Image Invariant Robot Navigation Based on Self Organising Neural Place Codes

Kaustubh Chokshi *, Stefan Wermter, Christo Panchev, and Kevin Burn

Centre for Hybrid Intelligent Systems, School of Computing and Technology, University of Sunderland St. Peter's Campus, Sunderland SR6 0DD. United Kingdom.
http://www.his.sunderland.ac.uk

Abstract. For a robot to be autonomous it must be able to navigate independently within an environment. The overall aim of this paper is to show that localisation can be performed even without having a pre-defined map given to the robot by humans. In nature place cells are brain cells that respond to the environment the animal is in. In this paper we present a model of place cells based on Self Organising Maps. We also show how image invariance can improve the performance of the place cells and make the model more robust to noise. The incoming visual stimuli are interpreted by means of neural networks and they respond only to a specific combination of visual landmarks. The activities of these neural networks implicitly represent environmental properties like distance and orientation to the visual cues. Unsupervised learning is used to build the computational model of hippocampal place cells. After training, a robot can localise itself within a learned environment.

1 Introduction

Despite progress made in the fields of AI and Robotics, robots today still remain vastly inferior to humans or animals in terms of performance [1]. One reason for this is that robots do not possess the neural capabilities of the brain. Human and animal brains adapt well to diverse environments, whereas artificial neural networks are usually limited to a controlled environment, and also lack the advantage of having millions of neurons working in true parallelism.

In an mammal's brain place cells fire when the animal occupies a familiar portion of its environment, known as its place field. However, the activity of cells, or even a collection of such cells, simply indicates different locations; it informs the animal where it is, but it cannot directly inform the animal where it should go [2–4]. One role of the place cells is to associate a path integrator and local view so that when an animal enters a familiar environment, it can reset its path integrator to use the same coordinate system as during previous experiences in the environment [5].

To navigate in familiar environments, an animal must use a consistent representation of its positions in the environments. In other words, the animal must localise in order to navigate within the environment. Visual clues that support a local view to inform the animal of its initial position may be ambiguous or incomplete and there must be

* Corresponding Author: kaustubh.chokshi@sunderland.ac.uk

a way to settle on a consistent representation of localisation [4]. The evidence from neurophysiology suggests that place cells are well suited for this role. Spikes fired by dentate granule cells, CA1 and CA3 pyramidal cells, are strongly correlated with the location.

In a experimental environment, place cells have clearly shown a firing rate that relates to that environment. From the experimental evidence [5] we can summarise their properties as follows:

1. When distinct landmarks move, place fields also move proportionately.
2. Place cells continue to show clean place fields when landmarks are removed.
3. The firing rate correlates to more than the location of the animal.
4. Place cells show different place fields in the environment.
5. Place cells are directional when the animal takes a limited path, but non-directional when wandering around randomly in open fields.
6. Place cells are multi-modal. They can integrate various input sensors to localise with vision being the primary one. In the case of no vision or restricted vision, they localise using other sensors such as odour, or whiskers.

In this paper we evaluate our computational place code model in a realistic context, using a Khepera robot. Visual information is provided by a linear vision system. Eight infra-red sensors are used to provide reactive behaviour. This paper is structured as follows: we describe the basis of the model in section 2, outline of the model in section 3, following with the experiments and results in section 4.

2 Self Organising Map for Localisation

In the brain, hippocampal pyramidal cells called place cells have been identified that fire when an animal is at a certain location within its environment. In our model, we show that place cells based on SOMs have potential to provide locations to the path integrator and place cells can localise the robot in a familiar environment. Self-localisation in animals or humans often refers to the internal model of the world outside. As seen in a white water maze experiment [4], even though a rodent was not given any landmarks, it could still reach its goal by forming its own internal representation of landmarks of the world outside. It is seen in humans and animals that they can create their own landmarks, depending on the firing of place cells [6]. These cells change their firing patterns in an environment when prominent landmarks are removed. With this evidence from computational neuroscience, it is reasonable to assume that a model of place cells might prove to be an efficient way of robot localisation using vision.

One possibility is to build a place code model that is based on Self Organising Maps (SOM). SOM [7] networks learn to categorise input patterns and to associate them with different output neurons, or a set of output neurons. Each neuron, j, is connected to the input through a synaptic weight vector $\mathbf{w}_j = [w_{j1}....w_{jm}]^T$. At each iteration, the SOM finds a winning neuron **v** by minimising the following equation:

$$v(x) = arg\ min_j \|x(t) - w_j\|, \qquad j = 1, 2, ...n \tag{1}$$

x belongs to an *m*-dimensional input space, $\|.\|$ is the Euclidean distance, while the update of the synaptic weight vector is done in the following way:

$$w_j(t+1) = w_j(t) + \alpha(t)h_{j,v(x)}(t)[x(t) - w_j(t)], \qquad j = 1, 2, ...n, \qquad (2)$$

This activation and classification are based on features extracted from the environment by the network. Feature detectors are neurons that respond to correlated combinations of their inputs. These are the neurons that give us symbolic representations of the world outside. In our experiments, once we get symbolic representations of the features in the environment we use these to localise the robot in that environment.

The sparsification performed by competitive networks is very useful for preparing signals for presentation to pattern associators and auto associators, since this representation increases the number of patterns that can be associated or stored in such networks [8, 9]. Although the algorithm is simple, its convergence and accuracy depend on the selection of the neighbourhood function, the topology of the output space, a scheme for decreasing the learning rate parameter and the total number of neuronal units [10].

The removal of redundancy by competition is thought to be a key aspect of how the visual system operates [8, 9]. Competitive networks also reduce the dimensions of the input vector as a set of input patterns, in our case pixels of the input image vector. The representation of a location is achieved by activation of a neuron.

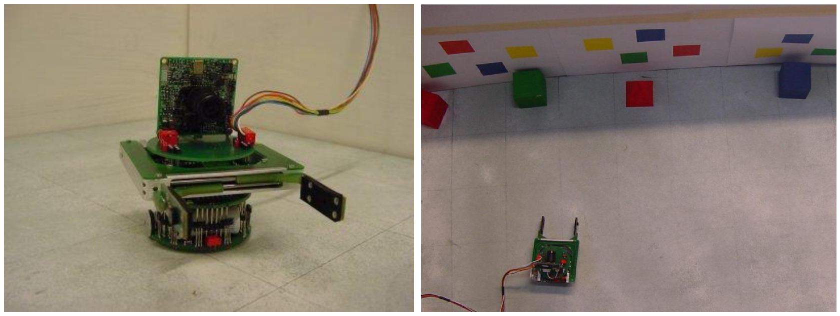
An important property of SOMs is feature discovery. Each neuron in a SOM becomes activated by a set of consistently active input stimuli and gradually learns to respond to that cluster of coactive inputs. We can think of SOMs as feature discovery in the input space. The features in the input stimuli can thus be defined as consistently coactive inputs and SOMs thus show that feature analysers can be built in without any external teachers [8]. This is a very important aspect of place cells, as they have to respond to unique features or landmarks in the input space in order to localise the robot.

3 Scenario and Architecture

Our approach is to try to model aspects of neural visual localisation present in human and animal brains. The main emphasis of this research is to build a robot that uses robust localisation, with the objective that it has learning and autonomy. The central objective is on natural vision for navigation based on neural place codes. This section summarises the scenario and architecture of our approach.

3.1 Scenario

In our experiments the overall goal for the robot (a Khepera robot, figure 1(a)) was to to localise itself between two desired locations. In order to facilitate natural vision experiments, we provided random colour-coded squares on the wall, along with some distinguishable features like cubes, cylinders and pyramids randomly kept in the environment as shown in figure 1**(b)**. During experimentation, the robot should be able to create its own internal representation of the world model based on unsupervised learning for neural place codes.



(a) **(b)**

Fig. 1. (a) A Khepera robot used during experimentation. **(b)** A birds eye view of the overall experiment setup and the cage in which the robot was allowed to move in.

3.2 Overall Architecture of the Model

Humans and animals use various sensors to navigate [11, 12]. In our robot model, we are primarily using vision as a global navigation strategy and for our local navigation strategy we have employed the use of infra red sensors.

Our approach is based upon functional units each of which uses a neural network. An overview of the different functional units can be seen in figure 2. SOMs are used for the visual landmarks, which enable the robot to generate its internal representation of the world based on the most salient features in its visual field. A primitive visual landmark allows us to implement simple, visually-based behaviour. The transform invariance and pattern completion modules are based on MLPs, the output of which forms the input to the SOM. Furthermore, self-localisation and target representation are based on SOMs.

In figure 2, 'visual information derivation' is a module which is responsible for getting the images from the robot's camera. The Visual information derivation module is responsible for image pre-processing and normalising the images for the network. Transform invariance, a part of our localisation module (figure 5) makes use of associative memory and pattern completion for noise reduction. The localisation module is responsible for the localisation of the robot in the environment.

Effective navigation depends upon the representation of the world the robot is using [11]. In our architecture the world representation is called 'spatial representation'. This provides the path planning module with necessary information from the localisation module and visual target module. It maps both the current location and the location of the target into the same map and enables the path planning module to compute the most appropriate path. Once we can map both the visual target and the current location of the robot into the same spatial representation, the 'path-planning module' directs the robot to its goal. The path planning can derive a path which is the shortest and quickest way towards the goal.

There are various ways in which the robot can be instructed as to where its target for navigation is. We are exploring how to translate the place code output and target

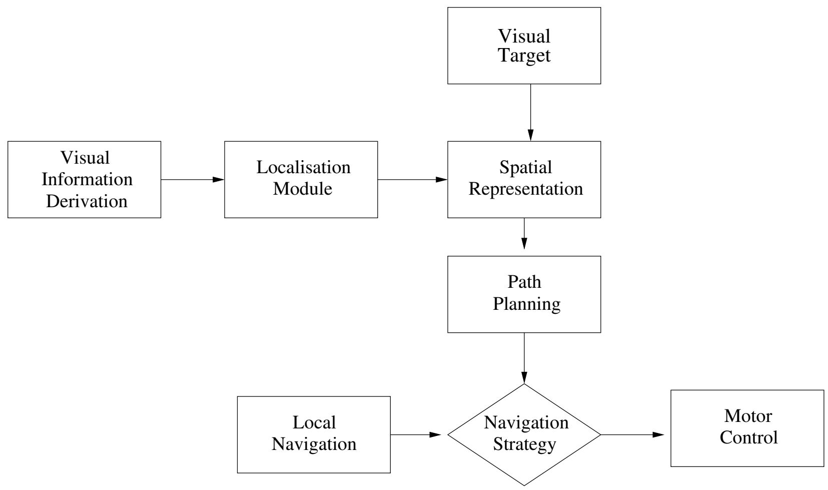


Fig. 2. Overall architecture used in the visual navigation strategy of the robot. This shows the flow of the model.

representation into a spatial representation. The path planning module provides output to the 'motors control'. This forms the global navigation strategy.

We have implemented the local navigation strategy using reactive behaviour. Both the global and local navigation strategies meet each other in the navigation strategy module, which is mostly responsible for choosing motor commands either from local or global behaviours. Accordingly, it chooses the output from either the global navigation strategy or local navigation strategy to generate motor control commands.

Real-time processing of artificial neural networks requires vast amounts of computational power, especially those algorithms that require real-time vision. Therefore, we make use of distributed and decentralised processing. The robot onboard computer is primarily responsible for robot control. At the same time we are making use of various off-board computers in order to achieve the real time navigation. Each module in figure 2 can run on a different computer/CPU as a part of distributed architecture.

3.3 Overview of the Model Implementation

The focus of this paper is on robot localisation and therefore in this section we will describe in detail the implemented models. It consists of a hierarchical series of five layers of hybrid neural networks, corresponding to the transform invariance layers and place code layer. Figure 4 shows the forward connections to individual layers derived from the modular arrangement of the layers.

Local Navigation: Reactive Behaviour A lot of recent research in intelligent robotics involves reactive behaviour [13, 14]. In a reactive robot system, all sensors are wired to the motor controls. This enables the motors to react on the sensory state. In these systems internal representations play a limited role or no role at all in determining the motor control output for the robot. Even though reactive behaviour robots do not have an internal representation of the outside world, they are able to solve many complex tasks, since the robot can react to different sensory states in a different manner based upon coordination of perception and action [15, 16].

As in biological systems, reactive behaviours have a direct mapping of sensory inputs to motors actions [11]. The reactive behaviour emerges as a result of **SENSE** and **ACT** strongly coupled together. Sensing in reactive behaviour is local to each behaviour, or in other words it is behaviour-specific. One behaviour is unaware of what the other behaviour is doing, i.e. the behaviours are independent of each other and they do not interact with each other. This is the fundamental difference between local and global navigation strategies.

Our neural network design for reactive behaviour (figure 3) is based on Braitenberg's Vehicle [17, 16], with eight infrared sensors forming the input layer. The inputs were pre-processed to toggle the actual input between 0 and 1. The output layer had two nodes, one connected to the left wheel, another to the right wheel and direction was determined by the value of activation between -1 and 1: positive activation for the forward direction and negative activation for backwards.

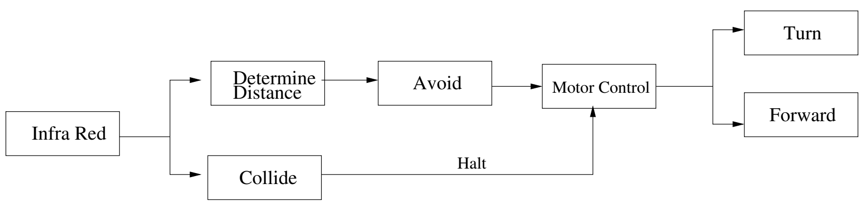


Fig. 3. Control system for robots reactive behaviour.

Constructing a local navigation system by behaviours is often referred to as programming by behaviour [11], since the fundamental component of any implementation is a behaviour. Behaviours are inherently modular and easy to test in isolation i.e. they can be tested independently of the global navigation. Behaviours also support incremental expansion of the capabilities of the robot and a robot becomes more ”intelligent” with more behaviours in it. The reactive behaviour decomposition results in an implementation that works in real time and is computationally inexpensive. If the behaviours are implemented poorly, then the reactive implementation can be slow. But generally, the reaction speed of a reactive behaviour is equivalent to the stimulus-response time in animals [11].

Global Navigation and Self Localisation The global navigation strategy is crucial for how systems behave. Global navigation requires perception and motor skills in order to provide complex sensor-motor integration enabling the system to reach its goal. The global navigation strategy is the strategy which uses an internal representation, or map, of the environment while local navigation does not make use of such representations or maps. Many of these global planning methods are based on paths without obstacles

[18] and their main advantage is to prove the existence of a solution that will permit the robot to reach its destination. Thus both reactive and deliberate planning are needed, not only bottom-up reactive, but also top-down predictive behaviour.

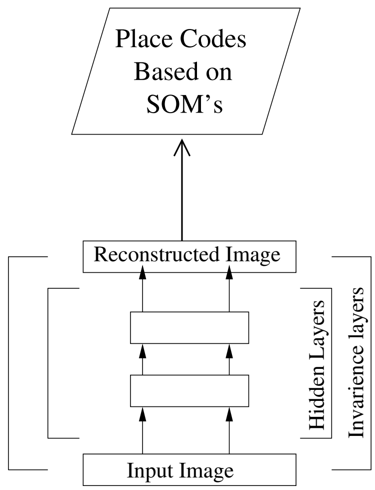


Fig. 4. Overview of the neural model that is being used in our experiments. Hybrid neural networks for localisation based on vision. The first part of the hybrid neural networks is associative memory based on associative memory for invariance and higher layers are SOM for place codes.

Although localisation has been investigated for more then a decade, there is still no universally accepted solution [19]. Some general methods have been employed for encoding prior knowledge of the environment and matching it with local sensor information. Some of the previous methods of localisation are *(i) Topological Maps*: the environment is mapped into a number of distinct locations, usually connected with each other [20]. Typically these maps are learned during the exploration stage. *(ii) Evidence grids* [21]: in this method each location in the environment is represented by a grid point in the global map. For localisation, the system constructs local grip maps with occupancy probability for each grid point which are matched to the global map. *(iii) Markov Models*: in this method of place code localisation the probability distribution is computed for all possible locations in the environment [22]. *(iv) Landmarking*: in this method the robot encodes a number of distinctive locations [23, 20, 24, 25, 2].

Our method of localisation is based on landmarks. We use landmarks for localisation, mainly because this enables us to make internal representation of the environment and does not involve human interference for determining the landmarks in the environment. As the robot generates its own landmarks depending on the features in the environment, we call it "Self-Localisation". Our method of localisation is distinct from other methods described above because there are not maps given to the robot and the neural network creates an internal representation of the world based on the visual stim-

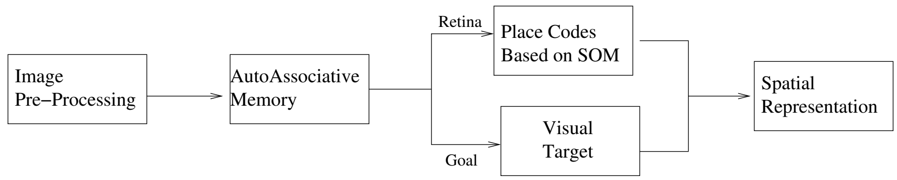


Fig. 5. An overview of the localisation in our model. This is the overall implementation of the localisation module as been before in figure 2. The image pre-processing is responsible for getting the images from the robot camera and resizing. Then the associative memory is responsible for image-invariant processing. Localisation of the robot is done by place codes. Visual Target represents the main goal for the navigation. Spatial representation will take activations from both the neural network regions and represent it in the environmental space.

uli. This network is inspired from neural place codes, making the model more robust and efficient.

4 Experiments and Results

Once the environment was learned by the robot, and when the landmarks were presented to it, it could be seen clearly that the activation in the SOM's map would represent a place code. There were some interesting observations made, including that the landmarks were not only self-generated, but also that when the robot starts to approach the landmark there was activation in neighbouring fields before it reached it, as discussed in section 4.5. Another observation was that the neuron responsible for the landmark would have increasing levels of activation when it was approached by the robot.

4.1 Experimental Setup

The experiments were conducted on a Khepera robot. The robot was introduced in a closed environment, as seen in figure 6 of about 2m x 1.5m, which was divided into four parts: north, south, east and west. The environment was further divided into a grid of 10 cm x 10 cm squares. This grid was only used for the purpose of calculation of the error by the place cells. All the landmarks were placed against the wall of the cage. There were cubes and pyramids of different colour codes spread across the walls of the cage randomly. The walls also had randomly colour-coded figures on it.

Each square represents a place code. Each cell was given a name, depending on where it was located, for example a cell in the southern part within the eastern part was given name "*se10*". The naming convection was simple; the first letter represents which hemisphere, the second letter which block and the numbers indicate the x and y coordinates. This information was purely for our use in order to test the results and set up the experiments. This information was not provided to the robot. For training purposes there were 4 images taken from each of the place codes. For testing there were 10 new images from each place code in the environment.

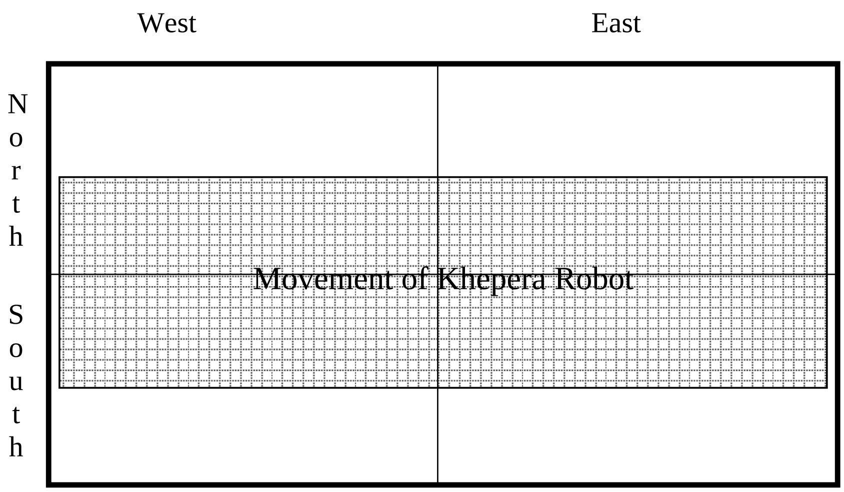


Fig. 6. The arena of the robot. The robot was allowed to moved in a limited space, as we did not want it to be too close to, nor too far away from, the landmarks. The area of movement of the robot was divided into smaller grids of 10cm x 10cm, giving us an approximate position of the robot.

4.2 Representation of Visual Input

It is more convenient to perform neural analysis on smaller versions of an image while retaining all the essential information of interest, in our case to detect a landmark in an image using neural networks. If the landmark is just as evident in the smaller image, it is more efficient to reduce the image size before applying neural methods. Thus computational time saving occurs as the smaller images contain fewer pixels and the recall time for associative memory and self organising map is reduced, since the number of the neurons is reduced.

There are number of techniques that can be used to enlarge or reduce images [26–28]. These generally have a tradeoff between speed and the degree to which they reduce salient visual features. The simplest methods to reduce the image keep every n^{th} pixel. However, this results in aliasing of high frequency components. Therefore, a more general case of changing the size of an image by a given factor requires interpolation of colours. The simplest method is called "nearest neighbourhood", which is currently used by us. Using this method one finds the closest corresponding pixel in the original image (i, j) for each pixel in the reduced image (i', j'). If the original image has the dimensions width (w) and height (h), and the reduced image would be of w' and h', then the point in the destination is given by

$$i' = iw'/w \tag{3}$$
$$j' = jh'/h \tag{4}$$

where the division (equation 4) is a integer, the remainder being ignored. In other words, in the nearest neighbour method of resizing, the output pixel is assigned the value of the pixel that the point falls within. The number of pixels considered affects the complexity of the computation.

Once the images are resized, they are arranged to a single dimension vector from a two dimensional vector. All images were in 24 bit colour RGB (Red Green Blue)format (equations 8), N represents the whole image.

$$N = (a_{ijk}) \in \mathbb{A}^{3mn}, \quad i = 1,2,\ldots,m \quad j = 1,2,\ldots,n \quad k = 1,2,3 \tag{5}$$

$$R = (r_{ij}) \in \mathbb{A}^{mn}, \quad i = 1,2,\ldots,m \quad j = 1,2,\ldots,n \quad r_{ij} := a_{ij1} \tag{6}$$

$$G = (g_{ij}) \in \mathbb{A}^{mn}, \quad i = 1,2,\ldots,m \quad j = 1,2,\ldots,n \quad g_{ij} := a_{ij2} \tag{7}$$

$$B = (b_{ij}) \in \mathbb{A}^{mn}, \quad i = 1,2,\ldots,m \quad j = 1,2,\ldots,n \quad b_{ij} := a_{ij3} \tag{8}$$

$\mathbb{A}$ is the set of possible pixel values. The values are between 0 and 255, and can be represented as shown in 9.

$$\mathbb{A} = \{0,1,2,\ldots,255\}, \qquad m = 17, \qquad n = 27 \tag{9}$$

Each image was reduced to a size of 17 x 27. This image in turn was converted into a single vector to be presented to the network. It was done as explained from equations 10 to 12.

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \qquad A_i = (a_{i1},\ldots,a_{in}) \qquad i = 1,2,\ldots,m \tag{10}$$

$$V = (v_l) := (A_1,\ldots,A_m) \in \mathbb{A}^{mn} \qquad l = 1,2,\ldots,mn \tag{11}$$

Equation 11 is a concatenation of A_i of A. In other words,

$$v_{(i-1)n+j} := a_{ij} \qquad i = 1,2,\ldots,m \qquad j = 1,2,\ldots,n \tag{12}$$

4.3 Training and Testing Procedure

The stimuli used for training and testing our model are specially constructed to investigate the performance of localisation using the self organising maps. To train the network, a sequence of 200 images was presented to represent over 20 landmarks. At each representation the winning neuron was selected and the weight vector of the winning neuron was updated along with the distance vector. The presentation of all the stimuli across all the landmarks consists of one epoch of training. In this manner the networks were trained using backpropagation in Multi-Layered Perceptrons. Invariance and the place code networks were trained separately.

	Training	Testing
No. of Images	200	485
No. of Epoch (Invariance)	1800	-
No. of Epoch (Place Code)	3000	-

Table 1. Training and Testing procedure

4.4 Results for Transform Invariance

This method of representation shows promise, although it is not ideal for invariance such as size, view etc. It is observed that, compared to the traditional template matching methods, it is computationally and memory efficient. After the neural networks memorised the various landmarks, and after a new image has been given at the retina, our method finds the image nearest to the image previously memorised.

The main purpose of transform invariance was to reconstruct the image that was on the retina for the SOM. It was seen that due to this process of reconstructed images, we could also achieve a certain degree for independence from light conditions. The independence of this was achieved due to the generalisation feature of neural networks, which would generalise the effect of light over various colours in the reconstructed image. Transform invariance has improved the performance of place codes based on SOMs in various way, and the results will be described in sections 4.5. and 4.6.

4.5 Discussion of Results for SOMs Place Codes

Activation Activity of a Neuron When an animal approaches a desired landmark, the place cells representing the landmark increase activation and, when the animal is at that the desired landmark, the activation is maximum. This property is observed in biological place cells and has been discussed in 1.1.

To show that our model also follows the same principles of biological place cells, we have taken readings of activation of various neurons and we are presenting here activation levels of a neuron responsible for different place codes. In figure (7) we can see that when the robot starts to approach the landmark, there is a sudden steep rise in the activation of the neuron responsible. As the robot gets closer, the activation keeps on rising, until it is at the desired landmark. Once there, the activation is 1. As the robot moves away from the desired landmark, there is a gradual fall in the activation of the neuron, and as soon as the landmark is out of sight of the robot, the activation is set to 0. This is shown in figure 7.

Activation Activity of a Place Code Cluster Another property of place cells is that when the animal is approaching the desired landmark, the neighbouring neurons would also be active as described in section 1. These results are shown in figure 8. It is seen that when the robot is within a region there is a cluster of place codes responding to the robot's location. In section 4.6, we will see the advantages of having clusters of place cells for noise handling. The activation in the cluster provides us the robot's grid

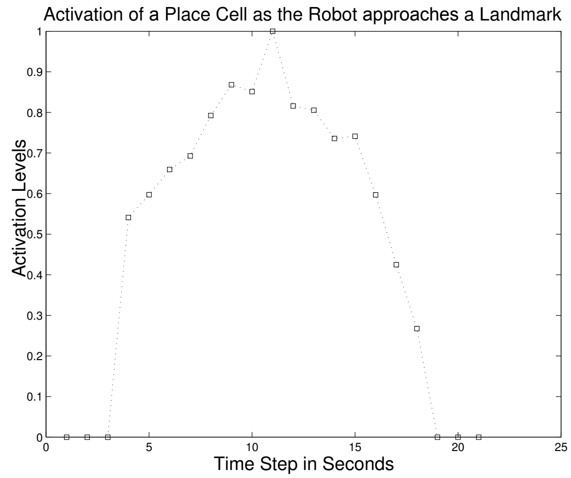


Fig. 7. This figure shows activation levels and activity for a single place cell. As the robot approaches the place code, the activation rises and when the robot is at the location represented by the place cell the activation is maximum.

location. Each neuron within the cluster provides us with a more precise location of the robot, within 2cm of precision.

Clustering of Place Codes The basic property of a SOM network is to form clusters of information relating to each other, in our case landmarks. A cluster is a collection of neurons which are next to each other representing the same landmark. Figure 8 shows that when the robot was approaching the desired landmark, there were activations in the neighbouring neurons. This is due to clustering of similar images around the landmark. There are multiple similar images that are being represented by a single neuron, making the cluster smaller and richer in information. This is achieved with the invariance module.

On the other hand, figure 8(c) shows the landmarks which were at a distance to the location represented in figure 8(d). Two landmarks that were given to the robot at a distance would be mapped not only into different clusters, but also distant from each other. By their very definition, landmarks are features in the environment. This was the reason behind a formation of these clusters by SOMs. The landmarks that were chosen by the SOM were quite significant in the image and distinguished features from the rest of the environment, and other landmarks.

Distinction between North and South We have observed that there is a clear distinction between the north and the south on the place code map. Figure 9(a) shows all the neurons that are responsible for landmarks in the north and figure 9 (b) represents the neurons responsible for the south. The reason for this distinction is that the field of view

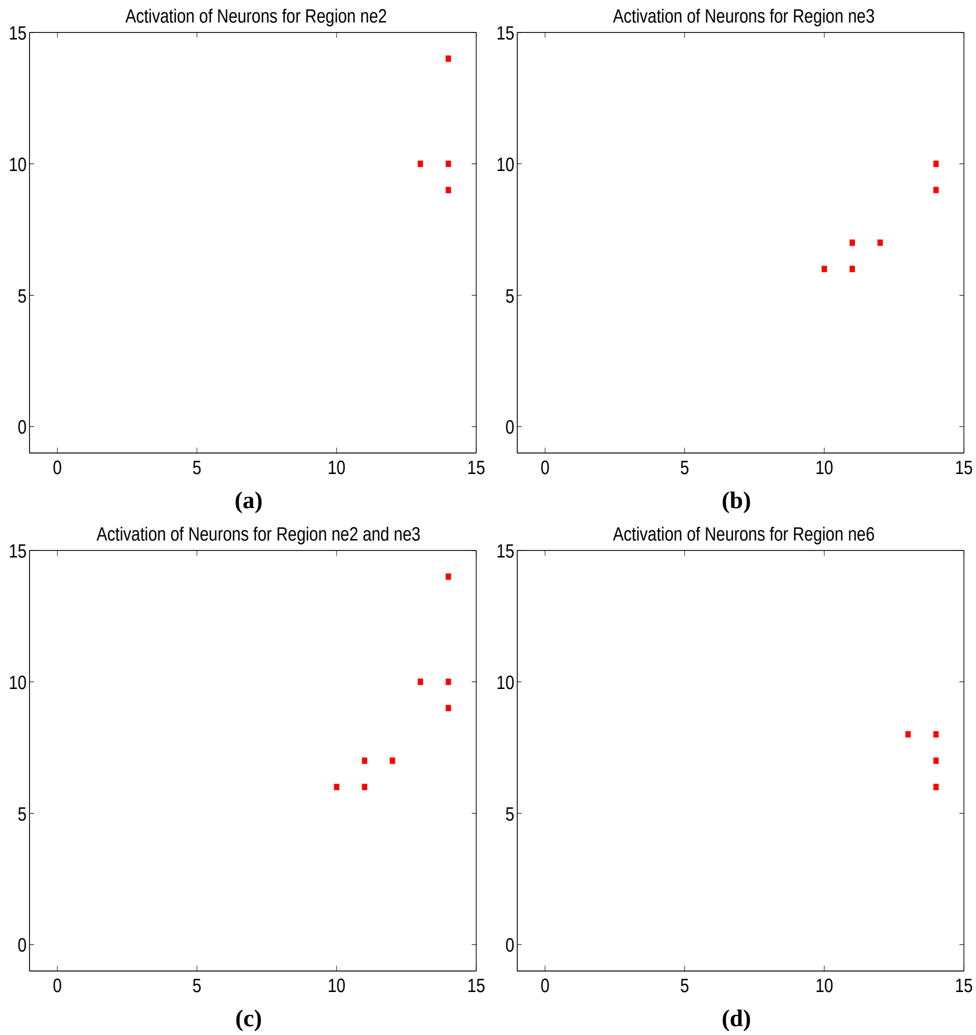
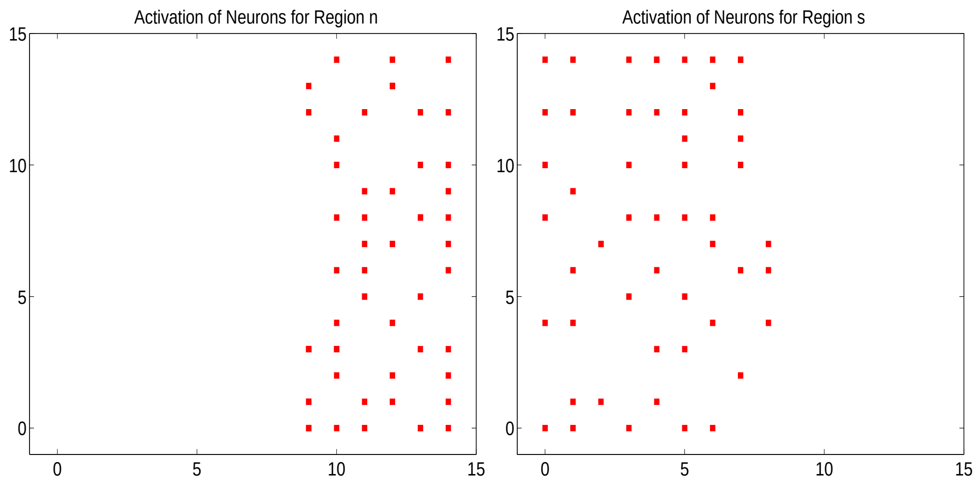


Fig. 8. Each of the graph shows activation of winning neurons. It is seen in the images that neighbouring regions in environment are neighbours to each other on the place code map. There is also a clear overlap of a neuron in both regions. The reason for the overlap is because in the field of view of the robot between both locations, both prominent landmarks can be seen.

is quite different in both hemispheres. It has been observed that in the northern hemisphere the object of attention for the landmark selection was very much limited to an object in the field of view. In contrast, in the southern hemisphere, the robot had a much larger field of view, therefore the object of attention was not focused only on a single object, but on various objects.

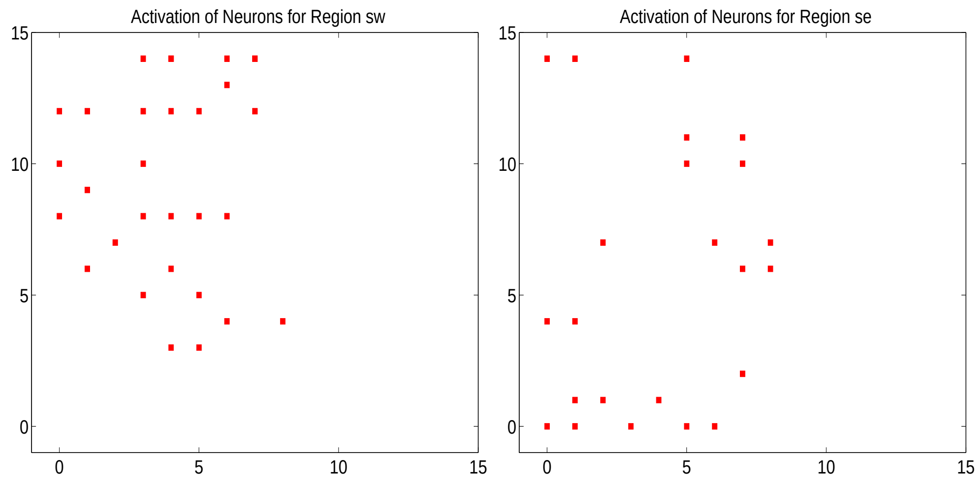


(a) **(b)**

Fig. 9. The various activations of neurons that represent landmarks in the northern and southern hemisphere. We can also see that they are not topological.

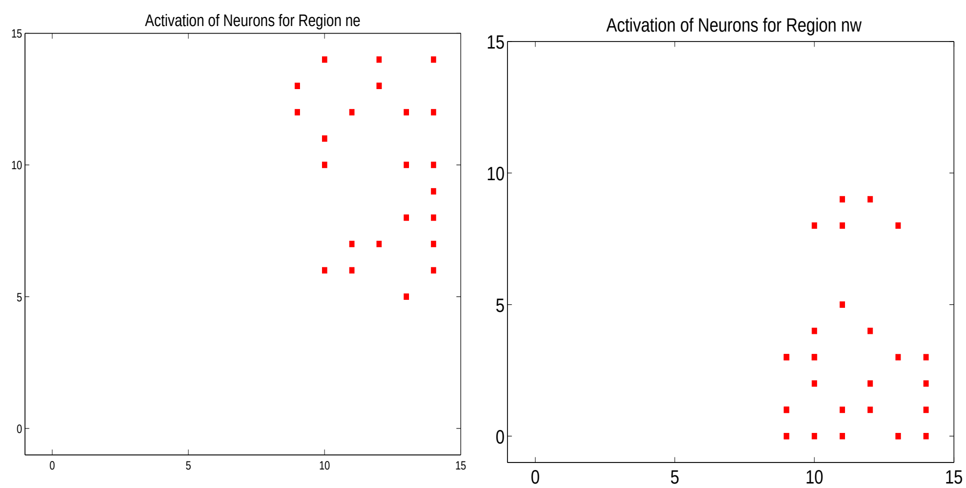
Overlap of East and West in the South Section In the north it was observed that there was a clear distinction between the east and west, whereas in the south, there is overlap. The overlap was caused by the field of view of the robot retina. From the south section of the cage, the robot is able to see more landmarks than in the north section. After small movements in the north, the object of attention changes, whereas in the southern hemisphere there are various objects of attention that lie in the field of view. Therefore, minor movements in the southern hemisphere, do not lead to drastic changes in the visual field. The overlap is caused by landmarks which are nearer the borders of east and west.

Directional Independence It was clearly observed that the model was directionally independent. It was seen that in whichever direction the robot travelled within the environment, as it came across the landmark, it would activate the place cells responsible for the landmark. Therefore it did not matter in which direction the robot travelled; it could localise itself as soon as it came across a significant landmark.



(a) **(b)**

Fig. 10. There is a clear overlap of the regions in the (a) south-eastern and (b) south western. The overlap is due to the sharing of landmarks between both the regions.



(a) **(b)**

Fig. 11. In the images above, there is a clear distinction between the (a) north eastern and (b) northwestern. The reason for this is that the field of view here is restricted to a particular landmark.

For testing purposes, the model was presented with random images with different landmarks. It was even seen that once the landmark was seen, the place cell responsible for it would be activated.

Manipulating the Landmarks There were two experiments that were conducted for testing robustness with the robot. In the first experiment, we removed two blocks from eastern part and blocks from western part randomly. It was observed that when the robot was in the northern hemisphere and came across the missing blocks, it would activate an unknown cluster or completely wrong neuron. This happened because the field of view within the northern hemisphere is very limited to one or two objects. So, when the blocks were removed, the environment was not rich enough to provide the robot with enough clues as to where it was, but as soon as the robot came across a known landmark, it localised itself again. This was not observed in the southern hemisphere. The southern hemisphere visual field was large, hence the removal of blocks did not affect the activations of neurons. The visual field was rich enough to provide the necessary clues to localise.

In the second experiment, the blocks that were removed were now replaced, but not in their original positions. It was observed that the activations in the southern hemisphere were still representing the right location. In the northern hemisphere, the activations were not in the unknown cluster, but for the neurons representing those landmarks.

Reduction in Cluster Size It was observed in [23], that the main cause for large clusters of place codes was due to the SOM trying to handle transform invariance by having the neighbouring neurons responding to the invariance. With the use of associative memory for transform invariance, the size of the clusters was reduced. In the present model, the SOM does not represent the invariance, rather it represents the place codes. Images were collected at every 10th frame i.e. approximately half a second between images. This causes large amounts of overlap and large amounts of transform invariance. The associative memory clustered the similar images and reduced the transform invariance. The number of neurons per location reduced, since there were fewer neurons required to represent the same location if there was a shift in the images. This also has additional benefits, mainly now SOM can represent more place codes without actually growing or increasing the size of the map.

4.6 Performance of Network

To test the performance of the network, we tested it with white noise with a mean noise ranging from 0.0 to 0.8 and variance of 0.01 to the image. The effects of the noise on the images can be seen in figure 13. The aim of the neural network is to localise the robot within its environment, based on the internal representations it has formed. As the place cells are based on SOMs, there is a cluster of neurons responsible for a place code in the environment. The neuron representing that place code would be more accurate then the neighbouring neurons. To have a more precise localisation, we need the neuron responding to the place to be active.

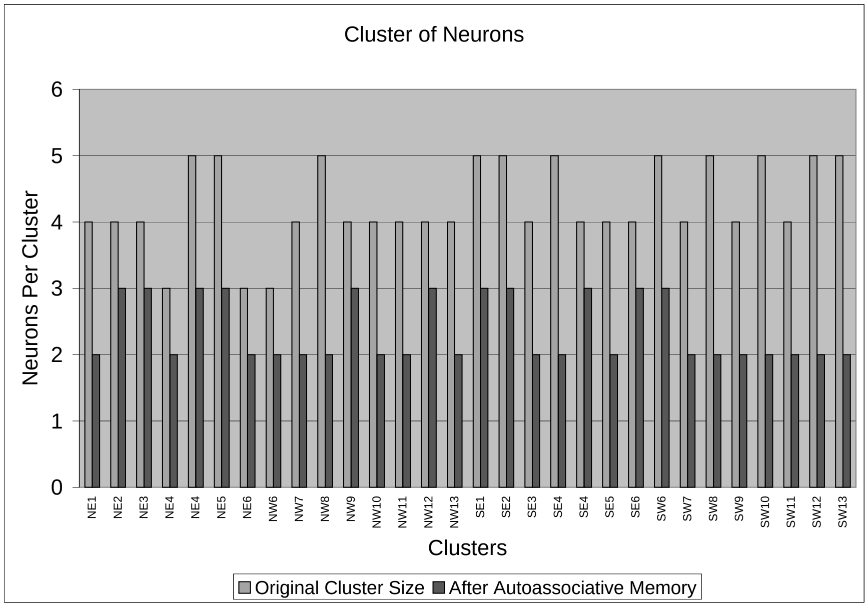


Fig. 12. The cluster sizes i.e. number of neurons per cluster representing a particular location of the robot. It is seen that There is a drastic reduction in the size of the clusters after the use of autoassociative memory before the place code map. This makes it possible for us to have more place code on the same size of the map.

There are various reasons where we would need an approximate localisation. It was noted that with the increasing noise, it was more likely for the robot to be ‘lost’ (unable to localise itself) [23]. During these times, approximate coordinates would help the robot to localise itself. We consider two options for localisation with noise handling: one being that a neuron responsible for the robot response and another being another neuron in the cluster of neurons responding for the same place responding. In the later case, localisation may not be very accurate.

As seen in figure 14, the clusters are more robust than the neurons with regard to noise handling. As the amount of noise increases, the neurons or cluster response to localisation becomes more random. However the network performance is quite impressive until 0.5 mean deviation of noise where the error for localisation is below 30%. The cluster for a place codes still performs much better and is still below 20%.

Noise handling by the neural networks was also improved by adding an additional layer of associative memory below the place codes. The associative memory reduces the noise before the outputs are given to the place codes layer in the architecture. It can be seen in figure 14 that associative memory helps place cells to perform better, giving less error per neuron.

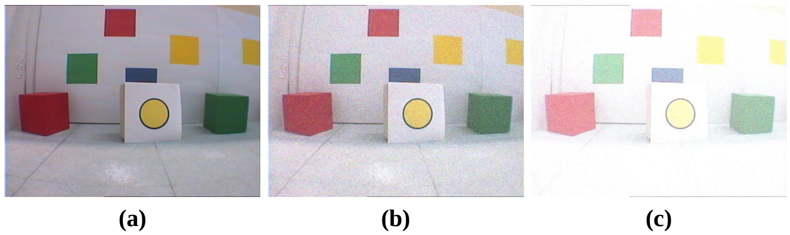


Fig. 13. Effects of different noise levels added to the image. **(a)** Image without any noise. **(b)** Image with 0.2 mean deviation **(c)** Image with 0.5 mean deviation

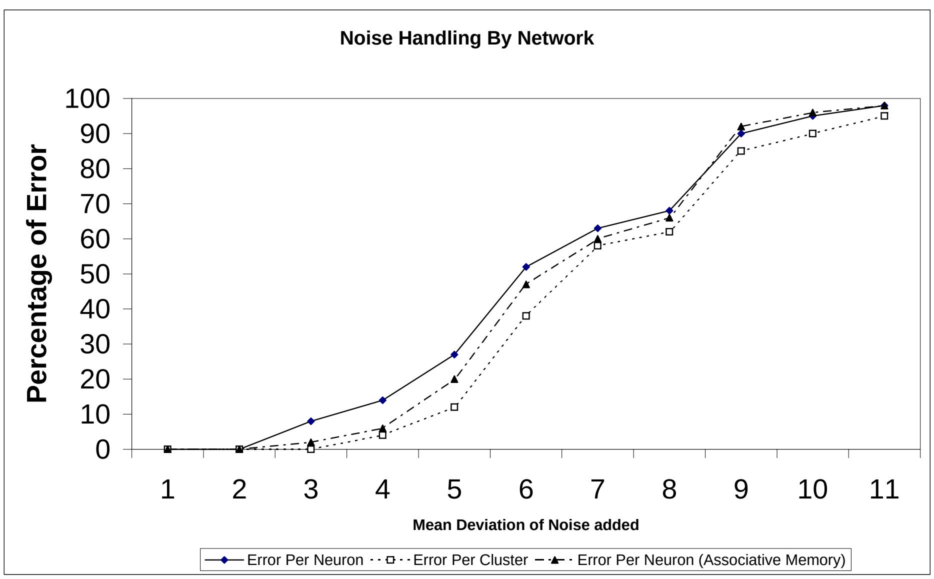


Fig. 14. This figure shows the noise handling by the network, with transform invariance, based on associative memory, and also without transform invariance. It shows that the network performs much better with transform invariance.

It was also noted that as the noise level increased, the performance of the network decreased along with associative memory. This was mainly seen for localisations where higher levels of noise were present. With noise of 70%, the performance was not significantly different, with or without associative memory. At higher levels of noise of more than 80% the noise handling of the associative memory failed, making the place codes performance performs worse than without the layer. At can be expected, with 80% noise level, the performance of the SOMs is completely random to get the right localisation.

For our experiments we were not expecting more than 30% of noise levels. Since this would mostly be caused by interference in wireless signals. Even if the noise levels for a few frames is more then 30%, it would be possible for the robot to localise itself with the following input frames. With the use of the associative memory, the performance up to 30% noise level improved substantially.

5 Conclusion

The aim of this research, was to investigate whether the reliability of robot localisation can be improved using SOMs based on place codes. In this paper we have described a place cell model based on a SOM for localisation. The model was successful in learning the locations of landmarks even when tested with distorted images. Visual landmarks were associated with locations in a controlled environment. This model clusters neighbouring landmarks next to each other. The landmarks that are distant from each other are also relatively distant in the place code map. Rather than pre-programming localisation algorithms as internal modules, our place code based SOMs architecture demonstrates that localisation can be learnt in a robust model based on external hints from the environment. This model was developed to learn landmarks in an environment, by having maps divided into clusters of neurons for different parts of the environment. It is considered to have a lot of potential for learning the localisation of the robot within an environment.

References

1. J. Ng, R. Hirata, N. Mundhenk, E. Pichon, A. Tsui, T. Ventrice, P. Williams, and L. Itti, "Towards visually-guided neuromorphic robots: Beobots," in *Proc. 9th Joint Symposium on Neural Computation (JSNC'02), Pasadena, California*, 2002.
2. D. Foster, R. Morris, and P. Dayan, "A model of hippocampally dependent navigation, using the temporal difference learning rule," *Hippocampus*, vol. 10, pp. 1–16, 2000.
3. A. Arleo, F. Smeraldi, S. Hug, and W. Gerstner, "Place cells and spatial navigation based on 2d visual feature extraction, path integration, and reinforcement learning," in *NIPS*, pp. 89–95, 2000.
4. A. D. Redish, *Beyond Cognitive Map from Place Cells to Episodic Memory*. London: The MIT Press, 1999.
5. A. D. Redish, *Beyond Cognitive Map*. PhD thesis, Carnegie Mellon University, Pittsburgh PA, August 1997.
6. E. T. Rolls, S. M. Stringer, and T. P. Trappenberg, "A unified model of spatial and episodic memory," *Proceedings of the Royal Society B*, vol. 269, pp. 1087–1093, 2002.

7. T. Kohonen, *Self-organizing Maps*. Springer Series in Information Sciences, Berlin, Germany: Springer-Verlag, 3rd ed ed., 2001.
8. E. Rolls and G. Deco, *Computational Neuroscience of Vision*. New York: Oxford University Press, 2002.
9. S. Wermter, J. Austin, and D. Willshaw, *Emergent Computational Neural Architectures based on Neuroscience*. Heidelberg: Springer, 2001.
10. M. Haritoppulos, H. Yin, and N. M. Allinson, "Image denoising using self-organisising map-based non-linear independent component analysis," *Neural Networks: Special Issue, New Developments in Self Organising Maps*, vol. 15, pp. 1085–1098, October - November 2002.
11. R. R. Murphy, *Introduction to AI Robotics*. London, England: The MIT Press, 2000.
12. R. A. Brooks, *Cambrian Intelligence: The Early History of the New AI*. Cambridge, Massachusetts: The MIT Press, 1999.
13. N. E. Sharkey, "The new wave in robot learning," *Robotics and Automation system*, vol. 22, pp. 179–186, 1997.
14. S. Wermter and R. Sun, eds., *Hybrid Neural Systems*. Berlin: Springer, 2000.
15. S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self Organizing Machines*. The MIT Press, 2000.
16. R. Pfeifer and C. Scheier, *Understanding Intelligence*. The MIT Press, 1999.
17. V. Braitenberg, *Vehicles*. Cambridge - Massachusetts: MIT Press (A Bradford Book), 1984.
18. H. Maaref and C. Barret, "Sensor based navigation of mobile robot in an indoor environment," *Robotics and Automation Systems*, vol. 38, pp. 1–18, 2002.
19. U. Gerecke, N. E. Sharkey, and A. J. Sharkey, "Common evidence vectors for reliable localization with som ensembles," in *Proceedings of Engineering Application of Neural Networks*, EANN-2001, 2001.
20. C. Owen and U. Nehmzow, "Landmark-based navigation for a mobile robot," in *Simulation of Adaptive Behaviour*, (Zurich), The MIT Press, 1998.
21. H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *In. proc. IEEE International Connference on Robotics and Automation*, pp. 116–121, 1985.
22. W. Burgard, A. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, pp. 3–55, 1999.
23. K. Chokshi, S. Wermter, and C. Weber, "Learning localisation based on landmarks using self organisation," in *ICANN 03*, 2003.
24. U. Nehmzow, *Mobile Robotics: A Practical Introduction*. London: Springer Verlag, 2000.
25. S. Marsland, *Online Nolvelty Detection Through Self Organisation, With Application to Inspection Robotics*. PhD thesis, University of Manchester, Manchester UK, December 2001.
26. K. R. Castleman, *Digital Image Processing*. New Jersey: Prentice Hall, 1996.
27. L. G. Shapiro and G. C. Stockman, *Computer Vision*. New Jersey, USA: Prentice Hall, 2001.
28. M. Seul, L. O'Gorman, and M. J. Sammon, *Practicle Algorithms for Image Analysis: Decribtion, Examples, and Code*. Cambridge, UK: Cambridge University Press, 2000.