Expectancy-Based Robot Localization through Context Evaluation

Maria E. Niessen, Gert Kootstra, Sjoerd de Jong, and Tjeerd C. Andringa
Artificial Intelligence, University of Groningen, The Netherlands

Abstract— *Agents that operate in a real-world environment have to process an abundance of information, which may be ambiguous or noisy. We present a method inspired by cognitive research that keeps track of sensory information, and interprets it with knowledge of the context. We test this model on visual information from the real-world environment of a mobile robot in order to improve its self-localization. We use a topological map to represent the environment, which is an abstract representation of distinct places and the connections between them. Expectancies of the place of the robot on the map are combined with evidence from observations to reach the best prediction of the next place of the robot. These expectancies make a place prediction more robust to ambiguous and noisy observations. Results of the model operating on data gathered by a mobile robot confirm that context evaluation improves localization compared to a data-driven model.*

Keywords: Spreading activation, knowledge network, cognitive science, robot localization

1. Introduction

Agents that operate in a real-world environment have additional challenges compared to agents that operate in a simulated or controlled environment. They have to process an abundance of information, of which not everything is necessarily relevant. Moreover, sensory information may be ambiguous or noisy. To be able to make sense of its environment, an agent needs to identify and structure the sensory information it gathers. We developed a method inspired by cognitive research that keeps track of sensory information, and interprets it with knowledge of the context. Human perception is also not strictly data-driven. Knowledge of the context helps humans to form predictions and guide their perception of the environment (e.g. [1], [2]).

Applications of cognitive research, such as handwriting recognition (e.g. [3]) and information retrieval (e.g. [4], [5]), often employ a spreading activation semantic network to recognize a particular item or retrieve specific information. Spreading activation networks are based on models of human memory [6]. They are built of nodes that represent pieces of information or concepts, and the connections represent the prior probabilities that the nodes are encountered together. Spreading activation networks are typically static, because the data in these application domains can be accessed completely and simultaneously. In contrast, for agents operating in a dynamic environment the available information continuously changes.

To be able to deal with continuous data, we apply a dynamic network model. This dynamic network is similar to a spreading activation network, but instead of being static, it is updated when new data are encountered. The model continuously makes an estimation of its current state, based on sensory input and knowledge of the context. The dynamic network model has been applied previously to sound input [7], but is developed to process any type of sensory input. Therefore, as we will show in this paper, it can also be applied to visual information from the real-world environment of a mobile robot.

A basic task for an autonomous mobile robot is to build a map of its environment for self-localization. For this reason, Simultaneous Localization and Mapping (SLAM) has received considerable attention in the last decade. Most SLAM approaches use range or vision sensors to construct a detailed metric map of the environment (e.g. [8]). These maps contain the Cartesian coordinates of a large number of structural features present in the environment. Other approaches build topological maps of the environment (e.g. [9]). Instead of representing the environment in detail, it is represented more abstractly in topological maps, as distinct places and the connections between them. The advantage of such an abstract representation is that it is less susceptible to noise, and ambiguous observations and situations. Moreover, it results in a computationally less demanding system.

In topological mapping, a general idea of the location of the robot can help to form an expectancy of the path of the robot. This expectancy can be combined with evidence from observations to form a hypothesis of the place of the robot. Furthermore, an expectancy of the place of the robot can resolve ambiguous observations. In this way, the place in a topological map where an observation is made can be considered as the context of that observation. When the robot is moving and making observations, an evaluation of the context can improve its localization. The evaluation of the context entails that the recent history of visited places is used to predict the place that follows. Furthermore, using knowledge of the context makes localization more robust to noise in the observations.

In the next section we describe the design of the model, and how it processes observations made by a mobile robot. In section 4 we present the results of two experiments that are described in section 3. The first experiment demonstrates that the model is more robust to noise when the context is used. The second experiment shows that predictions in real data with many ambiguous observations and noise are also better with context evaluation than without. We end with a discussion on the performance of the model and give an outlook on future work.

2. Model

The model we present processes visual input of a moving robot. These visual observations, which are explained in section 2.1, provide evidence about the place of the robot. However, ambiguous or noisy observations might lead to erroneous place predictions. To improve these predictions, contextual information about the environment is learned in a supervised training phase and stored in a static knowledge network. In the operation phase this knowledge is used in a dynamic network model, which computes expectancies of the place of the robot.

The knowledge about the environment, in the form of nodes in the knowledge network and the strength w of the connections between them, is computed in the training phase. We refer to this knowledge as long-term memory, since it reflects invariant knowledge. Therefore, it is stored as a static network, which is constructed from learning relations in the training data. This knowledge network is similar to semantic networks used in information retrieval. In section 2.2 we describe in more detail how the knowledge network is created.

In contrast to the knowledge network, the dynamic network reflects short-term memory. Information represented by nodes in this network is added and forgotten more quickly, since the nodes pertain only to the current state of the robot. Nodes in the dynamic network are called hypotheses, because they represent possible explanations for input data. The dynamic network has three levels that all represent a different type of information: hypotheses of observations, landmarks, and places in the environment. Figure 1 shows an example of a dynamic network at one moment, namely when oberservation 2 has been made. The network configuration represents the knowledge of the environment at that moment. This knowledge consist of two observations, their connections to landmarks hypotheses, and the connections of the landmarks to hypotheses of places in the environment. In section 2.3 we explain the construction of the dynamic network, and how context is used to compute expectancies of the place of the robot.

2.1 Observations

The robot (a Pioneer 2 DX mobile) uses a video camera to observe the world. Visual interest points are detected in the camera images, which serve as landmarks to represent the environment. The interest points are detected and described using the Scale-Invariant Feature Transform (SIFT) [10]. The SIFT algorithm detects points that stand out from their surroundings. These points are described using histograms

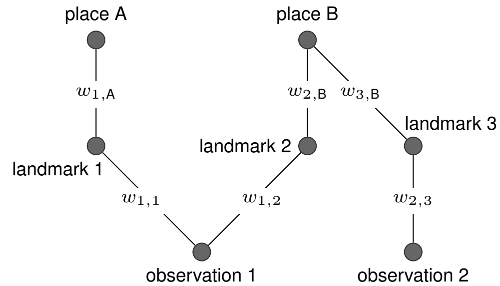

Fig. 1: Example network configuration at one instant, of two observations that are matched to three landmarks, each in turn connected to a place.

of gradients. A drawback of SIFT is that it results in a large number of interest points, many of which are not re-detected in subsequent images. Therefore, we use a visual buffer to test the stability of the interest points over a number of successive images [11]. Only interest points that are stable enough are used as landmark observations. The descriptor of an observation is then compared to that of previously observed landmarks. Based on the descriptor distances, the observation is matched with one or more landmarks or labeled as a new landmark.

The data set used in one of the two experiments (see section 3.2) was collected by the robot while it drove a closed loop of eight by ten meters in an office-like environment. The data was logged by the robot while driving four laps. The map of the loop was manually divided into nine places, as depicted in Figure 2. Half of the data set, that is, the observations made in the first two laps, is used to determine which landmarks are observed in which place. The other half is used to test the model. Because of the variability of the images in different laps, the robot might have observed landmarks in the last two laps that are not present in the training data.

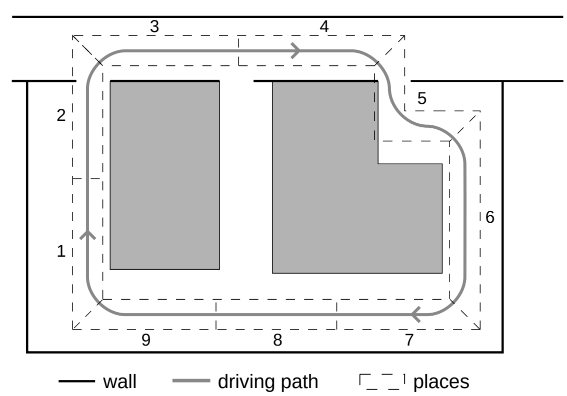

Fig. 2: Environment where the robot drove four laps. The size of the loop is 8 by 10 meters, divided into nine places. The gray area consists of objects the robot cannot drive through.

2.2 Knowledge Network

Three classes of information are stored in a knowledge network: the descriptors of the landmarks, the relations between the landmarks and the places in the environment, and the transitions between the places. This knowledge network represents the context, which is slowly changing or invariant. Therefore, it is referred to as the long-term memory of the model.

The connection strengths between landmarks and places in the training data are calculated according to a term-weighting approach used in automatic document retrieval [12]. In this method the importance of a term (word or phrase) in a document is determined by multiplying its frequency in the document with the inverse frequency it occurs in other documents. Hence, the term is important for a document if it occurs often in that document and infrequently in other documents. Since the connection strength (weight) between a landmark and a place should reflect the specifity of the landmark to that place, we adopt the term-weighting approach. The landmarks can be treated as terms, and the places as documents. Accordingly, the weight of the connection between landmark l and place r is:

$$w_{r,l} = w_{l,r} = \mathrm{tf} \cdot \log\left(\frac{N}{n}\right), \tag{1}$$

where N is the total number of places, n is the number of places in which landmark l is observed, and the normalized term frequency is given by:

$$\mathrm{tf} = \frac{f_{l,r}}{\sqrt{f_l}}, \tag{2}$$

where $f_{l,r}$ is the observation frequency of l in r, and f_l is the total observation frequency of l.

The connections between observations and landmarks are not stored in the knowledge network, because all observations are unique. Therefore, the weights of these connections are computed at the moment when an observation is made, both in the training and the operation phase.[1] The connection strength between an observation and a landmark should represent the likelihood of a correct matching between their descriptors. If these descriptors are far apart, the observation and landmark are less likely to have been matched correctly. Therefore, the weight of a connection between an observation and a landmark is inverse to the distance between their descriptors:

$$w_{l,o} = w_{o,l} = 1 - \frac{d}{\theta_d}, \tag{3}$$

where d is the distance between the descriptor of observation o and landmark l, and θ_d is the maximum distance at which an observation is still matched to a known landmark.

[1] Observed information is not necessarily always unique. In other domains or applications areas it could be useful to store observations in the knowledge network. However, in the presented application it would be useless to do so.

The transition probability that the robot moves from one place to another is calculated by normalizing the number of times the robot moves from one place to another in the training data (the first two laps). As can be seen in Figure 2, the robot can move within place i or move from place i to place $i \pm 1$. Since the robot is driving the loop in one direction, the transition probabilities to all places other than i and $i+1$ are generally zero. However, there are a few exceptions when no observations are made in a place in one of the laps, and thus the probability to move to $i{+}2$ is greater than zero. The complete matrix of probabilities serves as the context that helps to compute an expectancy about the next location of the robot.

To summarize, the knowledge network consists of the matrix with the a priori transition probabilities between all places. Furthermore, it stores the labels of all landmarks that are observed in the training data, along with their connections to the places in which they are observed.

2.3 Dynamic Network of Hypotheses

Once the knowledge network is fully trained after the learning phase, it is used in the operation phase, together with evidence from observations, to predict the place of the robot. The algorithm for the construction and updating of a dynamic network is summarized in Table 1. Every level in the network consists of hypotheses of a single type of representation (see Figure 1). The landmark observations are the lowest level of the dynamic network. As described in section 2.1, observations are matched to one or more previously observed landmarks, or labeled as a new landmark, which are at the middle level. (The current version of the model only processes known landmarks in the operation phase. The possibility to add new landmarks will be discussed in section 5.) The highest level in the network holds hypotheses of places in the environment.

Each node in the network represents a hypothesis of one of the three different types of representation. When an observation is made, a hypothesis is added to the dynamic network (step 1). Next, its matched landmarks (that are stored in the knowledge network) are initiated as hypotheses (step 2), and they are connected to the observation hypothesis (step 3). Subsequently, these landmark hypotheses retrieve their place connections from the knowledge network (KN)(step 4). These places are also initiated as hypotheses (step 5) and connected to the landmark hypotheses that initiated them (step 6). Every time new observations are made, the network is updated and the dynamics change.

The connections in the dynamic network are symmetrical, and only between hypotheses at different levels. For instance, the landmark hypotheses are connected to the observations that initiated them, and to hypotheses of places in which they may lie, but not to each other (see Figure 1). Connections between hypotheses at the same level would be redundant, since they can reinforce each other through shared par-

Table 1: Algorithm for updating the dynamic network configuration at times when observations are made by the robot.

	For all observations o at time t:
1.	Add observation hypothesis o
2.	Add matched landmark hypotheses l
3.	Connect o and l with strength $w_{o,l}$
4.	Places $r \leftarrow$ receive place connections of l from KN
5.	Add place hypotheses r
6.	Connect l and r with strength $w_{l,r}$
7.	Spread data-driven activation
8.	Spread context-based activation
9.	Evaluate activation values

ent hypotheses. Furthermore, the hierarchy of the network is now captured by the connections. Therefore, it is not necessary to store a global representation of the complete network. Instead, each hypothesis contains information of its relative position in the network, that is, it contains its direct connections. The only information that is stored globally is which hypotheses are active.

2.3.1 Activation Spreading

After the connections in the network are updated, the activation of the observation hypothesis spreads through the network. The computation of the spreading activation is similar to the method used in McClelland and Rumelhart's model of letter perception [13]. The input activation first spreads upward to the place hypotheses at the highest level in the network, and is called data-driven spreading (step 7). Subsequently, the activations of the place hypotheses spread downward to other connected hypotheses, for example landmarks in the same place that are observed previously. We call this context-based spreading (step 8). As a consequence of context-based spreading, a landmark hypothesis of a particular observation can be reinforced by later observations. For example, in Figure 1 the first observation is matched to landmarks 1 and 2, where landmark 1 lies in place A and landmark 2 in place B. Another landmark observation made in place B will increase the support for the hypothesis that the first observation was of landmark 2, and not of landmark 1.

The input activation $n_i(t)$ of the individual hypotheses is the weighted sum of all connected hypotheses, either from the level below, for data-driven activation spreading (step 7), or from the level above, in case of context-based spreading (step 8):

$$n_i(t) = \sum_j w_{ji} A_j(t), \tag{4}$$

where j is a hypothesis connected to i, $A_j(t)$ is its activation, and w_{ji} is the connection strength between hypotheses j and i, retrieved from the knowledge network.

2.3.2 Activation Evaluation

After the activation has spread through the network, the activation value of each hypothesis is evaluated (step 9). The activation evaluation is different for different types of hypotheses, because the context is only relevant for the highest level in the dynamic network, that is, the place hypotheses. The activations of the hypotheses that are not at the highest level in the network are normalized, similar to the model of McClelland and Rumelhart [13]. The activations of the place hypotheses at the highest level are a weighting of evidence from the input and an expected value. The result of the activation evaluation of a hypothesis is treated as the likelihood that the hypothesis is true.

The activation evaluation is an accumulation of current input and the previous activation corrected with a decay. The decay represents that items in short-term memory are forgotten without reinforcement, in contrast to information in long-term memory [6]. The activations of the hypotheses decay exponentially with time toward a default situation. Therefore, the decay function is dependent on the a priori probability of a hypothesis:

$$f_i(\Delta t) = \mathrm{e}^{-\frac{\Delta t}{D}}(1 - P(i)) + P(i), \tag{5}$$

where $P(i)$ is the a priori probability of hypothesis i, which is computed as the normalized number of observations the robot made in this place in the training data. The sum of the a priori probabilities of all place hypotheses is 1. For all other hypotheses $P(i) = 0$. Furthermore, D is a constant parameter controlling the rate of decay, set to 0.015, and Δt is the elapsed time since hypothesis i is evaluated last. As a result, hypotheses deactivate when they do not receive input activation from other hypotheses. When the activation value decreases below a minimum, the hypothesis is no longer evaluated, and removed from the dynamic network. A new hypothesis will be initiated when new evidence is found for a particular landmark or place. Therefore, every hypothesis in the two higher levels in the network corresponds to one occurrence of a landmark or place. Every hypothesis at the lowest level corresponds to a unique observation.

The activation value of the observation and landmark hypotheses is normalized to the maximum input activation, so that it is scaled between 0 and 1:

$$\begin{aligned} A_i(t) = & f_i(\Delta t) A_i(t - \Delta t) + \\ & n_i(t)(M - f_i(\Delta t) A_i(t - \Delta t)) \text{ for } i \notin R, \end{aligned} \tag{6}$$

where M is the maximum activation level, and $A_i(t - \Delta t)$ is the activation of hypothesis i when the network was last updated, multiplied with a decay $f_i(\Delta t)$, computed according to (5). Furthermore, $n_i(t)$ is the input activation as calculated in (4), and R the subset of hypotheses that represent places. It should be noted that the activation of the observation hypotheses will decay quickly, because they do not get any more input activation ($n_i(t) = 0$) after being initiated. In contrast, landmark hypotheses may get reinforced by new evidence from subsequent observations, and thus can stay active for a longer period of time.

For place hypotheses an expected activation is computed, which represents the expectancy to be at a place given the context. It is calculated using the information about the place transitions in the environment (see Figure 2). The expected activation of place i is the sum of all possible options to drive to place i:

$$\hat{A}_i(t) = \sum_j f_j(\Delta t) A_j(t - \Delta t) P(j \to i) P(j) \qquad (7)$$

$$\text{for } i, j \in R,$$

where $A_j(t - \Delta t)$ is the previous activation of place hypothesis j, multiplied with a decay $f_j(\Delta t)$, $P(j \to i)$ is the transition probability from place j to place i, , including $j = i$, the probability to stay in the same place. Finally, $P(j)$ is the a priori probability to be in place j.

The a priori transition probabilities from (7) are retrieved from the knowledge network. The probabilities are adjusted in the dynamic network of hypotheses, because the probability that the robot leaves a place increases as it is longer in that place. More specifically, the probability to stay in the same place decreases as a function of the age T_i (how long it is active) of the place hypothesis: $P(i \to i)(T_i) = P(i \to i)^{T_i}$. The probabilities to move to other places are increased proportionally to their a priori connection strength. For example, suppose the initial transition probability between place A and place B is 0.2, and the probability to stay in place A is 0.8. After the robot has observed landmarks in place A at four subsequent times, $P(A \to A) = 0.8^4 = 0.4$ and $P(A \to B) = 0.6$. When the robot returns to the same place, the probabilities are re-initialized to the probabilities in the knowledge network.

The expected activation is combined with evidence from the current input to compute the activation evaluation of the place hypotheses:

$$A_i(t) = \hat{A}_i(t) + K\left(\frac{n_i(t)}{\max(n(t))} - \hat{A}_i(t)\right) \text{ if } i \in R, \quad (8)$$

where $\hat{A}_i(t)$ is the expected activation according to (7), $n_i(t)$ is the input activation of i as calculated in (4), $n(t)$ is a list with the input activations of all active place hypotheses, and K is the gain factor. The gain factor is dependent on the noise in the observations. If the observations are very reliable, its value should be high. However, the current data set is relatively noisy. Therefore, the gain factor is set to 0.25, which entails that the model responds relatively slowly to new observations, and is guided more by expectancies.

The final activation values of all active place hypotheses are compared, and the one with the highest activation is the current best hypothesis of the place of the robot. Hence, the sequence of best hypotheses at each update gives the estimation of the model of the path of the robot.

3. Experiments

To illustrate the benefit of context evaluation in robot localization, we show the place predictions of two models. In the first model the predictions are based on instant observations alone, which implies that only information from the knowledge network is used. Accordingly, context-based spreading is not applied, because the data-driven model does not remember previous predictions. In other words, hypotheses of the place of the robot are deactivated after the data-driven activation spreading. In the second model, the expectancy-based model, the place prediction is based on a combination of instant observations and expectancies, which are computed through context evaluation, as discussed in section 2.3.

We discuss the results of both models running on two types of data. In the section 3.1 we present an experiment with simulated data, which can be controlled in their complexity. The simulated data are a simplification of the real data described in section 2.1. The experiment with the real data is discussed in section 3.2.

3.1 Simulated Data

We generated a data set to measure the performance of the model on data with different levels of noise. The noise simulates observations that are so similar that they are matched to the same landmark, although the observations are made at distinct places. These types of ambiguous observations occur often in the real data due to reoccurring objects and structures in office-like environments. At every time step one observation is simulated, which is matched to one landmark. The distance between the descriptor of the observation and the landmark is set to the same value for all observations. In the first lap all 240 landmarks are observed and connected uniformly to one of eight places. No noise was applied in the training part of the data, the first two laps, so there are no ambiguous landmarks in the a priori knowledge network. In the test data we applied a varying amount of noise on the landmarks. When no noise is applied to the data, the test set is identical to the training set. As the amount of noise increases, the place at which a landmark is observed becomes more random, until it is completely random at a noise level of 100 %.

3.2 Real Data

In the real data, as described in section 2.1, 225 unique landmarks are observed in the first two laps (the training data). In the operation phase, 107 of these landmarks are re-observed and used as input to the dynamic network. 114 new landmarks are detected in the operation phase, which are not processed by the current version of the model. 24 % of the landmarks in the knowledge network is ambiguous, that is, these landmarks are observed in more than one region in the training phase. The real data are quite challenging, because they contain noisy and erroneous observations, hold many ambiguous landmarks, and landmarks that are unequally distributed in the environment.

4. Results

The results of the model on the simulated data are shown in figure 3. Since the model keeps track of all hypotheses, there is a list of hypotheses with a decreasing activation value, not only a single winner. Hence, it is possible that the true place is not the best hypothesis, but the second best. Therefore, the performance of the model can be evaluated not only by comparing the true place to the best place hypothesis, but also to the top two or top three. Figure 3 depicts the single best result for the expectancy-based model and the data-driven model, and the top two and three of the expectancy-based model. The results of the data-driven model are identical for the top one, the top two, and the top three, because the simulated data set contains only one observation per time step, resulting in one possible hypothesis.

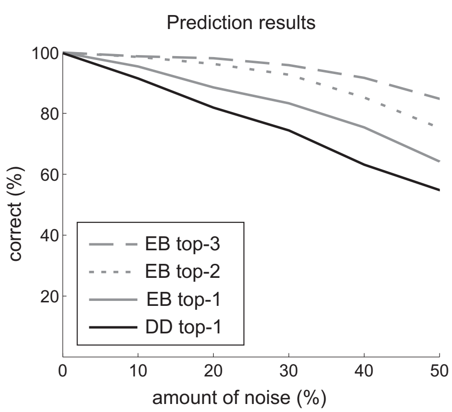

Fig. 3: Results of the model tested on data with a varying amount of noise. The single best result of the expectancy-based (EB) model and the data-driven (DD) model are shown, and the top two and top three of the EB model.

As expected, the data-driven model performs at chance level. When an incorrect observation is made, the place prediction is also false. The expectancy-based model performs better than the data-driven model, especially for low amounts of noise (< 50 %). (High levels of noise are not included in the figure, because the results are less meaningful if the noise is more prominent than the observations.)

In the experiment on the real data, of which the results are given in table 2, the expectancy-based model also outperforms the data-driven model. The difference between the score of the best hypothesis of both models is not very large, but consistent in multiple tests. However, the high scores on the top two and three are promising for future improvement.

It should be noted that the predictions of both models are based solely on visual observations, and odometric information is ignored. Therefore, the results of the two models can be compared by their performance on visual information, and we can show the advantage of the expectancy-based model. If one would aim at a best possible robot localization, odometric information should be included.

Table 2: Results of the data-driven and the expectancy-based model on data collected by a moving robot.

	Top-1	Top-2	Top-3
Data-driven	56%	69%	76%
Expectancy-based	63%	81%	88%

5. Discussion

We presented a model that dynamically manages a spreading activation network. This network represents the environment of an agent based on sensory information and knowledge of the environment. To test the applicability of the model in a real-world environment, we tested it on visual observations gathered by a mobile robot, with the goal to improve its localization. Learned knowledge about the environment of the robot is used to compute expectancies of its location. These expectancies are combined with instant observations to form a prediction of its location. Including expectancy in the prediction enhances the stability of the model, since it prevents unexpected landmarks from disrupting the place prediction.

The information about the environment is learned in a supervised training phase, and stored in a knowledge network, the long-term memory of the model. The short-term memory is represented by a dynamic network. Hypotheses in the dynamic network are more transient, because they represent the current state of the robot. The network and the deduced location prediction are updated when the robot gathers new evidence about its environment. The results of the experiments confirm that context evaluation improves the performance compared to data-driven evaluation on both the simulated and the real data.

Although the expectancy-based model outperforms the data-driven model, the difference on the top one in the experiment on the real data is not very large. This can be explained by the fact that more than half of the landmarks that the robot encounters during the operation phase are new. Hence, the information the model can base its prediction on is limited. Therefore, it will be useful to integrate an algorithm in the model that includes new landmarks in the knowledge network during the operation phase. For example, the Growing When Required (GWR) network of Marsland *et al.* adds new nodes to a network based on the (mis-) match between the data and the network [14]. Such an algorithm would make it possible to learn new information during the operation phase. Furthermore, incremental learning can be used to update existing connections based on new observations.

Another possible improvement can be made in the determination of expectancies. In the current version of the model we only update the network when observations are made. This can pose problems to the model, especially when the data is not equally distributed over the environment, causing some places to be poorly represented by landmarks. Based on temporal and odometric information, expectancies of the path of the robot can be made even without observations. Therefore, we are working on including this information in the model.

In conclusion, the presented model can improve robot localization through context evaluation. It is computationally efficient and needs little memory storage. Therefore, it can be easily scaled to larger environments. Moreover, the model is general, because the sensory information in the model is not limited to visual observations. Hence, it can be used for state estimation in other domains (see [7]), or even combine information from different modalities to make predictions.

Acknowledgment

This work is supported by SenterNovem (Dutch Companion project grant no. IS053013).

References

[1] M. Bar, "The proactive brain: using analogies and associations to generate predictions," *Trends In Cognitive Sciences*, vol. 11, no. 7, pp. 280–289, 2007.

[2] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends In Cognitive Sciences*, vol. 11, no. 12, pp. 520–527, 2007.

[3] M. Côté, E. Lecolinet, M. Cheriet, and C. Y. Suen, "Automatic reading of cursive scripts using a reading model and perceptual concepts," *International Journal on Document Analysis and Recognition*, vol. 1, no. 1, pp. 3–17, 1998.

[4] F. Crestani, "Application of spreading activation techniques in information retrieval," *Artificial Intelligence Review*, vol. 11, no. 6, pp. 453–482, 1997.

[5] L. Van Maanen, H. Van Rijn, M. Van Grootel, S. Kemna, M. Klomp, and E. Scholtens, "Personal publication assistant: Abstract recommendation by a cognitive model," In press, 2008.

[6] M. R. Quillian, "Semantic memory," in *Semantic Information Processing*, M. Minsky, Ed. Cambridge, MA: MIT Press, 1968, pp. 216–270.

[7] M. E. Niessen, L. van Maanen, and T. C. Andringa, "Disambiguating sound through context," *International Journal on Semantic Computing*, vol. 2, no. 3, pp. 327–341, 2008.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, R. C. Arkin, Ed. Cambridge, MA: MIT Press, 2005.

[9] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots – an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359–371, 2007.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal Of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[11] G. Kootstra and S. De Jong, "Using local symmetry to select landmarks for visual SLAM," Submitted, 2009.

[12] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[13] J. L. McClelland and D. E. Rumelhart, "An interactive activation model of context effects in letter perception: I. an account of basic findings," *Psychological Review*, vol. 88, no. 5, pp. 375–407, 1981.

[14] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural Networks*, vol. 15, no. 8-9, pp. 1041–1058, 2002.