Horus: 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 original 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 as well as the identity of the robot. 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.

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 marker ball, 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 HORUS, the new video server of the All Botz. The results of an empirical evaluation comparing the performance of HORUS 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 F-180 league of the RoboCup initiative use a global vision system to obtain information about objects in the domain, including the robots, the opponents, and the ball.

There are three important pieces of information that the global vision system must provide: position, orientation, and identification. The following subsesctions describe related work in obtaining the necessary information.

2.1 Position

The rules of the F-180 league require each robot to mount a coloured table tennis ball in the centre of the robots. Each team is assigned a colour (either yellow or blue). The two goal boxes are also painted yellow and blue respectively. The yellow team shoots on the blue goal and vice versa.

The position of a robot can easily be determined by the image coordinates of this marker ball. Given the height of the robot as well as the extrinsic and intrinsic camera parameters, this location can be mapped back to real world coordinates. The All Botz use a pinhole camera model with two non-linear lens distortion parameters.

The geometry of the All Botz video camera setup makes the accuracy of the camera parameters more important and the computation of these parameters more difficult than that of other teams. However, this side view setup is general and versatile.

Briefly, a calibration pattern is used to find real world coordinates for a number of image coordinates. This mapping from image to real world coordinates is computed using an automatic iterative method. Given this set of calibration points and their real world coordinates, the Tsai camera calibration method is used to compute the parameters of the camera model [4].

2.2 Orientation

Although a single point on the robot is sufficient to determine its position, additional information (e.g., a second point or a vector) is needed to determine the orientation of a robot.

Most teams in the RoboCup competition use additional coloured markers to create a second point 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.

The distance between the two points determines the accuracy of the orientation: the further apart the two points, the better the orientation. The maximum length of the robot is limited by the rules.

The All Botz used this method previously with good success. The variance in the orientation for a static object was less than 10 degrees at the far side of the field.

2.3 Identification

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" or manual tagging.

Most teams identify their robots through different colours. The major problem is that 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 game starts. For example, the user may click on robot one through five in turn. The vision server then continues to track the robot until there is an occlusion or the robot is occluded. This occurs usually during a stoppage in play.

This procedure is time consuming and error prone. Assigning an identification takes about 30 seconds, but needs to be done in every stoppage in play. Also, in the heat of battle it is easy to mistake two robots. Furthermore, as the skill level increases and stoppages in play become less common, there will be fewer chances to change an erroneous assignment.

3 The Horus 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. 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 ball and no other patterns on the robot.

Position information in the current implementation is still based on the marker ball on top of the robot. Since the rules require this marker, it seems reasonable to use it for position information. Since the processing of the orientation (described in more detail in the next section) is computationally more expensive than simple blob detection, the position information is used to "anchor," that is to constrain the following computation to a small subimage (approximately 64 by 64 pixels).

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. The maximum size of this window depends on the geometry of the camera position. In most "practical" situations, the size of the window is less than 64 * 64 pixels.

All further processing is limited to this local neighborhood. The image is divided into four regions, which are shown in the Fig. 2.

- Pixels that are more than half a diameter away from the position. These can not be part of the robot and are ignored.
- Pixels that belong to the marker ball or are very close to it. These pixels are usually noisy and are ignored.
- Pixels that match the top colour of the robot.
- Pixels that belong to the contour of the robot. These pixels are determined after tracing the contour of the robot using a standard edge walking algorithm.

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 worst case conditions.

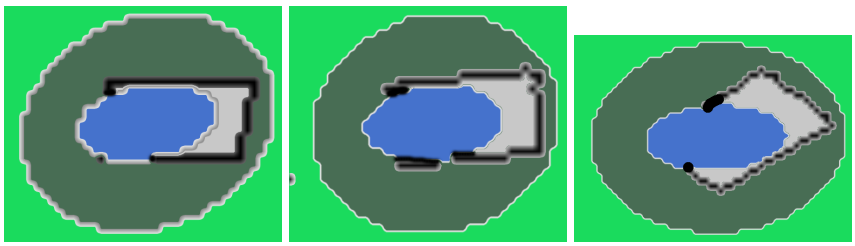

Fig. 2. The image of the robot after preprocessing. Pixels that are too far or too close are ignored. Pixels matching the colour of the top of the robot and pixels on the contour.

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

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

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

$$E_1 = \theta + \beta$$
$$E_2 = \theta - \beta$$
$$E_3 = \theta + \alpha$$
$$E_4 = \theta - \alpha$$

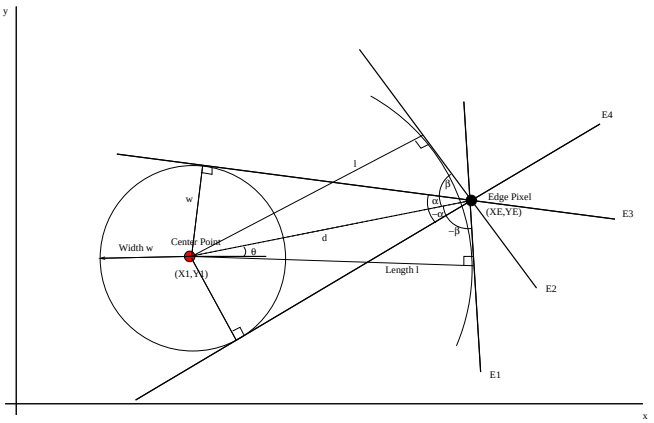

Fig. 3. The four possible edges E_1, E_2, E_3, and E_4 (and therefore, the possible orientations) for an edge pixel (X_E, Y_E).

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.

In theory, this information is sufficient to determine the orientation of the robot +/- 180 degrees. In practice, we normalize the angles to within 90 degrees, since the distance between the edge pixel and the center point is noisy. This makes little difference in resolving the disambiguity about the quadrant, since for a car-like robot two of the four possible orientations can immediately be rules out, since a car can not move sideways.

The hough space consists of a one-dimensional array with 18 entries, which gives us a resolution of 5 degree. For each edge pixel, the value in the array for that position is incremented. Finally, the angle corresponding to the maximum value is returned.

4 Identification Using Bayesian Probability

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

HORUS maintains a probability for the identity of each robot. We use a simple Bayesian model to update this probability when new evidence is encountered.

Firstly, HORUS 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, HORUS observes the motion of the robot over a number of frames and assigns it one of seven states: not moving, forward left, forward straight, forward right, backwards left, backwards straight, and backwards right. The actual steering angle is not determined, so there is no difference between, for example, full left and gently left.

Initially as well as after some errors in the assignment, a robot will have an unknown identity. If a robot has an unknown identity, HORUS will assign it the first free identity that matches the observed behavior of the robot. The initial probability of this identity assignment is 0.5.

5 Evaluation

The performance of HORUS 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. The performance of the vision processing is not a limiting factor in the overall system.

5.1 Horus' Processing Speed

The Hough transform is a compute 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 Horus' Accuracy

Evaluating the accuracy of the orientation information is more difficult. HORUS is unable to determine the orientation completely from just a single image or from a stationary object.

Knowing the orientation of static objects is rarely useful though. 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 HORUS by driving a simple pattern (a circle to the left in the centre of the playing field at constant speed) 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 HORUS was slightly less (less than approx. 5 degrees) than that of our original videoserver (less than approx. 10 degrees).

Another factor that determines the quality of the videoserver is its interaction with the control algorithm. For example, it is unclear if the maximum or average errors are more important. Therefore, we tested the interaction of the orientation information with a non-holonomic control algorithm based on Egerstedt's controller ([3]. This control algorithm was used in time-trials on the Aucklandianapolis race track ([2]) using orientation information from the two video servers. The times were identical in both cases. The limiting factor for the speed was in this case not the accuracy of the video server information, but rather the latency in the control loop, which is at least 20ms.

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 marker balls, which 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 angles and identify each robot.

References

1. D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
2. Jacky Baltes and Yuming Lin. Path-tracking control of non-holonomic car-like robots using reinforcement learning. In *Proceedings of the IJCAI Workshop on RoboCup*, Stockholm, Sweden, July 1999.
3. Markus Egerstedt, X. Hu, and A Stotsky. Control of a car-like robot using a dynamic model. In *Proceedings of the 1998 IEEE Conference on Robotics and Automation*, Leuven, Belgium, 1998.
4. Roger Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–374, Miami Beach, FL, 1986.