UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Robot companion localization at home and in the office

Visser, A.; Sturm, J.; Groen, F.C.A.

Publication date
2006

Published in
Proc. 18th Dutch-Belgian Artificial Intelligence Conference, BNAIC'06

Link to publication

Citation for published version (APA):
Visser, A., Sturm, J., & Groen, F. C. A. (2006). Robot companion localization at home and in the office. In *Proc. 18th Dutch-Belgian Artificial Intelligence Conference, BNAIC'06* (pp. 347-354)

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:15 Oct 2026

Robot companion localization at home and in the office

Arnoud Visser Jürgen Sturm Frans Groen

Intelligent Autonomous Systems, Universiteit van Amsterdam

http://www.science.uva.nl/research/ias/

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 to explore and map extensive public areas with large holonomic robots on wheels, less attention is paid on the localization of a small robot companion in a confined environment as a room in office or at home. In this article, a localization algorithm for the popular Sony entertainment robot Aibo inside a room is worked out. This algorithm can provide localization information based on the natural appearance of the walls of the room. The algorithm starts making a scan of the surroundings by turning the head and the 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 is used to determine the orientation (including a confidence value) relative to the learned spot for other points in the room. When multiple spots are learned, an absolute position estimate can be made. The applicability of this kind of localization is demonstrated in two environments: at home and in an office.

1 Introduction

1.1 Context

Humans orientate easily in their natural environments. To be able to interact with humans, mobile robots also need to know where they are. Robot localization is therefore an important basic skill of a mobile robot, as a robot companion like the Aibo. Yet, the Sony entertainment software contained no localization software until the latest release[1]. Still, many other applications for a robot companion - like collecting a news paper from the front door - strongly depend on fast, accurate and robust position estimates. As long as the localization of a walking robot, like the Aibo, is based on odometry after sparse observations, no robust and accurate position estimates can be expected.

Most of the localization research with the Aibo has concentrated on the RoboCup. At the RoboCup[2] artificial landmarks as colored flags, goals and field lines can be used to achieve localization accuracies below six centimeters [6, 8].

The price that these RoboCup 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.

The interest of the RoboCup community in more general solutions has been (and still is) growing over the past few years. The almost-SLAM challenge[3] of the 4-Legged league is a good example of the state-of-the-art in this community. For this challenge additional landmarks with bright colors are placed around the borders on a RoboCup field. The robots get one minute to walk around and explore the field. Then, the normal beacons and goals are covered up or removed, and the robot must then move to a series of five points on the field, using the information learnt during the first

[1] Aibo Mind 3 remembers the direction of its station and toys relative to its current orientation

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

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

minute. The winner of this challenge [6] reached the five points by using mainly the information of the field lines. The additional landmarks were only used to break the symmetry on the soccer field.

A more ambitious challenge is formulated in the newly founded RoboCup @ Home league[4]. In this challenge the robot has to safely navigate toward objects in the living room environment. The robot gets 5 minutes to learn the environment. After the learning phase, the robot has to visit 4 distinct places/objects in the scenario, at least 4 meters away from each other, within 5 minutes.

1.2 Related Work

Many researchers have worked on the SLAM problem in general, for instance on panoramic images [1, 2, 4, 5]. These approaches are inspiring, but only partially transferable to the 4-Legged league. The Aibo is not equipped with an omni-directional high-quality camera. The camera in the nose has only a horizontal opening angle of 56.9 degrees and a resolution of 416 x 320 pixels. Further, the horizon in the images is not a constant, but depends on the movements of the head and legs of the walking robot. So each image is taken from a slightly different perspective, and the path of the camera center is only in first approximation a circle. Further, the images are taken while the head is moving. When moving at full speed, this can give a difference of 5.4 degrees between the top and the bottom of the image. So the image seems to be tilted as a function of the turning speed of the head. Still, the location of the horizon can be calculated by solving the kinematic equations of the robot. To process the images, a 576 Mhz processor is available in the Aibo, which means that only simple image processing algorithms are applicable. In practice, the image is analyzed by following scan-lines with a direction relative the calculated horizon. In our approach, multiple sectors above the horizon are analyzed, with in each sector multiple scan-lines in the vertical direction. One of the general approaches [3] divides the image in multiple sectors, but this image is omni-directional and the sector is analyzed on the average color of the sector. Our method analysis each sector on a different 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^o circular panorama image of its environment while it is in the learning phase. After that, it should align each new image with the stored panorama, and from that the robot should be able to derive its relative orientation (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 at the point where the panorama was originally learned [11].

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[5] on an Aibo. So, a reduced circular 360^o panorama model of the environment is learned. Figure 1 gives a quick overview of the algorithm's main components.

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 [9]. 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)). These initialization steps are worked out in more detail in [10].

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

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

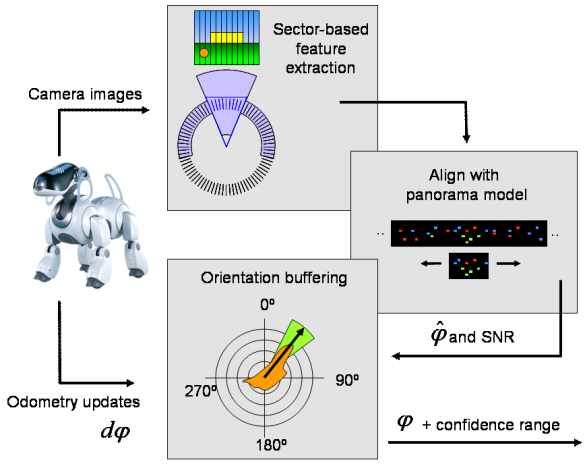
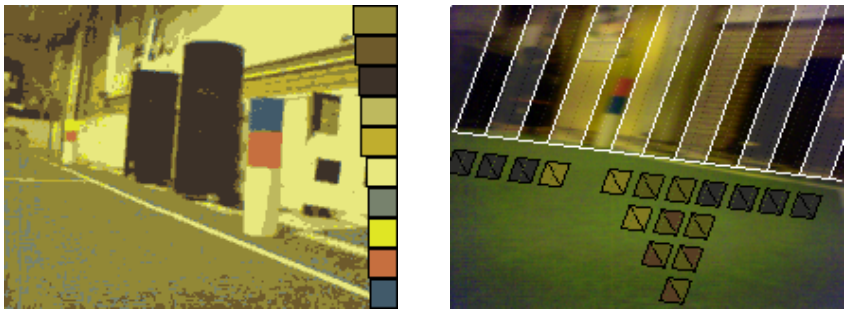


Figure 1: Architecture of our algorithm



(a) Unsupervised learned color segmentation.

(b) Sectors and frequent color transitions visualized.

Figure 2: Image processing: from the raw image to sector representation. This conversion consumes approximately 6 milliseconds/frame on a Sony Aibo ERS7.

2.1 Sector signature correlation

Every incoming image is now divided into its corresponding sectors[6]. The sectors are located above the calculated horizon, which is generated by solving the kinematics of the robot. 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 the histograms of 10x10 transition frequencies per sector, which we subsequently discretize into 5 logarithmically scaled bins. In figure 2(b) we displayed the most frequent color transitions for each sector. Some sectors have multiple color transitions in the most frequent bin, other sectors have a single or no dominant color transition. This is only visualization; not only the most frequent color transitions, but the frequency of all 100 color transitions are used as characteristic feature of the sector.

In the **learning phase** we estimate all these 80x(10x10) distributions[7] by turning the head and body of the robot. We define a single distribution for a currently perceived sector by

$$P_{current}(i,j,bin) = \begin{cases} 1 & discretize(freq(i,j)) = bin \\ 0 & otherwise \end{cases} \quad (1)$$

where i, j are indices of the color classes and bin one of the five frequency bins. Each sector is seen multiple times and the many frequency count samples are combined into a distribution learned

[6] 80 sectors corresponding to 360^o; with an opening angle of the Aibo camera of approx. 50^o, this yields between 10 and 12 sectors per image (depending on the head pan/tilt)

[7] 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.

for that sector by the equation:

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

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

$$Corr(P_{current}, P_{learned}) = \prod_{\substack{i,j \in colorClasses, \\ bin \in frequencyBins}} P_{learned}\left(i,j,bin\right) \cdot P_{current}\left(i,j,bin\right) \tag{3}$$

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. This minimal path is indicated as green dots in figure 3. The path is extended to a green line for the sectors that are not visible in the latest perceived image.

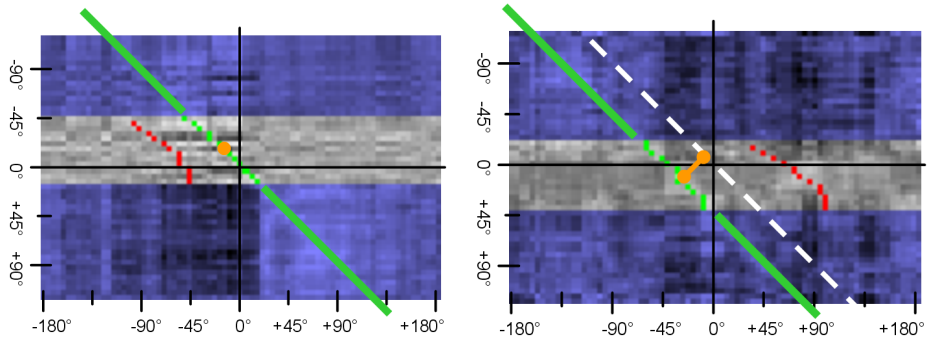
We consider the fitted path to be the true alignment and extract the rotational estimate φ_{robot} from the offset from its center pixel to the diagonal ($\Delta sectors$):

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

This rotational estimate is the difference between the solid green line and the dashed white line in figure 3, indicated by the orange halter. 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 noise path is indicated in figure 3 with red dots.



(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)

Figure 3: Visualization of the alignment step while the robot is scanning with its head. The green solid line marks the minimum path (assumed true alignment) while the red line marks the second-minimal path (assumed peak noise). The white dashed line represents the diagonal, while the orange halter illustrates the distance between the found alignment and the center diagonal ($\Delta sectors$).

2.3 Position Estimation with Panoramic Localization

The algorithm described in the previous section can be used to get a robust bearing estimate together with a confidence value for a single trained spot. As we finally want to use this algorithm to obtain full localization we extended the approach to support multiple training spots. The main idea is that the robot determines to which amount its current position resembles with the previously learned spots and then uses interpolation to estimate its exact position. As we think that this approach could also be useful for the RoboCup @ Home league (where robot localization in complex environments like kitchens and living rooms is required) it could become possible that we finally want to store a comprehensive panorama model library containing dozens of previously trained spots (for an overview see [1]).

However, due to the computation time of the feature space conversion and panorama matching, per frame only a single training spot and its corresponding panorama model can be selected. Therefore, the robot cycles through the learned training spots one-by-one. Every panorama model is associated with a gradually changed confidence value representing a sliding average on the confidence values we get from the per-image matching.

After training, the robot memorizes a given spot by storing the confidence values received from the training spots. By comparing a new confidence value with its stored reference, it is easy to deduce whether the robot stands closer or farther from the imprinted target spot.

We assume that the imprinted target spot is located somewhere between the training spots. Then, to compute the final position estimate, we simply weight each training spot with its normalized corresponding confidence value:

$$position_{robot} = \sum_i position_i \frac{confidence_i}{\sum_j confidence_j} \tag{6}$$

This should yield zero when the robot is assumed to stand at the target spot or a translation estimate towards the robot's position when the confidence values are not in balance anymore.

To prove the validity of this idea, we trained the robot on four spots on regular 4-Legged field in our robolab. The spots were located along the axes approximately 1m away from the center. As target spot, we simply chose the center of the field. The training itself was performed fully autonomously by the Aibo and took less than 10 minutes. After training was complete, the Aibo walked back to the center of the field. We recorded the found position and kidnapped the robot to an arbitrary position around the field and let it walk back again.

Please be aware that our approach for multi-spot localization is at this moment rather primitive and has to be only understood as a proof-of-concept. In the end, the panoramic localization data from vision should of course be processed by a more sophisticated localization algorithm, like a Kalman or particle filter (last not least to incorporate movement data from the robot).

3 Results

3.1 Environments

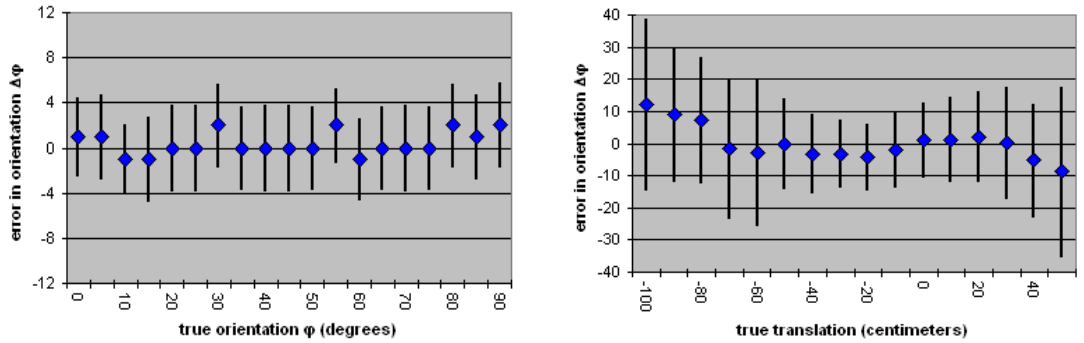
We selected four different environments to test our algorithm under a variety of circumstances. The first two experiments were conducted at home and in an office environment[8] to measure performance under real-world circumstances. The experiments were performed on a cloudy morning, sunny afternoon and late in the evening. Furthermore, we conducted exhaustive tests in our laboratory. Even more challenging, we took an Aibo outdoors (see [7]).

3.2 Measured results

Figure 4(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[9]; moreover, the size of the confidence interval can be translated into maximal two sectors, which corresponds to the maximal angular resolution of our approach.

[8] XX office, DECIS lab, Delft

[9] full circle of 360^0 divided by 80 sectors



(a) Rotational test in natural environment (living room, sunny afternoon)

(b) Translational test in natural environment (child's room, late in the evening)

Figure 4: Typical orientation estimation results of experiments conducted at home. In the rotational experiment on the left the robot is rotated over 90 degrees on the same spot, and every 5 degrees its orientation is estimated. The robot is able to find its true orientation with an error estimate equal to one sector of 4.5 degrees. The translational test on the right is performed in a child's room. The robot is translated over a straight line of 1.5 meter, which covers the major part of the free space in this room. The robot is able to maintain a good estimate of its orientation; although the error estimate increases away from the location where the appearance of the surroundings was learned.

Figure 4(b) shows the effects of a translational dislocation in a child's room. The robot was moved onto a straight line back and forth through the room (via the trained spot somewhere in the middle). The robot is able to estimate its orientation quite well on this line. The discrepancy with the true orientation is between +12.1 and -8.6 degrees, close to the walls. This is also reflected in the computed confidence interval, which grows steadily when the robot is moved away from the trained spot. The results are quite impressive for the relatively big movements in a small room and the resulting significant perspective changes in that room.

Figure 5(a) also stems from a translational test (cloudy morning) which has been conducted in an office environment. The free space in this office is much larger than at home. The robot was moved along a 14m long straight line to the left and right and its orientation was estimated. Note the error estimate stays low at the right side of this plot. This is an artifact which nicely reflects the repetition of similarly looking working islands in the office.

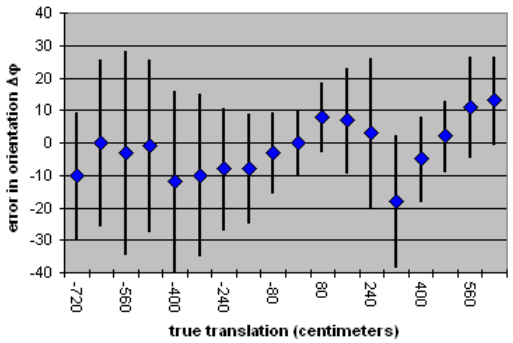
In both translational tests it can be seen intuitively that the rotation estimates are within acceptable range. This can also be shown quantitatively (see figure 5(b)): both the orientation error and the confidence interval increase slowly and in a graceful way when the robot is moved away from the training spot.

Finally, figure 6 shows the result of the experiment to estimate the absolute position with multiple learned spots. It can be seen that the localization is not as accurate as traditional approaches, but can still be useful for some applications (bearing in mind that no artificial landmarks are required). We recorded repeatedly a derivation to the upper right that we think can be explained by the fact that different learning spots don't produce equally strong confidence values; we believe to be able to correct for that by means of confidence value normalization in the near future.

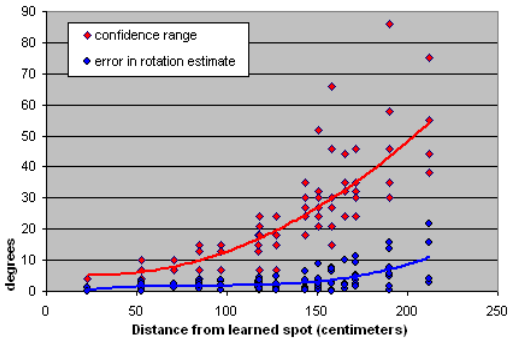
4 Conclusion

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. 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 comparatively small).

The lack of an accurate 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.



(a) Translational test in natural environment (office, cloudy morning)



(b) Signal degradation as a function of the distance to the learned spot (measured in the laboratory)

Figure 5: Challenging orientation results. On the left a translational test in office environment over 14 meters along a line 80 centimeters from the learned spot (only one). A translation to the left of the office increases the error estimate increases, as expected. When translating to the right of the office to the orientation estimate oscillates, but the error estimate stays low. This is due to repeating patterns in the office, after 4 meters there is another group of desks and chairs which resembles the learned appearance. Normally the error estimate increases as a second order polynome of the distance, as demonstrated with our qualitative measurements on the right. For this measurements the rotation estimate was recorded on a grid of 99 locations regulary distributed over our laboratory.

For example, a robot companion can highly benefit from a robust bearing estimation towards the front door. With this bearing estimating the robot can walk into the right direction. Only when the front door is nearly reached, and the robot has to search for a newspaper, additional distance estimation could be advantageous.

The requirements of this approach 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 carry, as shown, enough panoramic information the algorithm can lock on to. Therefore, this method becomes 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, robot research can move outside the laboratory into non-prepared places. Progress in this domain facilitates the advancement of mobile robots - and thereby robotics research itself - into more natural environments.

Acknowledgements

The Dutch AIBO Team is financially supported by DECIS Lab, University of Amsterdam, Technical University of Delft, Saxion University of Professional Education, University of Groningen and University of Utrecht. We are grateful for the contributions by all of our team members, working on RoboCup related research and other research. Special thanks to Olaf Booij, David Knibbe and Niek Wijngaards.

References

[1] R. Bunschoten. *Mapping and Localization from a Panoramic Vision Sensor.* PhD thesis, Universiteit van Amsterdam, November 2003.

[2] J. Gonzalez and S. Lacroix. Rover localization in natural environments by indexing panoramic images. In *International Conference on Robotics and Automation 2002*, Washington, DC (USA), 2002.

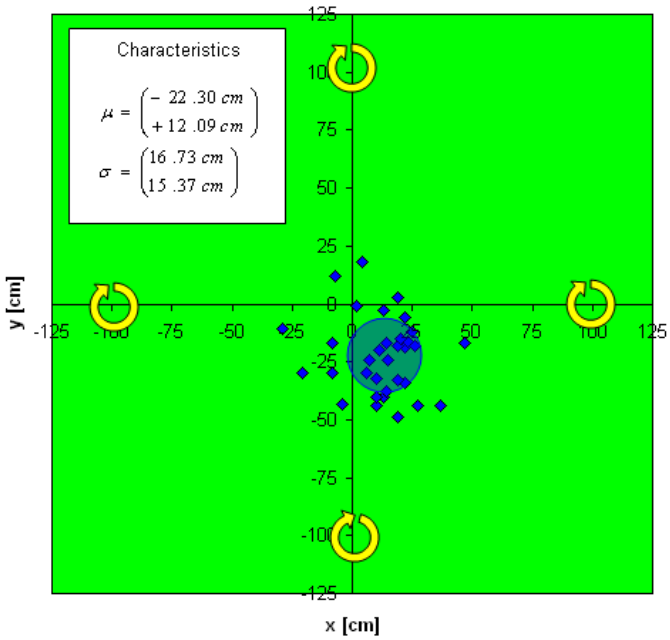


Figure 6: The position estimation after learning the appearance of the surroundings on 4 spots (indicated with yellow arrows). The small dots indicate the distribution of positions reached every time the robot tried to return to the central spot after been placed randomly somewhere on the field.

[3] Horst-Michael Gross et al. Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In F.C.A. Groen, editor, *IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS 2003)*, pages 1505–1511. IEEE omnipress, 2003.

[4] Matjaz Jogan and Leonardis Ales. Robust localization using panoramic view-based recognition. In *Proceedings 15th International Conference on Pattern Recognition ICPR'00*, pages 136–139, Barcelona, Spain, 2000.

[5] T. Pajdla and V.Hlavac. Zero phase representation of panoramic images for image based localization. In *Proceedings of the 8th International Conference on Computer Analysis of Images and Patterns: CAIP'99*, volume 1689 of *Lecture Notes on Computer Science*, pages 550–557. Springer Verlag, September 1999.

[6] T. Röfer et al. *GermanTeam RoboCup 2005*. 247 pages, 2005.

[7] J. Sturm, P. van Rossum, and A. Visser. Panoramic localization in the 4-legged league. In *Proc. 10th RoboCup International Symposium*, Bremen, June 2006. To be published in the Lecture Notes on Artificial Intelligence series, Springer Verlag, Berlin.

[8] J. Sturm, A. Visser, and N. Wijngaards. Dutch aibo team: Technical report robocup 2005. Technical report, Dutch Aibo Team, October 2005.

[9] J.J. Verbeek. *Mixture models for clustering and dimension reduction*. PhD thesis, Universiteit van Amsterdam, December 2004.

[10] Arnoud Visser, Paul van Rossum, Joost Westra, Jürgen Sturm, Dave van Soest, and Mark de Greef. Dutch aibo team at robocup 2006. In *Proceedings CD RoboCup 2006, Bremen, Germany*, June 2006.

[11] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection, June 2003.