Sophisticated vision system for multiagent robots

Radim Bernatík

Department of measurrement and controll, FEI, VŠB – Technical university of Ostrava
17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
radim.bernatik@vsb.cz

Abstract. Proposed vision system is used for recognizing the robots positions during the game “Robotsoccer”. Each robot has small mark with colors. The trend is that the number of robots is increasing and therefore detection problems comes up (these problems are currently solved in this visual system). This system also has mesh-using transformation for wide usability. Mesh eliminates fish-eye, axis-unsymmetrical and also trapezoid distortion.

Keywords: color, coordinates, distortion, fish-eye, lens, mesh, multiagent, transformation, trapeze

1 Application scheme

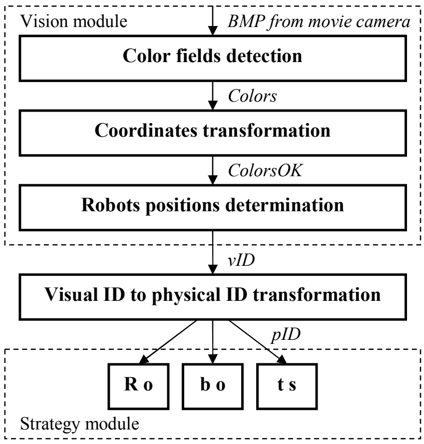

Figure 1. Vision system main scheme.

Robots on playground are marked by color. Each robot has two colors: team color, which is similar for all robots of one team, and robot color, which is different for each robot. Because the number of robots is still increasing, there are coming up problems with choosing suitable colors and with setting up the vision system to be able to right recognize simple robots. Implemented system is solving the problem by using same colors for each robot (two colors on one robot means two colors for all team robots). Vision system is splitted to four blocks (main scheme is shown on Figure 1).

1.1 Color fields detection

The first block recognizes single color fields of all robots on the playground regardless of their pertinence to particular teams. Each color has specified the number of their representations. The detection uses three-dimensional look up table. Indexes of this table are the color values (R, G, B) and value of table is color ID.

1.2 Coordinates transformation

In consequence of robots count increase the area of playground are also increased. A movie camera is used for sensing. The camera is placed above the playground and the height of placement can't be too high. Therefore the image is distorted by lenticular curvature in wide-angle objective. The second block recalculates coordination in image to physical coordination on real playground. The curvature is eliminated by mesh (explained below).

The coordinates of color fields are guaranteed. Each field is traced and their position predicted. In each frame there are all field position compared with equivalent predicted position. If difference is small, this position is guaranteed and therefore the actual and previous values of position are known. This property provides identification of each robot, although all robots have same colors.

1.3 Robots positions determination

The third block completes robots' positions. Robots' positions are calculated on base of color fields' positions. Difference between colors on one robot is constant. This difference is important for finding related colors of one robot. There is also used the system of guarantee, which inherits properties from previous block.

1.4 Visual ID to physical ID transformation

The last block makes adaptive identification of single robots. From visual system are transferred positions of single robots. Positions are guaranteed on the part of their order (*vID*). But each robot has his physical address (*pID*). The main task of this block is right determination of *vID* = *pID* equivalence. This just provides the system of positions guarantee. The system remembers previous *vID* = *pID* and if *vID* is guaranteed the *pID* is automatically guaranteed too. The *pID* is used for robots' identification in strategy module. Adaptive identification is done by special movement. If robot is identified, control of robot's movement is passed to strategy module.

2 Sophisticated adaptive mesh

2.1 Distortion

Not in all cases is this distortion uniform in each way. If this distortion is axis-unsymmetrical, the simple calculation can’t be used. It comes up the distortion, which is like trapeze. Trapezoid distortion is also perceptible when camera is placed outside the playground center and is aimed to the center of playground in small angle.
Figure 2 presents the normal physical state of playground, borders are symmetric and planar without distortion (can be seen by human eye). This is desired and optimal state but isn’t real. To achieve this state is only one way: place the camera very far above the playground and use small-angle objective. This situation cannot be reached, because too height room is needed.

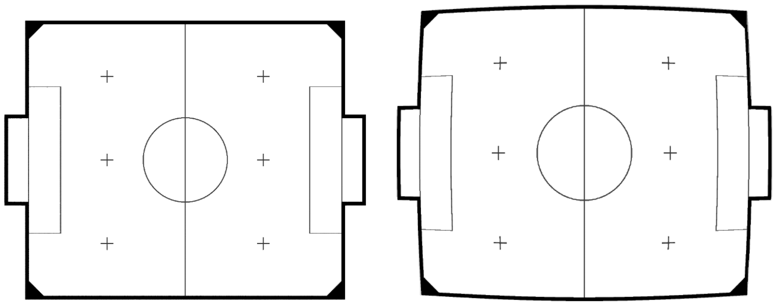

Figure 2. Optimal and distorted image.

Therefore camera is placed only 2 meters above playground. Playground is 2.2 meters wide, so the angle of view is about 50 degrees. Objective with this wide-angle always have fish-eye distortion. Typical image from camera with wide-angle objective is shown in Figure 2.

2.2 Mesh

For these cases was developed transformation of coordination, which uses adaptive mesh. Adaptation of mesh is done by placing their points on specific places in image. User places in sum 9 points of mesh to the image. Four points are on corners of playground, four points are on center of playground edges and one point is placed on center of playground.
Because main points of mesh can be placed anywhere on the image this transformation can eliminate many distortion, which occurs during normal camera-sensing process. There are a few of them:

- **Fish-eye distortion** (symmetrical or unsymmetrical). It comes up from lens.
- **Rotation** of image. Small rotation can be eliminated.
- **Zoom**. This mesh can be also used for zoomed image.

* **Trapezoid distortion** (also special case of axis-unsymmetrical fish-eye distortion from lens). It comes up from lens (incorrectly made lens) or from bad placed camera (camera is aimed on scene with small angle).
* **Excentric placement** of camera. During Robot Soccer game are 2 cameras above the playground and only one of them can be placed above the playground center (but in most of cases no camera is placed directly above the playground center).

Advantage of mesh is that the only one algorithm is needed. All distortions are eliminated in one algorithm by one run of this algorithm.

2.3 Transformation using mesh

With reference to application requirements of real-time control the image isn't whole transformed, but is analyzed in camera image state. After the image analysis the coordinates of robots and ball are transformed from image coordinates to real coordinates (millimeters on playground).

Coordinates definition

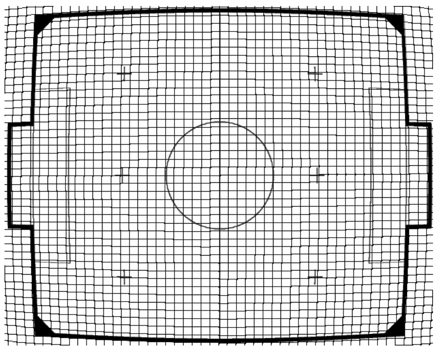

Figure 3. Mesh over playground.

Transformation mesh isn't defined by only 9 points, but by 55 points in both axes, it is 3025 points. In Figure 3 is shown most of points in mesh applied over the playground distorted by fish-eye distortion. Each point has two sets of coordinates. One is coordinates in image pixels and second set is in millimeters on playground. Playground coordinates calculation is really simple, because single points are away of their self the same length. The mesh is well-balanced on the playground (playground isn't distorted in real). More problematic is calculation of image pixels coordinates (image from camera is distorted). This calculation is shown below.

Coordinates calculation
a) Single points on edges (always 3 points) are inset by circle. These 3 points are placed by user. On this circle are calculated and placed always 55 points. These points

lie on the specific part of circle, part between points. By this procedure are calculated points on playground edges and on curves, which split the playground in horizontal and vertical way.

b) There comes up set of 3 vertical and 3 horizontal curves of points. For calculation other points inside the mesh are used these 6 curves. From curves are picked 3 corresponding points. 1st point from left border curve, 2nd point from middle vertical curve and 3rd point from right border curve, points are inset by horizontal circle. For vertical circle are choused points from top border circle, middle horizontal circle and form bottom border circle. Also these points are inset by circle.

c) Now, there are picked 3 and 3 points from both axes (if you like horizontal and vertical circle). Resulting point is calculated as point of intersection of these 2 circles. By this procedure are calculated all inside points of mesh and mesh is prepared for using in transformation algorithm.

Transformation algorithm

a) Transformation algorithm determines from input pixel image coordinates the rectangle, in which this coordinate lies. The way of finding rectangle is shown in Figure 4. Algorithm is optimized for high speed, so in 1 step is jumping between rectangles in each way (x and y). Jumping is allowed in both directions (left or right and up or down). For increasing the speed the algorithm remembers last position of object and uses it as start point in next frame. So if the robot or ball doesn't move, rectangle isn't changed and algorithm found the right rectangle in one step. If object is moving with low speed (most cases) the new rectangle is almost neighborhood of previous. This type of finding the right rectangle is needed because the image (and mesh) is distorted and this distortion is non-linear.

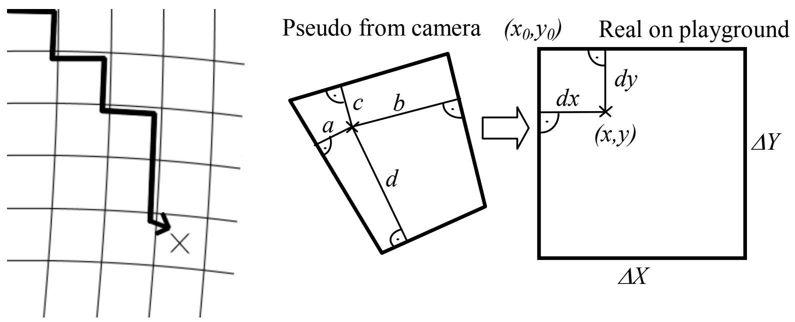

Figure 4. Rectangle finding and transformation.

b) After location of corresponding square it is counted precise offset of point in square (relative offset of square edges and corners). With using of pseudo coordinates (in image from camera) is calculated the real coordinate on playground. The main idea is that relative offset of edges and corners are the same in image and real coordinates.

Calculation is developed for right function in case if the square or whole mesh is slightly rotated (showed in Figure 4). From the main idea comes out equations for additions *dx* and *dy*.

$$x = x_0 + dx$$

$$\text{ri0}\; y = y_0 + dy \qquad (1)$$

$$dx = \frac{a \cdot \Delta Y}{a + b} \qquad dy = \frac{c \cdot \Delta Y}{c + d} \qquad (2)$$

3 Conclusion

Vision system is implemented in robotsoccer application, but its utilization is independent on this application. It was developed as robust library module (Windows32 DLL). Number of colors and teams isn't limited to two, which occurs in robotsoccer. Proposed solution provides right function of robot soccer application also in difficult conditions, when is impossible to place camera to the center of playground, or if the camera can't be placed high enough and must be aimed in small angle. The main idea for transformation isn't fine, but if mesh is dense enough error of object position is relative small.

This research was supervised by Doc. Ing. Vilém Srovnal CSc., Department of Measurement and Control, VŠB-Technical University of Ostrava. This research was supported by the **FRVŠ grant No. 1735-G1-a (2003)**.

4 Publications

1. Bernatík, R., Kovář, P., Horák, B.: Quick image recognize algorithms. International workshop R-MAS 2001, Ostrava, Czech Republic, 2001.
2. Bernatík, R., Kovář, P., Horák, B.: Advanced quick image recognize algorithms. International workshop EC of robot soccer 2002, Vienna, Austria, 2002.
3. Penhaker, M., Ožana, Š., Bernatík, R., Štula, T.: Využití MATLABu ve výuce Signálů a Soustav a v biomedicínckém inženýrství na Katedře měřicí a řídicí techniky na VŠB - TU Ostrava. Sborník příspěvků 10. ročníku konference MATLAB 2002, Praha, 2002, vol. II, 419-425, HUMUSOFT s.r.o., ISBN 80-7080-500-5.
4. Bernatík, R., Horák, B., Srovnal, V., Štula, T.: Mobile Robots Control Using of Multi-agent Technology, Agent days 2002, Belfort, France, 2002.
5. Bernatík, R., Horák, B.: Sensing and processing of phonocardiographic signal by PC sound card, IFAC workshop PDS 2003, Ostrava, Czech Republic, 2003, ISBN: 0 08 044130 0
6. Bernatík, R.: Vision systém for determining and controlling multiagent robots system, POSTER 2003, Prague, Czech Republic, 2003.
7. Horák, B., Bernatík, R., Štula, T.: Transformation of image coordinates by using of adaptive mesh, FIRA Robot World Congress 2003, Vienna, Austria, 2003