Integrated Cognition
AAAI Technical Report FS-13-03

# Towards a Neurocognitive Model of Visual Perception

**Arpan Chakraborty, Robert St. Amant**
{achakra, stamant}@ncsu.edu
North Carolina State University

## Abstract

Natural and artificial vision systems differ considerably in their underlying *hardware* and their method of information processing. Nevertheless, biological concepts are relevant, adaptable and useful in solving hard computer vision problems. This paper presents a biologically-inspired active vision framework that emulates early visual processing at the neuronal level to accomplish a range of visual tasks. Its emergent behavior is found to be qualitatively similar to humans in certain contexts, and performance is shown to be comparable to computer vision algorithms on a saliency detection task. A neurocognitive model of visual perception based on this framework is motivated.

## Introduction

The neurophysiology of visual perception paints a very fascinating yet mysterious picture of information processing in biological systems. The computer vision community has come to adapt this growing body of knowledge in different algorithms and architectures, at varying levels of abstraction, to address challenges in visual processing.

Some high-level generalizations have led to popular theories such as Marr's $2\frac{1}{2}$-D primal sketch (Marr 1982), scale-space theory (Lindeberg 1994), Treisman's feature-integration theory (Treisman and Gelade 1980) and Ullman's visual routines (Ullman 1984). But low-level simulation of biological vision systems remains inherently complex due to the sheer number of parallel processing units (neurons) involved, and their organic functioning (Armstrong and van Hemert 2009).

Researchers trying to implement biomimetic intelligence in practical systems are thus faced with a trade-off – simplify the nature of each individual computational unit, resulting in connectionist models such as artificial neural networks, or severely limit the number of units while maintaining a faithful replication of biological function, making them unusable except for applications with very low sensory complexity.

Modeling of higher-level visual functions and the interaction between corresponding areas of the brain make up a much more popular subject of study (Kosslyn et al. 1990; Vidyasagar 1999; Itti and Koch 2001; Wilson 2003). Cognitive architectures tend to use abstract models of early visual processing based on data derived from largely external observations (Byrne 2001; Lohse 1991). While such models are sufficient for their intended purpose of estimating human performance and timings in controlled settings, they do not reflect the underlying complexity involved in carrying out an arbitrary visual task.

The framework presented in this research models biological function at the neuronal level, and in the hierarchical organization of layers of neurons consistent with neurophysiological evidence. It provides mechanisms for orchestrating populations of neurons to perform low-level visual tasks, such as saliency detection and saccade generation, making it suitable for integration as an early-vision module within a larger cognitive system. The framework can be extended to provide higher-order visual functions such as object detection and tracking; one example of this, a scene reconstruction task, has been demonstrated.

## Foundations

**Neuronal architecture** Neurons in this framework communicate according to neurophysiological facts about membrane potential accumulation, firing and decay, making them fundamentally different from ones found in a traditional artificial neural network. Layers of neurons are organized hierarchically, the base layer being that of photoreceptors, with a foveal distribution. Other retinal neurons are modeled according to their functions, feeding up information to higher-level neurons with progressively larger receptive fields. Neurons also have inhibitory functions to prevent degeneration and implement attentive selection.

**Active vision and attention** For agents with active components, or which operate in dynamic environments, traditional passive approaches to vision can be inadequate (Aloimonos, Weiss, and Bandyopadhyay 1988). Visual attention is one of the mechanisms used by the framework to embody active vision. In humans and most other animals, attention is a necessity because of the foveal nature of our eyes. The illusion that we can see the world as a uniform high-acuity picture is maintained by frequent shifts of attention – saccades – once every few seconds (or less), and a yet unresolved

process of transsaccadic integration (Deubel, Schneider, and Bridgeman 2002).

**Frameless processing** An important characteristic of most computer vision systems, due to their inherent discrete nature, is that they treat dynamic visual input as a sequence of frames. This is also a result of the fact that most real-time or video processing algorithms are extensions of static image processing versions. As a result, computationally intensive procedures often suffer from synchronization problems when run at real-time. On the other hand, biological neurons function in a true parallel fashion, without any notion of globally synchronized *frames*. We take a cue from nature, and computer graphics (Watson and Luebke 2005), by employing an adaptive frameless sampling technique to update neurons pseudo-parallely using a priority measure defined by their own level of activity. This also lets the system performance degrade gracefully when less resources are available.

**Perceptual grounding** Cognitive models with symbolic reasoning have long suffered the problem of keeping symbols associated with appropriate percepts (Harnad 1990). A number of ways have been suggested to deal with this, including visual indexes (Pylyshyn 2001), a dual coding theory that integrates metric and symbolic information (Paivio 1990), and a theory of activity involving deictic representations (Agre and Chapman 1987). This framework provides an implicit perceptual grounding solution by exposing top-level object neurons that abstract out visual indexing and provide a consistent interface to higher-level architectures.

## Active vision framework

The design of our artificial vision framework begins with a simplified yet biologically plausible model of a neuron. Synaptic connections are modeled to the extent that they affect the functioning of neurons. A hierarchically connected structure is generated, starting with a base layer of retinal receptors, to model the early visual processing pathway. Finally, top-level neurons are designed to provide a functional interface to application-specific cognitive modules.

### Neuron model

Within our framework we define a *neuron* to be a computational unit with a small constant-sized storage, including a floating-point variable to store its *membrane potential*. Each neuron has one *axon*, and multiple *dendrites*. A dendrite can only connect with one axon and acts as an input line, whereas an axon can connect with multiple dendrites from different neurons and acts as the output line. Each *synapse* is modeled as a passive unit that serves as a connection between an axon (from the *presynaptic neuron*) and a dendrite (from the *postsynaptic neuron*). A synapse also has a single internal value quantifying the strength or weight of the connection.

We characterize the information flow across synapses as a simplified mechanism. When the membrane potential of a neuron crosses a certain threshold, it self-depolarizes rapidly and fires an *action potential*. This action potential travels through its axon to all synapses. For each activated synapse, the postsynaptic neuron's membrane potential is increased by an amount determined by the strength of the synapse. This transmitted potential is known as excitatory postsynaptic potential (EPSP).

A *photoreceptor* is a specialized neuron with no dendrites. It is excited by the intensity of light falling it, here obtained by sampling the value of image pixels in its receptive field. A trace of the membrane potentials of two modeled neurons connected by a single synapse is presented in Figure 1 to illustrate the information flow. The presynaptic neuron (top) is a photoreceptor that is exposed to a constant stimulus between $t = 2$ and 12 secs. Each action potential it generates (denoted by a spike in the trace) increases the membrane potential of the postsynaptic neuron by a small amount, which in turn fires when it crosses a threshold. Note that both neurons try to return back to an equilibrium value known as the *resting potential*.

Dashed horizontal lines in the plot mark, from top to bottom, (i) typical action potential peak, (ii) action potential threshold, (iii) resting potential, and (iv) typical action potential trough.

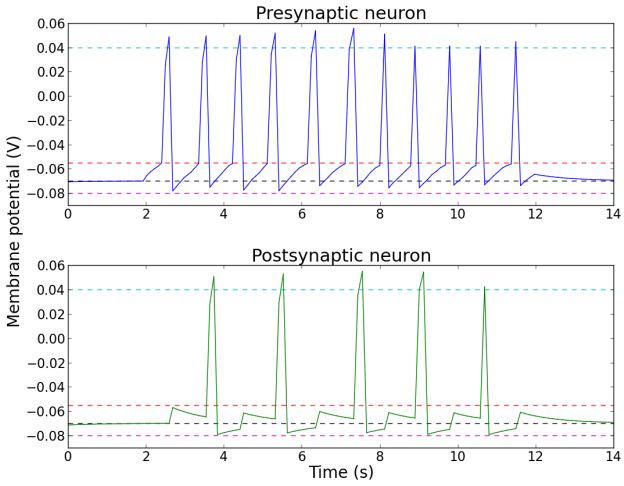



Figure 1: Neuron membrane potential traces

### Inhibition and synaptic gating

Inhibition is one important characteristic that enables neurons to perform a wide range of processing operations, including the control over attentional focus and modulating the spread of excitatory activations (Aron and others 2007). In addition to excitatory synapses mentioned above, neurons are known to form inhibitory synapses such that the firing of one neuron suppresses another – known as an inhibitory postsynaptic potential (IPSP). Inhibition has also been observed at synapses by a process called *synaptic gating* where a third "gatekeeper" neuron is connected to the synapse and blocks information flow when it is excited. Although the low-level process of inhibition is fairly well understood, neurophysiologists have not yet arrived at a con-

sensus on the type and mechanism of inhibition at different stages in visual processing.

The inhibition landscape is further complicated by specialized *interneurons* that form a network dedicated to channelizing information flow by inhibiting large collections of neurons. Taking note of these biological facts, our framework implements inhibition in the following way: At the individual neuron level, we use gatekeeper neurons at each synapse to control information flow; at the architectural level, we organize these gatekeeper neurons into a connected hierarchy to control groups of neurons by spreading inhibitory spikes. An example of synaptic gating is illustrated in Figure 2. The same setup is maintained as in Figure 1, with an additional gatekeeper neuron inhibiting the neurotransmission between $t = 5$ and $8$ secs.

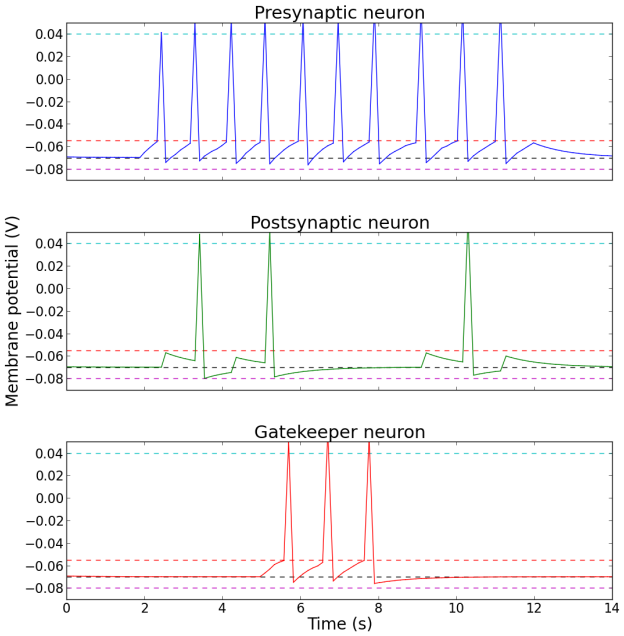



Figure 2: Synaptic gating

## Hierarchical organization and attentional control

As mentioned before, the interconnected structure of neurons in this framework is hierarchical in nature. Visual information is processed in a bottom-up fashion going from photoreceptors through progressively higher level neurons. Top-level neurons are specialized to fixate on salient regions, generate saccades and implement attentional control by sending back inhibitory signals through gatekeeper neurons.

Top-level neurons also interface with application-specific modules, providing both desired information (such as location and scale of the currently attended region) as well as methods to change attentional focus (e.g. to a specified spatial region, visual property etc.). The overall network is generated by providing foveal distribution and connectivity parameters. One such network is shown in Figure 3, although the networks used for our applications were more dense.

Each layer of the hierarchy is generated by sampling neuron positions from a bivariate normal distribution $(\mu, \sigma)$ where $\mu$ is the foveal center and $\sigma$ controls the degree of spread. Adjacent layers are connected by simulating dendritic growth from higher layers to lower layers with bounded length. Photoreceptors in the bottom layer are associated with input image pixel positions based on their location in the layer. Pixel intensity and color sensitivity drive their membrane potential.

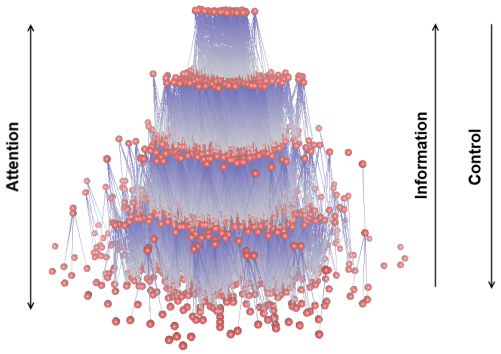



Figure 3: A sparse 5-layer hierarchical network depicting neuronal connectivity

## Computational challenges and solutions

Simulating a neuronal network capable of general visual processing is prohibitively expensive due to the massively parallel nature of computation involved. Fortunately, effects of neuronal activity can be integrated over short durations to closely approximate real-world behavior. The framework accumulates incoming potentials for each neuron as excitatory and inhibitory impulses are received. When a neuron is next updated, it first simulates time-based exponential decay of current potential and then factors in the potential values accumulated since the last update.

It then checks this stored potential value, in case it has crossed the action potential threshold. To model an action potential, a flag is set to perform additional self-depolarization on each update till a maximum is reached, after which potential falls abruptly and then slowly stabilizes. This crudely yet effectively estimates the process of ion exchange across a neuron's cell membrane. Note that updates occur in a bottom-up fashion, but the results are integrated over time. If the time period is sufficiently short, this simulation mimics the behavior of biological neurons up to a degree of abstraction and a margin of approximation that is appropriate for our purposes.

To further minimize the need to compute and update membrane potential values, we exploit the fact that neurons receiving less incoming potentials are likely to need infrequent updates, since our primary concern is to identify when a neuron crosses the action potential threshold. The framework maintains an update probability for each neuron that is correlated with its level of activity. A neuron is selected

to be updated by a random sampling based on this probability. A side-effect of this scheme is that visual input with a lot of temporal change causes an increased need for updates, sometimes overwhelming the framework.

One important difference remains between biological systems and our framework – the time scale of operations is an order of a magnitude longer, for example, while the time course of an action potential in biological neurons is typically under 5 ms (including the rising, falling and recovery phases), computationally it is only possible to achieve a duration of about 50–100 ms depending on the population of active neurons. For reference, we ran all our tests on an Intel® Core™ i5 3.4 GHz quad-core computer with 8GB RAM, and the generated networks contained 50,000–60,000 neurons. Nevertheless, the pattern of activation across our neuronal architecture and overall behavior closely resembles early visual processing in biological systems.

## Applications

### Studying change blindness

The biological plausibility of the framework makes it a useful tool for understanding human visual behavior and explaining certain peculiar phenomena. Change blindness (Simons and Levin 1997; Simons and Rensink 2005) is one such aspect of human vision. A change detection system built using the framework exhibited similar results as humans, although at a different time scale. More importantly, by continuously monitoring the activation levels of neurons, we got better insight into why this phenomenon occurs.

It has been observed that if two identical images with few deliberately introduced changes are presented immediately one after the other, then it is easy for humans to identify the change. However, if the visual array is blanked out (i.e. an empty image is presented) in between the two test images, then our ability to detect the change reduces drastically. Repeated presentation of such pairs of images with interleaved blanks is known as the "flicker" paradigm in change blindness research (Rensink, O'Regan, and Clark 1997). There are other interventions that also elicit change blindness behavior (e.g. mudsplats, and even real-world scenarios), but we have limited our current testing to flicker presentations only, specifically, with a single localized change (no image-wide color changes, etc.).

Since the framework already implements the central mechanisms for active vision, i.e. attention and gaze control, the additional work required for the change blindness system is minimal. A higher-level process simply monitors the current level of temporal variance in the focused region compared to other regions previously fixated, and also changes in gaze direction. If the current region maintains high temporal variance, and if the framework does not shift its gaze for a certain period of time, the agent identifies this as the location of change. This threshold is a parameter that can be varied to study its effect on the accuracy of the agent's response and the time it takes to detect changes.

Correctness of the agent's response is judged by checking if at least one-third of the changed area in the image is within the bounds highlighted by the agent. This means, if an object moves in the scene from one position to another, the agent only needs to identify one of the two locations. Time taken to detect the change is noted, subject to a timeout period. The agent is tested with both versions of a pair of change blindness images – with and without the intermediate flicker.

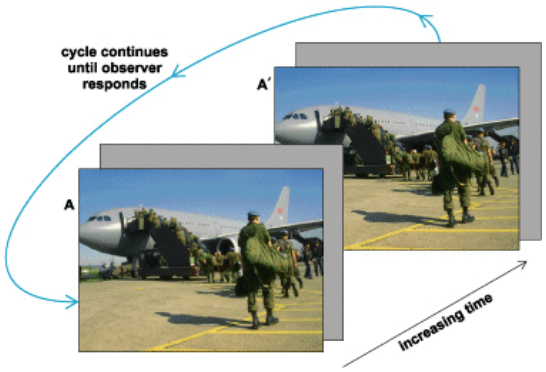
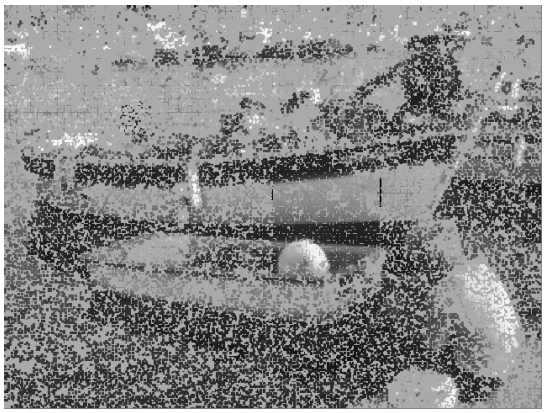



Figure 4: Illustration of the flicker paradigm in change blindness research

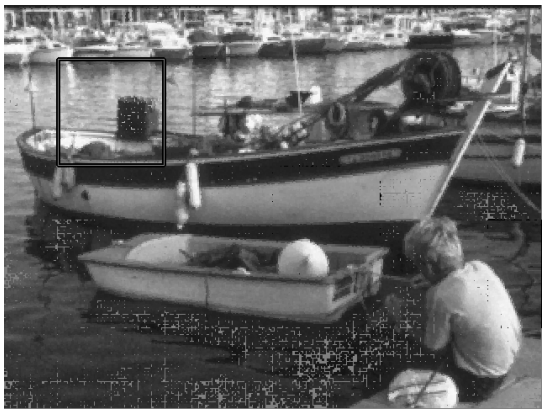
Figure 5 visualizes the internal representation that is built up over time for the test image pair "Harborside". It is clear that in the flicker condition, the perceived structure is not fully reflective of the actual scene, and does not contain enough information to allow the system to detect localized changes. Whereas, a definite structure seems to evolve in the no flicker case – without large temporal variations to overwhelm visual processing, the system is able to focus on areas that are spatially rich. Gradually, a clear enough representation emerges that enables identification of any localized temporal change. Figures 5b shows how the system draws a rectangle around the area it believes the change to be in.

Results indicate a clear difference between the flicker and no flicker conditions. The rate of identification is significantly high for the no flicker case, and the corresponding time to detect changes is low. In fact, the system was only able to detect a change correctly in 1 out of the 5 image pairs tested ("Airplane") in the flicker case (tests were only run for a duration of 60 secs. each, beyond which the system essentially never converged). This qualitatively agrees with human behavior – some people give up trying to find a change for certain difficult flicker image pairs, or use higher-level cognitive strategies to methodically scan the entire image (this has not been modeled in our system). For comparison, human participants have been observed to take on an average of 10.9 secs. to notice a change, requiring more than 50 secs. in some cases (Rensink, O'Regan, and Clark 1997).

Table 1 summarizes the time it took the system to detect changes in the different conditions. 10 runs for each image pair was conducted. The figures reported in the table are means. Standard deviation was under a second for each of these cases, hence has not been reported. Only the airplane case, in flicker condition, had a significant s.d. of 2.53 seconds. Variation in performance across different images is due to their respective complexity, significance of the change introduced, and its location in the image.



(a) Flicker



(b) No flicker

Figure 5: Visualized output of what the system perceives

## Scene structure description

The change blindness agent above is an example application that can be used for psychological and cognitive modeling research, by testing how the framework reacts to a given controlled interface/image. On the other hand, we would also like use the framework on physical agents that interact with the real world. Identifying the current scene structure description is one of the basic capabilities that is required for most such systems. Note that within this application, we are only interested in finding out the relative locations of different salient items in view, and not assigning semantic information with them. That is the focus of a separate pathway in visual processing, and when combined with scene structure description, results in scene understanding.

Table 1: Change blindness results summary

| Test image pair | Flicker | No flicker |
|---|---|---|
| Harborside | Failed | 9.37s |
| Corner | Failed | 11.59s |
| Airplane | 26.46s | 14.58s |
| Couple | Failed | Failed |
| Farm | Failed | 15.89s |

Our scene description agent simply identifies regions of interest in the given visual input and visits them sequentially. While making each saccade, it tries to perform trassaccadic integration. The scene description agent achieves this by sequentially linking percepts using their relative distances, and storing snapshots of each percept externally (i.e. outside the framework). Once the system believes it has studied all relevant parts of the scene, it reports back a description of the scene in terms of the relative distances recorded. This description is used to recreate a visual representation of the scene using stored snapshots. When used in a dynamic scene or with video as input, the resulting scene description is a temporally-integrated summary. Timestamps with each percept are maintained so that a progressive scene description can be generated in this case.

Figure 6 shows one of the complete live scenes that the system was tested on.

Figure 6: Test scene used for description task

Below is an example of the description presented by the system (this is only a sampling of one test run). The columns "pre_img" and "post_img" identify snapshots that were stored at that time.

| pre_img | post_img | x_disp | y_disp |
|---|---|---|---|
| 000 | 001 | 15.6471 | -86.0589 |
| 001 | 002 | 29.8825 | -67.2356 |
| 002 | 003 | 29.9599 | 29.9599 |
| 003 | 004 | 20.7304 | -6.91014 |
| 004 | 005 | 22.4968 | 44.9937 |

The reconstructed scene is shown in Figure 7. The blue dashed line segments in the image illustrate saccadic 'eye' movements of the system. Comparing with Figure 6, we can see that the reconstruction is not a very precise representation of the global scene. But the relative placements of salient objects in the scene were maintained (the painting and the lamp stand, for instance). This representation, combined with the ability to focus on specific things, gives the framework enough capability to be useful for active agents.

One possible extension to this agent will be the ability to look back at a previously focused region. The challenge here will be to design a higher-level process that can keep track of different perceptual as well as proprioceptive sensory information to correctly integrate visual percepts over longer

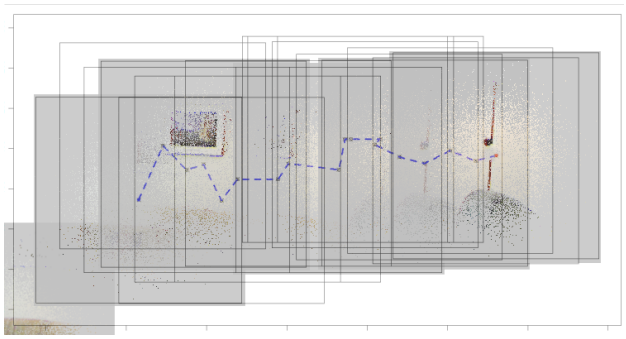


Figure 7: Reconstructed scene showing saccades

periods of time. This is beyond the scope of the framework, and is intended to be implemented in a higher-level agent.

### Saliency detection

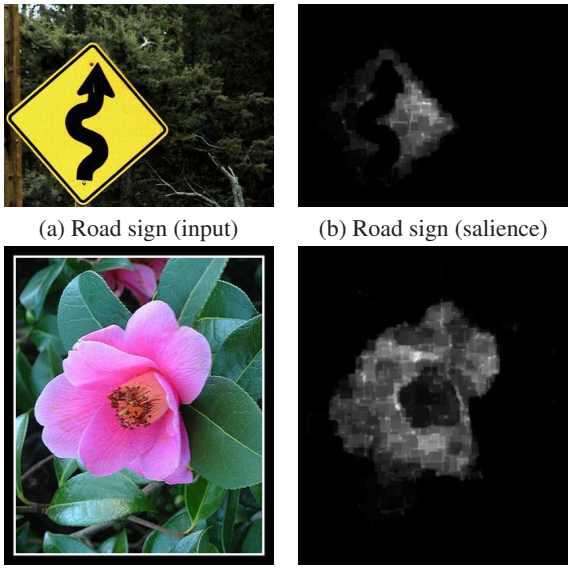
In both the above applications, the primary output of the framework is the location and scale of interesting regions, i.e. with localized spatial and/or temporal variance. One way of interpreting this is the concept of saliency. Itti, Koch et al. (Itti, Koch, and Niebur 1998) use saliency as a term to refer to the uniqueness of an area of visual input along one or more feature dimensions. Their computational method is inspired by evidence from neurobiology that indicates the presence of feature-specific neurons in different areas in the brain, but it only focuses on the analysis of static images.

We extended this work by implementing saliency detection in our framework using specifically modeled neurons that generate motor impulses for performing saccades and fixating on regions based on their saliency. Each image was presented to the application, and the saliency value, measured as the ratio of activity of neurons within the currently attended region to overall saliency in the image, was used to pick out the most interesting region. The performance of this application was evaluated on a published dataset (Liu et al. 2011) [1], and has been found to be comparable with some of the leading solutions. Figure 8 shows the salience map obtained by rendering neuron activity for two input images from the dataset.

Table 2 lists the results of running our saliency detection application on the above mentioned dataset that contains 5000 images labeled by nine users each to obtain mean ground truth salient object regions. The dataset is subdivided into ten input sets. Precision and recall figures compare the image areas covered by generated salience maps against areas marked by users. F-measure has been obtained with $\alpha = 0.5$, giving a balanced interpretation of precision and recall. BDE (Boundary Displacement Error) is a measure of how far (in total pixels) was the rectangular boundary found by our application from ground truth.

Mean precision and BDE results of our application are better than two popular saliency detection algorithms (Itti, Koch, and Niebur 1998; Ma and Zhang 2003) compared

[1]Dataset 'B' from this source: http://research.microsoft.com/en-us/um/people/jiansun/SalientObject/salient_object.htm



(a) Road sign (input) (b) Road sign (salience)

(c) Flower (input) (d) Flower (salience)

Figure 8: Salience maps generated from neuron activity

in (Liu et al. 2011) on the same dataset. When rendering a salience map from neuronal activity, we only consider values above a certain level, and this thresholding step is somewhat arbitrary. It is necessary in order to obtain results in a form that is comparable to the ground truth data, but unintuitive from a neuronal point of view. The resulting salience maps thus sometimes have holes or incomplete sections in them, leading to low recall rates, even though the detected external boundaries are relatively accurate (as shown by low BDE).

Table 2: Salience detection results summary

| Input set | Precision | Recall | F-measure | BDE |
|---|---|---|---|---|
| 0 | 0.760 | 0.544 | 0.641 | 33.351 |
| 1 | 0.736 | 0.547 | 0.630 | 33.052 |
| 2 | 0.763 | 0.546 | 0.647 | 32.827 |
| 3 | 0.775 | 0.523 | 0.644 | 33.377 |
| 4 | 0.784 | 0.527 | 0.653 | 31.986 |
| 5 | 0.794 | 0.536 | 0.661 | 31.322 |
| 6 | 0.796 | 0.542 | 0.666 | 31.085 |
| 7 | 0.755 | 0.545 | 0.646 | 30.313 |
| 8 | 0.682 | 0.625 | 0.634 | 28.572 |
| 9 | 0.742 | 0.594 | 0.656 | 28.944 |
| Mean | 0.759 | 0.553 | 0.648 | 31.515 |

## Conclusions and future work

As illustrated in this paper, the field of computer vision can learn a lot from biological vision systems. With increasing progress in our understanding of neurophysiological functions, we are slowly becoming capable of emulating natural processes with great precision and efficiency. Visual processing tasks that are hard computational problems today may turn out to be much easier to solve when considered within a fundamentally different neuronal architecture.

Having demonstrated some applications in saliency detection and related areas, this research has a long way to go in addressing other vision problems within a unifying framework, including object recognition, tracking, visual memory representation and transsaccadic integration.

To formalize the concepts embodied within this framework, we plan to formulate a neurocognitive theory of visual perception. It will enable us to reason about the behavior of the system as a whole, and provide provable results regarding its operational characteristics. This is the primary direction of future work, along with implementation of other applications. With some effort, the framework can also be used to extend a cognitive architecture such as ACT-R (Anderson et al. 2004; Anderson, Matessa, and Lebiere 1997) to provide neurologically-grounded estimates of execution times for visual tasks. For this purpose, the ACT-R P/M module is being investigated for possible modification. In this way, the framework can serve its dual purpose of reflecting on biological vision as well as providing a platform for active vision.

## References


Agre, P., and Chapman, D. 1987. *Pengi: An implementation of a theory of activity*.

Aloimonos, J.; Weiss, I.; and Bandyopadhyay, A. 1988. Active vision. *International Journal of Computer Vision* 1(4):333–356.

Anderson, J.; Bothell, D.; Byrne, M.; Douglass, S.; Lebiere, C.; and Qin, Y. 2004. An integrated theory of the mind. *Psychological review* 111(4):1036.

Anderson, J.; Matessa, M.; and Lebiere, C. 1997. Act-r: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction* 12(4):439–462.

Armstrong, J., and van Hemert, J. 2009. Towards a virtual fly brain. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 367(1896):2387–2397.

Aron, A. R., et al. 2007. The neural basis of inhibition in cognitive control. *Neuroscientist* 13(3):214–228.

Byrne, M. 2001. Act-r/pm and menu selection: Applying a cognitive architecture to hci. *International Journal of Human-Computer Studies* 55(1):41–84.

Deubel, H.; Schneider, W. X.; and Bridgeman, B. 2002. Transsaccadic memory of position and form. In *Progress in Brain Research*, 140–165.

Harnad, S. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena* 42(1-3):335–346.

Itti, L., and Koch, C. 2001. Computational modeling of visual attention. *Nature reviews. Neuroscience* 2(3):194.

Itti, L.; Koch, C.; and Niebur, E. 1998. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(11):1254–1259.

Kosslyn, S.; Flynn, R.; Amsterdam, J.; and Wang, G. 1990. Components of high-level vision: A cognitive neuroscience analysis and accounts of neurological syndromes. *Cognition* 34(3):203–277.

Lindeberg, T. 1994. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics* 21(1-2):225–270.

Liu, T.; Yuan, Z.; Sun, J.; Wang, J.; Zheng, N.; Tang, X.; and Shum, H. 2011. Learning to detect a salient object. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33(2):353–367.

Lohse, J. 1991. A cognitive model for the perception and understanding of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, 137–144. ACM.

Ma, Y.-F., and Zhang, H.-J. 2003. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia*, 374–381. ACM.

Marr, D. 1982. *Vision: A computational investigation into the human representation and processing of visual information*. WH Freeman and Co., San Francisco.

Paivio, A. 1990. *Mental representations: A dual coding approach*. Oxford University Press.

Pylyshyn, Z. W. 2001. Visual indexes, preconceptual objects, and situated vision. *Cognition* 80(1-2):127–58.

Rensink, R. a.; O'Regan, J. K.; and Clark, J. J. 1997. To See or not to See: The Need for Attention to Perceive Changes in Scenes. *Psychological Science* 8(5):368–373.

Simons, D., and Levin, D. 1997. Change blindness. *Trends in cognitive sciences* 1(7):261–267.

Simons, D. J., and Rensink, R. a. 2005. Change blindness: Past, present, and future. *Trends in cognitive sciences* 9(1):16–20.

Treisman, A., and Gelade, G. 1980. A feature-integration theory of attention. *Cognitive psychology* 12(1):97–136.

Ullman, S. 1984. Visual routines. *Cognition* 18(1):97–159.

Vidyasagar, T. 1999. A neuronal model of attentional spotlight: parietal guiding the temporal. *Brain Research Reviews* 30(1):66–76.

Watson, B., and Luebke, D. 2005. The ultimate display: where will all the pixels come from? *Computer* 38(8):54–61.

Wilson, H. 2003. Computational evidence for a rivalry hierarchy in vision. *Proceedings of the National Academy of Sciences* 100(24):14499–14503.