AISB 2011

Architectures for Active Vision

Editors:
Dimitar Kazakov &
George Tsoulas

THE UNIVERSITY of York

Foreword from the Convention Chairs

The AISB'11 call for symposium proposals particularly encouraged events drawing more strongly on the cognitive science aspect of the AISB remit. The result is a coherent programme with a very strong interdisciplinary character, which is also matched in the choice of plenary speakers. The three symposia looking at the interaction between Computing and Philosophy, the prospect of machine consciousness and the quest for a new, comprehensive intelligence test, form a coherent unit where the eternal questions of who we are and what makes us so are asked from a dual Human-Machine perspective. The Symposia on Active Vision, Computational Models of Cognitive Development and Human Memory for Artificial Agents demonstrate how better understanding of the nature and basis of cognitive processes can advance work on Artificial Intelligence and, inversely, how computational models of these processes can help better to understand them. The prominent multi-agent design and modelling paradigm links the Symposium on Social Networks and Multi-agent Systems with the one on AI and Games. Finally, the Symposium on Learning Language Models from Multilingual Corpora, which brings together some of the first attempts in this area, can also be seen through the prism of such a general notion in Philosophy and Linguistics as semiosis, and the dual role of sign and interpretant that text plays in translations.

We are delighted that after another ten successful years in its long history, the AISB convention is returning to the University of York. The 2011 convention takes place on the brand-new Heslington East campus, the result of a multi-million pound expansion that is now the new home of the Department of Computer Science, and hosts the Excellence Hub for Yorkshire and Humber, a new incubator for interdisciplinary research and interaction between academia and industry. The last few years have seen a strong involvement of the Computer Science Department in such interdisciplinary collaboration through the York Centre for Complex Systems Analysis (YCCSA), and we hope that this convention will provide a boost for more synergy between York departments, with other institutions conducting AI-related research in the region, and beyond. As the programme shows, we have also made an effort to promote cooperation with industry and use the convention to support school outreach. The convention format makes it perfect for establishing dialogue and collaboration in new areas of research, as well as across disciplines, and we hope that this year, it will play again this role to the full. We want to thank everyone who has contributed to it or otherwise made this event possible and wish all participants a fruitful and enjoyable time in York.

Dimitar Kazakov and George Tsoulas

AISB 2011 Convention

4th-7th April 2011
University of York

Proceedings of the
AISB 2011 Symposium on

Architectures for Active Vision

Published by
The Society for the Study of Artificial Intelligence and Simulation of Behaviour

http://www.aisb.org.uk/convention/aisb11/

ISBN 978-1-908187-00-0

Contents

Preface to the Proceedings of the Symposium on architectures for Active Vision

This Symposium of the AISB Convention took place on 4 - 5 April 2011 at the University of York, United Kingdom, supported by the Society for the Study of Artificial Intelligence and Simulation of Behaviour.

The symposium theme was “Architectures for Active Vision” – that is, the control of vision and visual attention. The Symposium has been organized partly as a result of the activity of the Active Vision Network, a collaboration between the universities of Leeds, Sheffield and York supporting work spanning psychology and computer science. The Network is funded by the White Rose Consortium. Some of the results of work carried out by the Network are presented in this Symposium.

Vision is arguably the most researched function of the brain. Nonetheless, high level visual information processing is still poorly understood. A major problem in perception is the volume of information acquired by the body's sensors. Passive approaches to selection of information may deal with the overload by focussing processing on particularly salient inputs. Active vision takes the further step of directing the acquisition of information in a goal-directed manner, in which top-down information plays an important role, possibly overriding saliency in selection of actions. This shift in perspective connects vision with important issues for cognitive systems as a whole, such as action selection, planning and goal-driven behaviour.

The symposium brought together researchers with interests in brain architectures for active vision, the neural basis for action selection in vision, and in the high level modelling in software of structure or mechanisms from the visual system. Contributions spanned theory and experiment from neurobiology, through cognition to bio-inspired software applications.

Simon O’Keefe

Invited presentations

Functional brain architecture underlying eye movements

Melanie Burke[1], Claudia Gonzalez[1] and Graham Barnes[2]

[1]Institute of Psychological Sciences, University of Leeds, U.K. [2]Faculty of Life Sciences, University of Manchester, U.K.

Abstract: In everyday life, in order to perceive the world around us we make thousands of eye movements to people and objects of interest. We make these eye movements in order to place the high acuity region of the retina, known as the fovea, onto the region of interest. Furthermore, when watching a moving target our brain must be able to predict (or anticipate) the target motion in order to place the fovea on the moving target of interest. This prediction is a feature of all motor systems in the body and is thought to help avoid the inherent neural delays observed in the processing of information in the brain. A network of brain areas involved in the process of generating, inhibiting and predicting eye movements to stationary and moving targets will be presented.

1 INTRODUCTION

In order to see the world around us we need to place the high acuity region of the fovea in the eye onto the area or object of interest. The high acuity region of the eye allows us to view the world in full colour and high definition with a resolution close to 576 megapixels [1]. This high acuity region of the fovea is limited in spatial extent to the area of high density cones and subtends only around a 1mm diameter of the back of the eye. Due to this spatial limitation of high acuity within the eye, the eye must move in order to inspect a scene, read a newspaper or watch TV. Based on mainly attentional processes, we make hundreds of thousands of goal directed eye movements every day which mainly comprise a series of saccades and smooth pursuit.

Saccadic eye movements (SAC) are a fast ballistic type of eye movement commonly between 300 and 600˚/s [2]. Their primary function is to re-direct gaze from one area to another. These gaze shifts can be either *involuntary* (or externally driven) based on attentionally salient cues such as looking towards a loud bang or brightly coloured poster, or *voluntary* (or internally driven) based more on purposeful behaviour such as reading [3,4].

Smooth pursuit (SP) on the other hand involves much slower eye movements with speeds between approximately 10 and 40˚/s [2]. These eye movements are primarily used for looking at a single item or object when either the object or ourselves are in motion. The ideal

function of smooth pursuit is to match the speed of the eye with the speed of the motion so the object remains within the foveal region. It was previously assumed that smooth pursuit cannot be initiated in the absence of a visual target (i.e. must be reflexive) [5]. More recently it has been found that smooth pursuit can be *voluntary/volitional* (internally driven) [6] but only the target is predictable.

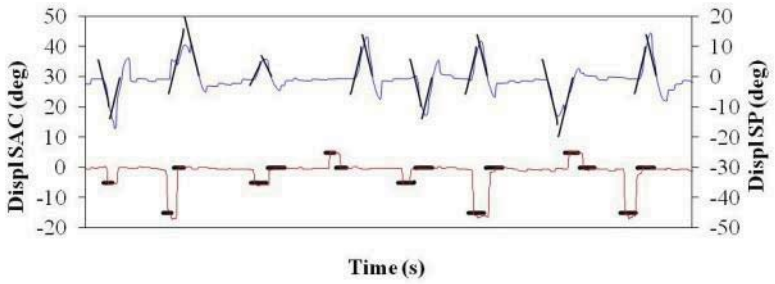


Figure 1. An example of reflexive smooth pursuit (SP, blue trace) and saccadic (SAC, red trace) eye movement traces for randomly presented targets appearing or moving to the left or right of the screen. The black line indicates target motion/position (taken from Burke and Barnes, 2006 [7]).

In order to address the differences between a more internal versus a more external generation of goal-directed behaviour we used a visually-guided (externally driven) and memory-guided (internally driven) paradigm in our study. To do this we took advantage of an important feature of both smooth pursuit and saccadic eye movements have (and in fact most other motor systems), by using their ability to predict or anticipate motion. Without this ability to predict we would not be able to perform simple behavioural operations such as catch a ball, cross the road or drive a car. This ability is a fundamental unconscious feature of motor responses and uses an internal model (derived from memory) to drive the response. The visually-guided reactive responses on the other hand, rely on vision and external cues to drive the response, and is thus more externally driven.

This paper will focus on these two types of eye movements (SP and SAC) under both visually-guided (reflexive) and memory-guided (predictive) conditions to visual targets. It will compare and contrast their relative behavioural hallmarks and neuro-physiological underpinnings.

2 METHODS

Methods are as reported previously in [7] and [8]. Twelve healthy participants performed experiments in both the lab, and the fMRI scanner. Subjects were between the ages of 20 – 39 years and 7 were female.

Paradigms: Subjects performed 5 different tasks in 5 blocks containing 8 individual trials in a row. Tasks were random saccade (RND SAC), predictive saccade (PRD SAC), random smooth pursuit (RND SP) and predictive smooth pursuit (PRD SP) and a control (CON). The pursuit tasks involved following a smoothly moving step-ramp target either left or right at 15 or 30˚/s outward and

then back to the centre of the screen. The saccade task induced a saccade by a target appearing either to the left or right of the fixation cue and then back to centre. Details of the timing of the individual trials are shown in figure 2. Each block contained 8 target presentations either to the left or right of the fixation 8 times in a row (see figure 1 for an example of a SP and SAC random block). If all 8 targets were the same, this comprised a PRD task else each target was random and comprised the RND tasks. The control task (CON) used the same timing as the other tasks (as shown below), however only fixation was required throughout the 8 trials.

For details of data methods and analysis for the eye data please refer to Burke and Barnes 2006 [7].

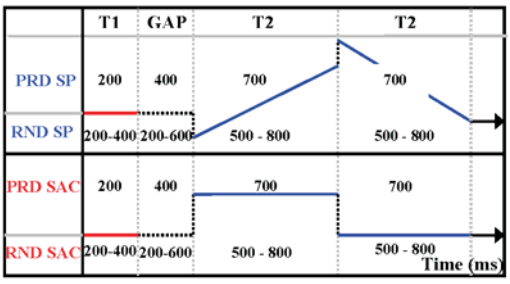
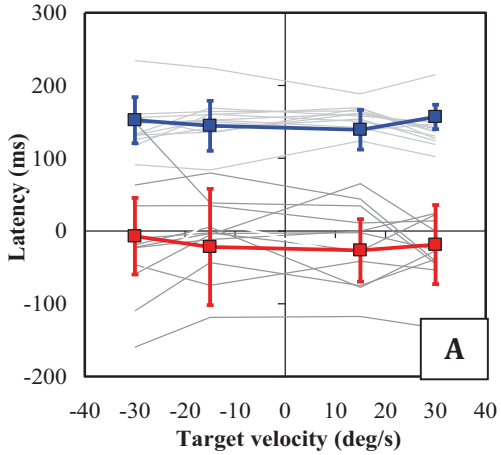


Figure 2: A representation of the timings a single trial for RND SP, PRD SP, RND SAC and PRD SAC. T1 is initial central fixation, this is followed by a gap (blank screen) and then T2 is target motion (for SP) or position (for SAC) (image from Burke and Barnes, 2006 [7]).

Experiments were performed inside the laboratory using an infra-red limbus eye-tracker (IRIS Scalar Medical, BV, CRS Ltd, UK) to record eye movements (see figure 3). Likewise a limbus eye-tracker was also used in the scanner to record eye movements (MR-eyetracker, CRS Ltd, UK) alongside BOLD related brain activations. We recorded activity in the brain using a 1.5T Philips Intera Scanner with SENSE head coil. Details of the fMRI methods, parameters and analysis can be found in Burke and Barnes (2008) [8].

3 BEHAVIOURAL RESULTS

We found a clear dichotomy of data for the eye movements when the target was presented either randomly (visually-guided) (shown in blue in figure 3) or repeated to induce prediction (memory-guided) (shown in red in figure 3). Both pursuit (graph A) and saccadic eye movements (graph B) revealed this effect. This behavioural data reveals a clear prediction, by all subjects, to the predictable targets, and reflexive responses to the randomize targets as expected. For further details refer to Burke and Barnes 2006 [7].



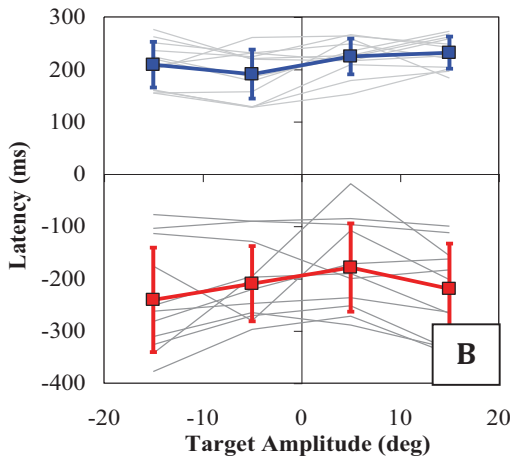


Figure 3: The mean latency for each subject is shown as the grey lines for each target velocity (15 and 30˚/s) or amplitude (15 or 30˚) for the pursuit trials (A) and the saccade trials (B). The Blue squares indicate the mean and standard deviation of all subjects to the random trials, and the red squares the mean and standard deviation to the predictive trials (data from Burke and Barnes, 2008 [8]).

4 BRAIN RESULTS

We found both overlapping and segregated activity in the production of smooth pursuit and saccadic eye movements to visual targets. Pursuit revealed higher activity than baseline (CON task) in; frontal eye fields (FEF), inferior temporal gyrus (ITS), prefrontal cortex (PFC), middle temporal cortex (MT), the cerebellum and brainstem. The saccades revealed higher activity than baseline (CON task) in the supplementary eye fields (SEF), middle temporal gyrus (MTG), frontopolar regions (FP), prefrontal cortex (PFC) and cerebellum. In both types of eye movement negative activity was observed in early visual areas V1 and V2.

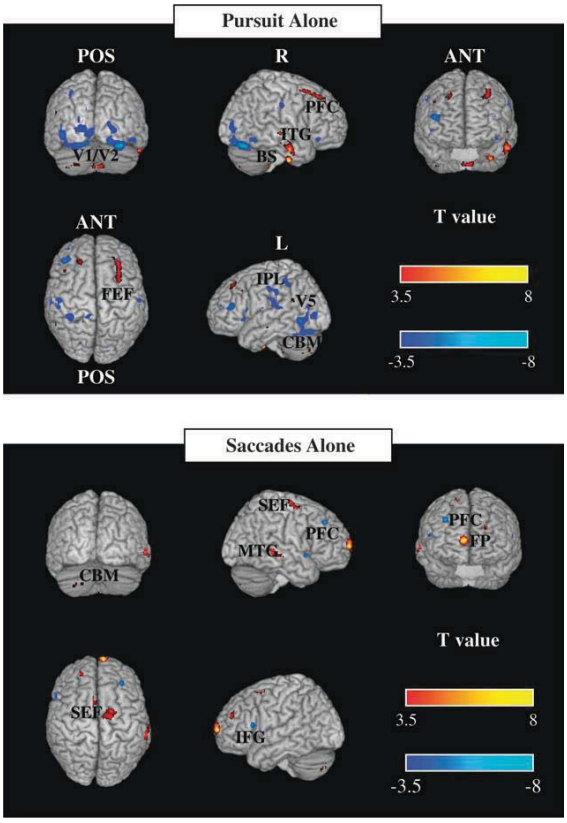


Figure 4: Mean results from all subjects for the pursuit only tasks and the saccade only task in which data for the random and predictive trials have been united and a baseline (CON) condition removed. The warm colour indicates positive activity in comparisons to baseline and the cooler colours are more negative activity (taken from Burke and Barnes, 2008[8]).

5 CONCLUSIONS

Based on the findings published in Burke and Barnes 2008 and 2006, the following conclusions can be drawn (as summarised in figure 5). There are clear overlaps in the brain during the production of smooth pursuit and saccadic eye movements to visual targets, despite difference in their behaviour and function to visual targets. We have also found additional distinctions in the

network of brain areas involved in generating more external (reflexive or direct) versus more internally (memory or indirect) driven responses during vision.

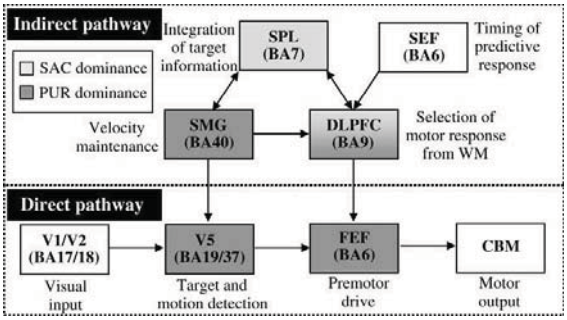


Figure 5: A diagrammatic representation of the brain areas involved in generating internally-driven responses (indirect pathway) or externally driven responses (direct pathway) for both saccades and smooth pursuit. Saccade dominant areas are shown in white and pursuit dominant in gray (figure taken from Burke and Barnes, 2008 [8]).

6 ACKNOWLEDGEMENTS

This work was supported by the Medical Research Council (MRC) and Manchester Wellcome Trust Clinical Research Facility (WTCRF). We also acknowledge the help and support of the staff at the WTCRF.

7 REFERENCES

[1]http://www.clarkvision.com/articles/eye-resolution.html).

[2] Robinson DA. (198l) Control of eye movements. In *Handbook of physiology, Section I: The nervous system, 2.* American Physiological Society: Bethesda, MD.

[3] Remington, R. (1980). Attention and saccadic eye movements. *Journal of Experimental Psychology: Human Perception and Performance, 6,* 726-744.

[4] Hoffman JE, Subramaniam B. (1995). The role of visual attention in saccadic eye movements. Percept. Psychophysics. 57(6):787-795.

[5] Carl JR, Gellman RS. (1987) Human smooth pursuit: stimulus-dependent responses. J Neurophysiol. 57:1446-1463.

[6] Barnes GR, Asselman PT. (1991) The mechanisms of prediction in human smooth pursuit eye movements. J Physiol (Lond) 439: 439-461.

[7] Burke MR, Barnes GR (2006) Quantitative differences in smooth pursuit and saccadic eye movements. Exp Brain Res, 175(4):596-608.

[8] Burke MR, Barnes GR (2008) Brain and behaviour: A task dependent eye movement study. Cerebral Cortex, 18(1): 126-135.

Vision in natural behavior

Benjamin W. Tatler

University of Dundee

Successful completion of many everyday activities requires that foveal vision is allocated to the right place at the right time. Models of gaze allocation in complex scenes are derived mainly from studies of static picture-viewing. From these studies the dominant theoretical framework to emerge has been that of image salience or visual conspicuity: that properties of the stimulus play a crucial role in guiding the eyes. It is now clear that salience-based schemes are poor at accounting for many aspects of picture-viewing and fail completely in the context of natural task performance.

These failures of the basic image salience model have led to the development of more complex models that incorporate higher-level factors, such as expectations about where objects will be and what they will look like. However, there are more problematic issues for developing models using the picture-viewing paradigm. First, there are problematic conceptual assumptions that do not stand up to scrutiny. Second, models based on the picture-viewing paradigm are unlikely to generalize to a broader range of experimental contexts, because the stimulus context is limited, and the dynamic, task-driven nature of vision is not represented. A particular problem with picture-viewing has been the use of the "free-viewing" task as an attempt to isolate "task free" vision. Models developed from static scene viewing paradigms may be adequate models of how we look at pictures, but are rather inappropriate models of how we use gaze in other situations.

Videos are increasingly used to provide more realistic, dynamic scenes for developing models. Accounting for dynamic properties is important and recent models based on video viewing show considerable promise. However, unnatural aspects of movies such as editorial cuts and the framing within a monitor still pose problems for models.

For ecologically valid models of gaze allocation it is important to study vision in the context of natural behaviour in real environments. Developing computational models of gaze allocation that can generalise across many instances of natural behaviour is a difficult goal. However, we see already from studies of gaze selection in natural behaviour that there is an emerging and consistent set of principles for gaze selection. A key principle is that of anticipation or prediction by the oculomotor system. Forward models have been implicated as important principles for models of perception, but do not feature in models of gaze allocation. Framing fixation selection

in terms of predictions from forward models allows us to explain ubiquitous aspects of fixation selection that cannot be explained within conspicuity-based models.

Gaze allocation on the basis of predictions from internal forward models highlights the need to understand how these internal models are constructed. Building internal forward models requires learning and as such modelling efforts for gaze allocation must encompass this learning. The reward system offers a promising candidate for the neural implementation of the learning that is required for deploying gaze in natural behaviour. Sensitivity to reward is found throughout the neural circuitry involved in controlling eye movements. Models of gaze allocation on the basis of reward are emerging and have been used to successfully describe aspects of complex behaviour. Reward-based models of gaze allocation provide a promising direction for the field and offer the building blocks for developing a theoretical model of eye guidance in natural behaviour.

Refereed papers

A biologically based model of active vision

Alex Cope and **Kevin Gurney** [1]

Abstract

How do we decide where to look next in a cluttered visual environment? How are top-down and bottom-up information combined to guide the fovea to points of interest? Here we seek to address these questions using a biologically based model of primate vision. In this model task driven object selection in the visual system's ventral, 'what', stream influences spatial selection in the dorsal 'where' stream, by deploying visual attention in the form of feedback down the ventral stream. Our model predicts the form of this attentional feedback, reproduces trends from a wide range of visual psychophysics data, including that of [5] and [28], and provides a framework for understanding the neural mechanisms behind visual search.

1 Introduction

Active vision, the interrogative approach that most higher visual organisms take to investigating the visual world, is a subset of a larger competency - that of action selection. In most interactions with an organism's external environment multiple motoric choices will be available, and the selection of the correct motor action to suit the organism's current environment is of great importance. In the visual system this action selection problem can be seen as follows: given a complex visual scene, with several areas of detail that require foveal fixation to resolve, the organism must decide which area to fixate first. How this selection is made is an interesting and challenging problem, even given that in the visual motor system there is only one way to fixate a given visual area - this being in stark contrast to the reach motor system, where multiple motor sequences can achieve the same final state (e.g. reaching for a pen can be done in many ways).

Selecting the most useful fixation requires combining the bottom-up information about the visual scene (for example, high contrast areas contain more infomation), and the top-down requirements of the current behavioural task (for example, locating a red pen). This model seeks to investigate how the anatomy of the primate visual system produces these types of task driven behaviour, and in doing so answer the following questions: what is the biological substrate of action selection as undertaken in active vision, and how does task information influence action selection, and thereby mediate the bottom-up sensory information, in the primate brain?

By primarily focusing on replicating the key anatomy of the visual system, and biological accuracy, rather than a model of a specific task, the work presented here aims to elicit a greater understanding of mechanisms involved in primate vision.

[1] University of Sheffield, UK, email: k.gurney@sheffield.ac.uk

1.1 Psychophysics

Visual psychophysics has several well established paradigms that can provide behavioural metrics for active vision, and chief amongst these is visual search [23, 26]. This paradigm involves a participant, human or animal, locating a target stimulus amongst distractor stimuli. A wide range of task variations exist, though for the purposes of this paper the predominant variation is used. Here a single, unique, target is placed amongst a field of distractor stimuli on a blank background. The subject must locate the target as quickly as possible and report success. The time from trial onset to the report of success (reaction time) is recorded as the principal behavioural metric, and the number of distractors that are present in the search array (array size) is the principal variable.

Behaviour in these visual search tasks is traditionally divided into two categories; *efficient,* in which reaction time does not vary with array size, and *inefficient,* in which there is a positive correlation between reaction time and array size. Traditional models [23, 26, for example] modelled these two behaviors by having two stages of visual processing, an initial parallel stage to perform efficient search, and an inefficient serial stage if the first stage could not make a clear decision. The idea of a serial process has little support in neurophysiology [3] however, and further evidence indicates a continuum of search behaviours from inefficient to efficient [4, 5].

1.2 Anatomy and neurophysiology

In the primate brain, the visual system is traditionally divided into two processing 'streams' [24]. After the entry of visual information into the cortex in primary visual cortex (V1) one stream travels in a dorsal direction, and the other in a ventral direction. The dorsal stream is often referred to informally as the 'where' stream, and maintains a topographic retinotopic mapping of visual space. The ventral stream is informally referred to as the 'what' stream, and is characterised by the receptive fields (RF) of neurons further along the stream responding to more complex stimuli, while spatial mapping is simultaneously lost (resulting in larger neuron RF). The ventral stream can therefore also be considered as an 'object recognition' system. Since the ventral stream discards spatial information regarding the objects it recognises, the location on the visual field of task relevant objects must be obtained in order to accurately guide a saccades to them. Evidence suggests [13, 7], and it has been previously proposed [25] that visual attention via feedback down the ventral stream is the mechanism used by the primate brain to recover an object's location. Spatial information is encoded in a 'log-polar' mapping [22], with radial distance from the fovea - distorted by a 'cortical magnification factor' (CMF) (e.g. [18]) - on one axis, and transverse angle on the other.

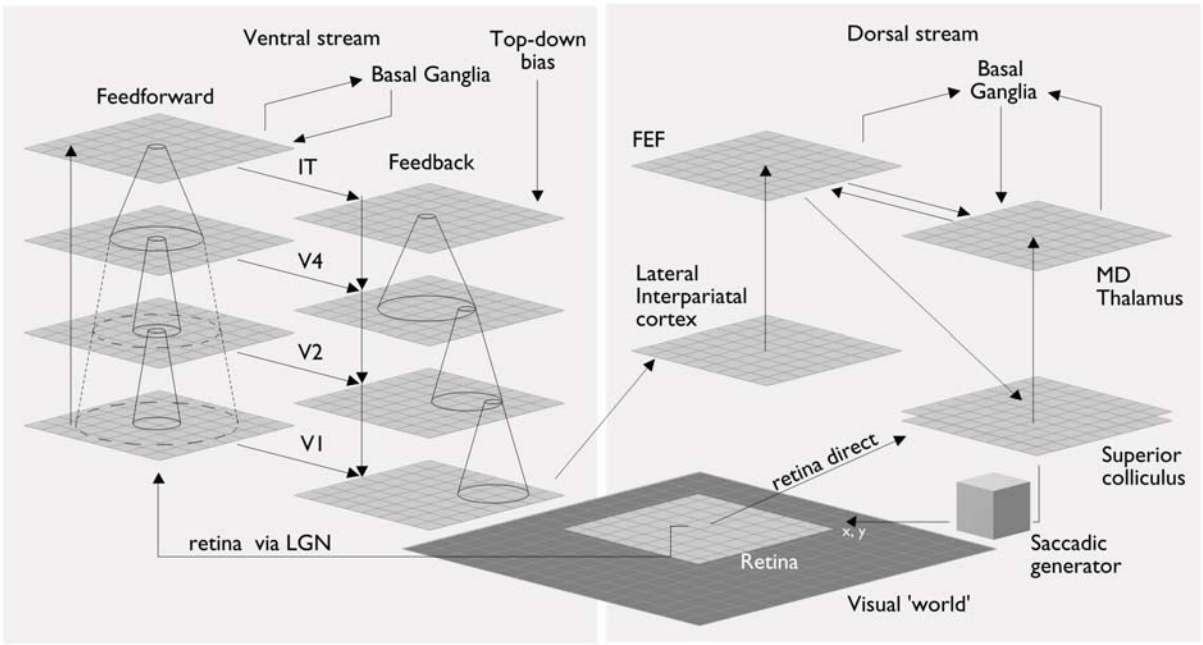


Figure 1. Diagram of the model showing modelled regions, excepting basal ganglia intrinsic connectivity, and connection patterns. See section 2. Grids represent modelled areas, arrows represent connections between areas. Cones are representative the spatial extent of connectivity in the model's ventral stream hierarchy.

2 The Model

In this section an overview of the model will be presented. The model replicates the anatomy (see section 1.2) with a bidirectional ventral stream object recognition hierarchy up to anterior inferior temporal cortex (AIT) from V1 [6], and a dorsal stream to the frontal eye fields (FEF). It integrates a modified model of the oculomotor system from [1]. Spatial mapping is in a biological log-polar form, as described in section 1.2. Selection is performed in AIT [2], where there is an object-based, largely spatially invariant visual map [19, 10], and in FEF [21], where there is a retinotopic spatial visual map. This selection is performed by closed loops of neural connections involving cortex, the basal ganglia, and thalamus. For details see Figure 1. The combined behaviour of these two selection mechanisms in determining saccade targets was investigated.

In the model the dorsal stream consists of Lateral Intrapariatal cortex (LIP), and the Frontal Eye Fields (FEF). Both areas have a spatial, retinotopic, mapping and have been found to represent the locations of behaviourally relevant objects in their activity [9][20], with FEF showing activity reflecting the visual decision making process [15]. In the model this selection is performed by a closed loop of neural connections through cortex, the basal ganglia, and thalamus. The basal ganglia has been implicated in selection between competing action channels [17] and gating oculomotor actions [11]. Selection in the dorsal stream of the model acts upon a spatial map, representing the likelihood of behaviourally task-relevant objects being present (the behavioural salience) at each location on the visual field. This salience map is produced as the output of the ventral stream of the model.

The model's ventral stream consists of a hierarchy of increasingly spatially invariant cortical areas linked by anatomically distinct feedforward and feedback excitatory connections. The receptive fields of the neurons increase up the stages of the feedforward hierarchy. The complexity of the stimuli the neurons are tuned to likewise increases by combining stimuli represented at the previous stage together. Spatial invariance increases by means of a MAX operation at each stage (see [12]). Within each receptive field in the feedforward pathway there is competitive processing to represent the strongest, and thus most likely, stimulus representation for that region. At the top of the hierarchy, in IT, there is almost no spatial information. Here the task information - the current goal of the model, what it is trying to achieve - is introduced by modulating the activity of the neuron representing the task-relevant object by a hand crafted top-down bias signal.

As described in section 1.2, the highest layers of the ventral hierarchy contain little spatial information about where objects are on the visual field, and this information must be recovered to direct gaze to behaviourally relevant objects. Evidence suggests that this is achieved by visual object-based attention [13, 7]. In experiments, not reported here, several methods of achieving object-based attention through feedback connections in the ventral stream were tested, finding that distinct feedforward and feedback paths were necessary to recover spatial information about the chosen objects reliably.

The method used in the model to recover the spatial information proceeds as follows. In the highest visual area of the model very coarse spatial information is used, in combination with the most complex representations of stimuli, to drive attention to the segments of the visual field containing evidence of the objects biased by the top-down signal. Since the most complex representations are used, this attention provides high specificity to distinguish the chosen objects from task irrelevant objects, which compensates for the coarse spatial scale. At the next level down in the hierarchy this attentional signal is now constrained to certain segments of the visual field by the higher visual area. This coarse attention is then combined with

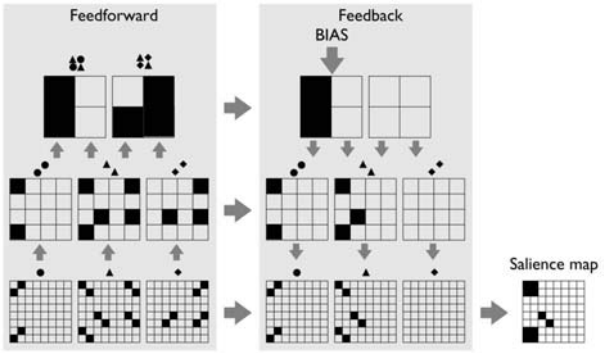


Figure 2. A simple example of the mechanism for feedback. Black represents active neural units, white inactive. Three base features are combined into intermediate features, then full objects. One object is biased. The product of the bias, and the evidence at each location, provides the attentional bias to the next level. In the next level the product of the attentional bias and the bottom-up evidence is once again combined, and the product of the two at each spatial location, and for each sub-object, provides the attentional bias from the final level, thus highlighting strongly the desired objects, and weakly a close object that shares a feature. Selection in the dorsal stream will drive gaze to the correct objects.

recognition of less complex stimuli, consisting of parts of the objects in the higher layer. Where there is top-down attention AND a part of the object, feedback continues. While these object parts are less specific to the objects to be located, the constraint of the coarse attention from the higher level compensates for this, and there will be less attentional bias erroneously provided to parts of object shared by other, task irrelevant, objects, as to be biased these must now lie in the same part of the visual field as the objects to be located. This process repeats down the hierarchy, slowly recovering the likely target identity. Figure 2 shows a simplified example.

In addition to the object salience map from the ventral stream, selection in the FEF operates on a salience map from the superior colliculus (SC). The SC responds to changes in luminance: the appearance, disappearance, and movement of objects [27]. A salience map of this information also enters the FEF selection loop.

In order to focus the investigation on visual selection we chose a highly limited feature set (much less complex than that of human vision). The objects recognised by the model consist of seven segment depictions of numbers, similar to those found on digital clocks (see Figure 3). These objects therefore consist solely of unique spatial arrangements of vertical and horizontally oriented line segments.

Figure 3. The object set used with the model.

The model is dynamic, using leaky integrator, rate-coded model neurons. Connection patterns between neurons in the model are hand crafted, as this allows a greater understanding of the behaviour of the model. The model is implemented using a set of custom created neural simulation tools (ModLIN, Modular Leaky Integrator Neurons) developed against the BRAHMS modular execution framework [14].

3 Experiment 1: Visual search - perceptual learning

3.1 Methods

The visual search task was to locate a '6' numeric digit amongst '5' and '9' digits and respond by making a single saccade to within a 10 pixel radius of the correct location. Stimuli consist of 5 by 9 pixel digits with a fixed luminance, modulated by noise on a per pixel basis, arranged on a 300 by 300 pixel 'world'. A subregion of 150 by 150 pixels is taken from this world as the extent of the visual field, saccades move this subregion within the world. The stimuli are evenly spaced around the circumference of an imaginary circle 60 pixels in diameter. The log-polar mapping of the visual field was calibrated to give the diameter of the imaginary circle as 24 degrees. This gives a stimulus size of 2 by 3.6 degrees.

Two main conditions were used. First is the 'naive' condition. In this condition the model is put into a state mimicking that of a participant when first presented with a new visual search task. The target and distractors are novel, and the model holds no complete internal representation for the target as a '6'. Instead, in order to bias the '6', the top-down bias must influence the parts of the '6', the top - which is a 'c' shape, and the bottom - which is an 'o' shape. The '6' shares one of these parts with each of the '5' and the '9'. In the second, 'trained', condition the model mimics a participant after hundreds of trials on the visual search task. The model has a complete, but weakly tuned, internal representation of a '6', and top-down task information is provided by biasing this target representation. Since the representation is weakly tuned, the model still finds some evidence for a '6' at the locations of the distractors. In all other aspects these two conditions are identical. See figure 5 for a diagram of the two bias conditions. These two conditions attempt to reproduce the effect of perceptual learning as demonstrated in [5].

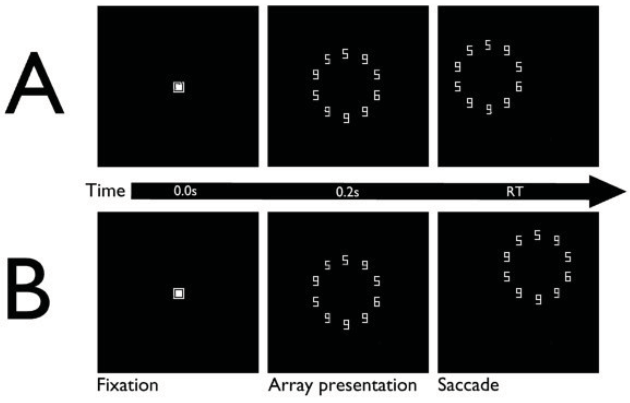


Figure 4. Top: Diagram of experimental procedure and example trials. Fixation is shown, then at 0.2 seconds into the trial the search array is shown. The model responds with a saccade with latency RT (reaction time). The saccade is either (A) correct if the '6' is fixated or (B) incorrect if a distractor is fixated.

Array sizes of 2, 4, 8 and 10 stimuli were used for each main condition. Twelve evenly spaced locations around the circle were used

Figure 5. Diagram of the neural representations biased in the 'naive' (A) and 'trained' (B) conditions.

for the target location, and the distractor locations evenly distributed around the circle based on the target location, with equal numbers of each distractor type where possible. 10 repetitions were performed for each target location. The experiment proceeded as follows. A fixation point was presented at the centre of the circle. After 0.2 seconds, by which time the model had reached a steady state, the fixation point was removed, and the search array presented. The trial was terminated when the model made a correct saccade. If the model failed to make a correct saccade within 4 seconds of the search array being presented the trial was considered an error and terminated. See Figure 4 (left).

3.2 Results

Figure 6 (right) shows the results of this experiment. Linear regression of the data gives search slopes of 73.3ms/item for the 'naive' condition, and 42.4ms/item for the 'learned' condition. This shows a flattening of the search slope with perceptual learning in the model.

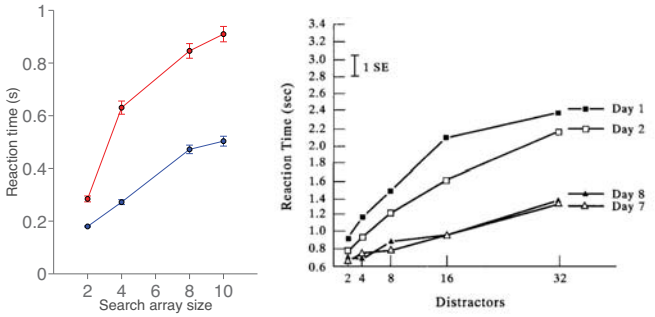


Figure 6. Graph showing the change in search time with varying number of distractors for the 'learned' (red), and 'naive' (blue) conditions. Error bars reflect standard errors. Bottom right: Graph from [5] showing perceptual learning over subsequent days of trials.

4 Experiment 2: Visual search - efficient vs inefficient

4.1 Methods

Two visual search tasks are used. For the condition A search task the model is required to find a horizontal bar amongst vertical bar distractors. The condition B task is to locate a '6' numeric digit amongst '5' and '9' digits, with biasing as in the 'trained' task. In both tasks a correct response is determined by the model making a saccade to within a 10 pixel radius of the the correct location, regardless of how many saccades this correct saccade takes to make.

In condition A the stimuli consist of five pixel by one pixel vertically and horizontally oriented bars. Stimuli in condition B consist of 5 by 9 pixel numeric seven segment display digits. The stimuli are randomly placed in a 150 by 150 pixel search array, making sure that there is a minimum centre to centre separation between stimuli of 17 pixels. The environment is as described in Section 3.1.

Array sizes of 2, 6, 12 and 24 stimuli were used for each condition. 480 repetitions were performed for each condition and array size. A fixation point was presented at the centre of the array. After 0.2 seconds, by which time the model had reached a steady state, the fixation point was removed, and the search array presented. The trial was terminated either when the model made a correct saccade, in which case the time was noted, or when four seconds from array presentation had elapsed, in which case the trial was considered an error. For each case (condition and search array size) 480 trials were run.

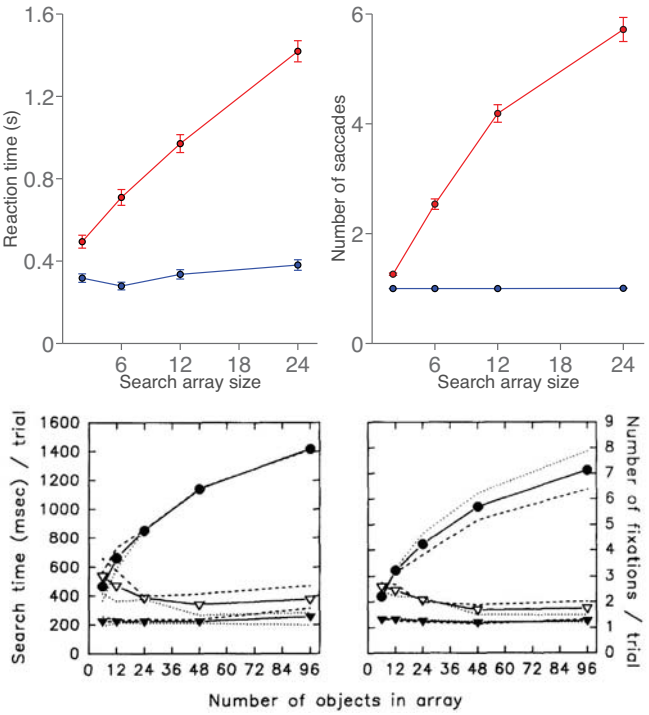


Figure 7. Top left: Graph showing average reaction time against search array size for condition A (red, triangles) and condition B (blue, circles) visual search tasks. Top right: Graph showing average number of saccades taken to locate the target in the for condition A (red, triangles) and condition B (blue, circles) visual search tasks. Error bars reflect standard errors. Bottom: Graph from [16], showing inefficient, (circles) and two types of efficient (triangles), search tasks (dashed lines are two subjects, solid is average).

4.2 Results

The principal result is shown in Figure 7 (left). The behaviour of the model parallels that found in human and primate subjects, with search times for the simple features in condition A being essentially flat or independent of the search array size, with a search slope of 3 ms/item. Condition B, on the other hand, in which the target and distractors shared partial features, on the other hand, showed a strong dependence of search on the number of stimuli in the search array, with a search slope of around 42 ms/item. The number of fixations taken also demonstrates the same pattern found in human subjects (Figure 7, right).

5 Experiment 3: Visual search - effects of stimulus onset

5.1 Experiment 3a: Stimuli and procedure

The visual search task was to locate a '6' numeric digit amongst '5' and '9' digits and respond by making a saccade to within a 10 pixel radius of the the correct location. The stimuli and procedure are chosen to match as closely as possible those used in [28]. Stimuli and the world are as described in Section 3.1. The stimuli are evenly spaced around the circumference of an imaginary circle 42 pixels in diameter. The visual field was calibrated to give the diameter of the imaginary circle as 4.0°. The stimulus are 0.48° by 0.86°, which maintains the stimulus height from [28].

There are two main conditions. In the 'onset' condition, a fixation point appears at the centre of the imaginary circle. This is maintained for 1.0 seconds as in the original paper. When the fixation point is extinguished the search array of three or seven digits appears. In the second, 'non-onset', condition an array of figure-eight placeholders surround the fixation point, and on removal of the fixation point after 1.0 seconds segments of the placeholders are removed to reveal the search array. Trials are terminated when the target is located by a correct saccade, or considered an error if after four seconds the target is not located (see Figure 8). As in the original paper, reaction times (RT) greater than twice the mean were discarded.

Twelve locations for the target were used, and forty trials were run for each target location, each display size, and each condition. Equal numbers of each distractor type were used and the locations of the distractors were randomised for every trial. This gives a total of 1920 trials.

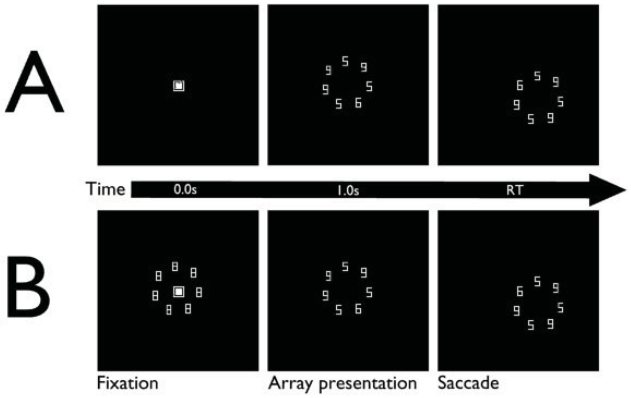


Figure 8. Diagram of the procedure for experiment 3a - (A) Onset condition: there are no masks over the search array positions. (B) Non-onset condition: figure '8' masks appear over the search array positions.

5.2 Experiment 3b: Stimuli and procedure

The task, stimuli and environment are as in section 5.1. Once again, there are two main conditions. In both conditions a fixation point appears and figure-eight placeholders appear on six of the seven search array locations. In the onset condition the target appears at the unmasked search location, in the non-onset condition one of the distractors appears at the unmasked location (See Figure 9).

Twelve locations for the target were used, and forty trials were run for each target location, each display size, and each condition. Equal numbers of each distractor type were used and the locations of the distractors were randomised for every trial. This gives a total of 1920 trials.

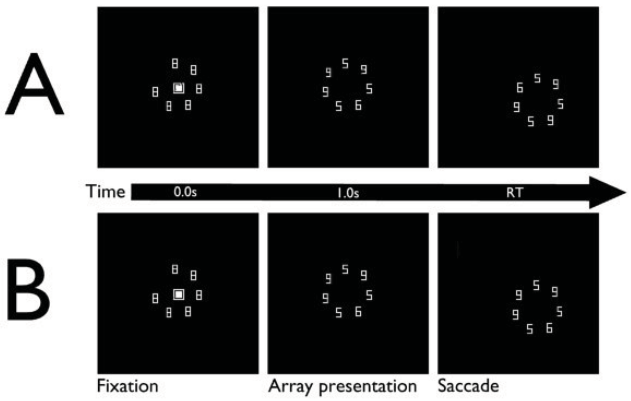


Figure 9. Diagram of the procedure for experiment 3b - (A) Onset condition: the target appears at the unmasked location. (B) Non-onset condition: a distractor appears at the unmasked location.

5.3 Results

For experiment 3a the mean RT for the onset and non-onset conditions for displays of three and seven elements, along with standard errors, are shown in Figure 10. These results show the same effects as those reported by [28] and [8].

The mean RT for the onset and non-onset conditions along with standard errors, for displays of three and seven elements, are shown in Figure 11. As can be seen from the graph, the data follows the same trends as those reported by [28].

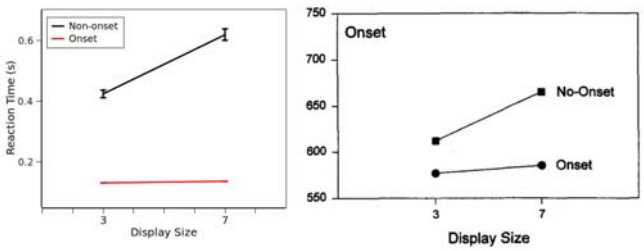


Figure 11. Results for Experiment 3b (bottom left) and experimental data from [28] (bottom right). Reaction time is plotted as a function of display size for the onset (red / circles) and non-onset (black / squares) conditions, respectively.

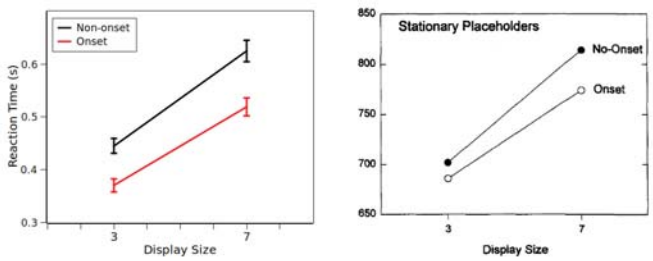


Figure 10. Results for Experiment 3a (top left) and experimental data from [28] (top right). Reaction time is plotted as a function of display size for the onset (red / unfilled circles) and non-onset conditions (black / filled circles), respectively.

6 Discussion

The results above demonstrate that the model, constrained by the anatomical and functional evidence from the brain, is capable of reproducing a range of psychophysical data. The power of the model to explain these results arises from the anatomical constraints of the model, which provide a clear correspondence between the model and brain function.

The mechanisms that give rise to efficient and inefficient search behaviours in the brain, as well as the shift from inefficient to efficient search obtained through extensive repetition of a specific visual search target and distractor stimulus set [5], can be explained by this model.

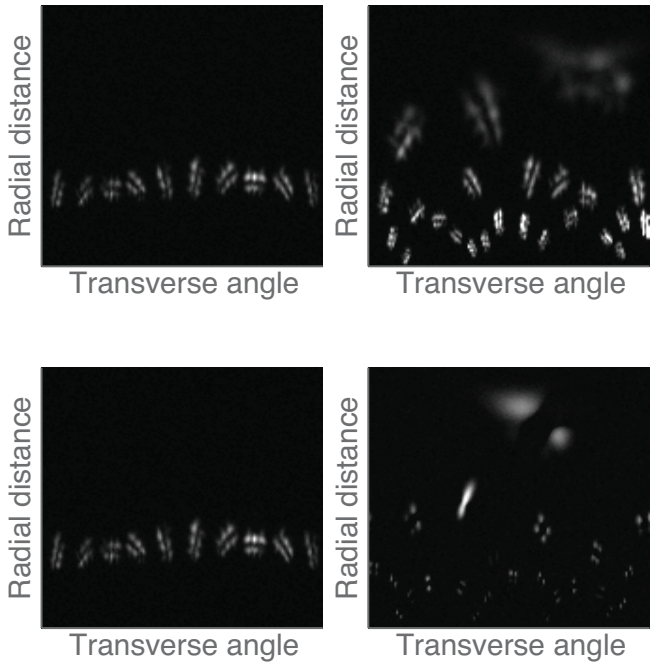


Figure 12. Activity in layer LIP of the model. The layer is log-polar mapped and activity is averaged over 100 iterations to remove noise. Top left: trained model on perceptual learning task. Bottom left: naive model on perceptual learning task. Top right: inefficient stimuli on efficient vs inefficient task. Bottom right: efficient stimuli on efficient vs inefficient task.

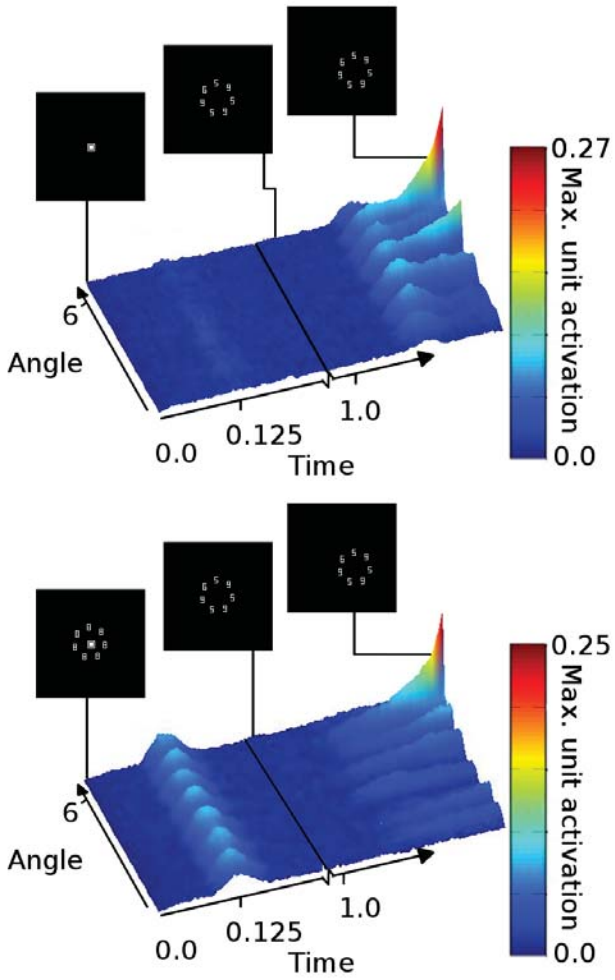


Figure 13. Plot of maximum cell activity on each radial column of layer FEF of the model through time. Top: Example trial for Onset Experiment 3a - onset condition. Bottom: Example trial for Onset Experiment 3a - non-onset condition. Phasic activity due to onset moves from mask onset (bottom) to array onset (top), thus speeding selection of the first saccade.

In inefficient search the ability of the ventral stream to differentiate between target and distractor is poor. The model suggests that this effect is caused in part by there being no single target representation, but instead representations for parts of the target, e.g, a red vertical bar could be represented as the features red, and vertical bar. Additionally poor tuning of the target representation can affect the efficiency of the search. In this case the representation of the target and distractors would span a greater amount of feature space, with the tuning curves overlapping each other. The result in both of these cases is the same, the distractors sharing close features with the target receive a portion of the bias intended for the target, due to the shared feature representations, or to the overlap in feature space. This results in a salience map input into the FEF from the ventral stream that contains more distributed activity, and therefore more ambiguity in target choice. This leads to greater difficulty locating the target, both during covert selection in the brain, and overt shifts of gaze, leading to greater reaction times as the number of stimuli increases. After hundreds of trials of the same target and distractor stimulus set, the ability of the ventral stream to differentiate between target

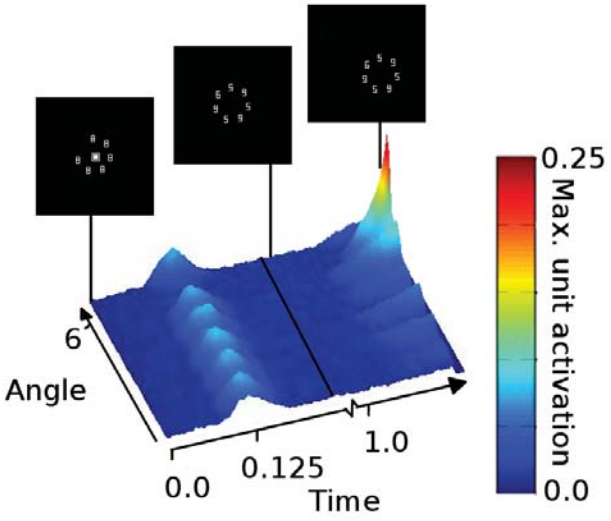


Figure 14. Example trial for Onset Experiment 3b - onset condition. Phasic activity due to onset boosts activity of the unmasked object, thus increasing the chance of it being fixated first saccade. This leads to efficient search behaviour when the target is unmasked object, as in this trial.

and distractor improves, and the salience map provided to the FEF is less ambiguous. The result of this is that the number of distractors has a reduced effect on reaction time. Efficient targets, are able to be distinguished very easily from the first encounter by being distant in feature space. Figure 12 shows these two cases. In the case the perceptual learning task the improvement is due to a slightly less ambiguous salience map, so the effect on the reaction time slope is weaker. In the case of the inefficient vs efficient search task the difference in ambiguity in the two salience maps is much greater, and therefore the difference in behaviour is much more pronounced.

The results from the onset task can also be explained by the model. These effects are caused by the combination of phasic salience map from SC with the object salience map from the ventral stream. The results for experiment 3a (see Figure 10) arise from the phasic onset being moved from the array onset to the mask onset. This leads to slower selection of the first saccade target, and thus a shift in the time taken to find the target. Since the onset drives the first saccade this shift is largely independent of the number of distractors. Figure 13 shows the neural activity in the FEF during example trials.

In experiment 3b (see Figure 11) a similar effect occurs, however in this case the phasic onset only affects one object, increasing the chance of that object being fixated on the first saccade. If this object is the target then the behaviour becomes similar to efficient search, however if the object is not the target, normal inefficient search proceeds. Figure 14 shows the neural activity in the FEF during example trials.

These explanations are possible due to the strong biological basis of the model, which provides a clear link between the behaviour and the neural mechanisms involved.

REFERENCES

[1] J. M. Chambers, *Deciding where to look: A study of action selection in the oculomotor system*, Ph.D. dissertation, Sheffield: University of Sheffield, 2007.

[2] L. Chelazzi, E. K. Miller, J. Duncan, and R. Desimone, 'A neural basis for visual-search in inferior temporal cortex', *Nature*, **363**, 345–347, (1993).

[3] Jeremiah Y. Cohen, Richard P. Heitz, Geoffrey F. Woodman, and Jeffrey D. Schall, 'Neural basis of the set-size effect in frontal eye field: timing of attention during visual search.', *Journal of Neurophysiology*, **101**(4), 1699–1704, (2009).

[4] J. Duncan and G. W. Humphreys, 'Visual-search and stimulus similarity', *Psychological Review*, **96**, 433–458, (1989).

[5] A. Ellison and V. Walsh, 'Perceptual learning in visual search: Some evidence of specificities', *Vision Research*, **38**(3), 333–345, (1998).

[6] Daniel J. Felleman and David C. Van Essen, 'Distributed hierarchical processing in the primate cerebral cortex', *Cerebral Cortex*, **1**(1), 1–47, (1991).

[7] Angela Lee Gee, Anna E. Ipata, and Michael E. Goldberg, 'Activity in V4 reflects the direction, but not the latency, of saccades during visual search.', *Journal of Neurophysiology*, **104**(4), 2187–2193, (2010).

[8] B. S. Gibson, 'Visual quality and attentional capture: a challenge to the special role of abrupt onsets.', *Journal of Experimental Psychology: Human Perception and Performance*, **22**(6), 1496–1504, (1996).

[9] J. P. Gottlieb, M. Kusunoki, and M. E. Goldberg, 'The representation of visual salience in monkey parietal cortex.', *Nature*, **391**(6666), 481–484, (1998).

[10] Charles G. Gross, 'Single neuron studies of inferior temporal cortex', *Neuropsychologia*, **46**, 841–852, (2008).

[11] O. Hikosaka, Y. Takikawa, and R. Kawagoe, 'Role of the basal ganglia in the control of purposive saccadic eye movements.', *Physiological Reviews*, **80**(3), 953–978, (2000).

[12] Ilan Lampl, David Ferster, Tomaso Poggio, and Maximilian Riesenhuber, 'Intracellular measurements of spatial integration and the max operation in complex cells of the cat primary visual cortex.', *Journal of Neurophysiology*, **92**(5), 2704–2713, (2004).

[13] S. D. Mehta, I. Ulbert, and C. E. Schroeder, 'Intermodal selective attention in monkeys. i: Distribution and timing of effects across visual areas', *Cerebral Cortex*, **10**, 343–358, (2000).

[14] Ben Mitchinson, Tak-Shing Chan, Jon Chambers, Martin Pearson, Mark Humphries, Charles Fox, Kevin Gurney, and Tony J. Prescott, 'Brahms: Novel middleware for integrated systems computation', *Advanced Engineering Informatics*, **24**(1), 49–61, (2010).

[15] Ilya E. Monosov, Jason C. Trageser, and Kirk G. Thompson, 'Measurements of simultaneously recorded spiking activity and local field potentials suggest that spatial selection emerges in the frontal eye field', *Neuron*, **57**, 614–625, (2008).

[16] B. Motter and E. Belky, 'The zone of focal attention during active visual search', *Vision Research*, **38**(7), 1007–1022, (1998).

[17] P. Redgrave, T. J. Prescott, and K. Gurney, 'The basal ganglia: A vertebrate solution to the selection problem?', *Neuroscience*, **89**, 1009–1023, (1999).

[18] J. Rovamo and V. Virsu, 'An estimation and application of the human cortical magnification factor', *Experimental Brain Research*, **37**(3), 495–510, (1979).

[19] K. Tanaka, 'Inferotemporal cortex and object vision.', *Annual Review of Neuroscience*, **19**, 109–139, (1996).

[20] K. G. Thompson and N. P. Bichot, 'A visual salience map in the primate frontal eye field', *Development, Dynamics and Pathology of Neuronal Networks: From Molecules to Functional Circuits*, **147**, 251–262, (2005).

[21] K. G. Thompson, N. P. Bichot, and T. R. Sato, 'Frontal eye field activity before visual search errors reveals the integration of bottom-up and top-down salience', *Journal of Neurophysiology*, **93**, 337–351, (2005).

[22] R. B. Tootell, E. Switkes, M. S. Silverman, and S. L. Hamilton, 'Functional anatomy of macaque striate cortex. ii. retinotopic organization.', *Journal of Neuroscience*, **8**(5), 1531–1568, (1988).

[23] A. M. Treisman and G. Gelade, 'Feature-integration theory of attention', *Cognitive Psychology*, **12**, 97–136, (1980).

[24] L. G. Ungerleider and M. Mishkin, *Two cortical visual systems*, 549–586, Cambridge, MA: MIT Press, 1982.

[25] F. Van der Velde and M. de Kamps, 'From knowing what to knowing where: Modeling object-based attention with feedback disinhibition of activation', *Journal of Cognitive Neuroscience*, **13**, 479–491, (2001).

[26] J. M. Wolfe, 'Guided search 2.0: A revised model of visual search', *Psychonomic Bulletin & Review*, **1**(2), 202–238, (1994).

[27] R H Wurtz and J E Albano, 'Visual-motor function of the primate superior colliculus.', *Annu. Rev. Neurosci.*, **3**, 189–226, (1980).

[28] Steven Yantis and John Jonides, 'Attentional capture by abrupt onsets:

New perceptual objects or visual masking?', *Journal of Experimental Psychology: Human Perception and Performance*, **22**(6), 1505–1513, (1996).

Multi-modal visual attention for robotics active vision systems - A reference architecture

Martin Hülse, Sebastian McBride and **Mark Lee** [1]

Abstract. This work introduces an architecture for a robotic active vision system equipped with a manipulator that is able to integrate visual and non-visual (tactile) sensorimotor experiences. Inspired by the human vision system, we have implemented a strict separation of object location (where-data) and object features (what-data) in the visual data stream. This separation of what- and where-data has computational advantage but requires sequential fixation of visual cues in order to create and update a coherent view of the world. Hence, visual attention mechanisms must be put in place to decide which is the most task-relevant cue to fixate next. Regarding object manipulation many task relevant object properties (e.g. tactile feedback) are not necessarily related to visual features. Therefore, it is important that non-visual object features can influence visual attention. We present and demonstrate visual attention mechanisms for an active vision system that are modulated by visual and non-visual object features.

1 Introduction

Robotic systems interacting in a truly autonomous fashion in unconstrained environments is one of the most challenging research and engineering topics in robotics. Solving these problems has been a major driver for the development of the field of cognitive robotics whereby biological systems, where many of the problems of autonomy have been resolved, are used as putative templates for robotic architectures. In the context of animals, cognition is defined as the process of acquiring and using knowledge about the world for goal-orientated purposes. One of the most fundamental aspects of this process is the ability to locate (where) and identify (what) objects within the environment and, to be able to do this in the context of an ever-changing visual scene due to body, head and eye movement. With this attribute, however, comes the problem of data handling due to the immense amounts of visual information that potentially has to be processed. Biological systems have again resolved this issue through the phenomenon of visual attention whereby objects and events of high relevance are placed at the centre of the visual scene at high resolution, whilst less relevant visual information are either kept at low resolution or not maintained within the visual field. In other words, visual attention is a mechanism that allows the allocation of resources to the most critical components of the agent's environment. The ability to do so is largely due to the mechanism of saccade and the graded number of visual sensors across the retina (highest at the centre [fovea] decreasing outwards) with continuous fixations allowing a single egocentric precept of the agent's immediate environment to be created. In addition, critical components of the agent's environment that drive visual attention can be largely categorized into two areas; objects and events that have high intrinsic saliency (due of high luminance, contrast or movement) or those that are relevant to the current task. This is commonly referred to as bottom-up versus top-down visual attention respectively and has often been used as an inspiration or framework for the development of robotic architectures, e.g. [8]. A potential extension of this type of architecture is use of non-visual features to modulate visual attention. For example, using non-visual sensorimotor experiences (e.g. hardness of object) to modulate visual attention in the context of a set task (reach for hard objects). In this paper, we present a computational architecture for a robotic active vision system and a manipulator that encapsulates bottom-up versus top-down visual attention but expands this concept towards added multi-modal (visual and tactile) modulation of the system.

The architecture presented here is termed cognitive in the sense that it is inspired by cognitive science, brain research and also developmental psychology. Furthermore, it adheres to the generalised mechanics of human retina as described above, with high resolution in the center with low resolution on the periphery, to create an active vision system where saccades allow visual information to be gathered from an extended egocentric space. A critical feature of this architecture is the existence of a common reference frame to allow a) the use of an active vision system, b) the ability of cross-modal modulation and c) the transformation of visual stimuli into reach target coordinates. The reader will see that from an engineering perspective this exploitation of such a common reference frame saves computational costs with respect to processing time and memory.

The three objectives of this paper are, therefore, firstly, to present a computational framework that enables an active vision system equipped with a manipulator to integrate robustly multi-modal visual and non-visual sensor experience. Secondly, to demonstrate the different ways in which non-visual sensorimotor experiences can modulate visual attention in terms of fixation patterns. Thirdly, to promote this architecture as a reference architecture for humanoid and anthropomorphic robot systems for visual attention and object manipulation. As we will outline, this reference architecture offers an experimental setup for validating different cognitive models and therefore it is a promising tool for developing new hypotheses and deriving insight about the nature and biology of visual attention and object manipulation.

2 Robotic Setup

Our robotic scenario includes an active vision system and a manipulator, i.e. a robot arm equipped with an 3 finger hand system, Fig. 1.

[1] Intelligent Robotics Group, Department of Computer Science, Aberywtwyth University,SY23 3DB, Wales, UK, email: {msh, sdm, mhl}@aber.ac.uk

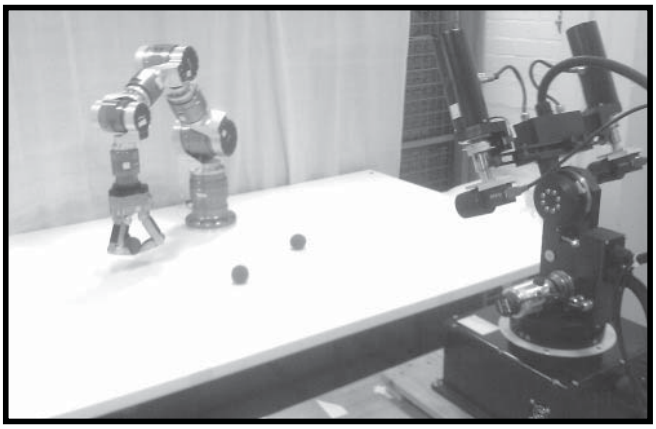


Figure 1. Robotic setup including robotic arm and hand systems and an active vision system.

Together, robot arm and hand systems (SCHUNK GmbH & Co. KG) have 14 DOF (degree of freedom). Each finger has two segments and each of these segments is equipped with one pressure sensitive sensor pad providing tactile feedback when grasping an object.

The active vision system consists of two cameras (both provide RGB 1032x778 image data) mounted on a motorised pan-tilt-verge unit. Here, only two DOF, verge and tilt of the left camera, are used. The motors are controlled by determining their absolute target position p, or the change of the current position Δp, given in radians (*rad*). In its default position the vision system is oriented towards the robotic manipulator which is mounted on a table where it can grasp and replace objects (coloured balls).

3 Computational Architecture

Our architecture combines four basic functions: image processing, object fixation, reaching towards and finally, grasping of objects (Fig 2). Central element is the *spatial memory* which modulates the current visual input represented in the gaze space. The spatial memory stores object locations and is itself modulated by object features stored in the *feature memory*. In the following this architecture and the processes of feature modulation are be explained in detail.

3.1 Basic functions

Image processing. The image processing of the original RGB camera image data is aimed at mimicking the retina of the human eye. These properties are simulated by dividing the original camera RGB data into two data streams. One stream is fed by image data from a small predefined region around the image center having maximal resolution. The second stream starts with a scaled down resolution of the image. Hence, we get one low resolution image of the whole visual scene and a high resolution image of the image center. At this point both streams contain colour information.

The low resolution color image is further filtered with respect to three colours components (red, blue and green) and changes in the image data (movement). The individual filter outputs are linearly combined and normalised generating a saliency map. Each pixel in this map has a real value in the range $[0.0, 1.0]$ indicating image regions of red, blue, green, or movement. The domain this saliency map is represented is called *retinotopic space* or *retinotopic reference frame*.

Due to the linear combination and normalisation the resulting saliency map can highlight specific colour components. Hence, it might be that red image regions are amplified while blue regions appear with low intensity, even if in the original image the blue regions show much higher intensity than red regions. The process of saliency map generation is here also called *spatial filtering* because it provides image regions which later will be used to derive potential fixation and reaching targets.

It is worth noting that this simple purely colour and movement based saliency map can be replaced by any other mechanisms generating more advanced saliency maps (see for instance the classical approach [7]). Furthermore, it must be emphasised that the final spatial filter output does not have any reference to the original colour data. The specific object features can be derived from the high resolution image data only.

Feature filters are applied to the image center of the high resolution image data. Here, specific feature filters can indicate properties about shape or texture. In this scenario again we made use of colour filters only. Hence, a 3-dimensional vector (*feature vector*, v_v) is provided indicating the intensity of red, green and blue. Obviously, this feature vector can be easily extended by other feature measures.

At this point one can see that the original RGB image data is transformed into two data streams: one delivers a low resolution saliency map and the other a feature vector v_v. Although very simplified and considering colour components (R, G, B) and image changes only this implementation mimics the separation of what (feature) and where (spatial) information.

Object fixations. Since visual features can only be detected from the image center, the camera must fixate the object in order to get access to its particular feature vector v_v. The fixation of an object is generated by saccadic camera movements or eye saccades which bring a selected image region into the image center, the fovea. In former experiments we have demonstrated how such eye saccades can be learned by a robotic active vision system [3]. In this scenario the saliency map refers to potential fixation targets. An eye saccade mapping maps a specific image (X,Y)-coordinate to the relative motor movements Δp. When executing these relative motor movements, the selected image regions is placed into the image centre, where feature vector v_v is derived from.

In more recent studies we have also shown that if the current absolute motor positions of the active vision system p are given then the relative motor movements Δp derived from the eye saccade mappings can be used to estimate the final absolute motor positions if a saccade towards the stimulus would be executed [4]. Applying this estimation for each stimulus in the saliency map we actually map all stimuli $(X,Y)^*$ from the retinotopic reference frame into the *gaze space*. The gaze space is defined by the range of absolute motor positions of the active vision system. The set of motor positions $< p >^*$ derived from the saliency map is called *local gaze space* since it represents the stimuli in the current visual field which is only a small part of scene visually accessible.

The gaze space is the domain of the *action selection process* for saccadic camera movements. Action selection is the process of selecting the most salient stimulus p in gaze space, followed by the execution of the corresponding motor command which drives the camera motors into position p. The eye saccade is said to be successful if after the execution the image centre of the saliency map contains non-zero entries. This *state* information is detected and provided by the spatial filter (see Fig. 2). If the eye saccade was successful then

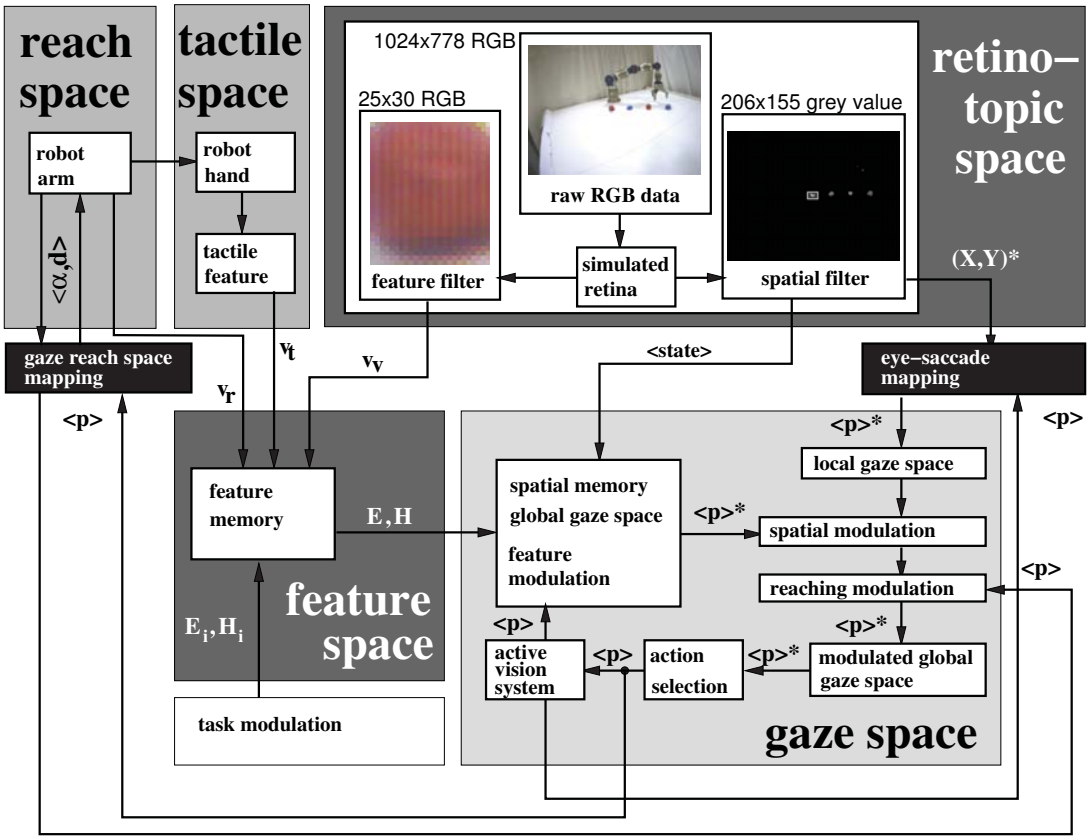


Figure 2. Computational architecture for active vision and visual guided reaching and grasping.

the absolute position p of the current active vision system (a point in the gaze space) is stored in the *spatial memory*. The domain of the spatial memory is global because it can also contain stimuli which refer to locations outside the current visual field of the camera.

The spatial memory is necessary to prevent exclusive fixations of the same stimuli (IOR). Stimuli in the spatial memory inhibit stimuli in the local gaze space having the same p-coordinates. Thus, current saliency values are modulated resulting in lower saliency values if the camera has already fixated these stimuli. The entries in the spatial memory have a decay value and get removed after a defined period of time. Therefore, the camera system will repeatedly saccade towards salient objects but will never "get caught" by the most salient object.

In addition, the intensity of the decay is modulated by the features each p is associated with. Features can be visual and non-visual and are stored in the visual memory where it is also defined how each feature modulates the decay. The parameters E and H are given for each feature class which determine the saliency of each object once the system has fixated it. Thus, after a saccade the saliency of an objects is defined by the features and their E and H values. This is what we call *task modulation* and we will explained this process later in detail.

Reaching, grasping, and tactile feedback. Having all potential fixation targets represented in the gaze space we had shown that these coordinates in gaze space can be mapped into the reach space and vice versa by a previously learned mapping [5, 6]. Thus, for any selected fixation target the corresponding reach coordinates can be derived instantly. Once the arm systems starts a reaching action the related gaze space coordinates modulate the gaze space data fed into the action selection process. In our architecture this is called the *reaching modulation* of the gaze space leading to an increased saliency of the stimuli representing the reaching target while all the other stimuli are inhibited. Consequently, the camera system remains fixated to the spot where reaching and grasping are performed until these manipulator-object interactions are finished. The continuous fixation of the target object during the whole reaching and grasping process guarantees that the current fixation point p can easily be associated with the corresponding visual v_v and tactile features v_t and the reach target coordinates v_r. These associations of p with specific visual and non-visual feature values are essential to generate a task driven visual attention and visual search.

The grasping provides tactile data. Here, we only measure the sum of all pressure values in order to indicate the "hardness" of an object. Very soft objects with pressure values lower than a given threshold are mapped to value 0 otherwise this value is 1.

3.2 Gaze space modulation

The spatial memory stores motor configurations representing fixation locations. As we have already mentioned, this is used to modulated the current visual input, in terms of highlighting task relevant visual and non-visual stimuli. In the following we provide the formal description of the modulation applied in our robotic system.

Starting at the point where the set of non-zero entries in the current saliency map $(X,Y)^*$ are transformed into a set of absolute motor po-

sitions $< p >^*$ which we also refer to as the local gaze space $\mathscr{G}_{local}$. Each p in $\mathscr{G}_{local}$ has the saliency value of the corresponding (X,Y)-coordinate. In our implementation the spatial filter is a summary of four different filter processes: three colour component and the movement filters. The final saliency value s for each pixel is calculated as follows:

$$\begin{aligned} s &= \frac{\vec{w}\cdot\vec{s}}{4} \\ &= \frac{1}{4}\cdot\begin{pmatrix} s_R \\ s_G \\ s_B \\ s_V \end{pmatrix}\cdot\begin{pmatrix} w_R \\ w_G \\ w_B \\ w_V \end{pmatrix}, \end{aligned} \tag{1}$$

where $s_i, w_i \in \mathbb{R}$ and $0.0 \leq s_i, w_i \leq 1.0$. The elements $s_{R,G,B,V}$ are the intensity values delivered by the specific filters (red, green, blue and movement). Each intensity is scaled with a given parameter w_i and finally, the normalisation guarantees that the saliency value s remains in the closed interval $[0,1]$. In the following we call this saliency values also activation values.

For any p in the local gaze space the activation value $f_o(p)$ is defined as:

$$f_o(p) = \begin{cases} s, & p \in \mathscr{G}_{local} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $s \in \mathbb{R}$ and $0.0 \leq s \leq 1.0$, the value refers to the output of the spatial filter / saliency map.

A simple inhibition mechanism for the stimuli the system has already fixated is provided by the spatial memory $\mathscr{G}_{sm}$. It stores p-coordinates representing successful saccades. Each of these coordinates is associated with an activation value f_{sm} which is determined by the the time a which is the time passed since p was added to $\mathscr{G}_{sm}$:

$$f_{sm}(p) = \begin{cases} \left(1-\frac{a}{M}\right), & p \in \mathscr{G}_{sm} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $M, a \in \mathbb{N}$, $1 \leq M$ and $0 \leq a \leq M$. Here, the variable a is the age of p while M is the maximal time a coordinate is stored in the spatial memory, both given in seconds. If $a > M$ then p is removed from $\mathscr{G}_{sm}$.

The process we call *spatial modulation* is the modulation of the activation values f_o by the activation values f_{sm}. This is done by creating a new set $\mathscr{G}_{global}$:

$$\mathscr{G}_{global} = \mathscr{G}_{local} \cup \mathscr{G}_{sm}. \tag{4}$$

We refer to the set $\mathscr{G}_{global}$ as the global gaze space since it can contain any configuration of the active vision system, in particular configurations outside the current visual field of view.

The activation values f_s for all $p \in \mathscr{G}_{global}$ are calculated as follows:

$$\begin{aligned} f_s(p) &= f_o(p) - f_{sm}(p), \\ &= s - \left(1 - \frac{a}{M}\right), \end{aligned} \tag{5}$$

where $-1.0 \leq f_s(p) \leq 1.0$ since $0.0 \leq f_o, f_{sm} \leq 1.0$.

Obviously, the resulting activation value f_s is determined by the original activation or saliency value and the time passed since the stimulus was fixated by the system. The diagram in Fig. 3 illustrates the linear change of this activation value over the time a, if the stimuli was not fixated since p has been stored in the spatial memory. One can see, when the maximal remaining time is reached ($a = M$), the activation value is back to its original value s. While at the beginning, when the stimuli was fixated and the corresponding p-coordinate stored, $a = 0$, the activation value is $(s-1)$. One can also see that the activation values can also be zero or negative at the beginning. In the diagram of Fig. 3 the two extreme cases are shown. The dashed line shows the case when the original saliency value is maximal $s = 1.0$, while the doted line represents minimal saliency values, i.e. s close to zero. Furthermore, in the diagram a threshold T, $0 \leq T \leq 1$ is indicated. T is a parameter of the action selection process. If the selected stimuli has an activation value smaller T then no eye saccade is executed. In other words p values having activation values below T are not fixated. The white region in the diagram indicates the domain of activation values which can trigger a fixation action.

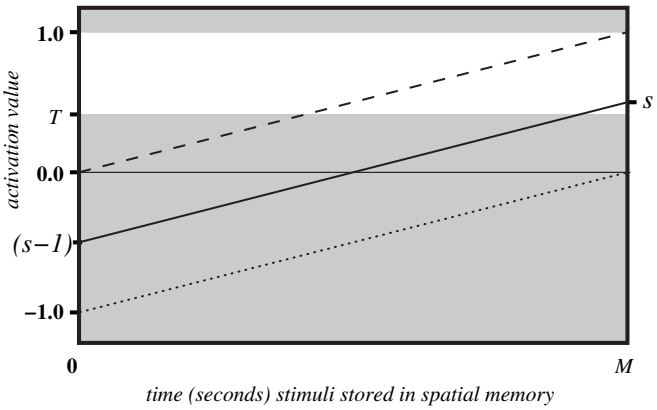


Figure 3. Activation values over time undergoing spatial modulation.

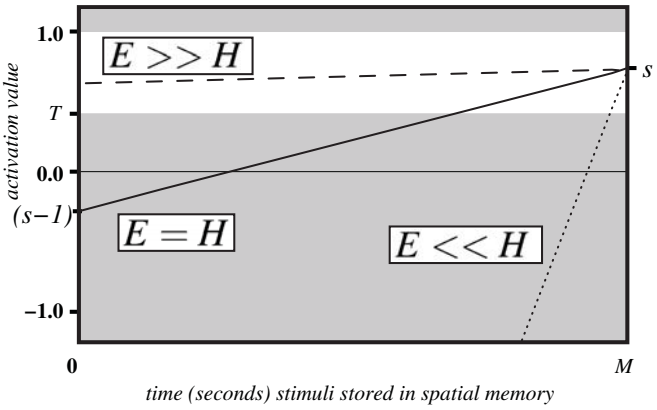


Figure 4. Activation values over time undergoing spatial and feature modulation for different excitation and inhibition levels.

The activation values f_s only consider the spatial relation. The modulation by object features includes an additional scaling factor for the activation values in the spatial memory:

$$\begin{aligned} f_f(p) &= f_o(p) - f_{sm}(p)\cdot\left(\frac{H+1}{E+1}\right), \\ &= s - \left(1-\frac{a}{M}\right)\cdot\left(\frac{H+1}{E+1}\right), \end{aligned} \tag{6}$$

where $H, E \in \mathbb{R}$ and $0 \leq H, E$. There can be different E and H values for each p. The diagram shown in Fig. 4 illustrates the evolution of activation values over time for different parameter settings of E and H while s is fixed.

Obviously, the activation value never exceeds the original activation value s generated by the spatial filter. For any parameter settings of E and H the final activation value ($a = M$) is equal s. The change of the activation values over a is determined by the ratio of H to E. For $H << E$ we get high activation values at the beginning and a gradient close to zero, while for $H >> E$ the gradient is large and the initial activation values are low. Interesting, for any given threshold $T < s$ there is always a parameter settings of E and H guaranteeing that the activation value is never below T. In such a case the corresponding stimuli p can always trigger a fixation action.

3.3 Task modulation

The parameter values E (level of excitation) and H (level of inhibition) are determined by the features associated with the corresponding stimuli p. Therefore, we call the calculation of f_f the *feature modulation*. Feature modulation can be seen as being processed on top of the spatial modulation since it is just an additional scaling of the activation values in the spatial memory.

During the interaction with an object (fixation, reaching, grasping) feature values are classified by the feature memory. Hence, while interacting with an object its location p (where data) can be linked with its features (what data). Each feature class i refers to specific excitation and inhibition values E_i and H_i. Thus, the final E and H values result from the combination of the individual excitation and inhibition values associated with p.

Since individual excitation and inhibition values *and* their combination determine the final modulation of the activation value of p, we call the assignment of E_i and H_i for each feature class and the calculation of the final E and H values for each p in the spatial memory the *task modulation*. Task modulation determines the activation values for each p and therefore, the likelihood that an object is selected for interaction (fixation, reaching and grasping). This selection probability can be influenced by the features associated with the object, which can be visual and non-visual features.

4 Experiments and results

In the following we present experiments demonstrating different types of feature modulation. In all the experiments the capacity of the spatial memory is 20 seconds, $M = 20$; while the threshold for triggering eye saccades T is fixed as well with $T = 0.1$.

4.1 Direct colour feature association

The first set of experiments shows feature modulation by colour features derived from feature vector v_v (see Fig. 2).

The system behaviour is measured in terms of fixation patterns. Over a period of 500 seconds the number of saccades and the fixation time in seconds is recorded. The fixation time is the time between two saccades. In addition, for each saccade we record the p-value and the corresponding features class. Hence, for each saccade we know which object the system has fixated, the feature classes perceived and how long the object was fixated.

Out of these data we have derived the absolute number of saccades, total fixation time and average fixation time for each object present. However, these measures are summarised with respect to the feature classes. The way we have measured fixation patterns is in accordance with [9] where fixation patterns of humans are analysed.

Before data are recorded the system is running for 100 seconds in order to let it settle for the specific parameter configurations. In this scenario four balls are placed on the table, two red and two blue ones. No reaching or grasping actions are executed. Thus, the objects are not moved. The excitation and inhibition values are pre-defined for each colour class (red [R] and blue [B]).

The results are shown in Fig. 5. The data of one parameter setting are summarised in one column. Each column represent one run over 500 seconds.

4.1.1 Spatial modulation only

The data presented in column A to G show the runs without direct feature modulation, i.e. $E = 0$ and $H = 0$. Only the weighting $\vec{w}$ in the spatial filter is different (see Eq. 1). The saliency weighting parameters are indicated by the filling of the circles labelled "saliency weighting". Black filled cycles refers to colour blue and grey to red. For column A to D we have $w_B = 1.0$ and $w_R = 0.25, 0.5, 0.75, 1.0$. This means that the blue regions in the saliency map have their maximal value, i.e. the intensity as it is perceived by the camera. While the original intensity of red regions is reduced by factor w_R. Vice versa, starting at column D we have $w_R = 1.0$, while the blue regions are scaled down following the same regime.

The different weightings in the spatial filter lead to different saliency maps with respect to the sensitivity to red and blue regions. For column A the saliency value for blue is maximal, while red regions show low saliency values. Hence, blue objects are more likely to have high activation values and therefore, are more likely to be selected as fixation target. We have the opposite in column G, where red regions are highlighted while the blue have low saliency values.

Consider the different saliency weightings in column A - G, Fig. 5. The absolute number of saccades towards a colour class $C = \{R, B\}$ increases if the weighting parameter w_R or w_B increases.

The total fixation time for a colour feature increases 1.) with the corresponding w-value *and* 2.) with decreasing w-value of the other colour class. See for instance the total fixation time for colour class red (second diagram in Fig. 5 column A - G). Here, the total fixation time increases when w_R is increased. With column D the total fixation time continuous to rise although the w_R remains the same. Therefore, the additional increase must be caused by the decrease of w_B.

Regarding the average fixation time, it seems that this value is not determined by its weighting term w. The mean value of the fixation time increases only if the weighting of the other colour class is reduced, i.e. other features become less salient.

The white labelled regions in the diagrams refer to unknown colour features. This refers to the cases where an object was not completely centered leading to the classification of the feature vector v_v as unknown (colour feature class U).

As base line for all the feature modulation experiments we selected the saliency weighting in column E, $w_B = 0.75$ and $w_R = 1.0$. Applying this saliency weighting, the system produces the most balanced response to red and blue objects. The number of saccades towards the two colour classes is nearly the same and the average fixation time is quite similar compared with the other weightings.

4.1.2 Feature modulation

Direct feature modulation was tested in three different variations that biased the system towards a particular colour class, either red or blue.

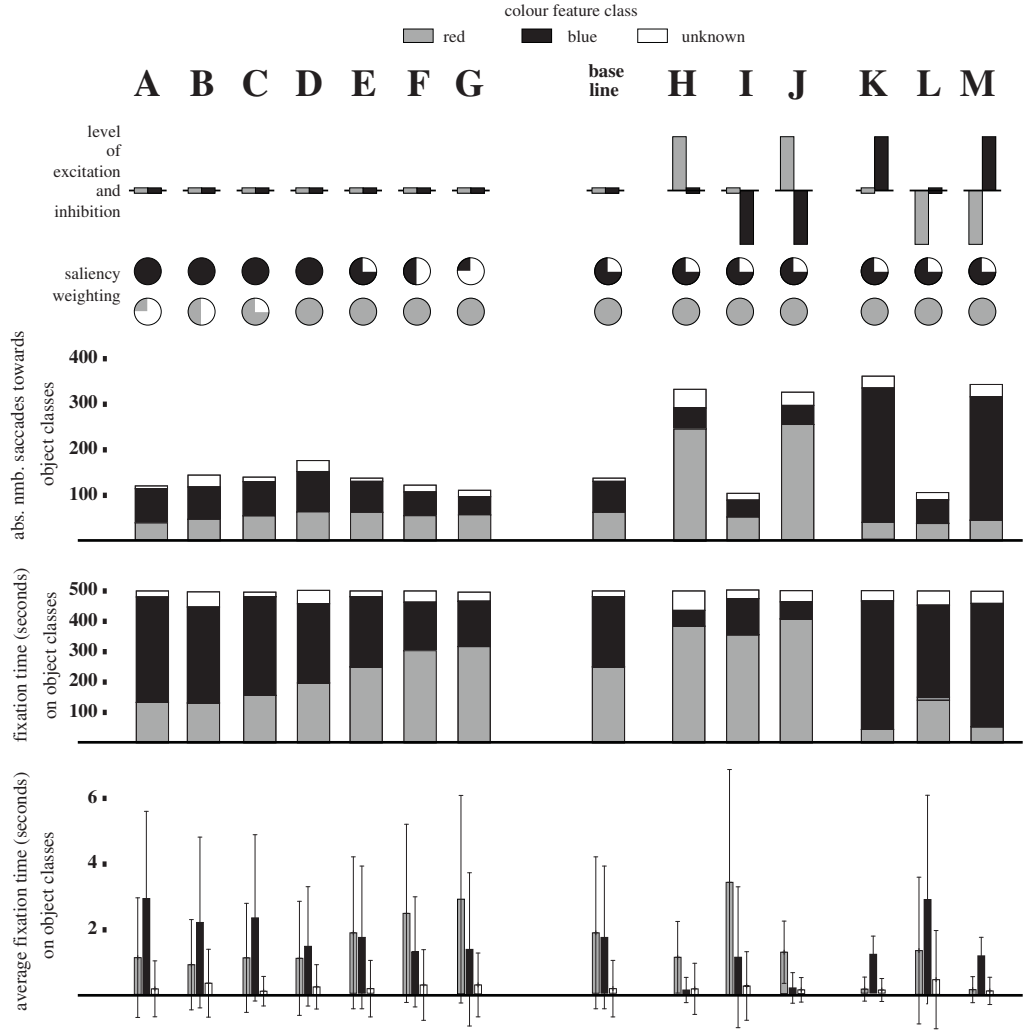


Figure 5. Bottom-up and direct feature modulation

As an example, assume we want to bias the system towards colour class red, R. This can be done by three different strategies:

1. **excitation only:** $E_R = 9$ while other E_i and H_i values are zero (column H in Fig. 5),
2. **inhibition only:** $H_B = 9$ while other E_i and H_i values are zero (column I),
3. **excitation and inhibition:** $E_R = H_B = 9$ while other E and H values are zero (column J), and

calculating the final excitation and inhibition values E and H as follows:

$$E = \sum_i E_i, \qquad H = \sum_i H_i.$$

For biasing the system towards colour feature class B (see column K, L and M) the corresponding values of E and H have to be altered accordingly.

Comparing direct feature modulation with the base line (spatial modulation only) then the fixation patterns change decisively (Fig. 5). Feature modulation in terms of the excitation of a specific feature class (column H, J, K and M) leads to a rise in the absolute number of saccades towards objects of this colour class. The total fixation time towards objects associated with the excited colour feature class increases too, while a decrease of the mean fixation time for all feature classes can be observed.

When having inhibition only (column I and L), one can see a decrease in the numbers of saccades towards the inhibited feature class. There is also a decrease of the total fixation time and a lower average fixation time, compare to the non-inhibited feature classes. With respect to the base line there appears no change in the total number of saccades, while the average fixation time and total fixation time increase for all colour classes.

Comparing the feature modulation data with each other, there seems to be no noticeable difference between doing excitation only (column H and K) and the combined excitation-inhibition strategy (column J and M).

There are also similar patterns between feature modulation data and spatial modulation. Having feature modulation by inhibition only then patterns emerge which have similarities with the two ends of the spectrum of spatial modulation data (column A and G). The data presented in column G are quite similar to the data in column I. The

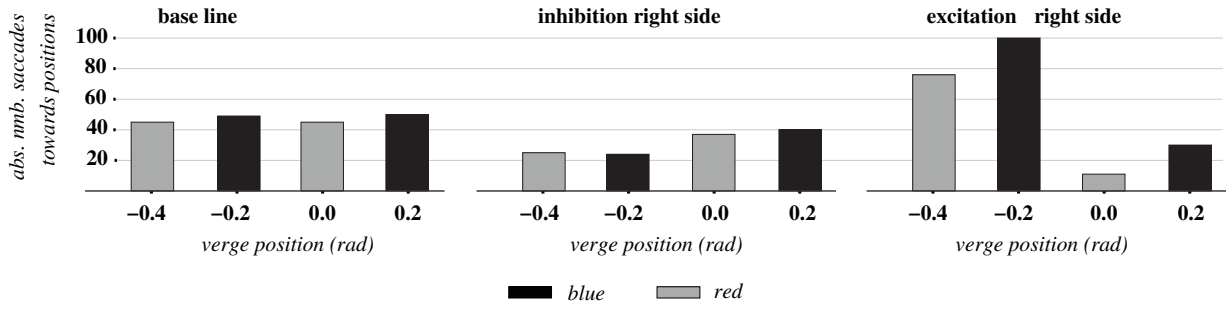


Figure 6. Absolute number of saccades towards red and blue objects while spatial habituation takes place, recording time 600 seconds, $M = 20$. **Left:** spatial modulation only. **Middle:** Inhibition of the right side in reach space, $H_{RG} = 9$ and $E_{RG} = 0$. **Right:** Excitation of the right side in reach space, $H_{RG} = 0$ and $E_{RG} = 9$.

same can be said about column A and L, where the system is biased towards blue by saliency weighting (A) and via the inhibition of red (L). Hence, saliency weighting and direct feature modulations can produce similar fixation patterns.

4.2 Direct reach space feature association

In this scenario we have use feature associations with the reach space to bias the vision system towards the "right side" (feature class RG) of the robot manipulator which defines the reach space. In this scenario on both sides of the robot arm one red and one blue object is placed. Since reach location can directly associated with gaze space location, no reaching and grasping action is executed, and therefore the scenario is static. The capacity of the spatial memory was set to 20 seconds. The saliency weighting was set up as in column E of Fig. 5, $w_B = 0.75$ and $w_R = 1.0$. There were no feature associations established between the colour classes. Feature associations were only established between the two reach space classes LE (left) and RG (right). Hence, the fixation patterns are only modulated by the direct feature associations with the reach space classes.

We present three runs, each one over a time of 600 seconds. The first provides the base line, where there is no bias towards feature class LE or RG. This is formally written as: $E_{LE} = E_{RG} = H_{LE} = H_{RG} = 0$. Thus, the final excitation and inhibition values are zero since: $E = E_{LE} + E_{RG}$ and $H = H_{LE} + H_{RG}$

In the second run, the system was biased towards the left side of the robot arm by inhibiting the right side only: $H_{RG} = 9$ while all the other values are zero. Therefore, we have $H = 9$ and $E = 0$.

Finally, in the third run we biased the system towards the robot's arm right side by excitation of the corresponding feature class RG: $E_{RG} = 9$ while all the others values are zero and therefore, we have $H = 0$ and $E = 9$.

The resulting fixation patterns are presented in Fig. 6. These diagrams show the number of saccades towards the four individual objects. The spacial position is indicated by gaze space coordinates. Absolute motor positions of the verge left motor larger $\approx 0.1 rad$ represent the left side of the robot arm (LF); while verge positions less or equal 0.1 represent the right side of the arm (RG). When inhibition or excitation of the right side takes place then the number of saccades towards the objects on the left and right side differ significantly. Since blue and red objects are on both sides it is obviously the spatial association that causes these differences. Hence, difference in the total number of object fixations between the manipulators left and right side can only be generated by non-visual spatial feature associations, not by the visual features.

4.3 Indirect tactile feature association

Cross-modal feature modulation is here demonstrated in a scenario where we have two red and tow blue balls. The blue ones are soft and the red ones hard. Hence, while grasping them they can easily be classified by the system according to the two tactile features classes *SO* (hard) and *HA* (soft). The excitation values for the tactile feature classes are set as follows: $E_{SO} = H_{HA} = 0$ and $E_{HA} = H_{SO} = 9$. which expresses a preference towards hard objects while soft should be avoided. All other excitation and inhibition values are zero. Furthermore, the saliency weighting we apply is $w_R = w_B = 1.0$. Thus, without any feature modulation the active vision system is slightly biased towards blue objects, see column D in Fig. 5.

The task of the system is to fixate an object, to pick it up and put it back on the table at a new position. As we have already mentioned in the introduction of our architecture due to *reaching modulation* the active vision keeps the object fixated while reaching and grasping are executed. After the object is placed back there is a time period of 35 seconds where the system is not allowed to trigger a reach action. After this period of time reaching and grasping actions are triggered as soon as an object is fixated. It is by chance which object is picked up since objects are repeatedly fixated. Thus, a preference of objects picked up should reflect the preference in the object fixated which is here modulated by cross-modal feature associations between colour and tactile features. Whether this is the case is tested in this experiment where objects are continuously re-located by the robot.

The system's behaviour is measured in terms of the objects picked up. Here we recorded only the first 25 reach and grasp actions (see Table 1).

Two runs are presented. In the first run, the base line, there is spatial modulation only. Consequently, object fixation and reaching driven by the saliency weighting only which has a bias towards blue objects.

In the second run cross-modal feature associations between colour and tactile feature classes are generated as well as direct feature associations with the colour classes. Other direct feature associations are not established. Therefore, a bias of fixating and picking up hard objects can only be caused by the cross-modal association between tactile and colour classes. It can not be caused by the direct features association with the tactile feature classes because they are not generated in this setup.

The results are shown in Table 1. The left part of the table shows the run of spatial modulation and the right part summarises the results for cross-modal learning. Each row indicates which object the robot picked up. The individual balls are referred to here as A and B for

each colour (red and blue). At the bottom of the table the total number of picked up objects are given for each individual object as well as for the colour class. For spatial modulation only we see that more blue than red objects are picked up. This correlates with the bias towards blue objects generated by the saliency weighting. If cross-modal learning takes place (right part of the table), then more red than blue objects are picked up, because the system has learned to associate the red objects with "hard tactile feedback" which leads to a highlighting of red objects due to feature modulation.

Table 1. Results of cross-modal learning

	base line				cross-modal			
	red object		blue object		red object		blue object	
grasp action	A	B	A	B	A	B	A	B
1			×		×			
2	×					×		
3	×					×		
4	×					×		
5				×	×			
6			×		×			
7				×			×	
8	×					×		
9			×		×			
10				×	×			
11				×			×	
12	×							×
13			×					×
14				×	×			
15		×			×			
16				×			×	
17	×							×
18				×				×
19		×						×
20				×		×		
21	×						×	
22			×			×		
23				×	×			
24			×			×		
25		×			×			
total	7	3	6	9	9	7	4	5
total	**10**		**15**		**16**		**9**	

5 Conclusion

We started with the assumption that if visual attention is task dependent it ought to be modulated by visual and non-visual features. In this work we have introduced a computational architecture for a robotics active vision system equipped with a manipulator that is able to demonstrate such a modulation of visual attention in terms of observable fixation patterns.

Central element of this architecture is the usage of the gaze space defined by the active vision motor system. This gaze space approach is computational very efficient because it provides a global reference frame for object locations. When applying the gaze space the amount of data needed to create a global reference frame is several orders of magnitude less compared with a retinotopic reference frame [1, 10].

The gaze space also provides a robust computational substrate for the integration of what (visual and non-visual features) and where (location) information. Having solved the problem of synchronisation of visual and non-visual what- and where-data in general, we are able to separate feature filtering from spatial filtering in the visual data (similar to the human retina). This separation has additional computational advantage and better scalability when applying visual feature filters of higher complexity and computational costs.

All these points make our architecture a promising reference framework for the modelling and engineering of robotics active vision and visual guided reaching and grasping. Nonetheless, the current robotic implementation presented shall be seen as only the first complete system which validates the our computational architecture. The modularity of the architecture allows to replace all the modules by more advanced processes.

First candidate of such replacements are the modules of the visual feature and spatial filter. They could be replaced by other and more comprehensive methods without changing the essentials of this architecture.

Furthermore, instead of pre-defining feature classes in the feature memory they could be the result of self-organised learning processes.

The action selection process we applied is a simple winner-takes-all mechanism, which is currently subject of being replaced with a more effective process based on an artificial neural network implementation [2].

All the feature associations, the modulation processes of the gaze space and the mappings could also be subject to be implemented by a different computational paradigm, e.g. by artificial neural networks.

Finally, future developments of this architecture needs to address the representation of more complex objects. Here, we are able to integrated visual and non-visual multi-modal sensorimotor experience. However, for more complex objects more visual and non-visual features must be processed which will include hierarchical organisations providing different combinations of low-level features.

ACKNOWLEDGEMENTS

This work was supported by the EC-FP7 projects IM-CLeVeR and ROSSI, and through UK EPSRC grant EP/C516303/1.

REFERENCES

[1] F. Alexandre, 'Cortical basis of communication: Local computation, coordination, attention', *Neural Networks*, **22**, 126–133, (2009).
[2] R. Bogacz and K. Gurney, 'The basal ganglia and cortex implement optimal decision making between alternative actions', *Neural Computation*, **19**(2), 442–477, (2006).
[3] F. Chao, M. H. Lee, and J. J. Lee, 'A developmental algorithm for ocular-motor coordination', *Robotics and Autonomous Systems*, **DOI information: 10.1016/j.robot.2009.08.002**, (2010).
[4] M. Hülse, S. McBride, J. Law, and Mark Lee, 'Integration of active vision and reaching from a developmental robotics perspective', *IEEE Transactions on Autonomous Mental Development*, **2**(4), to appear, (2010).
[5] M. Hülse, S. McBride, and M. Lee, 'Robotic hand-eye coordination without global reference: A biologically inspired learning scheme', in *In: Proc. Int. Conf. on Developmental Learing 2009, China, 2009, IEEE Catalog Number: CFP09294*, (2009).
[6] M. Hülse, S. McBride, and Mark Lee, 'Fast learning mapping schemes for robotic hand-eye coordination', *Cognitive Computation*, **2**(1), 1–16, (2010).
[7] L. Itti and Ch. Koch, 'A saliency-based search mechanism for overt and covert shifts of visual attention', *Vision Research*, **40**, 1489–1506, (2000).
[8] B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic, 'An active vision system for detecting, fixating and manipulating objects in the real world', *the International Journal of Robotics Research*, **29**(2-3), 133–154, (2010).
[9] C. Rothkopf, D.H. Ballard, and M. M. Hayhoe, 'Task and context determine where you look', *Journal of Vision*, **7**, 1–20, (2007).
[10] J. Vitay, N.P. Rougier, and F. Alexandre, 'A distributed model for spatial visual attention', *In: Wermter et al. (Eds.): Biomementric Neural Learning, LNAI 3575, Springer*, 54–72, (2005).

A visual novelty detection filter based on bag-of-words and biologically-inspired networks

Y. Gatsoulis[1] and **E. Kerr** and **J.V. Condell** and **N.H. Siddique** and **T.M. McGinnity**

Abstract. The ability of a robot system to learn continuously until the end of its cycle is a desired feature and consists a difficult challenge for the robotics research community. One of the main components necessary for effective continuous learning is the behaviour of a robot of identifying and focusing its attention to novel patterns, and has been an active area of research over the last decade, considering the large number of surveys that have been published recently.

This paper presents the initial steps of a larger work which is concerned with continuous learning driven by novelty detection as an intrinsic motivation. For the learning structure and the novelty detection filter we use a bag-of-words model combined with unsupervised biologically inspired neural networks, both for the generation of the vocabulary and the learner/classifier.

1 INTRODUCTION

One of the main challenges in artificial intelligence is to build robotic systems that are capable of continuous operation and learning of new skills. Key research issues for the realisation of such artificial agents are knowledge representation structures and methods that support cumulative learning and novelty detection.

In its primitive form the problem of novelty detection is to identify new, novel patterns that have never been seen before [8, 10, 15]. It consists an important ability of a number of biological cognitive organisms as it reduces computational load by selecting and guiding attention to areas of "interest", and it has seen an increasing interest in the last decade considering the number of works and surveys that have been recently published [8, 9, 10, 2, 4].

A more formal description of the problem of novelty detection is as follows. An agent is trained on a set of perceptual patterns $X = x_1, x_2, \ldots, x_n$ using a training method F and forming a knowledge database $K = F(X)$. At time t an observation o is considered novel if it differs significantly from what is already known, i.e. from K, using a novelty detection filter N to identify the level of novelty and the particular parts that are novel. The observation o is then inserted in the training set X as a new training pattern x_k, updating the agent's knowledge K. In the majority of cases the novelty detection filter and the training methods are one and the same ($N \equiv F$) because the diversity error of o with K, computed and needed by the learning step of F, is considered to be the novelty metric.

Previous published work [3] has discussed some key issues and characteristics needed for the effective operation of novelty detection filters in cumulative learning tasks, and also compared different categories of novelty detection methods as presented from the recent surveys. In particular it was identified that the novelty detection filters should operate online, and be dynamic and expandable.

This paper presents the first steps of a biologically inspired novelty detection filter based on previous work with habituated self-organising maps, combined with a bag-of-words classifier of images.

[1] University of Ulster, Intelligent Systems Research Centre, UK, email: i.gatsoulis@ulster.ac.uk

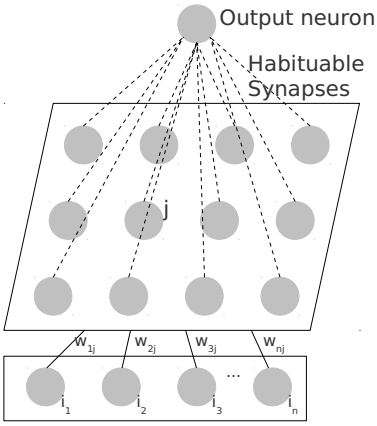


Figure 1. Habituated self-organising map (image adopted from [11])

2 METHODS

2.1 Habituable Neural Networks

A habituable neural network is a novelty filter that has the purpose of learning to recognize known features and evaluate their novelty based on the frequency with which these input stimuli have been seen recently was presented in [11]. It is based on a Kohonen map with habituated synapses linking the nodes of the network to an output node, as shown in Figure 1. A habituated synapse decreases in strength as its connected nodes fire, and increases in strength when already known stimuli are not seen for some time. The behavioural phenomenon of habituation has its roots in biology, and as mentioned by [11], cross-citing [17], it is thought to be one of the simplest forms of plasticity in the brain of a large number of organisms.

The model of habituation and dishabituation being used in this work is the one suggested by Stanley [16] who used the first-order differential equation shown in Equation 1.

$$\tau \frac{dh(t)}{dt} = \alpha[h_0 - h(t)] - S(t) \quad (1)$$

where,

h_0 : is the initial value of the habituation level.

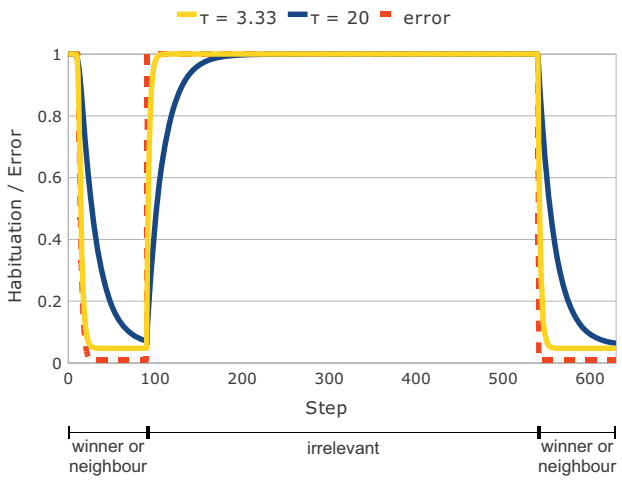


Figure 2. Behaviour of a node's habituation for different values of the habituation constant τ

τ, α : are constants controlling the habituation and dishabituation rate.
$S(t)$: is the activity of the unit.

This habituation model was integrated in unsupervised learning with self-organising maps (SOM) and successfully validated in earlier related experimental work [12], in which a robot learns the sonar representation of corridor environments. In these experiments the robot was able to learn and habituate over the training environment, but when a feature was changed and not observed for some time, then the novelty filter showed increasing levels of novelty over this particular feature.

The habituated SOM works as follows. Each unit of the SOM has an associated habituation variable that changes according to Equation 1. The habituation variable decreases when the winner unit of the SOM and its neighbours that best match the perceived object fire, while the habituation variables of the rest "irrelevant" units remain either unaffected, or increase so that the objects they represent are being forgotten. This is achieved by propagating the activity of the unit $S(t) = e^{-||\xi - w_s||}$, where ξ is the input pattern and w_s is the weights vector of the unit s [13]. This behaviour is graphically shown in Figure 2 by the blue or the yellow straight line. Regarding the rates of habituation and dishabituation, these are controlled by the two constant variables τ and α respectively. It is mentioned in [11] that in their experiments a value of $\tau = 3.33$ reduces the habituation value to 90% of its original value within 5 iterations. Figure 2 shows that when τ increases, the system takes longer to habituate on a perceived pattern.

We have initially experimented with habituated neural networks with visual systems [3]. In this paper we extend it as we combine the habituated neural networks with a bag-of-words model in vision, more specifically the classifier part of the bag-of-words. The bag-of-words model is explained next.

2.2 Bag-of-words model for vision

The bag-of-words (BoW) model has its roots in natural language processing where it was used to represent and classify documents according to the frequency of particular words existing in a dictionary. The produced histograms are then the representations of the documents.

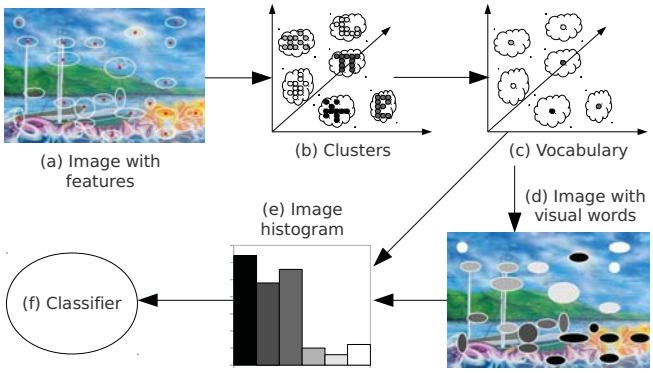


Figure 3. Bag-of-words technique

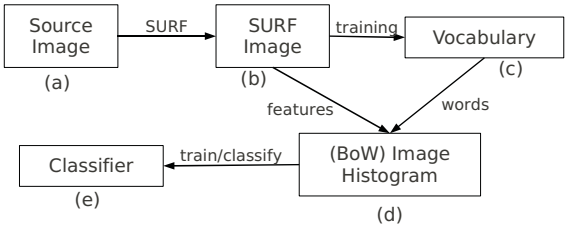


Figure 4. Experimental procedure

The BoW technique has been adopted by the machine vision research community to describe and classify images in the same manner, by using histograms of the frequencies of "visual words" from a dictionary that exist in the image.

The bag-of-words technique consists of the following generic steps, shown in Figure 3.

1. Extract a set of feature descriptors, such as SIFT, SURF, etc., from a perceived image (Figure 3.(a)).
2. Learn a visual vocabulary (Figure 3.(b) & (c)), by training an unsupervised structure (e.g. k-means, SOM, etc.) to the extracted features of the perceived image.
3. For a given vocabulary and a set of feature descriptors of an image, compute the histogram of the frequency of visual words that match these feature descriptors (Figure 3.(d) & (e)).
4. Train a classifier (e.g. support vector machine, SOM, etc.) with the produced histograms (Figure 3.(f)).

2.3 Experimental procedure

The experimental procedure used is described next, and it is also shown schematically in Figure 4.

1. The source images (Figure 4(a)) are drawn from three different categories of objects, these being 32 forks, 22 mugs and 8 plates (see Figure 5). All images are rectangular and rescaled to a fixed resolution of 300×300 $pixels$. Three quarters of them are used for training (24 forks, 16 mugs and 6 plates), and the remaining one quarter is used for validation purposes (8 forks, 6 mugs and 2 plates).
2. The SURF (Speeded Up Robust Features) [1] descriptors are identified next (Figure 4(b)). These descriptors are scale and rotational

Figure 5. Sample objects

invariant. The number of features was between 3-122 ($mean = 28$) for the forks, 81-270 ($mean = 142$) for the mugs, and 63-454 ($mean = 209$) for the plates.

3. The SURF descriptors are then used for training the dictionary of visual words ((Figure 4(c)), which is Kohonen SOM. The size of the SOM is 10×10, and it is sufficient for this small number of objects, however, it is noted that a dynamically expandable learning structure is needed for long-term operation and learning of many objects. A radius of 1 unit around the best-matching unit was selected as the size of the training neighbourhood, in order to avoid destructive learning as much as possible. Finally, a learning rate of $\eta = 0.2$ was selected [14].
4. A histogram representing the image in terms of the frequency of the visual words from the dictionary that correspond to each one of the SURF descriptors of the image is generated (Figure 4(d)).
5. The generated image histograms are used for training a classifier, which is a habituable neural network of size 10×10, and trained with a fixed learning rate $\eta = 0.2$, and a training neighbourhood of radius of 1 unit around the best-matching unit, similarly to the vocabulary. Training proceeds until the system is habituated with all perceptions, controlled by a fixed threshold set to 0.2. The habituation levels are calculated according to Equation 1, with $h_0 = 1$, $\alpha = 1.05$ and $\tau = 3.33$, similarly to [12].
6. The system is validated using the validation set with learning switched off. Performance is measured by checking whether the winner node and its neighbourhood of radius of 1 unit corresponds to a cluster representing the object correctly. This is done using a simple voting mechanism we employed. In more detail this is as follows, by counting the number of nodes in the winner's neighbourhood that correspond to winner nodes in the training set. So for example, for one mug in the validation set there were 3 nodes that correspond to winner nodes of mugs from the training set, while 1 node corresponded to a fork training image. As such the "classification confidence" would be in this case $3/4 = 0.75 > 0.5$, and hence classified correctly as a mug.

3 RESULTS

The training of the classifier was completed after 2899 epochs, with 5 iterations over the complete training data until all habituation levels dropped below the fixed threshold (0.2). It was then validated with the validation data set according to the procedure described in Section 2.3, step 6. The classification performance of the habituated neural network is shown in Table 1.

As the results show the network has an accuracy of 81.25%. All the forks were classified correctly and with absolute confidence

Table 1. Validation data (F: Fork, M: Mug, P: Plate)

Name	Nodes (F:M:P)	Confidence (p)	Classification
Fork 1	2:0:0	1	✓
Fork 2	8:0:0	1	✓
Fork 3	1:0:0	1	✓
Fork 4	6:0:0	1	✓
Fork 5	4:0:0	1	✓
Fork 6	6:0:0	1	✓
Fork 7	2:0:0	1	✓
Fork 8	4:0:0	1	✓
Mug 1	1:3:0	.75	✓
Mug 1	1:3:0	.75	✓
Mug 2	1:2:0	.67	✓
Mug 3	0:4:0	1	✓
Mug 4	3:1:0	.25	✗ – fork
Mug 5	0:1:2	.33	✗ – plate
Mug 6	0:4:0	1	✓
Plate 1	0:2:1	.33	✗ – mug
Plate 1	0:0:1	1	✓
Forks		1	100%
Mugs		.665	66.67%
Plates		.665	50%
Total		.833	81.25%

$p = 1$. Both in the mugs and the plates categories were some mis-classifications and even when the objects were correctly classified the confidence score for the mugs, where there is more than once case, it has not always been an absolute agreement. In fact as it is seen from the results plates can be mis-classified as mugs and vice-versa. A possible explanation for these mis-classifications is that some of the mugs and some of the plates may have similar "decorations", and therefore share some common feature descriptors. The inability to tell with confidence what these objects are, signify that these objects are novel. The robot can learn about them, and starting with a high value of dishabituation it will be trained with these patterns as inputs until it habituates on them, i.e. until the robot becomes "bored" on observing and learning these patterns.

Overall, it can be said that despite the few mis-classifications the system has performed satisfactory and provides a promising start for further expansions.

4 DISCUSSION

This paper has presented a bag-of-words model where the vocabulary and the learner/classifier are replaced with unsupervised learning equivalents. Specifically, the vocabulary was a Kohonen SOM, and the classifier was a habituated neural network [11].

The most related work is that of Kinnunen et al. [5, 6, 7], which describes a bag-of-words model based on unsupervised Kohonen maps. The model that was implemented is a typical bag-of-words model except that Kohonen self-organising maps were used for learning the visual vocabulary. In their previous work [6] they have shown that the SOM approach outperforms the typical k-means algorithm for vocabulary generation.

The work presented in this paper can be seen as an extension to Kinnunen et al. research, where we are using a habituated self-organising map as the learner/classifier of a bag-of-words model; the habituation behaviour directs the robot to learn novel areas. It consists initial steps in a bigger project investigating cumulative learning using novelty detection filters as the driving mechanisms for exploratory learning.

We are planning to investigate next the effects of online generation of the vocabulary with simultaneous online training of the classifier

with a real robot manipulator equipped with a camera.

ACKNOWLEDGEMENTS

This work was supported by EU FP7-ICT-2007-3: 2.2 Cognitive Systems, Interaction and Robotics PROJECT: IM-CLeVeR (2009-2013).

REFERENCES

[1] H Bay, A Ess, T Tuytelaars, and L Vangool, 'Speeded-Up Robust Features (SURF)', *Computer Vision and Image Understanding*, **110**(3), 346–359, (June 2008).
[2] V. Chandola, A. Banerjee, and V. Kumar, 'Anomaly detection: A survey', *ACM Computing Surveys*, **41**(3), 1–58, (2009).
[3] Y. Gatsoulis, E. Kerr, J.V. Condell, N.H. Siddique, and T.M. McGinnity, 'Novelty detection for cumulative learning', in *Proc. of Towards Autonomous Robotic Systems 2010 (TAROS'10)*, pp. 62–67, Plymouth, UK, (2010).
[4] V. Hodge and J. Austin, 'A survey of outlier detection methodologies', *Artificial Intelligence Review*, **22**(2), 85–126, (October 2004).
[5] Teemu Kinnunen, *Unsupervised visual object categorization*, Ph.D. dissertation, Lappeenranta University of Technology, 2008.
[6] Teemu Kinnunen, JK Kamarainen, Lasse Lensu, and H, 'Bag-of-features codebook generation by self-organisation', in *Advances in Self-Organizing Maps*, (2009).
[7] Teemu Kinnunen, Joni-Kristian Kamarainen, Lasse Lensu, and Heikki Kalviainen, 'Unsupervised Visual Object Categorisation via Self-organisation', in *2010 20th International Conference on Pattern Recognition*, pp. 440–443. IEEE, (August 2010).
[8] M. Markou and S. Singh, 'Novelty detection: a review part 1: statistical approaches', *Signal Processing*, **83**(12), 2481–2497, (December 2003).
[9] M. Markou and S. Singh, 'Novelty detection: a review part 2: neural network based approaches', *Signal Processing*, **83**(12), 2499–2521, (December 2003).
[10] S Marsland, 'Novelty detection in learning systems', *Neural Computing Surveys*, **3**, 157–195, (2003).
[11] S. Marsland, U. Nehmzow, and J. Shapiro, 'A real-time novelty detector for a mobile robot', in *Proc. of the EUREL Conference on Advanced Robotics Systems*, (2000).
[12] S. Marsland, U. Nehmzow, and J. Shapiro, 'On-line novelty detection for autonomous mobile robots', *Robotics and Autonomous Systems*, **51**(2-3), 191–206, (May 2005).
[13] S. Marsland, J. Shapiro, and U. Nehmzow, 'A self-organising network that grows when required', *Neural Networks*, **15**(8-9), 1041–1058, (October 2002).
[14] U. Nehmzow, *Mobile Robotics: A Practical Introduction*, Springer–Verlag, London, 2nd edn., 2003.
[15] R. Saunders and J.S. Gero, 'The importance of being emergent', in *Proc. of the Conference on Artificial Intelligence in Design*, (2000).
[16] J. C. Stanley, 'Computer simulation of a model of habituation', *Nature*, **261**(5556), 146–148, (May 1976).
[17] R. F. Thompson and W. A. Spencer, 'Habituation: A model phenomenon for the study of neuronal substrates of behaviour', *Psychological Review*, **73**(1), 16–43, (1966).

A modular reinforcement learning model for human visuomotor behavior in a driving task

Brian Sullivan*
brians@mail.utexas.edu

Leif Johnson†
leif@cs.utexas.edu

Dana Ballard†
dana@cs.utexas.edu

Mary Hayhoe*
mary@cps.utexas.edu

March 7, 2011

Abstract

We present a task scheduling framework for studying human eye movements in a realistic 3D driving simulation. Human drivers are modeled using a reinforcement learning algorithm with "task modules" that make learning tractable and provide a cost metric for behaviors. Eye movement scheduling is simulated with a loss minimization strategy that incorporates expected reward estimates given uncertainty about the state of environment. This work extends a previous model that was applied to a simulation of walking; we extend this approach using a more dynamic state space and adding task modules that reflect the greater complexity in driving. We also discuss future work in applying this model to navigation and fixation data from human drivers.

1 Introduction

Humans formulate and execute complex visuomotor action sequences while performing real-world tasks like driving or playing sports. Previous work has explored the role that visually "salient" [5] features play in making saccades, but this research has focused largely on 2D images or videos where human subjects are observing the scene and not actively participating in a visuo-motor task. In contrast, when performing tasks in natural environments, humans interact with the world and high-level cognitive goals and reward [3, 6] play an important role in the execution of eye movements. However, the mechanisms underlying these task-driven saccades are not well understood.

This paper presents a high-level, task-based scheduling framework for studying human eye movements in a realistic, 3D driving simulation. Our primary aim is to present an abstract framework for interpreting human eye movement behavior that explicitly represents task demands, reward and perceptual uncertainty. This approach allows modeling of visual behavior over long time scales that has not been typically addressed in vision science. Our model is quite abstract in that no image processing is used and major simplifications are made to ease the process of modeling driving behavior. The model is still in development and we focus here on providing a technical report of our methodology and a review of the current state of ongoing research.

We model human drivers computationally using a reinforcement learning algorithm that breaks the complex state space of driving into several "task modules" that make learning com-

*Department of Psychology, University of Texas at Austin

†Department of Computer Science, University of Texas at Austin

putationally tractable [7]. Modules also provide a cost metric that allows direct comparison of the relative values of different behaviors. In our model, eye movement scheduling attempts to minimize the expected loss of reward given the current knowledge of the state of the world and the uncertainty in the state estimate. From a high level, eye movements are directed toward targets in order to reduce uncertainty about potentially high-reward portions of the state space.

2 Background

Prior research suggests that although human vision has massively parallel inputs from the retina, due to attentional and memory limitations many visuo-motor computations are serial [3]. Studies have found that humans often employ active vision strategies of gathering specific and discrete pieces of visual information as a task develops [4, 2]. These data suggest that one approach to model goal directed human vision is to use serial "visuo-motor task modules," sometimes referred to as visual routines [1, 11]. These modules perform very specific computations in isolation (e.g., finding a road landmark to control steering), but when coordinated over time with other modules, complex behaviors can be achieved.

With this type of approach, a scheduling problem arises: What visuo-motor computations should be carried out and when should they be executed? Our work presents one solution to this scheduling problem by using reward values and uncertainty to solve the arbitration of visual computations. This work extends a previously developed reinforcement learning model [9] that has been successfully applied to a simulated three-task walking world with static obstacles and goals.

Sprague and Ballard simulated a humanoid walking down a sidewalk with obstacles and "litter" to be picked up. Their algorithm has distinct perceptual and motor components. Visual computations were broken down into components for avoidance of obstacles, "picking up" items and sidewalk following. Each of these modules has a dedicated visual computation that finds the distance and angle to the sidewalk, obstacles and litter. The motor system uses this state information to navigate (turn left, turn right or go straight) using a control policy learned via reinforcement learning. Only one visual module at a time can run to get a new update of state information. Idle modules are allowed to update their representations via a Kalman filter, introducing uncertainty into their state estimates. The perceptual arbitration system selects a module to be updated with new sensory information. Crucially, the perceptual arbitration algorithm uses reward estimates from the motor component and estimates of state uncertainty in the perceptual system to choose which module to update.

The present research applies a similar methodology to a simulated driving task. In comparison to walking, the driving task requires a more complex and dynamic state space and has more task modules to address the greater variety of available tasks while driving. After briefly introducing reinforcement learning and describing the driving simulation, we present some preliminary results and then conclude with a description of future work in applying this model to navigation and fixation data from human drivers in a realistic 3D car simulation.

3 Reinforcement Learning

Reinforcement learning (RL) [10] is a goal-focused learning framework that directly models the interaction between learner and environment. RL finds a mapping between a current environmental state and an appropriate action to execute in that state. In our application, we use RL to find a control policy that maps environmental states to actions to control steering and velocity of a simulated car. Our specific implementation of the state and actions spaces is presented in section 4.

Here we present a brief primer on the RL framework, focusing on the Q-learning [12] vari-

ant of the algorithm. A learning agent (LA) maintains a vector s_t of discrete variables describing the state of the world over a series of discrete time steps $t = 1 \ldots T$. At each time step, the LA chooses a discrete action a_t that will maximize the available reward. Positive or negative reinforcement r_t is given to the LA whenever s_t is a state that achieves some goal or subgoal, specified by the modeler as part of the construction of the world. The LA receives supervision only in the form of these explicit reward values, which are often nonzero only for a small fraction of world states.

During training, the LA constructs an exhaustive Q table $Q(s_t, a_t)$ of the expected rewards that are attainable by taking each action from each state in the world. If the LA takes an action a_t when the world is in state s_t, it observes the resulting state s_{t+1} and its associated reward r_{t+1} on the following time step. Using a learning rule, the LA can then update $Q(s_t, a_t)$ so that over time this Q value becomes closer to the "expected future reward" for (s_t, a_t). The LA adjusts the Q values by following the gradient of the error in Q:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \Delta Q(s_t, a_t)$$

where $\alpha \in [0, 1]$ is a learning rate parameter (set to 0.2 for our simulations) and ΔQ is the direction of the greatest observed change in Q at (s_t, a_t).

The optimal Q values reflect both the immediate reward in a state, and all future rewards attainable from that state, discounted exponentially by the number of time steps required to reach those future states. Thus ΔQ takes the form

$$\Delta Q(s_t, a_t) = r_{t+1} + \gamma \hat{Q}(s_{t+1}) - Q(s_t, a_t)$$

where r_{t+1} is the reward available in state s_{t+1} (which follows state s_t after taking action a_t) and γ is a parameter called the discount factor. Values of γ near 0 cause the LA to rely more on immediate rewards for the Q values, while values near 1 blur the distinction between immediate and future reward, allowing the agent to postpone immediate rewards for potentially larger future rewards. For our simulations, we set $\gamma = 0.95$.

A simple yet powerful learning rule for $\hat{Q}(s_{t+1})$ is simply the Q value associated with the subsequent action selected by the agent:

$$\hat{Q}(s_{t+1}) = Q(s_{t+1}, a_{t+1})$$

This rule, called SARSA learning, ensures that, along any given sequence of state/action pairs that are actually chosen by the agent, the expected rewards obey the discounting enforced by the γ parameter. Our simulation uses the Q values to choose an action using a softmax rule, where the probability of choosing action a_t when the world is in state s_t is given by

$$p(a_t|s_t) = \frac{\exp\left(Q(s_t, a_t)\right)}{\sum_a exp\left(Q(s_t, a)\right)}.$$

3.1 GM-SARSA

Traditional RL operates within a single joint state space that must represent all task-relevant aspects of the world simultaneously. Because the LA must visit each state/action pair multiple times during learning to formulate an accurate estimate of the Q values, a large state space leads to slower convergence during learning. In complex environments, RL is much more efficient if a learner is allowed to focus just on the state variables that are relevant for a particular task. Instead of running the driving simulation in a joint state space that represents all possible variables of interest simultaneously, we used the technique GM-SARSA [8] to split the world into small task modules.

In GM-SARSA, each task module $i = 1 \ldots N$ has a separate state space and Q table, $Q^i(s_t^i, a_t)$, but the tasks share a common action space. When the LA needs to select action a_t, it uses the state estimates $s_t^1, \ldots, s_t^N$ for each task to retrieve the corresponding vectors of Q values $Q^1(s_t^1, \cdot), \ldots, Q^N(s_t^N, \cdot)$. These vectors are

summed, and the result

$$Q^*(a_t) = \sum_{i=1}^{N} Q^i(s_t^i, a_t)$$

is used in the decision rule to select the best action.

The SARSA learning rule maintains the correctness of task learning with multiple modules. Because the action a_t is shared among all modules, the Q tables can be updated correctly even though a_t might not have corresponded to the highest-reward action for any of the individual task modules.

4 Modular RL for Driving

Our driving world consists of $C \approx 20$ cars that drive in the lanes of a simulated world including a four-lane road (two lanes in each direction), cars, and pedestrians. Two of the cars in the world have special roles: car 1 is controlled by the learning agent in the simulation, and car 2 is called the "pace car" and is described in more detail below. Cars $3 \ldots C$ serve mostly as obstacles for the learning agent. Figures 1 and 4 show screenshot of the simulations, before and after we have projected it into the human driving environment, respectively.

All non-learning agent cars move in the same direction and are constrained to drive along one of two tracks that represent the two available lanes on the road. Each car thus maintains three scalar variables that describe its state in the world: δ_c represents the distance (in meters) traveled along the track by car c, σ_c represents the speed (in meters per second) of the car on its track, and $\lambda_c \in \{0, 1\}$ represents the lane that car c currently occupies. In the 3D simulation for humans, described below, these scalars are mapped to lane positions in a realistic virtual world.

All agents other than the learner move at a fixed speed along one track, but these states change randomly on average every 1000 time steps to prevent the LA from overlearning a static world. When choosing new values, cars $2 \ldots C$ draw a new speed uniformly from $[0, \Sigma]$ (where Σ is the maximum speed allowed for any car, set to 20 m/s in our simulations) and a new track uniformly from $\{0, 1\}$.

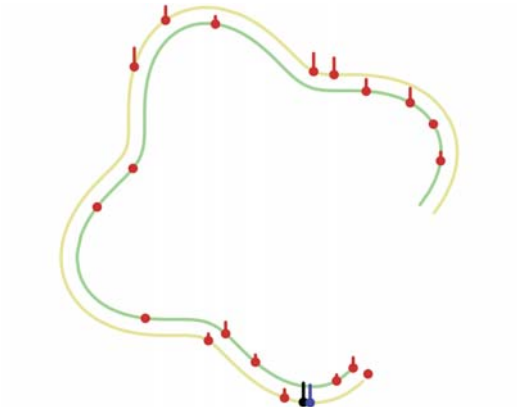


Figure 1: The simulated RL driving world consists of the LA (black dot), a pace car that the LA is rewarded for following (blue dot), and several other agents that the LA is punished for hitting (red dots). Each car has a "flag" whose length indicates the car's speed. Lanes for driving are shown as curvy colored lines, even though to the RL agents the lanes are one-dimensional.

4.1 Task Modules

The RL model uses several modules coordinated over time to drive, limited to a basic set that could be applied to data from human drivers. These modules are dedicated to tasks for avoidance, following another car, and simply driving forward. Figure 2 shows a graphical representation of the state space used by each module discussed below. Additionally, Figure 3 (page 8) shows a high level overview of the scheduling model and how modules are coordinated.

4.1.1 Forward Progress

With predetermined lanes, the LA is encouraged to move around the track by a task module that

provides a small positive reward R_Σ whenever the LA is moving at speed greater than $\frac{\Sigma}{2}$. Without this task module, the LA tends to stop moving, which no humans do in the 3D driving simulator. The state space for this task is simply the speed of the LA, divided into N_σ uniformly spaced bins. In active lane following, the LA keeps track of the relative speed between the closest portion of road, the angle of this scalar, the distance to the road and the angle between the LA and the road.

4.1.2 Car Following

The LA receives a positive reward R_f for following the pace car at a fixed distance of 10 m, with a relative speed of 0 m/s (i.e., whenever the LA is following behind the pace car and both cars are going the same speed). The state space for this module consists of three dimensions: the lane indicator, the relative distance, and the relative speed. The lane indicator is an ordered pair from $\{0,1\} \times \{0,1\}$ that represents the lanes for the LA and the pace car. The relative distance is given by $\max(\min(\delta_2 - \delta_1, D), -D)$, where D is a constant (set to 200 m in our simulations) that represents the maximum distance the LA can discriminate. This dimension is quantized into N_δ uniformly spaced bins. Similarly, the relative speed is given by $\max(\min(\sigma_2 - \sigma_1, \Sigma), -\Sigma)$ and is quantized into N_σ uniformly spaced bins.

4.1.3 Car Avoidance

The LA receives a negative reward R_c for colliding with any of the other cars in the world. A world state is considered a collision whenever

$$|\max(\min(\delta_* - \delta_1, D), -D)| < \frac{2D}{N_\delta}$$

and the relative speed between the LA and the obstacle is less than 0. This task uses the same state space as the following task described above. The driving simulation includes one task module that tracks the closest obstacle (including the pace car) to the LA at every time step, but could easily include more such modules representing the states of the next-closest obstacles.

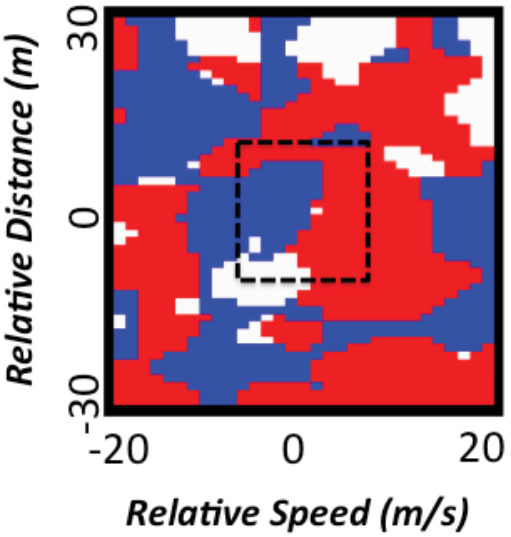


Figure 2: A learned policy for pedal control in the following task when the LA and Pace car are in the left lane. Blue indicates slow down, red speed up and white do nothing. The box with dashed lines indicates the portion of state space most frequently visited by the LA.

4.2 Action Space

While each module tracks different information in the world, they share the same set of actions. Given the current state of the world, the LA can read out the Q estimates for each module and evaluate the optimal action via GM-SARSA. The action space for each module contains a steering component and a velocity component. The actions are discretized such that steering control has three options: staying in the same lane, changing to the right lane, or changing to the left lane. Similarly, velocity control features three options: speed up, stay at the same speed, or slow down.

4.3 Training and Evaluation

Training takes place over 1000's of episodes that start with a randomly configured world. Whenever a task module is in the same state for 10 time steps or collides, the world is then reset and a new training episode begins. In reset, all cars $2 \ldots C$ are placed by randomly selecting a lane, position, and speed from uniform distributions across the full ranges of these variables. The LA is placed behind the pace car and this distance grows with the number of training episodes.

Results with the RL model are preliminary, an example of a learned policy for following pedal control is in Figure 2. Due to state space complexity, it only shows when the LA and the pace car are in the same lane. Since the state space is large, the dashed box indicates the area frequently visited and well-learned. The LA pedal control decreases speed when relative speed is less than zero and increases when greater. Additionally, when the pace car is close it slows down and speeds up when far away. Results with other modules are similar but more in depth analysis and simulation are in progress.

5 Eye Movements

5.1 Perceptual Arbitration

In the learning framework presented so far, the LA always has access to accurate state information. This is not the case in a real driving task, where a human driver with limited visual resources must fixate specific targets over time to resolve their true locations or speeds. Therefore, we follow the approach developed for a more static walking task [9] and incorporate the notion of state uncertainty into our model.

Instead of making a decision based on perfect state knowledge, the LA maintains an estimate of $\tilde{s}_t^i$ for each task module i in the driving simulation. This state estimate consists of a probability distribution over the entire state space; the most likely state of the world corresponds to the mode of this distribution, but the world might have changed since the LA last took an accurate state measurement (e.g., by foveating some object like the pace car).

When choosing an action, the LA multiplies its state estimate distribution with the learned Q tables, yielding an expected reward metric. For a given task module b, the loss ℓ^b incurred for not updating a module's state estimate is the difference in expected value between the reward that the LA might receive if it had perfect state information and the estimated value of the reward $\tilde{Q}^b$ given the action a^* that would be selected by the current (imperfect) state information:

$$\ell^b = E\left[max_a\left(Q^b(s^b,a) + \sum_{i\neq b}\tilde{Q}^i(s^i,a)\right)\right] - \sum_i \tilde{Q}^i(\tilde{s}^i,a^*)$$

This loss function can be used to guide the LA's perceptual resources during a simulation. Figure 3 (page 8) shows a diagram of the information flow in the computational model.

6 Future Work

The computational modeling work described in this paper forms part of a larger attempt to quantify and analyze human visual behavior in a realistic driving task. The model is in development and we are currently working to improve learning and add additional behaviors for dealing with pedestrians and oncoming cars. Additionally, because the model provides quantitative costs for various actions that the LA can take in the world, a major focus of our future work is to use the model to provide a plausible mechanism for explaining eye movements of human subjects navigating in a world involving multiple distinct tasks.

Our lab has a virtual reality driving simulator, consisting of driving platform with pedals and a steering wheel, a head tracking system, and a head mounted display (HMD). An eye tracker

Figure 4: The state space of the RL simulation can be projected easily into a 3D virtual reality driving simulation in the lab. Human subjects see this sort of view of the driving environment as they drive around in the virtual world.

is mounted on the HMD. Preliminary data has been collected from human subjects driving in a realistic urban environment with a pace car, other cars and pedestrians present; see Figure 4 for an example screenshot from the environment.

Preliminary analysis of human fixation data suggests that distributions of fixations are inconsistent with simple scheduling models (e.g. round robin), suggesting a scheduler like the one presented here may have more utility. While the current application of our methodology to driving is still in development, we believe that this general framework is a powerful and unique approach to understanding human vision and may also have broader application in the construction of computer vision systems.

References

[1] D.H. Ballard, M.M. Hayhoe, P.K. Pook, and R.P.N. Rao. Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences*, 20(04):723–742, 1997.

[2] J.A. Droll, M.M. Hayhoe, J. Triesch, and B.T. Sullivan. Task demands control acquisition and storage of visual information. *Journal of Experimental Psychology*, 31(6):1416–1438, 2005.

[3] M.M. Hayhoe and D.H. Ballard. Eye Movements in Natural Behavior. *Trends in Cognitive Sciences*, 9(4):188–193, 2005.

[4] M.M. Hayhoe, D.G. Bensinger, and D.H. Ballard. Task constraints in visual working memory. *Vision Research*, 38(1):125–137, 1998.

[5] L. Itti and C. Koch. Computational Modelling of Visual Attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.

[6] V. Navalpakkam, C. Koch, A. Rangel, and P. Perona. Optimal reward harvesting in complex perceptual environments. *Proceedings of the Nationalal Academy of Sciences*, 2010.

[7] C.A. Rothkopf and D.H. Ballard. Credit assignment in multiple goal embodied visuomotor behavior. *Frontiers in Psychology, Special Topic: Embodied and grounded cognition*, 2010.

[8] N. Sprague and D.H. Ballard. Multiple-Goal Reinforcement Learning with Modular SARSA(0). In *International Joint Conference on Artificial Intelligence*, volume 18, pages 1445–1447, 2003.

[9] N. Sprague, D.H. Ballard, and A. Robinson. Modeling embodied visual behaviors. *ACM Transactions on Applied Perception*, 4(2), 2007.

[10] R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*. The MIT press, 1998.

[11] S. Ullman. Visual Routines. *Cognition*, 18(1-3):97–159, 1984.

[12] C.J.C.H. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

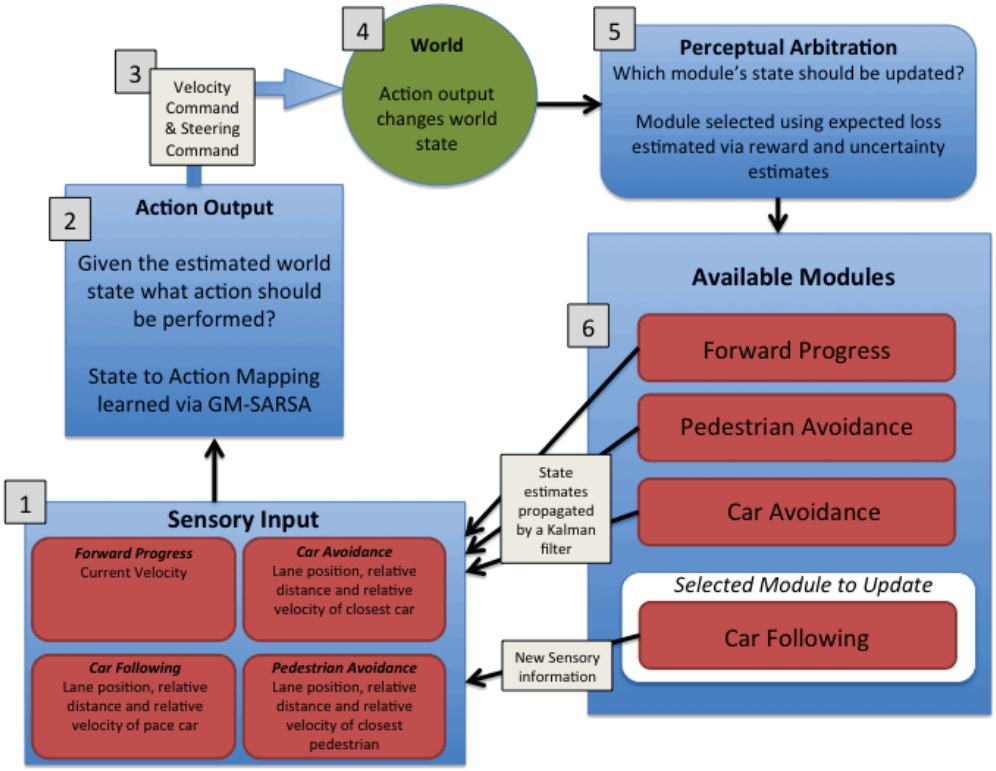


Figure 3: Flow diagram of the task-module scheduling architecture. Following the numeric labels, (1) the system initializes with a set of sensory readings about the world. Each task module has a representation of their state space that is (2) mapped to a learned action policy via the GM-SARSA algorithm. This mapping allows the driving agent to (3) output a steering and velocity command to drive the car. The actions taken (4) have some effect in the world that changes the world state. (5) Using information on potential rewards and state uncertainty, the perceptual arbitration algorithm chooses the module most in need of update to its world state estimate. (6) In this example the Car Following module is chosen to be updated and is able to gain access to new sensory information. The other modules cannot update and are forced to propagate estimates of their world state using a Kalman filter. This perception and action loops repeats itself each time step as the driving agent traverses through the environment.

Research student papers

A Dynamical Model of Feature-Based Attention with Strong Lateral Inhibition to Resolve Competition Among Candidate Feature Locations

David G. Harrison[1] and **Marc De Kamps**

Abstract. When subjects are instructed to attend to an object defined by a feature in the visual field, e.g. a shape element, a colour, or a direction of motion, one finds correlates of this feature in neural activity across all retinotopic locations in low level areas in visual cortex, even in locations far removed from the neurons representing the object of interest. Next to the well established phenomenon of spatial attention, there is another form of attention which is feature-based, but not location specific. While feature-based attention has been demonstrated convincingly, its role is not well established. In earlier work we have suggested that is role is to help to prepare eye movements (or other motor actions) to the object of interest. In order to be able to do so, the location of the defining feature must first be established. In a single-object scene this is trivial, but in a multi-object scene the feature must be located among distractors. While we showed there that the retinotopic position of the object can be retrieved by an interaction between top-down attention-driven activation and stimulus-driven activation in lower visual areas, we did not model the dynamics of the selection process explicitly. Here we show that if one does this, one must resolve a competition process between the location of interest and those of the distractors. The process is decided by feature-based attention, but in order to implement this competition lateral inhibitory connections are required, in line with the 'biased-competition model'. We will show that this process enables an unambiguous determination of the location of the target object and that the effect of the lateral inhibition is the effective shrinkage of receptive fields around an attended object which has been reported in several experiments.

1 INTRODUCTION

The amount of visual information entering the eye would overwhelm the brain's visual processing capability if the entire retinal input is processed equally at all locations [19]. In order to perceive visual objects in detail, the object of interest is brought into focus on the fovea. The mechanism by which we designate what is of interest is described as visual attention. The ability to visually attend to an object is so crucial to visual perception it has been argued that "to see is to attend" [25].

Three types of visual attention have been described: attention may be deployed to a location (spatial attention), an individual object (object-based attention), or to a collection of features (feature-based attention). Object-based attention may be described as an example of feature-based attention, as evidence [17] suggests that object representations are composed of distributed sub-object, feature building blocks bound together as the neural object representation as needed, or determined, by the visual stimulus.

[1] University of Leeds, UK, email: pab2dgh@leeds.ac.uk

1.1 SPATIAL ATTENTION

Motter [15] describes spatial attention as shrinking the receptive field around the attended to object. We achieve a similar effect with feature-based attention through lateral inhibition of cortical feature-binding circuits with receptive fields containing non-matching features to an endogenously initiated attentional template in the ventral stream. The lateral inhibition from non-matching populations subdues activity in neighbouring populations, effectively removing external stimuli in their receptive fields from further visual processing. Regions with few mismatches between the attentional template and stimulus driven activity project excitatory activity to a separate cortical area in the dorsal stream, where the saccade necessary to foveate the object may be generated.

In the contrast-gain model of spatial attention [18], the effective contrast of stimuli at an attended location are increased, providing a greater neural response of neurons with receptive fields at the attended location, compared to their response without attention.

In order for spatial attention to be applied, the location of interest must first be selected. If the location is unknown, a visual search needs to be performed to determine locations from known properties of the object before spatial attention can be engaged. Neural mechanisms to determine the location for spatial attention have been described in the literature, such as saliency [10] or priority maps [1]. Priority maps, like saliency maps, code for a location of interest from visual stimuli, but includes top-down influences in addition to bottom-up. The model we present here generates spatial saliency maps through the interaction of neural activity in top-down and bottom-up visual pathways. Influence of top-down flow is necessary to sustain output to the dorsal stream once a location for attention has been determined. See [24] for a review of visual search.

1.2 FEATURE-BASED ATTENTION

Feature-based attention describes the deployment of attention to known properties of the visual array. These properties are simple features such as colour, orientation and direction of motion [21]. Feature-based attention enhances the response of neurons which code for the attended to feature [13]. Feature-based attention is used to detect the presence of the features in the visual array, then uses the attentional map to resolve the location of those features. As the

location of these features is not know prior to the onset of feature-based attention, the feature templates must act across the visual field [22, 3, 13, 19, 4, 26].

The top-down feature template interacts with bottom-up activity in two ways. In the feature-similarity gain principle [12] the sensitivity of neurons which code for the presence of the attended feature is enhanced, whilst neurons which do not code for the attended feature are suppressed. In the biased competition model of feature-based attention [14, 7, 8], stimuli within a feature-selective neuron's receptive field compete for neural representation. When only a preferred stimulus is present within the receptive field and matching feature-based attention is applied, the response of the neuron is maximal. When an additional non-matching stimuli occurs in the receptive field the competition is biased in favour of those stimuli which match the coded for feature, causing a suppression of non-matching stimuli and increased response to the matching stimuli. When attention is applied to the non-preferred stimulus, the response of the neuron is suppressed, but still greater than the response of the neuron when the preferred stimulus is absent.

The existence of the feature-similarity gain principle and the biased competition model as mechanisms for feature-based attention are not mutually exclusive mechanisms. Rather, it has been argued that the feature-similarity gain principle predicts the biased competition model [3].

1.3 ATTENTIONAL CAPTURE

Visual attention may be deployed voluntarily, as in visual search or Posner cuing paradigms, but may also be initiated from external stimuli. Sudden changes in the visual input, such as an unexpected flash, will create a neural activity in populations coding for the location of the external change. This effect of breaking into conscious perception of a salient but irrelevant stimulus at a non-attended location is an example of 'implicit attentional capture' [20]. When attentional mechanisms are not engaged, attention may be briefly captured by this new stimulus, termed 'explicit attentional capture' [20]. Once the cause of stimulus has been ascertained, the stimulus may be ignored, with a concomitant reduction in the representative LIP activity, or actively attended, maintaining the location activity in LIP.

The neural correlates of visual attention have been much studied (see [1] for a recent review). We build upon a model of feature-based attention to resolve the binding problem which occurs from a distributed object representation [22], by adding a mechanism of lateral inhibition to resolve a collection of features to spatial locations.

2 THE MODEL

The model consists of two artificial neural networks, one modelling the bottom-up flow of stimulus activities and the other modelling the top-down flow. The bottom-up network is a feed-forward network of five layers corresponding to V1, V2, V4, posterior inferotemporal (PIT) and anterior inferotemporal (AIT) visual areas. A widening receptive field in higher layers allows AIT neurons to project across the entire V1 layer, allowing objects to be recognised in all locations. This network is trained using backpropagation to associate objects presented at the V1 layer, to individual neurons in AIT. V1 consists of 4 feature layers which detect lines of 45° orientations, and objects are presented by direct stimulation of neurons in the appropriate feature layer to simulate neural inputs from the lateral geniculate nucleus (LGN), which we do not model.

Once the forward network has been trained to recognise the objects, it is used to train the top-down network through Hebbian learning: each training pattern is evolved through the forward network and conditions reciprocal connection weights in the reverse network. This mechanism creates the attentional template.

With the forward and reverse artificial neural networks trained, they are then converted into a dynamical model as neural populations of Wilson-Cowan differential equations [23]. During this conversion the neurons in the forward and reverse networks are converted to a neural circuit separating out positive and negative activities in the ANN's to two spiking neuron populations, with one of the pair's populations implicitly coding for the negative activities. The architecture and conversion of the model into dynamical populations is detailed in [6].

Layers of neural circuits are created between layers V2, V4 and PIT of the converted forward and reverse networks to detect correlated neural activations in paired populations of the forward and reverse networks. Correlating activations in the forward and reverse networks is achieved through implementing the disinhibition mechanism described in [22]. We extended this disinhibition circuit with two inhibitory populations, to create a mechanism to inhibit the activity of neighbouring circuits when there is a mismatch of activities in the disinhibition circuit.

Figure 1 shows the populations of the disinhibition circuit in grey, and matching populations in the forward and reverse network as positive-forward (Pf) and positive-reverse (Pr), likewise Nf and Nr for the negative populations. Open triangles represent excitatory connections, and black triangles inhibitory. If we consider a stimulus-driven activation in Pf, the excitatory-positive (Ep) and gating positive (Gp) populations of the disinhibition circuit receive equal rates of excitatory spikes. The Ep population is inhibited by Gp, but there is a small delay in inhibition as the driving activity in Pf passes through Gp. This delay allows the attentional capture mechanism, as there is a brief output of Ep to LIP from increases in the spike rate of Pf. However, the inhibitory output of Gp is itself inhibited by the inhibitory-positive (Ip) population if there is matching positive activity in the reverse population, Pr.

The Ep population sends excitatory projections to the LIP layer, but also excites the inhibitory-lateral-inhibition population (ILI), which in turn inhibits the lateral-inhibition population (LI) from reducing the output to LIP of neighbouring circuits. Due to the mutually exclusive activities in the positive and negative populations, it is guaranteed that the disinhibition circuit can only receive strong excitatory spikes from one reverse and one forward population at a time[2].

In the situation of matching positive activity in the forward and reverse network just described (and analogously for matching negative activity), the circuit will output excitatory spikes to LIP from the Ep population, while the inhibitory spikes from LI to neighbouring circuits is prevented by inhibition from ILI. Now we consider a mismatch of activities in the forward and reverse networks. If, for example, Pf and Nr have high activity, Gp is not inhibited by Ip, shutting off the output to LIP and ILI from Ep in a few tens of milliseconds. With no inhibitory activity from ILI, the excitatory input to LI from Nr is unchecked and neighbouring populations of Ep and En neurons are inhibited.

It is worth stressing that the circuit only produces lateral inhibition on mismatches in the (implied) sign of the forward and reverse networks. We can consider the spike rate of populations in the for-

[2] Baseline activity, such as thermal noise, may cause occasional low level rates, but these are not significant.

ward network to be the weight of evidence from the visual array, and the spike rate of the reverse populations as the expectation of a positive or negative value in the receptive field of the circuit if that receptive field was to contain the searched for feature. Thus strong evidence and strong expectation output a strong excitatory activity to LIP, whereas some evidence and high expectation is inconclusive. Similarly, strong evidence which contradicts a strong expectation suggests the sought for feature is probably not in the receptive field, and this confidence is passed on to neighbouring circuits via lateral inhibition. If neighbouring circuits have strong evidence of a match, they can compete with the inhibition, and the most confident circuit (i.e. that with the highest initial activity) wins the competition.

The benefits of mismatches being determined as opposite signed activities allow a non-linear selection of matches and mismatches and allows for slight differences between instances of objects belonging to the same category. This non-linearity is desirable, so the use of lateral inhibition to inhibit neighbouring neurons on mismatches allows only neurons to survive the attentional template that are in regions with high correlation between stimulus and attentional template, without the mechanism running away and completely extinguishing matches throughout the visual pathway. In this way, spurious matches of individual populations are prevented from generating salient regions in LIP.

The model is implemented using MIIND [5], a computational neuroscience framework, for modelling the artificial neural networks and the simulation of the neural dynamics.

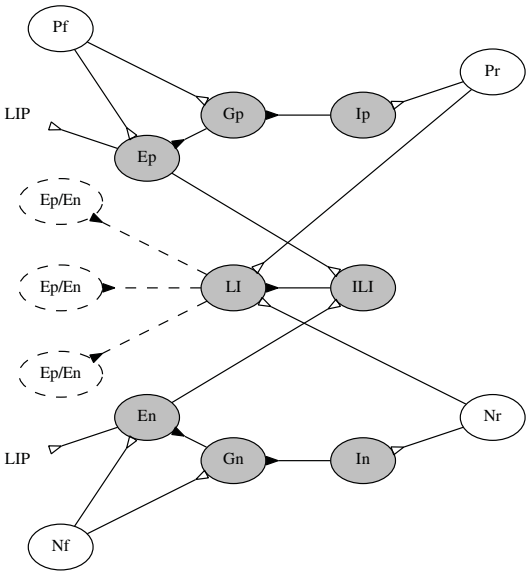


Figure 1. Disinhibition circuit with lateral inhibition. Grey nodes form the disinhibition circuit: Ep: excitatory population (positive), Gp: gating population (positive), Ip: Inhibitory population (positive). En, Gn, In denote negative populations. ILI: Inhibitory lateral inhibition population, LI: Lateral inhibition population inhibits positive/negative neighbouring populations of the forward network. 'f' suffix denotes populations from forward network, 'r' suffix denotes populations from reverse network. Black triangles denote inhibitory synapses, white shows excitatory synapse.

3 RESULTS

The forward network was created as a 16×16 grid of neurons, consisting of four feature detectors for lines at 45° orientations, and four AIT neurons to code for each of the square, diamond, horizontal cross and diagonal cross as seen in the top left layer of the following figures. Each shape was presented to V1 by direct stimulation of the appropriate feature neurons. Each shape was presented at every location in V1 which allowed the shape to be contained wholly, to avoid border effects. As the shapes are simple, training continued until the global network error, measured as the sum of differences between expected and actual AIT activity for every training exemplar, was below 10^{-5}.

Simulations were then run by converting the trained ANN's into dynamical networks as previously described. During the conversion the inputs patterns presented to V1 were varied, as was the attentional template, selected by activating an AIT population in the reverse network. The connectivity of the disinhibition and lateral inhibition circuit could also be modified at this stage to create simulations from identical inputs with and without lateral inhibition, by removing the projections from LI to neighbouring populations.

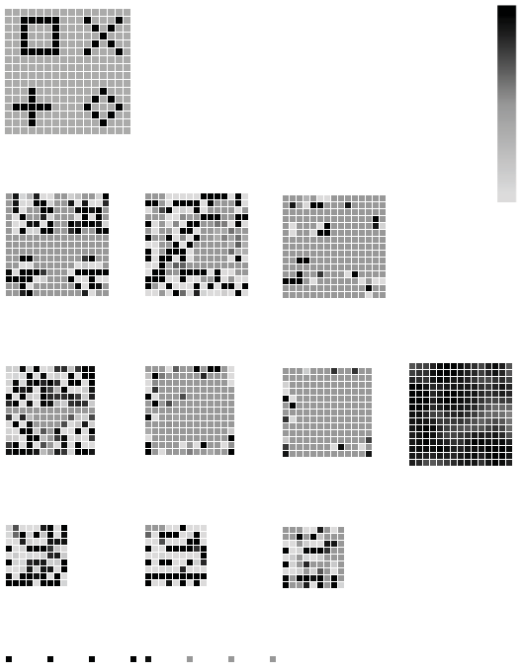


Figure 2. Final activity of network without lateral inhibition. The spread of activity across LIP shows no clear locus for spatial attention. From the left, the columns show the stimulus driven pathway of the ventral stream, the top-down stream, the disinhibition layers, then an LIP layer on the right. The rows of the first 3 columns show V1 (top), V2, V4, PIT and AIT (bottom).

Figure 2 shows an image of a simulation without lateral inhibition at the end of the two second simulation. Inputs to V1 and AIT were constant, so the image shows the steady state of the network. The bottom row of the image shows four AIT populations of the forward network, all showing high activity, demonstrating the presence of the four types of objects in the input array. Next to these are the four AIT populations of the reverse network, showing high activity in the left most population which codes for squares. The three layers above, are PIT, V4 and V2. The third column shows matching activity in the forward and reverse layers, and shows the activity of the Ep and En populations of the disinhibition circuits. Output of these populations is mutually exclusive, allowing activity of both populations to be visualised as a single element. Strong positive activity (Ep) is coloured

black, strong negative activity (En) the lightest grey, and zero activity the middle grey in the depicted colour bar.

The interesting image in figure 2 is the activity in the LIP layer. Without lateral inhibition, the high levels of activity in the forward network causes a high level of activity across the LIP layer, and this activity is poorly correlated with the location of activity in V1. This shows that busy visual scenes with stimuli across the visual array prevents a winning location being clearly resolved, and therefore the location to direct covert attention or generate a saccade to cannot be determined. This image shows the need for large scale inhibition to reduce the location noise in LIP.

Enabling lateral inhibition raises a number of interesting events in the neural dynamics. Visualisation software shows shifting patterns of activity in the disinhibition network and the LIP layer through the time course of the simulation. All simulations were run for 1.5 seconds and activation rates for each 5 millisecond period were recorded. Due to space constraints only three images are given in this paper, showing the more interesting features of the simulation.

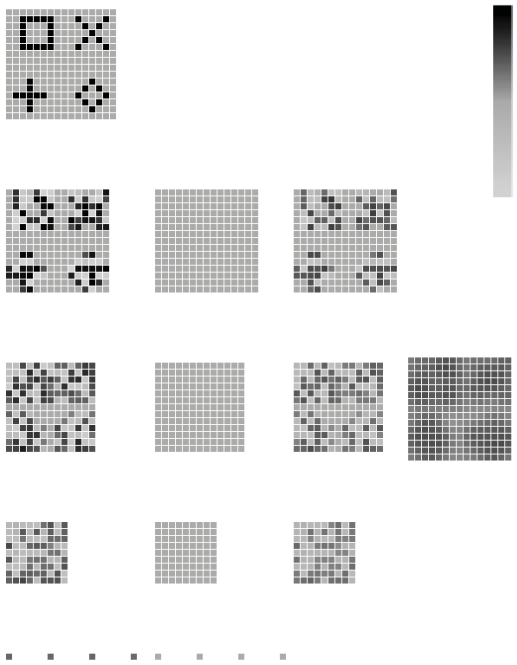


Figure 3. Network activity with lateral inhibition, trained on all locations, 30ms after stimulus onset. Prior to top-down activation, the stimulus activations cause location activity in LIP, demonstrating attentional capture. The layout is as described in figure 2.

Figure 3 shows a snapshot of the network 30 milliseconds after stimulus presentation. At this early stage the forward network causes activity in the Ep/En and Gp/Gn populations, but Ep/En is not yet inhibited by Gp/Gn so LIP receives excitatory spikes from all locations containing the presented objects. The image of the LIP layer shows high activity in the four quadrants corresponding to the activity in V1. This demonstrates the capture of attention by new visual stimuli. Spatial or feature-based attention would maintain activity in some (or all) of these active regions. However, in this simulation these activities are ignored.

After 500 milliseconds, the AIT population coding for squares becomes active and the attentional template is propagated to lower layers, gating the stimulus initiated activity to LIP at matching locations. The temporal nature of the dynamic simulation allows the time course of this disinhibition to be captured. While not shown in the images, location activity in LIP is first generated from activity in higher layers, and supplemented with activity from lower layers as the spread of activations from the attentional template descends the reverse network.

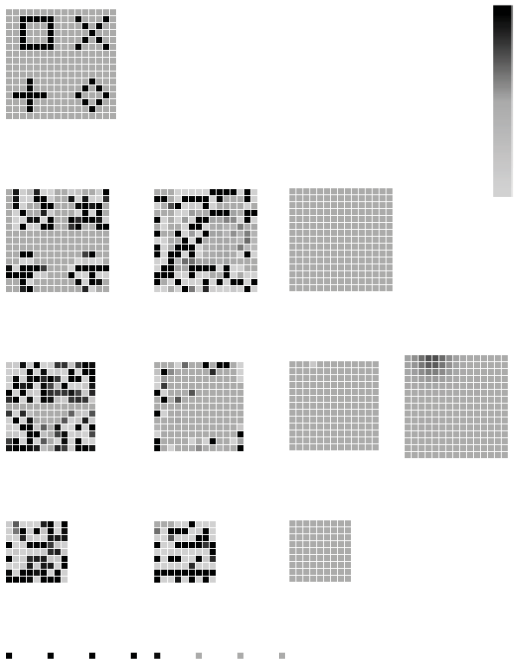


Figure 4. Network activity with lateral inhibition, trained on all locations, 350ms after simulation of the AIT population coding for the square. The layout is as described in figure 2.

Figure 4 shows the network activity 350 milliseconds after activation of the AIT population in the reverse network. Competition in the disinhibition layers has finished, leaving a single population of high spike rates being output to LIP. Neighbouring populations are also elevated above baseline, as their influence on LIP can be seen. However, this activity is barely visible on the image. Similarly, the areas of LIP activity seen in figure 3 are slightly elevated above baseline, as the inhibition of the Ef/En populations is not total.

Figure 5 shows the state of the simulation 1 second after activation of the cue population. This image demonstrates the location of the searched for feature persists with the application of attention. This contrasts with the LIP activity shown in figure 3, which is rapidly subdued.

In order to test the robustness of the model, simulations were generated with one to five objects in various positions of the visual array. In all cases the location of the resulting LIP activity covered at least part of the target object in V1. The model was able to resolve the location of objects even when distractor objects were overlaid on the target. Figure 6 shows an example of this, with the target square being partially obscured by an overlaid diamond. Comparison of figures 5 and 6 shows the activity in LIP to be slightly reduced. This reduction in activity occurs due to neurons with both the target and distractor in the receptive field experiencing more inhibition, and less excitatory stimulation as their receptive fields are effectively reduced in size.

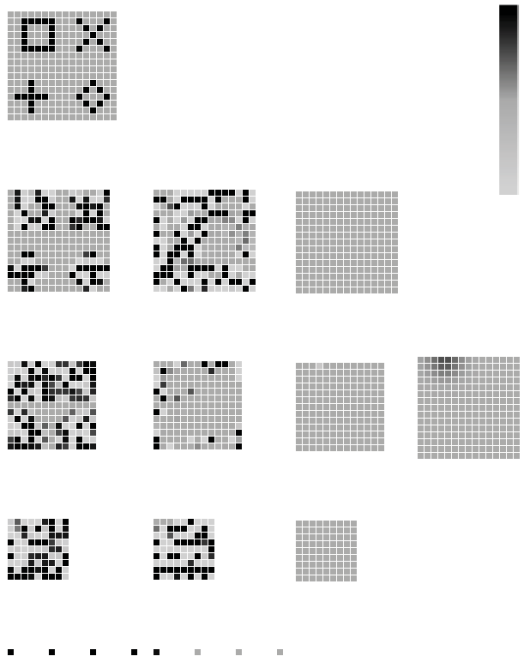


Figure 5. Network activity with lateral inhibition, trained on all locations, 500ms after stimulation of the square AIT population in the reverse network. The image shows maintained activity in LIP for locations of attended features. The layout is as described in figure 2.

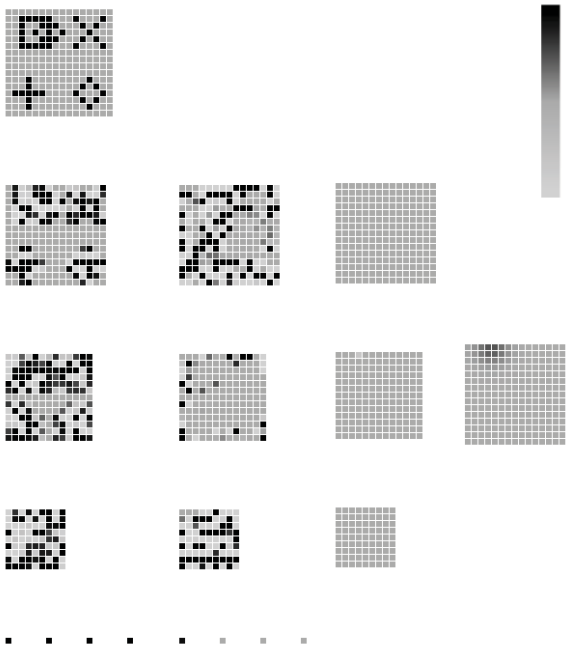


Figure 6. Network activity with lateral inhibition, trained on all locations, 500ms after stimulation of the square AIT population, with the target square (top left) partially obscured. The layout is as described in figure 2.

4 DISCUSSION

Motter [15] showed that the shrinking of the receptive field around an attended object is accompanied by an enhancement of stimulus activity (in the forward network) in neurons within that receptive field in the ventral pathway. Currently the model shrinks the receptive field to inside the boundary of the attended object. Increasing activity in the forward network will help borderline circuits neighbouring the receptive field to win their competitions with their inhibitory neighbours, and expand the receptive field. This could be achieved in the ventral stream by augmenting the disinhibition circuit to project excitatory connections from Ep/En to neighbouring Pf/Pn populations, or by reciprocal excitatory connections from the dorsal stream. As lateral inhibition affects the Ep/En populations of the disinhibition circuit, modulation of the Pf/Pn populations through such mechanisms is feasible with the current architecture. Future work will investigate this process.

By the same argument, the model provides the effect described by the contrast-gain model of spatial attention [18]. Although adaptation of the contrast response function of neurons is not modulated by the model, an inhibition of populations which do not contain the attended feature causes a gain in the signal to noise ratio. This is achieved by reducing the noise, rather than an active increase in the signal.

Despite the model not incorporating a method of directly elevating the neural representation of a feature through increased excitatory spike rates, both the feature-similarity gain and biased competition behaviours are exhibited. Feature-similarity is achieved by reducing the activity in non-matching locations, boosting the matching neurons' representations by a relative increase to background activity levels, rather than an absolute increase in spike rates. The same is true for biased competition: the representation of mismatched features is inhibited, so total incoming spike rates in the receptive field is reduced when a distractor object inhabits part of the receptive field compared to the receptive field containing only matching stimuli.

The attentional capture mechanism was not demonstrated in the simulations depicted here. However, we have run trials with activity in the reverse network prior to presentation of stimuli to V1. This models a Posner-like paradigm of a cue being presented and held in memory briefly before presentation of an array of targets and distractors. In simulations of this type, the location of the matching objects are not extinguished in LIP after the stimulus driven pulse of activity. While we do not provide direct evidence here, this can be seen from the architecture of the disinhibition circuit: stimuli matching the attentional template never receive the inhibition from Gp/Gn populations as they are already inhibited by activity from the top-down (reverse) network. Evidence for the modulation of activity in the the ventral stream through the action of attention without bottom-up visual stimuli has been demonstrated to act throughout the visual field [19], as exhibited by this model. Furthermore, this model predicts a reduction in LIP activity resulting from sudden stimuli onset when feature-based attention is already applied and the new stimulus does not contain the attended feature.

When the input array contains multiple objects to be attended, the objects' locations are found by appropriate activations in LIP. A form of inhibition of return could be implemented to resolve the multiple candidate locations to an ordered list of eye movements, based on criteria such as largest spike rate, or largest area of activity. The model as presented does not use the activity in LIP to generate actions or saccades, but the required neural information for such actions is available for use by these mechanisms.

5 CONCLUSION

We have shown that the use of an attentional template coupled with lateral inhibition of circuits neighbouring mismatched bottom-up and top-down populations can provide a resolution of targets from distractors in a mixed visual array. The lateral inhibition and disinhibition mechanism employed by the model allows for lateral inhibition for feature-based attention, while still supporting attentional capture: excitatory output to the dorsal stream (LIP) only occurs when the stimulus-driven activity is large enough (low pass filter), or when the attentional template matches.

We interpret our use of a top-down attentional template in combination with the bottom-up activity as a simple form of a priority map [2]. Attention effectively gates the output of the ventral stream [14] to visual areas in the dorsal stream, where planning of actions can be initiated. The inhibitory mechanism in effect causes the receptive field of neurons to shrink around the attended to object, as described by Motter [15, 16] and others [9, 1], while implementing a biased competition [7] between neurons coding for different features within a receptive field, and inhibiting activity of neighbouring neurons with distractor objects with their receptive field [11]. It should be possible to model other neural correlates associated with the deployment of attention, such as elevated bottom-up activity from attended stimuli, within the presented framework.

REFERENCES

[1] J. W. Bisley, 'The neural basis of visual attention', *The Journal of Physiology*, **589**(1), 49–57, (2011).

[2] J. W. Bisley and M. E. Goldberg, 'Neuronal activity in the lateral intraparietal area and spatial attention', *Science*, **299**(5603), 81 – 86, (2003).

[3] G. M. Boynton, 'Attention and visual perception', *Current Opinion in Neurobiology*, **15**(4), 465–469, (2005).

[4] G. M. Boynton, 'A framework for describing the effects of attention on visual responses', *Vision Research*, **49**(10), 1129–1143, (2009).

[5] M. de Kamps and V. Baier, 'Multiple Interacting Instantiations of Neuronal Dynamics (MIIND): aLibrary for Rapid Prototyping of Models in Cognitive Neuroscience', in *International Joint Conference on Neural Networks*, pp. 2829 – 2834, (2007).

[6] M. de Kamps and F. van der Velde, 'From artificial neural networks to spiking neuron populations and back again', *Neural Networks*, **14**(6-7), 941–953, (2001).

[7] R. Desimone and J. Duncan, 'Neural mechanisms of selective visual attention.', *Annual Review Neuroscience*, **18**, 193–222, (1995).

[8] R. Desimone and J. Duncan, 'Neural mechanisms of selective visual attention', *Annual Review Neuroscience*, **18**, 193 – 222, (1995).

[9] F. H Hamker and M. Zirnsak, 'V4 receptive field dynamics as predicted by a systems-level model of visual attention using feedback from the frontal eye field', *Neural Networks*, **19**(9), 1371 – 1382, (11 2006).

[10] L. Itti and C. Koch, 'A saliency-based search mechanism for overt and covert shifts of visual attention', *Vision Research*, **40**, 1489–1506, (2000).

[11] S. Kastner, M. Pinsk, P. De Weerd, R. Desimone, and L. Ungerleider, 'Increased activity in human visual cortex during directed attention in the absence of visual stimulation', *Neuron*, **22**, 751–761, (1999).

[12] J. C. Martinez-Trujillo and S. Treue, 'Feature-based attention increases the selectivity of population responses in primate visual cortex', *Current Biology*, **14**(9), 744 – 751, (2004).

[13] J. H. R. Maunsell and S. Treue, 'Feature-based attention in visual cortex', *Trends in Neurosciences*, **29**(6), 317 – 322, (2006).

[14] J. Moran and R. Desimone, 'Selective attention gates visual processing in the extrastriate cortex', *Science*, **229**(4715), 782 – 784, (1985).

[15] B. C. Motter, 'Focal attention produces spatially selective processing in visual cortical areas V1, V2, and V4 in the presence of competing stimuli', *Journal of Neurophysiology*, **70**(3), 909 – 119, (1993).

[16] B. C. Motter, 'Neural correlates of feature selective memory and pop-out in extrastriate area V4', *The Journal of Neuroscience*, **14**(4), 2190 – 2199, (1994).

[17] R. A. Rensink, 'Seeing, sensing and scrutinizing', *Vision Research*, **40**, 1469–1487, (2000).

[18] T. Reynolds, J.H. Pasternak and R. Desimone, 'Attention increases sensitivity of v4 neurons', *Neuron*, **26**(3), 703–714, (2000).

[19] J. T. Serences and G. M. Boynton, 'Feature-based attentional modulations in the absence of direct visual stimulation', *Neuron*, **55**(2), 301 – 312, (2007).

[20] D. J. Simons, 'Attentional capture and inattentional blindness', *Trends in Cognitive Sciences*, **4**(4), 147 – 155, (4 2000).

[21] A. M. Treisman and G. Gelade, 'A feature-integration theory of attention', *Cognitive Psychology*, **12**(1), 97–136, (1980).

[22] F. Van Der Velde and M. De Kamps, 'From knowing what to knowing where: Modeling object-based attention with feedback disinhibition of activation', *Journal of Cognitive Neuroscience*, **13**(4), 479–491, (2001).

[23] H. R. Wilson and J. D. Cowan, 'Excitatory and inhibitory interactions in localized populations of model neurons', *Biophysical Journal*, **12**, 1 – 24, (1972).

[24] J. M. Wolfe and J. Reynolds, 'Visual search', in *The Senses: A Comprehensive Reference*, eds., Allan I. Basbaum, Akimichi Kaneko, Gordon M. Shepherd, Gerald Westheimer, Thomas D. Albright, Richard H. Masland, Peter Dallos, Donata Oertel, Stuart Firestein, Gary K. Beauchamp, M. Catherine Bushnell, Jon H. Kaas, and Esther Gardner, 275–280, Academic Press, New York, (2008).

[25] S. Yantis, 'To see is to attend', *Science*, **299**(5603), 54 – 56, (2003).

[26] W.W. Zhang and S.J. Luck, 'Feature-based attention modulates feedforward visual processing', *Nature Neuroscience*, **12**(1), 24–25, (2009).

Coordination of multi-layered neural computation - a Neural Pipeline approach

Rebecca Naylor[1], Simon O'Keefe[1], Jim Austin[1] and Netta Cohen[2]
1. Computer Science Dept., University of York, Heslington, York
2. School of Computing, University of Leeds, Leeds
becky@cs.york.ac.uk

Abstract.

We present the Neural Pipeline - a novel multi-layered computational system that can control information flow without recourse to an external or internal clock. Information flow is demand-driven and the multi-layer coordination problem is solved in a distributed fashion. The architecture, structure and dynamics of the layers and their interconnections are inspired by biological neural networks, that operate robustly even in the absence of rhythmic control.

The system consists of multiple recurrently connected layers of leaky integrate-and-fire neurons with feed-forward excitation and feed-back inhibition to the previous layer. Given an appropriate balance of excitation and inhibition, the network will respond to inputs by sequentially propagating the input signal across the layers. The backward inhibition temporarily silences additional inputs while a layer is performing a computation. The behaviour in this operational regime is dubbed 'correct'. Under- or over-inhibition lead to two other behavioural regimes of this system. The parameters that most influence the type of behaviour are found to be the range of internal weights, the overall strength of external inhibition and the degree of connectivity.

We introduce supervised learning in a three-layer Neural Pipeline, by interpreting each of the three layers as a liquid state machine. The system is demonstrated to successfully recognise a set of six shapes on each of its layers independently. As would be expected in this set up, the specific patterns of connectivity within a layer are not important for learning to take place.

The potential for future work is discussed based on these preliminary results. Ideas include exploring the storage capacity of the network, separation of input patterns and the uses of such an architecture including the separation of a stream of inputs into its component parts.

1 Introduction

Visual recognition tasks using digital computers tend to be split into separate processing stages, with the timing of the flow between stages controlled by a global clock. It is important in computational systems that data appear in the correct order. When a result from one process is fed through as the input to another function, it is imperative that the first computational process completes successfully before the second function reads the value. If the second function starts before the first is complete it will use whatever is on its input, either an old value or worse a junk value.

The more general term for such a computational architecture is a pipeline. In a computer pipeline each processing task is split up into a number of subprocesses. Each of these subprocesses takes the same amount of time to execute and the computer hardware allows all of the subprocesses to operate at the same time [5]. This is advantageous when compared to a single process because of improved throughput when sequential inputs are provided. For example if the entire process takes 50ms to complete and it is split into 5 subprocesses that take 10ms each to run. Once the pipeline is 'full' (so that all of the processing stages are in use) one result can be provided every 10ms, rather than every 50ms if there was no pipeline. A global clock is used to keep all of the processing units in time.

In computer pipelines it is up to the designer to break down the overall function into same sized tasks. The processing stages of the visual stream, rather than being designed, have evolved. They may have evolved so each stage takes the same time to complete, but biological systems may use other methods for coordinating their information flow. In the same way it could be an advantage to have a computer system itself construct the timing of flow through the processing stages, because the design problem would be simplified by removing the global clock. Asynchronous systems (for an introduction see [13]) do not have one global clock, rather each process operates using its own clock. This can be of particular use when the system components operate in parallel or are distributed. As nervous systems are parallel and distributed they are of interest for inspiration when designing these types of system.

Nervous systems must also process information in a particular order. For example; when processing sound such as speech the syllables must be processed in the right order. It is not known whether the brain uses a clock in order to process this type of information. There are different rhythms present in the brain from alpha rhythms to delta rhythms. It is possible these rhythms may perform a role in timing but it is not generally understood how the brain regulates time. An overview of the different experiments and models to identify this is given in [6].

Biologically inspired computation borrows and simplifies biological components where they may be useful while remain-

ing free from the limitations imposed upon biological systems. There are many examples of biologically inspired computation, but the field of interest here is artificial neural networks. A good introduction to the field is provided in [3]. Using this paradigm we construct a biologically inspired system that can process data, through different computational stages, in a particular order, without the aid of a clock. The 'Neural Pipeline', introduced in this paper, combines the controlled flow of a computational pipeline with coordination using local activity in groups of neurons. The architecture is similar to the synfire chain controlled by inhibition presented in [12]. The two main differences between the two architectures are the presence of lateral connections within a layer in the Neural Pipeline, and the use of the layers themselves to provide inhibitory input rather than an external source.

A Neural Pipeline is structured so that it behaves as a computational pipeline. Each layer is a laterally connected group of neurons. Each layer passes information forward to the next layer. The neural architecture differs from a traditional computing pipeline in that the timing of information flow is not designed around a clock, rather flow is gated by the activity in each layer.

The Neural Pipeline has been used to perform character recognition and it learns as a multi-layered Liquid State Machine (LSM). Liquid State Machines were introduced by Maass et al in [8]. They are neural networks with two distinct parts; the 'liquid' is a set of randomly interconnected neurons and the 'readout map' is a set of output neurons that are connected to the liquid. Inputs are presented to the liquid layer and it becomes active. It is then possible for the output neurons to be trained to identify which input was presented. Only the connections between the readout layer and the liquid are trained. LSMs have two important properties: 'separation' relating to the liquid and 'approximation' relating to the readout. Separation is a measure of how distinct two different inputs appear to be in the liquid. Approximation is how easily the internal states can be transformed to a particular output on the readouts.

The remainder of the manuscript is organised in the following way. The Neural Pipeline architecture is described in section 2). The simulation environment used to test the architecture along with our specific implementation of the neurons and synaptic connections is introduced in section 3. The three fundamental behaviours of the pipeline are described and analysed (section 4). Section 5 presents our learning experiments and results. Directions for future work are presented with the Conclusions in section 6.

2 Architecture

The Neural Pipeline architecture is a layered neural network composed of leaky integrate and fire (LIF) neurons (for a neuron model description see [7] chapter 1). LIF neurons were chosen for several reasons: they represent a simple model of spiking neurons and therefore strike a balance between computational efficiency and biological realism. They are also used in LSMs in [8] allowing for easier comparison with these results.

Each layer has an equal number of excitatory and inhibitory neurons. The type of neuron dictates the sign of all its outputs, so an excitatory neuron has only positive outputs and an inhibitory neuron only negative ones. This format has been chosen to follow Dale's principle [1] as most neurons in vertebrates follow this principle.

The internal connections are the lateral connections within a layer. Each neuron in a layer has the same number of output connections. For each neuron, each of its output connections is attached to a neuron within the same layer. The target neuron for each connection is chosen at random from all of the neurons in the layer, with each neuron having the same probability of being chosen. Self connections and multiple connections are permitted. The level of connectivity is an important parameter in governing the overall behaviour of the pipeline as shown in section 4. The weights on these connections are all set to $+w$ for excitatory connections and $-w$ for inhibitory ones. All delays are set to have the same value.

A variation is to use weight values randomly chosen from a range *0 to* $+w$ for excitatory connections and *0 to* $-w$ for inhibitory connections. This variation matches the structure of the liquid layer that LSMs use in [8]. This allows the same liquid to be applied to different computational problems without the need for training. It is, however, possible to train the liquid in order to improve its performance for a specific task. Hebbian learning [10] and Particle Swarm Optimisation (PSO) [4] have been used to increase the separation of inputs within the liquid.

The pipeline is constructed by connecting the layers with external connections. There are excitatory feedforward connections between consecutive layers (X connections in figure 1). These connections propagate the signal through the neural pipeline. The connections on the input layer provide the system input to the 'input neurons' in the first layer. For example in the shape experiment in section 5, a 9 by 9 grid is provided as an input, so the first 81 neurons in the first layer serve as input neurons. Between all other layers each neuron in layer n provides input to a different randomly chosen subset of neurons from layer *n+1*.

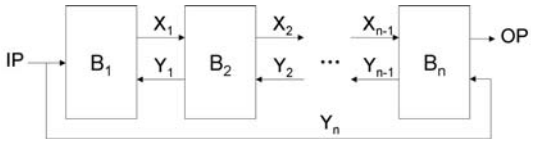


Figure 1. The external connections between layers

There are also feedback connections between layers, labelled Y in figure 1. These connections are inhibitory and are used to suppress activity in layer *n-1* when layer n becomes active. There is a time delay on these inhibitory connections to allow the layers some time to operate before they are shut off. This delay could be placed on either the feedforward or feedback connections. The inhibitory connections run from all neurons in layer n to all neurons in layer *n-1*.

This external connectivity appears to violate Dale's principle, because all of the neurons, including the excitatory ones have inhibitory connections. This could be corrected using inhibitory inter-neurons from the excitatory neurons. For ease of implementation however these neurons are not present in the simulations, rather the connection is inhibitory.

Connection Y_n (figure 1) is used to inhibit the final layer. This inhibitory connection has a suitable delay to cause inhibition after the last layer has had time to become active. When uninhibited the layer continues to spike rapidly after the first stimulus, until the end of the simulation. This causes a problem when sequential inputs are provided, because the last layer continuously inhibits the previous layer. This prevents any subsequent inputs reaching the output. This type of behaviour can also be seen when the inhibition is too low as described in section 4.

The inhibition is added from the input so that it is the arrival of a stimulus that triggers inhibition in the final layer. Clearly it is possible that the last layer could inhibit itself, or be inhibited by any of the other layers. The reason for choosing the input is to allow dynamic operation. If the delay on line Y_n (in figure 1) is longer than the time taken for the activity to reach B_n, then the stimulus will suppress its future self. If the delay is shorter than the time for activity to reach B_n then the last layer (B_n) is permitted to remain active until another input is presented to the pipeline. This allows for inputs that may take different times to compute.

3 Simulation Environment

Neuron Model		
Leaky IAF neuron		
Variable name	Value	Description
V_th	-69.931	Threshold voltage in mV
E_L	-70.0	The resting potential of the membrane in mV
C_m	250.0	Membrane capacitance in pF
tau_m	10.0	The time constant of the membrane in ms
t_ref	2.0	Length of the refractory period in ms
V_reset	-70.0	The reset voltage in mV
tau_syn	2.0	Synaptic alpha function rise time in ms

Table 1. Neuron parameters

Model Summary	
Internal excitation	0.5
Internal inhibition	-0.5
External excitation	5.0
External inhibition	-0.3
Internal delay	1.0 ms
External excitatory delay	1.0 ms
External inhibitory delay	5.0 ms
External excitatory connectivity	10%

Table 2. Connection parameters

The neural pipeline architecture has been simulated using the Neural Simulation Tool (NEST) [2]. There are other simulation environments that could have been chosen, but the objective here is not to review which would be best in this case. The choice is to use a simulation environment rather than not use one.

The parameters of the Neural Pipeline have not been tuned to enhance performance for each experiment. The parameters were set to achieve desired behaviour for a Neural Pipeline with 100 neurons per layer. They were unchanged for the experiment using different sizes of layer as described in section 5. This demonstrates some robustness in the system.

The neuron parameters for the *iaf_neuron* used in simulations are given in table 1 and the connection parameters in table 2. The equations for this model can be found in [9]. The synapses used are of the type *static_synapse*.

4 Behaviour

When the pipeline is presented with a stream of two or more sequential inputs it can exhibit one of three types of behaviour. Examples of these three types are illustrated in figure 2 when a stream of two inputs is presented 30ms apart. All neurons are initialised to be silent, having experienced no prior activity and there is no background noise.

The desired behaviour is shown in figure 2 b). In this case activity from both stimuli can be seen in each of the layers, and importantly the activity is suppressed again after activation. 'Over inhibited' behaviour is shown in figure 2 a). In this case the inhibition from layer 2 is too strong, because the second input does not produce any activity in the first layer and therefore any subsequent layers. This is the more preferable of the two undesired types of behaviour, because it is possible to resend the second input at a later time and for the pipeline to behave correctly. Therefore the definition of over inhibited behaviour is dependent on the required time between inputs, here 30ms. The least desirable type of behaviour is shown in figure 2 c), this is known as 'under inhibited'. When there is too little inhibition between layers *n-1* and *n*, layer *n* fails to suppress the activity in *n-1*. This means that layer *n-1* continues to fire and prevents any other activity from being provided as input. If this occurs then the pipeline needs to be flushed of activity before any further inputs can be provided.

The random connectivity in the network means that even when using the same parameters there is variability in the behaviour between runs. Although it is not possible to guarantee 'correct' behaviour there are certain parameters that can be used to improve the number of runs that operate in the region of correct behaviour. The three most important parameters used to influence the behaviour are the internal weights, external inhibition and the connectivity. Figure 3 shows the importance of the connectivity and internal weights. For all three internal weight values (graphs a, b and c) the number of correct runs decreases as the connectivity is increased. Each bar represents a total of 100 individual runs. With weights of +3 and -3 (Figure 3 graph c) only the lowest tested connectivity (10 connections per neuron) has 100% correct behaviour. The importance of internal weights is illustrated when this is contrasted with (Figure 3 graph a) where the internal weights are +1 and -1. Here there is 100% correct behaviour with up to 60 connections per neuron.

It is also important to note that the parameters choices are fairly robust, for example when the internal weight is well chosen, there is a much wider choice of connectivity where there

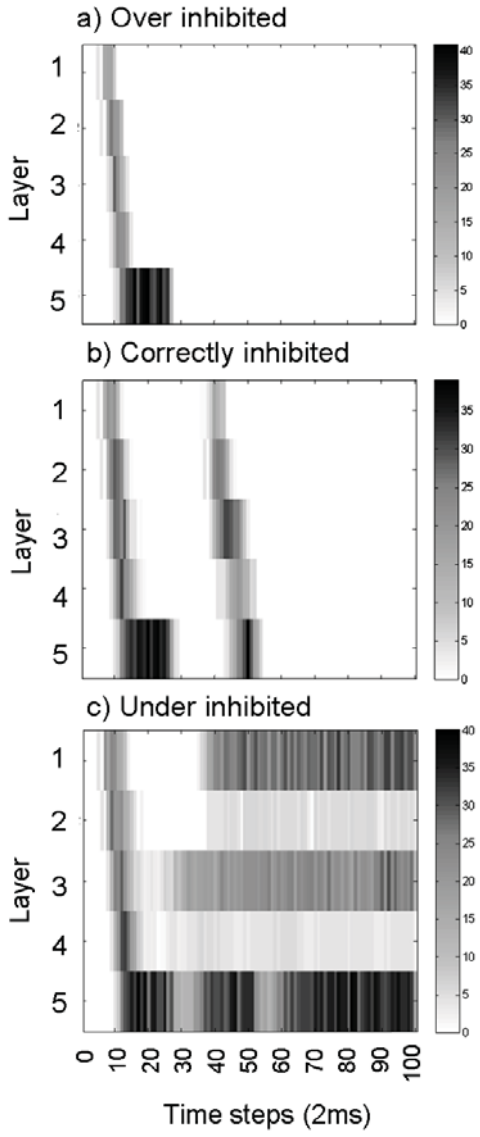


Figure 2. Examples of the three types of behaviour that can be exhibited by the neural pipeline

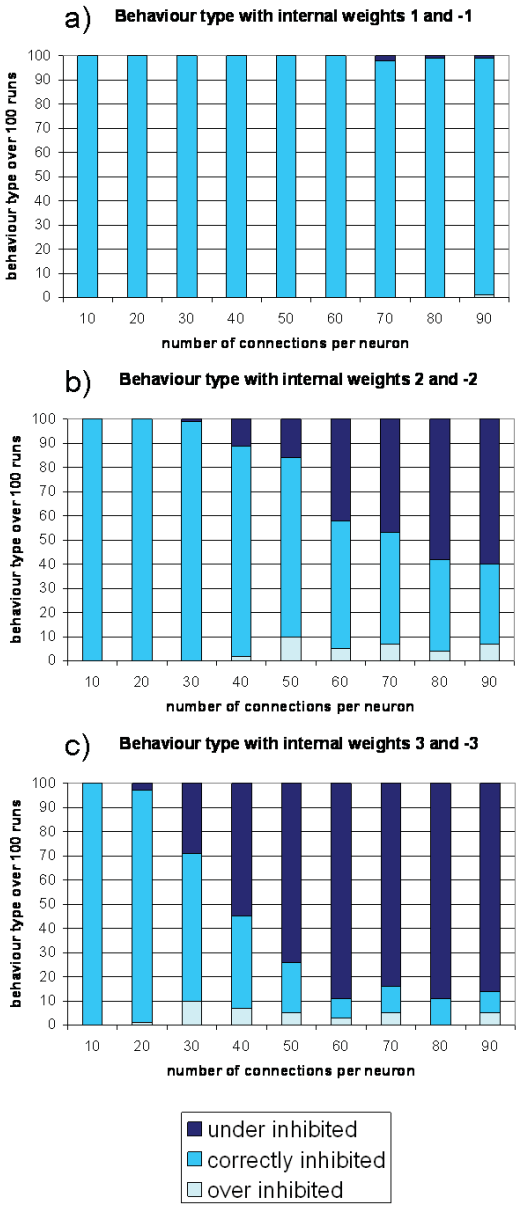


Figure 3. How the behaviour changes as the internal connectivity and internal weights are changed. Each bar represents 100 individual simulations.

is 100% correct behaviour. Also even with poor parameter choices (e.g. internal weights at +3 and -3 and 90 connections per neuron) some runs are correct, so if the time taken to optimise parameters was at a premium poorer parameter settings could be used with more runs of the experiment to achieve enough correct runs.

The importance of the external inhibition parameter was confirmed when beginning the shape learning experiment outlined in section 5. In this instance the number of inputs was increased from 5 (used to test the code initially) to 81. This large increase in inputs caused an increase in the activity in each layer and increased the number of 'over inhibited' runs. Reducing the inhibition from -0.6 to -0.3 allowed correct behaviour with 81 inputs.

5 Learning

The Neural Pipeline has been trained to recognise a set of six shapes on each of three layers as shown in figure 4. These six shapes are shown in figure 5 and were chosen so that there is some overlap, in active inputs, between them. The shapes are presented (at 1 spike per ms) for 10ms to layer 1 at the start of the simulation. Before this all neurons are silent. The 81 inputs are connected to 81 of the 100 neurons in the first layer. There are 100 neurons per layer with 6 readout neurons on each layer, one for each of the input shapes. The simulation is run for 100ms.

The readout neurons are fully connected to the layer and the initial weights on these connections are randomised be-

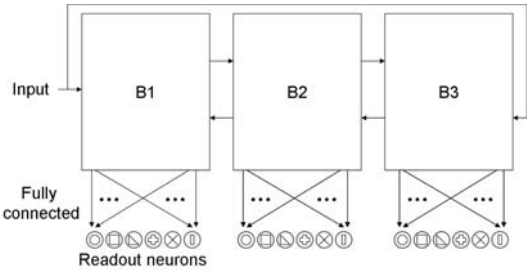


Figure 4. The neural pipeline architecture with readout neurons used for learning six shapes

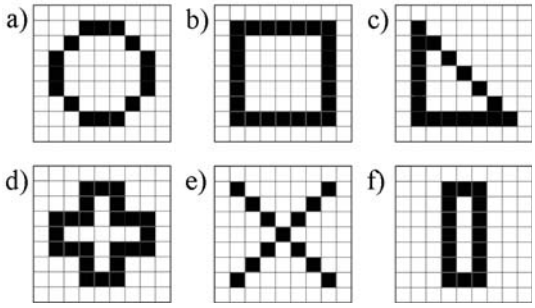


Figure 5. The shapes that the pipeline has been trained on a) circle, b) square, c) triangle, d) plus, e) cross and f) rectangle

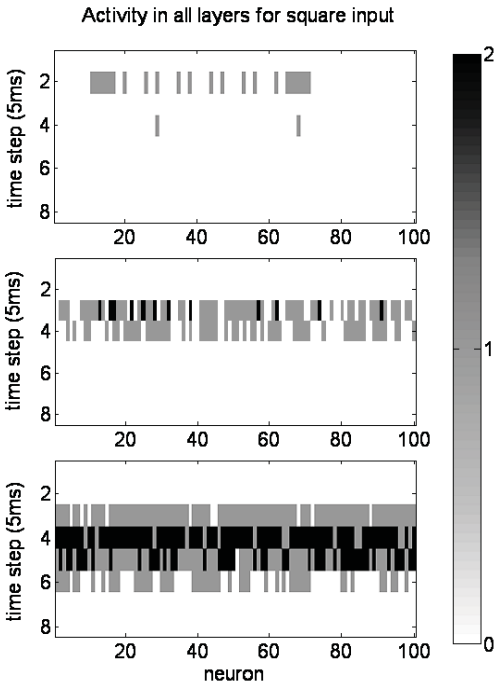


Figure 6. The activity in each layer produced when a square is applied to the input.

tween 0 and the 'internal excitation' value of 0.5. The weights are trained using the delta learning rule [11] to identify the input shapes at a particular time window. The times of all of the spikes that occur in the layer are recorded and divided into time windows of 5ms (as shown in figure 6). The first chronological set of unique windows all with non-zero values is used to train the network. The windows must be unique for each shape so that the system can recognise that pattern as belonging to a single shape. They must be non-zero because with no spikes it is not possible for the readouts to fire.

The readout neurons are trained to spike any number of times when their shape is the presented input, but to remain silent when the input presented is not their shape. So for each shape only one readout neuron will spike. Training is carried out until this is true for all of the readout neurons.

Figure 6 shows the response in each layer when the square shape as shown in figure 5b is presented at the input. Each unit of the graph represents the number of times that each neuron has fired in the time window. The square is presented only to layer 1 and can be seen appearing at time 2 as a regular pattern, when rearranged into a grid this is identical to the square input shown in 5 so is still identifiable as a square by eye. By the time the square has reached the second layer it is no longer identifiable by eye as a square, but the system can be trained to associate that pattern with 'square'.

The six shapes have been tested on one set of arbitrarily chosen internal connections. To determine that a set of inputs can be learnt with different sets of internal connections a smaller experiment was run 100 times. 50 neurons were used per layer, with 10 different inputs of size 40. Of the 40 input bits each pattern had 5 active bits and 35 inactive ones. The reduction in neurons per layer and the number of active input bits (when compared to the shape experiment) was chosen to decrease the simulation time as this experiment was to be repeated 100 times. Only the first layer readout was trained.

All 100 trials successfully learnt the series of inputs, showing that a specific connection structure is not necessary to allow patterns to be learnt. The average connectivity is important in determining behaviour, as outlined in section 4.

6 Conclusion and Future Work

It has been shown that the Neural Pipeline exhibits three different types of behaviour; under inhibited, correctly inhibited and over inhibited. Three of the system parameters, the internal weight values, connectivity and external inhibition, were identified as being important when trying to increase the level of correct behaviour.

The Neural Pipeline has been successfully trained to recognise a set of six shapes on each of three layers. The influence of specific connection choice on the ability to learn successfully was tested, to identify whether it is likely that the system will fail to learn the presented inputs. All 100 runs successfully learnt a set of 10 inputs. This illustrates that the system is not dependent on a particular set of connections.

The work presented here is an introduction to the architecture itself and an illustration of the potential that it has for future work. There are several avenues to explore from this work. These include the capacity of the layers or how many unique input shapes they can learn. This is related to the

separation property of LSM. Here a maximum of 10 different shapes have been learnt but it would be useful to consider what the maximum possible is depending on the parameters that are used. The separation of input patterns in itself is also a consideration for future work, particularly whether separation decreases in later layers. The influence of the variation (described in section 2) to use a range of values for internal weights upon separation can also be examined.

A hypothesis that could be tested is that; the Neural Pipeline with n neurons in each of l layers may be able to improve capacity when compared to a single layer LSM with nl neurons. Each layer could be used to identify a different subset of the patterns, this may take the form of a coarse filter on the earlier layers and finer ones in later layers. This would allow the system to remove the inputs that are 'definitely' a particular shape earlier and focus later layers on shapes that are harder to identify. The readout neurons from earlier layers could be used to influence the output from the later layers. Should this hypothesis be shown to be true it would provide an advantage that this multi-layered LSM has over a single layered one.

Multi-layered LSMs have been considered in the context of vision in [14] in their model of the mammalian visual system. The Neural Pipeline maps naturally onto a multi-layer process such as vision, with the different layers each representing a layer of the visual cortex. It would be interesting to investigate whether the Neural Pipeline architecture provides an explanation for the control of processing in structures such as the visual cortex.

Possible applications of a Neural Pipeline would be to separate a stream of inputs into its individual component parts or for each of the layers to identify different characteristics of the input. For example attributes such as the shape of an object, its location and colour.

The introduction to Neural Pipelines presented here serves as a stepping stone from which to show that the architecture can be applied to real computer vision problems.

7 Acknowledgements

We acknowledge the support of the White Rose Consortium through the support of a studentship as part of the Active Vision Network.

REFERENCES

[1] John Eccles. From electrical to chemical transmission in the central nervous system. *Notes and Records of the Royal Society of London*, 30:219–230, 1976.

[2] Marc-Oliver Gewaltig and Markus Diesmann. Nest (neural simulation tool). *Scholarpedia*, 2(4):1430, 2007.

[3] Simon Haykin. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, July 1998.

[4] Jiangshuai Huang, Yongji Wang, and Jian Huang. The separation property enhancement of liquid state machine by particle swarm optimization. In *Proceedings of the 6th International Symposium on Neural Networks: Advances in Neural Networks - Part III*, ISNN 2009, pages 67–76, Berlin, Heidelberg, 2009. Springer-Verlag.

[5] Kai Hwang and Faye A. Briggs. *Computer Architecture and Parallel Processing*. McGraw-Hill, Inc., New York, NY, USA, 1st edition, 1986.

[6] Uma R. Karmarkar and Dean V. Buonomano. Timing in the absence of clocks: Encoding time in neural network states. *Neuron*, 53(3):427 – 438, 2007.

[7] Wolfgang Maass and Christopher Bishop. *Pulsed Neural Networks*. MIT Press, 1999.

[8] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, 14(11):2531–2560, November 2002.

[9] Abigail Morrison, Sirko Straube, Hans Ekkehard Plesser, and Markus Diesmann. Exact subthreshold integration with continuous spike times in discrete-time neural network simulations. *Neural Comput.*, 19:47–79, January 2007.

[10] David Norton and Dan Ventura. Preparing more effective liquid state machines using hebbian learning. In *in: IJCNN2006, Int. Joint Conf. on Neural Networks, IEEEINNS*, 2006.

[11] Phil Picton. *Neural Networks*. Palgrave, 1994.

[12] Takashi Shinozaki, Hideyuki Câteau, Hidetoshi Urakubo, and Masato Okada. Controlling synfire chain by inhibitory synaptic input. *Journal of the Physical Society of Japan*, 76(4):044806, 2007.

[13] Georgios Theodoropoulos. *Strategies For The Modelling and Simulation of Asynchronous Computer Architectures*. PhD thesis, University of Manchester, 1995.

[14] G.M. Wojcik and W.A. Kaminski. Pattern separation in the model of mammalian visual system. In *Parallel Computing in Electrical Engineering, 2006. PAR ELEC 2006. International Symposium on*, pages 309 –312, 2006.

Visual search performance can be enhanced by instructions that alter eye movements

David J. Yates and **Tom Stafford**[1]

Abstract.

Recent evidence suggests that subjects perform better on some visual search tasks when they are instructed to search the display passively rather than actively [3, 4]. We have extended this finding in two ways: an additional neutral instructions condition established a baseline result and the subjects eyes were tracked during the experiment. Our results show that passive instructions lead to faster searching in a hard visual search task compared to either neutral or active instructions. This result indicates that we adopt a more active strategy by default and can be made to improve in some tasks by simply following instructions to search more passively. Our eye tracking analysis shows that the experimental instructions, which make no reference to the eyes, lead to systematic differences in the way the subjects search the display. Specifically, the subjects in the passive group take longer to initiate their first saccade, locate the target more quickly, and are faster to make a button press once the target is located. Much like the visual search results, the neutral instructions led to eye movements that were much more like the eye movements of the active group. This finding suggests that the instructions alter our search performance by changing the way we move our eyes. The potential implications for real-world visual search are discussed.

1 Introduction

Anecdotal evidence suggests that we are more efficient at visual search tasks when we are relaxed or adopt a more passive approach. In 2006, Smilek and colleagues [3] sought to formalise and demonstrate this effect by giving subjects different instructions prior to a visual search task. Half the subjects were given passive instructions, which told the subject "to be as receptive as possible, and let the unique item "pop" into your mind". The other half were given active instructions, which asked the subjects "to be as active as possible and "search" for the item" (the full instructions can be found in [3] p.548-549). The results of the experiment were clear: in line with the anecdotal evidence, subjects given the passive instructions performed significantly better than the subjects given active instructions.

Smilek concluded that the different instructions lead to the subjects adopting different "cognitive strategies", with the passive instructions giving fast automatic processes more influence over spatial attention and active instruction leading to a greater reliance on slow and unnecessary executive control processes. This explanation was further evidenced in a second experiment where they showed that subjects searched more efficiently on the hard visual search task when given a concurrent memory task.

The current study seeks to address two issues with the original Smilek experiment. The first issue is the lack of a control condition, which would establish a baseline result. Without this it is difficult to conclude whether passive instructions made the subjects better, active instructions made them worse, or both. If the passive instructions improve performance over a baseline result, this suggests that our performance on this visual search task is sup-optimal and can be improved. Such a finding could have implications for some real-world, two-dimensional visual search tasks, such as CCTV operators.

The second issue is that eye movements were not recorded during the experiment. If the instructions are leading to systematic changes in the way the subjects move their eyes, then this could be informative of the mechanism for the passive versus active advantage. For example, length of fixation has been shown to have a systematic effect on search. Hooge and Erkelens [2] found that subjects who made longer fixations selected better locations for their next fixation, and so located the target faster. This suggests that longer fixations allow subjects more time to accumulate evidence on the best location to move their eyes to next.

We predict, therefore, that the passive instructions are leading to subjects taking longer on each fixation, which will have the knock-on effect of them making better eye movements and locating the target faster compared to active subjects. Furthermore, if the instructions have their effect by moving the subjects along a passive–active continuum, we predict that the subjects given neutral instructions will fall somewhere between these two extremes.

During the running of this experiment, Watson and colleagues [4] published a replication of the Smilek experiment with eye tracking. Their hypothesis was similar to that proposed here, and their results will be reviewed alongside ours in the Discussion.

2 Methods

The experiment is a replication and extension of Experiment 1 in [3], where the reader is referred to for further details. The subjects were required to search for a circle that had a gap on either the left or right hand side amongst distractor circles that had a gap on both sides. The difficulty of the target discrimination was altered by having either a large gap in the target (easy condition) or a small gap (hard condition) (see Figure 1 for examples of display types).

The easy and hard conditions were presented in two blocks of 144 trials and the order of presentation was counterbalanced between subjects. A target was present in every trial accompanied by either one, three or five distractors, and subjects were asked to indicate what side the gap was on as quickly and accurately as they could. Subjects were given 12 practice trials at the start and were given a break between the two blocks.

[1] University of Sheffield, Sheffield. Email: t.stafford@sheffield.ac.uk

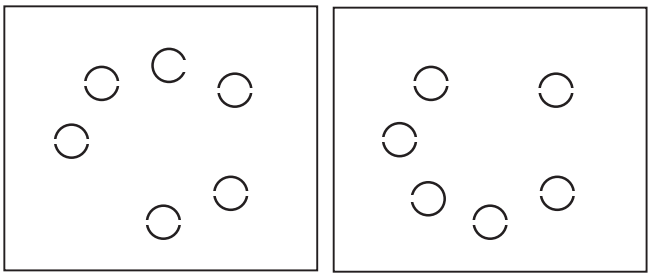


Figure 1. Examples of easy (left) and hard (right) search displays (taken from [3].

The current experiment differs from the Smilek experiment in two ways. First, a 'neutral' instructions condition was added, which simply stated:

> "The best strategy for this task, and the one that we want you to use in this study, is to respond as quickly and accurately as you can."

Second, we included eye tracking throughout the experiment. The centre of the left pupil of the subject was tracked at a sample rate of 60 Hz using a head mounted, infra-red video-based eye tracker (ISCAN RK-500). The addition of eye tracking meant that a nine-point calibration procedure was carried out before each block of trials, and head movements were restricted during the experiment using a chin rest.

42 subjects aged between 16 and 31 ($M = 20.0$ years, $SD = 3.0$ years) participated in the experiment (35 Female, 7 Male). 35 of the subjects took part in return for course-credit towards their undergraduate Psychology degrees. All subjects had normal or corrected-to-normal vision and were naïve as to the purpose of the experiment. The subjects all gave their informed consent to take part in the experiment and the procedures were in accordance with the ethical standards of the Department of Psychology Ethics Sub-Committee and British Psychological Society Guidelines.

The 42 subjects were split evenly between the instruction conditions and between the order of difficulty. However, seven subjects were removed from the eye movement analysis because their eye movement data was not reliable.

3 Results

3.1 Visual Search

The mean correct reaction times (RTs) for the easy and hard conditions can be seen in Figures 2 and 3 respectively. The mean correct RTs were analysed by a mixed analysis of variance (ANOVA) that assessed the within-subject factors of search difficulty (easy, hard) and set size (2, 4 and 6) and the between-subject factors of instruction (neutral, passive and active) and order (easy first, hard first). A multiple comparisons post-hoc test with Bonferroni correction was then performed to determine any significant differences between the three instructional groups. The full results of these tests can be seen in Table 1.

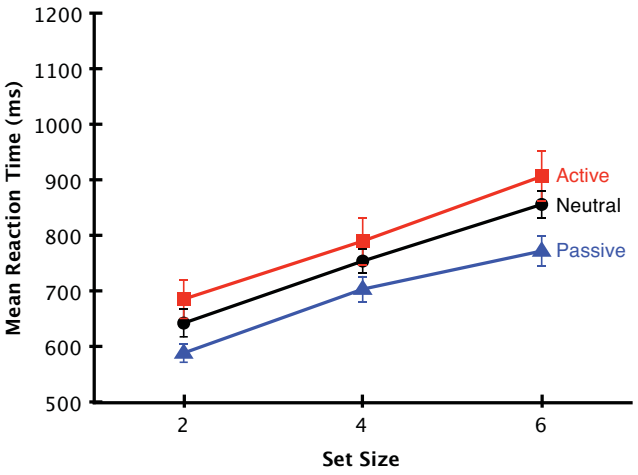


Figure 2. Points and error bars represent the mean correct RTs for the easy condition, ± SEM

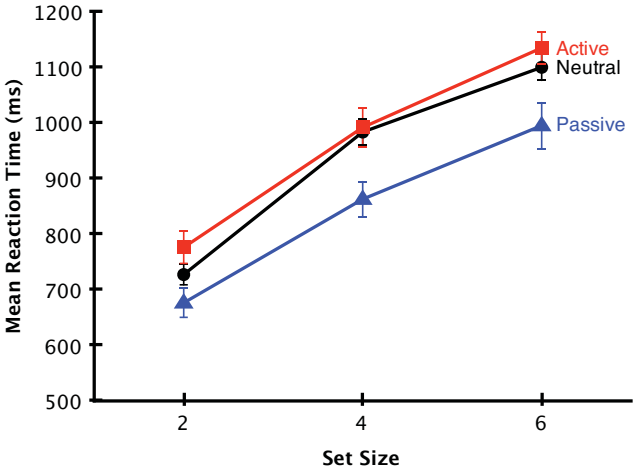


Figure 3. Points and error bars represent the mean correct RTs for the hard condition, ± SEM

The ANOVA and post-hoc tests revealed that there was a significant effect of instructions, with passive instructions leading to significantly faster performance than active instructions. However, unlike the results of [3], there was no significant interaction with set size, indicating that passive subjects are faster at searching but there is no significant change to the slope of their search function.

To look at the effect of the experimental instructions in more detail, we analysed the data separately in the easy and hard conditions. This revealed that instructions had a substantial effect on the speed of search when search was hard ($F(2,36) = 6.70, p = .003$), but the effect of instructions was only approaching significance in the easy condition ($F(2,36) = 3.24, p = .051$). In the hard condition, the passive instructions led to significantly faster performance than either the active ($p = .004$) or neutral ($p = .037$) instructions, but there was

no difference between the active and neutral conditions ($p = 1$). In the easy condition the passive subjects were significantly faster than the active subjects ($p = .048$), but there was little or no difference between neutral and passive ($p = .427$) or neutral and active ($p = .927$).

There was a significant effect of order of difficulty in the hard condition ($F(1,36) = 9.24, p = .004$) but not the easy condition ($F(1,36) = 0.38, p = .541$). Subjects performed significantly better on the hard task if they completed it after the easy task rather than before. However, this effect did not interact with set size or instructions.

The same ANOVA was also run on the number of errors made in each condition to ascertain whether the subjects in the passive condition were trading response speed for accuracy. This analysis showed that there was no significant difference between the number of errors made between instructional groups, ($F(2,36) = 1.00, p = .379$). In addition, combining errors and reaction times to form a single "search inefficiency" measure does not significantly alter the results so has not been presented here.

Table 1. Mean Correct RT: Mixed ANOVA Results.

Source	df	F	p
Difficulty	**1,36**	**205.28**	**.000**
Diff×Instructions	2,36	.51	.606
Diff×Order	**1,36**	**20.27**	**.000**
Diff×Instructions×Order	2,36	1.08	.349
Set Size	**2,72**	**648.18**	**.000**
Set Size×Instructions	4,72	1.96	.109
Set Size×Order	2,72	.16	.852
Set Size×Order×Instructions	4,72	.10	.982
Diff×Set Size	**2,72**	**61.03**	**.000**
Diff×Set Size×Order	4,72	1.30	.279
Diff×Set Size×Order	2,72	1.64	.200
Diff×Set Size×Order×Instructions	4,72	.40	.807
Instructions	**2,36**	**5.37**	**.009**
Order	1,36	1.27	.267
Instructions×Order	2,36	.41	.669
Multiple Comparisons			p
Neutral versus Passive			.109
Neutral versus Active			.920
Passive versus Active			**.008**

3.2 Eye Tracking

In order to make a direct comparison with the results of [4], each trial was split into three epochs, which are defined as follows. Epoch 1 is the time it takes the subject to initiate the first saccade, or the saccadic latency. Any saccadic latencies shorter than 100ms were considered anticipatory and removed. Epoch 2 is the time between the initiation of the first saccade and the eye fixating the target quadrant. The target quadrant was defined as a square region centred on the target and extending 0.5° beyond the edges of the item. Epoch 3 is the time between the eye fixating the target for the first time and the subject making their button press response.

Tables 2, 3 and 4 shows the results of the same mixed ANOVA and post-hoc tests used on the visual search data for epochs 1, 2 and 3 respectively. Figure 4 shows the collapsed data for time spent in each epoch by experimental instructions.

Epoch 1: The ANOVA revealed that passive subjects took significantly longer to initiate their first saccade ($M = 166.95$ ms, $SD = 23.33$ ms) compared to either neutral ($M = 148.95$ ms, $SD = 17.46$ ms) or active subjects ($M = 148.06$ ms, $SD = 14.80$ ms).

Table 2. Epoch 1: Mixed ANOVA Results.

Source	df	F	p
Difficulty	**1,29**	**8.38**	**.007**
Diff×Instructions	2,29	.72	.493
Diff×Order	1,29	.06	.810
Diff×Instructions×Order	2,29	.15	.860
Set Size	**2,58**	**47.80**	**.000**
Set Size×Instructions	4,58	.72	.583
Set Size×Order	2,58	.98	.381
Set Size×Order×Instructions	4,58	.44	.780
Diff×Set Size	2,58	.44	.647
Diff×Set Size×Order	4,58	.39	.814
Diff×Set Size×Order	2,58	1.15	.324
Diff×Set Size×Order×Instructions	4,58	1.52	.208
Instructions	**2,29**	**5.04**	**.013**
Order	1,29	1.32	.260
Instructions×Order	2,29	.18	.840
Multiple Comparisons			p
Neutral versus Passive			**.042**
Neutral versus Active			1.000
Passive versus Active			**.027**

Epoch 2: The passive subjects found the target significantly faster ($M = 260.65$ ms, $SD = 70.11$ ms) than the active subjects ($M = 315.69$ ms, $SD = 84.53$ ms). The subjects given neutral instructions fell between these two points ($M = 297.54$ ms, $SD = 51.42$ ms), but were more closely aligned with the active group.

Table 3. Epoch 2: Mixed ANOVA Results.

Source	df	F	p
Difficulty	**1,29**	**97.36**	**.000**
Diff×Instructions	2,29	.22	.800
Diff×Order	**1,29**	**9.89**	**.004**
Diff×Instructions×Order	2,29	.25	.783
Set Size	**2,58**	**264.82**	**.000**
Set Size×Instructions	4,58	1.06	.386
Set Size×Order	2,58	2.08	.134
Set Size×Order×Instructions	4,58	2.04	.100
Diff×Set Size	**2,58**	**10.57**	**.000**
Diff×Set Size×Order	4,58	.52	.720
Diff×Set Size×Order	2,58	.96	.390
Diff×Set Size×Order×Instructions	4,58	.17	.952
Instructions	**2,29**	**3.64**	**.039**
Order	1,29	1.15	.293
Instructions×Order	2,29	.90	.418
Multiple Comparisons			p
Neutral versus Passive			.279
Neutral versus Active			1.000
Passive versus Active			**.039**

Epoch 3: The passive subjects responded significantly faster once they found the target ($M = 340.80$ ms, $SD = 53.93$ ms) compared to the active subjects ($M = 416.96$ ms, $SD = 92.98$ ms). The subjects given neutral instructions again fell between these two points ($M = 402.06$ ms, $SD = 70.57$ ms) but were more closely aligned with the active group.

Table 4. Epoch 3: Mixed ANOVA Results.

Source	df	*F*	*p*
Difficulty	**1,29**	**81.05**	**.000**
Diff×Instructions	2,29	.21	.809
Diff×Order	1,29	3.56	.069
Diff×Instructions×Order	2,29	1.21	.312
Set Size	**2,58**	**5.60**	**.006**
Set Size×Instructions	4,58	.35	.840
Set Size×Order	2,58	1.62	.207
Set Size×Order×Instructions	**4,58**	**2.96**	**.027**
Diff×Set Size	2,58	1.53	.225
Diff×Set Size×Order	4,58	.69	.604
Diff×Set Size×Order	2,58	.76	.472
Diff×Set Size×Order×Instructions	4,58	.71	.587
Instructions	**2,29**	**4.49**	**.020**
Order	1,29	1.10	.303
Instructions×Order	2,29	.56	.578

Multiple Comparisons	*p*
Neutral versus Passive	.103
Neutral versus Active	1.000
Passive versus Active	**.025**

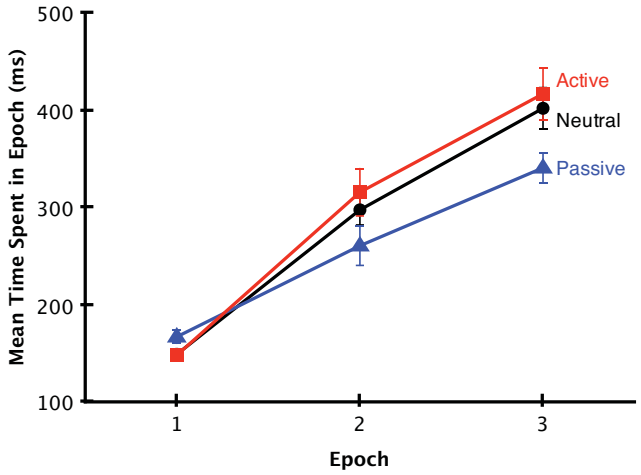


Figure 4. Points and error bars represent the collapsed group means for each epoch by experimental instructions, ± SEM.

4 Discussion

This study first of all confirms the findings of both Smilek [3] and Watson [4]: there was a clear speed advantage to being in the passive group compared to the active group. However, the neutral, baseline condition allow us to extend these findings, and conclude that 1) the passive instructions led to significant speed advantages compared to the baseline in the hard visual search task, and 2) the visual search performance of the neutral instructions group was much more closely aligned to the performance of the active group. This suggests that, on this task at least, we adopt a more active search strategy by default and can be made to improve by simply being instructed to search more passively beforehand.

During the running of this experiment, Watson and colleagues [4] published a replication of the Smilek experiment, with some minor alterations, and with the addition of eye tracking. They predicted that subjects would spend more time on individual fixations in the passive group, sampling fewer locations of the display but in more detail. Active subjects, they predicted, would spend more time "looking", sampling more locations but at a cost in fidelity. Their results were in line with their prediction: passive subjects gazed at the centre of the display for longer before making their first saccade. Furthermore, passive subjects were more likely than active subjects to fixate the target in three or fewer saccades and were faster to respond once they had found it.

Our eye tracking analysis intentionally copied the analysis methods of Watson and colleagues [4], splitting the data in to three epochs so that the two sets of results could be directly compared. Our results are clear and follow the same pattern: passive subjects took longer to initiate their first saccade, they located and fixated the target quicker and responded with a button press faster. However, we are able to extend Watson's findings in the same way that we were able to extend Smilek's findings: by showing that the neutral instructions led to eye movements that more closely resembled the eye movements of the active group, which mirrors the differences in performance between the instructional groups in the visual search results.

This finding suggests that the instructions are having their effect on search speeds by altering the way the subjects move their eyes. Therefore, we predict that the advantage of passive instructions would carry over to other visual search tasks that are primarily carried out by moving the eyes, but would be drowned out in search tasks that involve more gross movements of the head and/or body. Indeed, early evidence from Waton's lab indicates that this is the case: the effect of passive and active instructions does not carry over to real-world situations that involve head and body movements [1]. However, in instances where search takes place on a small, 2-dimensional display such as a monitor, we predict that there would still be an advantage to instructing searchers to do so more passively. This finding could have potential implications for jobs that require this kind of search, such as CCTV operators or baggage scanning at airports. We plan further work to establish the conditions that are necessary to elicit the passive versus active advantage.

REFERENCES

[1] A. A. Brennan, M. R. Watson, A. Kingstone, and J. T. Enns, 'From lab to life: Cognitive strategy fails to influence real-world search', *Journal of Vision*, **9**(8), 1203, (2009).

[2] I T C Hooge and C J Erkelens, 'Peripheral vision and oculomotor control during visual search', *Vision Research*, **39**(8), 1567–1575, (1999).

[3] D Smilek, J T Enns, J D Eastwood, and P M Merikle, 'Relax! Cognitive strategy influences visual search', *Visual Cognition*, **14**(4), 543–564, (2006).

[4] M R Watson, A A Brennan, A Kingstone, and J T Enns, 'Looking versus seeing: Strategies alter eye movements during visual search', *Psychonomic Bulletin and Review*, **17**(4), 543–549, (2010).

Proceedings of AISB `11: Architectures for Active Vision
Dimitar Kazakov and George Tsoulas (eds.)
ISBN 978-1-908187-00-0

Published by the Society for the Study of Artificial Intelligence and the Simulation of Behaviour
Printed by the University of York, York, UK