Intelligent Autonomous Systems 6, E. Pagello et al. (eds), pp 589-595, IOS Press, 2000

An Efficient Representation of the Robot's Environment

Ben Kröse
Real World Computing Partnership, Autonomous Learning Functions Lab. SNN
Department of Computer Science, University of Amsterdam
Kruislaan 403, NL-1098 SJ Amsterdam, The Netherlands
krose@science.uva.nl

Abstract. In this paper we describe a method for modeling the environment of a mobile robot. Recently a number of methods have been presented which implicitly model the environment as the relation between the robot position and the sensor observations instead of using an explicit geometric 'CAD'-like representation. Localization now is a sort of lookup process. We go one step further, and show that if range information is available, the number of prototypes which is needed to model the environment is reduced drastically.

1 Introduction

An internal representation of the environment is needed for optimal mobile robot navigation. Traditionally such a model is represented as a geometric model in the task space of the robot, indicating admissible and non-admissible areas. The robot has to know its location within such a model. Odometry is accurate on short distances but fails to give an accurate position estimate if longer distances have been traveled. Sensors which observe the environment have to be used for localization. One approach is to find the location which has an optimal fit between the sensor measurements and the global model (*model matching methods*). This approach requires an extensive modeling which is only feasible if sufficient prior knowledge about the environment is available.

Other methods try to model a direct relation between the location of the robot and the observations. Since no geometric model of the environment is built, but only a model of the sensor values ('appearances') as a function of the robot position these methods are often called *appearance based methods*. The building of such a model requires a data set consisting of a large number of observations and corresponding robot positions: the method is a supervised method. Recently much work in this are has been presented. Probabilistic models [9, 5] have been proposed, neural networks [7], radial basis functions [10] or look-up tables [1]. The observations could be visual [6, 8, 4], range scans [1] or linear or nonlinear features derived from these.

The problem with supervised methods is that the robot has to be positioned at a large number of known positions. A solution to this problem was presented in [1], where based on a number of measured range scans, new 'synthetic' scans were generated using the measured data. This can be done by first constructing a 'composite' scan, which is then re-sampled to construct synthetic scans. All these synthetic scans are now used to build the model on the basis of which the localization can take place. Note that -although the dimensionality of the measurements is reduced with a Principal

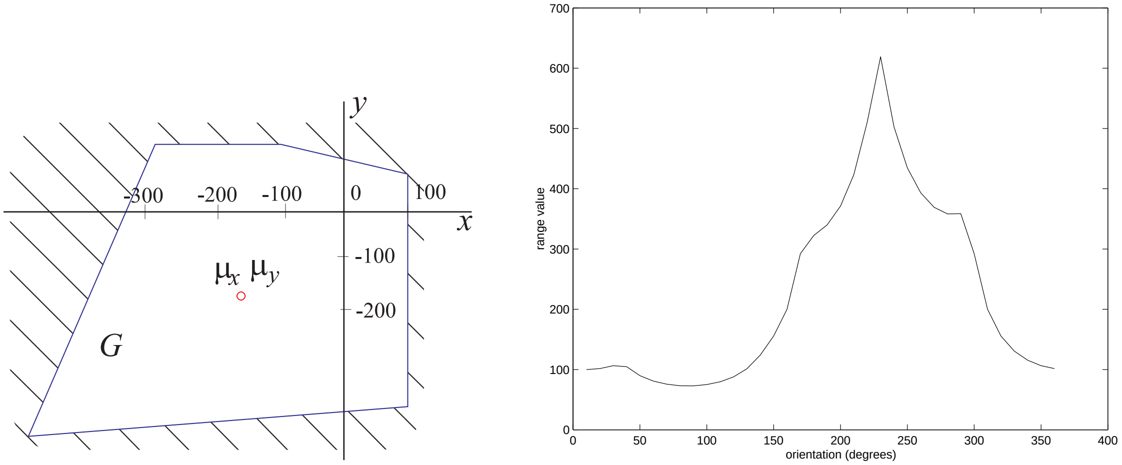

Figure 1: Convex room with the robot at the position $(0, 0)$. Right we see the range scan taken from the robot, giving the distance to the nearest object as a function of the orientation of the sensor. Zero degrees is in the direction of the positive X-axis. The circle in the left figure depicts the estimated center of the room.

Component Analysis- all measurements have to be stored and many distances have to be computed, particularly if a nearest neighbor method is applied.

We introduce a method to represent the environment of a robot with less examples. The method is an *unsupervised* method, in which no information about absolute position of the robot is used. The localization is always relative to some location which is denoted as a *prototype* location. The method is based on the same principle that [1] used to generate a dense field, with the difference that we do not generate synthetic scans but we check whether a new scan, from an unknown location, can be derived from one of a restricted number of prototype scans. First we will explain our method with a localization task in a convex space, then we show that we can distinguish between different types of convex spaces and finally we will will show the localization in a more complex environment.

2 Convex spaces

We assume that the robot has available range measurements to nearest surfaces in the environment. These can be obtained with for example optic flow (taking two images at different positions), ultra-sonic range sensors or laser range sensors. The latter are rather common and provide a one-dimensional range scan. Suppose we have such a sensor which provides a scan $\mathbf{r}$ consisting of R range measurements taken under equidistant angle.

In appearance-based methods, the model of a convex room as depicted in figure 1 typically consists of a large number of stored scans and corresponding positions ([1] use 2880 scans per square meter, we have done experiments with about 100 image-position pairs per square meter [5]) . However, for a convex room a *single* range scan, taken from an arbitrary position in that room, provides all the information about the *shape* of the room (which can be used to tell the robot in which room it is) and about the robot's *position* relative to the center of the room.

Instead of representing the room with *many* prototypes we represent the room with a *single* 'prototypical' range scan, taken from the center of the room. Two questions have to be answered:

- How can the center of the room be estimated from a range scan from an arbitrary position?
- How can we warp this range vector to the scan which would have been perceived from the center of the room?

2.1 Estimation of the center point

Suppose the convex environment is represented by some area G:

$$G : \{x, y | f(x, y) < 0\}. \tag{1}$$

As the center of the room we define the expected value over all admissible values. For the x this is:

$$\begin{aligned} \mu_x &= \int\int x f(x,y) dx dy \\ &= \frac{1}{A}\int_G\int x dx dy \end{aligned} \tag{2}$$

We can write the convex room in polar coordinates:

$$H : \{r, \phi | r < r(\phi), 0 \leq \phi < 2\pi\}. \tag{3}$$

so that for the expected x we can write (using $x = r\cos\phi$ and a change of integration variables):

$$\begin{aligned} \mu_x &= \frac{1}{A}\int_H\int r\cos\phi r dr d\phi \\ &= \frac{1}{A}\int_0^{2\pi}\int_0^{r(\phi)} r^2 dr \cos\phi d\phi \\ &= \frac{1}{A}\int_0^{2\pi} \frac{1}{3} r(\phi)^3 \cos\phi d\phi. \end{aligned} \tag{4}$$

If we have R uniformly distributed range measurements with an interval $\Delta\phi$ we can approximate this with

$$\mu_x \approx \frac{1}{A}\sum_{i=1}^{R} \frac{1}{3} r_i^3 \cos\phi_i \Delta\phi. \tag{5}$$

In the same way we can estimate μ_y. In figure 1 the estimated center of the room is depicted derived from a range scan based on 36 range measurements taken at intervals of 10 degrees. So for convex rooms the robot can determine its location relative to the center of the room from a single range scan.

2.2 Scan warping

Now we know the relative position of the center of the room we can warp the range scan taken from an arbitrary position to the range scan taken in the center. First we write the R range measurements as points in the coordinate system of the robot. We then write these points in the coordinate system with μ_x, μ_y as origin (this is a simple translation) and then we linearly interpolate and re-sample to get an equidistant range scan.

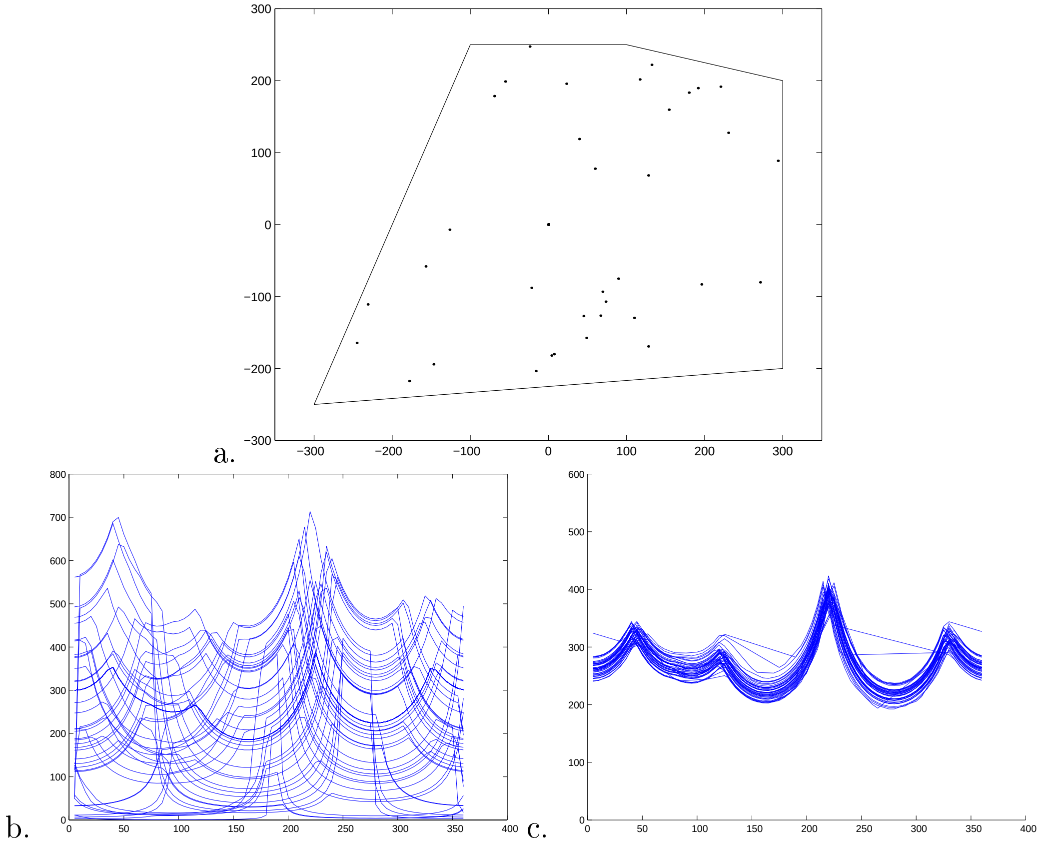

Figure 2: a.) The same room as the previous figure and 50 random positions. b.) 50 range scans from those positions. c.) The warped range scans.

In figure 2 a. we see the same room as in figure 1, with the dots depicting 50 random positions in the room. Figure 2 b. shows the range scans taken from these positions. Figure 2 c. shows the warped scans. We see that all scans are warped to approximately the same prototypical scan. Some problems with warping arise if the robot is positioned very close to a wall (see Figure 2 a). Because in these situations the sampling of the 'opposite' wall is very sparse the interpolation between the warped range measurements causes inaccuracies.

3 Localization: "Where" and "what"

Suppose the environment now consists of a restricted number of convex spaces, as shown in figure 3a. (Note that this is not a very useful environment because the robot can not go from one room to another, but it is for illustration). If the robot has to model this environment the only thing it has to learn are the prototypical range scans for each individual space.

In a learning phase we randomly choose a position in the room, estimate the center of the room and warp the range scan to the prototypical scan. Next we again select a random position and compute the warped scan. If the difference between the two scans is below a certain threshold we believe that the two positions are in the same convex room and the prototypical scan and the center of the room remain the same. If not, the robot is believed to be in a new room and a new prototype and center is defined. Figure 3b. shows the warped scans from the positions in the left room while c. shows the warped scans from the positions in the right room. They clearly differ. In this case

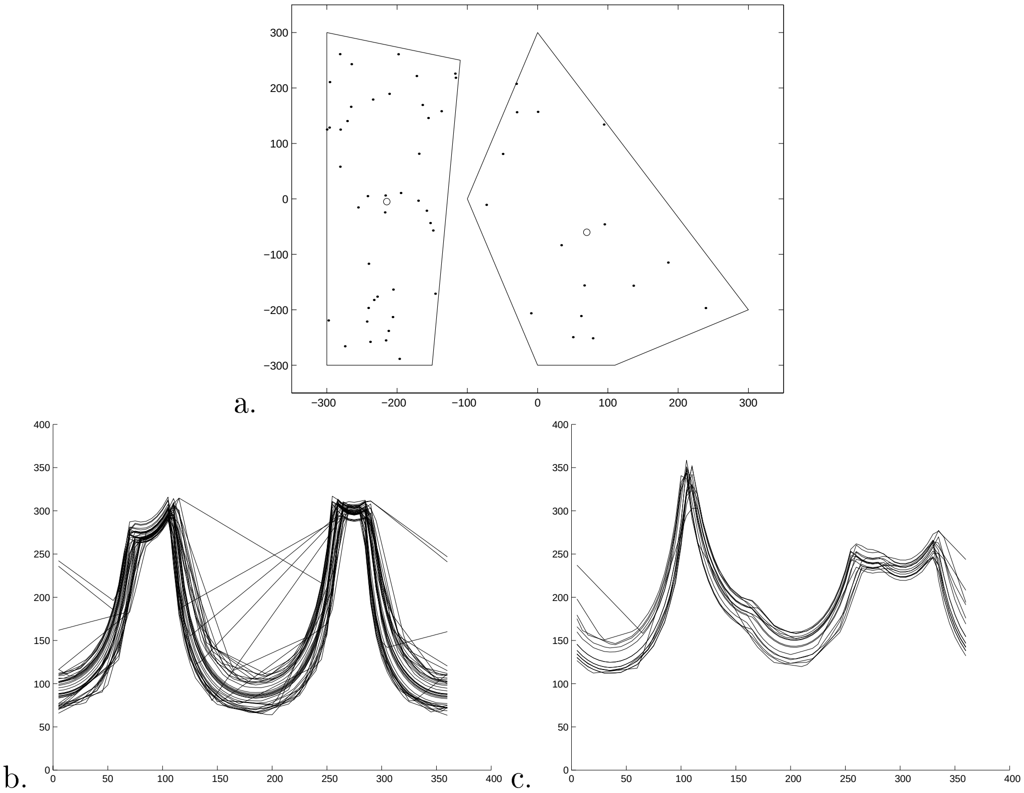

Figure 3: a.) Random robot positions in two separated rooms. b.) The warped scans from the positions in the left room . c.) The warped range scans from the positions in the right room.

we found two centers (depicted as circles in figure 3a.) and two prototypical scans.

Localization now involves two steps: first it detects *where* (relative to the center of the room it is), and subsequently (after warping its range scan) detects *what* room it is in.

4 More complex environments

As discussed in the previous section most realistic indoor environments do not consist of a collection of separated rooms but have connections between the spaces, which introduce non-convexities. An example is given in Figure 4. We followed the procedure described in the previous section to model this environment: about 150 random locations were generated and prototypical scans and their positions were derived from the data. The centers (the locations of the prototypical scans) are shown as small circles. We see that two of the prototypes are placed in the center of the two parts of the environment. However, the algorithm also places prototypes near the connection between the two parts. Near this connection the changes in the range profile can not be contributed to one single convex space and new prototypes are needed.

5 Navigation

The goal of the robot is usually to move to some sort of desired state. Because we work with an *unsupervised* method we can not express the desired state as a position in an absolute coordinate frame. We have to express the desired state in the sensor domain. From the desired state (for example desired range scan) we can derive the desired part

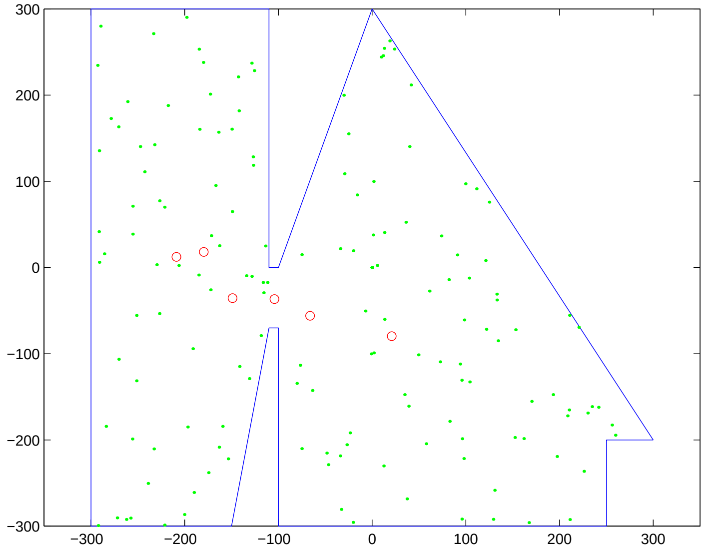

Figure 4: A more complex environment. The positions of the prototypes are depicted as small circles.

of the environment where the robot has to go to (desired prototype). We also know the actual sensor state of the robot and thus the actual prototype. For navigation we need to know the topological relations between the prototypes. Or, more accurate, which actions need to be taken to get from one prototype to another. This can be estimated in the learning phase by navigation of the robot. We used a similar method (Markov Decision Process) in our earlier work [3] which gave good results.

6 Discussion

The main idea behind the paper is that, because we have range information available, we can predict range scans at other locations. We showed that by this procedure we can describe convex rooms with a single range scan and do not need some learned model which gives the relation between robot displacement and changes in the sensor domain: *all information is in the range scan.* The idea works perfect in convex spaces but not perfect in realistic non convex spaces. We are now investigating whether approaches from computational geometry (for example the 'visibility skeleton', presented in [2]) can be used.

The main interest in our group is visual navigation, for which we use panoramic images from an omnidirectional vision system [5]. In our earlier work we used visual representations which had to store many images. From this paper it is clear that if we have range information, a single omnidirectional image represents all information in a convex room. Our current work focusses on a representation of an environment which consists of a number of prototypical images and range vectors, and where we can use visual homing as navigation technique.

References

[1] Crowley, J.L., F. Wallner and B. Schiele "Position estimation using principal components of range data", Proc. 1998 IEEE Int. Conf on Robotics and Automation, pp 3121-3128.

[2] Oliver Karch and Hartmut Noltemeier, “Robot Localization-Theory and Practice”, Proc. IROS 97, pp 850-856, 1997.

[3] B.J.A. Kröse, M.Eecen, “A Self-Organizing Representation of Sensor Space for Mobile Robot Navigation”, Proc. of the 1994 IEEE IEEE/RSJ Int. Conf. on intelligent Robots and Systems, Munich, Germany, 1994, pp. 9-14.

[4] B. Kröse and R. Bunschoten. Probabilistic localization by appearance models and active vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2255– 2260, 1999.

[5] Ben Kröse, Roland Bunschoten, Nikos Vlassis, and Yoichi Motomura. Appearance based robot localization. In G. Kraetzschmar, editor, *IJCAI-99 workshop on Adaptive Spatial Representations of Dynamic Environments*, pages 53 – 58, 1999.

[6] S. Maeda, Y. Kuno, and Y. Shirai. Active navigation vision based on eigenspace analysis. In *Proc. IROS'97, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1018–1023, 1997.

[7] S. Oore, G. E. Hinton, and G. Dudek. A mobile robot that learns its place. *Neural Computation*, 9:683–699, 1997.

[8] F. Pourraz and J. Crowley. Continuity properties of the appearance manifold for mobile robot estimation. In *Proc SIRS'98*, Edinburgh, 1998.

[9] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.

[10] Nikos Vlassis and Ben Kröse. Robot environment modeling via principal component regression. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 677–682, 1999.