As can be seen, the most prominent features are the edges along the top of the robot. Other features (e.g., the wheels are not always visible and hard to distinguish). Therefore, we decided to use these edges as features and to infer the orientation of the robot from them.

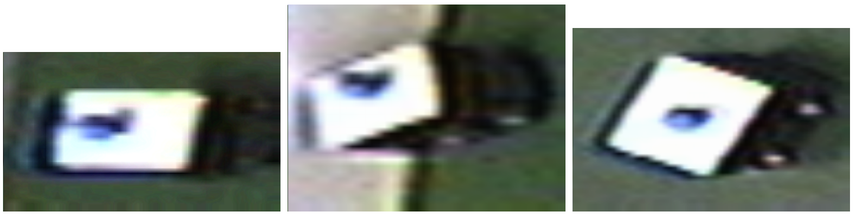

Fig. 1. Some sample images of our robots taken at the far side of the field.

This idea faces an immediate problem, since the robots are almost square. This means that it is *impossible* to determine the orientation of the robot from a single image. Given the angle of the edge, there are four possible orientations for the robot, which can not be distinguished without further information.

Furthermore, since all robots have exactly the same shape, it is impossible to identify the robot. Therefore, we decided to use additional information (e.g., history of the cars, current commands, motion of the robot) available to the video server to disambiguate the orientation and to identify the robot. This part of the system is described in section 4.

Given the real world coordinates of the robot, the surrounding image corresponding to a square area of the diameter of the robot is extracted. All further processing is limited to this local neighborhood.

Figure 2 shows the output for the three sample images given in Fig. 1. The contour of the robot is shown in black. As can be seen, using even a very coarse colour calibration, the edges of the robot can be traced accurately even under bad conditions.

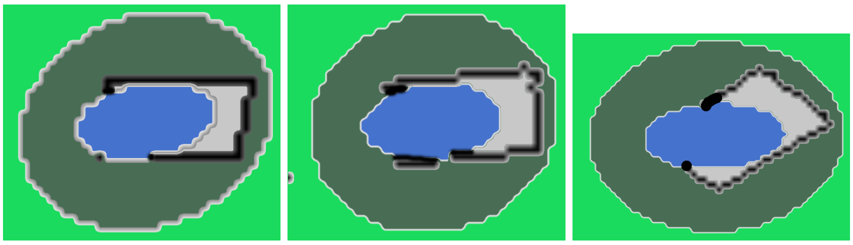

Fig. 2. The image of the robot after preprocessing. Pixels that are too far or too close are ignored. Pixels matching the colour of the top of the robot and pixels on the contour.

Given the position of the edge pixels, a general Hough transform is used to compute the possible orientation of the robot in the first quadrant [1].

The Hough transform is a popular method in computer vision to find lines and other shapes. The basic idea for the Hough transform is to collect evidence to support different hypothesis. The evidence for different hypotheses is accumulated and the hypothesis with the strongest support is returned as the solution.

Figure 3 shows an example of the geometry in our problem. Each edge pixel can be at most on four possible edges (E_1, E_2, E_3, E_4 in the figure). It is easy to see that

$$\begin{aligned} \alpha &= sin^{-1}(w/d) \\ \beta &= sin^{-1}(l/d) \end{aligned}$$

Therefore, the corresponding angles for the edges can be computed as:

$$\begin{aligned} E_1 &= \theta + \beta \\ E_2 &= \theta - \beta \\ E_3 &= \theta + \alpha \\ E_4 &= \theta - \alpha \end{aligned}$$

Edges E_1 and E_2 are only possible solutions if the distance d between the center point (X_C, Y_C) and the edge pixel (X_E, Y_E) is greater than the length of the robot l. Similarly, edges E_3 and E_4 are only solutions if d is greater than the width w of the robot.

The algorithm works by allowing each pixel to "vote" for the orientation supported by this pixel. Once all pixels have been processed, the orientation with the most number of votes, that is supported by the most pixels is chosen as the correct orientation. The hough space (vote space) consists of a one-dimensional array with 18 entries, which gives us a resolution of 5 degree.

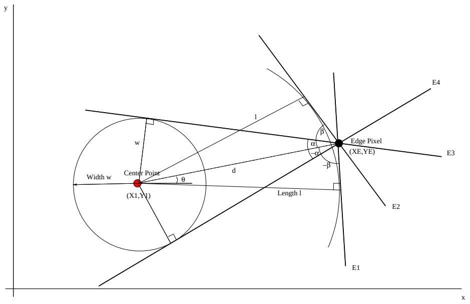

Fig. 3. The four possible edges E_1, E_2, E_3, and E_4 (and therefore, the possible orientations) for an edge pixel (X_E, Y_E).

4 Identification Using Bayesian Probability

As mentioned previously, since all robots in our team look identical, the vision information is insufficient to identify them. YUE-FEI uses two additional sources of information to determine the identity.

Firstly, YUE-FEI predicts the motion of the robot and tracks it. Should the robot be found in the predicted position, its identity and its associated probability is not changed. If the robot is found in the neighborhood of the predicted position, its identity is not changed, but the probability in the identity is reduced by a factor of 0.9 or 0.7, dependent on how far the robot was found from the predicted position.

Secondly, YUE-FEI observes the motion of the robot over a number of frames and assigns it one of seven states: not moving, accelerating, decelerating, turning clockwise, not turning, and turning anti-clockwise. The actual steering angle is not used, so there is no difference between, for example, full left and gently left.

YUE-FEI listens to the communication between the clients and the executors. When an executor receives a command (e.g., Robot 3 forward left!), YUE-FEI stores the last command sent to the robot and observes the motion of the robots. Should the motion of the robot agree with the command, the probability of the identity is slightly increased (a factor of 1.1). On the other hand, there are many reasons why a robot does not follow the command: malfunction, an unknown obstacle in the path, noise in video processing, the robot is being pushed by another robot, occlusion, etc. In this case, YUE-FEI slowly decreases the probability of the identity assignment by a factor of 0.9. When the probability of an identity drops below a certain threshold, the identity of the robot is marked as unknown.

5 Evaluation

The performance of YUE-FEI was compared against the performance of our original video server, both with respect to speed and accuracy. The evaluation shows that the performance of the new videoserver is at least as good as that of our original video server.

5.1 Yue-Fei's Processing Speed

The Hough transform is a computationally intensive method, which as a rule should be avoided in real time applications. However, since the position of the robot is known, only a small neighborhood (64x64 pixels) of the image needs to be considered. Also, the number of edge pixels in that neighborhood is small. In the worst case, there are 256 edge pixels. Also, the Hough space is reduced, since we are not interested in the location of the line and since the possible orientations are only between 0 and 90 degrees. These factors explain why there is no noticeable difference in the processing speed of the two videoservers. Both videoservers are able to maintain a 50 fields/second frame rate in the RoboCup domain with eleven objects.

5.2 Yue-Fei's Accuracy

Knowing the orientation of static objects is rarely useful. We are interested in moving our robots to their targets, so the accuracy of the orientation information for a dynamic object is much more important. A dynamic evaluation is more difficult than a static one, since we have no way of knowing the correct orientation for a moving object. We tested YUE-FEI by driving a simple pattern and by observing the orientation information. The correct information was inferred from a kinematic model of the robot. This test showed that the average error of YUE-FEI was slightly less (less than approx. 5 degrees) than that of our original videoserver (less than approx. 10 degrees).

6 Conclusion

This paper presents a new approach to vision in the RoboCup domain. Instead of coloured markers, the system uses geometrical features of the robots to determine their orientation. This means, that the only coloured marker on the robots are markers are used to determine the position of the robot. The orientation is determined by the projection of the robot in the image. The system uses a generalized Hough transform to find edges in the neighborhood of the position of the robot. These edges are used to determine four possible angles (offset by 90 degrees) for the orientation of the robot.

The videoserver correlates the movement of the different robots with the observed behavior to disambiguate the quadrant of the orientation and the identity of the individual robots.

References

1. D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.