Panoramic Localization in the 4-Legged League
Removing the dependence on artificial landmarks

Jürgen Sturm[1], Paul van Rossum[2], and Arnoud Visser[1]

[1] Universiteit van Amsterdam
[2] Technische Universiteit Delft

http://www.dutchaiboteam.nl

Abstract. The abilities of mobile robots depend greatly on the performance of basic skills such as vision and localization. Although great progress has been made in the 4-Legged league in the past years, the performance of many of those approaches completely depends on the artificial environment conditions established on a 4-Legged soccer field. In this article, an algorithm is introduced that can provide localization information based on the natural appearance of the surroundings of the field. The algorithm starts making a scan of the surroundings by turning head and body of the robot on a certain spot. The robot learns the appearance of the surroundings at that spot by storing color transitions at different angles in a panoramic index. The stored panoramic appearance can be used to determine the rotation (including a confidence value) relative to the learned spot for other points on the field. The applicability of this kind of localization for more natural environments is demonstrated in two environments other than the official 4-Legged league field.

1 Introduction

1.1 Context

Mobile robots need to know where they are. Robot localization is therefore an important basic skill of mobile robots, e.g. when playing robot soccer. Many other processes of the robot's cognition - like world modeling and action planning - strongly depend on fast, accurate and robust position estimates.

In the 4-Legged league[3] of the RoboCup, teams consisting of four Sony Aibo robot dogs play soccer fully autonomously against each other on a field of 6x4m. Colored flags, goals and various field lines can be used to achieve localization accuracies below six centimeters [1, 2].

The price that these approaches pay is their total dependency on artificial landmarks of known shape, positions and color. Most algorithms even require manual calibration of the actual colors and lighting conditions used on a field and still are quite susceptible for disturbances around the field, as for instance produced by brightly colored clothes in the audience.

---

[3]RoboCup Four Legged League homepage, last accessed in April 2006, http://www.tzi.de/4legged

The interest in more general solutions has been (and still is) growing over the past few years. The almost-SLAM challenge[4] of the 4-Legged league and the upcoming challenges in natural environments of the newly founded RoboCup @ home league[5] reflect this growing interest within the robotics community.

1.2 Related Work

As many teams of the 4-Legged league use similar approaches for vision and localization (from the raw camera image to the robot's pose), we briefly want to point out typical main processing steps and their corresponding considerations.

1. In the **image processing** step, the image pixels are **divided** into color classes. Coarse scanning through the image then yields object candidates that need to be post-processed by **object specialists** for object recognition. Finally the complete set of detected objects is filtered by a second-order **sanity checker** to remove invalid percepts. Representative and well-working implementations can be found in [1, 3].
2. In the **localization** step, the perceived landmarks are filtered over time and merged with the data from the robot's body odometry. Common approaches include Kalman Filters, Monte-Carlo approaches or combinations thereof. For a good comparison see [4].

Other interesting approaches can be found in the Mid-Size league[6] : as the hardware of a Mid-Size robot is only limited to what it can carry, both different types of sensors as well as different classes of algorithms become feasible. Most Midsize robots use omni-directional high-quality cameras whereof each camera image carries enough information to localize the robot almost perfectly [5]. These approaches are not applicable on an Aibo with its limited camera angle; therefore panorama images have to be constructed of several images.

Also inspiring, but also only partially transferable to the 4-Legged league is the work that has been done on the SLAM problem in general, for instance on panoramic pictures [6–9]. One of these approaches [10] divides the panoramic image in multiple sectors, but uses as characteristic feature the average color of the sector. Our approach combines sectors with as characteristic feature the frequency of colortransitions.

2 Approach

The main idea is quite intuitive: we would like the robot to generate and store a 360° panorama image of its environment while it is in the learning phase. After

---

[4]Details about the Simultaneous Localization and Mapping challenge can be found at http://www.tzi.de/4legged/pub/Website/Downloads/Challenges2005.pdf

[5]RoboCup @ Home League homepage, last accessed in March 2006, http://www.ai.rug.nl/robocupathome/

[6]RoboCup Mid-Size league homepage, last accessed in April 2006, http://www.idt.mdh.se/rc/Mid-Size/

that, it should align each new image with the stored panorama, and from that the robot should be able to derive its relative position (in the localization phase). This alignment is not trivial because the new image can be translated, rotated, stretched and perspectively distorted when the robot does not stand anymore at the point where the panorama was originally learned.

Of course, the Aibo is not able (at least not in real-time) to compute this alignment on full-resolution images. Therefore a reduced feature space is designed so that the computations become tractable[7] on an Aibo. Figure 1 gives a quick overview of the algorithm's main components.
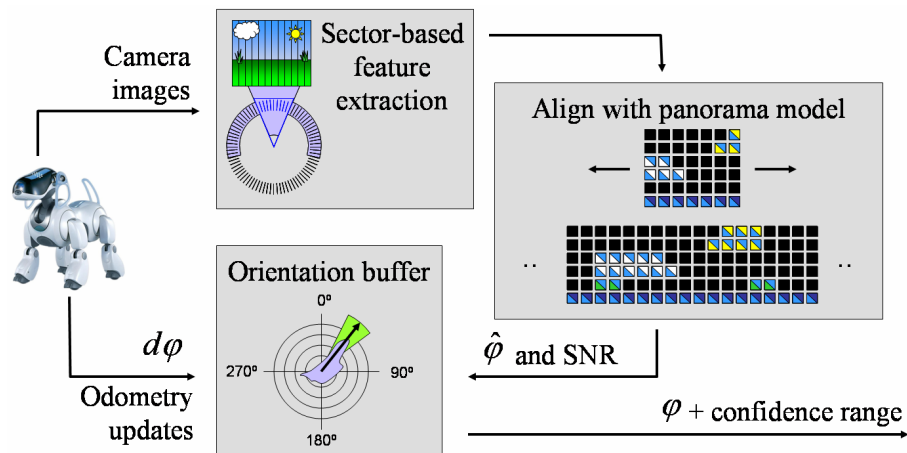

Fig. 1. Architecture of our algorithm

The Aibo performs a **calibration** phase before the actual learning can start. In this phase the Aibo first decides on a suitable camera setting (i.e. camera gain and the shutter setting) based on the dynamic range of brightness in the **autoshutter** step. Then it collects color pixels by turning its head for a while and finally clusters these into 10 most important color classes in the **color clustering** step using a standard implementation of the Expectation-Maximization algorithm assuming a Gaussian mixture model [11]. The result of the calibration phase is an automatically generated lookup-table that maps every YCbCr color onto one of the 10 color classes and can therefore be used to segment incoming images into its characteristic color patches (see figure 2(a)). As these initialization steps are not in the focus of this article we kept the explanation short.

---

[7]Our algorithm consumes per image frame approximately 16ms, therefore we can easily process images at the full Aibo frame rate (30fps).

(a) Unsupervised learned color segmentation.

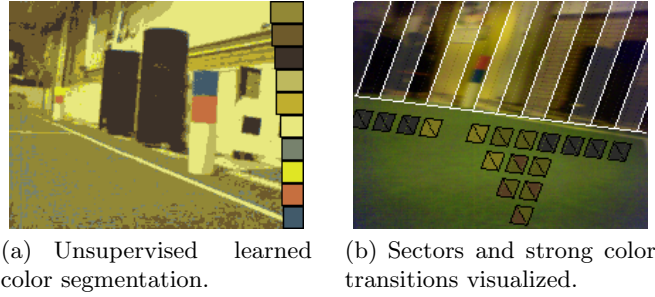(b) Sectors and strong color transitions visualized.

Fig. 2. Image processing: from the raw image to sector representation. This conversion consumes approximately 6ms/frame on an Sony Aibo ERS7.

2.1 Sector signature correlation

Every incoming image is now divided into its corresponding sectors[8]. Using the lookup table from the unsupervised learned color clustering, we can compute the sector features by counting per sector the transition frequencies between each two color classes in vertical direction. This yields 10x10 transition frequencies per sector, which we subsequently discretize into 5 logarithmically scaled bins. In figure 2(b) we displayed the strongest color transitions (bin 5) for each sector.

In the **learning phase** we estimate these 80x(10x10) distributions[9]. We define a single distribution for a currently perceived sector by

$$P_{current}\left(i, j, bin\right) = \begin{cases} 1 \; discretize\left(freq\left(i, j\right)\right) = bin \\ 0 \qquad\qquad otherwise \end{cases} \tag{1}$$

and the distribution learned from many frequency count samples of a certain sector by

$$P_{learned}\left(i, j, bin\right) = \frac{count_{sector}\left(i, j, bin\right)}{\sum\limits_{bin \in frequencyBins} count_{sector}\left(i, j, bin\right)} \tag{2}$$

Now we can simply multiply the current and the learned distribution to get the correlation between a currently perceived and a learned sector:

$$Corr(P_{current}, P_{learned}) = \prod\limits_{\substack{i,j \in colorClasses, \\ bin \in frequencyBins}} P_{learned}\left(i, j, bin\right) \cdot P_{current}\left(i, j, bin\right)$$

$$\tag{3}$$

---

[8]80 sectors corresponding to 360°; with an opening angle of the Aibo camera of approx. 50°, this yields between 10 and 12 sectors per image (depending on the head pan/tilt)

[9]When we use 16bit integers, a complete panorama model can be described by (80 sectors)x(10 colors x 10 colors)x(5 bins)x(2 byte) = 80 KB of memory.
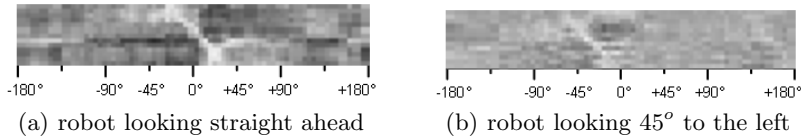
(a) robot looking straight ahead   (b) robot looking 45° to the left

Fig. 3. Resulting sector correlation matrix between the 12 seen sectors of the current image and the 80 stored panorama model sectors. The bright diagonal line corresponds to a minimum in the correlation and constitutes the alignment we want to find.

2.2 Alignment

After all the correlations between the stored panorama and the new image signatures were evaluated, we would like to get an alignment between the stored and seen sectors so that the overall likelihood of the alignment becomes maximal. In other words, we want to find a diagonal path with the minimal cost through the correlation matrix.

We consider the fitted path to be the true alignment and extract the rotational estimate φ_{robot} from the offset from its center pixel to the diagonal ($\Delta sectors$):

$$\hat{\varphi}_{robot} = \frac{360°}{80} \Delta sectors \tag{4}$$

Further, we try to estimate the noise by fitting again a path through the correlation matrix far away from the best-fitted path.

$$SNR = \frac{\sum\limits_{(x,y) \in minimumPath} Corr(x,y)}{\sum\limits_{(x,y) \in noisePath} Corr(x,y)} \tag{5}$$

The results of eq. 4 and eq. 5 can be found in figure 4.

3 Results

3.1 Environments

We selected five different environments to test our algorithm under a variety of circumstances. The first two experiments were conducted at home on a sunny afternoon and in an office environment[10] to measure performance under real-world circumstances. Furthermore, we conducted exhaustive tests on a (classical) 4-Legged field to test the performance under RoboCup circumstances. Then, we repeated the same measurements on a Midsize field[10], which could become interesting for future 11-against-11 games. Even more challenging, we took an Aibo to a real-world soccer field outdoors. This could be interesting especially for public demonstrations because until now all demonstrations had to be given in closed rooms with artificial lights.

---

[10]results omitted due to the lack of space, both located at Delft

(a) Robot standing on the trained spot (matching line is just the diagonal)

(b) Robot turned right by 45 degrees (matching line displaced to the left)
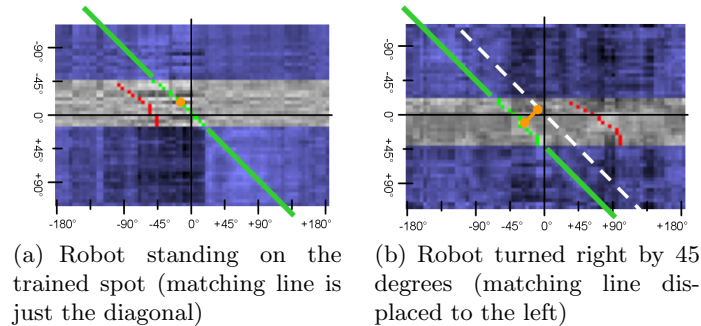
Fig. 4. Visualization of the alignment step while the robot is scanning with its head. The green line marks the minimum path (assumed true alignment) while the red line marks the second-minimal path (assumed peak noise). The grey line represents the diagonal, while the orange line illustrates the distance between the found alignment and the center diagonal ($\Delta sectors$).

3.2 Measured results

Figure 5(a) illustrates the results of a rotational test in a normal living room. As the error in the rotation estimates ranges between -4.5 and +4.5 degrees, we may assume an error in alignment of a single sector; moreover, the size of the confidence interval can be translated into one and two sectors respectively, which corresponds to the maximal angular resolution of our approach.

The next experiments were conducted on different types of soccer fields, exemplarily we present the measurements recorded on a real outdoor soccer field (fig. 5(b)) and a regular 4-Legged field indoor (fig. 6(a)). We developed a suitable visualization that resembles a magnetic field where we can display estimated rotations and confidence ranges in an intuitive way. The direction of an arrow shows the estimated rotation (with respect to the trained spot) and the grey arc shows the 80%-confidence interval around this estimate. Interestingly, it can be seen that the (imaginary) elongations of the arrows all run together in a single spot corresponding to the intersection point with the panoramic index.

In both cases it can be seen intuitively that the rotation estimates are within acceptable range, in the sense that all arrows point in the direction of the original 0-sector. This can also be shown quantitatively (see figure 6(b))): both the rotational error and the width of the confidence interval increase slowly and in a graceful way when the robot is moved away from the training spot.

4 Conclusion

Although at first sight the algorithm seems to rely on specific texture features of the surrounding surfaces, in practice no dependency could be found. This can be explained by two reasons: firstly, as the (vertical) position of a color transition is not used anyway, the algorithm is quite robust against (vertical) scaling.
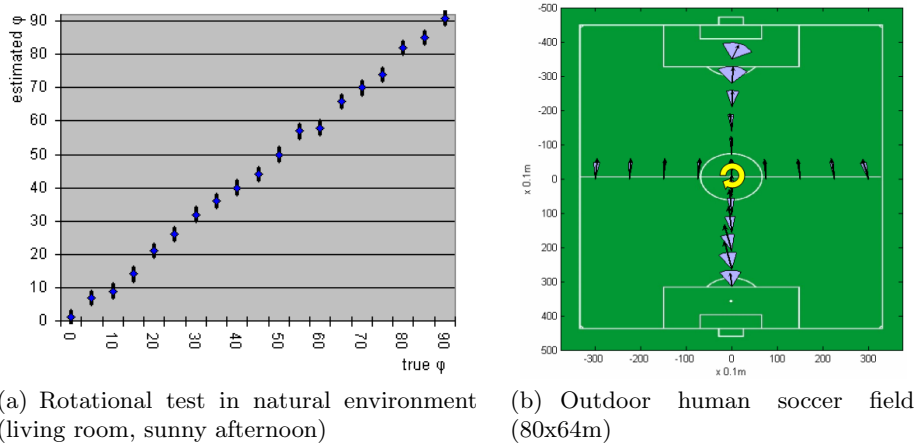
(a) Rotational test in natural environment (living room, sunny afternoon)

(b) Outdoor human soccer field (80x64m)

Fig. 5. Results of a rotational experiment conducted in a regular living room (left picture) and a translational test conducted on an outdoor soccer field.

Secondly, as the algorithm aligns on many color transitions in the background (typically more than a hundred in the same sector), the few color transitions produced by objects in the foreground (like beacons and spectators) have a minor impact on the match (because their sizes relative to the background are comparably small).

The lack of absolute position estimates seems to be a clear drawback with respect to the other methods, but bearing information alone can already be very useful for certain applications. For example, an attacking robot can highly benefit from a robust bearing estimation towards the goal. With this bearing estimating the robot can rush into the right direction. After 3 seconds the robot has to shoot, at that moment an additional distance estimation could be advantageous.

Further, this approach of panoramic localization is actually interesting for a broader spectrum than soccer. The requirements for both the robot as well as for its environment are quite moderate (on a Sony Aibo ERS7, the computation time is below 20ms/frame). The robot itself needs only a simple camera and medium computational power, while most natural environments (both indoors and outdoors) carry, as shown, enough panoramic information the algorithm can lock on to. Therefore, this method becomes for example interesting for the newly established RoboCup @ Home league, where fast localization information is needed in natural but completely unknown environments.

As the training on a single spot can be completed in less than one minute on a Sony Aibo in an arbitrary place, small demonstrational games (for example a striker versus a goalkeeper) could be set up more easily especially at non-prepared places. Progress in this domain facilitates the advancement of mobile robots - and thereby robotics research itself - into more natural environments.
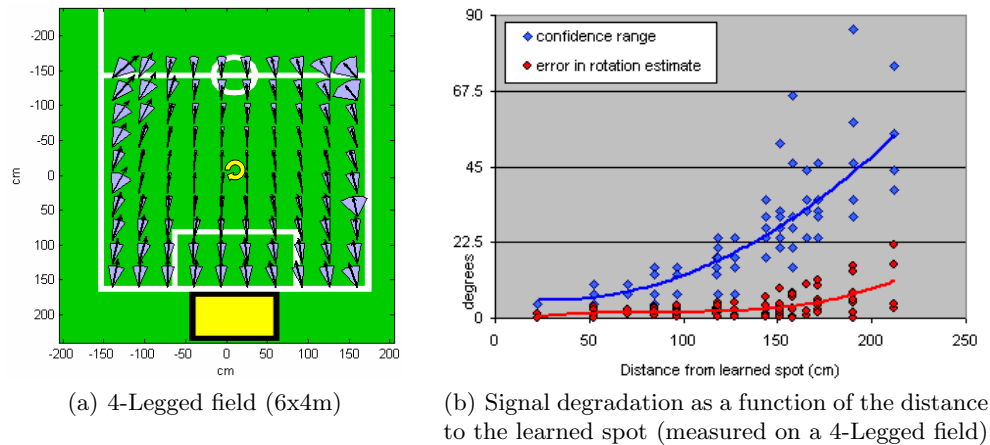
(a) 4-Legged field (6x4m)

(b) Signal degradation as a function of the distance to the learned spot (measured on a 4-Legged field)

Fig. 6. Test results on an official field of the 4-Legged league.

References

1. Röfer, T., et al.: GermanTeam RoboCup 2005. Online, 247 pages (2005)
2. Sturm, J., Visser, A., Wijngaards, N.: Dutch aibo team: Technical report robocup 2005. Technical report, Dutch Aibo Team (2005)
3. Quinlan, M.J., et al.: Nubots team report. Technical report, University of Newcastle (2005)
4. Gutmann, J.S., Fox, D.: An experimental comparison of localization methods continued. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02), Lausanne, Switzerland (2002)
5. Terwijn, B., Porta, J., Kröse, B.: A particle filter to estimate non-markovian states. In Groen, F., ed.: International Conference on Intelligent Autonomous Systems, IAS'04, IOS Press (2004) 1062–1069
6. Jogan, M., Ales, L.: Robust localization using panoramic view-based recognition. In: Proceedings 15th International Conference on Pattern Recognition ICPR'00, Barcelona, Spain (2000) 136–139
7. Pajdla, T.: Robot localization using shift invariant representation of panoramic images (research report k335-cmp-1998-170). Technical report, Czech Technical University, Prague (1998)
8. Bunschoten, R.: Mapping and Localization from a Panoramic Vision Sensor. PhD thesis, Universiteit van Amsterdam (2003)
9. Gonzalez, J., Lacroix, S.: Rover localization in natural environments by indexing panoramic images. In: International Conference on Robotics and Automation 2002, Washington, DC (USA) (2002)
10. Gross, H.M., et al.: Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In Groen, F., ed.: IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS 2003), IEEE omnipress (2003) 1505–1511
11. Verbeek, J.: Mixture models for clustering and dimension reduction. PhD thesis, Universiteit van Amsterdam (2004)