Obtaining the distance map for perspective vision systems

António J. R. Neves, Daniel A. Martins and Armando J. Pinho
Signal Processing Lab, IEETA / DETI
University of Aveiro, 3810–193 Aveiro, Portugal
an@ua.pt,dam@ua.pt,ap@ua.pt

The distance map is widely used in robotic applications which include robotic vision systems as sensorial elements. This mapping between the pixels of the acquired image and real coordinates allows the vision system to characterize the objects of interest, like its position or size. In this paper, we describe a tool to create the distance map for the perspective vision system of the RoboCup Middle Size League (MSL) soccer team CAMBADA (Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture) from the University of Aveiro. With this tool, the user input is reduced to a few parameters that can be easily measured or obtained by the intrinsic parameters of the digital camera. This avoids measures of angles that could otherwise be difficult for the user to acquire and seriously corrupt the distance map if not accurate. The distance mapping approach proposed in this paper assumes that all the objects found in the image are in one plane (the ground plane), allowing to map the real world distances in the image pixels. From a ray-trace like perspective we solve the distance mapping problem considering some simplifications without loss of usability. To simplify the problem, it was assumed that the lens and the CCD from the camera were centered and in parallel planes, so the center of the image would be the center of the CCD. Two more approximations were made during this approach to the problem. First, the pan angle was ignored, so the camera is considered to be pointing towards the front of the robot and second, the rotation of the camera was also ignored, considering it horizontally aligned. Experimental results are provided showing the effectiveness of the proposed approach.

1 Introduction

The Middle Size League (MSL) competition of RoboCup, one of the biggest worldwide robotic competitions, is a standard real-world test for autonomous multi-robot systems. Being a color-coded environment, recognizing colored objects such as the orange ball, the black obstacles, the green field and the white lines are a basic and yet important ability for robots (see Fig. 1).

Figure 1: An example of the CAMBADA team playing a game of the MSL League .

One problem domain in RoboCup is the field of Computer Vision, responsible for providing basic information that is needed for calculating the behavior of the robots. Catadioptric vision systems (often named omnidirectional vision systems) have captured much interest in the last years, because they allow a robot to see in all directions at the same time without having to move itself or its camera. However, due to the last changes in the MSL rules, the playing field became larger, bringing some problems to the omnidirectional vision systems, particularly regarding the detection of objects at large distances.

The vision systems are used to provide the robotic agents with information about their surroundings, and therefore the position of the objects founded in the environment. To successfully complete this task, we must create a relation between the pixels in the image (acquired by the robotic vision system) and a position in the real world. The "Distance Mapping" is a technique widely used which allows to create a map between real world positions and the pixels of the image. This technique assumes that all the objects found in the image are in one plane, usually the ground plane, in order to retrieve such information from the mapping process.

Note that in MSL, the perspective vision systems are a common sensorial element due to its overall simplicity and good results. Despite the central place occupied by the catadioptric vision systems (Lima, Bonarini, Machado, Marchese, Marques, Ribeiro, and Sorrenti 2001; Lu, Zheng, Liu, and Wang 2008; Lunenburg and v.d. Ven 2008; Hafner, Lange, Lauer, and Riedmiller 2008; Zweigle, Kappeler, Ruhr, Haussermann, Lafrenz, Schreiber, Tamke, Rajaie, Burla, Schanz, and Levi 2007; Azevedo, Lau, Corrente, Neves, Cunha, Santos, Pereira, Almeida, Lopes, Pedreiras, Vieira, Martins, Figueiredo, Silva, Filipe, and Pinheiro 2008; Neves, Corrente, and Pinho 2007), the perspective vision is very useful to provide information about distant objects (Neves, Martins, and Pinho 2008). Besides greater distance scope, the perspective systems introduce less distortion in the image when compared with the catadioptric systems. This is mainly due to the mirror distortion inherent to the catadioptric systems.

With this strong dependence from the vision systems, grows the necessity of automatic solutions for vision systems calibration (Neves, B. Cunha, and Pinheiro 2009).

In this paper, we describe a tool to automatically create the distance map for the perspective vision system of the MSL team CAMBADA from University of Aveiro (called after this point of *PerspectiveMapCalib* tool). With the proposed tool, the user input is reduced to simple measures that can be easily acquired. This avoids measures of angles that could be difficult for the user to acquire and seriously corrupt the distance map if not accurate. We solve the distance mapping problem from a ray-tracing approach and considering some simplifications without loss of usability. The detailed description of the tool is followed by a field test of the tool using robots from team CAMBADA. The results show that this tool can be used to provide a fast calibration of the distance map without loss of accuracy.

2 Distance Mapping

The use of a distance map image is a method commonly used in the RoboCup domain (Cunha, Azevedo, Lau, and Almeida 2007). This method assumes that all the objects found in the image are in one plane (the ground plane), allowing to map the distances in the image with only one camera. To simplify the problem, it was assumed that the lens and the CCD from the camera were centered and in parallel planes, so the center of the image would be the center of the CCD. Two more approximations were made during this approach to the problem. First, the θ rotation was ignored, so the camera is considered to be pointing towards the front of the robot and second, the φ rotation of the camera was also ignored, considering it horizontally aligned (see Fig. 2).

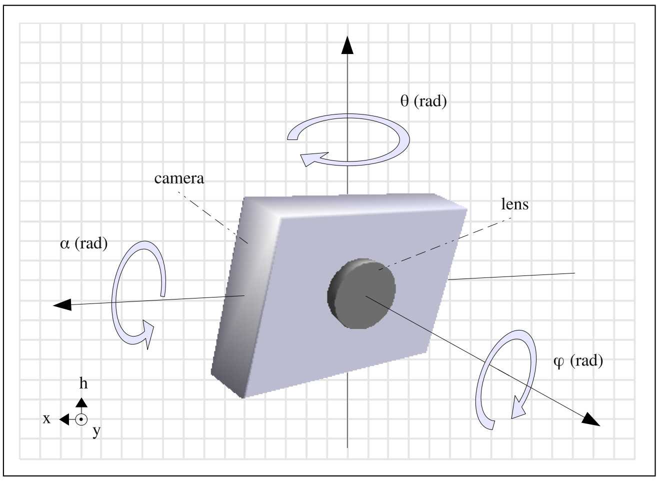


Figure 2: Schematic representation of the perspective camera with its rotation axis.

To summarize, the mathematical model of this system needs information about the camera, the robot and the field. Some of this information, such as the h_{offset} and r_{offset} represented in Fig. 3, can be measured directly with good precision. Other kinds of information, like the $pixel\ height$, $pixel\ width$ and the $focal\ length$ represented in Fig. 5, may be obtained from the camera datasheet, given by the manufacturers. However, it is still missing necessary information, namely the α_{offset} (see Fig. 3). This information can be measured, but, in one hand, measures like the h_{offset} and r_{offset} may be used from one robot to another without compromising the resulting map image and, on the other hand, distance map results are very sensible to α_{offset} variations and it is very time consuming to obtain good measures of this variable. One way to overcome this time consuming process (of measuring the α_{offset} in each robot) is to include an automatic measuring process of this parameter into the *PerspectiveMapCalib* tool.

In Fig. 3 and Fig. 4 it is presented a side and top schematic view of the perspective vision system. A schematic view of a detail over the perspective system is presented in Fig. 5. Follows a short explanation of the measures shown in these figures:

- h_{offset} - distance from the camera to the ground;

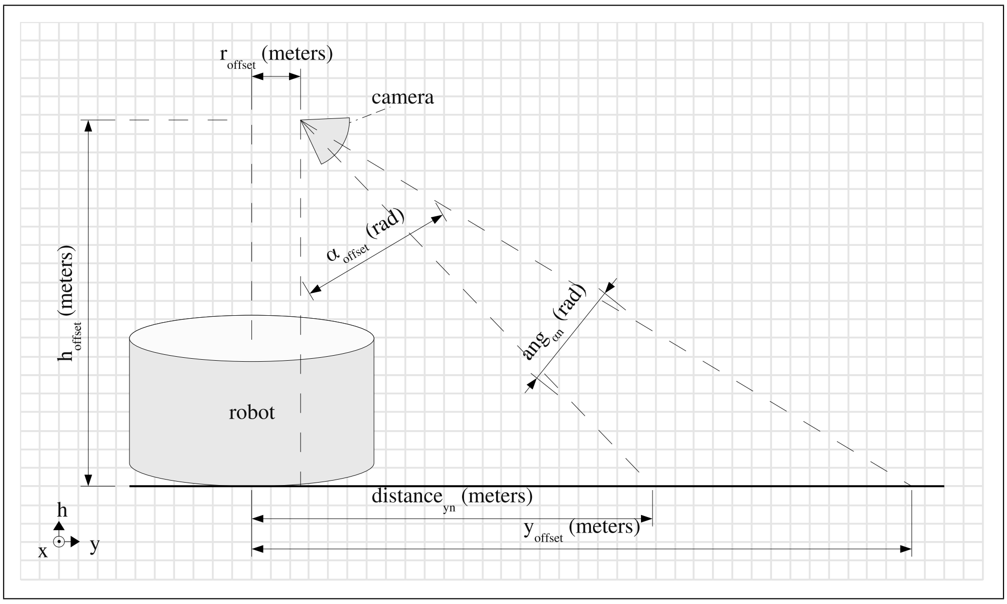


Figure 3: Robot and its perspective system schematic side view.

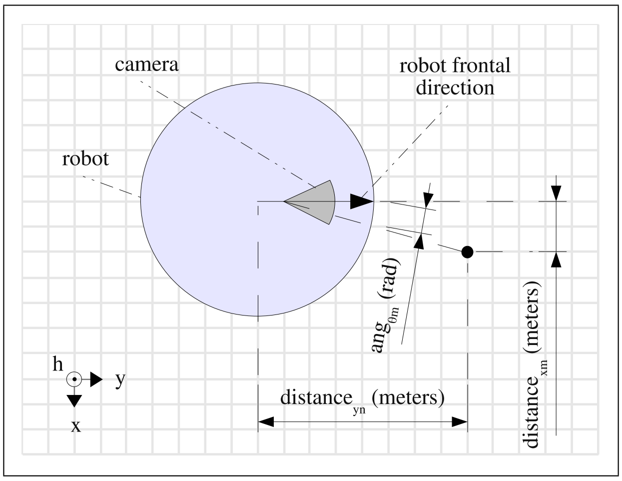


Figure 4: Schematic top-down view of the robot and perspective system.

- r_{offset} - radial distance from the camera to the robot center;
- α_{offset} - angular offset of the camera along α axis;
- y_{offset} - distance from the center of the robot to the point in the center of the image projected on the ground;
- $ang_{\alpha\text{n}}$ - angle measured from the α_{offset} along α axis, relative to $pixel_{\text{n}}$;
- $ang_{\theta\text{m}}$ - angle measured from the robot front along θ axis, relative to $pixel_{\text{m}}$;
- $distance_{\text{yn}}$ - distance from the center of the robot to the $pixel_{\text{n}}$, projected on the ground;
- $distance_{\text{xm}}$ - distance from the center of the robot to the $pixel_{\text{n}}$, projected on the ground;
- $focal\ length$ - distance between lens and CCD;
- $pixel_{\text{n}}$ - number of pixels (n) along a CCD column;
- $pixel_{\text{m}}$ - number of pixels (m) along a CCD row.

Using an image taken from a known position (over the goal line, aligned with the kick-off mark, as shown in Fig. 6), the tool acquires some samples based on image analysis and some user input data. The tool highlights the white lines found, asking the user for the $distance_{\text{yn}}$ between the robot center and the white line being processed. The search for white lines is conducted over the central row of the image, because the θ_{offset} is being ignored.

Figure 6: Example of an image processed by the tool *PerspectiveMapCalib*. Magenta spots in the image show the white lines found. Orange circles highlight the spot being processed.

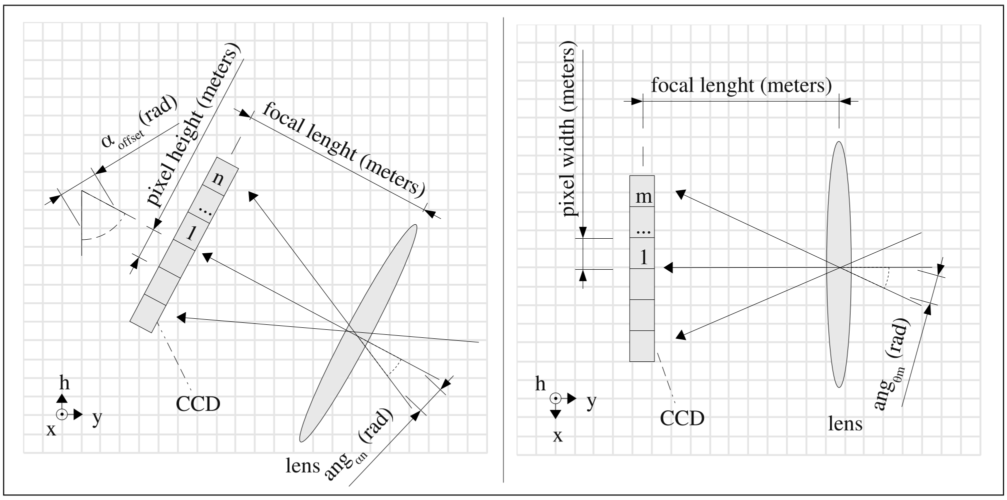


Figure 5: Schematic view of a detail over the lens, CCD and focus point from the perspective system. On the left, a side view. On the right, a top-down view.

The following equation shows the relation between the y_{offset} and the angle α_{offset}, both referred in Fig. 3,

$$\alpha_{\text{offset}} = \arctan\left(\frac{y_{\text{offset}} - r_{\text{offset}}}{h_{\text{offset}}}\right). \qquad (1)$$

From (1), the relation is generalized to an angle $ang_{\alpha\text{n}}$ centered in α_{offset},

$$ang_{\alpha\text{n}} = \alpha_{\text{offset}} - \arctan\left(\frac{distance_{\text{yn}} - r_{\text{offset}}}{h_{\text{offset}}}\right). \qquad (2)$$

Attending to Fig. 5 (on the left), using simple trigonometric rules the following equation can be obtained, relating a generic angle ang_{n} centered in α_{offset} and a pixel along a vertical column in the CCD,

$$pixel_{\text{n}} = \frac{\tan(ang_{\alpha\text{n}}) \times focal\ length}{pixel\ height}. \qquad (3)$$

Substituting (2) in (3) results in,

$$pixel_{\text{n}} = \tan\left[\alpha_{\text{offset}} - \arctan\left(\frac{distance_{\text{yn}} - r_{\text{offset}}}{h_{\text{offset}}}\right)\right] \times \frac{focal\ length}{pixel\ height}. \qquad (4)$$

Manipulating (4), it is possible to isolate the α_{offset} in the first member, i.e.,

$$\alpha_{\text{offset}} = \arctan\left(\frac{pixel_{\text{n}} \times pixel\ height}{focal\ length}\right) + \arctan\left(\frac{distance_{\text{yn}} - r_{\text{offset}}}{h_{\text{offset}}}\right). \qquad (5)$$

With (5) and the samples (pairs of values $pixel_{\text{n}}$ and $distance_{\text{yn}}$ introduced by the user) previously acquired, the value α_{offset} can be found. To achieve a better result, the tool *PerspectiveMapCalib* uses a mean filter with, at least, three calculated α_{offset} angles to obtain the final result.

Using the previously discovered α_{offset} angle and with some manipulation of (4), the distance corresponding to each pixel can be found using

$$distance_{\text{yn}} = r_{\text{offset}} + h_{\text{offset}} \times \tan\left[\alpha_{\text{offset}} - \arctan\left(\frac{pixel_{\text{n}} \times pixel\ height}{focal\ length}\right)\right]. \qquad (6)$$

Through (6) it is possible to obtain the real $distance_{\text{yn}}$ of a pixel projected in the ground along the y axis.

Based on Fig. 4, the following equation can be deduced, creating a relation between a generic angle $ang_{\theta\text{m}}$ and a $distance_{\text{xm}}$,

$$ang_{\theta\text{m}} = \arctan\left(\frac{distance_{\text{xm}}}{distance_{\text{yn}}}\right). \qquad (7)$$

Now, relating $ang_{\theta\text{m}}$ with $pixel_{\text{m}}$ along the x axis we obtain

$$pixel_{\text{m}} = \frac{\tan(ang_{\theta\text{m}}) \times focal\ length}{pixel\ width}. \qquad (8)$$

Using (7) to substitute in (8) and manipulating it, results in a relation between a $pixel_{\text{m}}$ and a $distance_{\text{xm}}$, both along the x axis, i.e.,

$$distance_{\text{xm}} = \arctan\left(\frac{pixel_{\text{m}} \times pixel\ width}{focal\ length}\right) \times \qquad (9)$$
$$distance_{\text{yn}}.$$

Summarizing, making use of (5), and some user input, its possible to obtain the α_{offset}. This permits the creation of the *distance map image* using (9) and (6) providing the x and y coordinates associated to each $pixel_{(\text{n,m})}$.

3 Results

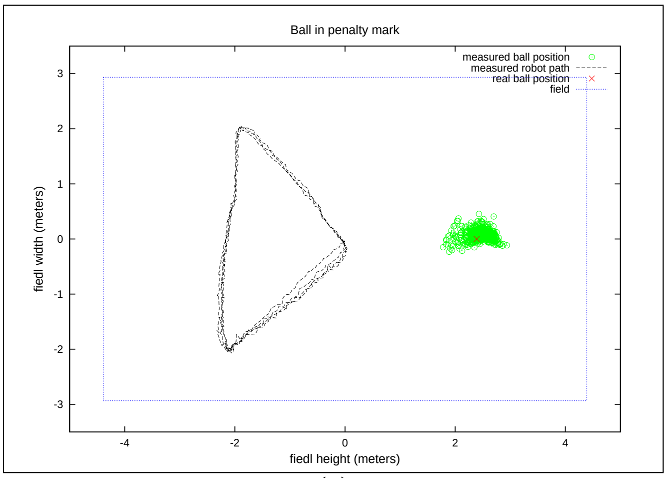
To measure the reliability of this tool, tests were made analyzing the resulting *distance map image*. To do so, the robot was moved along a predefined path through the game field leaving the ball in a known location. The ball position given by the robot is then compared with the real position of the ball. The results in this test may be affected by the errors of the localization algorithm and the robot bumps while moving, but they provide a realistic view of the problem, recreating the RoboCup environment.

In Fig. 7 it is possible to see the robot path along the field and the measured ball position. In Table 1, it is presented an analysis of the data measured during the test. Already visible in Fig 7, the average position of the ball is near the real position, being this result confirmed in Table 1.

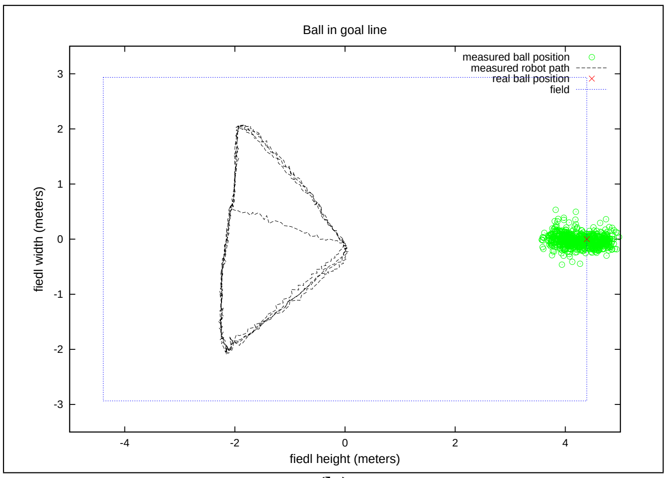
Furthermore, the standard deviation of the measured ball position is low compared with the distance between the ball and the robot, which is directly related to the quality of the distance map used. Being these measures affected by external noise as referred above, this result is more than acceptable for its purpose, that is, to detect the ball at long distances.

4 Conclusions

In this paper, we proposed an efficient tool to create the distance map for the perspective vision system of a RoboCup Middle Size League soccer team.



(a)



(b)

Figure 7: Experimental results obtained by the proposed perspective vision system using the *PerspectiveMapCalib* tool. In (a) the ball was positioned in the center of the penalty mark. In (b) the ball was positioned in the center of the goal line. The robot has performed a defined trajectory and the position of the ball was registered. Both axis in the graphics are in meters.

Experiment	Real Position	Measures	
		Average	Std
Penalty	(2.39, 0.00)	(2.47, 0.07)	(0.19, 0.09)
Goal Line	(4.39, 0.0)	(4.27, -0.03)	(0.32, 0.11)

Table 1: Some measures obtained for the experiments presented in Fig. 7. All the measures are in meters. The column under the label "Average" means the average position where the ball was detected by the robot, whereas the column under the label "Std" means the the standard deviation of the measured ball position.

With this tool, the user input is reduced to a few parameters that can be easily measured or obtained by the intrinsic parameters of the digital camera. This avoids measures of angles that could be difficult to acquire and seriously corrupt the distance map if not accurate. We solved the distance mapping problem from a ray-tracing approach and considering some simplifications without loss of usability.

The experimental results obtained in a real test situation, show that the mapping obtained by the proposed approach is very accurate, allowing its use in real competition. Moreover, the prosed algorithms have been used by the team in the last competitions. The CAMBADA team distinctively achieve the 1st place in the RoboCup 2008 and in the Portuguese robotics open ROBOTICA 2009, ROBOTICA 2008 and ROBOTICA 2007.

REFERENCES

Azevedo, J. L., N. Lau, G. Corrente, A. Neves, M. B. Cunha, F. Santos, A. Pereira, L. Almeida, L. S. Lopes, P. Pedreiras, J. Vieira, D. A. Martins, N. Figueiredo, J. Silva, N. Filipe, and I. Pinheiro (2008). CAMBADA'2008: Team description paper. Technical report, Universidade de Aveiro, Portugal.

Cunha, B., J. L. Azevedo, N. Lau, and L. Almeida (2007). Obtaining the inverse distance map from a non-SVP hyperbolic catadioptric robotic vision system. In *Proc. of the RoboCup 2007*, Atlanta, USA.

Hafner, R., S. Lange, M. Lauer, and M. Riedmiller (2008). Brainstormers tribots team description. Technical report, Institute of Computer Science, Institute of Cognitive Science. University of Osnabru, Germany.

Lima, P., A. Bonarini, C. Machado, F. Marchese, C. Marques, F. Ribeiro, and D. Sorrenti (2001). Omni-directional catadioptric vision for soccer robots. *Robotics and Autonomous Systems 36*(2-3), 87–102.

Lu, H., Z. Zheng, F. Liu, and X. Wang (2008, June). A robust object recognition method for soccer robots. In *Proc. of the 7th World Congress on Intelligent Control and Automation*, Chongqing, China.

Lunenburg, J. J. M. and G. v.d. Ven (2008). Tech united team description. Technical report, Control Systems Technology Group. Eindhoven University of Technology, Netherlands.

Neves, A. J. R., A. J. P. B. Cunha, and I. Pinheiro (2009, June). Autonomous configuration of parameters in robotic digital cameras. In *Proc. of the 4th Iberian Conference on Pattern Recognition and Image Analysis, ibPRIA 2009*, Pvoa de Varzim, Portugal.

Neves, A. J. R., G. Corrente, and A. J. Pinho (2007). An omnidirectional vision system for soccer robots. In *Progress in Artificial Intelligence*, Volume 4874 of *Lecture Notes in Artificial Inteligence*, pp. 499–507. Springer.

Neves, A. J. R., D. A. Martins, and A. J. Pinho (2008, April). A hybrid vision system for soccer robots using radial search lines. In *Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBOTICA'2008*, Aveiro, Portugal, pp. 51–55.

Zweigle, O., U. P. Kappeler, T. Ruhr, K. Haussermann, R. Lafrenz, F. Schreiber, A. Tamke, H. Rajaie, A. Burla, M. Schanz, and P. Levi (2007). Cops stuttgart team description 2007. Technical report, IPVS. University of Stuttgart, Germany.