Automatic acquisition of robot motion and sensor models

A. Tuna Ozgelen, Elizabeth Sklar, and Simon Parsons

Department of Computer & Information Science
Brooklyn College, City University of New York
2900 Bedford Avenue, Brooklyn NY 11210 USA
{tuna,sklar,parsons}@sci.brooklyn.cuny.edu

Abstract. For accurate self-localization using probabilistic techniques, robots require robust models of motion and sensor characteristics. Such models are sensitive to variations in lighting conditions, terrain and other factors like robot battery strength. Each of these factors can introduce variations in the level of noise considered by probabilistic techniques. Manually constructing models of noise is time-consuming, tedious and error-prone. We have been developing techniques for automatically acquiring such models, using the AIBO robot and a modified RoboCup Four-Legged League field with an overhead camera. This paper describes our techniques and presents preliminary results.

1 Introduction

Robots in RoboCup have two main requirements in order to play effective soccer. They have to be able to self-localize with reasonable accuracy [6], and they have to be able to detect and track the ball [13][1]. The current state-of-the-art in localization is to use Bayesian filter models [22, chap. 3–4], and a particularly popular approach is the particle filter [23]. This is especially popular in RoboCup because it allows robots to track multiple position hypotheses, helpful when robots are regularly kidnapped by referees, while running on modest computational hardware. To apply any Bayesian filter model, a robot requires a model of its own motion, which it uses to predict new poses from old ones following motion, and a model of its sensor behavior, which the robot uses to choose between multiple possible poses. The sensor model is clearly also important for detecting and tracking the ball.

Now, it is clear that the sensor and motion models are of importance to obtaining effective behavior from any robot, but they are especially important in vision-based soccer-playing robots. As a number of authors have pointed out, for example [6, 15], vision-based robots have much less sensor data to work with than robots equipped with sonar or laser range-finders (at least when the vision is based on landmark detection as it so often is in RoboCup). This comparative paucity of sensor data argues for the importance of making each datum as accurate as possible (though it should be noted

[1] Successful soccer-playing robots clearly need to be able to do a lot of other things as well, but these other things — effective moving of the ball, tactical positioning, and coordinated team play, for example — have good self-localization and ball-detection as pre-requisites.

Fig. 1. An AIBO with a color marker.

that if sensor data is too accurate, the performance of the particle filter degrades slightly [23]). The paucity of sensor data also argues for making the motion model as accurate as possible — with infrequent sightings of landmarks, robots have to run for several seconds at a time without sensor data [15], and during that time can only update their notion of where they are using motion data. Furthermore, when tracking the ball, the robot may not see a landmark for considerably longer, and so will have to rely on what is effectively dead-reckoning from its last confirmed position.

This requirement on the vision sensor model holds not only for models of the kind that we deal with here, which use information about distance and bearing to landmarks, but also for models that deal only with bearing [12] (and recent work [15] shows that distance information helps to improve the precision of localization provided that the distance information is adequately calibrated).

In this paper, we are concerned with the Sony AIBO ERS-7, the robot used by our Legged League team MetroBots[2]. To construct both motion and sensor models for the AIBO we are usually reduced to taking measurements "by hand and tape measure" [18]—running the robot for a given time and measuring how far it moved, or having the robot estimate how far it is from a landmark and comparing that with the measured distance. This gives relatively few measurements from which to construct and evaluate models, and the work described here is a response to that situation.

In this paper we describe how we have been using a global vision system, a system which uses an overhead camera, and from that image data determines the position of the robot, to automatically acquire motion and sensor data. This approach allows us to collect data sufficiently easily and rapidly — several hundred data points in an hour [3] — that we can use data-intensive machine learning techniques to construct models of motion and sensor error.

2 Experimental setup

For our experimental work, we have adapted a modified setup derived from the RoboCup E-League [1]. The E-League makes use of a simplified small-size league environment, where global vision data is provided by a common vision server. This data is sent to

[2] `http://agents.sci.brooklyn.cuny.edu/metrobots`

[3] A limit set, effectively, by the fact that at the moment we have to have the robot write image data to its memory stick, which takes several seconds, and then upload the image by ftp and that we use just a single robot. A group of several robots could collect data faster, as suggested in [11].

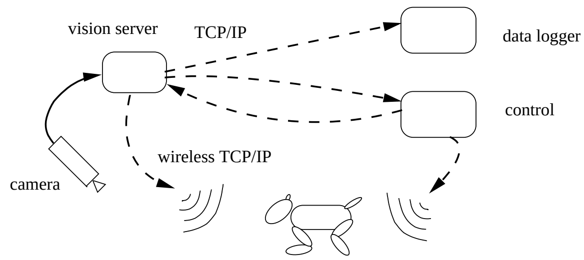


Fig. 2. The experimental setup.

both teams using UDP broadcast. Teams decide how to move their robots, and package instructions for the robots into a common format. These instructions are then combined into messages by the communication server, and broadcast to the robots via an infra-red transmitter. Each robot on each team unpacks the messages to find out what to do next.

At the heart of our setup is the Mezzanine visual tracking package [9]. Suitably calibrated, this software provides 2D tracking of objects — establishing x, y coordinates and orientation — provided that the objects are color coded for easy recognition from above. Mezzanine provides accurate tracking even with very unsophisticated camera hardware and can handle considerable image distortion. We currently use an XCam2 WideEye from X10, an inexpensive wide-angle security camera[4]. The original vision tracking system used by the E-League was Doraemon [2], which provides robust position estimates even when the camera is mounted at an angle rather than directly overhead. We are using Mezzanine because it more accurately handles the type of fish-eye images obtained from the wide-angle camera that is needed in order to get the whole soccer pitch in a single field-of-view.

As mentioned above, instead of the type of small, wheeled robots that have typically been used in the E-League, we have been working with Sony AIBO ERS-7 robots. To make them visible to Mezzanine, we simply attach a color marker to the back of the robot as in Figure 1. Since the AIBO is equipped with a wireless ethernet card, we can send data between the robot and the computer that is running the control code and the data logger (both are the same machine, though logically distinct), and we can send the position data from Mezzanine directly to the robot as well. The setup is as in Figure 2.

The idea of the experimental setup is to provide a completely automated mechanism for data-collection. The control module polls Mezzanine for location data and simultaneously sends instructions to the AIBO telling it how to move around the pitch, and when to gather data from its internal camera. When the robot is moving, we can continuously collect data about its position, and collate this position data with the motion commands sent to the robot. As we discuss below, this data can be used, amongst other things, to learn a motion model for the robot.

In addition to collecting this motion data, we can collect sensor data from the robot. Of particular interest, given the fact that the data used by the robot for self-localization

[4] `www.x10.com`

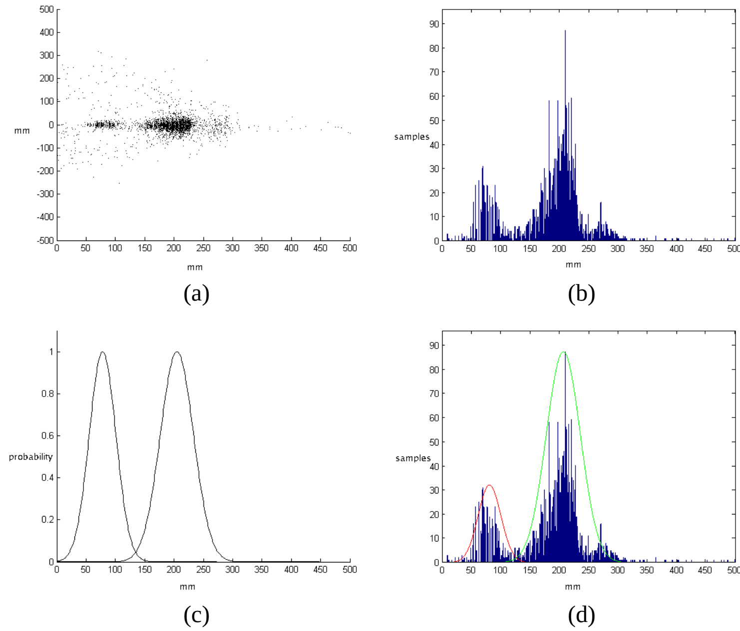


Fig. 3. The motion model for walking forward: (a) scatter plot of rates of motion in the x and y directions (x on the horizontal axis, y on the vertical) when walking forward; (b) histogram of motion in the x direction when walking forward; (c) Gaussian mixture fitted to the forward motion data; and (d) Gaussian mixture scaled and plotted with histogram of forward data.

is visual data, is the collection of camera images. Currently we do this by causing the robot to pause—thus allowing us to get an accurate idea of where each picture was taken without having to synchronise the clocks on the robot and the machine running Mezzanine—and then take a picture (which takes a few seconds to write to the robot's memory stick) and then upload the picture to the data logger.

3 Results

We used the setup described in the previous section to construct models for the robot's standard trot gait and the error in its perception of the Legged League markers. The robot gait is that from the motion module of the Carnegie Mellon University Legged League team CMPack'04 from the 2004 RoboCup competition [4].

3.1 Motion model

Data for the motion model was collected by making the AIBO walk forwards and backwards for 10 seconds at a time, while Mezzanine measured the coordinates of the robot at one second intervals. From these measurements, we computed the velocity of the robot over the relevant period in the three coordinate directions of the global frame of reference used by Mezzanine[5]. Since the robot takes time to accelerate and decelerate, we effectively had two sets of data—measurements for the robot moving continuously, and measurements for the robot when it was speeding up or slowing down.

For both forwards and backwards motion, we then plotted a histogram of around 3600 velocity measurements, obtaining two-peaked distributions — the lower valued peak corresponding to times when the robot was changing veocity, and the larger peak corresponding to constant velocity motion — that were approximately Gaussian. We then learnt the parameters of a two-Gaussian mixture that fitted the data. This learning was carried out using the standard EM algorithm [5]. A sample of this procedure for the x component of forward motion (that is the component in the direction of motion) is provided in Figure 3. Looking at Figure 3 (a) the two sets of measurements are clear, and these emerge as two distinct peaks in the histogram in Figure 3 (b) and (d). As Figure 3 (c) and (d) show, the two-Gaussian mixture closely fits the data.

The two forward motion distributions have means of 77 and 204, and standard deviations of 22 and 28 respectively, while the two backward motion distributions have means of 87 and 174, and standard deviations of 27 and 23, respectively.

3.2 Sensor model

Our second use of the experimental setup was to measure the error in the robot's estimates of its distance from the Legged League beacons. To do this, we first used the experimental setup to have the robot move around the pitch taking pictures, and recording the robot's position when these pictures were taken[6]. We used these images to build a color map and to calibrate a distance coefficient, based on the number of pixels counted for each beacon shape and the robot's distance measured from the beacon by hand. We then used the experimental setup to have the robot take a much larger set of images, again recording the position at which each picture was taken. For each of this second set of images we had the perception system of the robot calculate the distance to the beacon, and we compared this with the real distance as measured by the global vision—the difference is then the error in the local vision system.

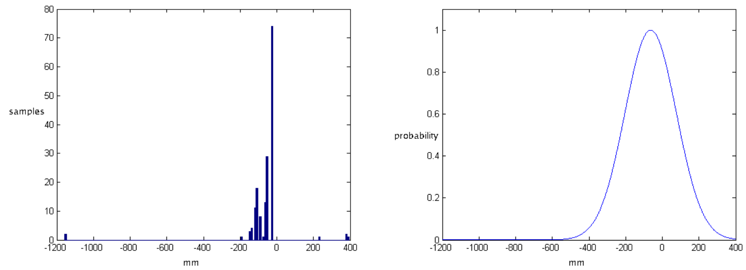
Given this error data, we then carried out exactly the same kind of learning as in the previous section, and the steps in this process are as depicted in Figure 4.

3.3 Discussion

The main thrust of the work described here has been the use of the external camera to measure robot pose and the subsequent use of this information, in conjunction with information computed on board the robot, to develop a motion model for the robot and a sensor error model. This is rather different to most existing work on developing vision models within RoboCup, for example [3, 10, 16, 17, 24], which has tended to concentrate on the automated segmentation of images, especially with an eye to handling

[5] Taking due account of the orientation of the robot in that frame of reference.

[6] In fact we combined taking pictures with the motion measurements required for the motion model.



(a) (b)

Fig. 4. The sensor error model (a) histogram of the error established from the global vision; (b) the model learnt from the data (adjusted for a measurement of 1200 mm).

changing illumination of the playing field, or work such as [21], which has concentrated on automatically identifying landmarks from sensor data.

Of course, there are problems with using the overhead camera as a measure of "ground truth", since, as [14] points out, overhead camera-based global vision systems tend to suffer from quantization problems and are adversely affected by noise in the image. However, these problems are much reduced for us in comparison with [14] thanks to the unique beacons used by the Legged League — [14] studies the small size league setup. These beacons greatly simplify the problem of localizing a robot by uniquely anchoring points on the image. Furthermore, while an occasional error in robot localization can have catastropic effects on the way that the robot plays soccer, which is the concern in [14], in our work an error will only introduce a little more noise, and create distributions with slightly more variance.

Our work described here is clearly related to the simultaneous learning of sensor and motion models described by [18, 19]. That exciting work promises to supercede what we are doing here, but for now is only capable (at least as reported in the literature) of learning models that work in the same single dimension — in the case of [18, 19] that is motion towards and away from a beacon, along with sensing of the distance to the beacon. In contrast our approach, while requiring data external to the robot — which is clearly a limitation in some domains — can aquire multi-dimensional models (and so, for example, can easily acquire models for the y direction and rotation).

4 Future work

We began this work not just to obtain data from which we could learn motion and sensor models off-line, but in order to be able to learn them on-line. In particular, we wanted to be able to run the robot, have it self-localize, and then adjust the parameters that control its motion and sensor models in order to improve its self-localization in much the way that [11] adjust parameters in order to optimize the robot gait (though clearly in a less autonomous way). This is still our aim, and we are continuing to work towards it. At the moment, as an intermediate between our overall goal and what we have reported here, we are using the experimental setup we have described to evaluate our use of particle filtering to localize the AIBO while it is playing soccer.

There have been many previous evaluations of localization. For example [7] examine a range of different probabilistic algorithms, while [6], and [20] evaluate RoboCup specific approaches, and [8, 12] look at the quality of localization on the AIBO in a RoboCup setting. However, all of these use rather contrived scenarios. For example, [6] required the robot to be manually placed around the pitch in order that the true location be known, while [8] controlled the robot with a joystick and obtained measurements by moving the robot over a known location and seeing where the robot thought it was as it passed over that location. [12] comes closest to what we are working on, using a laser range-finder to monitor continuously the real location of the robot, but never carried this out during a game (the addition to the robot to allow the laser to detect the robot presumably prevented this). As a result, we have no data on the extent to which actually playing, and thus, as described above, having to focus on the ball, affects the quality of the localization.

5 Summary

This paper has described the use of a global vision system as a means of automatically acquiring motion and vision sensor data for a legged robot. Despite the fact that these models are essential in order that robots can accurately self-localize, there has been little work to try and acquire them automatically. In addition to describing the process by which we collect the data in order to construct the motion and sensor models, we have demonstrated the kinds of results that it is possible to obtain in this way. In particular, we gave two components of the motion model for an AIBO ERS-7 that we learnt in this way, and the error model for the extraction of the beacons on the Four-Legged League pitch. While the learning process currently involves some human intervention, and is run on an off-board computer, there is no especial reason why the process could not be completely automated and run on-board.

Acknowledgements This work was partially supported by NSF CNS-0520989 and CUNY Collaborative Grant 80209-09-12. Thanks to Aleksandr Barkan for help with calibrating Mezzanine and setting up the overhead camera.

References

1. J. Anderson, J. Baltes, D. Livingston, E. Sklar, and J. Tower. Toward an undergraduate league for RoboCup. In *RoboCup 2003: Robot Soccer World Cup VII*, pages 670–677. Springer Verlag, Berlin, 2004.
2. J. Baltes. Yue-Fei: Object orientation and ID without additional markers. In *RoboCup 2001: Robot Soccer World Cup V*, pages 257–262. Springer-Verlag, Berlin, 2002.
3. D. Cameron and N. Barnes. Knowledge-based autonomous dynamic color calibration. In *RoboCup 2003: Robot Soccer World Cup VII*, pages 226–237. Springer Verlag, Berlin, 2004.
4. `http://www.cs.cmu.edu/~robosoccer/legged/`. (accessed Feb. 2nd, 2006).

5. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 34:1–38, 1977.
6. S. Enderle, M. Ritter, D. Fox, S. Sablatnög, G. Kraetzchmar, and G. Palm. Soccer-robot localization using sporadic visual features. In *Proceedings of the 6th International Conference on Intelligent Autonomous Systems*, 2000.
7. J-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.
8. J-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
9. A. Howard. Mezzanine user manual, version 0.00. Technical Report IRIS-02-416, Institute for Robotics and Intelligent Systems, USC, 2002.
10. M. Jungel, J. Hoffmann, and M. Lötzsch. A real-time auto-adjusting vision system for robot soccer. In *RoboCup 2003: Robot Soccer World Cup VII*, pages 214–225. Springer Verlag, Berlin, 2004.
11. N. Kohl and P. Stone. Machine learning for fast quadrupedal locomotion. In *Proceedings of the 19th National Conference on Artificial Intelligence*, San Jose, CA, July 2004.
12. T. Röfer and M. Jüngel. Fast and robust edge-based localization in the Sony four-legged robot league. In *RoboCup 2003: Robot Soccer World Cup VII*, pages 262–273. Springer Verlag, Berlin, 2004.
13. T. Schmitt, R. Hanel, S. Buck, and M. Beetz. Probabilistic vision-based opponent tracking in robot soccer. In *RoboCup 2002: Robot Soccer World Cup VI*, pages 426–434. Springer-Verlag, Berlin, 2003.
14. D. Sekimori, T. Usui, Y. Masutani, and F. Miyasaki. Evaluation of self-localization performance for a local vision robot in the small size league. In *RoboCup 2002: Robot Soccer World Cup VI*, pages 41–52. Springer-Verlag, Berlin, 2003.
15. M. Sridharan, G. Kuhlmann, and P. Stone. Practical vision-based Monte-Carlo localization on a legged robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, April 2005.
16. M. Sridharan and P. Stone. Autonomous color learning on a mobile robot. In *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, PA, July 2005.
17. M. Sridharan and P. Stone. Towards eliminating manual color calibration at RoboCup. In *RoboCup 2005: Robot Soccer World Cup IX*. Springer Verlag, Berlin, 2006.
18. D. Stronger and P. Stone. Simultaneous calibration of action and sensor models on a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, April 2005.
19. D. Stronger and P. Stone. Towards autonomous sensor and actuator model induction on a mobile robot. *Connection Science*, 18(2), 2006. (to appear).
20. A. W. Stroupe, K. Sikorski, and T. Balch. Constraint-based landmark localization. In *RoboCup 2002: Robot Soccer World Cup VI*, pages 8–24. Springer-Verlag, Berlin, 2003.
21. S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 33(1):41–76, 1998.
22. S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, Cambridge, MA, 2006.
23. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1–2):99–141, 2001.
24. J. C. Zagal, J. Ruiz del Solar, P. Guerrero, and R. Palma. Evolving visual object recognition for legged robots. In *RoboCup 2003: Robot Soccer World Cup VII*, pages 181–191. Springer Verlag, Berlin, 2004.