

An Omnidirectional Vision System That Finds and Tracks Color Edges and Blobs

Felix v. Hundelshausen, Sven Behnke, and Raúl Rojas

Freie Universität Berlin, Institut für Informatik
Takustr. 9, 14195 Berlin,
{hundelsh|behnke|rojas}@inf.fu-berlin.de

Abstract. We describe the omnidirectional local vision system developed for the FU-Fighters, a RoboCup F180 league soccer team. A small video camera mounted vertically on top of the robots looks at a concave parabolic mirror placed above the camera that reflects the field around. The image is sent via a radio link to an external PC for processing. Our computer vision system can find the ball and detect other robots as obstacles. The walls of the field are also recognized and are used to determine the initial position of the robot. In order to be able to process the video stream at full frame rate the movement of all objects is tracked, including the walls of the field. The key idea of our approach is to predict the location of color edges in the next frame and to search for such color transitions along lines that are perpendicular to the edge.

Introduction

We developed a team for the F180 RoboCup league, the FU-Fighters, that has taken part in the competitions held at Stockholm and Melbourne. To make the robots more autonomous, we replaced the global vision system by an omnidirectional local vision system where each robot carries its own camera.

Three tasks have to be accomplished by the computer vision software that analyzes the captured video stream: detecting the ball, localizing the robot, and detecting obstacles. These tasks are non-trivial, since sensor noise and variances, such as inhomogeneous lighting, are present in the images. The image analysis must be done in real time, which is not easy, due to the enormous data rate of video streams. Some teams need to reduce frame rate or resolution to match the available computing power, however, such an approach leads to less precise or less timely estimates of the game status. To be useful for behavior control, the system also needs to be robust. Unexpected situations should not lead to failure, but to graceful degradation of the system's performance.

Local vision is the method used by most teams in the F2000 league as the main sensor. Some of the successful teams adapted the omnidirectional vision approach. The Golem team [4] impressively demonstrated in Melbourne that, using an omnidirectional camera, sufficient information for controlled play can be collected. Another example for the use of omnidirectional cameras is the goalie of the ART team [3,6,7]. In our league, F180, only three teams tried to play in

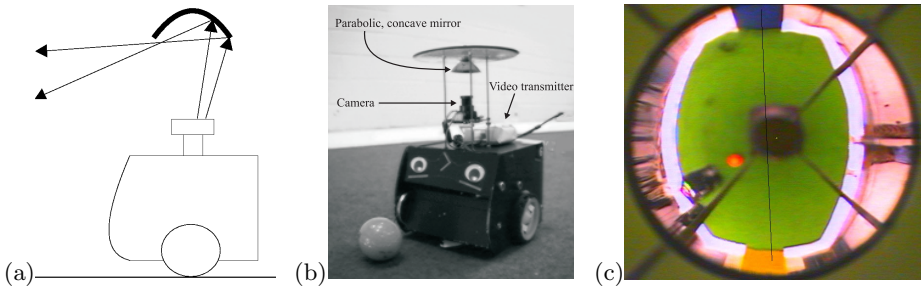


Fig. 1. Omnidirectional camera: (a) principle; (b) physical construction; (c) captured.

Melbourne with local vision. In the smaller form factor the implementation of a local vision system is more challenging than in the F2000 league. Due to space and energy constraints smaller cameras of lower quality must be used and less computing power is available on the robot. Recently, the OMNI team [5] demonstrated controlled play with omnidirectional local vision at the Japan Open competition. This team sends the video stream to an off-the-field computer that contains special purpose hardware for image processing.

We use a similar system design and mount a small video camera and a mirror on top of the robots, as shown in Fig. 1. The camera is directed upwards and looks into a parabolic mirror. The mirror collects light rays from all directions, reflecting them into the camera. The parabolic shape of the mirror produces less distortions, as compared to a spherical mirror. Far-away objects appear larger and are hence easier to detect [1]. The optical mapping preserves the angle of an object to the perception origin. The non-linear, increasing distance function of the camera can be calibrated easily by measuring distances in the image and in the world. To avoid carrying a large computer on the robots, the video stream is transmitted to an external computer via an analog radio link.

The main idea of the paper is to implement a tracking system for the analysis of the video stream produced by an omnidirectional camera that needs to inspect only a small fraction of the incoming data. This allows to run the system at full frame rate and full resolution on a standard PC. For automatic initialization of the tracking, an initial search analyzes the entire video frame.

The paper is organized as follows: The next section describes the initial localization of the robot, the ball and obstacles. The tracking system is described in Section 2. Some experimental results are reported at the end.

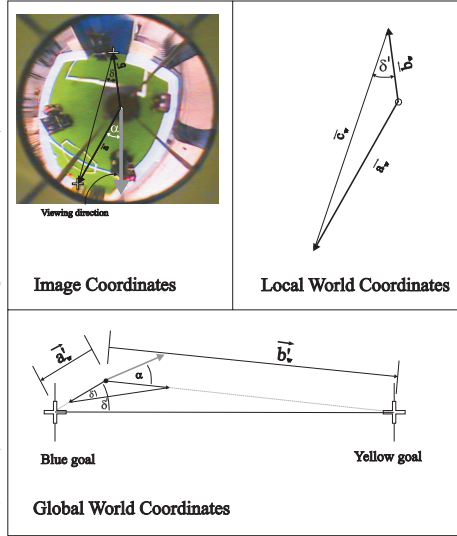
1 Initial Search

1.1 Initial Robot Localization

The task of initial robot localization is to determine the robot's position and orientation on the playing field, given a captured omnidirectional image. Two

different methods have been developed for localizing the robot. The first is fast, but is only applicable when both goals can be found. The second method is more flexible and robust, but also is slower.

Direct Localization uses the distances to the goals and the angle between the goal vectors to determine the robot's position and orientation. First, the vectors \mathbf{a} and \mathbf{b} from the perception origin to the goals have to be mapped to local world coordinates. If \mathbf{a}_w and \mathbf{b}_w denote these mapped vectors, we define $\mathbf{c}_w := \mathbf{a}_w - \mathbf{b}_w$. Now δ' is the angle between \mathbf{c}_w and \mathbf{b}_w . Hence, the position p of the robot lies on a line at an angle of δ' to the line connecting the goals. Knowing the distance to one goal determines p . The orientation is also known, because the angle α at which the other goal appears in the image in respect to the robot's viewing direction is preserved by the optical mapping.



Localization Using Evidence Aggregation consists of two steps. First, plausibility values for several positions are computed using evidence accumulation in a grid. In a second step, these positions are investigated in the order of their plausibility, until the correct position has been found.

If we recognize a goal and can estimate its distance from the robot, a circle segment with radius r_x around goal x is added to the grid. The circles will be drawn more fuzzy for great distances, as the estimation of r_x becomes worse.

Another feature used is the best visible wall that is found by a radial search followed by a Hough Transformation [2]. Lines are sent radially from the robot's perception origin and searched for transitions from the floor to the wall. The

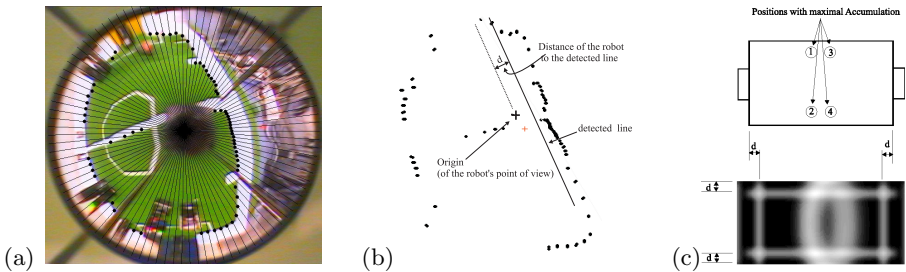


Fig. 2. Using the closest wall for localization: (a) robot next to a wall; (b) detected wall points transformed to world coordinates; (c) grid with goal-circles and wall-lines.



Fig. 3. The ball's appearance varies in color and shape and is similar to robot markers.

transitions found are transformed to local world coordinates, using the cameras' inverse distance function. The corresponding sinusoidal curves are accumulated in parameter space. The most significant local sinusoid maximum corresponds to the best visible wall. Since it is not known, which wall has been detected, for all walls parallel lines are drawn on the grid at the perceived distance (see Fig. 2).

Local maxima are now candidates for robot locations. They are evaluated by a registration procedure (see next section) that computes a quality measure for a model fit. The best candidate is used to initialize the tracking.

1.2 Initial Localization of Ball and Obstacles

Finding the ball is of essential importance for successful play. Although it seems to be an easy task, detecting the ball clearly demonstrates the difficulties of computer vision. Figure 3 shows some images of the ball. Its appearance varies greatly in size, shape, and color. Furthermore, it can be easily confused with robot markers. The detection of the ball can be summarized in three steps: (a) determine all clusters of the ball's color class and their sizes, (b) discard all clusters for which no transition to the floor can be found, and (c) choose the biggest cluster. Obstacles are detected as large clusters of the dark obstacle color. If clusters are close to each other, they are merged.

2 Tracking Objects

Tracking of color edges and blobs is key to the low computational load of our vision system. The idea is to utilize the fact that the world changes slowly, as compared to the frame rate. This makes it possible to predict the location where certain features will appear in the next frame. If most of these features are found close to their predicted positions, only small parts of the image need to be touched. The differences between the measured locations and the predictions can be used to update estimates of the parameters of a world model.

We use a 2D-model of the field, with the robots and the ball on it, as shown in Fig. 4. The model is matched sequentially to the video frames. For each model line, short orthogonal equidistant tracking lines form a tracking grid (Fig.5(a)).

The ends of each tracking line specify two color classes, according to the expected colors at both sides of the line. Each tracking line is mapped into the image, using the inverse camera function. This is done by mapping the endpoints of each transition line into the image and then reconnecting them with a line. Next, the pixels along the projection line are searched for the appropriate color

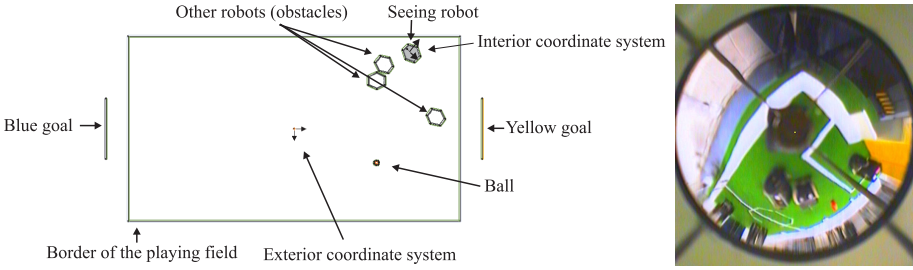


Fig. 4. Model seen from an external point of view.

transition. In Figure 5(b) detected transitions are marked with a black dot. It can be seen that the model does not fit precisely to the playing field in the image, due to a rotation of the robot. Sometimes false transitions may be found, e.g. at field lines. They need to be detected as outliers that must be ignored.

The next step is to calculate a 2-D rigid body transformation that brings the model in correspondence with the found transitions. In the case here, the model should be rotated slightly to the left. To determine the model's transformation, first a rotation and translation is calculated for each track grid independently. Then the results are combined to obtain a transformation for the whole model.

Repeating the above steps while perceiving a sequence of images yields the desired result: the pose of the field seen from the robots point of view is tracked and so the position and orientation of the robot is known by a simple coordinate transformation. Figure 6(a) shows the tracking while the robot rotates.

During initial search candidate positions are evaluated using the tracking mechanism. Given a position of the robot and the angle of a goal, the robot's orientation is calculated. The field can now be projected into the image and the ratio of found transitions can be used as quality measure for model fit. This ratio is also used during tracking to detect situations when the tracking fails and the initial search is needed to localize the robot again.

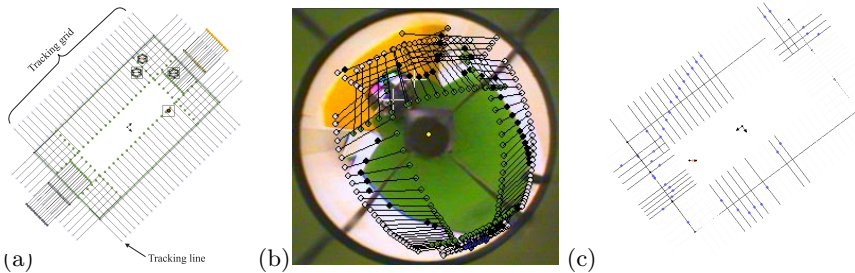


Fig. 5. Tracking of the field: (a) enhanced CAD model; (b) projected into the image with found transitions marked; (c) found transitions in local world coordinates.

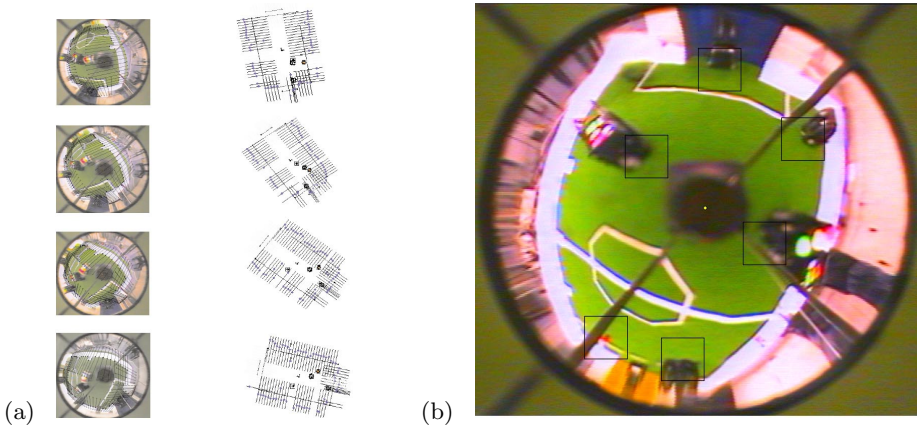


Fig. 6. Tracking of (a) the field while rotating; (b) ball and obstacles.

The system does not only track color edges, but also color blobs, such as the ball or obstacles. The blobs are only searched for in small square windows around their predicted positions, as shown in Fig. 6. If an object cannot be found within its rectangle, initial search is started to localize it again.

Conclusions

We implemented a local vision system for the F180 league that uses an omnidirectional camera. The system fits a world model to the input by finding and tracking color edges and blobs. It is able to process a full resolution, full frame rate video stream on a standard PC. For the production of the Seattle qualification video we used the extracted information about the world as input for behavior control. A single robot was able to drive behind the ball, and to kick it towards the goal, as well as to defend the goal. To control a team of robots, we plan to fuse multiple local views to a single global view. Currently, the image analysis is done on an external PC. As semiconductor technology advances, it will be possible to integrate a small computer on-board the robots.

References

1. Ryad Benosman, Sing Bing Kang (editors), *Panoramic Vision: Sensors, Theory, and Applications*, Springer, 2001.
2. Hough, *Machine Analysis of Bubble Chamber Pictures*, International Conference on High Energy Accelerators and Instrumentation, CERN, 1959.
3. ART, *F2000 team homepage*, <http://www.dis.uniroma1.it/~ART/>.
4. Golem, *F2000 team homepage*, <http://www.golemrobotics.com/gteam/gteam.html>.
5. OMNI, *F180 team homepage*, <http://robotics.me.es.osaka-u.ac.jp/OMNI>.
6. Bonarini, A., (2000) The Body, the Mind or the Eye, first? In M. Veloso, E. Pagello, A. Asada (Eds), *Robocup99 - Robot Soccer World Cup III*, Springer Verlag, Berlin.
7. Bonarini A., Aliverti, P., Lucioni, M. (2000). An omnidirectional sensor for fast tracking for mobile robots. *T. Instrumentation and Measurement*. 49(3). 509-512.