TOWARDS CONSTRUCTING AND USING SELFORGANIZING VISUAL ENVIRONMENT REPRESENTATIONS FOR MOBILE ROBOTS

Georg von Wichert *,[1] **Henning Tolle** *

* *Control Systems Theory & Robotics Laboratory, Institute of Control Engineering, Darmstadt University of Technology, Landgraf-Georg-Str. 4, D-64283 Darmstadt, Germany*
gvw/tolle@rt.e-technik.th-darmstadt.de

Abstract: Due to the upcoming applications in the field of service robotics mobile robots are currently receiving increasing attention in industry and the scientific community. Applications in the area of service robotics demand a high degree of system autonomy, which robots without learning capabilities will not be able to meet. Learning is required in the context of action models *and* appropriate perception procedures. Both are extremely difficult to acquire especially with high bandwidth sensors (e.g. video cameras) which are needed in the envisioned unstructured worlds. Selflocalization is a basic requirement for mobile robots. This paper therefore proposes a new methodology for image based selflocalization using a selforganized visual representation of the environment. It allows for the seamless integration of active and passive localization. *Copyright © 1998 IFAC*

Keywords: Mobile robots, Navigation, Vision, Scene analysis, Selforganizing systems

1. INTRODUCTION

The service robotics applications of the future demand a high degree of system autonomy also in unstructured (compared to industrial production sites) environments. In addition, these environments and their sensorical characteristics are hardly to be known in detail at the design time of the robot.

Intelligent service robots without learning capabilities will therefore not be able to suffice in these applications. It is important to note, that learning is not only required in the context of action models where a lot of work is being done (Thrun (1994),Cassandra et al. (1996) and others), but also in the context of appropriate perception procedures for extracting relevant information from the available sensor data. This is extremely difficult especially with high bandwidth sensors (e.g. video cameras). On the other hand powerful sensors are needed in the envisioned unstructured worlds.

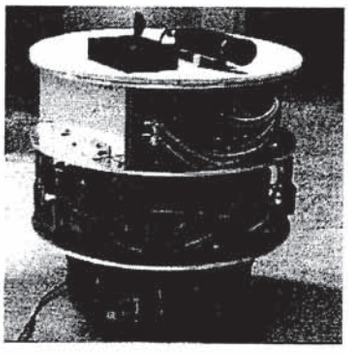

Fig. 1. The mobile robot ALEF

One basic requirement on mobile robots is the capability to localize themselves with respect to a global map of the work space. We therefore propose in this paper a new methodology for image based self localization using a selforganized visual representation of the environment. As a start, the next section will intro-

[1] This work was funded in part by the Deutsche Forschungsgemeinschaft under grant no. To-75/24-1.

Original publication in IFAC Proceedings Volumes (now IFAC-Papers online), https://doi.org/10.1016/S1474-6670(17)44116-4

duce our mobile robot ALEF, which was used for the experiments. Then section 3 will shortly discuss our approach to the problem of extracting relevant scene features from the video images perceived by the robot. The extracted scene features enable the robot to memorize and recognize distinct places in its environment. This capability is the foundation of a self-localization and navigation method presented in sections 4 and 5. Results from experiments are given along the presentation of our methods. The paper is summarized in section 6.

2. THE MOBILE ROBOT ALEF

Figure 1 shows our mobile robot ALEF. It is built upon a RWI-B12 platform and equipped with 24 ultrasonic sensors plus a CCD camera. A 486-based on board computer is responsible for basic tasks like collision avoidance (using the ultra-sonic sensors), odometric navigation and radio communication with an off board control computer. Via a separate analog radio link the video images are transferred to the same off board computer. Thus it is possible to perform all high level tasks off board.

3. SELFORGANIZATION OF PERCEPTUAL CAPABILITIES

The final goal of every image or scene analysis is to find a set of features which characterize the scene under examination and enable the system to compare it with other scenes perceived before. In 'natural' environments which can not be prepared for the robot's needs, it will at least be difficult or even impossible to define a standard set of features, which characterize the scene precisely enough and which can at the *same* time be extracted from the image with sensible effort. In addition, for service robots, it will not be possible to engineer each system for a specific environment, and the variety of possible environments makes it unrealistic to assume, that the feature extraction process can be defined in detail at the design time of the system.

We therefore propose, that the robot should be able to decide itself, which features are useful in a given situation. We formulate an unsupervised scene feature extraction process that enables the robot to create a visual representation of its environment by means of selforganization of its perceptual capabilities.

3.1 *Unsupervised Scene Feature Extraction*

The scene feature extraction is performed in three major steps:

(1) The first step consists mainly in extracting local pixel features. In case of a grey level image $s(x, y)$ these will be features computed from the local grey level distribution by means of a set of N_h feature extractors $h_i(x, y)$ with $i \in [1, N_h]$. We obtain N_h feature channels

$$s_h^i(x, y) = h_i(x, y) * *s(x, y). \quad (1)$$

For $h_i(x, y)$ we use texture energy features Laws (1980) on different scales (image pyramids), but other filters (e.g. Garbor Wavelets) could also be used. The single feature channels are then grouped into a multichannel image

$$\mathbf{S}_h(x, y) = \begin{bmatrix} s_h^1(x, y) \\ \vdots \\ s_h^{N_h}(x, y) \end{bmatrix} \quad (2)$$

When a color camera is available, these local pixel features can be derived from color cannels $\mathbf{S}(x, y) = [s_R(x, y), s_G(x, y), s_B(x, y)]^T$ instead (or in addition) by an appropriate color space transformation T_c (for an interesting physiologicaly motivated approach see Pomierski and Groß (1996)).

$$\mathbf{S}_h(x, y) = T_c\mathbf{S}(x, y) = T_c \begin{bmatrix} s_R(x, y) \\ s_G(x, y) \\ s_B(x, y) \end{bmatrix} \quad (3)$$

In both cases the local features should be independent from the absolute brightness, to suppress illumination effects.

(2) Step two essentially consists of a learning classification of $\mathbf{S}_h(x, y)$, the multichannel feature image computed in step one. With the pixelwise classifier $Cl(\cdot)$ using N_C classes c we get the classification result

$$s'(x, y) = Cl(\mathbf{S}_h(x, y)), s'(x, y) \in [1, N_C], \quad (4)$$

s' being an integer value. This results in an unsupervised segmentation of the input image, grouping areas of similar local appearance. Several methods can be considered to perform this quantization step, but a hierarchical arrangement of selforganizing feature maps Kohonen (1988) has proven to be well suited (see v. Wichert (1996)). The segmented image $s'(x, y)$ resulting from this procedure is robust to small variations of the camera (and thus also the robot) position. This robustness is crucial for the next step, which computes the desired scene features $\mathbf{x}(s'(x, y))$ from this segmentation of the image. Otherwise, these would not smoothly change as the robots moves so that existing neighborhood relations in the resulting "scene space" would get lost during the transformation to the feature space.

(3) As mentioned above the third step comprises the computation of a set of features which shall be used afterwards to robustly compare images from the video sensor. The basic assumption for this step is, that the distribution of the image segments computed in step two is a characteristic of the

scene under examination. Therefore we compute the scene features from this distribution. Again, a large variety of methods could be used here (local feature histograms,...) v. Wichert and Kleiner (1995), but good results have always been obtained using geometric moments $m_c^{p,q}$ of the segments belonging to each class c

$$m_c^{p,q} = \sum_x \sum_x x^p y^q \delta(c - s'(x,y)) w(x,y) \quad (5)$$

Where δ(x) is the Dirac-impulse function and $w(x,y)$ is a weighting function, that emphasises the center of the image and fades out pixels near the edges.

Geometric moments are often used as features for shape recognition problemsHu (1961). Their major advantage compared to other shape features (e.g. fourier descriptors) is that the segments do not have to be connected. This is a property that can not be guaranteed with our unsupervised segmentation method (Step 1 and 2).

The scene features up to order P and Q resulting from the third step are then grouped into scene feature vectors (**SFV**'s)

$$\mathbf{x}(s') = \left[m_1^{0,0}, ..., m_1^{P,Q}, ..., m_{N_C}^{0,0}, ..., m_{N_C}^{P,Q} \right]^T \quad (6)$$

They will later on – after some normalization – be used to compare different images without the need of any geometric model and any need to interpret the scenes represented by the video images. The neccessary computation requires a processing time of 1.2 sec on an Ultra-SPARC 1 workstation.

4. BUILDING MAPS FROM SCENE FEATURE VECTORS

The environment map used by the robot must contain all information necessary to navigate in the workspace. This comprises the representation of places and pathways as well as additional information the system might need to fulfill its task. Many learning systems use either grid-based maps (Moravec and Elfes (1985), Burgard et al. (1996)) or graph-like representations (Zimmer (1995), Kurz (1995), Simmons and Koenig (1995), Tolle and Kurz (1998)). Grid-based approaches can be appropriate for systems using one dimensional distance sensors (US, Lidar) and comparatively small [2] environments. Maps composed from places at which sensory information (images) was stored and path ways (experienced traversals between different places) as nodes and edges of a graph structure seem to be more natural in our context.

[2] This is of course limited only by the available memory.

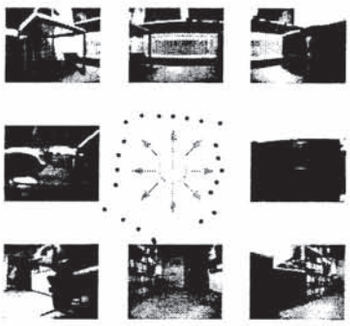

Fig. 2. Sensor data (vision, ultrasonics (·), odometric position) are used to represent a specific place. SFV's are stored instead of the original images.

4.1 *Representing Places and Pathways*

To represent places in our enironment we use an *omnidirectional place representation* which is generated by acquiring an omnidirectional view at every place represented by a node n in our graph-like map. This can be done taking a fixed number N_k of images $s_n^k(x,y)$ equidistant in all directions. The major advantage of such an omnidirectional representation is caused by the trivial fact, that – after taking all images at the same place – it is known that these images were acquired at the same position and which are the correct neighborhood relations with respect to the rotational axis of the robots three dimensional workspace. This reduces the amount of information to be inferred from experiments and thus one can expect that the map learning process is significantly simplified.

Along with the SFV's, also information from all other sensors (US, Odometry) is stored at each place. Fig. 2 gives an example of all the data used to represent a specific place in our environment. For each node $n \in [1, N_n]$ (N_n: Number of nodes in the map) the position $\mathbf{p}_n$ of the node, as derived from odometry, and the N_k SFV's $\mathbf{x}_n^k(s_n^k(x,y))$ are of relevance in this paper.

Pathways represent connections between places corresponding to experienced traversals and thus the topological structure of the environment. Information stored to represent them comprises mainly the mean positional difference of the adjacent places as measured by dead reckoning during past traversals.

4.2 *Localization in the Map – Where is the Robot?*

Having gathered all necessary information and having stored it appropriately in the map, it can now be used for robot selflocalization. The goal of the selflocalization process is to reposition the robot using the data stored in the map to compensate for a significant drift of the odometry caused by slippage of the wheels.

The optimum selflocalization process would be able to exactly specify the robots current position by means of sensor data gathered currently. However, this is very hard in practice, due to the ambiguity of the sensor

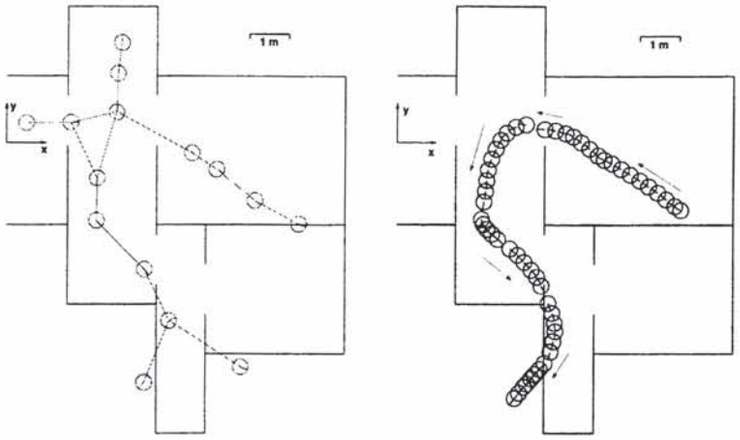

Fig. 3. Data from real experiments with ALEF. Left side: Places selected for representation in the map. The graph nodes are plotted at their cartesian position as derived from dead reckoning. Dead reckoning errors distort the vehicle's internal position hypothesis, which leads to nodes placed outside the actual rooms when plotting the map against a floor plan. Right side: Path traveled through the environment (dead reckoning).

images. Especially with the one dimensional distance sensors, often used in mobile robots, many places in the environment "look the same". This problem is sometimes also referred to as *perceptual aliasing*. During map learning, the map is built using position data from the dead reckoning process and at the same time the dead reckoning has to be corrected using the position data stored in the map. The major consequence is that the map building process tends to become unstable for large environments, because the system needs its cartesian position to distinguish between areas in the map that cannot be uniquely recognized using external sensors only. As a result of this, the capability to distinguish successfully between as many different places as possible – only by means of the external sensors – is crucial for map learning. It is therefore interesting to see how much our selforganized *visual* perception process (Section 3) can contribute to the solution of this problem. The left part of figure 3 shows a map acquired on our office floor. Small circles indicate places selected for inclusion in the map. The right part shows a path [3] followed by the robot.

4.2.1. *Localization at a glance*

Ideally it would now be possible to select at each time step the closest place to the true robot position, only by comparing the SFV's stored in the map and the one computed from the image grabbed at the current position.

A correct localization "at a glance" could be considered optimal. The left part of Figure 4 shows the result of such an unconstrained One-Step-Localization experiment. Grey lines are drawn from each recorded position during path execution to the place with the best matching SFV. In addition the orientation of the corresponding view direction is plotted in grey at each robot position. There are two things to note:

(1) A correct localization with only one image is possible in 70–80% of all cases. There are only a few mismatches and (at least here) at maximum two consequent errors.
(2) In those cases, where the current image was assigned to the correct [4] place, especially the orientation estimated from the map is correct. This is notable, because dead reckoning errors effect the orientation estimation most. As expected, the visual information is very orientation sensitive and thus well suited for updating the odometry.

It should also be noted that **only vision data were used** to perform the unconstrained localization. The robot performs a complete and unconstrained relocalization at every step. This shows, that the features provided by the SFV's are very rich compared to e.g. simple range measurements. The next section will present a simple but effective *voting mechanism*, which can be used to remove the remaining errors.

4.2.2. *Passive and Active Selflocalization by "Voting"*

Looking carefully at figure 4 (left), only a few and single misassingments can be identified. Therefore it seems reasonable to think about a *voting mechanism* which uses also past images for finding the best matching place in the map. The algorithm we use selects the place with the maximum votes in the past N images.

Naturally any decision to switch to the next node is delayed by this voting procedure for a large N, because new decisions have to be supported by several measurements before being accepted. However, in combination with distance thresholding it is possible to

[3] Map acquisition and the execution of the reference path were performed in completely different runs. The similarity between dead reckoning errors in fig. 3 is a pure coincidence.

[4] Correct in the sense, that one of the closest places in the workspace was selected.

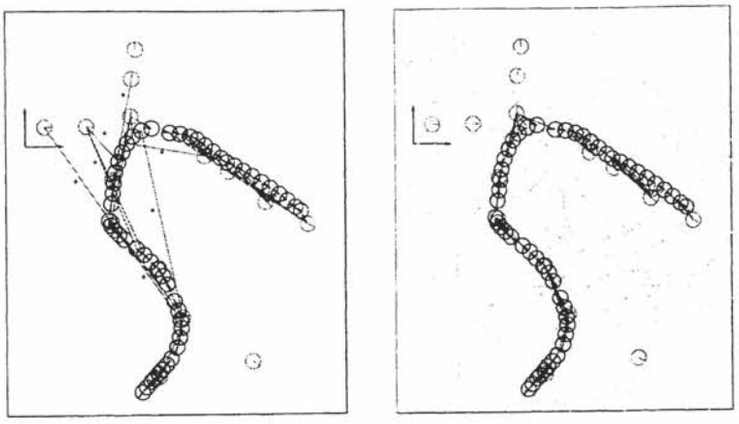

Fig. 4. Localization at a glance: One-Step-Localization with no constraints (left). Localization by voting: Constrained Three-Step-Localization with a distance threshold of $r = 2.5m$ (right). Erroneous localizations are marked with a '*'.

choose $N = 3$ assuming aditionally a minimum accuracy of the dead reckoning process and excluding all nodes further away than $r = 2.5m$ (marked by the large circles) from the search for the best candidate (fig. 4, right). Although this is rather straight forward and is sufficient in the given example, the voting mechanism allows additionally for a seamless integration of passive and active selflocalization strategies.

In the previous sections only a passive selflocalization was presented. The robot used images acquired during its trip without taking any special actions to establish or support a specific localization hypothesis. Such passive methods do not fully exploit the data stored in the map, since due to our omnidirectional place representation information on other directions than the current driving direction is also available. Humans tend to stop their movement when they get lost and start looking around to gather additional data to support or change their current idea of where they are. This type of behaviour can be added to our robot system by introducing a voting threshold. If the maximum number of votes falls below this threshold, the robot stops its default movement and turns the camera to capture additional views from its current position. These views are then replacing earlier images in the voting buffer. The voting procedure does not need to be changed to incorporate active relocalization.

4.3 *Choosing Places to Represent*

One problem, which was not discussed in the previous sections, is the selection of appropriate places for inclusion in the map. Our current mapbuilder uses only a distance criterion: If the exploration path leads the robot to a region further than an empirically determined threshold of one meter [5] away from all previously established graph nodes, a new node is generated and an omnidirectional view is acquired. As a next step we will add two additional conditions for node generation:

(1) If the voting procedure is not able to establish a clear position hypothesis with respect to the current map, i.e. if also after stopping and the acquisition of additional images no clear majority vote can be made, this indicates that a new node should be generated. Either because the vehicle reached a previously unknown region, or because the environment has changed. If the new node lies close to existing nodes, these should be removed to reflect changes in the robot's world.
(2) If the SFV changes significantly during traveling – this is determined from a learning quantization of the SFV's v. Wichert (1996) – a new node will also be inserted in the map. This condition enables an image "content" driven generation of map nodes in addition to the distance criterion used so far.

While the first criterion will allow for navigation in moderately dynamic environments, the second node insertion condition will strengthen the selforganization characteristics of the map building process as it adapts the node densitity in the map to the distribution of the data in the SFV-space.

5. INTERNAL VS. EXTERNAL CONCEPTS

The graph-like map of the environment acquired using the techniques describe above forms an internal representation of the system. It is not suited for communicating with the robot, because it is not transparent to the user.

The unsupervised acquisition of a representation that fits with the human users concepts is not realistic. Therefore it is neccesary to teach the external concept to the system in a supervised manner on a higher level of abstraction. A simple assignment of names to nodes is not satisfying. Our approach is based on the definition of an application dependent construction kit. For

[5] As measured by odometry.

an indoor application this kit might look like the one depicted in figure 5. Predefined structural conditions – we compel the system to place nodes in doorways – force the graph map to fit in the corresponding higer level structures. This way our system can automaticly translate its internal representation into the users way of perceiving things without the designer making more than very rough predefinitions. On the other hand this bridges the gap between subsymbolic continuous scene features and an object oriented higher level description that facilitates the communication with the system.

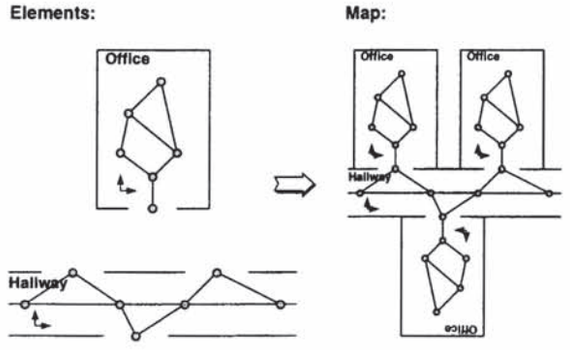

Fig. 5. The "representation construction kit" contains predefined and application dependent environmental structures, that are used to automaticly build representations transparent to the user.

6. CONCLUSION

The work presented in this paper has shown, that our selforganizing scene feature extraction process can successfully contribute to the problem of mobile robot selflocalization. The unsupervised nature of our approach is desirable, because true service robots operating in unstructured and apriori unknown environments do not only need to adapt their behaviour but also their perception procedures to the specific characteristics of their application environment.

In our system the complete sensor interpretation process is totally unsupervised. Nevertheless, the robot is able to relocalize itself with respect to a previously acquired map. The procedures introduced above allow for a seamless integration of passive and active self-localization strategies. As sketched above, next steps will extend our method to moderately dynamic environments. Our approach does not require any type of environment model to be supplied to the robot in advance. No restricting assumption neither on the structure of the workspace nor on the internal structure of the images (the existence of vertical line segments or something similar) have to be made. Easy communication with the robot requires the internal concepts of the system to be translated into the users way of thinking. We propose an approach using predefined environmental structures from a "representation construction kit" to build a transparent representation.

References

Wolfram Burgard, Dieter Fox, Daniel Henning, and Timo Schmidt. Position tracking with position probability grids. In *1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT'96)*, pages 2–9, Kaiserslautern, Germany, 1996. IEEE Computer Society Press.

Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete baysian models for mobile-robot navigation. In *Proc. of the Conf. on Intelligent Robots and Systems (IROS'96)*, pages 963–972, 1996.

Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8:179–187, 1961.

Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer, New York, London, Paris, Tokyo, 1988.

Andreas Kurz. ALEF: An autonomous vehicle which learns basic skills and constructs maps for navigation. *Autonomous Systems*, 14:171–183, 1995.

Kenneth J. Laws. *Textured Image Segmentation*. PhD thesis, University of Southern California, 1980. USCIPI Rep. 940.

Hans P. Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Intern. Conf. on Robotics and Automation*, pages 19–24, 1985.

T. Pomierski and H.-M. Groß. Biological neural architecture for chromatic adaptation resulting in constant color sensations. In *Proc. of the ICNN-96*, pages 734–739, Washington DC, USA, 1996. IEEE-Press.

Reid Simmons and Sven Koenig. Probabilistic navigation in partially observable environments. In *Intern. Joint Conf. on Artificial intelligence (IJCAI'95)*, pages 1080–1087, 1995.

Sebastian B. Thrun. A lifelong learning perspective for mobile robot control. In *Proc. of the IEEE/RSJ/GI Intern. Conf. on Intelligent Robots and Systems (IROS'94)*, pages 23–30, 1994.

Henning Tolle and Andreas Kurz. Learning aspects for mobile service robots. *accepted for: Intern. Journal of Intelligent Control and Systems (Special Issue: Intelligent Machines: Bridging the Gap between Theory and Practice)*, 1998.

G. v. Wichert. Selforganizing visual perception for mobile robot navigation. In *1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT'96)*, pages 194–200, Kaiserslautern, Germany, 1996. IEEE Computer Society Press.

G. v. Wichert and K. Kleiner. Selbstorganisierende Bildanalyse für die Navigation von mobilen Robotern. In R. Dillmann and T. Lüth, editors, *Autonome Mobile Systeme (AMS'95)*, Informatik Aktuell. Springer Verlag, Heidelberg, December 1995.

U. R. Zimmer. Self-localization in dynamic environments. In *IEEE/SOFT Workshop BIES'95*, Tokyo (Japan), May 1995.