Improving Vision-Based Distance Measurements Using Reference Objects

Matthias Jüngel, Heinrich Mellmann, and Michael Spranger

Humboldt-Universität zu Berlin, Künstliche Intelligenz
Unter den Linden 6, 10099 Berlin, Germany
{juengel,mellmann,spranger}@informatik.hu-berlin.de
http://www.aiboteamhumboldt.com/

Abstract. Robots perceiving their environment using cameras usually need a good representation of how the camera is aligned to the body and how the camera is rotated relative to the ground. This is especially important for bearing-based distance measurements. In this paper we show how to use reference objects to improve vision-based distance measurements to objects of unknown size. Several methods for different kinds of reference objects are introduced. These are objects of known size (like a ball), objects extending over the horizon (like goals and beacons), and objects with known shape on the ground (like field lines). We give a detailed description how to determine the rotation of the robot's camera relative to the ground, provide an error-estimation for all methods and describe the experiments we performed on an Aibo robot.

Keywords: RoboCup, humanoid robots, Aibo, camera matrix, reference objects.

1 Introduction

A main task in robotic vision is to determine the spatial relations between the robot and the objects that surround it. Usually the robot needs to know the angle and the distance to certain objects in order to localize, navigate or do some high-level planning. To determine the distance to an object is easy when the size of the object and the focal length of the camera are known. To determine the distance to an object of unknown size is possible using the knowledge about the height of the camera and the bearing to the point where the object meets the ground. This bearing is given by the position of the object in the image and the known orientation of the camera relative to the ground. Unfortunately this orientation is not known in a lot of cases. The calculation of the kinematic chain of a legged robot from the ground to the camera is usually difficult as the exact contact points of the robot and the ground are hard to determine. Additionally inaccuracies in the joint angle sensors sum up the longer the kinematic chain is. But also for wheeled robots the orientation of the camera relative to the ground can be unknown, especially when there is a suspension for the wheels. In this paper we show how to determine the orientation of the camera

U. Visser et al. (Eds.): RoboCup 2007, LNAI 5001, pp. 89–100, 2008.
© Springer-Verlag Berlin Heidelberg 2008

using reference objects in the image and how to calculate the distance to objects of unknown size. This work was inspired by our experience in RoboCup where using the field lines to localize a Sony Aibo was inaccurate due to large errors in the orientation and position of the camera which are calculated based on the sensor readings of the joint angles of the robot and assumptions on the contact points of the legs with the ground.

1.1 Related Work

There has been extensive work on the calibration of camera parameters. Typically authors try to infer intrinisic and extrinsic parameters of cameras using specially crafted calibration objects. A lot of work has been put in to reduce the complexity of this objects, i.e. their dimensionality or rigidness of pattern [1,2] or even allow completely other objects for the parameter estimation [3]. RoboCup teams have developed mechanisms to reduce the calibration time after transport of robots [4] or to calibrate ceiling cameras [5]. A lot of these methods involve off-line optimization of the estimated parameters regarding projection errors. In contrast to these methods our approach focuses on determining the camera pose relative to the ground during the operation of the robot. While intrinsic parameters do not change during operation, the extrinsic parameters of the camera are usually hard to determine using proprioception in a highly dynamic environment like RoboCup. We describe how information from the camera images can be used to determine the orientation of the camera and how additional information from the joint sensors can be incorporated.

1.2 Outline

This paper is divided into several parts. In section 2 we motivate our work by giving an error estimation for the bearing based distance measurement approach. In section 3 we describe several methods that determine the camera matrix by means of visual information in order to determine distances to other objects. In section 4 we examine the robustness of these methods concerning errors. Section 5 presents the results of some experiments which were conducted with an AIBO.

2 Motivation

A simplified version of the bearing based distance estimation approach of objects can be seen in figure 1. The model was used to estimate the significance of any correction approach in advance. From this simple mathematical model conclusions about the influence of measurement errors of the rotation angle φ and the estimated height h_{camera} on the calculated distance d_{object} were drawn.

The basic bearing based distance approach depicted in figure 1 calculates d_{object} from known h_{camera}, h_{object} and φ. From

$$d = \tan(\varphi) \cdot h_{camera} \quad \text{and} \quad d_{rest} = \tan(\varphi) \cdot h_{object}$$

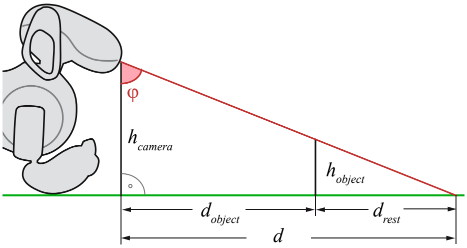

Fig. 1. Simple bearing based distance estimation model

follows with $d_{object} = d - d_{rest}$ that

$$d_{object} = \tan(\varphi) \cdot (h_{camera} - h_{object})$$

With known h_{camera} and h_{object}, d_{object} can be seen as a function depending on φ only, i.e. $d_{object} = d_{object}(\varphi)$. It can be immediately seen that it is also possible to infer the correct bearing φ from known h_{camera}, h_{object} and d_{object}. This simple model is only valid when $h_{camera} > h_{object}$ and $\varphi < \frac{\pi}{2}$. It allows to show the effect of estimation errors of φ on the estimated distance d_{object} of an object of height h_{object}. For an ex ante study suitable values for h_{camera} and h_{object} where chosen from the context of *RoboCup*. The error d_{error} is calculated by

$$d_{error}(\Delta\varphi) = |d_{object}(\varphi + \Delta\varphi) - d_{object}(\varphi)|$$

From the formulas provided it can be seen that even small changes of φ can result in big errors for the estimated distance d_{object}, which is shown in figure 2a) for fixed h_{camera} and h_{object}. For positive $\Delta\varphi$ the error is rising exponentially. Figure 2b) illustrates that this error rises with the growing correct distance of the object.

3 Using Reference Objects for Improving Distance Measurements

A lot of objects in the environment of a robot can be used as reference objects for distance calculation. In this paper we focus on the calculation of distances based on the height of the observing camera and its direction of view. As shown in section 2 this method is very prone to errors in the angle between the optical axis of the camera and the ground. We show several methods to estimate the position and orientation of the camera relative to the ground using different classes of objects:

- objects with known size (e.g. the ball)
- objects with known height, higher than the camera of the robot (e.g goals and beacons)
- objects with known outline on the ground (e.g. goals and field lines)

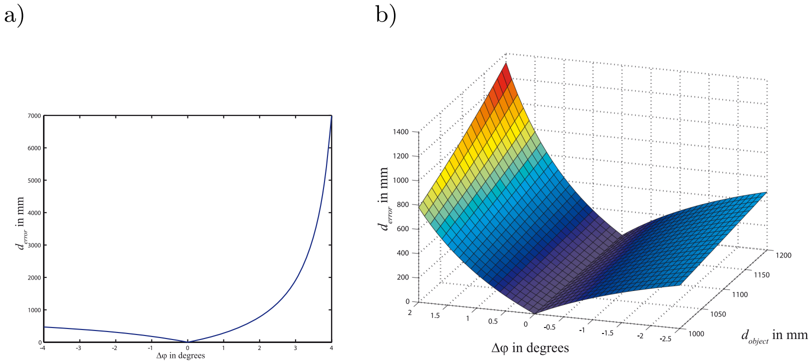

Fig. 2. Bearing based distance estimation error for fixed $h_{camera} = 160mm$ (which is a suitable camera height approximation for the Sony Aibo Robot) and an object height $h_{object} = 90mm$(height of the *RoboCup* ball in the 4-legged-league) a) Shows the effect of variations of φ (calculated from correct distance of $d_{object} = 1000mm$). Please note that the error gets as big as $7000mm$ for a variation of φ by $4degrees$. b) Shows the same effect as a) in a range for the object distance d_{object} from $1000mm$ to $1200mm$. For bigger distances the error rises dramatically.

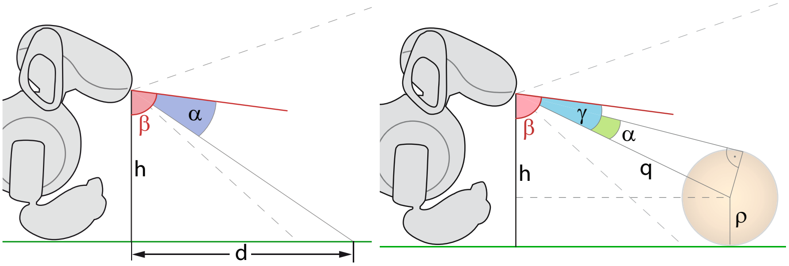

Fig. 3. Relation between the tilt of the camera and the ball used as reference object

All examples given in brackets are usually to be seen during a typical RoboCup game. Thus in almost every image at least one reference object can be used. The following subsections describe the different methods for all classes of reference objects. Given that the camera is not rotated on the optical axis we can limit our following considerations to a two-dimensional model, as shown in figure 3 (left).

3.1 Objects of Known Size

An easy approach in order to determine the camera tilt is to consider reference objects, whose distance can be determined based on their size. If the distance to a point is given, the tilt can be calculated as follows:

$$\beta = \arccos\left(\frac{q}{h}\right) - \alpha.$$

This formula can be deduced from figure 3 (left).

Figure 3 (right) illustrates the relation between the camera tilt and the ball as reference object[1]. Here, two simple formulas can be deduced as follows:

$$\frac{\rho}{q} = \sin(\alpha) \quad \text{and} \quad \frac{h-\rho}{q} = \cos(\beta - \gamma)$$

it can be deduced:

$$\beta = \arccos\left(\frac{h-\rho}{\rho} \cdot \sin(\alpha)\right) + \gamma.$$

This formula allows us to calculate the camera tilt using only the size of the ball without the need of any other sensor information.

3.2 Objects with Known Shape on Ground

If the height and the tilt of the camera are known, the image captured by the camera can be projected to the ground. If the used camera tilt corresponds with the real tilt, the outline of ground-based objects should appear without distortion in this projection. Should there be distortions (e.g. there is not a right angle between the field lines), this is a hint on the fact that the used tilt is incorrect. Thus it is an obvious idea to determine the camera tilt so that the projected field lines are perpendicular to each other.

This idea can be formulated as follows. Let p_1, p_2 and p_3 be the defining points of a corner in the image, p_1 being the vertex. The points are projected to the ground plane by means of the camera matrix $M(\alpha)$, α being the camera tilt. The resulting points are denoted $P_i(\alpha)$. For the angle φ, which is defined by these points, it holds:

$$\cos\varphi = \frac{\langle P_1(\alpha) - P_2(\alpha), P_1(\alpha) - P_3(\alpha)\rangle}{||P_1(\alpha) - P_2(\alpha)|| \cdot ||P_1(\alpha) - P_3(\alpha)||}.$$

However, it is known that $\varphi = \frac{\pi}{2}$ and hence $\cos\varphi = 0$, so that the formula for α is the following:

$$\langle P_1(\alpha) - P_2(\alpha), P_1(\alpha) - P_3(\alpha)\rangle = 0.$$

In general, this equation has an infinite number of solutions. However, in specific cases, as e.g. in the case of AIBO, there is often only one admissible solution due to the limits of the joints. By means of standard methods as Gradient Descent, this solution can be easily found.

This method works best if the corner is viewed on from the outside or the inside. However, if the robot is situated on one of the defining lines, the angle is not distorted by the wrong camera tilt any more and the method fails.

3.3 Tilt Correction Using Objects Higher than Eye-Level

The examination of the horizon yields another approach for the correction of the camera matrix. In many cases the horizon can be measured by means of objects

[1] The advantage of taking the ball as reference object is that it is easy to determine its size, as the ball looks equal from every direction. Furthermore, it can be seen on numerous images, being the central object of the game.

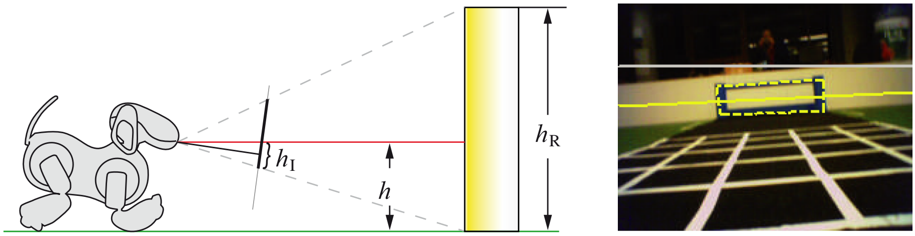

Fig. 4. (left) A landmark is projected on the image plane. The knowledge of the real height of landmark can be used to determine the height of the horizon in the image. (right) An image captured by the robot, containing the recognized goal and the calculated horizon.

that are higher than the focal point of the camera. For example, if the robot sees a landmark with a known real height h_R and if its height in the image h_I is known as well, it is easy to determine the height of the horizon in the image, as it equals $\frac{h \cdot h_I}{h_R}$, as can be seen in the figure 4. By definition, the line of the horizon goes through the center of the image, if and only if the camera tilt is exactly $\frac{\pi}{2}$. Thus the camera tilt can be determined as follows, in accordance to section 3.1

$$\beta = \frac{\pi}{2} - \alpha.$$

3.4 Roll Correction Using Objects Higher than Eye-Level

In the methods outlined in the sections 3.1, 3.2 and 3.3 we assume that the camera is not rotated on the optical axis (i. e. roll = 0).

Not only does this rotation have an effect on the calculation of the tilt; it also influences the following calculation of the distance to respective objects, if these are not located in the center of the image.

The effects of the rotation on the tilt are examined in detail in section 4.2[2]. In order to calculate the roll we can use the inclination of objects in the image. For example, in the case of a landmark of the 4-Legged League, the horizon is always perpendicular to it. Another method to calculate the slope of the horizon is to determine the height of the horizon by means of several objects, e.g. two goal posts as shown in figure 4 (left). The roll can be easily calculated with the slope of the horizon. If we think of the horizon as a line in the image, the roll of the camera is the gradient of the straight line.

3.5 Using Knowledge about the Kinematic Chain

In some cases, the kinematic chain is known so that the position of the camera can be deduced by means of the joint data. For example, this holds true for AIBO.

[2] The effects on the rotation of the camera on the distance become obvious in the section 2.

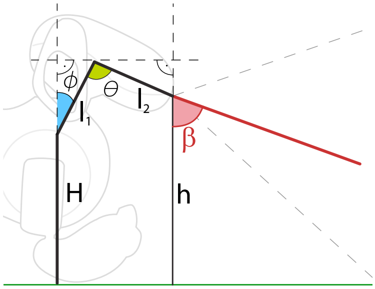

Fig. 5. Relation between the height of the camera and the angle of the neck-joint

In this case the whole kinematic chain can be determined via the joint data. However, the results are partly rather inaccurate. This is due to the fact that the contact points of the robot and the ground cannot be determined precisely. Furthermore, some of the joint sensors provide inaccurate data. In this section we want to examine how the knowledge of the kinematic chain can be combined with the outlined methods in order to achieve better results.

All the outlined methods provide us with the relations, or dependencies, between the different parameters of the camera, as e.g. the tilt and the height, which result from the respective observations. The kinematic chain also yields information on the relations of these parameters. Thus it is evident to try and determine the interesting parameters so that all given relations are fulfilled.

In many cases, there are not enough independent relations to determine all parameters. However, it is possible to write all camera parameters as a function of the joint angles. In turn, we can consider some of the joint angles as parameters and optimize them.

As an example, we use the method outlined in section 3.1 in order to correct the angle of the neck joint in the case of AIBO.

Application for Aibo Robot. According to our findings, AIBO's neck tilt is one of the most substantial error sources. This joint particularly has an effect on the determination of the camera's height. In the method outlined in section 3.1 the height of the camera is implied in the calculations so that this error also affects the results.

In order to completely eliminate the influence of the neck joint we have to make use of our knowledge of the relation between the neck joint and the height of the camera. This relation is depicted in figure 5. The interdependence of the height and the neck tilt can be formulated as follows:

$$h = H + l_1 \cdot \cos(\phi) - l_2 \cdot \sin(\frac{\pi}{2} - \theta - \phi)$$

and for the camera tilt β it holds $\beta = \theta - \phi$. Applying this to the formula outlined in section 3.1 the following function can be defined:

$$f(\phi) := \left(\frac{h(\phi) - \rho}{\rho} \cdot \sin(\alpha)\right) - \cos(\beta(\phi))$$

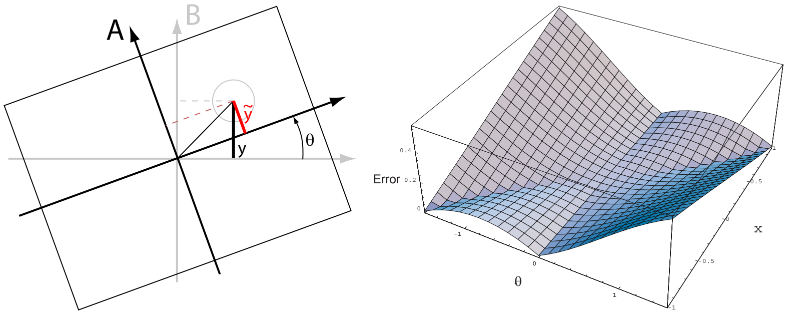

Fig. 6. (left) Camera rotated on its optical axes. (A) is the real coordinate system of the camera, (B) is the not-rotated coordinate system. The coordinate y is necessary for the calculation of the distance to an object. However, the coordinate $\tilde{y}$ measured in the image differs from y in case $\theta \neq 0$. (right) Error $|\tilde{\beta} - \beta|$ caused by ignoring the camera roll θ. The y-position is assumed as $y = 1mm$ (nearly the maximal y-position on the Aibo ERS7 camera chip) and the focal length as $f = 3.5mm$, θ and x-position are varied.

The angle ϕ can be determined as root of the function f. Thus the sensor data of the neck joint does not affect the determination of distances to other objects.

4 Error Estimation

In this section we want to analyze the effects of errors on the above mentioned methods in order to evaluate the quality of the results.

4.1 Errors of the Horizon-Based Methods

In many cases, the height of the robot is not known, e.g. if AIBO is walking. The method outlined in section 3.3 is particularly robust concerning this kind of errors. Let the error of the robot's height be h_e, the resulting error β_e of the roll angle is

$$\tan(\beta_e) = \frac{h_e}{d},$$

whereas d is the distance to the object that is used to measure the horizon. In the case of AIBO this would result in an error of $\beta_e = 0.03$, if the error of the height is $h_e = 3cm$ and the distance between the robot and the goal is $d = 1m$. The method becomes more robust with increasing distance to the reference object.

4.2 Estimating Errors Caused by Unknown Camera Roll

The camera tilt is essential for the determination of the distance to other objects. This is why all outlined methods deal with the correction of the camera tilt. Actually, there are cases in which the roll angle has a major influence on the result. The methods in the sections 3.1, 3.2 and 3.3 the roll angle is ignored. Thus we want to examine the effect of this on the results. The error estimation is only performed for the method using the ball as reference object, however for the other methods it can be done in the same way.

We consider all coordinates concerning the center of the image. Let $p = (x, y)^T$ be the center of the not-rotated image and θ the rotation of the camera on the optical axis as shown in figure 6 (left). We need the y-position of the ball's center in order to correct the tilt. After the application of the rotation we get the position of the ball's center as it would be detected in the image, in particular the measured height of the ball's center is then given by

$$\tilde{y} = x \cdot \sin\theta + y \cdot \cos\theta.$$

Thus it is obvious that the extent of the rotation's influence depends on the distance between the center of the image and the ball. Figure 6 (left) illustrates above treatments.

With the notation used in section 3.1 we can denote

$$\beta = \arccos\left(\frac{h-\rho}{\rho} \cdot \sin(\alpha)\right) + \arctan\frac{\tilde{y}}{f}$$

whereas f is the focal length of the camera. Figure 6 (right) illustrates the errors in case $\theta \neq 0$. As the figure shows, the error can be neglected if the angle θ is near zero. Thus, the method will yield acceptable results even though the roll of the camera is ignored if the roll is small enough.

5 Experiments

A number of experiments have been made with AIBO in order to evaluate the outlined methods under real conditions.

5.1 Projection Experiments

A good method to evaluate the accuracy of the camera matrix is to project images to the ground plane. In this subsection we describe two experiments using this methods. The first experiment evaluates the camera matrix obtained using the goal in images. In the second experiment a corner of field lines is used to correct the robots neck tilt joint.

Testing Accuracy of Horizon-Based Tilt and Roll Estimation. This experiment was performed in order to test the accuracy of the horizon based methods outlined in the section 3.3 and 3.4.

In the setup of this experiment the robot is situated in the center of the field and directed towards the blue goal. There is a calibration grid right in front of the robot.

During the Experiment the robot runs on the spot, the camera is directed towards the goal. The camera matrix is calculated and the correction is applied

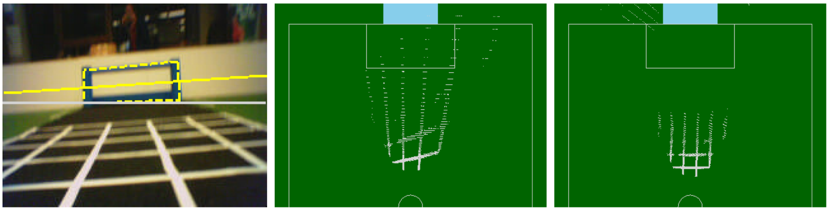

Fig. 7. (left) the scene from the view of the Aibo robot, (center) projection of the grid by means of the camera matrix calculated from the joint data, (right) projection of the grid by means of the corrected matrix

by calculating the tilt and roll angles with the help of the goal in the image (according to the method outlined in section 3.3). Figure 7 (left) shows the situation viewed by the robot. The image captured by the camera is projected to the ground plane by means of both matrices (the one calculated by the means of the kinematic chain and the corrected one).

Distortions occur if the values of the camera matrix do not correspond to reality, i.e. the lengths of the edges are not equal any more and the edges do not form a straight angle. All these effects can be increasingly observed in the case of a camera matrix that is calculated by means of joint data (figure 7 (center)). There are no distortions in the case of the corrected matrix, as can be seen in figure 7 (right).

Testing Field Line Corner Based Tilt Estimation. This experiment consisted of two parts. In the first part the robot was standing and looking straight ahead at a corner of field lines. The robot's hind legs were lifted manually by approximately 10cm resulting in a body tilt of up to 30 degrees. In the second experiment the robot was running on the same spot again looking at a corner of field lines. The running motion caused inaccuracies in the measurement of the neck tilt angle. The body tilt was estimated by the approach described in section 3.2. Both experiments have in common, that the distance to the corner does not change. To visualize the result of this estimation the images of the corner were projected to the ground using the camera matrix obtained from the readings of joint values and using the corrected camera matrix (see figure 8). The distance to the projected vertex of the corner using the corrected camera matrix was almost constant over time. Using the uncorrected camera matrix resulted in a large error in the distance and the angle of the projection of the corner. Thus the method was able to significantly increase the accuracy of bearing based distance measurements.

5.2 Distance Experiment

In this experiment we calculate the distance to another robot with the help of the bearing-based approach. Here, the parameters of the camera are corrected with different methods, which gives us the opportunity to compare them.

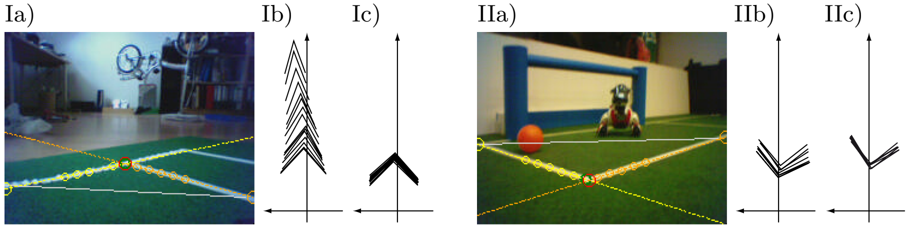

Fig. 8. This figure illustrates the correction of camera tilt by the means of corners of the field lines. The situation from the view of an Aibo and the perceptions of a corner projected to the ground are shown. In the first experiment the hind legs of the Robot were lifted manually, thus the resulting offset in the tilt angle can not be calculated from the joint data only. The figures Ib) and Ic) show the projections of the corner based on the joint data only, and using the corrected neck tilt respectively. IIb) and IIc) illustrate the not-corrected and corrected projections of the corner that was seen while by robot while walking on a spot.

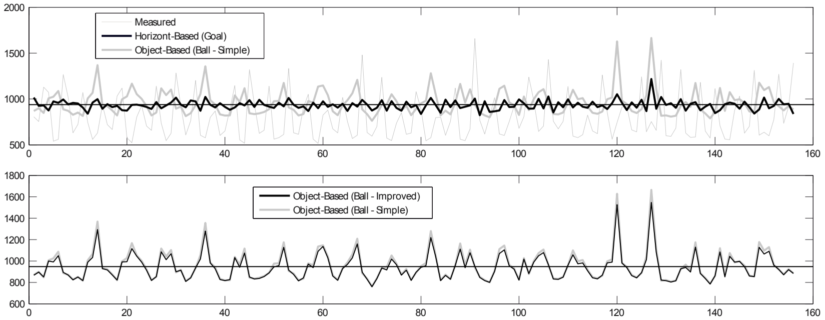

Fig. 9. (top) The distance determined by means of the camera tilt calculated from the joint data is shown in comparison to the distance determined with the help of the method using the size of the ball (outlined in section 3.1) and the method based on the horizon (outlined in section 3.3). (bottom) comparison between the results of the method using the ball as reference object and the combination of this method with the knowledge of the kinematic chain as described in section 3.5.

The setup is the same as in the experiment described above. However, there is no calibration grid in front of the robot. In addition, there are a ball and another robot in the robot's visual field.

In this experiment we correct the camera matrix with the help of the ball (directly and indirectly) and the goal, respectively. In order to compare the results we determine the distance to the other robot with the different corrected camera matrices, respectively. As the observing robot runs on the spot and the other robot does not move, the distance between them is constant. However, the calculated distance varies, due to errors. Figure 9 summarizes and compares the different results within the time scope of about 30 seconds.

The deviations in the not-corrected case are particularly very high. The best results were achieved by applying the horizon-based method. The two methods using the ball as reference object provide nearly identical results that are feasible.

6 Conclusion

We showed several methods to determine the camera pose of a robot relative to the ground using reference objects. This can help to improve bearing-based distance measurements significantly. Our work is relevant for all kinds of robots with long kinematic chains or unknown contact points to the ground as for these robots it is hard to determine the orientation of the camera using proprioception. As we provided methods for different kinds of reference objects there is a hight probability for a robot to see a suitable reference object. Experiments on Aibo showed that the methods work in practice.

References

1. Zhang, Z.: Camera calibration with one-dimensional objects (2002)
2. He, X., Zhang, H., Hur, N., Kim, J., Wu, Q., Kim, T.: Estimation of internal and external parameters for camera calibration using 1d pattern. In: AVSS 2006: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, Washington, DC, USA, p. 93. IEEE Computer Society, Los Alamitos (2006)
3. Junejo, I., Foroosh, H.: Robust auto-calibration from pedestrians. In: AVSS 2006: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, Washington, DC, USA, p. 92. IEEE Computer Society, Los Alamitos (2006)
4. Heinemann, P., Sehnke, F., F.S., Zell, A.: Automatic calibration of camera to world mapping in robocup using evolutionary algorithms (2006)
5. Benosman, R., Douret, J., Devars, J.: A simple and accurate camera calibration for the f180 robocup league. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 275–280. Springer, Heidelberg (2002)