Fault-tolerant self localization by case-based reasoning

Jan Wendler*, Steffen Brüggert, Hans-Dieter Burkhard, and Helmut Myritz

Humboldt University Berlin, Department of Computer Science,
Artificial Intelligence Laboratory, Unter den Linden 6, D-10099 Berlin, Germany
email: wendler/bruegger/hdb/myritz@informatik.hu-berlin.de,
WWW: http://www.ki.informatik.hu-berlin.de

Abstract. In this article we present a case-based approach for the self-localization of autonomous robots based on local visual information of landmarks. The goal is to determine the position and the orientation of the robot sufficiently enough, despite some strongly incorrect visual information. Our approach to solve this problem makes use of case-based reasoning methods.

1 Introduction

In order to enable robots to make goal-oriented decisions, their position and direction are very important. If the operational area of the robot is known in advance, e.g. given by geographical maps, the robot should be supplied with this information to improve its localization skills. Our approach relies on information about absolute positions of landmarks in the operational area of the robot. This information is suitable for a case base. With such a case base the robot can determine its position and orientation in its operational area based on perceived sizes and shapes of landmarks and on perceived angles between pairs of landmarks.

In this article we only describe the determination of the robots position. Its orientation can be determined easily in a next step using the already determined position or it can be included into the described approach.

We are using the fully autonomous "Sony Legged Robots" [3]. These are small dog-like robots, that play soccer in teams of 3 robots at the Robocup competitions. At the games the robots are entirely independent, global view or remote control is not allowed. Furthermore, extensions of the robots hardware are prohibited. The robots play on a small field, which contains some well defined markers and goals of different colors. The "Sony Legged Robots" utilize a camera for input and an integrated video processing unit for discriminating between the different colored markers and goals.

* This research has been supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK 316).

P. Stone, T. Balch, and G. Kraetzschmar (Eds.): RoboCup 2000, LNAI 2019, pp. 259-268, 2001.

In the next section we describe the properties of the "Sony Legged Robots" and present their environment. In section 3 we describe our approach in detail. Afterwards we give an overview to some related work before we demonstrate some results of our approach in section 5.

2 The Robot and its Environment

Sony kindly provided us with four of their four-legged robots (ERS-1100), which are marketed by Sony in a similar configuration, known as AIBO (ERS-110 and ERS-111). With this platform there come some restrictions which are very close to the properties of biological systems. On the one hand these restrictions raise some problems, on the other hand they represent interesting challenges for the use of intelligent techniques.

The robot contains a camera, which has a very limited field of view[1]. The head, to which the camera is affixed in the front, has three degrees of freedom. With these the robot is able to overcome the disadvantages of the small field of view of the camera. Therefore the robot has the possibility to support its self localization by turning its head into a direction where it expects additionally landmarks. This can usually be done without interfering with the robot's current actions[2]. The integrated video processing unit performs a fast color separation into 8 colors, which is highly sensitive to temporal and spatial changes of the light, like flashes, reflections and shading. Such changes can heavily influence the shape and size of a color region or, much worse, it is possible that a color region is assigned to the wrong color. Unfortunately the integrated color separation algorithm returns only information about the biggest coherent area for each of these 8 colors. So some information in the pictures maybe already lost at this stage.

As already mentioned, the robot uses four legs for the movement in contrast to common robot architectures. Each leg has 3 degrees of freedom, two at the shoulder joint and one at the knee joint. The walking causes some tilting during movement, which leads to virtual relative movements of objects on the field.

For all other computations besides the color segmentation the robot has a 100 MHz RISC-processor (MIPS R4300) and 16 MB main memory. Figure 1 shows the structure of the dog-like robot.

The field of the robot is equipped with different colored fixed objects, which allow the robot to orient itself. These objects consists of six two-color markers and two one-color goals, which are used for self localization. Furthermore, it is conceivable to use the borders of the field and some other field markings like the middle line or the penalty area line for localization. In figure 2 a picture of the field with its landmarks is given.

[1] The CCD-camera has a resolution of 176×120 pixels and a field of view of approximately $47° \times 30°$.

[2] Unfortunately this does not apply for actions which involves the head, as the search for the ball.

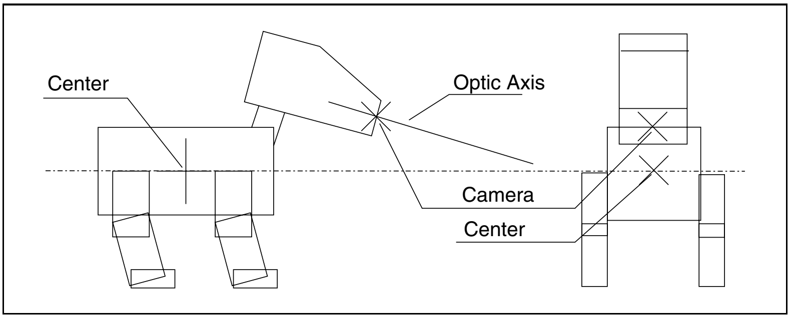

Fig. 1. Structure of the dog-like robot

The field has an effective length of 280 cm and an effective width of 180 cm. The dog itself is approximately 30 cm long and 14 cm wide.

3 Case Base Reasoning for the Self Localization

For the self localization we have divided the state space of field positions into a 14×9 (X, Y) grid. This corresponds to a raster of 20 cm $\times$ 20 cm. For each of these states a case was generated, that consists of information on the relative position of all landmarks for this state. To apply the technique of case base reasoning (CBR)[6] in this scenario, several design decisions have to be made. For that the following questions will be answered:

– What is a case? How will cases be represented?
– Where do the cases come from?
– How is the query to the case base determined?
– How is the similarity measure between the cases defined?
– How does the structure of the case base looks like?
– What is the final result?

3.1 Case Structure

Because the determination of the robots position depends on perceived sizes and shapes of landmarks and on perceived angles between pairs of landmarks, this information has to be a part of the case base. We are using the term landmark only for the elemental parts of markers and goals. Therefore a marker will be seen as two landmarks (one landmark for every color). With this view the uniqueness of landmarks seems to be lost, but it is still coded by small angles (angles of zero or near zero) between two landmarks that represent the same marker. The advantage of this view takes effect if parts of a marker are covered by other robots or if only parts of a marker are identified by the robot, due to shading for instance. In such cases the information about the identified part of the marker isn't lost. Furthermore this view is more universal, because it permits different landmarks with identical appearance. According to this view a case consists of the following properties:

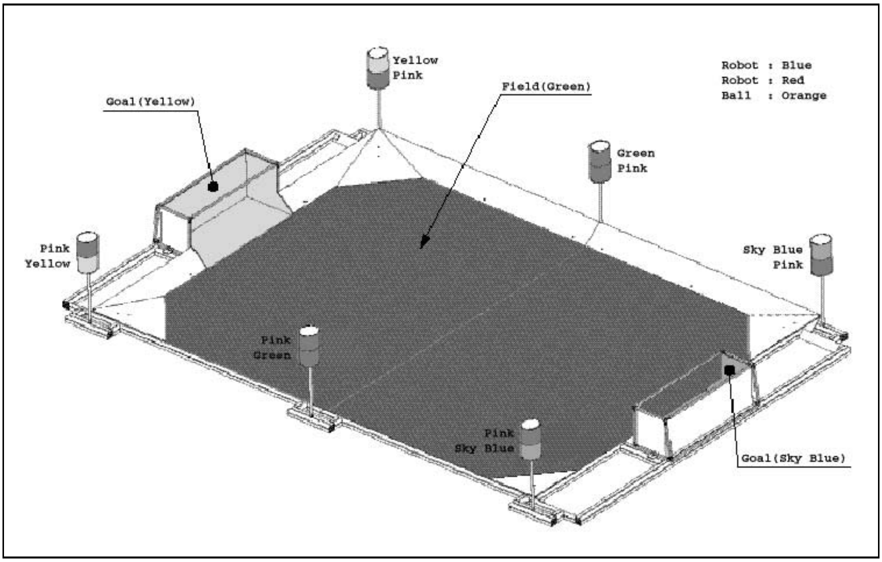

Fig. 2. The field with its landmarks

Size of all landmarks and their position in the picture: For every case the width and the height of all 14 landmarks (six times pink, three times yellow, three times sky blue and two times green) are saved. Width and height indicate the amount of pixels of that color in the corresponding dimension. With the separation of the markers into their elemental parts the information about whether the landmark corresponds to an upper or to a lower part of a marker is lost. For that reason this information is included in the case. For each landmark this is done by its position in the picture. If the landmark is a known part of a marker its position has the value "top" or "down", if it is an unknown part of a marker its position has the value "unknown". The last case only occurs in the queries. If the landmark is a goal its position has the value "irrelevant".

Angle between pairs of landmarks: Whereas the perceived size of landmarks may have a relatively high error because of changes of the light, the error of the angles between pairs of landmarks is more limited. If the robot is moving a lot the error can get a higher value. For that reason we decided only to look at the pairs of landmarks which are directly or indirectly (exactly one marker between the two landmarks) neighboring and at pairs which represent the same marker. So we have 6 pairs where each pair represent one marker, 24 direct and 24 indirect neighboring pairs.

Every property which is assigned with a fixed value will be called an information entity (IE) in the following. A case consists of a set of IEs, here there are 68 IEs. Theoretical we could have up to *number of cases multiplied by number of IEs per case* different IEs ($14 \times 9 \times 68 = 8568$). But many IEs are shared among different cases so we have a total of 859 different IEs in our case base. A case represents an omni-directional picture of the robot. The solution of the case is the position of the robot on the field from which the picture was taken.

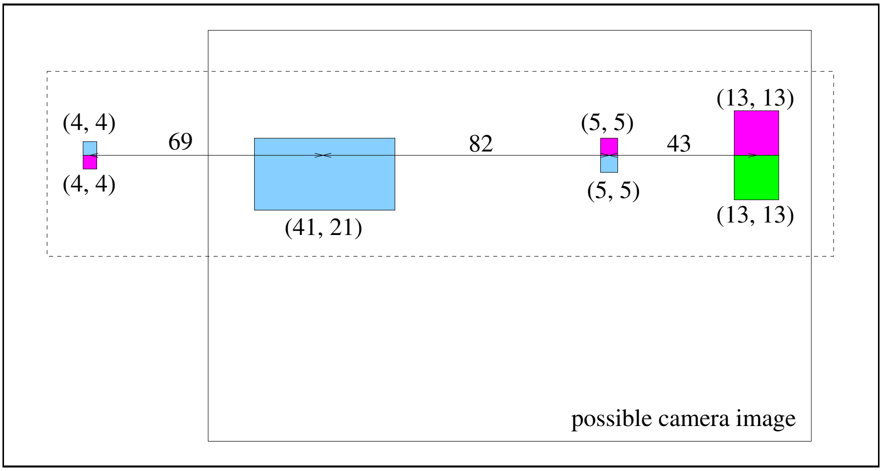

Fig. 3. Graphical excerpt of the case base

In figure 3, a graphical extract of the case base is presented for the case where the robot is at position (110, 60), which is the position marked with a black circle in figure 6. The values in figure 3 are given in pixels of the dog's camera. The figure consists already of 24 IEs, 7 of them are representing the size of landmarks and its position in the picture. The other 17 represent angles between pairs of landmarks (3 pairs represent the same marker, 8 pairs represent direct neighboring pairs, and 6 pairs represent indirect neighboring pairs).

3.2 Where do the Cases come from?

The case base was generated semi-automatically in two steps.

In the first manual step, a table was created which maps the distances of landmarks to perceived sizes of these landmarks. Using the robots camera, pictures of landmarks were taken from different distances (5 cm to 3,20 m distance) and the sizes (in pixel) of the landmarks were determined. These measurements were done with one marker and one goal three times and the mean value was calculated.

In a second automatic step, the distances to all markers and goals and the angles between all relevant markers and goals were calculated for all the 14×9 positions on the field. After that the markers were separated into landmarks and the distance to the landmarks were replaced by the determined sizes. Hence the case base for the self localization was developed.

3.3 Determination of the Case Base Query

The color detection module supplies the self localization module with data about landmarks in the picture. For every detected landmark the following values are supplied:

- its color identifying information,
- its size (width and height) in pixel,
- its balance point in the camera picture as (x, y) coordinate, and
- the orientation of the head at the picture creation time.

Using the x-coordinate of the balance points it is possible to determine the *angles between pairs of landmarks*. In a first step the angles will be normalized according to the optic axis of the robot. After that the angles of landmark pairs are calculated. Small angles indicate that this pair represents the same marker. By comparing the y-coordinates of such pairs of landmarks the *position of the landmark* ("top" or "down") *in the picture* can be determined. The position of any landmarks whose position isn't set in this manner is treated as "unknown". The orientation of the head is used to normalize the relative angles (normalized to the optic axis) of landmarks according to the orientation of the robot. With this normalization it is possible to use informations about landmarks from older pictures despite head motions. This holds as long as the robot is only moving marginally.

The case base query is generated using as much visual information about landmarks as possible to smooth the errors of the observed data which is sometimes very big and other times very small.

In figure 4 an observation of two landmarks is shown. It shows a pink (dark) landmark at the top with an balance point of (72,96) and a size of (44,38), and a sky blue (light) landmark at the bottom with an balance point of (70,67) and a size of (40,21). Because the x-coordinates of these two landmarks does not differ so much, the angle between these landmarks is very small and therefore the two landmarks represent the same marker. Because the y-coordinate of the pink landmark is greater its position in the picture has the value "top" whereas the value is "down" for the sky blue landmark.

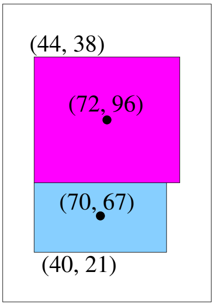

Fig. 4. Observation of two landmarks

3.4 Similarity Measure

The self localization module gets unambiguous color identifying information from the color detection module. Therefore a similarity comparison is only done between IEs with the same color identifying information. For this reason, we have separated the IEs into different categories.

The IEs, which represent the *size of landmarks and their position in the picture* has been separated into four categories (for every color, one category).

Two of these IEs of the same color are similar if their size difference is small and their position in the picture is equal or one of the positions is "unknown". The smaller the size difference, the more similar the IEs are. An "unknown" position of the landmark in the observed picture decreases the similarity to all IEs with positions "top" or "down" in the case base. If the query consists of the IE for the pink (dark) landmark of figure 4, all pink IEs whose position is "top" and who have a small size differences to the size (44,38) are similar to the query IE. The IE with the size (42,38) is more similar to the query IE than the IE with the size (40,36).

For the IEs which characterize *angles between pairs of landmarks*, the separation into categories is done according to their color combination. Because the combination (green, green) does not exist among the selected pairs, we get 9 categories here. A high similarity is given here between two IEs if its angle difference is small.

Given these local similarity measures, we could define a composite similarity measure which computes the weighted sum over all local similarities where all IEs have the same relevance. We instead use a different composite similarity measure where an IE has less relevance the more often this IE appears on the field. Given this, the IEs for the sizes of landmarks with the color pink has much less relevance than the IEs for the sizes of landmarks with the color green.

3.5 Structure of the Case Base and Retrieval

A crucial point of the self localization is that is has to be done in real time. An efficient organization of the case memory and of the retrieval procedure is needed. For that reason we apply the Case Retrieval Net (CRN) model[7].

In a CRN the case base is represented by a net of nodes for the IEs and by nodes which mark the cases. IE nodes might be connected among each other by similarity arcs. Every case node is connected with all its IEs nodes by relevance arcs. Based on this structure the retrieval works as follows:

1. activation of the IEs from the query,
2. propagation of the activation through the net along the similarity arcs, and
3. propagation of the activation reached so far to the case nodes along the relevance arcs.

Whereas the similarity arcs represent the local similarity measure, the composite similarity measure is represented by the relevance arcs. Different strength of similarity and relevance can be represented by weighting the arcs.

3.6 Determination of the Final Result

After having done the case based retrieval we get a list of cases which represents most likely positions of the robot. Every case is activated by some value, the higher the value the more likely the robot is at the corresponding position.

For the determination of the robot's position we consider all cases which have a similar activation to the maximum activated case (80% activation or more) and are not more than 50 cm away from the maximum activated case. A weighted sum over all these cases is computed to determine the robot's position.

4 Related Work

All the groups we are aware off, which focus on similar or even the same problems as we do in our approach, are using probabilistic methods for the self localization and robot navigation. In [8] the authors are using partially observable Markov models to robustly track a robot's location in office environments. Furthermore Markov models have been employed successfully in various mobile robot systems like [1, 4, 8]. Even in the same domain, for the "Sony legged robots", these models have been applied [5].

With our completely novel approach we want to evaluate whether we can reach similar results by applying case based techniques. We predict that our approach has some advantages and some disadvantages in relation to the Markov model approach. For instance we expect that our approach is faster if the information set of perceived landmarks is low whereas our approach maybe slower in cases where many landmarks can be perceived. Furthermore we think that active localization [2] can be supported more easily by a case base structure than by a Markov model.

5 Results and Future Work

To test and evaluate our approach we performed three simulated experiments and one real-world experiment. The simulated experiments were done automatically without using the robot or the field, whereas the real-world experiment was done manually with the robot on the field.

In our first simulated experiment we randomly chose 1000 positions on the field. For each of these positions, 10 virtual pictures were generated which consisted of 2 to 14 landmarks. For each of these 10000 pictures, IEs were created using the landmarks and incorporating an error between -20% and 20% of the correct value. These IEs were used as queries for our localization approach and the error between the correct position and the computed position was determined. The average error's dependence on the number of landmarks in the virtual picture is shown in figure 5 with the solid line. In the second experiment we incorporated a 50% possibility that the information about whether a landmark is the "top" or the "bottom" of a marker was lost. The results of this experiment is represented by the dashed line in the same figure. In the last simulated experiment, whose results are represented by the dash-dotted line, all 10 pictures of one position had the same number of landmarks (randomly chosen between 2 and 14). The IEs were created using the same error as above but this time every IE of each of these 10 pictures was used as localization query. This experiment is quite natural because the robot usually takes more then 10 pictures in a second and all these pictures have errors but different ones. In figure 5, only the average errors for the three experiments are shown, but sometimes the error was much

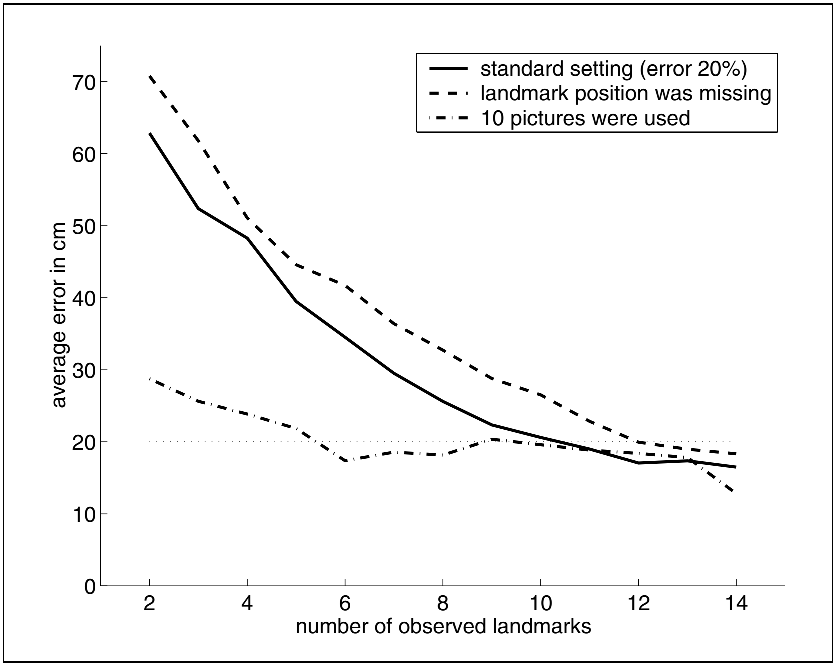

Fig. 5. Simulated Results

bigger. This happens if the incorporated errors modify the virtual picture in a manner such that it “looks” like a virtual picture from another position.

For our real-world experiments the robot was put onto the field and allowed some time to localize itself by turning its head and taking pictures. The actual location and the result computed by the robot were compared. Figure 6 shows three (out of 30) typical examples from this experiment. The dark squares show the actual and the light the computed positions of the robot. The figure also shows the given and the computed orientation of the robot. The orientation was computed using the determined position and one fully observed marker or goal. Although the robot sometimes had problems determining its position well enough (in example 3 it had an error of 75 cm), it was always pretty good at determining its orientation (maximum error of 10°).

The simulated and the real results already show the usefulness of the approach, even though we are not fully satisfied with it. During evaluation we discovered that the IEs which represent angles between landmarks don’t include the orientation of the angle, so we lost the information about which of the landmarks were right or left. Furthermore the algorithm to determine the final result out of the activated cases needs improvement. Up to now the robot localizes itself using only the most current information about landmarks. We also want to incorporate older information and the robot’s movements into the case base approach as well as enhancing our approach with active localization.

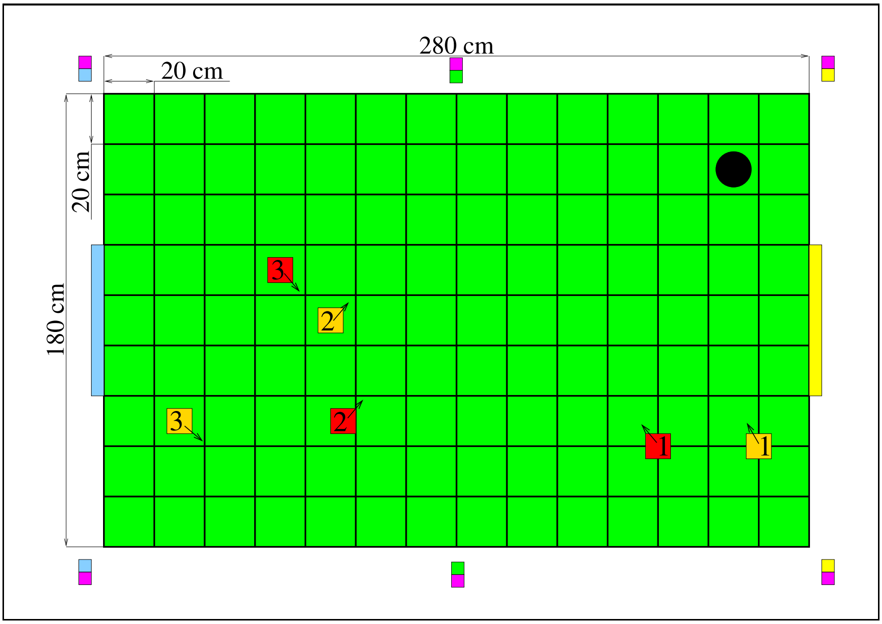

Fig. 6. Real Results

References

1. W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1996.
2. D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. In *Robotics and Autonomous Systems*, 1998.
3. M. Fujita, S. Zrehen, and H. Kitano. A Quadruped Robot for RoboCup Legged Robot Challenge in Paris '98. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, volume 1604 of *LNAI*, pages 125–140. Springer Verlag, 1999.
4. L. Kaelbling, A. Cassandra, and J. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 1996.
5. S. Lenser and M. Veloso. Sensor Resetting Localization for Poorly Modelled Mobile Robots. In *Proceedings of ICRA-2000*, 2000.
6. M. Lenz, B. Bartsch-Spörl, H. D. Burkhard, and S. Wess. *Case Based Reasoning Technology. From Foundations to Applications.*, volume 1400 of *LNAI*. Springer, 1998.
7. M. Lenz and H. D. Burkhard. Lazy propagation in Case Retrieval Nets. In W. Wahlster, editor, *12th ECAI 1996*, pages 127–131. John Wiley & Sons, 1996.
8. R. Simmons and S. Koenig. Probabilistic Robot Navigation in Partially Observable Environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1080–1087, 1995.