Robotics: Environmental Awareness Through Cognitive Sensor Fusion

T.P. Schmidt

March 2010

Master Thesis

Artificial Intelligence
Dept. of Artificial Intelligence
University of Groningen, The Netherlands

Internal supervisor:
Dr. M.A. Wiering (Artificial Intelligence, University of Groningen)

External supervisor:
Ir. A.C. van Rossum (Almende B.V., Rotterdam)

university of groningen / faculty of mathematics and natural sciences / artificial intelligence

Abstract

In order for autonomous mobile robots to survive in the real world they have to be aware of the environment. The self-assembling micro robots developed in the European project Replicator are destined for such a task. By using several modalities, these robots must be able to detect and recognize interesting objects in the environment. This thesis presents a biologically inspired cognitive sensor fusion architecture to create environmental awareness in these micro robots. This architecture consists of a bi-modal attention module and multi-modal sensor fusion. A state of the art visual saliency detection system has been optimized and combined with biologically based sensor fusion methods to obtain a visual-acoustic attention module. For multi-modal sensor fusion a new type of ARTMAP (self-organizing associative memory) called the Multi-directional ARTMAP (MdARTMAP) has been designed. With this MdARTMAP a module for unsupervised on-going learning of sound was developed by clustering states of an echo state network, which processes cochlear filtered audio. Also unsupervised visual object recognition was obtained with this MdARTMAP by clustering and associating salient SIFT keypoint descriptors. Based on these modules a multi-modal sensor fusion system was created by hierarchically associating the MdARTMAPs of the different modalities. Experiments conducted in a 3D simulator showed that a simulated robot was able to successfully perform a variety of search tasks with the cognitive sensor fusion architecture.

Contents

Chapter 1

Introduction

Cognitive sensor fusion is one of the mechanisms used in the European FP7 project: Replicator [17] to obtain environmental awareness in micro-robots. The Replicator project focuses on the development of mobile multi-robot organisms, which consist of a super-large-scale swarm of small autonomous micro-robots capable of self-assembling into large artificial organisms. Due to the heterogeneity of the elementary robots and their ability to communicate and share resources, they can achieve great synergetic capabilities. The goal of the Replicator project is to develop novel principles underlying these robotic organisms, such as self-learning, self-configuration and self-adjustment. By using a bio-inspired evolutionary approach the robots will evolve their own cognitive control structures so that they can work autonomously in uncertain environments without any human supervision. Eventually these robots will be used to build autonomous sensor networks, capable of self-spreading and self-maintaining in for example hazardous environments. For example in the event of an earthquake, the micro-robots could dissemble to enter a collapsed building and then reassemble once inside to crawl over obstacles and search autonomously for victims.

To obtain environmental awareness in the micro robots, cognitive sensor fusion will be used. Cognitive sensor fusion is a bio-inspired process, the equivalent biological system is responsible for our internal representation of the environment. The self-organization which takes place in biological sensor fusion is the research point of interest. This master project focuses on the development of cognitive sensor fusion through self-organization. With this project an answer to the following research question is to be found:

How can biologically inspired sensor fusion be used in an embodied self-organizing micro-system to increase environmental awareness?

Implementing bio-inspired cognitive sensor fusion on an embodied system can give insights in how to benefit from self-organization in a system which interacts with a dynamical environment. This project will also give new insights in how to develop a multi-modal saliency detection system on a mobile robot.

The cognitive sensor fusion system will be tested in a 3D simulator where visual-acoustic information is fused for object detection and recognition. In the experiment the robot must be able to distinguish other robots from other objects based on low quality sound and camera images. By using cognitive sensor fusion with different modalities the robot must be able to detect objects earlier and recognize objects better than without sensor fusion. If the robot is searching for a particular object, then if it hears a sound, it has to know what object, in the sense of a visual representation, is associated with it and the other way around. So if the robot is shown a picture of an object which he has to search for, then the robot should be able to find that object only based on the expected sound that it makes.

The remainder of this thesis is structured as follows; In the next chapter, the theoretical background for the parts of the cognitive sensor fusion architecture is given. In chapter 3, a description of the methodology and implementation of these parts is given. It starts with the attention module followed by self-organizing associative memory and eventually an implementation used for object recognition is described. In chapter 4, the implemented modules used for the experiments and the experiment setup are described. The results of the experiments are presented in chapter 5. In chapter 6, a summary and explanation of the results are given followed by the conclusion and recommendations made for future work.

Chapter 2

Theoretical Background

2.1 Biological Sensory Integration

Cognitive sensor fusion is a biologically inspired approach to integrate multiple sensor data. To find an architecture suited for mobile robots, taking a look at how biology has implemented such a mechanism is needed. Studies in the literature of multi modal integration (MMI) have been using different species to find out more about the underlying architectures in the brain. In mammals, integration has been found in the superior colliculus. Although much remains speculative, some general processes can be formulated. A better understood integration process is that of the insect brain. Neurobiological research on the insect's nervous system has identified essential elements like the mushroom bodies for multi modal integration. A description of these two biological ”architectures” will be given below.

2.1.1 Multi Modal Sensory Integration in Vertebrates

When looking for multi modal integration in vertebrates, the superior colliculus (SC) is found to be the main brain area involved in this integration. Neurons in the SC are responsive to audio-, visual-, somatosensory-, and multi sensory stimuli. In the barn owl, visual and auditory pathways are believed to be integrated in the deeper layer of the SC [25]. The deeper layer is also involved in orientation-initiated behaviour such as eye saccades. Most of the neurons in the SC are bimodal (Audio-Visual). Visual stimuli from the retina is projected (2D image map) to the superficial SC, in a way that a certain retina location corresponds to a neuron in the SC (retinotopic). The auditory stimuli to the SC comes from the external nucleus of the inferior colliculus (ICx). The auditory input shows frequency specific neural response in the central nucleus (ICc), and neural response to specific positions in space in the ICx. The neurons in both these areas are sensitive to interaural time differences (ITD). Frequency neurons (ICc) with the same ITD are mapped to a single ICx neuron. The auditory map formed in the ICx shows a map shift due to change (error) in the visual map, in contrary to ICc. An inhibitory network in the SC modulates the visual signal to allow adaptation only when auditory and visual maps are misaligned (Map Adaptation Cue: MAC) (figure 2.1).

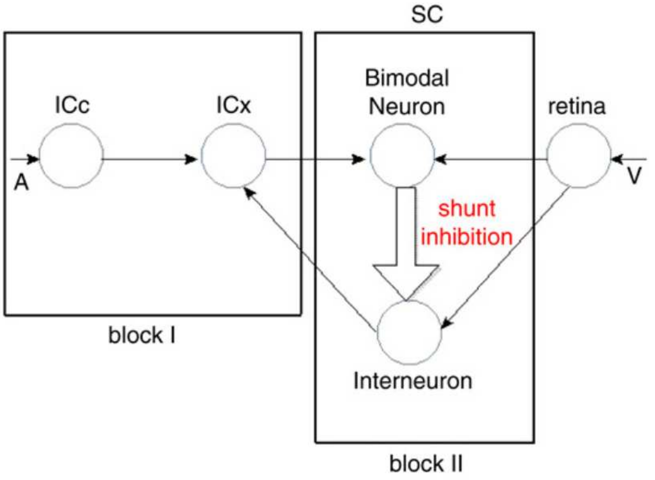


Figure 2.1: The schematic audio visual signal processing pathway. The circles represent neurons, the filled arrows excitatory connections and the open arrow represents the inhibitory connection between the SC's bimodal neuron and the interneuron. The salient auditory input is denoted by (A) and the spatial visual salient input by (V). If the inputs from A and V correspond, indicating aligned A and V stimuli, the connections to the bimodal neuron are strengthened and the interneuron is inhibited strongly. In contrast, when the A and V signals do not match, the connection strength is decreased and the inhibition of the interneuron reduced. (Taken from [25])

In a proposed model by [25], the MAC (which is adjusted by Spike Time Dependent Plasticity) resides in an ”interneuron” which is responsible for sending the visual signal to the ICx. The sensory pathway can be divided into two sections (figure 2.1). Block I with ICc connected to ICx, and block II with the detector of any shift between visual and auditory cues and the controller of the ICc/ICx mapping (interneuron). The neuron response in the visual or audio layer have a center surround profile. The firing rate of the neurons with the difference in spike timing encodes the location of objects in the environment.

2.1.2 Multi Modal Sensory Integration in Insects

Wessnitzer and Webb [56] [55] have done several studies on the nervous system/brain of insects. In [55] they have given a review about what is known about two specific higher areas in the insect (Dorsophila) brain, the mushroom bodies and the central complex. The mushroom bodies in most insects have similar and characteristic neuroarchitectures: a tightly-packed parallel organization of thousands of neurons, called Kenyon cells. The mushroom bodies are divided in: the calyces, the penduncului and the lobes. In most insect species the mushroom bodies receive significant olfactory input, and some also have connections from the optic lobes to the mushroom bodies. The neurons in the output regions of the mushroom bodies can be classified as: sensory, movement-related or sensorimotor. A large majority responds to multiple sensor stimuli and therefore seems to be involved with sensory integration. The mushroom bodies are not the only sensorimotor pathways, there exists a parallel pathway from sensors to the pre-motor unit (figure 2.2).

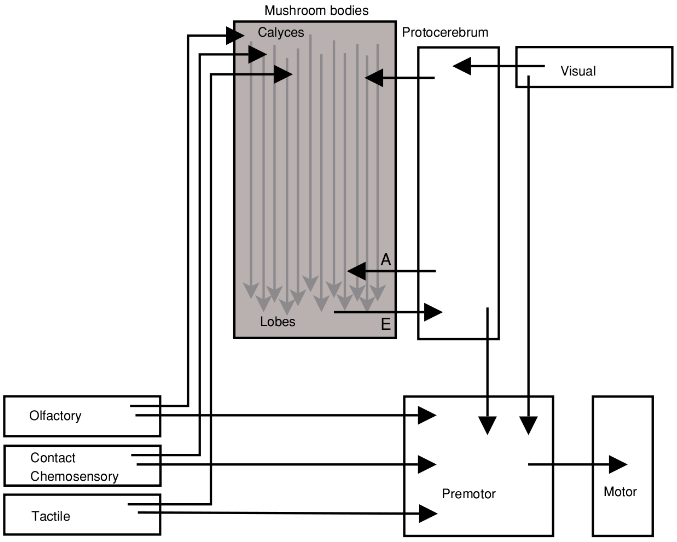


Figure 2.2: Multi-modal processing pathways of the Dorsophila nervous system. The mushroom bodies play an important role in the processing and integration of multi-modal information. Evidence suggests that mushroom bodies do not form the only sensorimotor pathway for any modality, sensory areas in the brain have direct connections to premotor areas.(Taken from [55])

A role of the mushroom bodies is pattern recognition. The Kenyon cells perform specific processing functions on the primary sensory input to mushroom bodies. The dendrites from the Kenyon cells to the lobes impose different filter characteristics. The Kenyon cells also act as delay lines which could provide a mechanism for recognizing temporal patterns in the input. The spatio-temporal properties of the Kenyon cells can act as a saliency detector using the correlations in the input spike trains. Kenyon cells receive direct sensory input from a modal lobe and indirect via the lateral horn arriving shortly after. The integration time for the Kenyon cells is limited to short time windows, making them sensitive to precise temporal correlations.

A second role is the integration of sensory and motor signals. Extrinsic neuron responses have been reported which were selective to directions of turning behaviour. A distinction in neural activity has also been reported for self-stimulation and externally imposed stimuli. It is thought that an indirect pathway involving the mushroom bodies converges with more direct pathways for hierarchical integration and modulation of behaviour.

Mushroom bodies also play an important role in associative learning and memory. Kenyon cells show structural plasticity by growing new connections during the insect's life time. The Kenyon cells seem to be a major site for the expression of 'learning' genes. Hebbian processes underlying associative learning could reside in the Kenyon cell dendrites.

A sensor fusion method based on the structure of the mushroom bodies should be able to perform: multi-modal saliency detection, pattern recognition using associative memory, and integration of sensory and motor signals.

2.2 Sensor Fusion

With respect to the sensor domain, sensor fusion can be divided in: fusion of information from different sensor modalities that have a similar representation (single domain), and fusion of complementary data from different sensor modalities that have a different representation (multi-domain). The first one can be used for extracting useful information out of a single sensory information domain, whereas the second creates a coupling between different sensory information domains. For example when combining vision with auditory localization cues (spatial domain), the position of a certain object can be determined more accurately. Fusing information from different domains can be done for example by fusing an audio pattern and an image of an object so that there is a visual and an auditory representation of the object. Hearing an object will then create a mental image due to association.

Single and multi domain sensor fusion are needed to enlarge environmental awareness and the complexity of an autonomous system. Single domain sensor fusion can be seen as an attentional mechanism, while multi domain sensor fusion can be seen as an associative process. The in the previous section described biological sensory integration systems are examples of single and multi domain sensor fusion. Some examples of architectures that can be used to create these types of sensor fusion systems will be described.

2.2.1 Self-Organizing Maps (SOM)

When thinking about associative memory, self-organization comes to mind. The link between multi-modal integration (MMI) and self-organization (SO) seems to be made because of the associative processes in MMI. In the pre MMI stage associative network structures can be found in for instance the retinotopical and tonotopical organization [31]. In *"Multi-modal Feed-forward Self-organizing maps"* by Paplinski and Gustafsson [42] a method is proposed to build a multi-modal classification system with hierarchically constructed SOMs. The construction is based on the modular hierarchical structure of the mammalian neocortex [31]. The first layer of the proposed structure is formed by three feed-forward SOMs, each for a modality, and these maps are connected to a single multi-modal SOM. This structure incorporates both types of fusion: in the feed forward SOMs uni-modal multi-sensory information is merged, and in the last map multi-modal information.

In [43] this structure was used to build a Multi-modal Self-Organizing Network (MuSON), consisting of several Kohonen maps. With the use of a feedback connection from the multi-modal SOM, perception of corrupted stimuli in the uni-modal SOM was enhanced (Top-Down). This feedback loop can be compared with the recalibration after integration misalignment of bi-modal information in the superior colliculus [25]. In [43] it was successfully implemented to enhance the perception of corrupted phonemes using a bimodal map which integrates phonemes and letters. The advantage of the MuSON in comparison with a single SOM is the parallel uni-modal processing converging into a multi-modal map. More complex stimuli can therefore be processed without a growing map size [43]. Bimodal integration and classification of phonemes and letters is not a complex task in comparison with unsupervised recognition and fusion of noisy auditory and visual data. This makes it rather uncertain whether this method is suitable for on-going learning in a dynamic and complex environment.

2.2.2 Reservoir Computing

Constructing a random recurrent topology with a trained single readout layer for pattern recognition is called reservoir computing. The idea behind it is that through pre-processing the input is transformed to the feature space which has a higher dimension and is possibly linearly separable. Echo state networks (ESNs) [29] and liquid state machines (LSMs) [39] are the best performing types of reservoirs. In *"An overview of reservoir computing: theory, applications and implementations"* by Schrauwen [46] a summary of the capabilities of these methods is given. ESNs and LSMs differ on the type of node they use, but which type of node is best suited for what task is not known. Evidence in [52] shows that spiking neurons might outperform analogue neurons for certain tasks, like speech recognition. There also seems to be a monotonic increase of the memory capacity as a function of the reservoir size [52].

In *"Dynamic liquid association: complex learning without implausible guidance"* by Morse and Aktius [41] a system is constructed where a liquid state machine is combined with an associative network for pattern recognition. The relations to the mushroom bodies are: saliency detection using a spatio-temporal mechanism (the micro columns as reservoir), and associating different sensor modalities (sensor-motor) using an associative network.

Morse and Aktius did several experiments with a mobile robot with infra-red and collision detection sensors. It managed to learn obstacle avoidance and showed complex behaviour. They also conducted a classical conditioning experiment where they used a camera with 10 x 10 x 3 pixel values but abandoned the LSM for reasons of computational speed on a SEER-1 robot. Instead they used an ESN microcircuit, which is comparable with an LSM but has a randomly generated continuous time or discrete time recurrent neural network with analogue neurons instead of spiking neurons. This raises questions about the usability of LSMs for computationally poor robots that use even more sensors with additional microcircuits.

In *"Training networks of biological realistic spiking neurons for real-time robot control"* by Burgersteiner [4], a real-time off-line LSM with one microcircuit of 54 leaking integrate-and-fire neurons was used to create two reactive Braitenberg controllers (linear and non-linear) on a Khepera robot. Using 6 IR sensors it was able to learn the desired behaviour. For training they stored sensor input and motor output during a test run of the robot with a preprogrammed Braitenberg architecture. They used this off-line on an LSM, with the desired motor response as target input for supervised linear regression learning. Although the used setup is not desirable and is quit complex, they were able to show that using one micro column was enough to imitate the linear and non-linear Braitenberg behaviour on a miniature robot.

2.3 Attention

Working with computationally poor systems requires the need of efficient processing of information. When it comes to sensor information processing, visual and acoustic data processing are the most demanding. Without selective attention sensory systems would be either overwhelmed or blind to important sensory information. Therefore implementing attention mechanisms derived from biology can be helpful.

2.3.1 Visual Attention

The visual system is not capable of fully processing all of the visual information that arrives at the eye. In order to get around this limitation, a mechanism that selects regions of interest for additional processing is used. This selection is done bottom-up, using saliency information, and top-down, using cueing.

The processing of visual information starts at the retina. The neurons in the retina have a center surround organization of their receptive fields. The shapes of these receptive fields are among others modelled by the difference of Gaussian (DoG) [45]. This function captures the ”Mexican hat” shape of the retina ganglion cell’s receptive field. These cells emphasize boundaries and edges (figure 2.3).

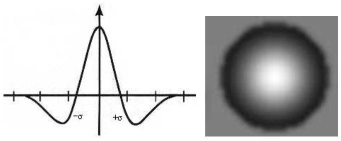


Figure 2.3: The difference of Gaussian, used to model retina cells. Left the Difference of a Gaussian is shown as a graph, and right as an intensity image.

Further up the visual processing pathway is the visual cortex area V1. Here are cells that are orientation-selective. These cells can be modelled by a 2D Gabor function (figure 2.4).

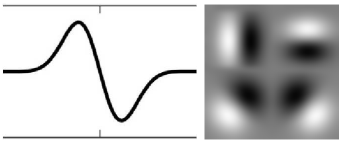


Figure 2.4: A steerable Gabor is used to model orientation selective V1 cells. Left an example of a steerable Gabor is shown as a graph. On the right four different steerable Gabor outputs are shown as an intensity image.

Itti and Koch’s implementation of Koch and Ullman’s saliency map is one of the best performing biologically plausible attention model [33] [28] [26]. Itti et al. [28] implemented bottom up saliency detection (figure 2.5) by modelling specific feature selective retina cells and cells further up the visual processing pathway. The retina cells use a center surround receptive field which is modelled in [28] by taking the DoG. They also model orientation selective cells using 2D Gabor filters. For each receptive field their is an inhibitory variant. For example if an on-center off-surround receptive field shows excitation on certain input, then the input will cause the opposite off-center on-surround receptive field to inhibit.

The sub-modalities that Itti et al. [28] use for creating a saliency map are intensity, color and orientation. For each of these sub-modalities a Gaussian scale pyramid is computed to obtain scale invariant features. For each of these image scales feature maps are created with a receptive field and its inhibitory counter part. For the intensity sub-modality on-center off-surround and off-center on-surround feature maps for different scales are computed based on the pixel intensity. For the color sub-modality feature maps are computed with center surround receptive fields using a color pixel value as center with its opponent color as surround. The color combinations used for this are red-green and blue-yellow. The feature maps for the orientation sub-modality were created using the 2D Gabor filters for the orientations 0, 45, 90, 135.

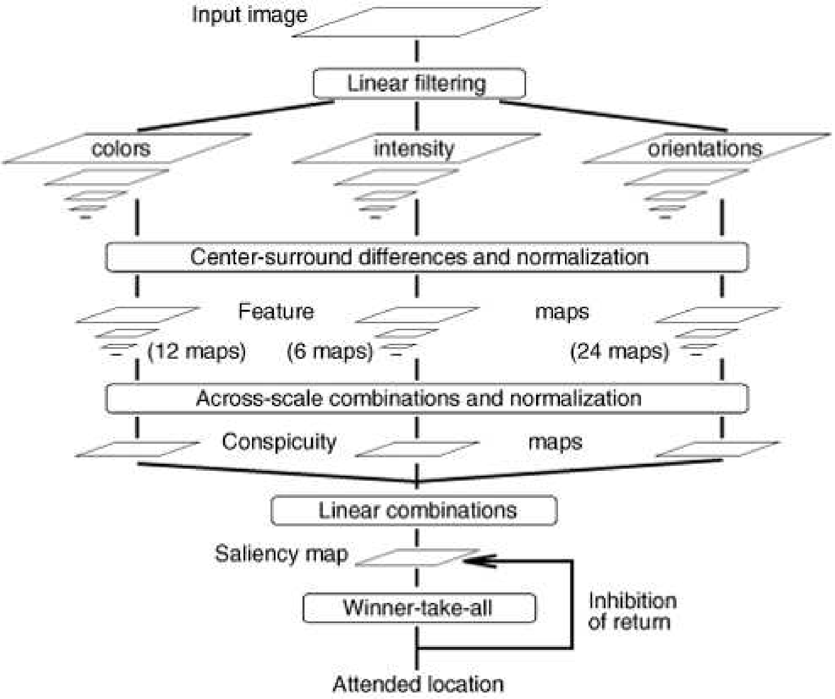


Figure 2.5: Saliency model Itti et al. [28]. This figure shows the processing pipeline of the saliency detection model. From top to bottom: an input image is filtered on color, intensity, and orientation using receptive fields on different scales of the input image. Using a weighting process (Center-surround differences and normalization) feature maps are created for different scales for each sub-modality (color, intensity, orientation). Through across-scale combinations and normalization conspicuity maps are created for the three sub-modalities. These three maps are subsequently combined into a saliency map. When modelling attentional focus with this model, the inhibition of return will cause the second most salient location to be attended. (Taken from [28])

To obtain a saliency map (figure 2.6) from all these features, a weighting process is executed in several stages to obtain the most salient features. In the first stage feature maps are weighted across the different receptive fields, in the second stage this is done across the scales, and in the final stage across the sub-modalities. By combining the feature maps obtained in the last stage (conspicuity maps) a saliency map is created.

Itti and Koch's model has been implemented in a real-time system called the Beobot (Neuromorphic Vision Toolkit; NVT). A real-time system which is based on their work is VOCUS [18]. VOCUS is used in several applications such as object recognition and visual localization [21] [20] [22]. Itti, Koch, and Ullman' s attention model is also used for applications that

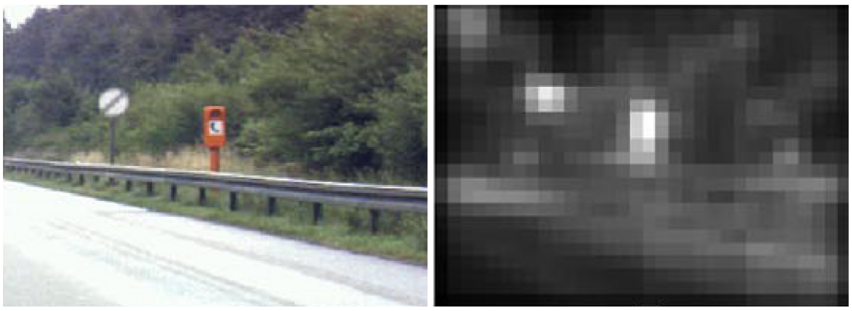


Figure 2.6: A saliency map computed with the visual attention system of Itti et al. [28] with the corresponding input image on the left. (Taken from [28])

are not used in real-time, such as text detection [34]. Chevallier et al. have implemented the model using a spiking neural network (SNN) [16].

Both the NVT and the SNN model need a lot of computational power. The Beobot is equipped with 4 PIII processors, and the SNN implementation has a good performance with 1 frame/sec (76x56 pixels) on a 2.8 Gig Core2Duo processor. The computational expensive part of the model is the feature calculation for the different scales (see section 3.1.1). In Frintrop's implementation [18] of Koch and Ullman's model [33], center-surround features are calculated using integral images (see section 3.1.1). With this optimization a comparable saliency detection performance can be obtained with 100 frames/sec (200x150 pixels) on a 2.8 Gig processor.

Spike-timing which seems to be an efficient and biologically plausible way to compute salient information [50], is computationally rather expensive for current computers. Therefore biologically inspired real-time visual attention systems seem to need algorithms from computer vision to create a system which is usable in real-time.

2.3.2 Auditory Attention

Just like visual information processing, audio processing is also influenced by attention. Mechanisms exist to bias attention towards salient events so that information rich data has a processing preference. In [32] Kayser et al. showed that visual saliency detection methods are suitable for allocating auditory saliency. To find salient information in temporal data, a transformation to a visual representation can be used to benefit from the more sophisticated visual saliency detection methods. In [32] they visualized an audio stream as an intensity image in a time-frequency representation. From this intensity image an auditory saliency map was computed using a visual saliency detection system based on work of Itti et al. [26]. The extraction of auditory salient features was based on three types of features: the sound intensity difference, the spectral contrast, and the temporal contrast. With these features they were able to predict which sound samples would be perceived as salient by humans and monkeys. Based on this Kayser et al. [32] concluded that saliency is determined either by implementing similar mechanisms in different uni-sensory pathways or by the same mechanism in multi-sensory areas. In any case, their results demonstrate that different primate sensory systems rely on common principles for extracting relevant sensory events.

Chapter 3

Methodology & Implementation

The developed cognitive sensor fusion architecture (figure 3.1) is broadly based on the earlier described audio-visual integration process found in the brain of vertebrates, and the multi-modal integration process in the well studied nervous system of the fruit fly (Drosophila Melanogaster). In this architecture environmental awareness is obtained through bi-modal attention, via audio-visual saliency detection and binaural localization; and through audio-visual object recognition via multi-modal associative memory (sensor-fusion).

In the cognitive sensor fusion architecture in figure 3.1 two types of sensor fusion are shown on the left and right. These are respectively multi-modal sensor-fusion using *Associative Memory*, and early stage sensor fusion used for *Bi-Modal Attention*. This architecture focuses on integrating visual and auditory information, but associating other sensory information is also possible.

The first step in early stage sensor fusion is saliency detection. *Visual saliency detection* is performed on the camera image (see section 3.1) and on the visual representation (cochleogram) of an audio stream (see section 3.2). Based on the saliency information from the camera image a spatial location is computed. The saliency information from the cochleogram is used to select the audio regions to compute the binaural cues from. The binaural cues and the visual salient location are used for *Bi-modal Attention* (see section 3.3). Based on the saliency information in both modalities, audio-visual object recognition is initiated. After audio pre-processing (see section 3.6) and image feature extraction (see section 3.7) both sensory data are fused using *Associative Memory* (see section 3.5).

In the next sections these modules will be described in more detail, starting with early stage sensor fusion: visual and bi-modal attention, followed by multi-modal sensor fusion: unsupervised visual and auditory object recognition and association.

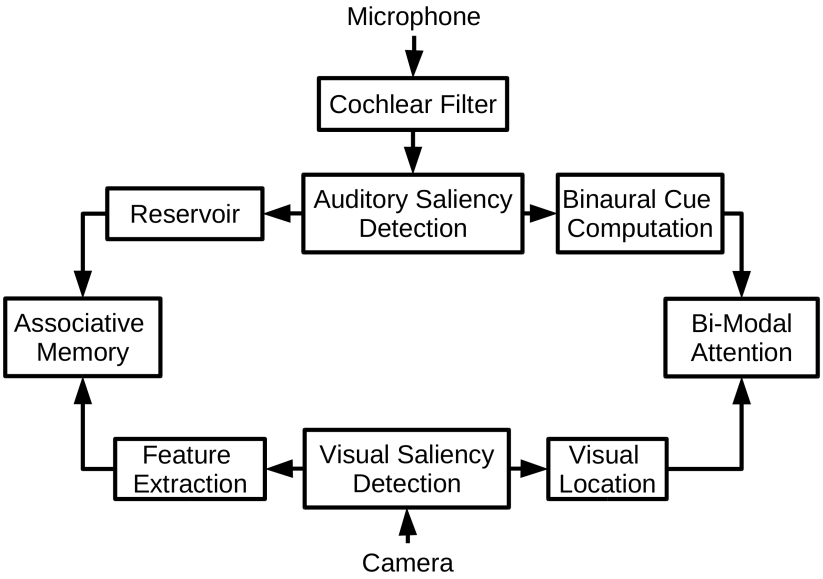


Figure 3.1: The cognitive sensor fusion architecture. In this abstract representation of the architecture, modules are visualized by blocks and information streams by arrows. *Bi-Modal Attention* pathway: The *Visual Saliency Detection* module receives a camera image and computes a saliency map. The *Visual Location* module returns the location of the most salient object. The *Cochlear Filter* filters the audio stream from the microphone. The *Auditory Saliency Detection* module computes a saliency map from a cochleogram, after which the *Binaural Cue Computation* module computes the binaural cues from the salient audio. The *Bi-Modal Attention* module integrates the binaural cues and the visual location. *Associative Memory* pathway: The *Feature Extraction* module computes image features from the salient image region. The *Reservoir* module transforms the cochlear filtered audio to feature space after which the audio and visual features are associated in the *Associative Memory* module.

3.1 Visual Saliency Detection

The visual saliency detection architecture that will be described in this section is derived from work of Itti et al. [28] and Frintrop et al. [19]. The proposed architecture is implemented in the 3D simulator Symbricator and will therefore be referred to as: Symbricator3D Image Saliency-based Computational Architecture (SISCA). Itti et al. [28] implemented bottom up saliency detection (figure 3.2) by modelling specific feature selective retina cells and cells further up the visual processing pathway. The retina cells use a center surround receptive field which is modelled in [28] by taking the difference of Gaussian (DoG). They also model orientation selective cells using 2D Gabor filters. The features that they use for creating a saliency map are intensity, color and orientation. For each of these features a Gaussian scale pyramid is computed to obtain scale invariant features using receptive fields.

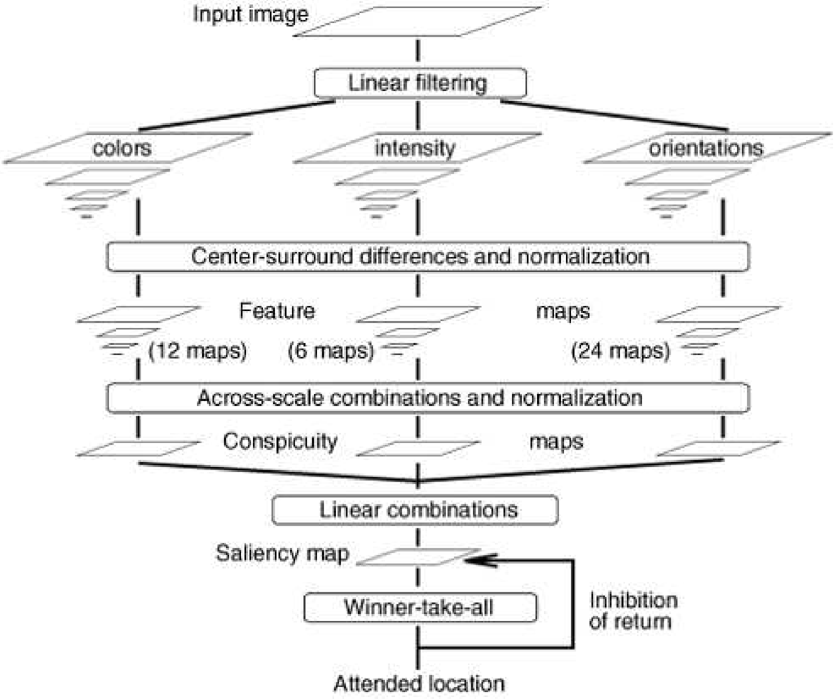


Figure 3.2: Saliency model Itti et al. (Taken from [28])

Frintrop et al. [19] created a modified version of Itti and Koch's model called VOCUS. The first version of VOCUS was aimed at creating a better performing system. Simplifications in Itti and Koch's model in comparison to the biological analogue were changed in VOCUS to obtain a biologically more plausible model and a better performance. The drawback of these changes was the high computational complexity of the system which made it not suitable for real-time usage. To obtain a real-time saliency detection system they changed one of the most computational expensive parts, the calculation of the center surround difference. Instead of using a Gaussian scale pyramid they used integral images and computed the center surround difference by taking the difference of mean (DoM) (figure 3.3).

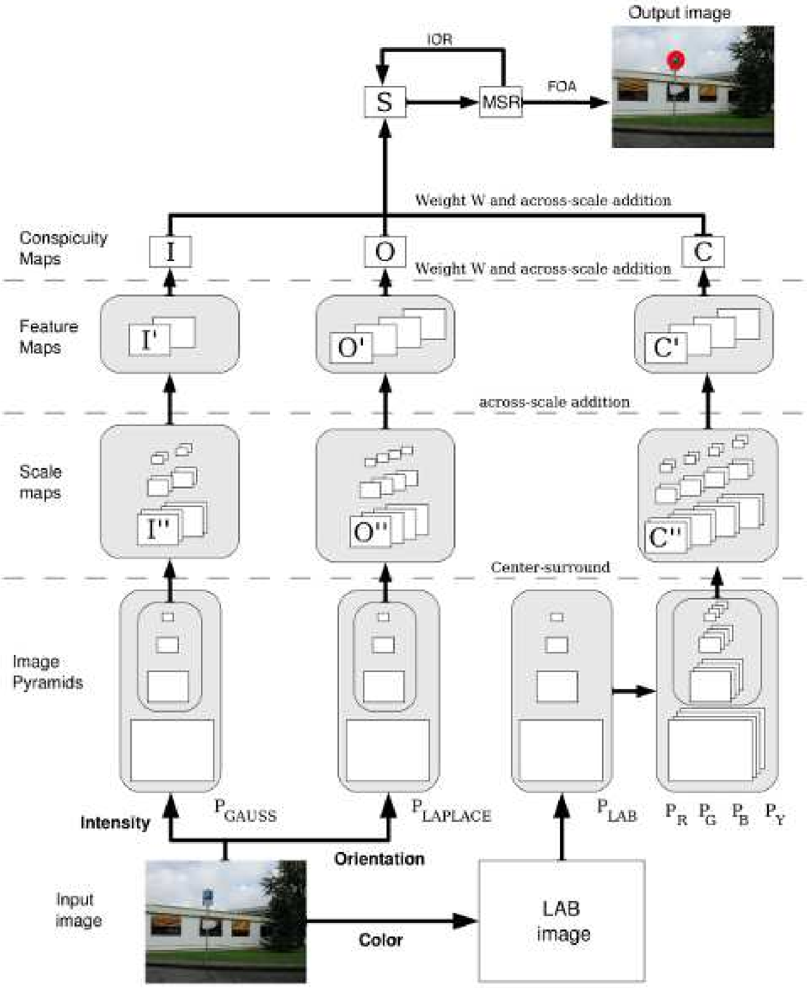


Figure 3.3: The visual attention system VOCUS. VOCUS is based on the saliency map computation of Itti et al. [28]) (figure 3.2). It has the same processing stages: linear filtering of the input image followed by the creation of image pyramids, scale maps, feature maps, conspicuity maps and the saliency map. The main difference in VOCUS is that the computation of the *Image Pyramids* for intensity and color is done with integral images. (Taken from [19])

Although the improved version of VOCUS has gained much processing speed there is still room for improvements. In order to preserve their original structure with scale pyramids they chose to use separate integral images for each scale instead of just one integral image. They also chose to keep the Gabor filter instead of an approximation for better performance.

SISCA (figure 3.4) is mostly based on VOCUS. It also uses integral images for faster center surround computations, but to increase computation speed the 2D Gabor filters are replaced by Haar-like features in combination with rotated integral images to compute the orientation feature maps. Other changes on different levels have been made for a better speed accuracy ratio. These will be discussed in the following sections.

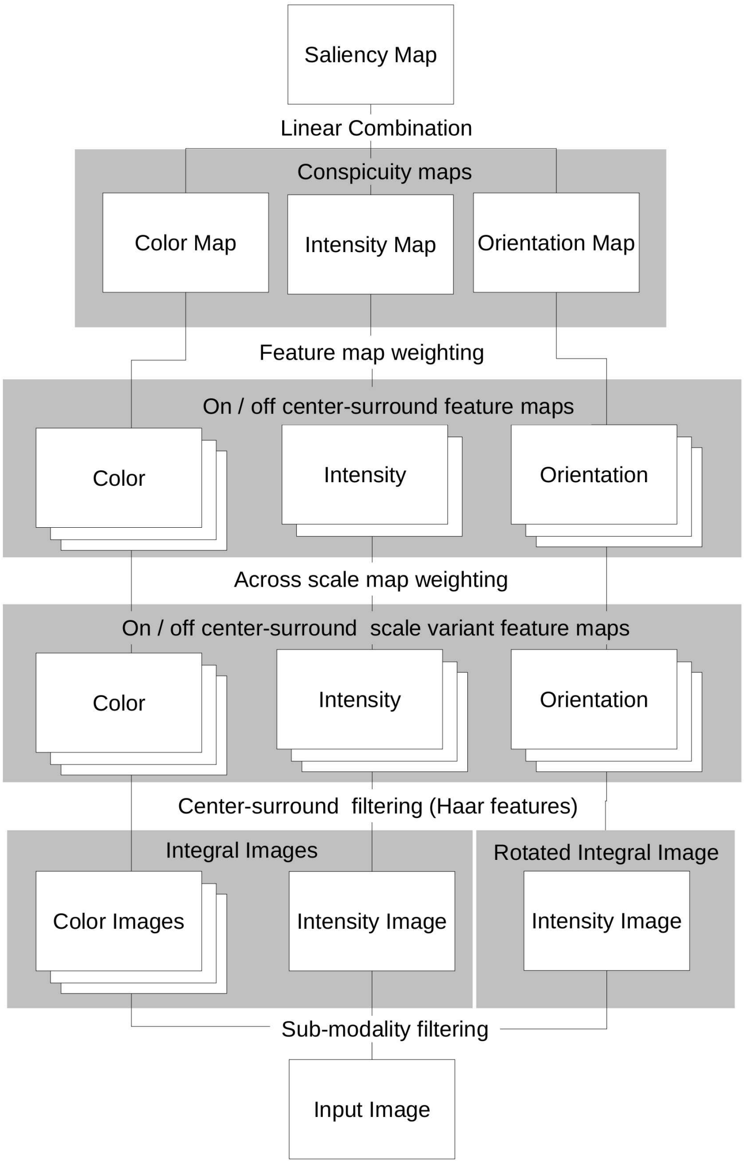


Figure 3.4: The implemented Symbricator3D Image Saliency-based Computational Architecture (SISCA). SISCA is mainly based the visual attention system VOCUS. The main differences between these systems are that in SISCA no image pyramids are computed to obtain the scale maps but instead integral images are used to compute the color and intensity features and a rotated integral image is used for to compute the orientation features. The different scales are obtained using different receptive fields sizes in the *Center-surround filtering using Haar-features*.

3.1.1 Scale Invariant Feature Extraction

The main difference between the visual attention system of Itti et al. [28], VOCUS [19] and the new proposed architecture SISCA is the computation of the scale invariant features. As described in section 2.3.1, features can be extracted using filters which are based on the receptive fields of retina cells and cells from the visual cortex area V1. Because the traditional calculation of these features with respectively a DoG filter and Gabor filters is computationally expensive, an approximation of these filters can be used. Haar-like features in combination with integral images [54] can be used to obtain such an approximation. In VOCUS only the DoG filter is approximated (figure 3.5). To decrease the computation time even further in SISCA, extended Haar-Like features with rotated integral images [36] are used to approximate the Gabor filters. In the next sections the different methods are elaborated on.

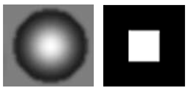


Figure 3.5: The center surround receptive field approximation of a retina cell. Left the DoG, and right the Haar-like equivalent.

Gaussian Scale Pyramids

In [28] Gaussian scale pyramids are used for scale invariant receptive field feature extraction (figure 3.6). It is a commonly used method in image processing, but it is computationally rather expensive. In VOCUS Gaussian pyramids are only used to compute scale invariant features. Different image scales are normally used so that the filter mask with which an image is convolved does not have to change. The convolution of an image with a larger mask is rather time consuming, O(nm) where n is the number of pixels in the image and m the number of entries in the filter mask.

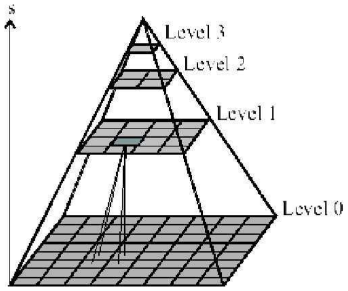


Figure 3.6: Gaussian scale pyramid. The layers of the image pyramid are obtained by subsampling or downsampling the previous layer (typically by taking every 2nd pixel), starting with the original image on level 0. (Taken from [18])

When a Gaussian pyramid is used, several processing steps have to be taken. First the input image needs to be scaled down, which can be done by sub-sampling. Sub-sampling can lead to aliasing and to overcome this problem the spatial frequencies of the image which are above the sampling frequency must be removed. This can be done by smoothing the image with a Gaussian filter before sub-sampling it. When the receptive field filter is applied the filtered image needs to be scaled up/back. In [28] they used 9 spatial scales and all filtered maps are resized to scale 4. In VOCUS they used 4 scales, 2 receptive field sizes, and all maps are resized to scale 2. When scaling up some sort of interpolation needs to be used for anti-aliasing. In the first version of VOCUS nearest neighbor interpolation was used, and in the later version bilinear interpolation, a more accurate but also more expensive method.

Integral Images

Computing scale invariant receptive field features with integral images is faster because the computation of the average value of a region only needs a few lookups and additions (figure 3.7), it is independent of the filter size, and creating an integral image requires only one scan over the input image.

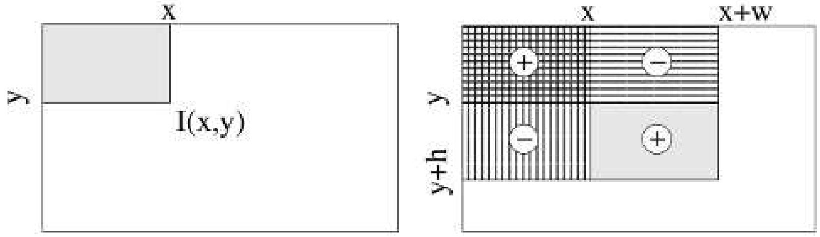


Figure 3.7: Integral image. Left: the value of pixel I(x,y) is the summation of the pixels in the grey area. Right: the computation of the shaded area based on four operations. (Taken from [19])

By using Haar-like features in combination with integral images, a fast and good approximation of the DoG and first order Gaussian filters can be obtained (figure 3.8).

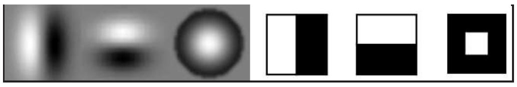


Figure 3.8: Receptive fields. Left: from left to right: 0 and 90 degrees first order Gaussian steerable filters (Gabor) and a 2D DoG. Right: the analog Haar-like filters.

3.1.2 Rotated Integral Images

When using integral images only simple rectangle Haar-like features can be created. In order to approximate second order Gaussian filters (see section 2.3.1) with Haar-like features, Rotated Integral Images (RII) (figure 3.9) can be used. The RII can be created using two scans over the input image. With a RII, 45 and 135 degree second order Gaussian filters can be computed (figure 3.10). These are called extended Haar-like features. With all these Haar-like features the three feature maps can be created.

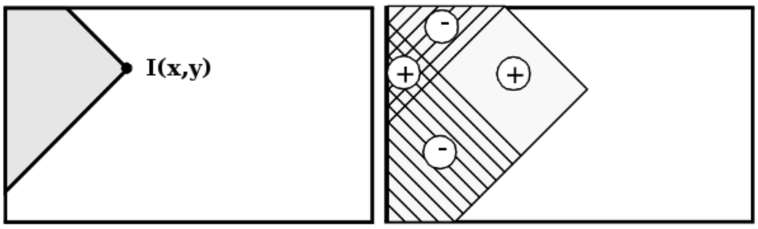


Figure 3.9: Rotated Integral image. Left: the value of pixel I(x,y) is the summation of the pixels in the Grey area. Right: the computation of the shaded area based on four operations. (Taken from [36])

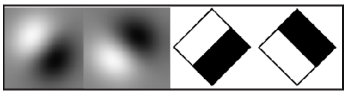


Figure 3.10: Receptive fields. Left: a 45 and 135 degree Gabor filter. Right the equivalent extended Haar-like features.

3.1.3 Receptive Fields (On-center Off-center)

The retina consists of cells which have an on-center off-surround or off-center on-surround receptive field. In [28] these two types of receptive fields are combined by taking the absolute value of the difference between center and surround. A problem with this approach, which is also addressed in [19], is that this will lead to a wrong pop-out when the difference with the background is the same for on-center and off-center. Therefore the computation of the on-center off-center receptive field in SISCA is done separately, and the map with the most information is promoted which leads to the right pop-out (figure 3.11).

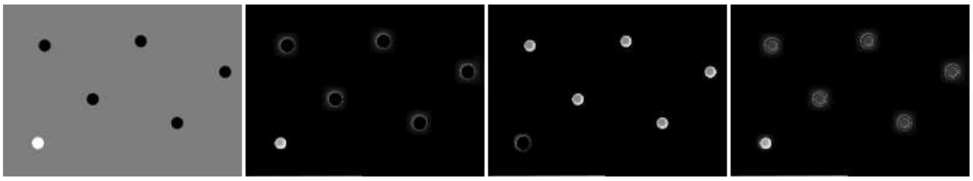


Figure 3.11: Saliency pop-out using separate on-center off-center computations with SISCA. (a) the input image (b) on-center off-surround intensity difference (c) off-center on-surround intensity difference (d) intensity feature map

3.1.4 Receptive Fields (Scales)

In order to obtain scale invariant features the Gaussian pyramid is replaced by different receptive field sizes. When using the Gaussian pyramid each scale reduction reduces the image dimensions from $(n * n)$ to $(\frac{n}{2} * \frac{n}{2})$, this is more or less equivalent with increasing the receptive field size by 2. Applying a larger receptive field size does not change computation time. It is faster than scaling down an image to find scale invariant features, because no anti-aliasing has to be applied. Another positive aspect of using the original size is that the output image has far more details (figure 3.12). This gives the possibility to use a lower resolution for the original image.

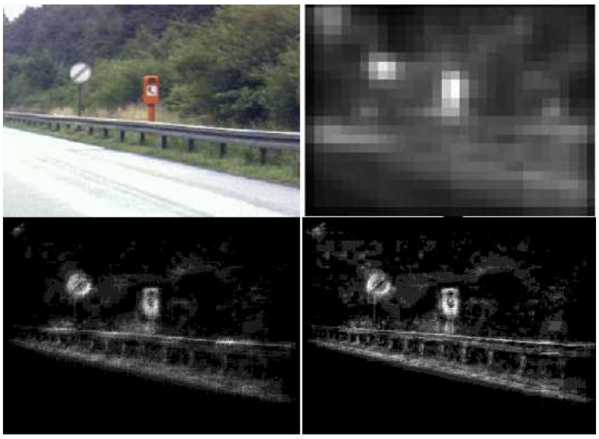


Figure 3.12: Saliency maps, white is more salient (normalized for printing). Top left: input image. Top right: Itti et al. [28] saliency map. Bottom left: SISCA saliency map using 4 scales, bilinear interpolation to scale 0 and 3 receptive field sizes: 2, 4 and 8 (without distribution as weight). Bottom right: SISCA saliency map using original scale and 9 receptive field sizes (without distribution measure as weight).

3.1.5 Feature Maps

For each sub-modality and receptive field a feature map will be created. SISCA uses three sub-modalities: intensity, color and orientation, and between 8-12 receptive field sizes. The intensity feature map set consists of feature maps with on-center off-surround and off-center on-surround receptive fields. The color map set is created using a system known in the cortex as ”color double-opponent”. In the center of the receptive fields, neurons are excited by one color and inhibited by another. This relation exists for: red/green, green/red, blue/yellow, yellow/blue. As in [28], these colors are broadly-tuned: red = r - (g + b) / 2, green =g - (r + b) / 2, blue = b - (r + g) / 2, and yellow = (r + g) / 2 - |r - g| / 2 - b. The orientation map set consists of 4 different orientation maps: 0, 45, 90, and 135 degrees, which are created using the corresponding Haar-like edge filters.

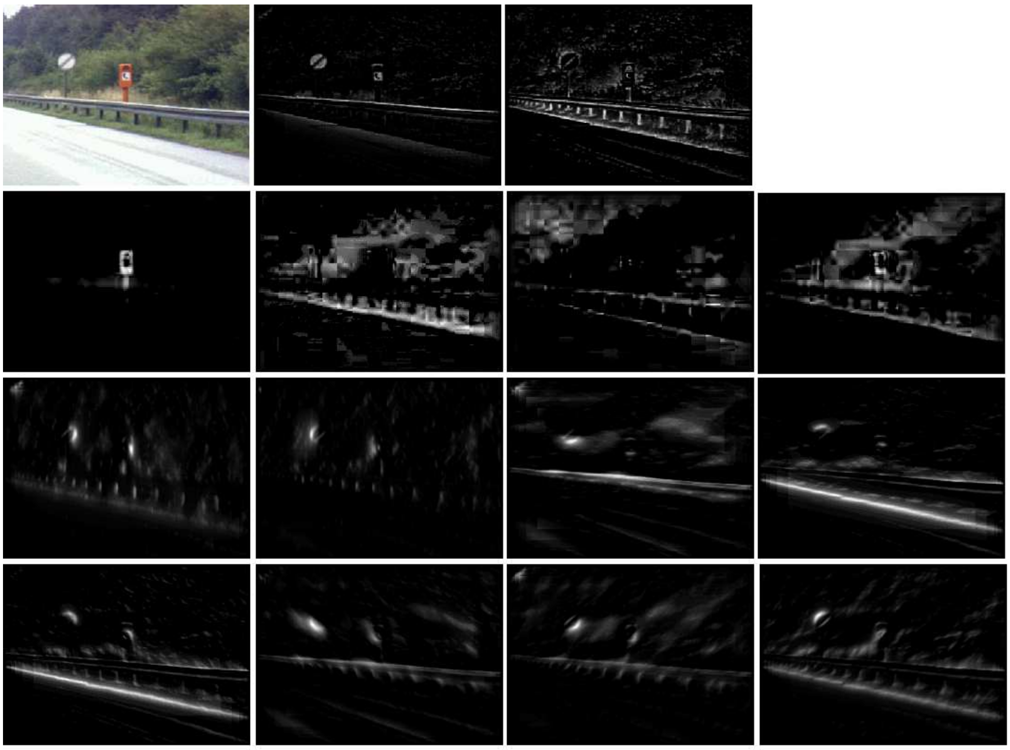


Figure 3.13: Feature maps created with SISCA (intermediate normalization). First row: the input image and the on-center and off-center intensity maps. Second row: the color maps, red/green, green/red, blue/yellow, yellow/blue. The third and fourth row the on-center and off-center maps for orientations: 0, 90, 45, and 135 degrees.

3.1.6 Fusing Receptive Field Specific Feature Maps

A feature map set with different receptive field sizes needs to be fused into one feature map (figure 3.13). Because there are a lot of feature maps, and some maps have less information than others, merging the maps can cause information to get masked (curse of dimensionality). Therefore the maps first need to be weighted to promote information rich maps and suppress maps that contain nothing unique (figure 3.11). After weighting the maps they are merged using point-to-point pixel addition.

Promoting information rich maps is an important aspect of the saliency detection system. Determining which map has the most information is not a trivial job. In [28], Itti et al. propose a map normalization operator. This operator works as follows:

- normalize the values in the map to a fixed range [0..M], in order to eliminate modality-dependent amplitude differences;
- find the location of the map's global maximum M and compute the average mu, of all its other local maxima;
- globally multiply the map by $(M - mu)^2$

One of the problems with this method was already pointed out in [27]. Taking the difference of the global and local maxima only works when there is just one strong peak. With two strong peaks the difference becomes zero which will result in suppressing the map. To overcome this problem they used a more complex iterative process, by local competition between neighbouring salient locations.

In the VOCUS system they used a more simple approach:

- Determine the global maximum M.
- Count the number of local maxima N above a predefined threshold from M.
- Divide each pixel by the square root of N.

The threshold was determined empirically and was set to 50% of the global maximum.

A reason given in [19] not to normalize the maps to a fixed range but only weigh them, is that normalizing maps to a fixed range removes important information about the magnitude of the maps. They only apply normalization to create the conspicuity maps, but not to a fixed range. Their motivation is that normalization is needed to make them comparable. Why this does not remove important information about the magnitude of the map is not mentioned.

3.1.7 Suppression, Promotion and Normalization

One of the main differences that can be seen when comparing both map weighting approaches is the promotion and suppression of maps. In [28] and [27] maps with more information are promoted more than maps with less information, while the information rich maps in VOCUS are suppressed less than maps with less information. This in combination with or without normalization gives remarkably different results when implemented in SISCA (figure 3.14 and 3.15). When considering maps with a lot of noise and not much information, suppression will wipe these maps out at an early stage by reducing the pixel values to 0 (due to the use of integer values) before creating a feature or conspicuity map. While promotion will let maps with only noise and not much information exist. Fusing these maps in the end by taking the sum or average will still give rise to the noise. This approach also leads to saliency maps where there is always a salient region even when there is nothing salient in the scene. Applying suppression will yield a totally black saliency map when there is nothing salient in the scene.

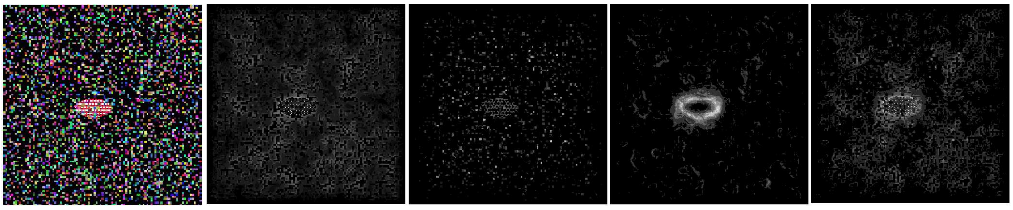


Figure 3.14: SISCA: Effect of noise on map weighting and normalization. From left to right: the input image, the intensity map, the color map, the orientation map and the saliency map.

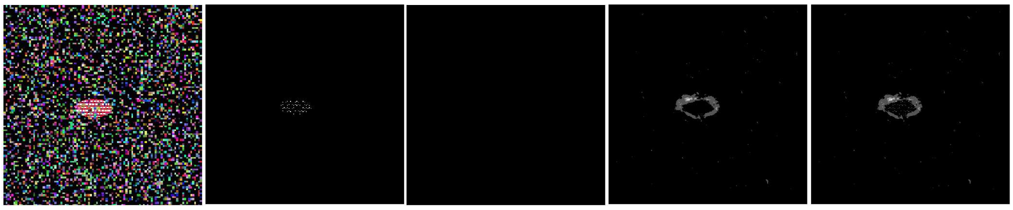


Figure 3.15: SISCA: Effect of only applying normalization to the feature maps when creating conspicuity maps. From left to right: the input image, the intensity map, the color map, the orientation map and the saliency map. The effect of only normalizing the maps when creating conspicuity maps like in [19] instead of normalizing all the maps like in [28], shows that noise has far less influence on the saliency map (figure 3.14). The color map which mostly consists of noise is totally suppressed in this figure.

3.1.8 Map Weighting

The weight methods used in [28] and [19] are both very sensitive to noise. If a few white pixels are encountered the weight value is set very high which results in promoting (or less suppression) the map due to a small amount of peaks while all other pixel values could be fairly low. In order to weight a map based on its maximum pixel value noise has to be removed. Because SISCA uses the original image size the image has to be smoothed first before it can be normalized and weighted, otherwise noise can mask the signal (figure 3.16).

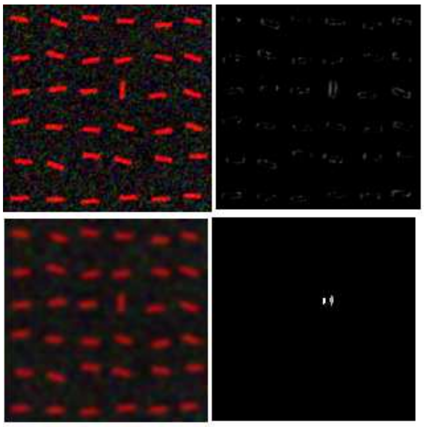


Figure 3.16: Effect of smoothing in SISCA (normalization for creating conspicuity maps only). Top row un-smoothed input image and saliency map. Bottom row: smoothed input image and saliency map.

Another drawback of the earlier mentioned weight functions is the bias for salient areas of small volume. A salient blob can contain a lot of pixels, and because only one peak is favoured this blob is considered less salient than a few pixels scattered around an image. This effect is especially noticeable in SISCA because it uses higher resolution feature maps than used in [28] and [19]. To overcome this problem another measurement has to be taken into account. A measure used in SISCA is the distribution of the peaks. A map is suppressed more when a lot of peaks are found that lie far from each other than when the same amount of peaks lie close to each other. The distribution is measured by taking the median of the squared Euclidean distances from the global maximum M to the other peaks. The other peaks are pixels with a value higher than a predefined threshold (50%) from M. Figure 3.17 shows the effect of taking the peak distribution into account. Without the distribution as weight the most salient location in figure 3.17 is on the middle red men.

Map weighting in SISCA is done as follows:

- Determine the global maximum M.
- Count the number of local maxima N above a predefined threshold from M.
- Calculate the squared Euclidean distances from M to N and find the median U.
- Divide each pixel by the square root of U times N.
- Multiply the pixel with the feature weight W.

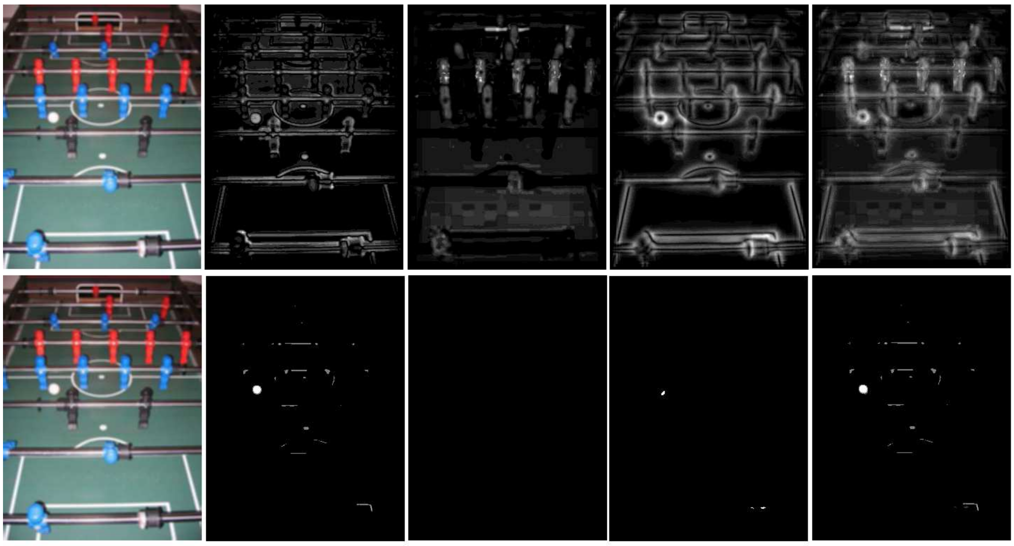


Figure 3.17: SISCA: Effect of peak counting as weight function (1e row) vs the addition of the distribution as weight value (2e row). From left to right: smoothed input image (sigma 2), intensity map, color map orientation map, and the saliency map.

3.1.9 Top Down Cueing

For top down saliency detection the map weighting method is equipped with a feature weight W. This weight value can be determined through learning in a particular environment, where a certain feature is more useful than others, or it can be set according to the search task. By setting a higher value for for example the red/green feature, red objects will become more salient.

3.1.10 Conspicuity Maps

Conspicuity maps are created for the three sub-modalities: intensity, color and orientation (figure 3.12). A conspicuity map is created by fusing the feature maps of a sub-modality. These maps are created in [28] by using the same normalization operator as with the feature maps. Their motivation for creating three separate channels, intensity, color, and orientation, and their individual normalization is the hypothesis that similar features compete strongly for saliency, while different modalities contribute independently to the saliency map. In [19] the conspicuity maps are created by first normalizing the feature maps before fusing. The values are normalized between 0 and the maximum pixel value of all feature maps of a sub-modality. SISCA uses fixed scale normalization and the same weight function as for creating the feature maps. Finally the saliency map is created by weighting the conspicuity maps and subsequently fusing the maps using point-to-point pixel addition (figure 3.18).

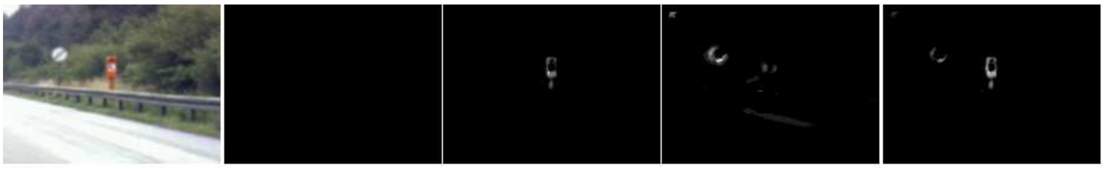


Figure 3.18: SISCA: *Conspicuity maps* and the *saliency map*. From left to right, the input image, the intensity map, the color map, the orientation map and the saliency map. The conspicuity maps are computed with smoothing factor 2, 8 receptive field sizes, peaks and distribution measure as weight function, and feature map normalization for creating the conspicuity maps.

3.2 Auditory Saliency Detection

The detection of salient audio is based on the earlier mentioned method of creating an auditory saliency map [32]. This auditory saliency map can be computed using the previously described saliency detection system. Because the visual attention system SISCA also allows top down cueing, higher weight values can be assigned to feature maps that highlight the appropriate auditory features.

Three auditory features are used for creating the auditory saliency map. The first feature is the intensity. In the visual representation of the audio data (figure 3.19) foreground sound is represented by the red color. Therefore giving the red-green feature maps in SISCA a higher weight value will result in finding salient audio based on intensity. The second feature is the frequency contrast. Frequencies are displayed along the vertical axis in the image, which means that a horizontal line represents a tone on a certain frequency. To detect the frequency contrast the feature maps that highlight horizontal edges is given a higher weight value. The last feature is temporal contrast. Because the horizontal axis represents time, the feature maps that highlight vertical edges is given a higher weight value.

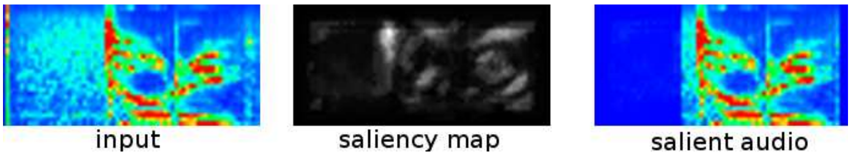


Figure 3.19: Salient audio detection. A cochleogram is used as *input* image for auditory saliency detection. Using the visual saliency detection system SISCA a *saliency map* is created from the cochleogram. Based on the salient region the start and end of the *salient audio* is determined.

3.2.1 Cochlear Filtering

The visual representation that is used for the auditory saliency map is a cochleogram. A cochleogram is a visual representation of audio that is filtered using a cochlea model. In a cochleogram audio is visualized using three dimensions: along the horizontal axis time, along the vertical axis frequency and through color the intensity.

The cochlea is a snail-shaped organ (figure 3.20) that is responsible for converting sound waves into a neural and spectral representation. The cochlea model performs a frequency analysis like that of a Fast Fourier Transform (FFT). But the advantage over a FFT is that a cochlea analysis has continuity in time and frequency.

The cochlear filtering method used here is Malcolm Slaney's implementation of Lyon's Cochlear model [48]. The model describes the propagation of sound in the inner ear and the conversion of the acoustical energy into neural representations. The cochlear has a strong compressive non-linearity over a wide range of sound intensities. This model unlike many other cochlea models takes the non-linearity into account and explicitly recognizes the purpose of the strong non-linearity as an automatic gain control (AGC) that serves to map a huge

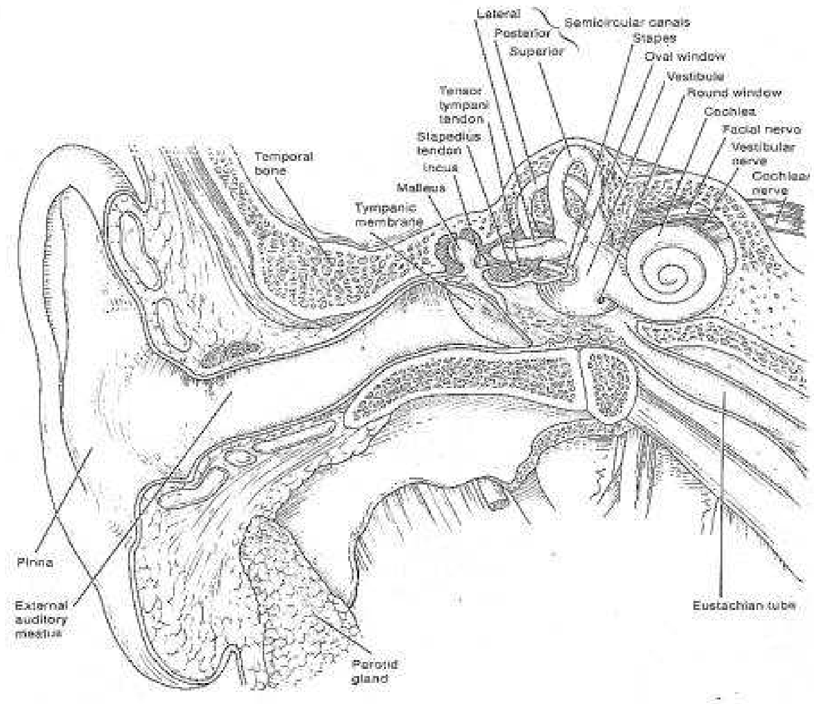


Figure 3.20: A schematic illustration of the human inner ear and cochlea

dynamic range of physical stimuli into the limited dynamic range of nerve firings [38]. The model combines a series of filters that model the travelling pressure waves with Half Wave Rectifiers (HWR) to detect the energy in the signal at several stages of the AGC (figure 3.21).

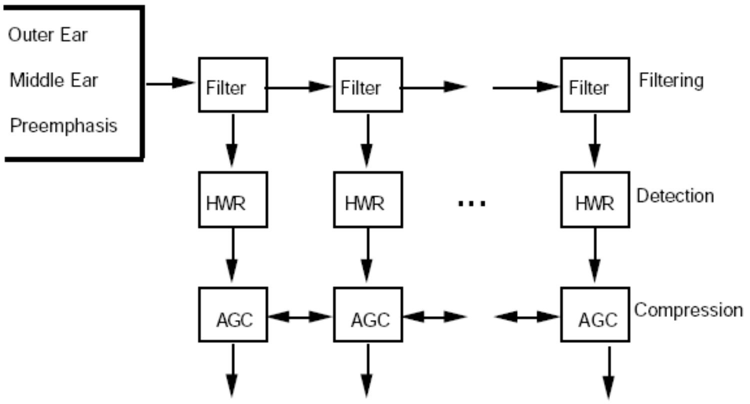


Figure 3.21: The structure of Lyon's cochlear model (figure from [48])

An important characteristic of the cochlea is that each part of the cochlea has its own resonance frequency. This has the result of mapping frequencies into the spatial domain.

3.3 Bi-Modal Attention

Early stage sensor fusion as can be found in the superior colliculus (section 2.1) lies at the basis of bi-modal attention in vertebrates. The superior colliculus is an integrator for auditory and visual information. It fuses these modalities in the spatial domain through bi-modal neurons which are responsive to interaural time differences (ITD) but also show a different sensitivity to changes in the retinotopic visual map. The mapping of the interaural cues to a spatial location (azimuth) is learned by aligning the visual location and the perceived auditory cues [25]. Learning this mapping in contrast to hard coding the relation is important when dealing with a morphodynamic organism like the Replicators. Interaural time and intensity difference are two cues which are often used for auditory localization which is then called binaural localization. The implemented bi-modal attention system is based on binaural cues and a visual salient location.

3.3.1 Binaural Localization

The localization of an object through sound is done via binaural localization of salient audio. In order to use binaural localization to steer the robot's attention, cues must me computed from salient audio, otherwise background and internal noise would cause unwanted behaviour and wasted processing time.

Salient audio is detected with the earlier described auditory saliency detection module. Based on the frequency of the input signal and the frequency bandwidth parameter, called step factor, a certain amount of channels for different frequencies are created for an audio sample. A channel contains the spike rate of the hair cells for a certain frequency in time. Another parameter to adjust the quality (and computational complexity) of the cochlear output is the decimation factor. With this parameter the output can be sampled at a different rate. Depending on the step factor and decimation factor a cochleogram of a certain size is computed for the audio samples of both audio channels. A parameter that can be set for the cochleogram is to use absolute energy or not. If absolute energy is not used the maximum intensity will be set to the highest value of the cochlear output. Because intensity is also a salient feature, absolute energy is used to keep the relative difference. From the cochleograms of the left and right channels salient regions are computed. Based on the start and end of the salient region a region of the cochlear filtered audio is used to compute binaural cues from.

Binaural cues

The first interaural cue used for binaural localization is the intensity difference. The difference in the salient region is computed by subtracting the left cochlear filtered audio from the right.

The other binaural cue, interaural time difference, is computed by means of cross-correlation. The time difference is computed by correlating frequency channels from the cochlear filtered left and right audio channel. In order to obtain a good measurement of the time difference between the two channels, every sample of the cochlear output must be used which is a decimation factor with value 1.

A Simple method to calculate the correlation is shown in the formula below. Consider two series $x(i)$ and $y(i)$ where $i = 0, 1, 2, ..., n-1$. The cross correlation r at delay d is defined as:

$$r(d) = \frac{\sum_i [(x(i) - mx) * (y(i-d) - my)]}{\sqrt{\sum_i (x(i) - mx)^2} \sqrt{\sum_i (y(i-d) - my)^2}} \tag{3.1}$$

Where mx and my are the means of the corresponding series, and delays $d = 0, 1, 2, ..., n-1$.

The location where the correlation has the maximum value is considered as the delay. This delay is measured in samples. If the maximum value lies to the left of the center then y is delayed, and if it lies to the right of the center then x is delayed. The length of the correlation series is twice the length of the original series if delays from 0 to n are used. Based on the computed cross-correlation a correlogram can be created as can be seen in figure 3.22. The values of the two binaural cues are normalized to a value between 0 and 1, where 0 means left and 1 means right. Because there is noise and no uniform distribution of cue value occurrences, it is important to at least determine where the boundaries of the center are to be able to make a good prediction of the location of an object.

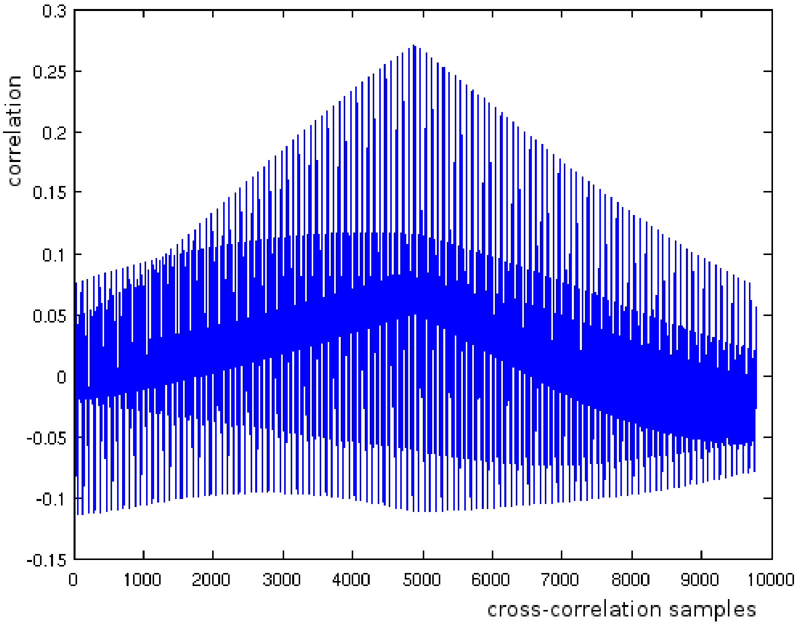


Figure 3.22: Correlogram of two identical signals x and y with $n = 5000$ where signal y is delayed.

Audio-visual integration

For binaural localization binaural cues need to be related to a spatial location. The mapping of cue values to a location is done through Hebbian learning. As in [25] audio-visual information is obtained from a visual salient object that emits an auditory salient sound. The spatial location is obtained from the visual saliency detection module by translating the salient location into a degree value in the field of view, which ranges from -60 to 60. This results in 121 locations which are used as input for a Hebbian network. The two binaural cues are also used as input and have the same amount of inputs as the amount of visual locations. Because the occurrences of cue values do not have a uniform distribution between 0 and 1, the boundaries of the cue values are first searched for by associating the minimum (-60 or 0) and the maximum (60 or 121) from the visual input to the calculated binaural cues. Because the field of view is only 120 degrees and sound is perceived in 360 degrees, all the values above these boundary cue values are classified as either left or right, respectively -90 or 90 degrees.

This Hebbian learning process is influenced by a few parameters. One of the parameters is the number of input neurons. To speed up the learning process the visual field can be divided in less than 121 locations, for instance when 5 locations are used then 2 decode the left half, one the middle half and 2 the right half (figure 3.23). This way lesser locations need to be visited in the visual field by the salient object to learn the associations of these locations. Other parameters are the learning rate and the update range. When a lot of input neurons are used updating nearby connections with a Gaussian function can also speed up the learning process. This method is suitable because of the relation between the real spatial location and the location of the input neurons.

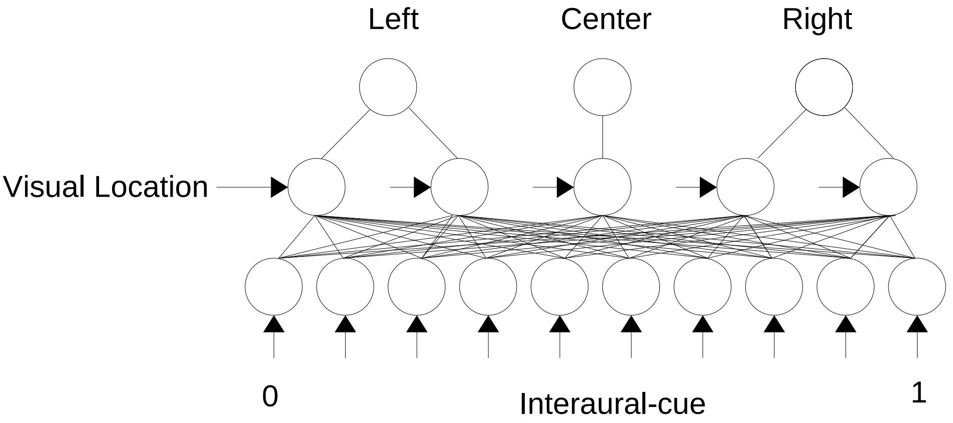


Figure 3.23: An abstract associative network for associating five visual locations to an interaural cue.

3.4 Associative Memory

When we look at biology, multi-modal sensor fusion as seen in the nervous system of insects is an associative process [55]. The modalities in which a perceived object is encoded have different dimensions in which they represent the features of the perceived object. These could be visual features, audio-temporal features, olfactory, tactile, etc. Fusing all this information will lead to the perception of that specific object or a category of objects. This way of fusing information could be based on a hierarchical architecture where there is on the highest level a single neuron that encodes an object in the brain based on a network of all the features from the different modalities at different abstraction levels. Whether a particular object (single neuron) is activated by a set of features depends on the associations that these features have with all other percepts of objects in the memory. A feature that is very distinctive for a particular object could by itself activate this object together with all its underlying features from other modalities that encode this object into consciousness. This is the proposed foundation for the multi-modal cognitive sensor fusion architecture which has as basis associative memory.

This proposed idea for multi-modal cognitive sensor fusion can be supported by recent discoveries of single neurons that encode multi-modal percepts in the human brain. Quiroga et al. [44] researched how different stimulus modalities can evoke the same ”concept” of for instance a famous person by seeing a picture or by hearing or reading the name. They showed that (1) single neurons in the human medial temporal lobe (MTL) respond selectively to representations of the same individual across different sensory modalities; (2) the degree of multi-modal invariance increases along the hierarchical structure within the MTL. With their current data it was not possible to provide a conclusive mechanistic explanation of how such abstract single-cell multi-modal responses arise, but evidence points toward a role of the MTL in forming associations by for instance linking faces with written and spoken names. Recognized abstract patterns from different modalities are thus associated in one location where the concept of an object is stored. In the lower part of the hierarchy uni-modal neurons like in the inferior temporal cortex (IT) (which respond to visual stimuli) encode percepts in a distributed way, and have a limited degree of invariance which makes them responsive to similar but also slightly different percepts. This type of information from multiple sensory modalities are associated into a single percept in the MTL.

In the following section all the separate parts of the proposed multi-modal sensor fusion architecture will be described. Starting at the bottom of the processing hierarchy, a distributed clustering and pattern recognition method will be described that resembles the function of the IT neurons / Kenyon cells, followed by the description of an associative memory module that creates a multi-modal percept like in the MTL / mushroom bodies.

3.4.1 Adaptive Resonance Theory

The Adaptive Resonance Theory (ART) is a theory about information processing and storage in the brain. It was developed by Grossberg and Carpenter [15]. Principles derived form an analysis of experimental literatures in vision, speech, cortical development, and reinforcement learning, including attentional blocking and cognitive-emotional interactions, led to the introduction of adaptive resonance as a theory of human cognitive information processing [15]. The first version of ART also called ART-1 is an unsupervised binary clustering or pattern matching system. The basic model of all the ART systems (figure 3.24) consist of a short term memory input pattern (F1) which is matched against patterns that are in the long term memory (F2). An input pattern could either be in resonance with a long term memory node, which means that the input pattern matches the pattern in memory to a satisfying degree, or there could be no pattern in memory that resembles the input pattern which then leads to the storage of the input pattern as a new memory node. This match-based process is the basis of the ART system that deals with the stability-plasticity dilemma.

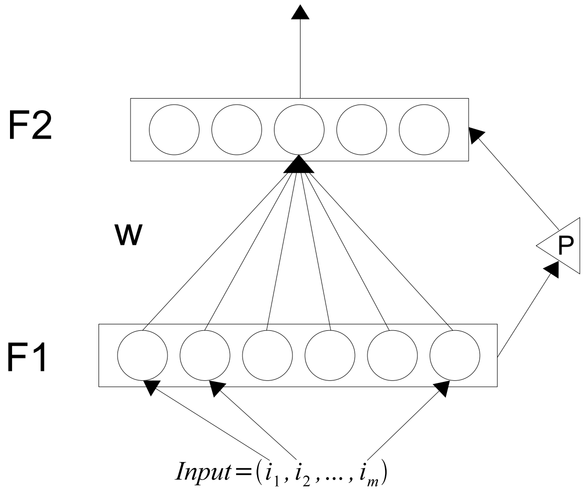


Figure 3.24: An abstract representation of the ART network. The input pattern has M elements and is put in short term memory F1. The pattern from F1 is compared to the patterns in long term memory F2. P is the vigilance parameter which specifies the amount of resemblance needed between F1 and a F2 node for a match.

Within the ART system an F2 memory or category node is chosen as possible candidate based on its similarity with the input pattern. The similarity is denoted by the signal value T_j (see equation (3.2)). The memory node with the highest signal value is selected for a resonance test. The ART system provides stability through the matching criteria parameter P called vigilance. With the vigilance parameter the amount of resemblance needed for a match can be set in the form of a minimum confidence value (see equation (3.3)). With a low vigilance value there has to be less resemblance to have resonance, this leads to fewer and more abstract memory nodes. Whereas a higher vigilance value will lead to more memory nodes that only have resonance with very similar input.

Learning within the ART system is done by storing a new input pattern if no resonance with F2 is found, or by updating the memory node which is in resonance with the input. Updating the weights of the existing node is done in such a way that it is monotonically non-increasing, it will always be able to classify earlier learned patterns. If fast learning is used the weights of the memory node are updated in a way that the input pattern just falls within the memory node's boundaries (see equation (3.4)). If slow learning is used then the memory node is updated only a small fraction in the direction of the presented input pattern.

ART 1
Category choice:

$$T_j = \frac{|I \cap w_j|}{\alpha + |w_j|} \tag{3.2}$$

where T_j is the signal value, I is the input vector, w_j the weight vector of the Jth F2 memory node, and α the signal rule parameter

Match criterion:

$$\frac{|I \cap w_j|}{I} \geq p \tag{3.3}$$

Fast Learning:

$$w_j^{new} = I \cap w_j^{old} \tag{3.4}$$

During the years several types of the ART systems have been developed. After the binary ART, ART-1, a variant was made to support continuous inputs which is called ART-2 [8]. A streamlined version of the former is ART-2A [11], this version needs less computation time and has only slightly worse qualitative results. Fuzzy-ART [7] uses fuzzy logic in pattern matching and has a means of incorporating the absence of features into pattern classifications through complement coding. In Fuzzy ART the logical AND $\cap$: intersection is replaced by the fuzzy AND $\wedge$: minimum.

Preventing category proliferation while monotonically non-increasing the memory node's weights is in Fuzzy-ART achieved by using a complement coded input (see equation (3.5)). A complement coded input pattern is a vector with normalized input values [0,1] where the second half of the vector consists of the complement values of the first half. The sum of the vector equals the length of the vector. In figure 3.25 it is shown that the cluster size is enlarged when the weight values are updated by taking the maximum vector values of two compared patterns.

$$I_c = (I_1, I_2, ..., I_m, 1 - I_1, 1 - I_2, ..., 1 - I_m) \tag{3.5}$$

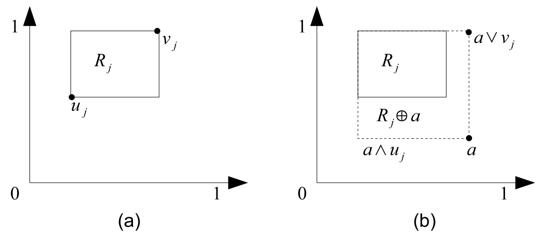


Figure 3.25: Fuzzy art cluster representation. (a) Having a two dimensional complement coded input vector, each weight vector w_j has a geometric interpretation as a rectangle R_j with corners (u_j, v_j). (b) Updating the weight for input a with fast learning, R_j expands to $R_j \oplus a$, the smallest rectangle that includes R_j and a, while satisfying the match criterion.

3.4.2 ARTMAP

An extension to the ART network for supervised learning is the ARTMAP [10]. The ARTMAP provides means to steer the clustering process by a secondary ART network with the correct output. The correct output classes for each input pattern is learned with an associative network. The structure of the ARTMAP is as follows, it consists of an ART network for classifying input patterns let's say ART$_a$, a secondary ART network with the correct output say ART$_b$, and an associative learning network that links ART$_a$ to ART$_b$ called the map field (figure 3.26).

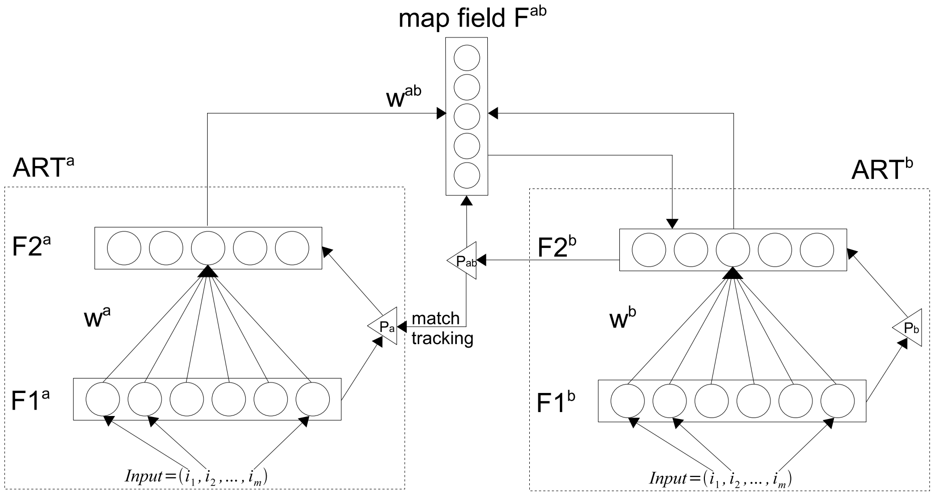


Figure 3.26: An abstract representation of the ARTMAP network. This ARTMAP consists of two ART networks ART^a and ART^b. ART^b is the supervisor network that is able to send a reset or match track request to ART^a when the output of ART^a is inconsistent with the expected output calculated by ART^b through map field F^{ab}.

The ARTMAP is trained by providing an input pattern a for ART$_a$ and the correct output b via ART$_b$. ART$_a$ processes the input pattern by finding a memory node J that is in resonance with the input based on a minimum confidence value p_a. When no match is found a new node is created in memory that resembles the input pattern. This memory node is then connected via a map field node X to the output node K of ART$_b$ which is established with the same matching processes. In the case where there is a match found in ART$_a$, the winning memory node J will activate via its weights w_j^{ab} a map field node X, if ART$_b$ is active, then only if output node K from ART$_b$ activates the same map field node X via its one-to-one pathway, F^{ab} will become active. Similar to ART, a vigilance parameter p^{ab} is used to determine if the activation is in resonance.

If ART$_a$ activated a different map field node than ART$_b$, equation (3.6) is not satisfied and a match tracking process is started in ART$_a$. A better match is searched for by slightly increasing the confidence or vigilance parameter p_a so that the previous winning memory node is no longer a candidate. This match tracking process will eventually end in a correct match or a new memory node that will be associated via the map field with the correct output node of ART$_b$. With fast learning the weights w_{jk}^{ab} from ART$_a$ node J to ART$_b$ node K is set equal to 1, which makes it a permanent association.

$$\frac{|X|}{|K|} \geq p^{ab} \tag{3.6}$$

where

$$X = \begin{cases} K \cap w_j^{ab} & \text{if the } J\text{th } F_2^a \text{ node is active and } F_2^b \text{ is active} \\ w_j^{ab} & \text{if the } J\text{th } F_2^a \text{ node is active and } F_2^b \text{ is inactive} \\ K & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is active} \\ 0 & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is inactive} \end{cases}$$

Testing is done by providing an input pattern for ART$_a$ after which a winning memory node is selected using a Winner Take All (WTA) method. This winning node activates the associated memory node of ART$_b$ via the connections in the map field. The output of this ARTMAP could be a class label associated with the input pattern via ART$_b$.

Many variants of the basic ARTMAP networks have been created to name a few: fuzzy ARTMAP [9], ART-EMAP [14], ARTMAP-IC [12], and the distributed ARTMAP [13]. Comparative analysis of these networks has led to the Default ARTMAP [6] and the Default ARTMAP 2 [2], which has a simplified design and a better performance in many application domains. The default ARTMAP is the same as fuzzy ARTMAP during training, but uses a distributed winner selection during testing. ARTMAP-IC is the same as the default ARTMAP plus instance counting, this biases a category node's test set output by the number of training set inputs coded by that node. The distributed ARTMAP uses a distributed winner selection during both training and testing and also uses instance counting. The Fuzzy ARTMAP is the basis for all these ARTMAP variants, and uses the earlier described WTA method. The difference between the Fuzzy ARTMAP and the first ARTMAP is the use of Fuzzy ART networks for pattern recognition. The default ARTMAP also implements the match tracking search procedure, with the baseline vigilance parameter equal to zero for maximal code compression, and uses fast learning.

The ARTMAP is well suited for unidirectional supervised learning tasks where there can be a many-to-one mapping of input patterns to output classes. Due to match-tracking and the match criteria in the map field, bidirectional and many-to-many associations are not possible. A variant of the Fuzzy ARTMAP that deals with this problem is the Bi-directional ARTMAP (BiARTMAP) [5]. By introducing a second map field that handles the associations and the match track process in the other direction, the BiARTMAP is able to handle bidirectional mappings ranging from one-to-one to many-to-many. With BiARTMAP both ART networks, ART$_a$ and ART$_b$, can be either used as input or as output network.

Although the associative capabilities of the BiARTMAP network starts to look more like the biological associative process, it still lacks on a few aspects. One main aspect that all the ARTMAPs have is supervised learning. In order to use associative memory in an ongoing learning setting, where there is no separate learning phase, unsupervised associative learning must be possible. Another important aspect is the lack of being able to associate multiple ART networks in a network that is multi-directional. BiARTMAP enables bidirectional association and lookup by means of a second map field. Increasing the number of map fields together with the number of input ART networks seems not only very unlikely from a biological perspective, but also unnecessary complex and computationally expensive. Because the BiARTMAP is based on the Fuzzy ARTMAP it also does not use a distributed winner selection in the testing phase. A distributed winner selection as can be seen in the default ARTMAP leads to a better performance and is also more plausible from a biological perspective.

In the next section a new variant of the default ARTMAP will be proposed that is capable of associating multiple ART networks with unsupervised learning. This new ARTMAP, called the Multi-directional ARTMAP (MdARTMAP), has more similarities with the biological associative processes described earlier. Also analogies between the hierarchical structure of this multi-modal sensor fusion architecture and the hierarchical structure of the MTL can be found.

3.4.3 Multi-directional (Un-)Supervised ARTMAP

ART networks have proven themselves to be useful as an unsupervised learning mechanism when ongoing learning is needed because they don't suffer from the stability-plasticity dilemma. And because ART is also used as a model to describe cognitive memory processes, it seems very suitable for the low level building blocks of the cognitive sensor fusion architecture. This sensor fusion architecture will consist of a hierarchy of ART networks which are linked by associative learning.

Multi-directional association

Associating the outputs of all the ART networks is done in a single associative network called the map field (figure 3.27). This network can associate multiple supervised and unsupervised ART networks. The difference with the default ARTMAP is the possibility to have one-to-many and many-to-many mappings in the associative network. The match-tracking process in the default ARTMAP only allows a match-tracking process for one network, ART$_a$, and only accepts the ART class that was already associated with this ”class label”. With that match-track process it is not possible to have an input pattern that belongs to multiple ”class labels”, which is possible in reality. For example a single sound pattern can be created by different objects, and thus must be associated to multiple classes.

The MdARTMAP is able to use supervised learning with match-tracking, but it also allows the binding of one ART class to multiple ART classes of another ART network. One-to-many mappings in multiple directions is made possible through distributed winner selection using multiple ART network classes. When input patterns are presented to all the ART networks, their output classes, which are connected to map field neurons, activate the associated neurons in the map field (associative network) (see equation (3.7)). The associative neuron with the highest activation value (see equation (3.8)) is selected as winner.

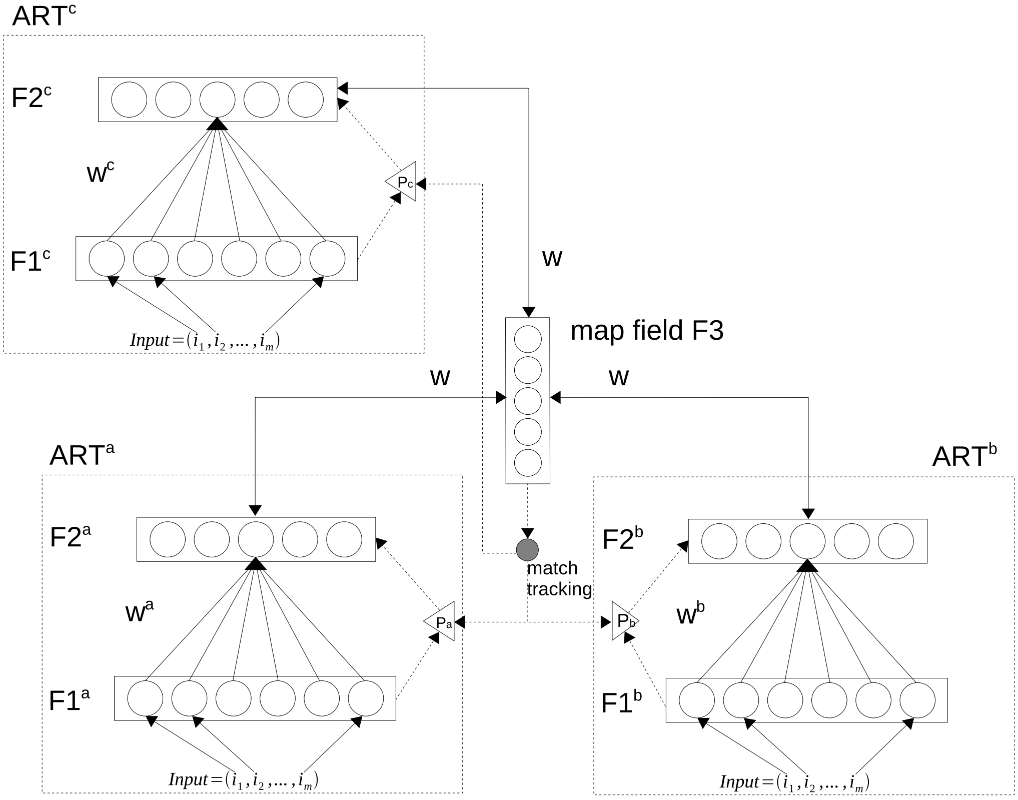


Figure 3.27: An abstract representation of the MdARTMAP network. This MdARTMAP consists of three ART networks ART^a, ART^b and ART^c. The output classes of each ART network are associated in the map field F3. A match tracking process can be initiated by one of the highest ranked ART networks when the output of a lower ranked ART network is inconsistent with the expected output calculated by the higher ranked ART network through map field F3.

Map Field node activation function:

$$Act(X) = |W| + \frac{\sum_{i}^{|W|} w_i}{|ARTnetworks|} \tag{3.7}$$

where W is a vector with all active connections to map field node X
and the normalized connection strength $w_i \, \epsilon \, [0, 1]$

Map Field winning node selection:

$$winning\_node = \arg\max_{X}(Act(X)) \tag{3.8}$$

where $X \, \epsilon$ Map Field nodes

The activation value of a map field neuron depends on the number of connections and the strength of these connections to associated classes. The connection strength is learned through hebbian learning. When a map field neuron is activated by at least two classes then the connection strength from the neuron to the classes is increased. Classes that are often perceived together have a high connection strength, but a connection with a high connection strength can never give a higher activation value than two connections with a low strength. This means that the number of ART networks that vote for a certain associative neuron weighs more than the strength of the association of a certain ART network class. This weight function ensures that very often perceived classes will not automatically lead to one ″output class″. If a system only encounters red robots that make a certain sound and ″give″ energy, the system will have a high connection strength for red robots and the ″give energy″ property. If a red robot is encountered that makes a different sound a new association will be made based on the findings whether this robot gives energy or not. If it learns that this robot does not give energy, seeing a red robot with that different sound activates the later learned association, based on the number of connections and not on the high connection strength of the red robots class to the neuron that is associated with the ″gives energy″ class. Therefore the map field is updated based on the following rules:

```
if Act(winning_node) >= |ART networks| then
   increase strength of active connections W to winning_node
else if (Act(winning_node) >= 2) and (Act(winning_node) >= |W|) then
   connect ART classes to winning_ node and increase connection strength of W
else
   create a new map field node and connect it to all ART output classes
```

(Un)-Supervised Learning

The ARTMAP is created to steer the clustering processes when training examples are available. But since the ARTMAP is based on the ART network, which is known for its unsupervised learning capabilities, it is also suitable for unsupervised learning. The hardest part of unsupervised learning with an ART network is finding the right vigilance parameter. Finding this parameter is an iterative empirical process where the trade-off between generalizing and abstracting has to be made. Such a process could be of evolutionary nature, but using knowledge about the future data could also give satisfying results.

Information from modalities differ in resolution, variance and thus in reliability. To cope with this the vigilance parameter can be tuned, but this only works to a certain extent. With knowledge about the reliability of the information sources / modalities, the individual ART networks in the MdARTMAP can be ranked and are able to steer each other′s search process. When using a high resolution camera and a low quality microphone, conflicting predictions made based on information from those modalities will be in favour of the camera.

Match-tracking for reliability ranked ART networks is only activated when there is a mismatch between the outputs of the ART networks. This mismatch occurs when the outputs of the ART networks both have associations to other output classes than the currently activated ones. The most reliable network will then initiate the match-track process for the other network(s). A new match will eventually be found, this could either be the output class that was already associated with the output of the supervisor network(s), or a new output class could be created and also associated with the output of the supervisor network(s). Because no initial zero vigilance is used, the match-track process does not force a specific outcome.

By adding a reliability measure to each ART network both supervised and unsupervised clustering can be obtained.

When the MdARTMAP consists of equally ranked ART networks then match-tracking is not used. Associations are made based on the classifications of the unsupervised ART networks. A well tuned vigilance parameter and a distributed winner selection method are important for this unsupervised process. The conditions needed to initiate match tracking are:

- Unequal ranked networks
- Highest ranked ART networks activate the same map field neuron X
- Lower ranked ART network does not activate map field neuron X

The association process

The association of ART network classes focusses on simultaneous perceptions and retrieving missing perceptions based on learned associations. In the following part a detailed description of the (un)-supervised association process will be given for multiple ART networks.

Based on the outputs of the ART networks, called ART classes, associations are learned using the following steps:

1. For each ART class calculate the activation of associated map field nodes
2. If no map node is activated, then associate the ART classes to a new map field node
3. If all the ART classes activated the same node then update W of the winning node X
4. If all ART classes activated X except the new connectionless ART classes then:
    (a) Associate the new ART classes to X if X has no associations with those ART networks and update W
    (b) Otherwise create a new map node and associate all the ART classes
5. If not all ART classes activate X then:
    (a) Create a new map node and associate all ART classes if:
        - All networks are equally ranked.
        - Or the highest ranked ART networks (supervisors) do not all activate X
    (b) Match-track all lower ranked ART networks if all supervisors activate X
        i. Update the connections W if all ART classes activate X
        ii. Associate new ART classes to X

Retrieving associated data is done using a distributed winner selection method. When for instance two of three ART classes are given as input, the third class is retrieved based on the learned associations. As in the learning process a winning map field node is selected based on the number of connections and secondly the connection strength. This winning map field node activates the ART class with the strongest connection from the ART network that was selected for retrieval.

3.4.4 Distributed Clustering

ART networks are known for dealing with the stability-plasticity dilemma but also for their lack of handling certain invariances. The classification algorithm of the used Fuzzy ART network performs a one to one comparison of the input vector with the stored memory nodes. Any shifts in the input pattern will lead to a wrong classification. Also partially observed patterns can not be classified. This puts a high constraint on ART for using it in real-time where ongoing classification is needed. For example when recognizing sound in a dynamic environment (real world), partly observed and shifted sound samples are often encountered.

To overcome these problems a distributed approach is used where features are clustered that are shift invariant. In this approach an ART network is used to recognize the features of a class which are all associated to that class via the previous described ARTMAP.

The features (F) that represent the class must of course be as descriptive and invariant as possible. Each feature can belong to multiple classes, and each class has multiple features. To be able to classify a set of orderless features a distributed winner selection method based on all the features is used for the MdARTMAP. Based on all the features there could either be a positive classification in which the class is known in memory, or there could be no classification in which a new object is encountered. Learning new patterns with this distributed clustering network (DCN) is done by first performing a test whether a set of features will lead to a reliable classification. If a class is found then all features are given as input to the ARTMAP subsequently together with the associated class. The connection strengths to associated features are then increased and connections to the class are created for new features.

The distributed winner selection method does not calculate the winning map field node, but the associated winning ART class which is the ”class label” for the input pattern (see equation (3.11)). This is done because multiple map field nodes are connected to one ART class. The activation value of the winning ART class (see equation (3.9)) is used to determine the probability whether this class belongs to the input pattern (see equation (3.10)). The probability is measured by dividing the activation value of the winning class by the total amount of class activations. This probability value will only be accepted if the amount of associated features is above a threshold which is dependent on the number of input features.

$$Act(class, F) = \sum_{i=1}^{n}(w_i) \tag{3.9}$$

$$\text{where } \{w_1, ..., w_n\} \,\epsilon\, ConEdge(activeNodes, class)$$
$$activeNodes \,\epsilon\, ConNode(F, MapNodes)$$

$$P(class|F) = \frac{Act(class, F)}{\sum_{i=1}^{n} Act(class_i, F)} \tag{3.10}$$

$$winning\_class = \arg\max_{class}(P(class|F)) \tag{3.11}$$

With this distributed classification method not only shift invariant but also sets of features can be classified. When classifying a temporal pattern each point in time can be used as separate feature for the feature set. The DCN does not learn the order of features in a set. Therefore in order to classify a pattern for which the sequence is important (e.g. audio) preprocessing is required.

3.4.5 Hierarchical Associations

With the previous described DCN it is now easy to see that a hierarchy of associations can be formed when combined with the MdARTMAP (figure 3.28). The Fuzzy networks used in the MdARTMAP can be replaced by DCNs, which will extend the MdARTMAP to be able to classify shift invariant and temporal patterns. This can be realized by using the output from the DCN, which is an ART class, as input for the MdARTMAP. For computation time and complexity only the winning ART class of the DCN is used for the associations, instead of a distributed output based on the activations. In figure 3.28 a hierarchical MdARTMAP is shown. In this figure the ART networks ART^a and ART^b from figure 3.27 are replaced by two individual MdARTMAPs with each two ART networks. Each MdARTMAP has a map field in which features from ART^{x1} are associated to a higher class in ART^x. These higher classes (from ART^a, ART^b, ART^c) are subsequently associated in the map field of the main MdARTMAP.

3.4.6 Conclusion

In this section a new type of ARTMAP was proposed which is capable of creating multi-directional supervised and unsupervised associations. By extending it with a distributed clustering network it is capable of classifying temporal patterns as well as being able to handle more invariances than former ARTMAPs. The hierarchical structure of this network resembles the structure of the medial temporal lobe in the human brain, and the mushroom-bodies in the Dorsophila nervous system. Analogue to those systems distributed uni-modal encoding of features is done with a limited degree of invariance to the feature patterns, followed by the association of the more abstract uni-modal percepts into a multi-modal concept. In the next sections the implementation of the distributed clustering network for sound and object recognition will be described.

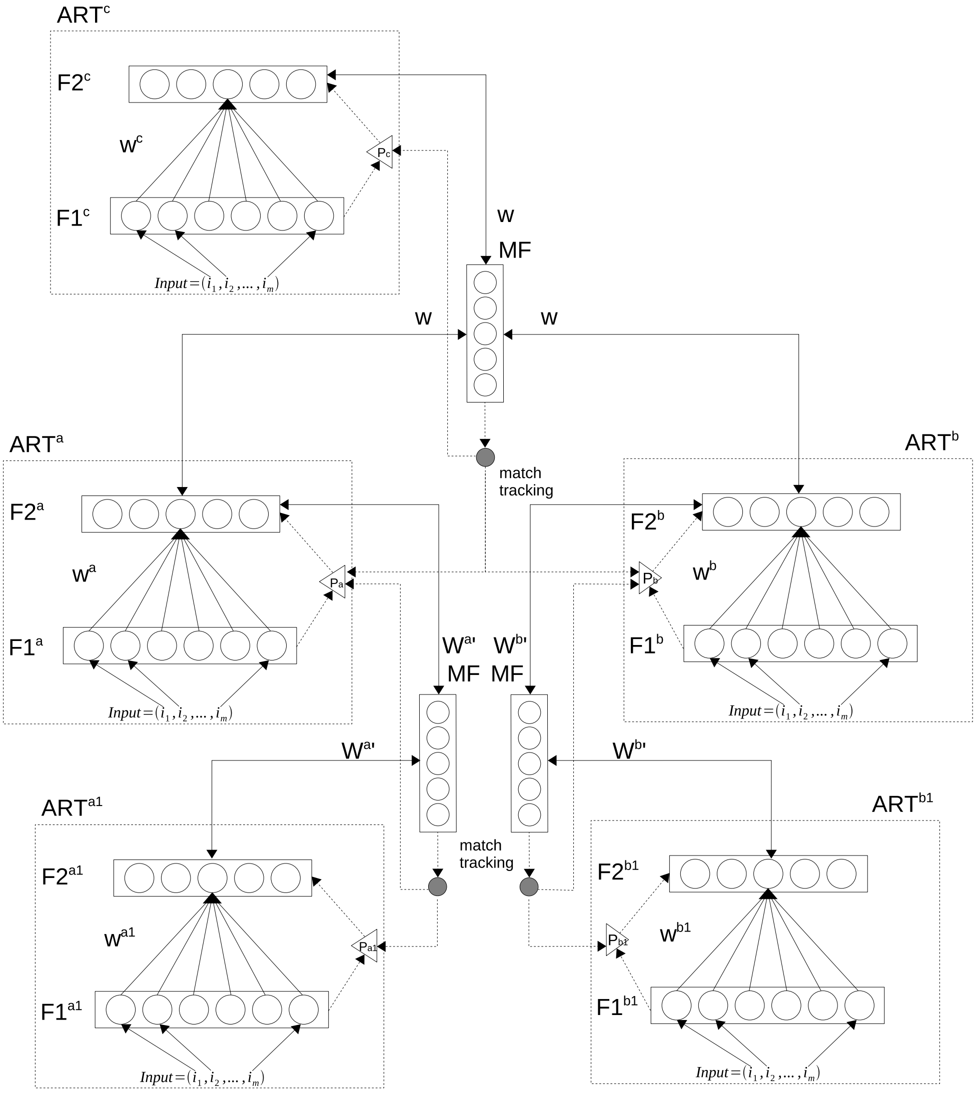


Figure 3.28: An abstract representation of the sensor fusion architecture using a hierarchy of combined MdARTMAP networks. This MdARTMAP consist of a ART network and two MdARTMAPs which each consist of two ART networks. The first MdARTMAP consists of ART^{a1} and ART^{a} and the second MdARTMAP of ART^{b} and ART^{b1}. The output values of ART^{a}, ART^{b} and ART^{c} are all associated in the map field MF with connections W.

3.5 Sound Recognition

One of the modalities used in cognitive sensor fusion is sound. When an interesting sound source is detected using the earlier described bi-modal attention system, sound patterns must be learned to be able to distinguish objects based on the sound they emit. To be able to learn and recognize sound in a un-constraint real-time setting, a robust un-supervised sound recognition method with ongoing learning is needed. In this section such a sound recognition method will be proposed. First the preprocessing phase with the use of reservoir computing will be described followed by a description of an un-supervised sound recognition system build with the previous described MdARTMAP.

3.5.1 Reservoir Computing

The Echo State Network (ESN) is one of the well known recurrent neural networks (RNN) used in reservoir computing. RNNs have the ability to model highly non-linear systems, and are capable of processing temporal information. The hard part of using RNNs is training the network. Three different types of RNNs have been described to overcome this problem, Echo State Networks [29], Liquid State Machines (LSMs) [39], and Back propagation Decorrelation (BPDC) [49]. With reservoir computing a randomly connected RNN is used as a reservoir that is not trained but read out by a simple classification layer. The reservoir has the function of a kernel, that is: projecting the input to a higher-dimensional space in which it is better separable. The advantage of a reservoir in comparison to kernel-based methods (e.g. SVM) is the ability to incorporate temporal information.

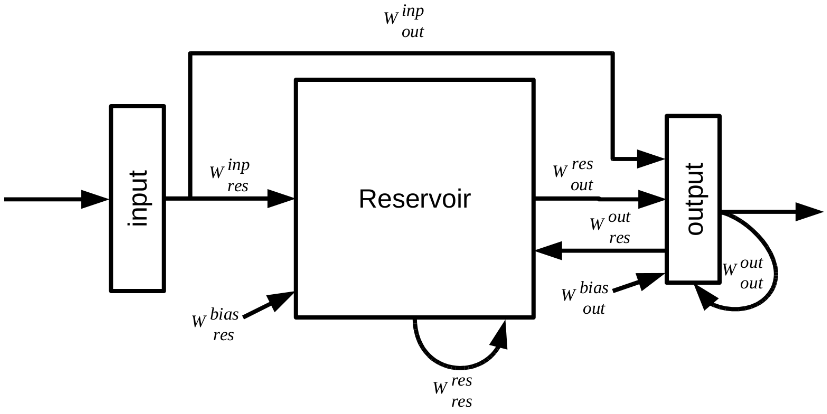


Figure 3.29: General reservoir computing architecture. The following connection weight vectors are labelled in the figure W^{inp}_{res}: input to reservoir, W^{inp}_{out}: input to output, W^{res}_{res}: reservoir to reservoir, W^{bias}_{res}: a bias value to the reservoir, W^{res}_{out}: reservoir to output, W^{out}_{res}: output to reservoir, W^{out}_{out}: output to output, W^{bias}_{out}: bias value to the output.

Reservoir computing has been successfully implemented in several application domains. It has for example been used in dynamic pattern classification, tone generation, object tracking and prediction, reinforcement learning, and also Digital Signal Processing (DSP). For an overview read [46]. Because of its temporal processing capabilities, and successful implementations in speech recognition [40, 47, 53], reservoir computing is chosen for preprocessing the microphone sensor data to obtain better classification results.

3.5.2 General Reservoir Model

A generic reservoir computing architecture is shown in figure 3.29. During reservoir simulation the reservoir states and output states with teacher forcing are computed with the following equations:

$$\mathbf{x}(t+1) = f(W_{res}^{res}\mathbf{x}(t) + W_{res}^{inp}\mathbf{u}(t) + W_{res}^{out}\mathbf{y}(t) + W_{res}^{bias}) \tag{3.12}$$

$$\hat{y}(t+1) = W_{out}^{res}\mathbf{x}(t+1) + W_{out}^{inp}\mathbf{u}(t) + W_{out}^{out}\mathbf{y}(t) + W_{out}^{bias}, \tag{3.13}$$

where $\mathbf{x}(t)$ is the reservoir neuron state vector for time step t, $\mathbf{u}(t)$ the input vector, f the neuron activation function, $\mathbf{y}(t)$ the teacher input vector, and $\hat{\mathbf{y}}(t)$ the state of the output neurons.

All the connection weights are randomly generated using some kind of distribution of connectivity and connection type, except for W_{out}^{res} which are obtained through learning.

3.5.3 ESN vs. LSM

The primary goal for developing an LSM was to provide a biologically plausible paradigm for computations in generic cortical microcircuits, while ESNs have been designed for high performance engineering tasks. LSMs therefore consist of biologically inspired spiking neurons with a small world interconnectivity pattern. Descriptions of ESNs can be found with analogue neurons and several different interconnection structures. For implementing reservoir computing in the sensor fusion model, the most important aspect is the performance in relation to computational complexity. Verstraeten et al. [52] compared reservoirs using different node types, for a broad range of parameter settings and tasks. They concluded that the computational cost of a spiking reservoir is higher but the performance was better on a speech recognition task of isolated digit recognition. They also showed that the memory capacity of both, spiking and analogue, reservoirs increases monotonically with the size, and found a strong dependence on the spectral radius for analogue neurons.

3.5.4 Implementation

Due to the computational constraints of the sensor fusion system, an ESN is used for preprocessing audio data. The main architecture of the ESN follows the generic reservoir design. The parameters used for the reservoir are based on work of Jaeger [30] and Venayagamoorthy [51] for the general ESN working, Holzmann [24] and Verstraeten [53] for the combination of an ESN with audio processing and recognition, and Morse and Ziemke for the combination with associative memory and robotics [41].

The connection weights are generated using a connectivity parameter. With a certain connectivity the connections between the neurons are generated with weight values between -1 and 1. The reservoir and output neurons can be built from different types of neurons they can be sigmoid or linear and can optionally be leaky integrator neurons. The state of a leaky integrator reservoir neuron with leak rate α is calculated using the following equation:

$$\mathbf{x}(t+1) = f(\alpha W_{res}^{res}\mathbf{x}(t) + W_{res}^{inp}\mathbf{u}(t) + W_{res}^{out}\mathbf{y}(t) + \mathbf{x}(t) * (1-\alpha)) \tag{3.14}$$

The implemented state update equation is different from the generic update equation, because no teacher forcing and feedback connections are used when simulating the reservoir. An important property of the reservoir is the spectral radius. The spectral radius is the largest absolute value of the eigenvalues of the reservoir weight matrix. To obtain the echo state property, the network must be on the edge of stability, this is obtained with a spectral radius between 0 and 1. A spectral radius of 0.8 seems to give a good performance for a variety of tasks [51]. The reservoir weights are first normalized using the largest eigenvalue of the reservoir, and subsequently scaled using the desired spectral radius.

Sound processing

In order to classify complex sound patterns an ESN can be used to transfer the non-linear separable sound patterns to a higher dimension in which it could be linearly separable. Verstraeten et al. [53] proved that cochlear filtered sound is a good representation to use for sound recognition with an LSM. The cochlear data can be fed into the reservoir by aligning the channels (frequencies) of the cochlear data to the input of the ESN. Each time step one frame of the cochlear data is entered into the ESN, which changes the states of the reservoir neurons that are connected to the input neurons. The state of other reservoir neurons that are not connected to the input neurons are changed due to the recurrent connections, no teacher input or feedback connections are used. Based on the states of all the neurons, patterns can be recognized with a readout function.

Verstraeten et al. [53] successfully used an LSM with cochlear filtered input data to recognize isolated spoken digits. The conversion of analogue cochlear values to spike trains was done using a filter encoding scheme (BSA). The best performance was obtained with a linear classifier as readout function. This showed that the LSM was capable of transferring the cochlear filtered data to a linearly separable representation. Previous studies [52] have shown that an LSM performs better on temporal patterns than an ESN. Therefore experiments must be conducted to determine whether an ESN is able to transfer the cochlear filtered data to a linearly separable dimension. The experiments must also show to what extent an ESN can handle noisy and partially observed input patterns. To only test the performance of the ESN a linear classifier is used as readout function.

3.5.5 Experiment

To test the separability of the ESN, a speech recognition task is created using a dataset with 250 samples of 5 english vowels spoken by 50 different male speakers, taken from the Hillebrand vowel dataset [23] . The words used are: ”hae”, ”her”, ”hih”, ”hoo”, ”huh”.

A second task is created for testing the robustness of the network. In this recognition task a spoken word needs to be recognized which is transformed with an effect. The network is first trained with 9 spoken digits in Māori after which it needs to recognize different instances of the spoken digit ”iwa”. The audio sample ”iwa” was edited and 11 different versions were used for testing the robustness of the the network. The applied editions are: reverberation (auditorium template), first half of the sample, last half of the sample, 150% amplitude, 125% pitch, telephone and AM effect, flanger and chorus effect, highpass filter (AM template), pink noise (SNR=-5.6), white noise (SNR=-0.5 DB), and ”iwa” spoken by a female.
The audio samples are first filtered using the earlier described cochlear model [48], with decimation factor 130, and step factor 0.25. The filtered audio is used as input for the reservoir, using 84 input neurons, and a fixed sample size of 60. The ESN parameters used are shown in table 3.1.

Table 3.1: Default parameters of the implemented ESN.

Default ESN parameters	
input connectivity	0.1
reservoir connectivity	0.5
feedback connectivity	0.0
reservoir activation function	tanh
output activation function	linear
spectral radius	0.8
input shift	0.0
reservoir shift	0.0
feedback shift	0.0
input scale	1.0
reservoir scale	1.0
feedback scale	1.0

3.5.6 Classification

To test the performance of the reservoir, a linear classification method, ridge regression is used. The weights of the readout neurons are calculated as followed:

$$W^{out} = (R + \alpha^2 I)^{-1} P, \tag{3.15}$$

where $R = S'S$ is the correlation matrix of the extended reservoir states $S = (X + U)$, α^2 is the smoothing factor, I the identity matrix, R^{-1} denotes the inverse of the matrix R, and $P = S'D$ is the cross-correlation matrix of the states S and the desired output D, which is obtained using Fisher labelling [3].

The actual classification is done using a linear projection of the input $\mathbf{u}(t)$ and reservoir states $\mathbf{x}(t)$ to the output $\mathbf{y}(t)$ using weights $\mathbf{w}$:

$$\mathbf{y}(t) = \mathbf{w} \cdot \mathbf{s}(t) \tag{3.16}$$

The winning class is selected using winner-take-all (WTA) selection, by taking the maximum value of the output **y**(t) over time. The classification performance was also tested using a winning class selection that computed the winner over a predefined amount of samples, and a final class selection using WTA. But the performance of this method for several samples sizes was not better than when computing the winning class over all the samples.

3.5.7 Results and Conclusion

For testing the reservoir separability, several reservoir sizes where used while conducting the speech recognition task with the 5 enlish vowels. The reservoir that had the best performance was the one with size T/10 (T=sample size), which is in the range that Jaeger [30] suggested. The average performance was a recognition of 96% on the training set and 95% on the test set, with 10-fold cross validation. Considering the dataset the network performance is average/good. Better results have been obtained using an LSM Verstraeten [53]. But considering the goal of the network: pre-processing with ″low″ computational costs, an ESN of 6 reservoir neurons is very suitable.

The robustness of the reservoir was tested by training the reservoir on every spoken digit once, and testing it several times on the edited versions of the word ″iwa″. For this task the best reservoir size was between between 300 and 400 neurons, which is more than the suggested amount by Jaeger. The average score on the training set was 100% and on the test set 65% (see table 3.2). The ESN was not always able to classify the word ″iwa″ with a pitch of 125%, or spoken by a female correctly. It seems that the reservoir is sensitive to shifts between neurons in the input vector. It also did not recognize the version with white noise correctly, and sometimes the pink noise version was mistaken for a word that looked like it (″wha″) because of the noise. The ESN showed to be able to generalize and still have great separable capabilities. But a problem encountered is the use of the ESN for different tasks. Different tasks demand different reservoir sizes. A larger size is needed to be able to recognize noisy samples, and smaller reservoirs are needed to generalize for inner class variance.

Table 3.2: Classifications scores of the robustness sound classification task.

Results robustness	
sample	score
female iwa	0%
reverb	100%
1st half	99%
last half	97%
150% amp	100%
125% pitch	0%
telephone and AM	100%
flanger and chorus	100%
highpass filter	100%
pink noise	20%
white noise	0%

3.5.8 The ART of Sound Recognition with Echo State Clustering

Sound recognition with an Echo State Network (ESN) was proven to be a suitable method for off line sound classification. But to use an ESN for sound recognition on an autonomous robot, unsupervised ongoing learning and classification is needed. A problem that is encountered with ongoing learning is the stability-plasticity dilemma. A method suitable for dealing with this problem is the earlier described MdARTMAP.
Instead of using linear read-out neurons or a linear classifier for an ESN, pattern recognition can be done by classifying each echo state with the Fuzzy ART network from the distributed clustering method. For this cochlear filtered sound data is first transferred into echo states which are then clustered separately and associated with the MdARTMAP.

Performance test for sound recognition with the MdARTMAP have been done for supervised and unsupervised learning. The difference with unsupervised learning is that the learning phase does not incorporate the distributed test to find the most likely class. It associates all the found echo state ART classes to the given sound class label. The overall performance of the system is very dependent of the parameters of the individual components which are related to each other. The parameters (see table 3.3) as well as the randomly generated reservoir topology are empirically determined.

Table 3.3: Parameters used for the echo state MdARTMAP sound recognition test.

Cochlear parameters	
step factor	0.6
decimation factor	130
ESN parameters	
input connectivity	0.2
input scale	3333
reservoir size	6.0
ART parameters	
vigilance	0.97
class probability threshold	0.5

With the parameters of table 3.3 a recognition score of 89% was achieved on the spoken word classification task, using a dataset with 250 samples of 5 english ”words” spoken by 50 different male speakers. The performance of the system with unsupervised classification was 70%. For these tests 10-fold cross validation with the same reservoir topology was used.

3.6 Visual Object Recognition

The second modality used in the cognitive sensor fusion architecture is vision. With sensor fusion the concept of an object must be formed using visual and auditory information. Before a visual object can be recognized it first has to be learned. Autonomously learning objects in a complex dynamical environment will need some guidance if there is minimal interaction with this environment and the objects in it. Getting to know an object in the real world is normally done by interaction, this could be for instance by touching it, picking it up, or by looking at it from different angles. This way all the different properties of the object can be learned. By interacting with the object a better segmentation of its properties and the properties of the surrounding can be established. When no interaction is possible with an object active sensing can be used to explore an object's visual properties. With active sensing an object is observed from different angles to get a more complete visual representation of an object. The current sensor fusion architecture will focus on the recognition of objects in a single camera image, which can obviously be extended to an active sensing system based on the control architecture. The first step in the object learning and recognition phase is the detection and segmentation of an object from its surrounding. After an object has been detected and extracted its characteristics are extracted, classified and associated to the concept that represents that object.

In the next sections the phases for visual object recognition from a camera image will be described. The first section will be about the detection and segmentation of an object, followed by the feature extraction method. In the last part a method for clustering and associating features will be described.

3.6.1 Detection and Segmentation

Extracting features and classifying each frame from a camera image is computationally expensive and biologically not realistic. An attentional system as described in previous sections is therefore needed to select relevant information for processing. This attentional system does not only point out the location of a visual interesting object but also highlights the most interesting parts via a saliency map. The saliency map points out a salient object which is inherently separated from its background. The saliency map is therefore used as segmentation method for objects in an image. The extracted salient location will be used for further processing which will be described in the next section.

3.6.2 SIFT Feature Extraction

One commonly used algorithm to detect and extract distinctive features from a visual object is Lowe's Scale-invariant feature transform (SIFT) [37]. The features extracted with SIFT are invariant to scale and rotation and partly invariant to illumination and 3D camera viewpoint. The SIFT features, called key-points, are highly distinctive and only a few features are needed to be able to recognize an object. Even when only a part of an image or object is visible, SIFT is able to recognize the object when the number of matching features is at least three. SIFT has proven to be a useful method for many image matching applications, to name a few, object recognition, robot localization and mapping, and 3D scene modelling. Because of its robustness and accuracy it is chosen to fulfil the visual object feature extraction task in the sensor fusion architecture.

The SIFT feature extracting process follows a cascading filtering approach, which means that operations that are expensive are only applied to locations that pass a test. A set of image features is created with the following process:

1. **Scale-space extrema detection:** First a Gaussian scale pyramid as described in section 3.1.1 is created to perform a Difference of Gaussian (DoG). To find interesting points every pixel in the DoG is compared to its eight neighbours and to eighteen neighbours in adjacent DoG levels.
2. **Keypoint Localization:** A detailed inspection is performed to see whether these interesting locations, keypoints, are stable. Keypoints are rejected when they have low contrast or are localized along an edge.
3. **Orientation assignment:** Based on local image gradient directions orientations are assigned to the keypoints. The keypoints are transformed relative to the scale and assigned orientations to make them invariant to these transformations.
4. **Keypoint descriptor:** A descriptor (figure 3.30) is computed for a local image region with the size determined by the scale at which the keypoint was detected. It is computed by calculating the gradient magnitude and orientation in the sub-regions of the region.

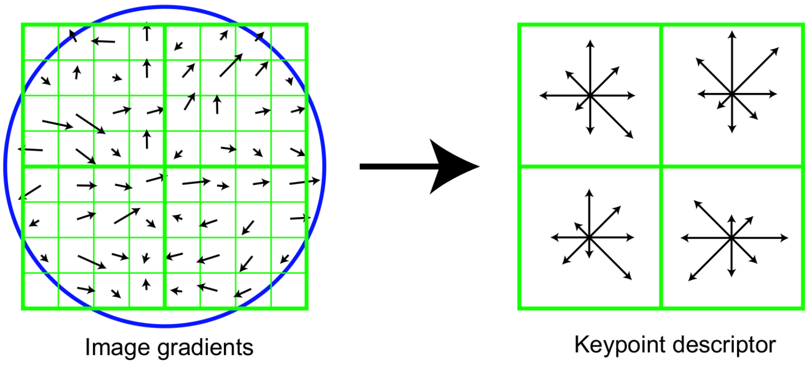


Figure 3.30: On the left the first step for computing the keypoint descriptor is shown. From the subregions around the keypoint the gradient magnitude and orientation are computed. The 2x2 subregions on the right contain orientation histograms which is a summation of the inner-subregions (taken from [37]).

3.6.3 The ART of 3D Object Recognition with SIFT Keypoint Clustering

Object recognition with SIFT is often done by matching the extracted keypoints to all the keypoints in a database using nearest neighbour classification. All the keypoints extracted from learned images are stored in the database. With a typical image of 500x500 pixels 2000 stable keypoints can be detected (dependent on the parameters and image content). These databases are therefore very large which leads to long search times and much storage consumption. To overcome these problems clustering of keypoints can be used to reduce the database size. Kootstra [35] showed that keypoint clustering can indeed be used to reduce the database size. The only drawback in Kootstra's implementation was the stability-plasticity dilemma. With the MdARTMAP keypoint clustering as well as matching can be performed without suffering from the stability-plasticity dilemma.

As with the echo states, individual keypoints are clustered and associated to a class based on the probability obtained via a test match using all the keypoints. If the probability of a class is high enough, then all the keypoints are associated to that class, else a new class is created and all the keypoints are associated to that class. With this method similar keypoints from different classes can be represented by a single cluster, which leads to a smaller database size. The dimension of the used keypoint descriptor is 128. These 128 values are used as input vector to the ART network. Due to the high dimensional input vector small changes in the vigilance parameter have a large effect on the amount of clusters that will be created. This effect is slightly minimized by first normalizing the input vector. Based on the SIFT parameters for keypoint filtering a vigilance parameter can be obtained empirically.

3.7 Summary

In this chapter the cognitive sensor fusion architecture consisting of a bi-modal attention and multi-modal sensor fusion module was described. The basis of the bi-modal attention module is a visual saliency detection system called SISCA. SISCA is an optimized version of the visual saliency detection systems of Itti et al. [28]. SISCA includes speed and performance optimizations through respectively a faster feature computation and an extended map suppression method. For auditory attention salient auditory features are extracted from a cochleogram through top-down cueing with SISCA. Based on salient audio the spatial location of a sound source is determined via binaural-localization. The mapping of the binaural cues to a spatial location is done by association the visual salient location of an object with the auditory cue information through Hebbian learning.

The presented multi-modal sensor fusion module is based on the multi-modal sensor fusion process found in the nervous system of a fruit-fly [55] and resembles the process found in the human medial temporal lobe [44]. The self-organizing associative processes found in these biological systems is the basis for the sensor fusion module. For the implementation of the associative memory a new type of ARTMAP called the Multi-directional ARTMAP (MdARTMAP) was presented. The MdARTMAP is based on the Default ARTMAP [6] and is extended with the possibility to have many-to-many associations, associated node retrieval in any direction, and un-supervised learning. The multi-modal sensor fusion module is created with a hierarchy of MdARTMAPs which enables shift invariant pattern recognition for the separate modalities through distributed clustering.

To use the proposed multi-modal sensor fusion module for the recognition and fusion of audio-visual information, feature extraction methods for these modalities have been presented. For the recognition of sound, features are extracted by processing cochlear filtered audio with an echo state network (ESN). The ESN transforms input to a higher dimension (feature space) which makes the recognition easier (linear separable). The echo states obtained from this process are clustered and associated using ″distributed clustering″ in the MdARTMAP. With ″distributed clustering″ all the echo states from one audio sample are clustered and associated separately to the same ″audio class″. The ″audio class″ is determined by calculating the probability whether this set of states belongs to a previously encountered ″audio class″ or not. If an ″audio class″ is found that has a high enough probability to belong to the echo states then all the states are associated to this class, otherwise a new ″audio class″ is created to which the echo states are associated. Experiments conducted with an ESN in combination with an MdARTMAP showed that it is a suitable method for un-supervised and on-going learning of sound.

For the recognition and fusion of visual objects with an MdARTMAP, SIFT [37] is used for image feature extraction. The features are extracted from the salient regions computed with SISCA. For object recognition these SIFT features, called keypoint descriptors, are clustered with an MdARTMAP using ″distributed clustering″. This means that each keypoint derived from the same object (region in an image) is clustered and associated separately to the same ″object class″. This ″object class″ is determined by calculating the probability whether this set of keypoints belongs to a previously encountered ″object class″ or not.

Chapter 4

Experiments

The previously proposed cognitive sensor fusion architecture is designed for modular micro robots (Replicators) [17] which are currently in development. To test the cognitive sensor fusion architecture the 3D simulator Symbricator (figure 4.1) is used. The whole cognitive sensor fusion architecture together with a simple control architecture are implemented in this simulator in C/C++. Different aspects of the implemented cognitive sensor fusion system are tested by conducing several experiments. In this chapter the experiments will be described.

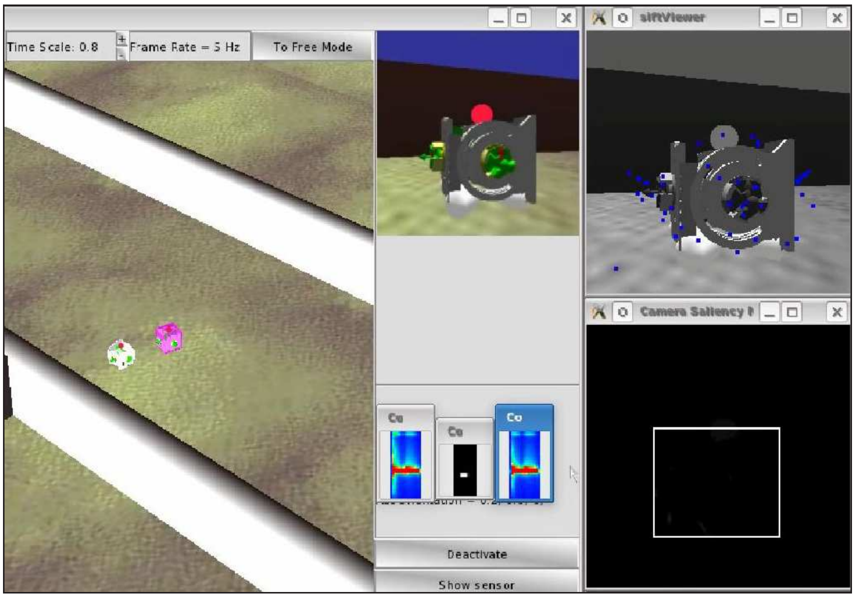


Figure 4.1: Symbricator3D interface. On the left there is the GUI with a birds eye view of the robots. In the middle and top right corner the camera view of the robot and the SIFT keypoints are displayed. A cochleogram with a saliency map is displayed below the center, and in the bottom right corner the salient region of the camera image displayed.

4.1 Simulator Implementation

The Symbricator3D simulator (figure 4.1) developed for the Replicators and Symbrion project is based on the open source gaming/simulation engine Delta3D [1]. This simulator consists of a 3D graphics toolkit: OpenSceneGraph; a physics engine: ODE; a skeletal animation: Cal3D; and a multichannel 3D positional audio library: OpenAL.

4.1.1 Robot

The robots used in the simulator have a cubical form and use two screw drives to translocate. They have unisex docking mechanism on four sides and a colour LED on top (see figure 4.2). The simulated sensors with which they are equipped are:

- Color Camera
- Depth camera
- Distance sensors
- Laser scanner
- Light sensor
- Stereo microphone

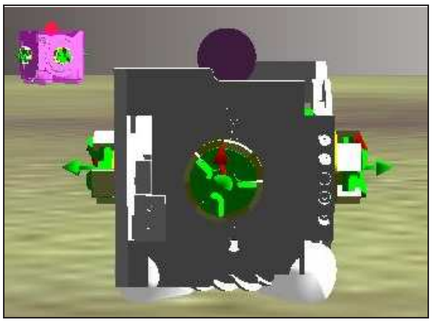


Figure 4.2: The simulated robot viewed from the front.

4.1.2 Implementation

To conduct the experiments the robot must be able to perform actions based on its perceptions and goals. For this a simple subsumption control architecture is used. This control architecture consists of:

1. Wander
2. Goal tracking
3. Obstacle avoidance
4. Docking
5. Object learning

1 Wander

On the lowest level of the control architecture is a wander module. This wander module lets the robot drive through the environment with some randomness. It controls the robot by setting a speed value for the two screw drives.

2 Goal tracking

The goal tracking module controls the main actions of the robot. With goal tracking the robot needs to go to interesting objects that are known or unknown to the robot.

2.1 Bi-modal saliency Goal tracking is done by first performing bi-modal saliency detection on the camera image and on a sound recording of 0.5 seconds from the microphone. If both modalities have salient data, then based on the predicted spatial locations of the visual object and sound source a classification is performed.

2.2 Perception alignment If both modalities perceive an object from the same location then both percepts are used as input to classify the same object. Otherwise each modality is used separately to predict the perceived object.

2.3 Target selection The robot only changes its heading to a percept if it is classified as the goal or as unknown. Objects in front of the robot have a higher priority and goal objects have the highest priority.

2.4 Motor control To be able to drive to a salient object the robot first learns to center a salient object in a reinforcement learning task. In this task a mapping is learned between the spatial location (state) and the direct effect of the speed values for the motors (action). The weight value between the motor values and the spatial location is increased if the speed value brought the salient location closer to the center of the camera image (goal), otherwise it is decreased.

3 Obstacle avoidance

Using its distance sensors the robot is able to avoid obstacles that are not classified as unknown or as goal objects. These distance sensors are simulated sonar sensors. The robot turns away from an obstacle when the distance is below a certain threshold.

4 Docking

The docking action is used to determine the energy properties of an object. This action is initiated when the robot is closely in front of a goal or unknown object. This docking mechanism is simulated and will return the value ”no energy property” if it is not able to dock.

5 Object learning

The robot learns the properties of an object through association after it has performed the docking action. It associates the visual and auditory representation together with the energy property of the perceived object.

4.2 Scenario

A scenario which will be used to test cognitive sensor fusion is visual-acoustic object recognition. This scenario will be tested using the 3D simulator Symbricator (figure 4.1). In this scenario the robot must be able to distinguish other robots from other objects based on low quality sound and camera images. The robot first learns to associate different sounds and objects by driving through the scene (self-learning). By using cognitive sensor fusion with different modalities the robot must be able to detect objects earlier, and recognize objects better. If the robot is searching for a particular object, then if it hears a sound, it has to know what object, in the sense of the object's properties (e.g. visual representation), are associated with it.

4.3 Task Description

There are two important search tasks that the micro robots must be able to perform, these are: finding other robots to dock to and exchange energy with, and finding power outlets to power up. The robots are equipped with a docking mechanism with which they can dock to power outlets or others to obtain or exchange energy. The performance of the cognitive sensor fusion module will be determined by measuring how well a search task is done in terms of successful trials and classification errors. The first task is to find energy, this can be a power outlet or another robot that also has energy. Robots that are running out of energy can be distinguished from robots that do have energy by the sound that they make. Robots that are almost out of energy make a beeping sound and switch of their red light, all other robots emit sound which is created physically through their motor and screw drive.

4.4 Experiment Setup

While the robot performs its search task it simultaneously learns: what objects are in the room, what kind of sound do they make, and whether the objects give, need, or don't use energy (through the docking mechanism). While the robot drives through the room it learns the objects that it encounters based on the used modalities (vision and sound). The robot also stores the energy property that an object has, this can be: ″energy sink″, ″energy source″, or ″no energy property″. A room will consist of a power outlet and one other robot that needs energy from time to time.

The robot starts with the task to power up. It tries to find power outlets or robots that have power. Once it has obtained power it wanders around and if it notices a robot without energy it tries to provide the empty robot with energy. When this succeeds the robot notices that the other robot has energy and proceeds its search. The other robot is always stationary even after it received energy. After a while the energy of both robots drop below a threshold. This happens for both robots at the same time. The stationary robot will then make a beeping sound, while the other robot tries to look for an energy source. Looking for an energy source can take as long as the duration of the experiment.

4.4.1 Conditions

In the experiment several properties will be tested. The added value of the sensor fusion module, and in particular the use of sound as second modality, is tested by performing the search tasks with and without audio. It is expected that without audio the time needed to find the other robot will increase, and also decrease the recognition performance. To determine the added value of the attention system, a condition is used in which the saliency module is not used (lesion experiment). The expected result is that it will take more time to find an energy source or sink when there is no attention mechanism. To test these properties four experiments are conducted with different conditions. Each experiment is conducted at least ten times for each condition and each experiment condition lasts thirty minutes.

During the experiments the amount of times the following actions are performed is counted:

1. Obtained energy from an outlet
2. Gave energy to the other robot
3. Try to give energy to a wrong (misclassified) object
4. Try to obtain energy from a wrong (misclassified) object

Experiment 1: Binaural-Localization

The added value of binaural-localization to the attention module is tested by using a large partly observable room (figure 4.3). The room consist of three parts, which all contain an outlet (figure 4.4). Two search task are performed, one with auditory and visual input, and one with visual input only. Because the other robot is not always visible, the added value of the binaural-localization module can be measured.

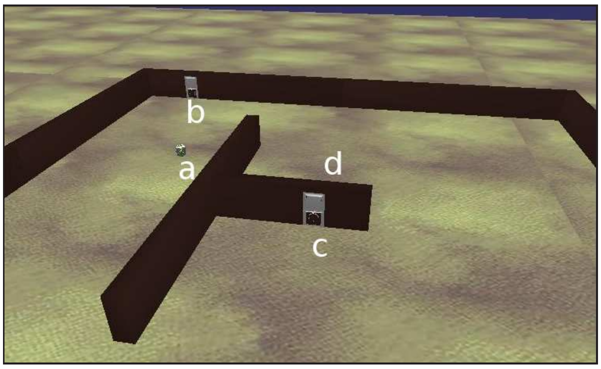


Figure 4.3: The room setup for experiment 1. In this figure (a) indicates the location of the empty robot, (b), (c), and (d) indicate the location of the outlets.

Figure 4.4: A picture of the outlet used in the experiment.

Experiment 2: Sensor Fusion

To test the sensor fusion architecture for its pattern recognition capabilities a single fully observable room with one outlet and robot is used as experiment environment (figure 4.5). In this task the emphasis does not lie on the searching performance but on making the right choice based on the classification. This task is also performed with two conditions, one time with auditory and visual input and one time with only visual input. In both cases the robot should be able to find the outlet and the other robot easily.

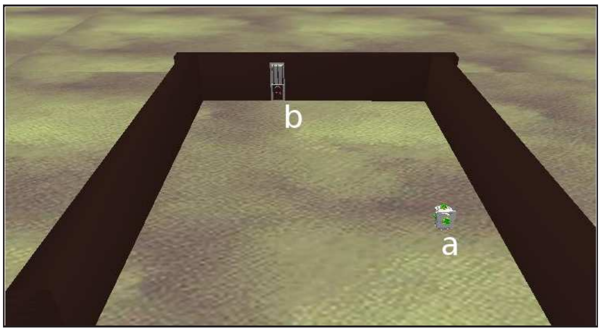


Figure 4.5: The room setup for experiment 2. In this figure (a) indicates the location of the empty robot and (b) location of the outlet.

Experiment 3: Sensor Fusion with Visual Distraction

A third experiment is used to test how well the sensor fusion system performs when there is another object in the room with visual distracting features. To test this an object (figure 4.7) was placed on the wall in the room (figure 4.6). This experiment was also conducted with two conditions, with auditory and visual information and with only visual information. For this experiment the amount of times the robot obtains and gives energy is measured together with the amount of times the robot goes to the distracting object.

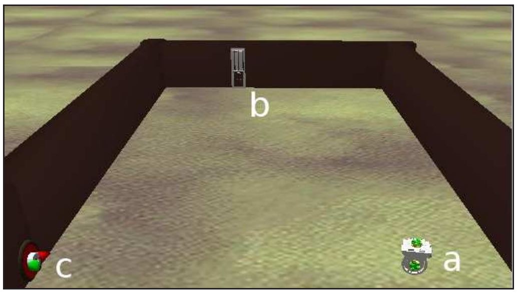


Figure 4.6: The room setup for experiment 3. In this figure (a) indicates the location of the empty robot, (b) location of the outlet, and (c) the location of the distracting object.

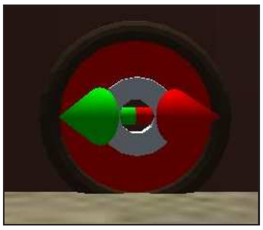


Figure 4.7: A picture of the distracting object.

Experiment 4: Bi-modal Attention

The added value of the attention module is tested by performing an experiment with and without the attentional mechanism. This is done by skipping the bi-modal saliency detection module, this causes the system to classify each camera image or sound sample. For this the same setup as in experiment 2 is used.

Chapter 5

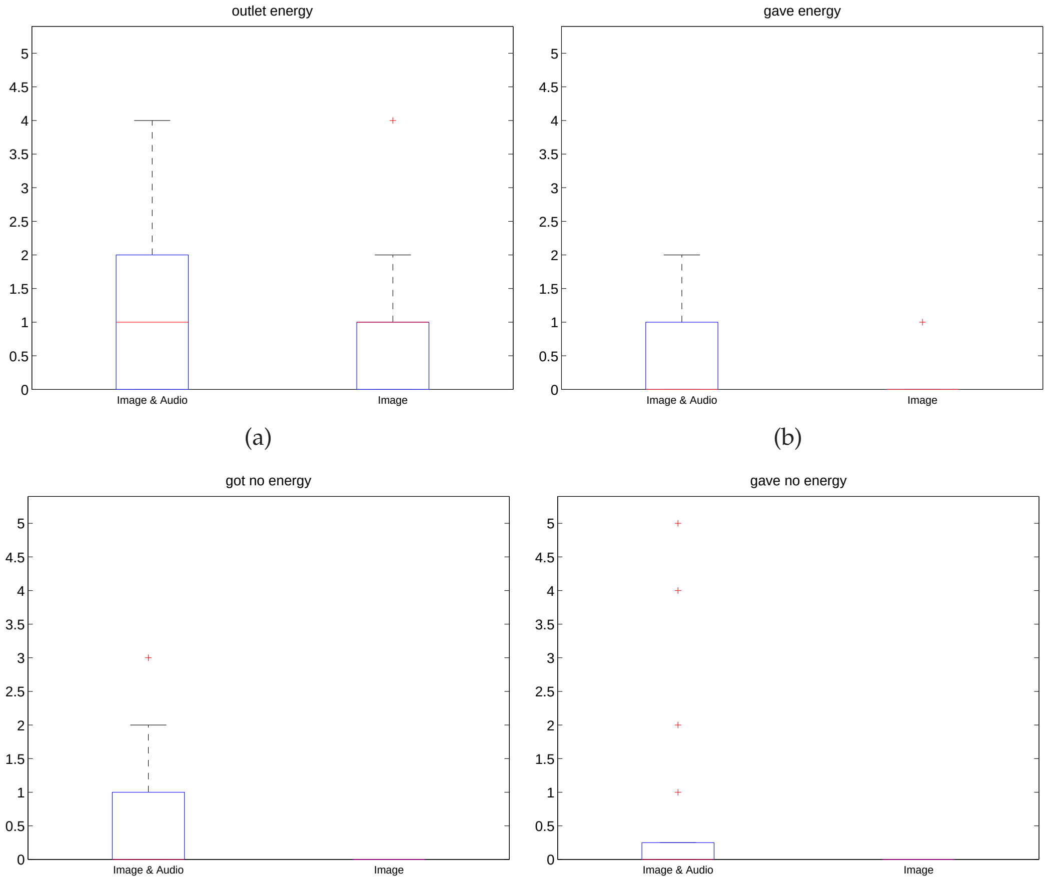
Results

5.1 Results of Experiment 1: Binaural Localization

In experiment 1 as described in section 4.4.1 binaural localization is tested by conducting an experiment where the robot needs to obtain energy and give it to another robot in a large environment. For this experiment a condition is used where auditory and visual information is used and a condition with only auditory information. The results are shown in table 5.1 and figure 5.1. In the table and figure the measured actions are indicated as *outlet energy* for the retrieval of energy from an outlet, *gave energy* for giving another robot energy, *gave no energy* when the robot tried to give energy to a wrong object, and *got no energy* when the robot tried to obtain energy from a wrong object. From the results it can be seen that the condition where both modalities are used scored higher on all the measured actions, inclusive on the wrong type of actions caused by false classifications. The significance of the difference between the actions is tested using a two-tailed Student's T-test with significance value $a = 0.05$ and null hypothesis: using audio as second modality does not have an effect on the measured property. Using audio did not lead to a significant difference in finding an outlet ($p = 0.35 > 0.05$), nor to a significant difference in trying to obtain energy from a wrong object ($p = 0.08 > 0.05$). A significant difference was found in finding and providing the other robot with energy ($p = 0.04 < 0.05$) and trying to give a wrong object energy ($p = 0.04 < 0.05$).

Table 5.1: Results for experiment 1: Binaural Localization.

conditions	outlet energy	gave energy	got no energy	gave no energy
Image & Audio	$\mu = 1.18$ $\sigma = 1.07$	$\mu = 0.65$ $\sigma = 0.79$	$\mu = 0.71$ $\sigma = 1.53$	$\mu = 0.47$ $\sigma = 0.87$
Image	$\mu = 0.82$ $\sigma = 1.07$	$\mu = 0.18$ $\sigma = 0.39$	$\mu = 0$ $\sigma = 0$	$\mu = 0$ $\sigma = 0$
H_0: no difference	$p = 0.35$	$p = 0.04$	$p = 0.08$	$p = 0.04$



(a)

(b)

(c)

(d)

Figure 5.1: Results of experiment 1: Binaural Localization. The graphs show the results for using both visual and auditory information and for using only visual information. (a) shows the amount of times the robot powered up at the outlet. (b) shows the amount of times energy was given to the other robot. (c) shows the amount of times the robot tried to obtain energy from a wrong object. (d) shows the amount of times the robot tried to give energy to a wrong object.

5.2 Results of Experiment 2: Sensor Fusion

The performance of the sensor fusion system was tested as described in section 4.4.1 by using a smaller single room. The results are shown in table 5.2 and figure 5.2. Also in this experiment the condition where both modalities are used scored higher on all the measured actions. The statistical significance of these differences is tested with a two-tailed Student't T-test with significance value $a = 0.05$ and null hypothesis: using audio as second modality does not have an effect on the measured property. A significant difference was measured for both the amount of times the outlet was found ($p = 0.01 < 0.05$) and the amount of times the robot was found and provided with energy ($p = 0.01 < 0.05$). No significant difference was found in the amount of misclassifications of objects where the robot tried to obtain energy from ($p = 0.53 > 0.05$) or give energy to ($p = 0.34 > 0.05$).

Table 5.2: Results for experiment 2: Sensor Fusion.

conditions	outlet energy	gave energy	got no energy	gave no energy
Image & Audio	$\mu = 3\ \sigma = 0.94$	$\mu = 2.2\ \sigma = 1.14$	$\mu = 0.6\ \sigma = 0.7$	$\mu = 0.1\ \sigma = 0.32$
Image	$\mu = 1.3\ \sigma = 1.57$	$\mu = 0.8\ \sigma = 0.92$	$\mu = 0.4\ \sigma = 0.7$	$\mu = 0\ \sigma = 0$
H_0: no difference	$p = 0.01$	$p = 0.01$	$p = 0.53$	$p = 0.34$

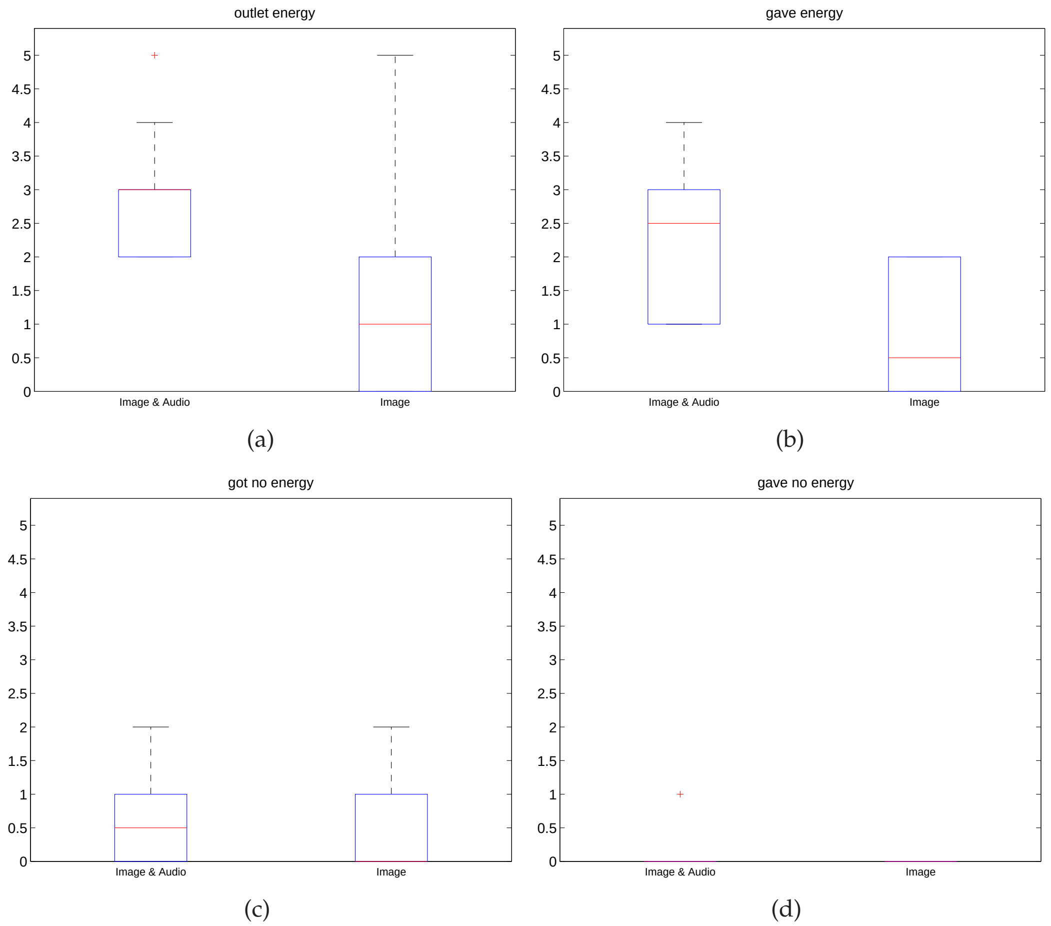


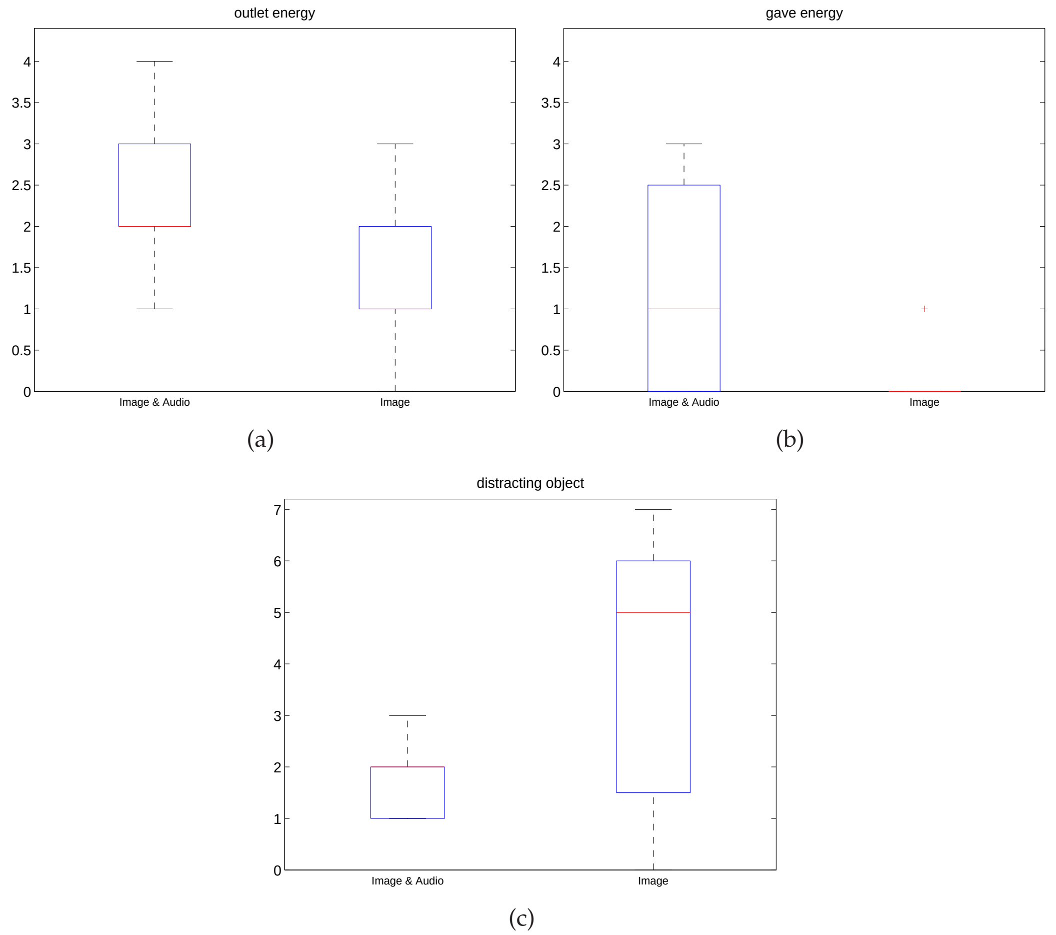
Figure 5.2: Results of experiment 2: Sensor Fusion. The graphs show the results for using both visual and auditory information and for using only visual information. (a) shows the amount of times the robot powered up at the outlet. (b) shows the amount of times energy was given to the other robot. (c) shows the amount of times the robot tried to obtain energy from a wrong object. (d) shows the amount of times the robot tried to give energy to a wrong object.

5.3 Results of Experiment 3: Sensor Fusion with Visual Distraction

Another test was conducted to test the robustness of the sensor fusion system as described in section 4.4.1. Table 5.3 and figure 5.3 show the results of this experiment. From the results it is clear to see that the condition in which both sound and visual information is used has a higher value for obtaining and providing energy, and a lower value for the amount of times the distracting object was visited. With a two-tailed Student T-test the significance of these differences is tested with significance value $a = 0.05$, and the null hypothesis: that the addition of audio as second modality does not have an effect on the measured properties. A significant difference was measured for all the values, the amount of times the outlet was found ($p = 0.03 < 0.05$), the amount of times the other robot was provided with energy ($p = 0.03 < 0.05$), and the amount of times the distracting object was visited ($p = 0.01 < 0.05$).

Table 5.3: Results for experiment 3: Sensor Fusion with Visual Distraction.

conditions	outlet energy	gave energy	distracting obj
Image & Audio	$\mu = 2.45$ $\sigma = 1.04$	$\mu = 1.09$ $\sigma = 1.3$	$\mu = 1.73$ $\sigma = 0.65$
Image	$\mu = 1.45$ $\sigma = 1.04$	$\mu = 0.09$ $\sigma = 0.3$	$\mu = 4$ $\sigma = 2.41$
H_0: no difference	$p = 0.03$	$p = 0.03$	$p = 0.01$



(a)

(b)

(c)

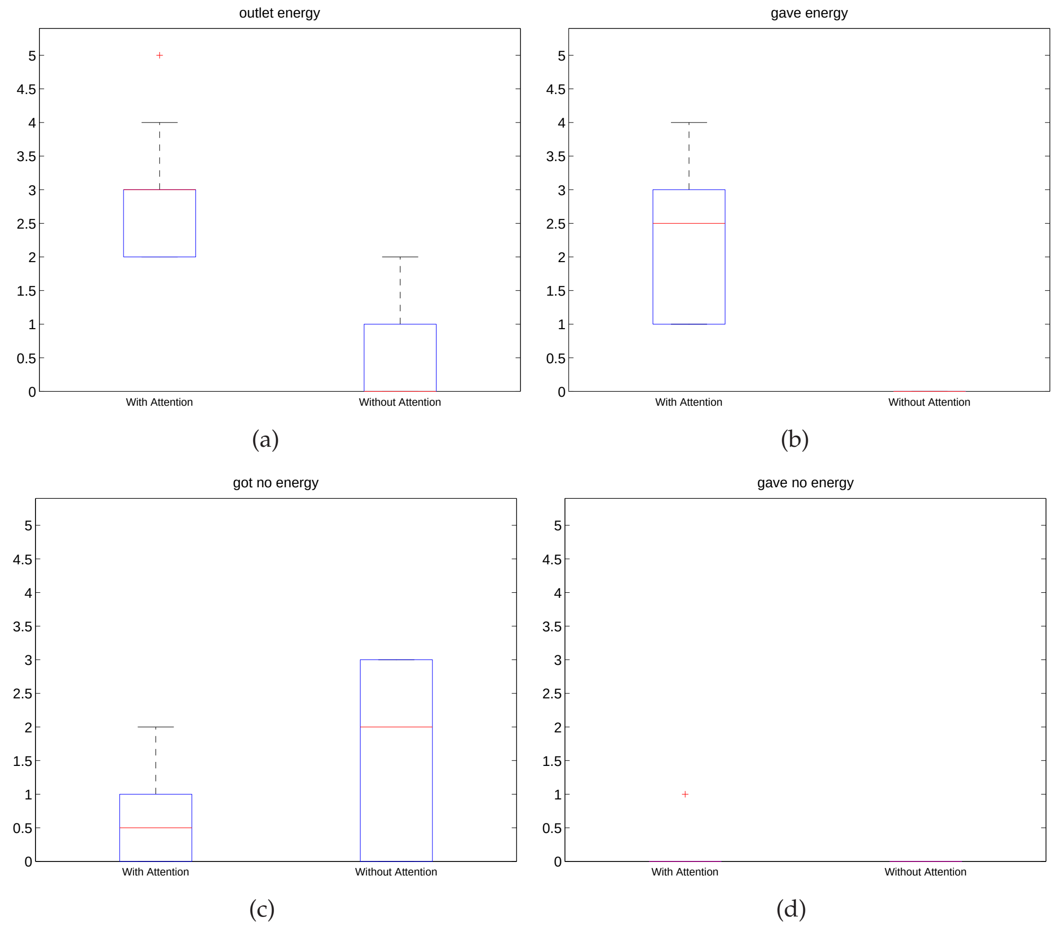
Figure 5.3: Results of experiment 3: Sensor Fusion with visual distraction. The graphs show the results for using both visual and auditory information and for using only visual information. (a) shows the amount of times the robot powered up at the outlet. (b) shows the amount of times energy was given to the other robot. (c) shows the amount of times the robot went to the distracting object.

5.4 Results of Experiment 4: Bi-modal Attention

The added value of the bi-modal attention module for the sensor fusion architecture is also tested as described in section 4.4.1. The same setup as with experiment 2 was used, but now without using the bi-modal attention module. The results of this condition are compared to the results of the experiment 2. The results of the experiment and a comparison with experiment 2 are shown in table 5.4 and figure 5.4. Again using a two-tailed Student's T-test with significance value $a = 0.05$ the significance of the difference is measured with null hypothesis: the addition of bi-modal attention module does not have an effect on the measured values. The Student's T-test showed that with the use of bi-modal attention the amount of times the robot got to the outlet is significantly higher ($p < 0.01 < 0.05$), and the amount of times the robot provided the other robot with energy is higher ($p < 0.01 < 0.05$). A significant difference in the amount of misclassifications was not found.

Table 5.4: Results for experiment 4: Bi-modal Attention.

conditions	outlet energy	gave energy	got no energy	gave no energy
With attention	$\mu = 3$ $\sigma = 0.94$	$\mu = 2.2$ $\sigma = 1.14$	$\mu = 0.6$ $\sigma = 0.7$	$\mu = 0.1$ $\sigma = 0.32$
Without attention	$\mu = 0.5$ $\sigma = 0.71$	$\mu = 0$ $\sigma = 0$	$\mu = 1.6$ $\sigma = 1.26$	$\mu = 0$ $\sigma = 0$
H_0: no difference	$p < 0.01$	$p < 0.01$	$p = 0.05$	$p = 0.34$



(a)

(b)

(c)

(d)

Figure 5.4: Results of experiment 4: Bi-modal Attention. The graphs show the results for the sensor fusion experiment with the use of the attention module and without the attention module. (a) shows the amount of times the robot powered up at the outlet. (b) shows the amount of times energy was given to the other robot. (c) shows the amount of times the robot tried to obtain energy from a wrong object. (d) shows the amount of times the robot tried to give energy to a wrong object.

Chapter 6

Discussion

6.1 Summary of the Results

In the first experiment the added value of the binaural-localization method was tested. An environment which consisted of three rooms was used so that the robot needed to search for its goal. The results showed that the addition of audio, used for binaural-localization, had a positive effect on the search task. As expected the addition of audio only had an effect on finding the robot, which emits sound, and not on finding the outlet which does not emit sound. Because the robot was drawn more often to the sound emitting robot the number of times it tried to give energy to that robot while it was no empty is also higher. From the results it can be concluded that the binaural-localization method has an added value to the cognitive sensor fusion architecture when it comes to finding an object that emits sound. To see whether sound as second modality only helps in finding the other robot or also results in better classifications results, a second experiment was conducted.

In the second experiment the sensor fusion system was tested by conducting the experiment in a smaller room where the robot did not need to search for its goal but merely make a good classification of the perceived objects. The results from this experiment show that with the addition of audio the other robot was more often successfully classified and given energy. Also the amount of times energy was obtained from the outlet was higher with the addition of audio. The reason why energy was also obtained more times is due the fact that when energy is given the robot's own energy is lowered, and therefore has to search for energy again. Another possible reason was observed during the experiments. When the robot perceived a salient object it could classify the object based on sound on a larger distance than with visual information, and when an object is classified earlier the robot loses less time on an uninteresting object.

In the third experiment a visual distracting object was added to the setup of the second experiment to test the robustness of the sensor fusion system. In this experiment the amount of times the robot went to the distracting object was counted instead of the amount of times it wanted to give or obtain energy from a wrongly classified object. The results from this experiment show that with the addition of audio the robot obtained energy more times, gave energy more times, and went fewer times to the distracting object than with vision as only modality. These results show even clearer than the second experiment that the early classification based on audio causes the robot to stay focussed on its goal instead of being distracted by even more salient objects.

The fourth experiment was conducted to see whether the saliency detection module was actually needed or that the robot could perform the task with all information equally salient. From the results of this experiment it can be concluded that the robot was not able to perform its task without the saliency detection module. One of the reasons for this result could be due to the fact that the robot is only able to classify objects from a certain distance. When no saliency detection module is used the robot has no need to drive to an object and will only make a successful classification when it happens to end up in-front of an object. Another aspect that could play a role in this result is the possible bad representation of a system without an attentional mechanism. Because the saliency detection module was an integral part of the sensor fusion system, leaving this module out could give a worse performance than a different sensor fusion system that was built not to have an attentional mechanism at all.

6.2 Conclusion

In this thesis a biologically inspired cognitive sensor fusion architecture was presented for the micro-robots developed in the Replicator project [17]. The goal for this architecture was to obtain environmental awareness through self-organization. By using several modalities these robots need to be able to detect and recognize interesting objects in the environment. However due to the limited processing capabilities of the individual micro robots, sensor information has to be processed efficiently. To find an efficient way for sensor processing, biological systems are examined for their sensor fusion capabilities. This has led to the development of a biologically inspired cognitive sensor fusion architecture which consists of a bi-modal attention module and multi-modal self-organizing associative memory.

State of the art visual saliency detection mechanisms [33] [18] were altered and combined with biologically based sensor processing and fusion methods [55] [25] to obtain the bi-modal attention module. The bi-modal attention module consists of audio-visual saliency detection with binaural-localization.

For the sensor fusion module a new type of ARTMAP (self-organizing associative memory) called the Multi-directional ARTMAP (MdARTMAP) was developed. This ARTMAP is used in a hierarchical manner to obtain shift invariant pattern recognition through distributed clustering. The MdARTMAP was used to cluster states from an echo state network (ESN) which used cochlear filtered audio as input in order to recognize sounds. Experiments conducted to test the sound recognition module showed that this new method is very suitable for un-supervised on-going learning of sound. Object recognition was obtained by clustering and associating SIFT keypoint descriptors with an MdARTMAP. A multi-modal sensor fusion module was eventually created by associating the MdARTMAPs of the auditory and visual object recognizers with a higher level MdARTMAP.

Experiments were conducted to test the performance of the cognitive sensor fusion system and its parts. The robot with its cognitive sensor fusion architecture was implemented in a 3D simulator in which the experiments where conducted. The results showed that the robot was able to successfully perform search tasks with the cognitive sensor fusion architecture. The robot performed significantly better on its tasks when fusing auditory and visual information than when only visual information was available. A significant better performance was also found when the bi-modal attention module was used than when only multi-modal sensor fusion was used.

With these results an answer is given to the earlier presented research question:

How can biologically inspired sensor fusion be used in an embodied self-organizing micro-system to increase environmental awareness?

The self-organizing process that underlies the associative memory, which is used as basis for multi-modal sensor fusion, has shown to be useful for on-going learning. The associative network was able to successfully re-associate learned features to multiple ”classes” when a change occurred in the environment (e.g. the association of features to a ”has energy” robot class and the partly re-association of features to the ”empty robot” class).

6.3 Future Work

The presented cognitive sensor fusion architecture is designed for modular micro-robots that can operate individually but also as a large organism consisting of multiple micro-robots. To be able to operate as one organism sensor information from all the micro-robots need to be shared, processed and fused. Further research needs to be conducted to find out what kind of changes need to be made to use this architecture for distributed cognitive sensor fusion.

Concerning the computational complexity of the system, an increase in the speed performance can be gained on at least one computationally demanding module, visual object recognition. The implemented SIFT module is for its good recognition performance and source code availability chosen as image feature extraction method. But a significant speed performance gain can be obtained when keypoints in SIFT are not searched for with the expensive *scale-space extrema detection* method of SIFT (see section 3.7.2) but instead are provided by the saliency detection module SISCA (see section 3.1).

Bibliography

[1] Delta3d. Available from: `http://www.delta3d.org/`.

[2] AMIS, G., AND CARPENTER, G. Default artmap 2. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'07)* (2007), pp. 777–782.

[3] ANTONELO, E., SCHRAUWEN, B., DUTOIT, X., STROOBANDT, D., AND NUTTIN, M. Event detection and localization in mobile robot navigation using reservoir computing. *Lecture Notes in Computer Science 4669* (2007), 660.

[4] BURGSTEINER, H. Training networks of biological realistic spiking neurons for real-time robot control. In *Proceedings of the 9th International Conference on Engineering Applications of Neural Networks* (2005), pp. 129–136.

[5] BUTZ, M., AND RAY, S. Bidirectional artmap: an artificial mirror neuron system. In *Neural Networks, 2003. Proceedings of the International Joint Conference on* (2003), vol. 2.

[6] CARPENTER, G. Default artmap. In *Proceedings of the international joint conference on neural networks (IJCNN'03)* (2003), pp. 1396–1401.

[7] CARPENTER, G., GROSSBERG, S., AND ROSEN, D. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks 4*, 6 (1991), 759–771. Available from: `http://dx.doi.org/10.1016/0893-6080(91)90056-B`.

[8] CARPENTER, G. A., AND GROSSBERG, S. Art 2: self-organization of stable category recognition codes for analog input patterns. *Applied Optics 26* (Dec. 1987), 4919–4930.

[9] CARPENTER, G. A., GROSSBERG, S., MARKUZON, N., REYNOLDS, J. H., AND ROSEN, D. B. Fuzzy artmap: A neural network architecture for incremental supervised learning of analog multidimensional maps. *Neural Networks, IEEE Transactions on 3*, 5 (1992), 698–713. Available from: `http://dx.doi.org/10.1109/72.159059`.

[10] CARPENTER, G. A., GROSSBERG, S., AND REYNOLDS, J. H. Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks 4*, 5 (1991), 565–588.

[11] CARPENTER, G. A., GROSSBERG, S., AND ROSEN, D. B. Art 2-a: an adaptive resonance algorithm for rapid category learning and recognition. *Neural Netw. 4*, 4 (1991), 493–504.

[12] Carpenter, G. A., and Markuzon, N. Artmap-ic and medical diagnosis: Instance counting and inconsistent cases. *Neural Networks 11*, 2 (1998), 323–336.

[13] Carpenter, G. A., Milenova, B. L., and Noeske, B. W. Distributed artmap: a neural network for fast distributed supervised learning. *Neural Networks 11*, 5 (1998), 793–813.

[14] Carpenter, G. A., and Ross, W. Art-emap: A neural network architecture for learning and prediction by evidence accumulation. *IEEE Transactions on Neural Networks 6*, 4 (July 1995), 805–818.

[15] Carpenter, G.A. & Grossberg, S. Adaptive resonance theory. *Michael A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, Second Edition* (2002), (pp. 87–90).

[16] Chevallier, S., and Tarroux, P. Covert attention with a spiking neural network. *Lecture Notes in Computer Science 5008* (2008), 56.

[17] EU. Fp7 ga no.216240 replicator. Tech. rep. Available from: `http://replicators.eu/`.

[18] Frintrop, S. Vocus: A visual attention system for object detection and goal-directed search. *Lecture Notes in Artificial Intelligence (LNAI) Vol. 3899* (2006).

[19] Frintrop, S., Klodt, M., and Rome, E. A real-time visual attention system using integral images. In *Proceedings of the ICVS* (2007), Citeseer.

[20] Frintrop, S., Nuchter, A., Surmann, H., and Hertzberg, J. Saliency-based object recognition in 3d data. Isle of Skye, Scotland.

[21] Frintrop, S., and Rome, E. Simulating visual attention for object recognition. In *Proceedings of the Workshop on Early Cognitive Vision* (2004), Isle of Skye, Scotland.

[22] Frintrop, S., Rome, E., Nuchter, A., and Surmann, H. A bimodal laser-based attention system. *Computer Vision and Image Understanding 100*, 1-2 (2005), 124–151.

[23] Hillenbrand, J., Getty, L., Clark, M., and Wheeler, K. Acoustic characteristics of american english vowels. *Journal of the Acoustical Society of America 97*, 5 (1995), 3099–3111.

[24] Holzmann, G. Echo state networks in audio processing. *Internet Publication* (2007). Available from: `http://grh.mur.at/sites/default/files/ESNinAudioProcessing.pdf`.

[25] Huo, J., and Murray, A. The adaptation of visual and auditory integration in the barn owl superior colliculus with spike timing dependent plasticity. *Neural networks : the official journal of the International Neural Network Society* (November 2008). Available from: `http://dx.doi.org/10.1016/j.neunet.2008.10.007`.

[26] Itti, L., and Koch, C. Computational modelling of visual attention. *Nature Reviews Neuroscience 2*, 3 (March 2001), 194–203. Available from: `http://dx.doi.org/10.1038/35058500`.

[27] ITTI, L., AND KOCH, C. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging 10*, 1 (January 2001), 161–169. Available from: `http://spiedl.aip.org/journals/doc/JEIME5-ft/vol_10/iss_1/161_1.html#F7`.

[28] ITTI, L., KOCH, C., AND NIEBUR, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (1998), 1254–1259. Available from: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.2366`.

[29] JAEGER, H. The "echo state" approach to analysing and training recurrent neural networks. GMD Report 148, GMD - German National Research Institute for Computer Science, 2001. Available from: `http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf`.

[30] JAEGER, H. Tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "echo state network" approach. Tech. rep., Fraunhofer Institute AIS, St. Augustin-Germany, 2002.

[31] KANDEL, E. R. *Principles of Neural Science*. McGraw-Hill Education, June 2000. Available from: `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0071120009`.

[32] KAYSER, C., PETKOV, C. I., LIPPERT, M., AND LOGOTHETIS, N. K. Mechanisms for allocating auditory attention: An auditory saliency map. *Current Biology 15*, 21 (November 2005), 1943–1947. Available from: `http://dx.doi.org/10.1016/j.cub.2005.09.040`.

[33] KOCH, C., AND ULLMAN, S. Shifts in selective visual attention: towards the underlying neural circuitry. *Hum Neurobiol 4*, 4 (1985), 219–227. Available from: `http://view.ncbi.nlm.nih.gov/pubmed/3836989`.

[34] KONUSKAN, F. Visual saliency and biological inspired text detection.

[35] KOOTSTRA, G., YPMA, J., AND DE BOER, B. Active exploration and keypoint clustering for object recognition. In *IEEE International Conference on Robotics and Automation (ICRA)* (2008), pp. 1005–1010.

[36] LIENHART, R., AND MAYDT, J. An extended set of haar-like features for rapid object detection. In *IEEE ICIP* (2002), vol. 1, Citeseer, pp. 900–903.

[37] LOWE, D. Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60*, 2 (2004), 91–110.

[38] LYON, R. Automatic gain control in cochlear mechanics. *The Mechanics and Biophysics of Hearing* (1991).

[39] MAASS, W., NATSCHLAGER, T., AND MARKRAM, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput 14*, 11 (2002), 2531–60. Institute for Theoretical Computer Science, Technische Universitat Graz, A-8010 Graz, Austria. maass@igi.tu-graz.ac.at.

[40] MAASS, W., NATSCHLAGER, T., AND MARKRAM, H. A Model for Real-Time Computation in Generic Neural Microcircuits. In *NIPS 2002* (2003), Advances in Neural Information Processing Systems, MIT Press, pp. 229–236.

[41] MORSE, A., AND AKTIUS, M. Dynamic liquid association: Complex learning without implausible guidance. *Neural Networks 22*, 7 (2009), 875–889.

[42] PAPLINSKI, A., AND GUSTAFSSON, L. Multimodal feedforward self-organizing maps. *Lecture Notes in Computer Science 3801* (2005), 81.

[43] PAPLINSKI, A. P., AND GUSTAFSSON, L. Feedback in multimodal self-organizing networks enhances perception of corrupted stimuli. *Lecture Notes in Computer Science 4304* (2006), 19–28.

[44] QUIAN QUIROGA, R., KRASKOV, A., KOCH, C., AND FRIED, I. Explicit encoding of multimodal percepts by single neurons in the human brain. *Current Biology* (July 2009). Available from: `http://dx.doi.org/10.1016/j.cub.2009.06.060`.

[45] RODIECK, R. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Res 5*, 11 (1965), 583–601.

[46] SCHRAUWEN, B., VERSTRAETEN, D., AND CAMPENHOUT, J. V. An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th European Symposium on Artificial Neural Networks* (2007), pp. 471–482.

[47] SKOWRONSKI, M. D., AND HARRIS, J. G. Automatic speech recognition using a predictive echo state network classifier. *Neural Networks 20*, 3 (April 2007), 414–423. Available from: `http://dx.doi.org/10.1016/j.neunet.2007.04.006`.

[48] SLANEY, M. Lyon's cochlear model. Apple technical report 13, Apple mComputer, Inc., Corporate Library, One Infinite Loop, Cupertino, CA 95104, 1988.

[49] STEIL, J. J. Backpropagation-decorrelation: online recurrent learning with o(n) complexity. In *Proc. IJCNN* (Jul 2004), vol. 1, pp. 843–848.

[50] VANRULLEN, R. Visual saliency and spike timing in the ventral visual pathway. *J Physiol Paris 97*, 2-3 (Mar-May 2003), 365–377. Available from: `http://www.hubmed.org/display.cgi?uids=14766152`.

[51] VENAYAGAMOORTHY, G. K., AND SHISHIR, B. Effects of spectral radius and settling time in the performance of echo state networks. *Neural Networks 22*, 7 (2009), 861–863.

[52] VERSTRAETEN, D., SCHRAUWEN, B., D'HAENE, M., AND STROOBANDT, D. An experimental unification of reservoir computing methods. *Neural Networks 20*, 3 (2007), 391–403. Available from: `http://dblp.uni-trier.de/db/journals/nn/nn20.html#VerstraetenSDS07`.

[53] VERSTRATEN, D., SCHRAUWEN, B., STROOBANDT, D., AND VAN CAMPENHOUT, J. Isolated word recognition with the liquid state machine: a case study. *Information Processing Letters 95*, 6 (2005), 521–528.

[54] VIOLA, P., AND JONES, M. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 1* (April 2001), 511–I–518 vol.1. Available from: `http://dx.doi.org/10.1109/CVPR.2001.990517`.

[55] WESSNITZER, J., AND WEBB, B. Multimodal sensory integration in insects:towards insect brain control architectures. *Bioinspiration & Biomimetics 1*, 3 (2006), 63–75. Available from: `http://dx.doi.org/10.1088/1748-3182/1/3/001`.

[56] WESSNITZER, J., AND WEBB, B. A neural model of cross-modal association in insects. In *Proceedings of the European Symposium on Artificial Neural Networks"(M. Verleysen, Ed.)* (2007), pp. 415–421.