

**LOOKING FOR CONCEPTS:  
UNSUPERVISED MAP CONSTRUCTION WITH UNKNOWN SENSOR CONFIGURATION**

**Philip Mächler (philip.maechler@epfl.ch)**

LAMI-DI, EPFL Swiss Federal Institute of Technology, Lausanne  
Tel: ++41 21 693 39 07 Fax: ++41 21 693 52 63

**ABSTRACT**

*The navigation method presented here allows a robot to find its position and its route without prior information about its environment. What is going to be used as landmark is initially unknown, and no specific preprocessing of the sensor signals is done. The robot nevertheless extracts meaningful information from its environment and uses it to build a map. Both landmark definitions and map are continuously adapted. Working without predefined categorization takes full advantage of the sensor abilities.*

*The result shows that a robot can reliably recognize self-defined landmarks suitable for its sensor capability and create a map of its environment.*

**1 INTRODUCTION**

For us, humans, it is not too difficult to get familiar with a new environment. We scan it for objects and remember their spatial relationship. When explaining the route to a tourist, we link successive objects (bridges, monuments, etc.) with spatial information (left, facing, 50 feet, etc.). Of course, what we describe as a striking object depends on our perception. Living in a different sensory world, a dog would rather select odor traces than monuments. The amount of such objects is also important, so trees are useful for orientation in a town, but not in a forest.

The purpose of this work is to find an algorithm to extract and identify *significant*<sup>1</sup> events from the raw sensor signal and combine them into *concepts*<sup>2</sup> suitable for the agent. These concepts can be compared to human concepts like “chair” or “door”, but should match the agent senses; they might have little value for humans.

1. Significant means “interesting” and describes a sensor event which appears not too rare (could be a sensor error or noise), not too often (representing uninteresting information) and not repeated in short time (which would not correspond to a single point).
2. Organization and interpretation of the robot’s sensory input in a way suitable for its capabilities and requirements.

These concepts are finally used to create a map.

The use of adaptive concepts is not new (see Pfeifer [12] and Scheier [14]): many recognition algorithms (especially in artificial intelligence) leave the creation of concepts to the machine in order to avoid human bias. But automatic concept creation normally begins at a high level, combining classically preprocessed sensor and status information. Such preprocessing makes things easier, but always entails a loss of (probably important) information.

The purpose of this paper is to extend the automatic concept creation as close as possible to the raw sensor signals (Kuipers and Pierce [7]). Ideally, such a system should be completely sensor-independent and free of any preprocessing. We want to investigate the limitations of such a method and see how close an experiment can approach this theoretical, ideal case.

**2 EXPERIMENTAL SET-UP**

As described in [9], I use the Khepera robot simulator developed by Olivier Michel [10]. A simulated robot moves randomly in an unknown, static environment. It avoids (is repelled by) obstacles and records its ongoing sensory information. It is equipped with eight proximity sensors (about 5 cm range), eight ambient light sensors, a compass and a frontal 64 pixel linear camera covering 36 degrees. The result of an edge extraction algorithm is used for further proceedings.

The environment is a square field about 15 x 15 x the size of the robot, containing obstacles and light sources.

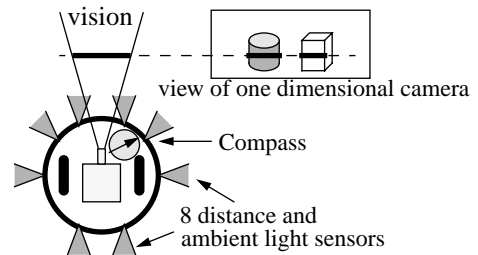


Fig. 2-1: Equipment of the extended Khepera robot

### 3 SIGNAL PROCESSING

The principal idea is to merge all raw sensor signals together and to treat them in the same way independently of the source. A self-organizing process clusters the signals in classes which correspond to the environmental situations. However, two exceptions have to be considered:

- Analog and digital signals have to be treated differently.
- The raw camera image is filtered and processed by a edge detection algorithm before further use.

#### 3.1 The classifier modules

A set of classifier modules organizes sensor samples into *classes*. In order to recognize rare events, unsupervised classifiers, implemented as neural networks (NN), must develop few and small classes with few entries. Such classifiers must neither forget such small classes nor fuse them too easily. The following unsupervised classifiers are used:

- An *Adaptive Resonance Theory* (ART) neural network (Grossberg and Carpenter [1], [4] [5]) which clusters *digital* vectors. A new cluster (neuron) will be created whenever a new input pattern doesn't match sufficiently to an existing one. This helps to overcome the *stability-plasticity dilemma*, that is, the difficulty of constantly classifying new experiences without forgetting old ones.
- A *Fuzzy ART* incorporates fuzzy logic in the ART structure and is capable of clustering *analog* input vectors. A very good description of the related supervised fuzzy ARTMAP algorithm is provided by Kasuba [6].
- A *Growing K-means* classifier which consists of a standard K-means classifier extended by the ability to acquire new output neurons if the existing ones are not suitable. It is similar *Growing Neural Gas* networks [3].

Increasing the number and types of classifying systems gives a more reliable characterization of the environment. The different types of cluster shapes displayed these classifiers increase the probability that all significant features of the environment are reliably enclosed (see fig. 3-1).

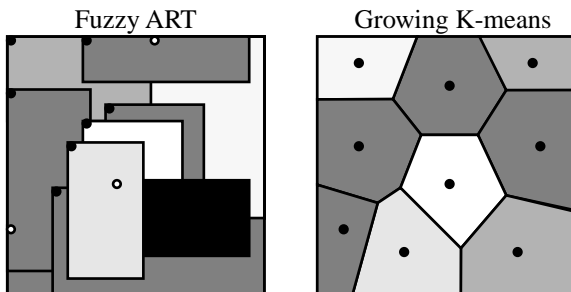


Fig. 3-1: Different cluster shapes for a 2D input space

#### 3.2 Extracting significant classes and landmarks

For a localization task, a class is considered significant if it represents a (possible unique) position in the environment and if the activation of this class is:

- not very rarely (because it might represent noise)
- not too often (it might represent different positions)
- not activated several times in succession (not a point)

Therefore a statistical analyzer is used in order to extract such significant classes from the complete pool of classes. This technique is inspired by Nake [11] and others [16] who suggest that most interesting data exhibits an ideal ratio between expected and unexpected information.

A *landmark* is a significant class combined with its adjacent classes (fig. 3-2) which increases the characteristics of a landmark and decreases the risk of confusion.

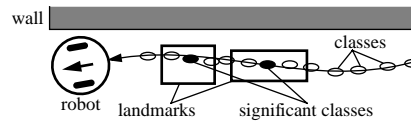


Fig. 3-2: Landmark: Significant class with the adjacent classes

#### 3.3 Quality versus quantity of significant classes

The euclidean distance between class prototypes must be short in order to separate sensory experiences corresponding to significantly different environmental situations. On the other hand, excessive many and compact classes might be affected by noise. A compromise for the amount of classes has to be found.

The diagram in fig 3-3 shows how often a significant class is activated during 10 minutes as a function of the total amount of classes. The descending lines in the diagram show that classifiers containing too many classes are less efficient because significant classes (which represent almost unique robot situations) are too compact for a reliable recognition.

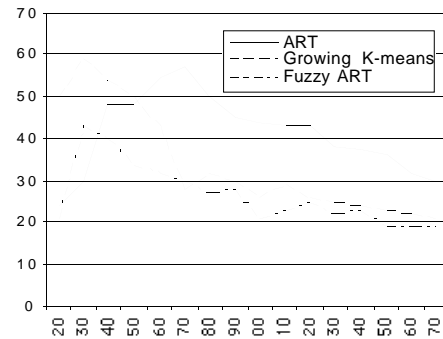


Fig. 3-3: Too compact (too many) classes decrease the quality

### 3.4 Landmark comparison

Landmarks (a string of classes) created by a robot in the same position and in equivalent circumstances are rarely identical; they always differ by a number of classes: some former features may be missing, new ones may have been added.

To compare two strings of discrete elements in a tolerant way (like text search) we use the *Weighted Levenshtein Distance* (WLD) developed by V.I. Levenshtein [8].

The WLD algorithm recognizes similar landmarks, even in moderately different class strings. The algorithm is defined as the minimum cost required to convert a string into another by deletion, insertion and substitution of characters (classes). The WLD can be easily explained by the following recursive formulation:

$$L(a_i, b_j) = \min \begin{cases} L(a_{i-1}, b_j) + \text{cost}_{del} \\ L(a_i, b_{j-1}) + \text{cost}_{ins} \\ L(a_{i-1}, b_{j-1}) + \begin{cases} 0 & \text{if } a[i] = b[j] \\ \text{cost}_{sub} & \text{if } a[i] \neq b[j] \end{cases} \end{cases}$$

$L(a_i, b_j)$  is the distance (difference) between the first  $i$  characters of string  $a$  and the first  $j$  characters of string  $b$ . Organizing temporary results in a table drastically reduces the calculation cost (Reuhkala [13]).

## 4 CONSTRAINING ROBOT MOVEMENTS

An autonomous robot moving randomly in a two dimensional space will not find enough landmarks to calibrate its odometry. Some hints could help it to take advantage of available landmarks. Steering the robot actively to landmarks requires the use of artificial concepts and coordination between sensors and motors, which offends the condition mentioned in the section "Introduction".

A possible solution is to let the robot be attracted by "interesting" signal events generated by its sensors; "interesting" meaning that the sensors are well stimulated.

The short range distance sensor of the robot supplies only interesting information, if the robot is close to an obstacle. This signal can be used in a fitness function to train a NN for a "wall following behavior". In the same way, the camera supplies only interesting information if the robot heads to an object presenting sufficient contrast. This signal can be used in a NN fitness function to visit edges and objects. Combining these behaviors allows the robot to explore the environment by following interesting stimuli as shown in fig. 4-1. Thanks to the camera, it can even overcome empty areas to reach other objects.

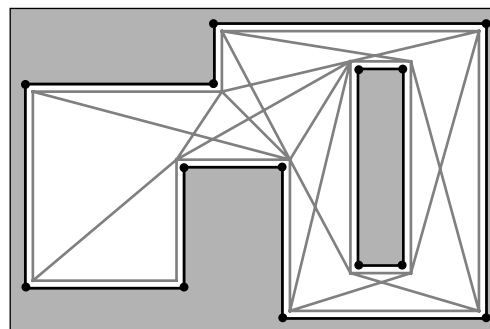


Fig. 4-1: Robot path attracted by walls and contours

The expected result by combining these behaviors is shown by gray lines in fig. 4-1.

Currently, this algorithm is directly programmed and not implemented by a NN to save time for more important parts of the experiment.

## 5 INTERMEDIARY RESULT

The described algorithm was tested on a simulated robot environment containing static obstacles (fig. 5-1). The freedom is reduced to some paths showed by gray lines. During this constrained movement about 150 landmarks are extracted from the camera, compass and distance sensor signals. The reason why some spots are chosen as landmark is not always intuitive for a human. Fig. 5-1 shows twenty of the most obvious ones. Some of them were generated as the robot was located in corners which activated more distance sensors than usual. Others were generated by the appearance of objects in the camera view.

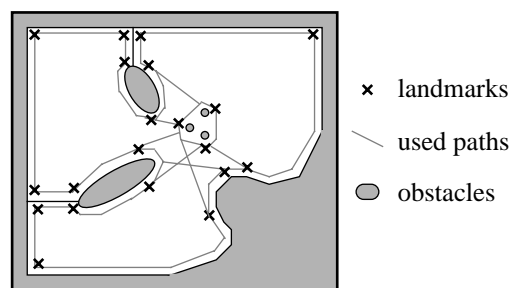


Fig. 5-1: Robot environment with created landmarks

Each landmark is stored with an associated estimated position which will be used to calibrate the robot's position while moving. Every time the robot recognizes a landmark in a plausible position, it corrects its odometry, setting it to the position which was associated to the landmark. The result is a bounded odometry error.

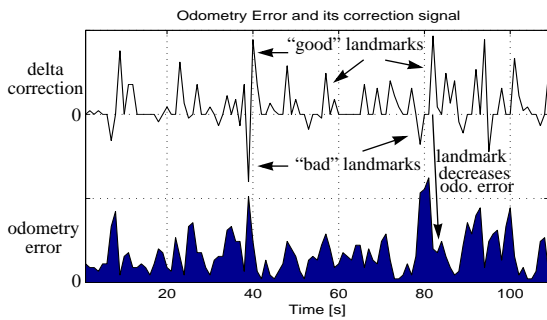


Fig. 5-2: Stabilized error due of landmark synchronization

The lower, filled graph in fig. 5-2 shows the absolute odometry error (maximum peak is about a third of the robot diameter). There is always a small absolute position error which never exceeds a certain limit if regularly some landmarks can be found. This limit describes the minimal distance between two identical landmarks and is an estimated parameter of about half the robot’s diameter. The upper graph shows the correction made by the recognized landmarks (same scale). Every positive peak of the curve means a right identification and therefore decreases the odometry error. Notice that there are also wrongly identified landmarks indicated by negative peaks. The reason for such faulty updates are several identical landmarks inside the described minimal distance. Such mistakes increase the odometry error, but they are quickly compensated.

## 6 LANDMARKS BECOME PLACES

The intermediary result shows that landmark recognition without any knowledge about sensors and environment can be done.

However, assigning absolute position information to landmarks is not realistic because of the continuously increasing odometry error. Randomly exploring the environment and assigning any found landmark to the actually estimated position would not result in a well distributed position accuracy. Spatially close landmarks could be assigned with disproportionately different positions.

A strategical exploration which discovers first the closest environment of the robot is difficult to implement because the robot has no concept of space. This means e.g. a wall on the left side produces a different stimuli (class) than the same wall on the right side. There is no orientation.

### 6.1 Recognizing places from the landmark stream

Assigning landmarks to places could solve this problem. A *place* is defined by a spatial accumulation of landmarks

and therefore always represents a significant robot’s position (e.g. corner, narrow corridor or rare optical event). The landmark position is defined relative to the position inside the *place*. Fig. 6-1 shows a robot’s path producing 12 landmarks (small rectangle). The corner groups 5 landmarks to a place A. In the same way, the rising light (hidden behind a bend) groups 4 landmarks to a place B.

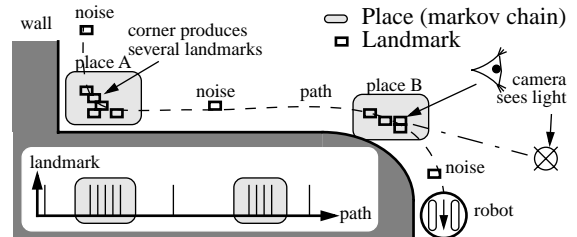


Fig. 6-1: Accumulations of landmarks become places

### 6.2 Place identification by Markov Chain

It is evident that a *place* (that is a sequence of landmarks) offers a better guarantee of proper identification than just one landmark. However, landmark sequences of the same *place* ( $S_{place}$ ) always differ due to robot trajectory drift. The experiment shows that only some *sub-parts* of the sequence stay identical, but their associated position may also have drifted. The Weighted Levenshtein Distance cannot distinguish between missing and shifted landmarks and is therefore unsuitable for place recognition.

*Markov chains* are a better tool to recognize sequences because they are sensitive to landmark transitions and not to their position in the sequence. So the probability that a place corresponds to a sequence of landmarks can be deduced. Fig. 6-2 shows a short overview of the Markov chain theory (see also Collins [2] and Ross [15]).

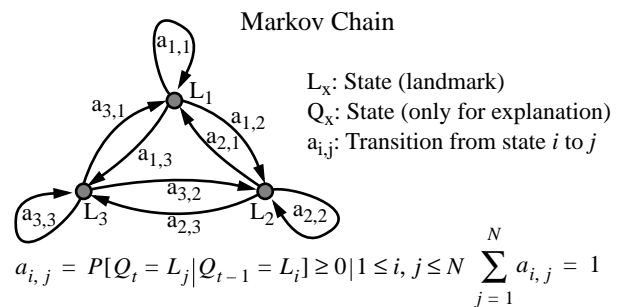


Fig. 6-2: Short definition of First Order Markov Chain

Each *place* is represented by a transition matrix:

$$M_{place} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

$a_{i,j}$ : Transition from landmark  $i$  to  $j$   
 $N$ : Maximal number of landmark  
 $M$ : Ergodic Markov chain for a place

The size  $N$  of the matrix is fixed and defined by the number of different landmarks. Each cell contains the probability of transition from landmark  $i$  to landmark  $j$ . Each new landmark sequence  $S_{new}$  has to be compared with all transition matrixes  $M_k$  (representing *places*). The following equation calculates the index  $w$  of the transition matrix  $M_w$  representing the most probable *place* for  $S_{new}$ :

$$w = \underset{k}{\operatorname{argmax}} P(M_k | S_{new})$$

index of winner place A posteriori probability of being in place  $k$  given the new sequence of landmarks.

Bayes transformation allows to calculate the a posteriori probability  $P(M_k | S_{new})$  from the likelihood  $P(S_{new} | M_k)$ :

$$P(M_k | S_{new}) = \frac{P(S_{new} | M_k) \cdot P(M_k)}{P(S_{new})}$$

A posteriori prob.      Likelihood      A priori prob.

Bayes identity:

The probability that  $M_k$  contains the sequence  $S_{new}$  is identical for every  $k$ , so  $P(M_k)$  is constant for all  $k$ .  $P(S_{new})$  is independent of  $k$  and can therefore be ignored as well. So  $w$  can be calculated as following:

$$w = \underset{k}{\operatorname{argmax}} P(S_{new} | M_k)$$

The winning transition matrix  $M_w$  represents the place with the highest probability of belonging to place  $S_{new}$ . Real correspondence has to be verified by a *normalized score*  $L(S_{new}, w)$  which has to be over the threshold  $T$ :

$$\text{Normalized score: } L(S_{new} | w) = \frac{P(S_{new} | M_w)}{P(S_{new} | M_{new})} > T$$

The new sequence  $S_{new}$  corresponds to the winner place  $M_w$  if the normalized score exceeds the threshold  $T$ , otherwise a new place  $M_{new}$  will be added to the database. The threshold  $T$  of about 0.008 was found empirically.

The environment showed in fig. 5-1 generates about 40 places. Fig 6-3 shows some of the most important places, sometimes overlapped. Place recognition is very reliable concerning the confusion between places. However, places are sometimes not recognized due to too few landmarks.

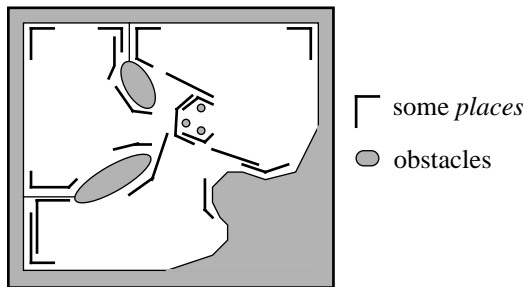


Fig. 6-3: Typical distribution of the most important *places*

## 7 PLACES BECOME A MAP

The abstraction level of *places* is now high enough to use it as robotics concepts which can be compared with simple human concepts like “chair”, “door”, “window”, etc. Therefore *places make* cartesian x,y position superfluous and can be used straight as spatial target definition.

Each place is linked with all reachable neighbor places. This network of links may be slightly distorted from reality and does not represent to scale the real environment. However, traveling from one place to another is possible and should be largely sufficient for navigation.

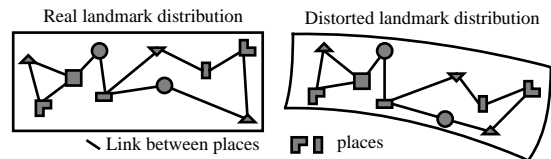


Fig. 7-1: Distorted representation of the environment

## 8 LEARNING PHASES

Acquiring “classes”, “landmarks” and “places” while randomly exploring the environment, cannot be done simultaneously; it implies a three-phase strategy of which the timing is shown in fig. 8-1. Their starting times are shifted and they never completely stop (see also fig. 8-2).

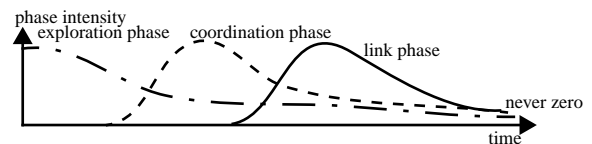


Fig. 8-1: Transition of the learning phases

A description of the phases can be found below. Please refer also to fig. 8-2 for better understanding:

- 1) In the *exploration phase*, signal events (see in fig. 8-2 sensor A-C) are classified and most landmarks are created. The classifiers becomes more and more rigid, stabilizing the classes in the sensor input space. After a certain time, the classes are rigid enough to allow the *coordination phase*. However, the robot will never completely quit the *exploration phase* in order to stay adaptive to new nonclassified sensor events.
- 2) In the *coordination phase*, accumulated landmarks are grouped into places (see in fig. 8-2 landmarks I, J, K and R become place  $Z_c$ ). A typical place contains about 20 landmarks, but only about 5 are always reliably discovered for each passing by. As before, the *coordination phase* will never completely freeze so that the robot can stay adaptive to future landmark changes.

- 3) In the *linking phase*, the places are linked together, which creates the map. The links result from the allowed movements of the robot. Each link contains a weight, which is reinforced after each use. Such reinforced links are preferred by the future path planner.

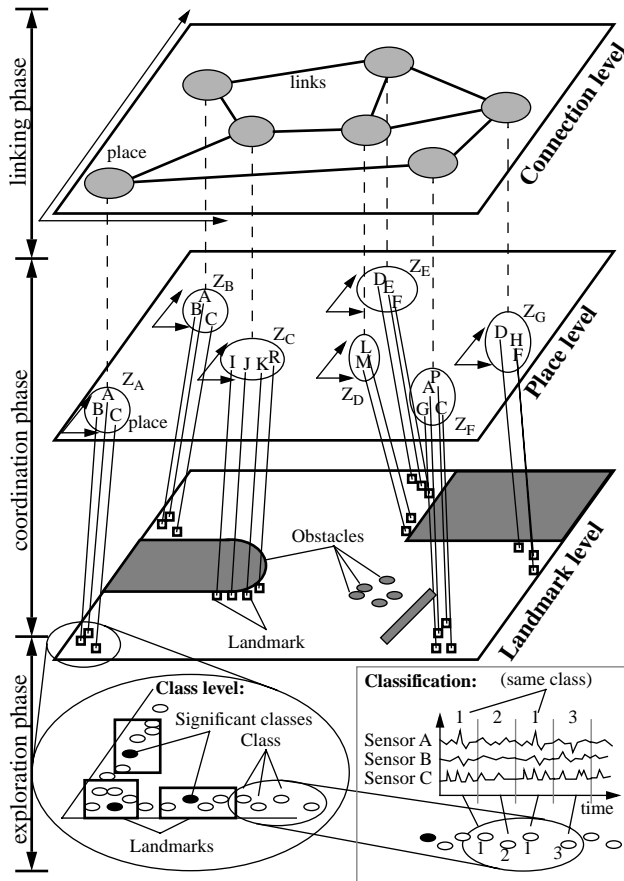


Fig. 8-2: Division into learning phases

## 9 CONCLUSION

This paper shows that a robot can extract and recognize suitable landmarks from pure sensor signals with almost no knowledge about the environment or its sensor configuration. These landmarks can be combined into concepts comparable to very basic human concepts like “chair” or “door”. The autonomous and unsupervised creation of concepts takes full advantage of the sensor abilities. The system can adapt to small changes in the environment, but it will not work in a highly dynamic one.

The concepts can be stored into a map and linked together, allowing a robot to navigate in its environment without the need of accurate cartesian position information. Odometry is only used to get a spacial idea about the place distribution and to follow place links.

## 10 REFERENCES

- [1] G.A. Carpenter and S. Grossberg, *ART 2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns*, Applied Optics, vol. 26 (1987), no. 23, page 4919-4930
- [2] Lyndhurst Collins, *An Introduction to Markov chain analysis*, Concepts and techniques in modern geography no.1, University of Edinburgh, Norwich, 1975
- [3] Bernd Fritzsche, *Some Competitive Learning Methods*, Inst. for Neural Computation, Ruhr-Universität, Germany, 1997, <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper>
- [4] S. Grossberg, *The ART of adaptive pattern recognition by self-organizing neural network*, Computer, Vol. 21, Mar., 1988, page 77-88
- [5] John Hertz, Anders Krogh, Richard G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley, Redwood City CA, 1991, ISBN 0-201-50395-6, 325 pages
- [6] Tom Kasuba, *Simplified Fuzzy ARTMAP*, AI Expert, Nov. 1993, 8 pages
- [7] Benjamin Kuipers, David Pierce, *Map Learning with Uninterpreted Sensors and Effectors*, Uni. of Texas, Austin USA, Artificial Intelligence Journal, 1997
- [8] V.I. Levenshtein, *On the minimal redundancy of binary error - correcting codes*, Information and Control, Vol 28, Nr. 4, August 1975, 23 pages
- [9] Philip Maechler, *Without a clue: Unsupervised robot navigation with unknown sensor configuration*, IAV'98, Madrid, Spain, Mars 1998, 6 pages.
- [10] Olivier Michel, *Khepera simulator package version 2.0*, Freeware mobile robot simulator, see <http://diwww.epfl.ch/lami/team/michel/khep-sim/index.html>
- [11] F. Nake, *Ästhetik als Informationsverarbeitung*, Springer, 1974
- [12] Rolf Pfeifer, *Building “Fungus Eaters”: Design Principles of Autonomous Agents*, SAB96, From Animals to Animats 4, 1996, page 3-12
- [13] E. Reuhkala, *Recognition of strings of discrete symbols with special application to isolated word recognition*, Acta Polyt. Scand. Ma 38. Dr. Tech. dissertation, Helsinki Univ. of Tech. 1983
- [14] Christian Scheier, *Categorization in a real-world agent using haptic exploration and active perception*, SAB96, From Animals to Animats 4, 1996
- [15] Sheldon M. Ross, *Introduction to probability models*, Academic Press, ISBN 0-12-598470-7, 1997
- [16] Jürgen Schmidhuber, *What's interesting?*, Technical report, IDSIA-35-97, Lugano, Switzerland, 23 pages, 1997, <http://www.idsia.ch/~juergen>