A Biologically Inspired Focus of Attention Model

A Thesis Submitted in Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Daniel I. Harris
dih0658@gmail.com
February 13, 2008

Supervised By

Dr. Roger Gaborski
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology

Approved By:

Dr. Roger Gaborski
Chair

Prof. Paul Tymann
Reader

Prof. Thomas Borrelli
Observer

Abstract

With high definition, high resolution, technology becoming ever more popular, the vast amount of input available to modern object recognition systems can become overwhelming. Given an image taken from a high resolution digital camera, a target object may be very small in comparison to the entire image. Additionally, any non-target objects in the input are considered unnecessary data, or clutter. While many modern object recognition systems have been created to be over 90% accurate in the recognition task, adding large amounts of clutter to an input quickly degrades both the speed and accuracy of many models.

To reduce both the size and amount of clutter in an input, a biologically inspired focus of attention model is developed. Utilizing biologically inspired feature extraction techniques, a feature based saliency model is built and used to simulate the psychological concept of a "mental spotlight". The simulated "mental spotlight" searches through each frame of a video, focusing on small sub-regions of the larger input which are likely to contain important objects that need to be processed in further detail. Each of these interesting sub-regions are then able to be used as input by a modern object recognition system instead of raw camera data, increasing both the speed and accuracy of the recognition model.

Contents

1 Introduction **3**

2 Cortical Structures **6**

2.1 Retina / Lateral Geniculate Nucleus 6

2.2 V1 8

2.3 V2 / V4 8

2.4 Inferior Temporal Cortex 9

3 Biologically Plausible Filters **12**

3.1 Center / Surround Filters 12

3.2 Orientation Sensitive Filters 13

3.3 Motion Sensitive Filters 14

4 Previous Research **18**

4.1 Koch and Ullman 18

4.2 Itti and Koch 18

4.3 VENUS 21

5 Model Definition **23**

5.1 Still Saliency 23

5.1.1 Intensity Contrast Feature Map 23

5.1.2 Orientation Contrast Feature Map 26

5.1.3 Color Difference Contrast Feature Map 26

5.2 Motion Saliency 28

5.2.1 Blink Filtering 29

5.2.2 Directional Filtering 31

5.3 Feature Map Integration 31

5.4 Focus of Attention 32

5.5 Attention Shifts . . . 36
5.5.1 Inhibition and Excitation of Attended Regions . . . 36
5.5.2 Salient Feature Feedback . . . 38
5.6 Model Output . . . 39

6 Simulations and Results **42**
6.1 Single Target . . . 42
6.1.1 Results . . . 43
6.2 Double Target . . . 44
6.2.1 Results . . . 44
6.3 Object Recognition Model Integration . . . 46
6.3.1 Biologically Inspired Object Categorization Model . . . 46
6.3.2 Region Extractions . . . 47
6.3.3 Categorization Results . . . 47
6.4 Attention Model Benchmarks . . . 49

7 Future Work **51**
7.1 Real Time Processing . . . 51
7.1.1 Biologically Inspired Filters . . . 51
7.1.2 Parallel Computing . . . 52
7.1.3 Programming Language Optimization . . . 52
7.2 Recognition Model Integration . . . 53
7.3 Pre-Cueing . . . 54
7.4 Object Tracking . . . 55

8 Summary and Conclusions **56**

1 Introduction

As computer vision continues to be a highly researched topic in computer science, object recognition remains one of the most studied aspects in the field. Many different object recognition systems have become extremely proficient at recognizing a wide array of objects that they have previously been trained to identify. However, one common problem many of these recognition systems share is that an input image that contains a target object to be identified needs to be relatively simple or uncluttered to reduce both error rates and processing time during the recognition task. Another important issue which affects some recognition models is their inability to ascertain the location of an object once recognized. Certain models simply determine that an object is present in the input, but cannot decipher the specific location of that object within the input. The inability of many object recognition systems to function both proficiently and efficiently on complex and cluttered scenes is a primary concern that must be addressed before such systems can be put to use in a broad range of tasks of everyday tasks.

One of the primary goals of computer vision has always been to build computational models which perform vision tasks as well or better than their human counterparts. To achieve this goal, researchers have began to look at the underlying physiology and psychology of the human recognition task to assist them in building computational models. Computational models which utilize biological and psychological information can been categorized into two groups: biologically inspired and biologically plausible models. Biologically inspired models utilize the ideas found in biological and psychological research as a template for the model, but the entirety of the model is not necessarily derived from these ideas. Biologically plausible models, however, are based entirely on biological and psychological research and can be thought of as a working copy of the way researchers believe a task is completed by humans. Striving to build a computational model based on the human vision task allows the building a more efficient vision model since it based on a highly accurate biological system. At the same time, modeling the human visual task can yield previously unknown information or new questions about the human visual task that had not been previously explored.

One of the earlier theories describing visual perception, first proposed by Marilyn and Peter Shaw, that has gained wide acceptance is the two-stage theory of perception [17]. The theory describes two different stages of perception, pre-attentive and attentive, which are used in conjunction to improve the vision task. The pre-attentive stage occurs early in the visual process and is used primarily to help focus cognitive resources onto a specific spatial location whereas the attentive stage occurs late in the visual process, using a significant amount of cognitive resources to process a specific spatial location in very high detail. A significant amount of research and development has been spent building models which perform the tasks of the attentive, or object recognition, stage of percep-

tion. Object recognition systems frequently require large amounts of time and processing power to accurately recognize an object similar to how the attentive stage of perception which utilizes large amounts of cognitive resources to process a spatial location in high detail. A significantly smaller amount research and development has been devoted to building computational models of the pre-attentive, cognitive focusing, stage of perception.

Psychologists, rather than biologists, have invested the most time into researching topics related to the pre-attentive stage of perception. Focusing cognitive resources onto a specific spatial location has become known to psychologists as visual attention which is important because the human brain is not capable of processing the enormous amount of data obtained from the estimated 125 million photoreceptors in each eye. Anne Treisman, a prominent psychologist from Princeton University, postulates that visual attention functions similarly to a spotlight [15] used in a stage show. A spotlight at a stage show serves to focus the attention of the audience to a single small point on the stage by illuminating that specific area while all other activity happening outside of the spotlight goes unnoticed by the audience. Treisman has put forward that visual attention functions in a very similar way: given a large amount of input, the brain is capable of directing a "mental spotlight" to a small region of the input for the higher brain to focus on, allowing the remaining, un-illuminated, input to go unnoticed.

The mental spotlight metaphor was the natural continuation of the "Feature Integration Theory" [16] developed earlier by Treisman and her colleagues. The "Feature Integration Theory" was developed by observing how humans completed various visual search tasks and postulates that in the pre-attentive stage of perception multiple primitive features, such as color, orientation and intensity, are quickly computed in parallel by the brain. The theory continues, describing that the attentive stage of perception is responsible for integrating these different primitive features from a small location in the input into more meaningful information. The "Feature Integration Theory" has become the foundation for the idea that the brain computes a saliency, or importance, map. The purpose of a saliency map is to show the importance of each individual area of a visual input. A saliency map can be used to assist in focusing a mental spotlight onto a much smaller region of the input for higher level processing.

Another concept Treisman has contributed to the fields of visual search and attention is the idea of perceptual grouping. While developing the feature integration theory, Treisman and Gelade proposed that attention might be directed towards groups of similar objects instead of individual objects [15]. The idea of perceptual grouping was confirmed through experimentation which showed that subjects would scan serially between groups, for instance searching through all the red objects before the green objects, instead of individual items. Perceptual grouping has been found to follow many of the Gestalt grouping principles, specifically similarity (color, orientation, etc.) and proximity. The effects of perceptual grouping on the visual search task has lead to the idea that the "mental

spotlight" may follow similar grouping principles when shifting between different spatial regions of the visual field.

As previously described, many different models have already been developed which are responsible for the second, object recognition, stage of the visual task. To complement these existing systems, a model inspired both by the psychology and biology of the pre-attentive stage of perception is developed which implements Treisman's "Feature Integration Theory" as well as her concept of a "mental spotlight" and shifting mechanism. This system can be used to determine and extract small areas of interest from video which may require further processing by one of the many existing object recognition systems. Used in conjunction with an accurate recognition system, the integration of the pre-attentive model developed in this research with the recognition model will theoretically yield more accurate results than the stand-alone recognition model when faced with a complex input.

2 Cortical Structures

When processing visual information, visual input is passed through a hierarchy of cortical structures which will be described in this section. Current research describes that there are two distinct streams of cortical structures (figure 1), dorsal and ventral, which visual information flows through during the visual task [9]. The dorsal stream, also known as a the "where" stream, retrieves data from the retina and lateral geniculate nucleus (LGN) sending it through the cortical structures V1, V2, MT, MST and eventually to the parietal cortex. The dorsal stream is primarily responsible for spatial awareness, determining the location of objects relative to the viewer as well as optical flow. While the dorsal stream does not have a direct impact on this research, the recognition and processing of optical flow is very important when processing motion. If the viewer is in motion, all objects within the visual field will also appear to be in motion, when in reality it is the viewer who is actually not in a fixed position. Understanding optical flow, or the movement of objects around the viewer, allows the brain to determine what is actually moving relative to the viewer. The ventral stream, also known as the "what" stream, retrieves visual information from the LGN which is then passed through the cortical structures V1, V2, V4, and finally the Inferior Temporal Cortex (IT). This research focuses specifically on the ventral stream since it is responsible for object and form recognition, including visual attention. The following sections will describe the structures of the ventral stream in further detail.

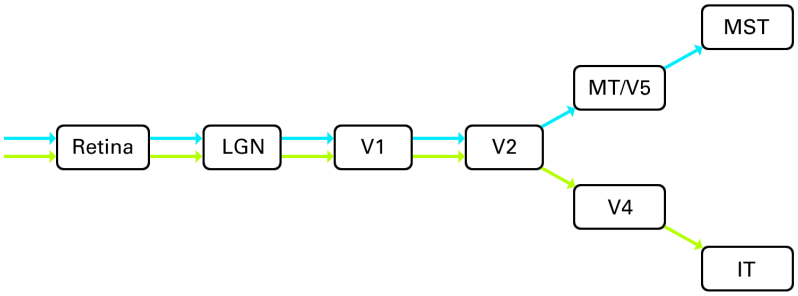

Figure 1: High level representation of the path visual information takes through the dorsal (blue/upper) and ventral (green/lower) streams of the visual task.

2.1 Retina / Lateral Geniculate Nucleus

While not part of the visual cortex, the retina is a very important structure involved in the visual task. The retina, technically part of the central nervous

system (CNS), is a small layer of photoreceptive cells in the back of each eye which responds to light and is primarily responsible for the collection of visual data. Visual information is first collected by the retina using two different photoreceptive cells, rods and cones. Rods are more numerous than cones and are much more sensitive than cones but cannot detect color. Cones are capable of determining color information, however they are not nearly as sensitive or numerous as rods. Small clusters of photoreceptors respond to visual stimuli by sending information either directly or indirectly to a bipolar cell. If sent directly, information is immediately sent from the photoreceptor straight into the bipolar cell. If sent indirectly, information is first passed through a horizontal cell, which inverts the signal before sending it to the bipolar cell. Since bipolar cells receive input from a cluster of photoreceptors, a center surround receptive field is created (figure 2) whose output is then sent along the optic nerve to the next structure in the ventral stream.

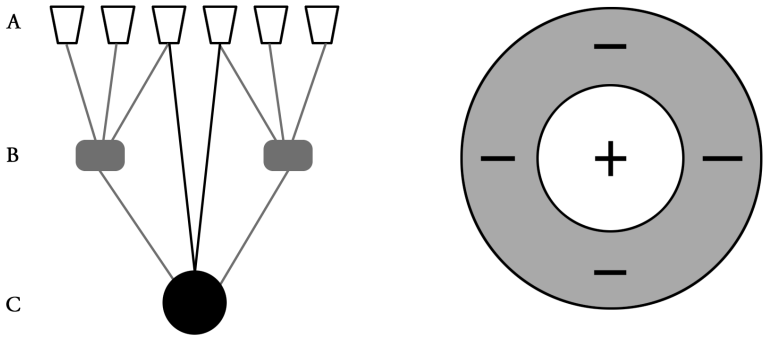

Figure 2: (Left)A: Photoreceptors receiving visual input B: Horizontal cells C: Bipolar cell receiving input from a cluster of photoreceptors (Right) Center/Surround receptive field that is created from these structures

The lateral geniculate nucleus (LGN) is the next area that visual information passes through on it's way through the ventral stream. The LGN is composed of six layers of cells which serve many different functions, but all together the LGN acts as a hub for information traveling to the visual cortex. The LGN receives input from the retina and sends output to the primary visual cortex, V1. In addition to receiving input from the retina, the LGN also receives strong feedback inputs from the primary visual cortex which allows it to have a sense of memory, or the ability to process new information along with information it has already received.

2.2 V1

Cortical area V1, otherwise known as the primary visual cortex has been the most studied region of the brain in regards to the visual task. The primary visual cortex is known to contain two types of cells, simple and complex, first discovered experimentally by Hubel and Wiesel [5]. Hubel and Wiesel determined that simple cells generate a receptive field that is sensitive to bars or edges which are orientation sensitive to a field of about 45 degrees. Hubel and Wiesel also speculated that the receptive fields they observed could be constructed by aligning a series of LGN center/surround receptive fields (figure 3). The receptive fields created by simple cells have been found to be effective at processing color differences, intensity contrasts and orientations.

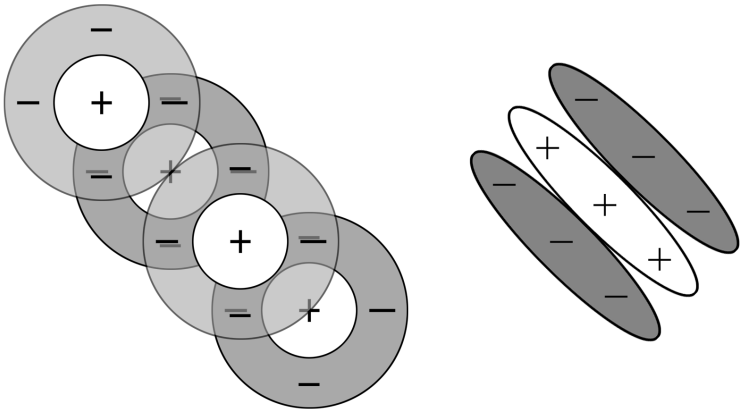

Figure 3: (Left)A series of center/surround receptive fields that have been aligned at a 45 degree angle (Right) The approximate receptive field that is created from such an alignment

Complex cells have been found to be similar to simple cells because they are also sensitive to bars or edges, however, complex cells are not as position sensitive as simple cells. The positional insensitivity of complex cells allows them to respond to moving bars or edges. It has been proposed by Hubel and Wiesel that the receptive field generated by complex cells can be created by aligning a series of simple cell receptive fields and integrating their responses over time (figure 4).

2.3 V2 / V4

Cortical areas V2 and V4 are the next two areas which visual information passes through as it traverses the ventral stream. Areas V2 and V4 have been found to

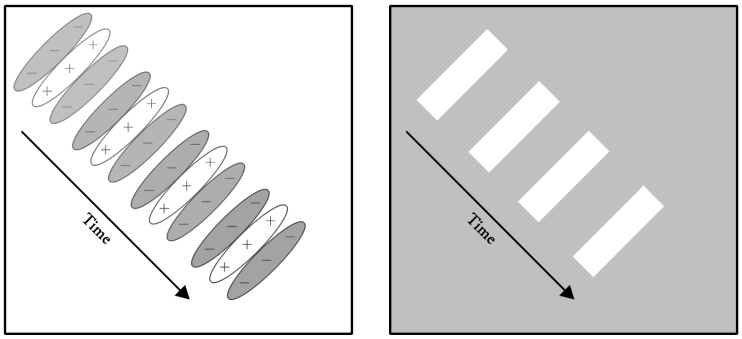

Figure 4: (Left)A series of receptive fields that are positioned differently at different periods of time (Right) A moving bar from top-left to bottom-right that would excite a receptive field described on the left

be functionally similar to V1 in that they process form, color, and orientation [12]. Even though areas V2 and V4 have a similar function to V1, they operate on the refined input coming from V1 which allows them to have a more global and higher level view of the information as it is being processed. In addition to performing similar functions to V1, neurons in V2 and V4 are also believed to be tuned for more complex properties such as the ability to determine whether or not a stimulus is part of the figure/background as well as simple geometric form recognition [18]. One of the more interesting aspects of V2 and V4 is that they are the first regions of the brain which show that their neuronal activities are modulated by by visual attention. Research has shown that the firing rates of V4 neurons can be affected by attended / unattended stimuli up to 20% whereas V1 neuron firing rates appear to not be affected by the type of stimulus [10]. Moran and Desimone's research can be used as strong physiological evidence that the attentional spotlight described by Tresiman may begin to occur in cortical areas V2 and V4.

2.4 Inferior Temporal Cortex

Once visual information reaches the Inferior Temporal Cortex (IT) basic features such as color and orientation have been extracted as well as a few more complex features such as shape. Visual attention has been shown to have an effect on the information arriving to IT from the other structures in the ventral stream. Research shows that the neurons in IT are very responsive to complex stimuli or parts of complex stimuli, leading to the belief that IT is responsible

for high-level object recognition. Edmund Rolls and his colleagues have been researching how and why neurons fire in IT by conducting studies on macaque monkeys [12]. Through experimentation they have determined that certain populations of neurons in IT encode for portions of faces or entire faces. They have also determined that the recognition in IT has a high amount of invariance, or the ability to detect objects regardless of size, contrast, position, or angle of view. Using this information, many different object recognition models have been developed which attempt to mimic IT's ability to recognize objects with a high level of invariance. Rolls himself has developed a recognition model called VisNet (Figure 5) which utilizes low level feature extractors, as seen in the Regina/LGN/V1, as the input for a hierarchical neural network which allows for object recognition with a high degree of invariance [13].

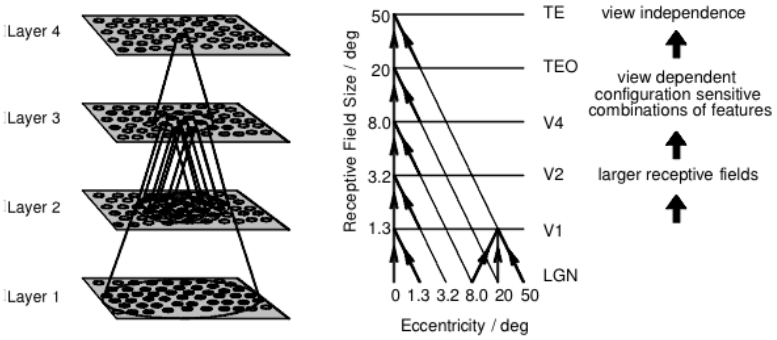

Figure 5: High level architecture of the VisNet model. Multiple features are detected at the bottom level with each successive level using groups of neurons from the previous level as input to build up feature invariance. (Source: Rolls07)

Thomas Serre and Tomaso Poggio from the Massachusettes Institute of Technology, have also developed a model which takes a similar biologically inspired approach by detecting and combining low level features using simulated simple and complex cells (figure 6) to build up invariance for object recognition [14]. Since these recognition models, as well as numerous others, utilize low level features extracted early in the visual task such as contrast, orientation, and color, a number of biologically plausible filters have been developed to mimic the low level feature detection found in the earliest stages of the visual task.

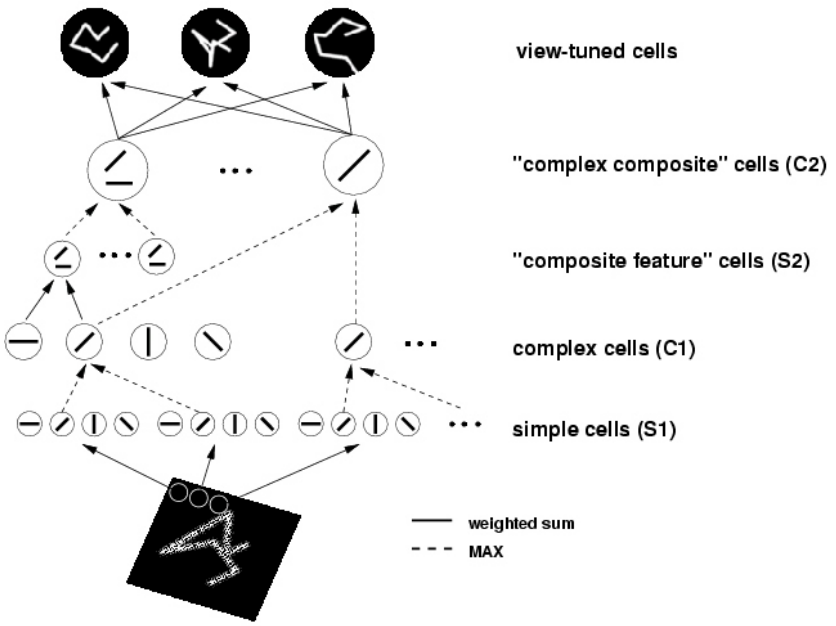

Figure 6: High level architecture of the Serre object recognition model. Layers of simple "S" and complex "C" cells are used to detect and combine similar low level features and positional information. Each layer adds invariance to position and scale. A neural network is used on the final layer of view tuned cells to determine what object is present in the input. (Source: Serre05)

3 Biologically Plausible Filters

By examining the effects that each cortical structure has on visual information as it is processed by the brain, several biologically plausible filters have been developed which model those effects. Used as the basic building blocks for more complex biologically based vision models, these filters have become staples in biologically inspired computing. Center/surround, orientation sensitive, and motion sensitive filters that very accurately portray their biological receptive field counterparts are explored in further detail in the following sections.

3.1 Center / Surround Filters

The center surround receptive fields formed in the earliest stages of the visual task have received the most scientific attention. Hubel and Weisel, who first began the study of receptive fields theorizing the existence of simple and complex cells, have had their research later expanded on by Christina Enroth-Cugell. Enroth-Cugell conducted experiments on cats, confirming the existence of these receptive fields showing that they display a Gaussian like tuning function [2]. By recording the responses from cat ganglion cell receptive fields in response to various visual grating patterns, Enroth-Cugell was able to show that the contrast sensitivity of the receptive fields could be accurately described as the difference of two Gaussian functions.

Using the discovery that the receptive fields can be described as a difference of two Gaussian functions, it is possible to create a biologically plausible filter which very accurately responds to the same input as the ganglion cell being studied. The difference of a wide Gaussian function and a narrow Gaussian function produces a new function that is very similar to a laplacian of Gaussian, or "mexican hat" function (figure 7).

The simple difference of Gaussian (DoG) function can be used to create a three dimensional wavelet suitable for use in computational modeling by rotating the function about it's center. The resulting wavelet (figure 8) can then be convolved with an input image and will respond strongly to regions of high center/surround contrast. The size of both the center and surround regions can be modified by altering the parameters to the DoG equation (Equation 1), which will then result in the filter being able to detect regions of different contrast. The type of contrast that the filter will detect can also be modified by changing the type of input that the filter is convolved with. Convolving the DoG filter with a intensity (grayscale) input will yield intensity contrast whereas convolving the filter with a red-green color difference input will yield the red-green contrast.

$$Equation 1: \qquad DoG(x,y) = \frac{c_{ex}^2}{2\pi\sigma_{ex}^2} e^{\frac{-(x^2+y^2)}{2\sigma_{ex}^2}} - \frac{c_{inh}^2}{2\pi\sigma_{inh}^2} e^{\frac{-(x^2+y^2)}{2\sigma_{inh}^2}}$$

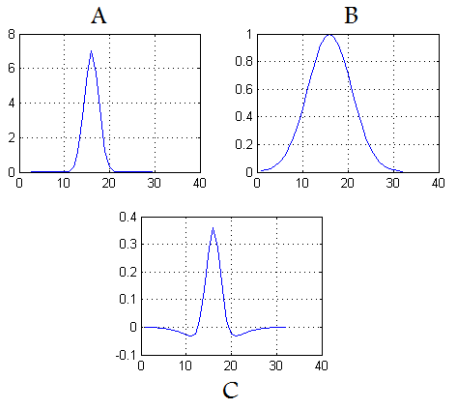

Figure 7: Taking the difference of a narrow Gaussian function (A) from a wide Gaussian function (B) creates a function that is very similar to a laplacian of Gaussian or "mexican hat" function (C)

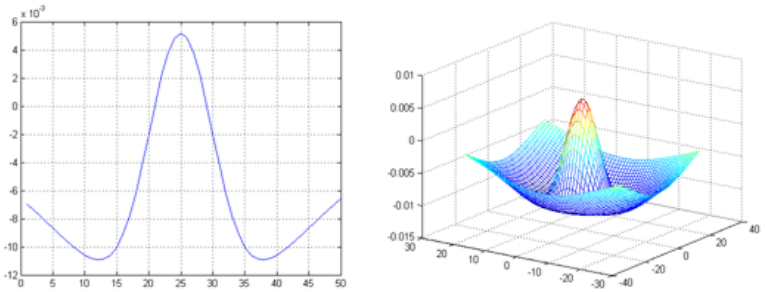

Figure 8: (Left) Side view of a DoG filter. (Right) Three dimensional plot of the DoG filter seen on the left.

3.2 Orientation Sensitive Filters

In addition to recognizing center/surround contrast, Hubel and Weisel also observed that some receptive fields responded strongly to bars or edges of high contrast instead of simple center/surround contrasts. As previously discussed, it is possible that a receptive field tuned to detect oriented bars can be generated by aligning and integrating the responses of a number of simple center/surround receptive fields. Gabor functions or wavelets have been found to be suitable approximations for these receptive fields (figure 9).

A Gabor function is defined as the product of a Gaussian function by a sinusoidal function (Equation 2). By altering the various parameters to the Gabor equation, the properties of the filter can be changed in a variety of ways.

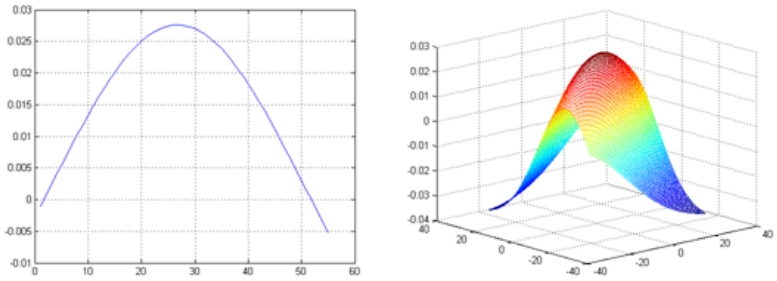

Figure 9: (Left) A simple plot of a Gabor function. (Right) Three dimensional plot of the Gabor filter seen on the left, suitable for detecting orientated bars.

By changing the θ value, the orientation of the bars the filter responds to can be modified. Altering the λ or σ values will change the wavelength and frequency of the sinusoidal function, making the filter to respond to differing numbers and sizes of bars

$$
\begin{aligned}
Equation2: \quad & G(x,y) = exp(\frac{-(X^2+\gamma^2Y^2)}{2\sigma^2})cos(\frac{2\pi}{\lambda}X) \\
& where \\
& X = xcos(\theta) + ysin(\theta) \\
& Y = -xsin(\theta) + ycos(\theta)
\end{aligned}
$$

The filters produced by Gabor functions have been experimentally shown to respond almost identically to their biological counterparts. John Daugman, from Harvard University, illustrated the accuracy of Gabor filter approximations by taking the differences of receptive fields experimentally measured in cats from the best fit Gabor filter approximations of those same receptive fields [1]. The differences in the experimentally measured filters and the Gabor approximations were found to be smaller than the amount of error introduced when measuring the receptive fields in the cats (Figure 10).

3.3 Motion Sensitive Filters

The complex cells hypothesized by Hubel and Weisel have been found to be similar to simple cells in that they detect oriented bars, however they are less positionally sensitive than simple cells allowing them to respond strongly to motion. Richard Young and Ronald Lesperance investigated these receptive fields and developed a model that creates filters which closely approximate many

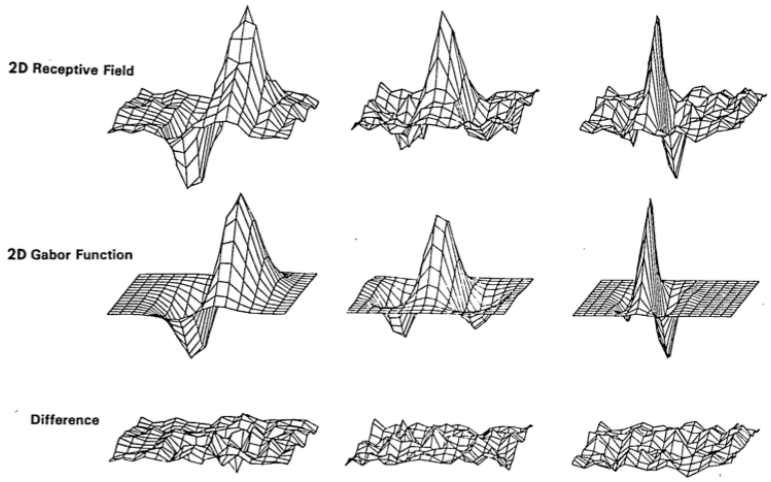

Figure 10: Top: Receptive fields measured in cat striate cortex. Middle: Best fitting Gabor function to each receptive field. Bottom: Difference between the actual receptive field and the Gabor approximation. (Source: Daugman85)

experimentally measured motion sensitive receptive fields [19]. The model created by Young and Lesperance, termed the Gaussian Derivative spatio-temporal model (GD Model), expands on the idea of using Gaussian tuning functions to detect bars or edges, by introducing the aspect of time. A three-dimensional spatio-temporal filter is created by taking the product of three one-dimensional derivative of Gaussian functions (Equation 3).

$$Equation3: \qquad G_{n,o,p} = (x', y', t') = g_n(x')g_o(y')g_p(t')$$

In the equation above x', y' and t' are the three axes of the final filter G, while n, o, and p are the number of derivatives to take for each of the component Gaussian functions $g()$. By changing the number of derivatives taken (Equations 4,5,6, and 7) or the θ / σ values from each of the individual base Gaussian functions, a multi-lobed filter is created (figure 11) which can be used as an accurate approximation to a number of biological receptive fields. Each lobe in the filter is either positive or negative. By taking slices of the filter along the time axis, a sequence of filters at different instances in time are created which are very similar to the Gabor filters used to detect stationary bars or edges (figure 12). Convolving each of these "slice of time" filters with input images taken at different instances in time and later combining the results from each individual convolution produces an effective motion sensitive filter. Similarly to how John Daugman compared the accuracy of Gabor function based receptive fields to their biological counterparts, Young and Lesperance computed the difference

between experimentally measured receptive fields and the model approximations [20] to show the accuracy of the GD spatio-temporal model (figure 13).

$$
\begin{aligned}
Equation4: \quad & g_1(x) = -xg_0(x) \\
Equation5: \quad & g_2(x) = (x^2 - 1)g_0(x) \\
Equation6: \quad & g_3(x) = -(x^3 - 3x)g_0(x) \\
Equation7: \quad & g_4(x) = (x^4 - 6x^2 + 3)g_0(x)
\end{aligned}
$$

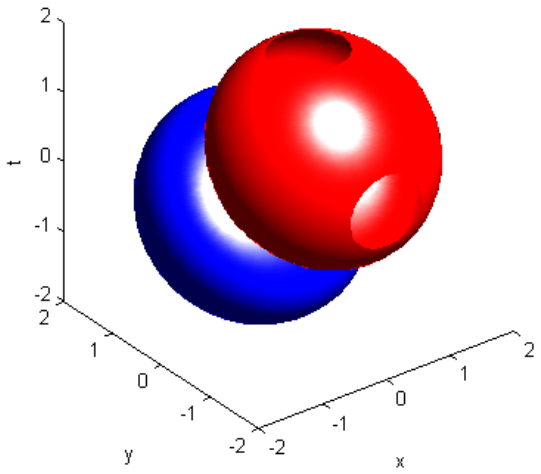

Figure 11: A multi-lobed receptive field created using n=1,o=0, and p=0. Red lobes represent positive values while blue lobes represent negative values.

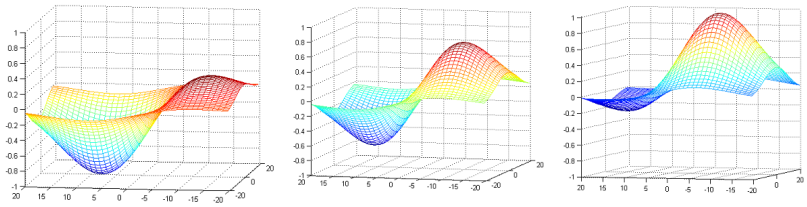

Figure 12: Slices of the filter seen in figure 11 taking at time instances t=13,t=21, and t=29. The filter has 41 total slices.

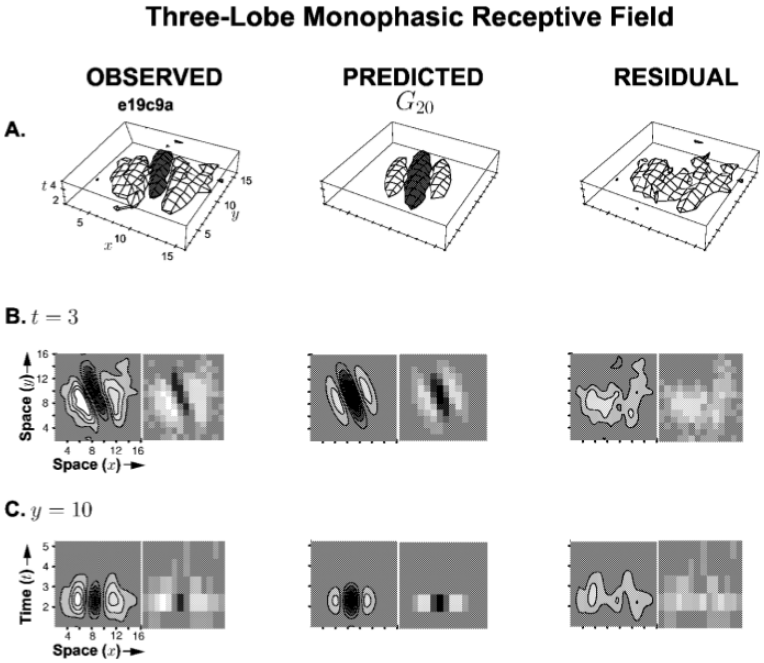

Figure 13: Left: Experimentally measured motion sensitive receptive field. Middle: Gaussian Derivative approximation receptive field. Right: Difference between the approximation and actual receptive field. (Source: Young01)

4 Previous Research

By combining low level feature extraction filters with a higher level knowledge of the vision task, several biologically inspired vision models have been developed which either calculate or utilize the concept of the saliency, or conspicuousness, of an input. Three iteratively more advanced saliency models are discussed in the following sections starting with a basic still saliency model developed by Koch and Ullman. A model developed by Itti and Koch, which uses a set of more advanced feature combination strategies, is then discussed in detail. Lastly, the saliency model used in VENUS, a modern object novelty and tracking model developed at the Rochester Institute of Technology, is explored.

4.1 Koch and Ullman

The concept of a computer generated saliency, or conspicuousness, map was first described by Koch and Ullman in their 1985 paper "Shifts in selective visual attention: towards the underlying neural circuitry" [8]. Having knowledge of the research conducted by numerous psychologists with regards to visual attention, Koch and Ullman proposed a computational model using three basic concepts, which in conjunction can be used to select regions of interest from visual input (figure 14).

The first concept introduced by Koch and Ullman, derived in part from Anne Treisman's Feature Integration Theory [16], proposes that multiple elementary features are computed in parallel and then stored in a number of different topographical maps. The second concept of the proposed model is that one of the functions of selective attention is to fuse the various topographical feature maps into a single cohesive map otherwise known as a saliency map. The final concept the model describes that only certain sections of the saliency map are selected for further processing. Selection can be accomplished by using a winner-take-all network based on the conspicuity of the locations on the saliency map. Koch and Ullman have also proposed that shifts in attended regions can be accomplished by inhibiting the currently selected region as well as implementing rules for similarity and proximity preference. Similarity and proximity preference rules follow the ideas of perceptual grouping in that shifts in attention are more likely to occur between regions that are spatially close to each other as well as btween regions that share similar features.

4.2 Itti and Koch

Using the model proposed by Koch and Ullman, Laurent Itti and Christof Koch have developed numerous saliency based attention models. As part of their ongoing research, Itti and Koch have explored various ways in which the large

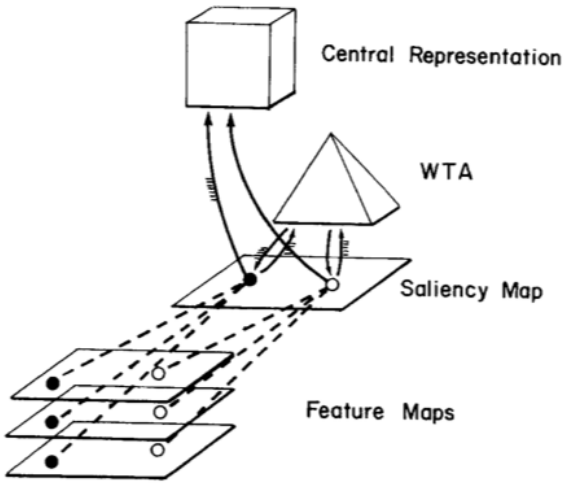

Figure 14: Multiple feature maps are computed at the lowest level of the model which are later combined into a saliency map. A winner-take-all network is then used to select a region of interest from the saliency map to send to the central representation for further processing. (Source: Koch85)

number of basic feature maps in a saliency model can be combined into a single, cohesive, saliency map [6]. Itti and Koch have described a number of different ways in which feature maps can be combined, from simple summation of features to the supervised learning of feature map weights. The "naïve" approach to feature combination described by Itti and Koch is to simply normalize each feature map between zero and one and then add them all together and re-normalize. The problem with the "naïve" approach is that each feature map has a different dynamic range, allowing a feature which might be prominent in one map to be "washed out" by a number of non-prominent features in different maps. Itti and Koch have explored three other methods for combining feature maps: learned weights, global amplification, and localized interactions.

The learned weights approach requires that the model undergo supervised learning to detect a specific class object. During training, each feature map is given a different weight based on the prominence of that specific feature in a set of training images of the specific items the saliency map should be able to find. The weights assigned to each map are then used to scale each feature map as they are combined into a saliency map. Itti and Koch have found that this approach is very successful, however it requires the model to be trained before hand which means that it can only recognize a specific class of object.

A second combination strategy explored by Itti and Koch, called global

amplification, has the goal of normalizing all feature maps to the same dynamic range before combining them into a saliency map. The global amplification is accomplished by finding the global maximum on each feature map, M, and the average of all other local maxima on each each feature map, $\bar{m}$. Each feature map can then be scaled by: $(M - \bar{m})^2$. A problem with this approach to feature combination is that a global maximum as well as the average of all local maxima must be found for each feature map. Calculating global maxima is not very biologically plausible since neurons early in the visual process only receive information from small spacial regions surrounding them.

The third approach explored by Itti and Koch takes a more biologically inspired approach by simulating local competition between the neurons in each feature map. After normalizing each feature map between zero and one, a wide difference of Gaussian filter is convolved multiple times with each feature map. The convolution of a DoG filter with each feature map simulates local competition, or lateral inhibition, by suppressing regions of uniform feature strength while enhancing regions of non-uniform strength (figure 15).

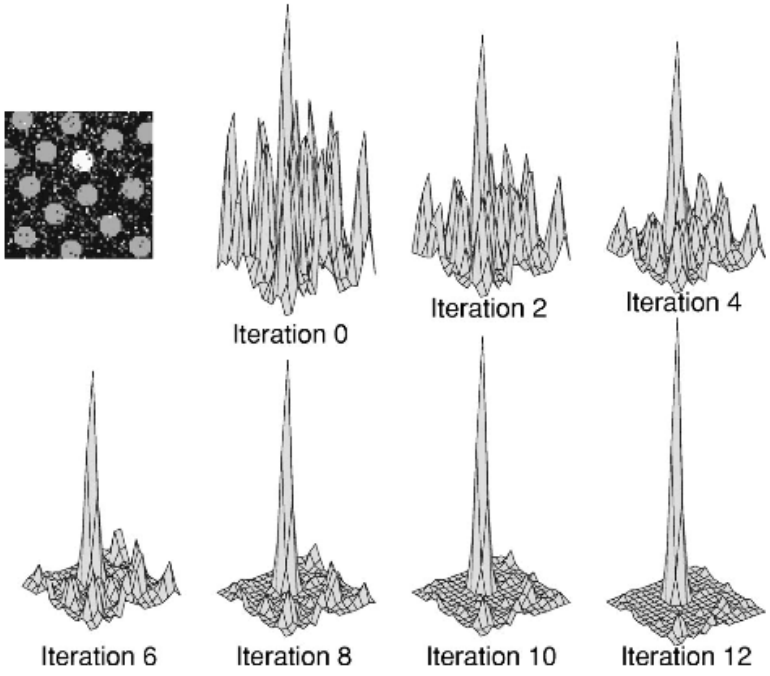

Figure 15: A feature map being iteratively convolved twelve times with a DoG filter. Each iteration strengthens the non-uniform area of the feature map while suppressing the uniform areas. (Source: Itti01)

By recording the amount of simulated time it took for saliency models using

each of the feature combination strategies to find a target object in an input, Itti and Koch were able to compare each of the feature combination strategies (figure 16). Their results show that the "naïve" method is consistently worse than any of the other three methods, while the learned weights method is most often the best strategy. The global amplification and and localized interactions combination strategies perform very similarly in many of the test cases. In cases where a model is not designed to detect specific classes of objects, the learned weights method will not perform well and one of the other strategies should be utilized.

	Naive	$\mathcal{N}(.)$	Iterative	Trained
Red can	2.90±2.50	1.67±2.01	1.24±1.42	0.35±1.03
Triangle	2.44±2.20	1.69±2.28	1.42±1.67	0.87±1.29
Traffic[a]	1.84±2.13	0.49±1.06	0.52±1.05	0.24±0.77
Traffic[b]	3.26±2.80	1.27±2.12	0.70±1.18	0.77±1.93

Figure 16: The amount of simulated time (in seconds) taken to find a target object in an image using each of the feature combination strategies. Note: $N(.)$ represents the global amplification strategy. (Source: Itti01)

4.3 VENUS

The Video Exploitation and Novelty Understanding in Scenes (VENUS) project is an ongoing project at the Rochester Institute of Technology under the direction of Dr. Roger Gaborski which utilizes a saliency based selection mechanism to find, extract and categorize novel events occurring in video [3]. The saliency model utilized in the VENUS project is similar to the the models developed by Itti, Koch, and Ullman which use color, orientation, and intensity feature maps to create a still saliency model. However, the VENUS saliency model also utilizes the motion sensitive filters developed by Young and Lesperance to create a motion based saliency map (figure 17). The VENUS project computes still and motion saliency as two different entities from a video stream which are then separately operated on by learning and novelty detection models.

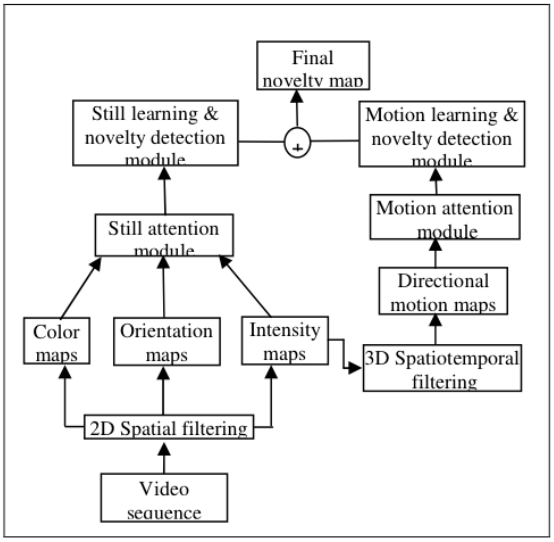

Figure 17: The saliency model utilized by the VENUS project. In addition to a saliency map created from contrast, orientation, and intensity feature maps the VENUS project utilizes motion saliency as well. (Source: Gaborski04)

5 Model Definition

The model created and tested in conjunction with this research is the combination and extension of many of the key ideas utilized by the developers of previous saliency based models. The purpose of this model is to produce an unconscious, target unspecific, search through a video, selecting small regions of interest for each frame. The selection of the target region is based on a number of different criteria including: still and motion saliency, habituation, proximity preference, and perceptive grouping. Using the regions of interest found by the models' search through a video, an object recognition model can then be used to classify the individual objects found in each of the small selected regions of the video without having to process the entire frame. The following sections will discuss the saliency model used to select regions of attention, how shifts between attended regions are accomplished, the final output of the model as well as how the attention model can be used in conjunction with a modern object recognition system.

5.1 Still Saliency

The still saliency module developed for this research is based on the model proposed by Koch and Ullman as well as the still saliency model used in the VENUS project. Using a single RGB color image as input, the purpose of the still saliency module is to produce three different feature maps: intensity contrast, color contrast, and orientation contrast (figure 18). Each of the three feature maps produced by the still saliency module are the combination of many smaller, sub-feature, maps. The three feature maps produced by the module can be later integrated to create a final saliency map. The following sections discuss the creation of each of the three feature maps in further detail.

5.1.1 Intensity Contrast Feature Map

As the name suggests, the purpose of the intensity contrast feature map is to show which areas in the input have regions of high intensity contrast. The input used for the still saliency module is a single RGB color image, usually a single frame taken out of a video stream. In order to calculate the intensity contrast of the input, the RGB image must first be converted into an intensity based, or grayscale, image. The creation of the intensity image can be accomplished by averaging each of the three component color planes of the image (equation 8, figure 19).

$$Equation 8: \qquad Im_{intensity} = \frac{Im_{red} + Im_{green} + Im_{blue}}{3}$$

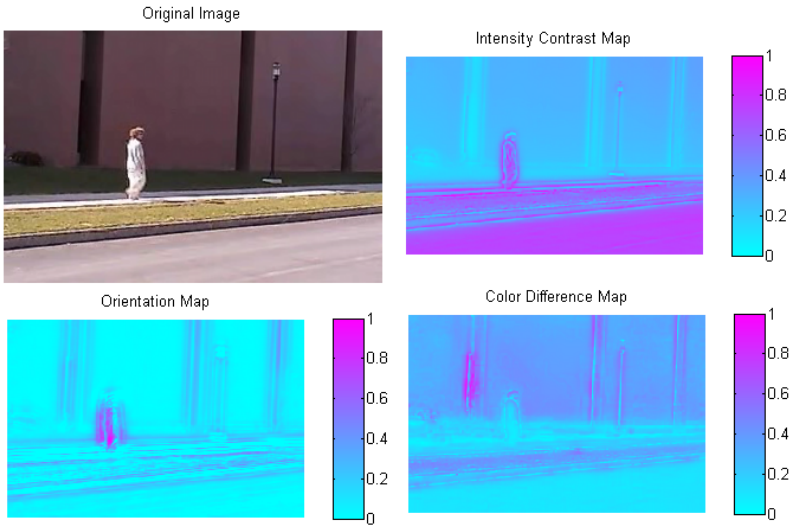

Figure 18: A frame of test video along with the intensity contrast, orientation contrast, and color contrast feature maps

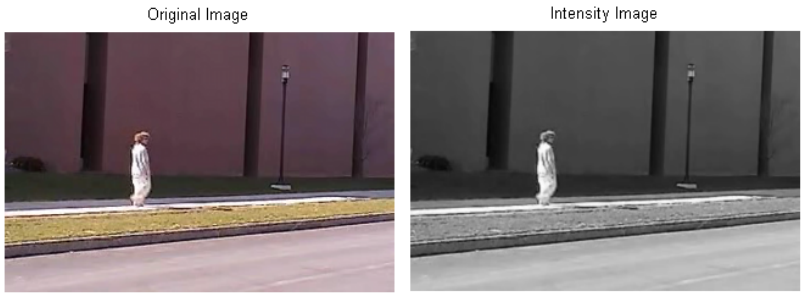

Figure 19: (Left)Original color image used as input to still saliency module (Right) Intensity based, grayscale, image

Once an intensity image has been created from the initial input, it is convolved with a series of three difference of Gaussian filters to find the regions of the image which have high center/surround contrasts. Three sub-feature maps are created by using differently sized DoG filters (table 1) for the convolutions, each with the purpose of finding areas of progressively larger contrast (figure 20).

Table 1: Difference of Gaussian Filter Parameters

Size	σ_{ex}	σ_{inh}	c_{ex}	c_{inh}
8x8	0.32	1.28	3.86	10.3
16x16	0.8	2.4	26.14	36.2
32x32	1.6	4.8	112.6	144.8

Each of the sub-feature maps are then scaled to the same dynamic range by dividing each map by maximum response of the DoG filter used in it's creation. The maximum response of each filter can be found by summing all the positive values in the filter. After each sub-feature map has been scaled to the same dynamic range, the maps are combined into the final intensity contrast feature map by adding them together and normalizing the resulting map's values between zero and one.

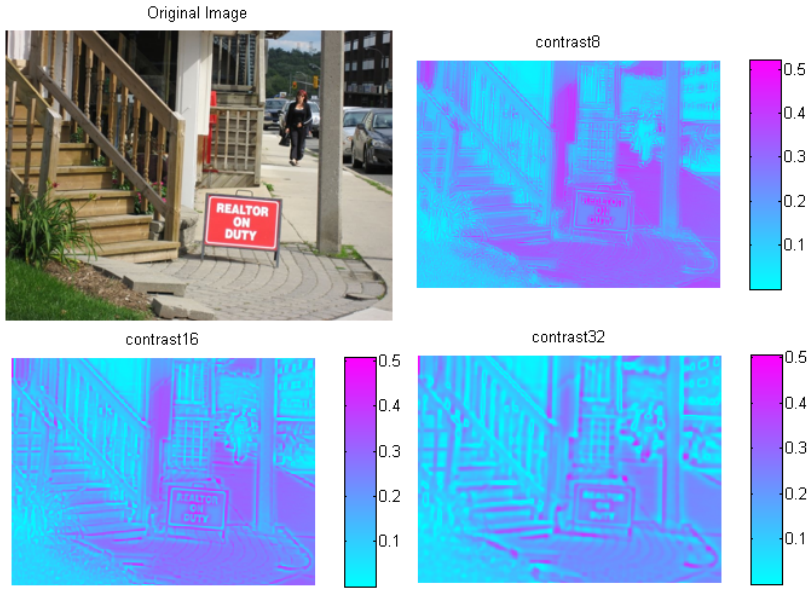

Figure 20: Original image and intensity contrast sub-feature maps created with different sized Dog filters.

5.1.2 Orientation Contrast Feature Map

The orientation feature map is created in a very similar way to the intensity contrast feature map. The input image is converted to an intensity image which is then convolved with a series of three progressively larger Gabor filters at four different orientations per filter size (table 2, figure 21). Similar to the intensity contrast process, the twelve sub-feature maps are scaled to the same dynamic range by dividing each map by the maximum response of the Gabor filter used to create it. Once scaled to the same dynamic range, each of the sub-feature maps are further convolved by the three DoG filters used in the intensity contrast calculations. By convolving the Gabor filtered results with center/surround filters, regions of high orientation contrast are able to be shown from the input. The final orientation feature map is created by adding all 36 of the sub-feature maps together, normalizing the results to the range of zero to one.

Table 2: Gabor Filter Parameters

Size	θ	σ	λ
7x7	0°	2.8	3.5
7x7	45°	2.8	3.5
7x7	90°	2.8	3.5
7x7	135°	2.8	3.5
15x15	0°	6.3	7.9
15x15	45°	6.3	7.9
15x15	90°	6.3	7.9
15x15	135°	6.3	7.9
31x31	0°	14.6	18.2
31x31	45°	14.6	18.2
31x31	90°	14.6	18.2
31x31	135°	14.6	18.2

5.1.3 Color Difference Contrast Feature Map

The color difference feature map is created by finding regions of high red-green and blue-yellow contrast. Red-green and blue-yellow contrast are used in the saliency computations since it has been found that humans, as well as other mammals, show sensitivity in V1 to these color contrasts [11] . Red-green and blue-yellow color difference maps are created by computing the red, green, blue, and yellow color channels (Equations 9,10,11,12) from the original input. Two sub-feature maps are then created by subtracting the green channel from the red channel, and the yellow channel from the blue channel (figure 22)

$$Equation 9: \qquad Channel_{red} = Im_{red} - \frac{(Im_{blue} + Im_{green})}{2}$$

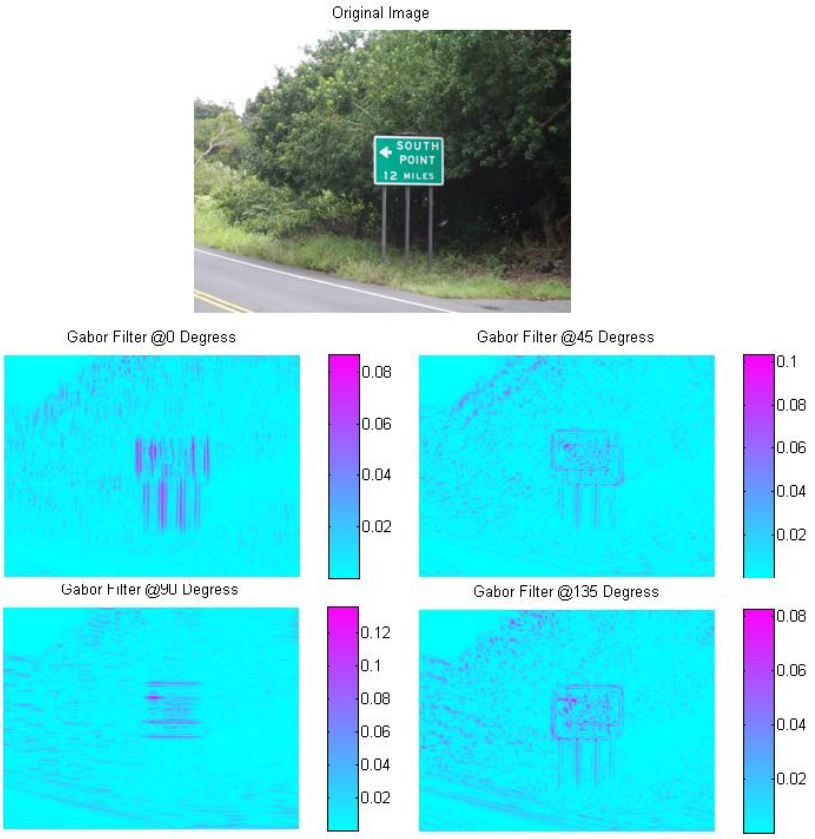

Figure 21: Original image and four different orientation sub-feature maps (0, 45, 90, and 135 degrees).

$$Equation10: \quad Channel_{green} = Im_{green} - \frac{(Im_{red} + Im_{blue})}{2}$$

$$Equation11: \quad Channel_{blue} = Im_{blue} = \frac{(Im_{red} + Im_{green})}{2}$$

$$Equation12: \quad Channel_{yellow} = Im_{red} + Im_{green} - (2 * abs(Im_{red} - Im_{green}) + Im_{blue})$$

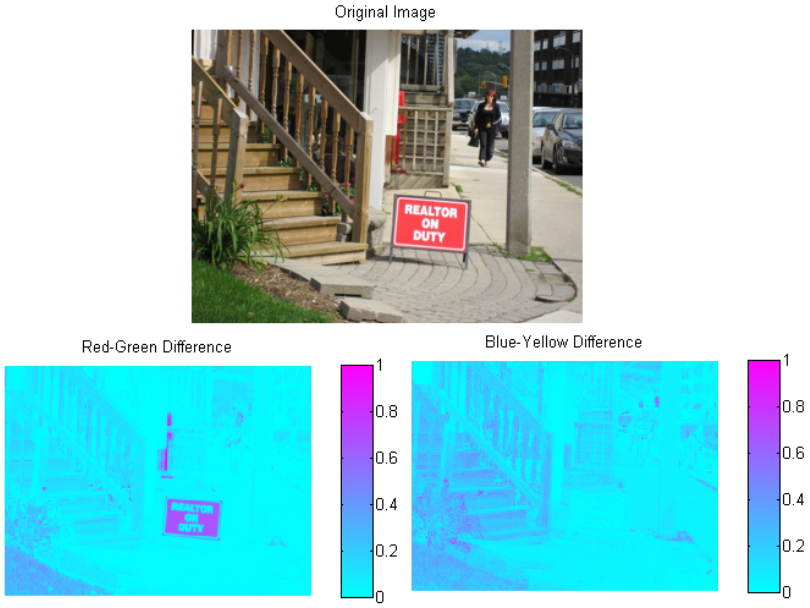

Figure 22: (Top)Original Image (Left)Red-Green Color Difference Map (Right)Blue-Yellow Color Difference Map

Using the two sub-feature maps created by taking the red-green and blue-yellow differences, regions of high contrast are extracted from each map by taking the absolute value of the convolution of each map with the different DoG filters used in the creation of the intensity and orientation feature maps. The absolute value of the convolution is used since it does not matter in terms of saliency weather or not an area is more red than green versus more green than red. The final color difference map is calculated by adding the six individual sub-feature maps together and normalizing the result to the range of zero to one.

5.2 Motion Saliency

The motion saliency module is used to produce a single feature map based on the movement of objects in a short segment of video. Unlike the still saliency module which operates on a single frame of video, the motion saliency module

requires that a series of frames from a video be used as input. The creation of the motion feature map requires that the section of video be first processed by a special case of spatio-temporal filter called a blink filter. The results from the blink filtering are then processed by four motion sensitive spatio-temporal filters to create four sub-feature maps, one for movement in each of the primary directions (0, 45, 90, and 135 degrees). The four directional maps are then scaled to the same dynamic range and integrated into a final motion map as output for the motion saliency module. Each of the stages to create a motion feature map are explored in further detail in the following sections.

5.2.1 Blink Filtering

The first step in creating a motion feature map is to use a blink filter to try to to remove the majority of the non-moving objects from the input. A blink filter is a special case of spatio-temporal motion sensitive filter, specifically tuned to detect objects that exist at one moment in time and then cease to exist the next. Called a blink filter for it's ability to detect blinking objects, the blink filter it is also capable of detecting moving objects since the edge of an object will exist in one location and then cease to exist at the same location shortly thereafter.

Creating and using a blink filter is accomplished the same way one would create any type of spatio-temporal filter (table 3). Directional sensitive motion filters are created with a σ value of about +/- 45 degrees which represents the speed of motion being detected. A blink filter, however, is created using a σ parameter value of 0 degrees meaning that no lateral motion is detected. The filter simply has a large positive lobe followed by a large negative lobe later in time (figure 23). If convolved with a non-moving object, each lobe will produce the opposite response of the other, producing no response for the filter.

Table 3: Blink Filter Parameters

Size	n	o	p	θ	ϕ
15x15x15	1	0	0	90°	90°

The result of processing a short segment of video with a blink filter is that each frame in the video will have the non-moving, non-blinking, objects removed. What remains in the video is an intensity based image where dark areas, values close to zero, represent non-moving objects and lighter areas (values near one) represent moving or blinking objects (figure 24).

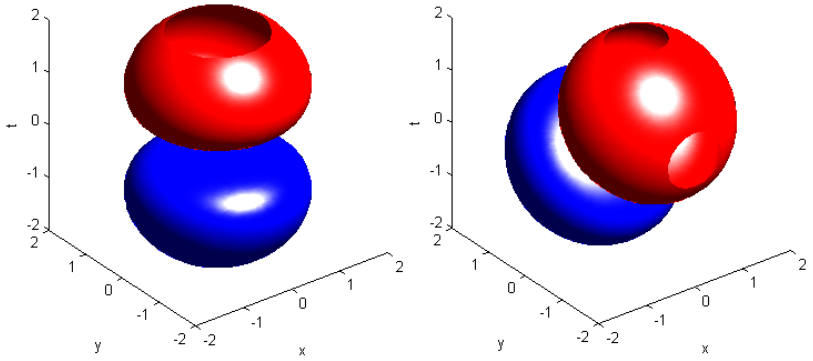

Figure 23: (Left) Blink filter: change only occurs along the time axis (Right) Normal spatio-temporal filter: change occurs along the X-Y plane

Figure 24: (Left)Cars moving across the screen with arrows overlayed to show the direction of motion (Right) Results of filtering the video on the left with a blink filter

5.2.2 Directional Filtering

After a video segment has been simplified to moving or blinking objects, regular spatio-temporal filters (table 4) are used to create four sub-feature maps for motion in each direction. Each of the spatio-temporal filters responds to movement in one of the four major directions of motion at a speed of approximately one pixel per frame. More spatio-temporal filters can be used to detect different speeds of motion or motion in other directions by changing the θ σ parameters. Only four spatio-temporal filters were used to create the motion feature map in this model because three dimensional convolutions are very computationally expensive and the added accuracy does not warrant the added computational complexity.

Table 4: Spatio-Temporal Filter Parameters

Size	n	o	p	θ	ϕ
15x15x15	1	0	0	0°	45°
15x15x15	1	0	0	45°	45°
15x15x15	1	0	0	90°	45°
15x15x15	1	0	0	135°	45°

The convolution of the blink filtered input with a spatio-temporal filter yields an intensity map whose values have a wide dynamic range in which high values represent regions which correspond strongly to the spatio-temporal filter it was convolved with. Similar to the creation of the other feature maps, all of the motion sub-feature maps are scaled to the same dynamic range by dividing each map by the maximum response of the spatio-temporal filter used to create it. Once each motion map has been scaled to the same dynamic range, the maps are added together and normalized between zero and one to create the final motion based feature map.

5.3 Feature Map Integration

Once the four feature maps have been created, the task of integrating them into a coherent saliency map remains. The VENUS model combines feature maps by adding them together and normalizing the result. However, as shown by the research of Itti and Koch, this naïve approach to feature map combination is less than optimal. Each of the individual feature maps has it's own dynamic range and by simply adding them together and normalizing the result, a prominent feature in one map may be washed out by a less prominent feature in a different map.

To avoid the problem of feature washout, local competition is used to bring each of the four feature maps into the same dynamic range before combining them. The local competition approach used in this model differs from the

method proposed by Itti and Koch [6] in that local competition is applied to only the four feature maps used to create the final saliency map instead of all 45 sub-feature maps used to create the feature maps. Local competition is simulated on each feature map by passing a wide difference of Gaussian filter across the map which inhibits regions of uniform intensity and excites regions of differing intensities (table 5).

Table 5: Local Competition DoG Filter Parameters

Size	σ_{ex}	σ_{inh}	c_{ex}	c_{inh}
0.25 * Im_{width}	0.02 * Im_{width}	0.25 * Im_{width}	0.5	1.5

Each feature map is convolved with the wide DoG filter a single time, differing from the iterative approach taken by Itti and Koch. The approach to local competition used in this model differs from the work of Itti and Koch mainly due to speed considerations. The convolution operation is a very time consuming process and by simulating local competition only among the four main feature maps the number of convolutions necessary for local competition can be reduced from 45 to 4. It is also not completely necessary to simulate local competition among each sub-feature map since sub-feature maps can be brought to the same dynamic range by scaling each sub-feature map by the maximum response of the filter used to create it. A single pass convolution is used to simulate local competition in this model instead of the twelve pass iterative method proposed by Itti and Koch for two reasons: speed and accuracy. Even though performing a single pass convolution is clearly a significant speed improvement over a twelve pass iterative convolution, both approaches yield similar experimental results.

Once each feature map has been processed using local competition, they are considered to be in the same dynamic range. The four feature maps can then be added together and normalized to create a cohesive saliency map (figure 25). Had the low cost implementation of Itti and Koch's local competition research not been used to bring the feature maps into the same dynamic range, the saliency map created using the naïve approach would contain a significant amount of excess noise as well as the possibility of washed out features (figure 25).

5.4 Focus of Attention

Once a cohesive saliency map has been created for a frame of video, a region of high salience can be extracted and later used by another system to classify the contents of that region. The process of extracting a region from each from of video is a difficult task since the purpose of the model is to produce an unconscious search through video. The approach taken by Itti and Koch to select regions of interest from their saliency models is to make all selected regions a uniform size around a single point of high saliency (figure 26). While the method

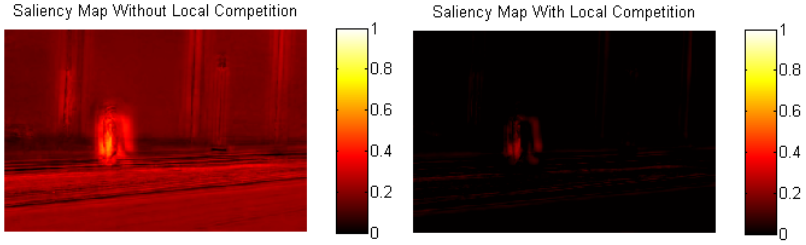

Figure 25: Before Competition, After Competition (Feature Maps) - With competition saliency map, without competition saliency map

used by Itti and Koch is very computationally efficient, it is very likely that objects of interest may be much larger or much smaller than the predetermined region size. For optimal use by a recognition system, regions of interest should be selected as closely as possible to the actual size of the objects they are focusing on.

Figure 26: Results from a saliency model created by Itti and Koch. All regions of attention are a uniform size that may or may not fit around the object being focused on. (Source: Itti00)

Without the knowledge of the content of the video or the objects being extracted, the model developed for this research improves upon the Itti and Koch method by estimating both the location and size of the region to extract using low level computer vision techniques. Due to the use of estimation when extracting regions of attention, each region may not always be optimal in regards

to the object being focused on. In the ideal case, the model focuses very tightly on a single object in the input (figure 27). It is possible, however, that a region of attention may be very large and encompass many smaller objects (figure 27),that a region only contains part of an object (figure 27), or that the region contains no objects of interest at all (figure 27). While not always producing optimal regions for each object to be extracted, the model produces optimal regions frequently enough that it should out perform the Itti and Koch method in most cases.

Figure 27: (Top Left) Optimal selection of object (Top Right) Partial selection of object (Bottom Left) Many small objects selected (Bottom Right) No objects of interested selected

To begin estimating the location and size of a region to focus attention on, the saliency map is first simplified and later enhanced. Initially, the saliency map contains any number of values ranging from zero (low salience) to one (high salience). The saliency map is simplified by thresholding the saliency map against a value of 0.5, converting the dynamic saliency map into a binary saliency map where salient areas are represented as ones and non-salient areas are represented as zeros (figure 28).

Once the saliency map has been simplified into a binary representation of the saliency, the map is then enhanced by performing an image closure on it. The purpose of an image closure is to remove the small holes in the binary representation of the saliency map as well as connect some of the smaller salient regions so that several distinct "blobs" are formed (figure 28). A structuring element in the shape of a disk with a diameter of 10% of the width of the image is used to close the image. The size of the structuring element used to perform

the image closure is somewhat arbitrary since it is not possible to know before hand the size of the objects that are to be extracted from the video.

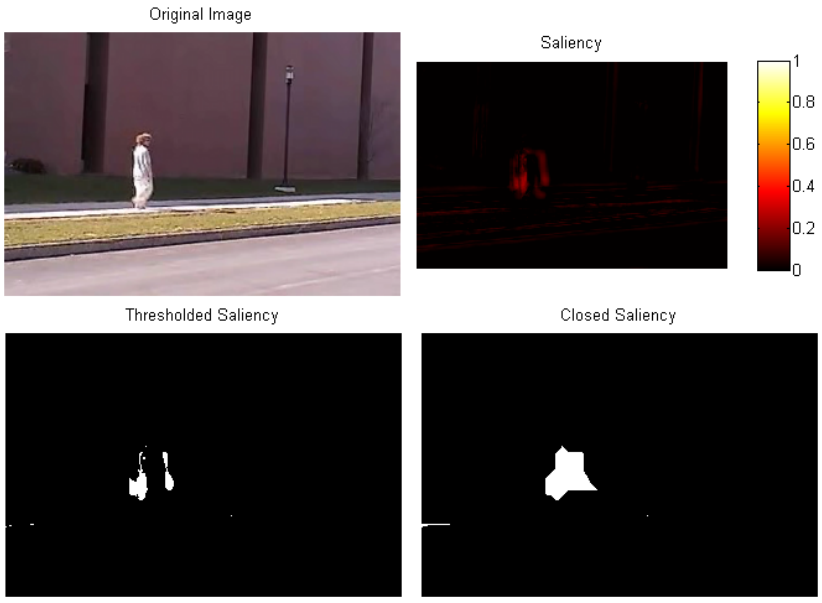

Figure 28: (Top Left)Original Image (Top Right)Saliency Map (Bottom Left)Thresholded Saliency Map (Bottom Right)Thresholded Saliency Map After Closure

Each individual region of salience is now considered an object that can be extracted. To determine which of these potential objects should be extracted, two factors are considered: average saliency and region size. Using the original saliency map in conjunction with the binary representation of the saliency map, the average saliency of each region can be calculated by finding the bounding box of each binary region and averaging the values in that bounding box on the original saliency map. The area to focus on is found by taking the region that has the highest average salience that also has an area larger than 25 pixels. A minimum area requirement is added to the focus of attention model because regions of very few pixels will most frequently have the highest average salience. Extracting very small regions is unwanted since objects of interest in video are infrequently very small and even if they were very small, most object recognition systems will most likely be unable to recognize objects of that size. The bounding box used to calculate the average salience of the object is used to extract the region of interest from the original input image. In the case that no suitable regions are found, the initial threshold for creating the binary representation of the saliency map is lowered by 0.1 and the process is repeated until a suitable region is found to focus on.

5.5 Attention Shifts

Once a region for the current frame has been focused on and extracted by the model, the next task is to shift the models attention to a new region in the next frame. It may seem, at first, that the easiest way to accomplish this is to just create a new saliency map for the next frame and select a new region using the same selection mechanism as before. Unfortunately, a significant problem with this approach is that if a video does not change from frame to frame the same saliency map will not change either, meaning that the same region will be selected from frame to frame. The purpose of the model is to produce a search through video, meaning that this problem must be overcome.

To force the model to find different regions each frame, the concepts of habituation and dishabituation are implemented. In psychology, habituation is the lowering of a response when repeatedly presented with the same stimulus. Dishabituation is the strengthening of a response that was previously weakened due to habituation. Dishabituation occurs when the stimulus that first evoked habituation is no longer present. Once a region has been the focus of attention, the model habituates to that region, lowing any future response produced by that region so that it is less likely to be focused on again. Once an attentional shift has occurred and a region is no longer the focus of attention, the region dishabituates, incrementally strengthening the regions' response back to it's original strength, giving that region a higher chance to be re-focused on.

In addition to habituating and dishabituating regions as they become the focus of attention, the model also implements two of the attentional shifting preferences described by Koch and Ullman: proximity preference and similarity preference. Proximity preference describes that shifts in attention are more likely to occur between regions that are spatially close together while similarity preference describes that shifts in attention are more likely to occur between regions that share similar basic features [8]. The following sections describe the implementation of each of these features.

5.5.1 Inhibition and Excitation of Attended Regions

The tasks of habituation, dishabituation, and proximity preference are all implemented in tandem by the model. Each time a region is selected by the model, information about that region is added to a list of recently selected regions, or short-term memory. Before the next region is selected, the model cycles through all of the previously selected regions in memory and applies a oversized, negative, difference of Gaussian filter (table 6, figure 29) to the location of each previously attended region on the saliency map.

The negative DoG function is created so that the entire negative portion of the filter encompasses the previously attended region while the positive areas

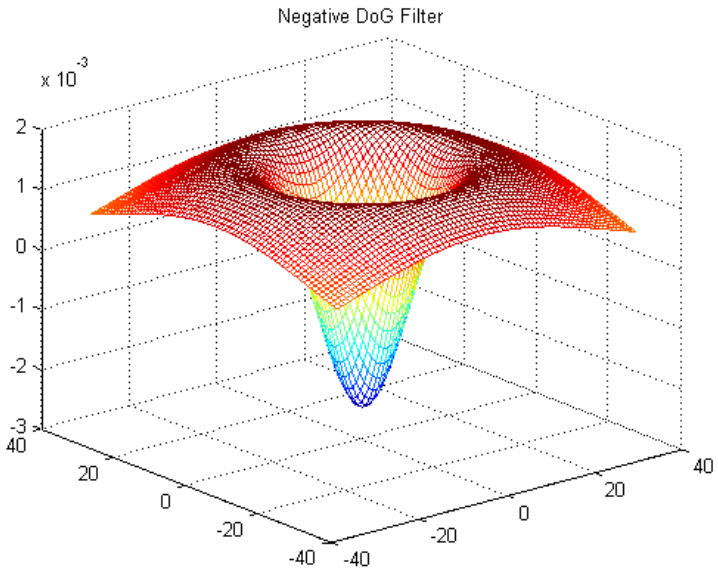

Figure 29: Negative DoG Function

Table 6: Negative DoG Filter Parameters

Size	σ_{ex}	σ_{inh}	c_{ex}	c_{inh}
3 * Im_{width}	0.416 * Im_{width}	1.25 * Im_{width}	3.5	1.5

of the filter are applied to the surrounding area. By applying the negative DoG filter to a previously attended region, the saliency of the previously attended region is inhibited while the saliency of the surrounding region is slightly excited. By simply applying the negative DoG filters to the saliency map for every previously attended region, the effects of proximity preference and habituation are implemented. The concept of dishabituation must now be addressed otherwise the saliency map will become filled with negative DoG filters as video is processed. If too many negative DoG filters are applied to the saliency map, the entire map will become inhibited, yielding little or no saliency information for future frames.

Dishabituation is implemented in the model by making the memory of previously attended regions a short term memory store only. Each selected region can only exist in this list of previously attended regions for a short period of time before it is no longer inhibited by the model. Once a region has lost the focus of attention, it is allowed to stay in memory for two seconds or 60 frames of video recorded at a rate of 30 frames per second. To incrementally bring each previously attended region back to it's "normal" saliency, each time the negative DoG filter is applied to the saliency map it is linearly scaled by a function of the elapsed time since it had been the focus of attention (Equation 13).

$$Equation13: \qquad Scale = 1.0 - (\frac{frames\_since\_attention}{2.0 * frames\_per\_second})$$

5.5.2 Salient Feature Feedback

The purpose of similarity preference is to make it more likely that a shift in attention will occur between two regions that share similar features. The model implements similarity preference by feeding back information about the salient features in the currently attended region so that the next frame being processed has some information about what was the last focus of attention. Four values are fed back to the model each frame, each value being the contribution of one of the component feature maps to the final saliency map for the attended region (Equation 14).

$$Equation14: FeatureFeedbackValue = \frac{\Sigma(FeatureMap)}{\Sigma(SaliencyMap) * 4.0}$$

Using the values representing the contribution of each feature to the final saliency of the previously selected region, the creation of the saliency map for the next frame is modified by weighting the contribution of each feature map to the final saliency map. Without feeding back salient features, the saliency map

is created by taking the average of the four component feature maps. Using the feed back values, each component feature map is increased up to a maximum of 25%, determined by the contribution of that feature to the previously attended region (Equation 15). By altering how much each feature map contributes to the final saliency map, features that are very prominent in a selected region one frame are made more salient in the next frame, making it more likely that a shift will occur between regions that share similar features.

$$Equation 15: \qquad Feature\_Map\_Scale\_Value = 1.0 + 0.25 * (feed\_back\_value)$$

5.6 Model Output

As a result of this model being primarily used as a pre-processor for a higher level recognition model, the output produced must be as generic as possible to accommodate a variety of recognition models while still being informative enough to be meaningful as a standalone model. The model produces two types of output: image output and textual output. Each frame of video processed by the model has a region that the model has determined to be the current focus of attention. The position, size, and frame numbers of the attended regions are the most important pieces of information to feed back to the user of the model.

The model produces three separate images for each frame processed. The first image produced is simply the the original frame of video before a region is selected (figure 30). The original frame of video is provided as an output of the model so that it is not necessary for an object recognition model to work with a video directly if the full frame is needed. The second type of image output produced is the original frame of video with the region that has the current focus of attention outlined in red (figure 30). The purpose of this output it to simply show at a glance which region has the focus of attention of the model in any given frame. The final image based output of the model is the extracted region which the model has it's attention currently focused on (figure 30). Each extracted region image will have different dimensions and can be used by an object recognition model to identify the content of each region.

In addition to producing image output, the model also produces textual output. The textual output of the model lists the position, size, and frame number of each region the model has focused it's attention on in a file called "BoundingBoxOutput.txt" (table 7). The (X,Y) values represent the upper-left corner of the region while W and H represent the width and height of the region. The textual based output of the model can be used to easily extract regions of attention from the original video if the image based output is unavailable or not useful "as is".

Figure 30: (A)Original frame of video before processing (B)Original frame with attended region outlined in red (C) Extracted region only

Table 7: BoundingBoxOutput.txt

Frame:	700
X:	267
Y:	228
W:	123
H:	123
Frame:	701
X:	300
Y:	400
W:	250
H:	250
...	...

6 Simulations and Results

The attention model described in the previous section was tested on a series of iteratively complex videos taken from a still camera. Each video segment lasts approximately three seconds and contains one or more target objects that are expected to be detected and extracted by the attention model. To accurately test the biological plausibility of the model, the video segments used should only be about one second in length because after that amount of time the brain begins to influence attention making the search no longer unconscious. However, the model is only capable of operating on and focusing on a region of interest one frame at a time. With the video taken at a frame rate of 30 frames per second, there are not sufficient frames to produce a complete search with only one second of video.

To improve processing speed on high resolution video, each input video is scaled to a width of 320 pixels and an appropriate height according to the aspect ratio the video was originally recorded in. By scaling the video, the overall speed of the system is increased while the saliency model remains intact. Regions are then extracted from the original high resolution video so that there are no scaling artifacts present if the output is reused by a recognition model.

To record the accuracy of the model, the simulated time taken to either partially or fully acquire each target object is recorded. Simulated time can be calculated by multiplying the number of frames elapsed by how much time each frame represents. All of the test videos were recorded at 30 frames per second, meaning that each frame represents 1/30 of a second. Target objects are defined subjectively since no eye tracking data is available for the test videos being used. In addition to simulating how well the model finds target objects, the attention model is also tested in conjunction with a modern object recognition system to see if adding attention improves the recognition task. Finally, the model is benchmarked for overall performance as well as individual component performance so that computational speed improvements can be made.

6.1 Single Target

The first series of simulations run on the attention model uses input video that contains a single target to be detected. The first video depicts a person walking from left to right across the camera's field of view. The person walking is the target object while the rest of the scene is considered clutter (figure 31-left). The second video used in the single target simulations is a video which uses a single road sign (figure 31-right) as the target. Each of the test videos were taken so that the scene, aside from the target, is relatively uninteresting in comparison to each of the target objects. It is hoped that each target object is found both quickly and optimally given that each scene contains only one target object along with very uninteresting clutter.

Figure 31: (Left) Video in which the moving person is the target (Right) Video in which the stationary road sign is the target

6.1.1 Results

Table 8: Single Target Simulation Results

	Partial Acquisition		**Optimal Acquisition**	
	Time Elapsed	Total Frames	Time Elapsed	Total Frames
Simulation 1	0.5 sec	26	1.8 sec	3
Simulation 2	-	-	0.3	6

As seen from the results shown in table 8, both simulations performed the task of finding the target object very well. Simulation 1, the person walking, was able to partially acquire the target object within 0.5 seconds, eventually finding the entire object within 1.8 seconds. Over the course of all 90 frames of the simulation, the target object was either partially or optimally acquired by the attention model for 29 of the 90 frames. The second simulation, the stationary road sign, faired somewhat differently than the first simulation. The sign was optimally acquired by the attention model with 0.3 seconds, however it was never partially acquired. The sign was only acquired by the attention model for 6 of the 90 frames simulated.

Based on the results from each of the first two simulations (figure 32), the model appears to be very promising. Because optimal acquisitions were found for each simulation it is likely that the model can be effectively used as an object extraction mechanism for a recognition system. Partial acquisitions of the target object did not occur during simulation 2, but this should not be seen as a problem since optimal acquisitions were found as well. Partial acquisitions only serve only to show that the model was close to finding an optimal solution. The actual extracted region from a partial acquisition cannot be reliably used by an object recognition system since it contains only a part of the target object.

Figure 32: (Left)Region selection from first simulation (Right) Region selection from second simulation

Figure 33: (Left) Video in which the 2 people are moving to the right (Right) Video in which 2 differently sized people are moving to the left. Scene also contains more complex clutter.

6.2 Double Target

The second series of simulations run on the attention model uses input video that simultaneously contains two different target objects. The first input video is a continuation of video used in the single target simulation showing a walking person. In this video the person continues to walk to the right, however, a second person enters the scene on the left (figure 33-left). The second simulation video is very similar to the first in that it has the goal of finding two different people as they walk across the screen. The second video, however, contains more clutter in the background to distract the attention model. Each of the targets to detect is also of different scale since one target is farther away than the other.

6.2.1 Results

The results from the first double target simulation were unexpected. The attention model was not able to fully acquire the first target (person farthest to

Table 9: Double Target Simulation Results

	Partial Acquisition		**Optimal Acquisition**	
	Time Elapsed	Total Frames	Time Elapsed	Total Frames
Simulation 1, Target 1	0.1 sec	18	-	-
Simulation 1, Target 2	N/A sec	N/A	1.6 sec	3
Simulation 2, Target 1	0.0 sec	28	0.76 sec	10
Simulation 2, Target 2	0.3 sec	12	2.6 sec	3

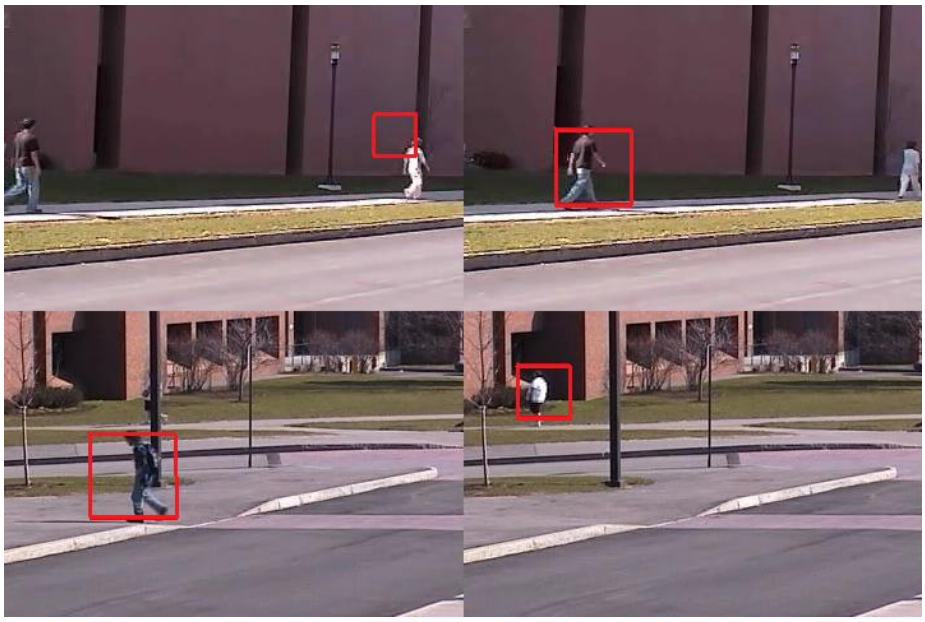

Figure 34: (Top) Region selections from first simulation (Bottom) Region selections from second simulation

the right), but it was able to fully acquire the second target (figure 34). While unable to fully acquire the first target, the model was able to partially acquire the target for 18 frames. Partial acquisition shows that the model was close to extracting the region, but determined the size of the region improperly. Partial acquisition of targets should not be seen as failures of the attention model, but rather as artifacts of not knowing anything about the targets that it is trying to focus attention on.

The second simulation performed much better than the first, being able to fully acquire both targets. Each target was partially acquired very quickly, within a fraction of a second for each target. Both targets were also partially or optimally acquired for 53 of the 90 frames which reveals that the attention model found the regions around both objects very salient which is what is expected.

6.3 Object Recognition Model Integration

The purpose of the attention model is not just to focus on interesting objects, but to focus on them so that a complex input is simplified for use by an object recognition model. By simplifying the input field, a recognition model's performance should increase when identifying target objects because there is less extraneous data in the input to negatively influence the recognition process. To test the influence the attention model can have on a modern recognition model, the extracted regions output from the attention model are used as input to a biologically inspired recognition model developed at the Rochester Institute of Technology by Theparit Peerasathein, Myung Woo, and Dr. Roger Gaborski.

6.3.1 Biologically Inspired Object Categorization Model

The goal of the biologically inspired object categorization model developed at the Rochester Institute of Technology is to be able to categorize objects presented to the model in the form of a color image [4]. The object categorization model is built on three different biologically inspired components (figure 35). The first component is a pre-processing component in which the color image is converted to an intensity image and then exposed to a battery of Gabor filters so that low level edge features are extracted from the image. After features have been extracted from the input, each feature map is passed through a four layer feature extraction neural network (FENN). The purpose of the FENN is to combine the low levels features extracted by the Gabor filters at each layer, slowly building invariance to scale and position of each feature. The final layer of the FENN is then processed by a neural network classifier that has been previously trained to categorize different objects.

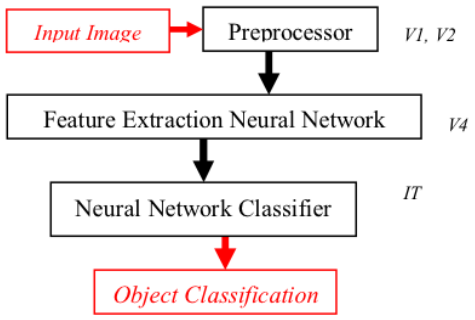

Figure 35: Architecture of the biologically inspired object categorization model. Each component represents a different area of the visual stream (Source: Gaborski07).

The categorization model has been trained to identify cat and dog faces as well as cars. The model has been experimentally shown to categorize novel cat and dog faces at 95% or better accuracy. However, the accuracy of the model degraded to around 71% accuracy when tested with car images. The drop in performance is partially attributed to the complexity of the urban scenes that the cars are present in (figure 36). The more complex backgrounds present in each car image have features extracted and propagated through the FENN which then cause errors in the neural network classification.

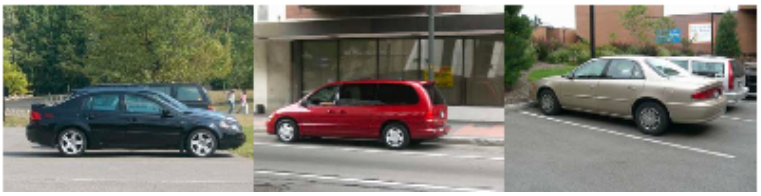

Figure 36: Images that were incorrectly classified by the object categorization model (Source: Gaborski07)

6.3.2 Region Extractions

Since the clutter in the car images cause problems to the classification model, using the attention model developed in this research seems a logical addition to the classification model. By using the attention model to extract cars from their cluttered backgrounds (figure 37), the model should have improved accuracy. To test this theory, the attention model was used on two different videos containing a number of cars driving through the cameras field of view. The cars which were extracted by the attention model, scaled to 128x128 pixels (a requirement of the categorization model), and then categorized. The full frame was also categorized so that the use of the attention model can be compared to the regular use of the classification model. It is expected that the results from the extracted regions will yield a higher categorization rate than the full frame of data.

6.3.3 Categorization Results

The results from the series of trials are shown in table 10. The neural network used in the categorization model produces a three decimal vector as output. The model categorization model was trained to output the vector [0 1 0] if a car is present in the input while [1 0 1] is output if no car is present.

As seen from the results, using the attention model to extract cars from their surrounding yields superior results than attempting to categorize the raw input. Trials 1, 2, and 3 yielded opposite results when using the extracted region

Figure 37: Sample images used to test the categorization model, extracted regions are shown in the lower left corner of each sample image.

Table 10: Car Categorization Results

	Low Resolution	
	Extracted Region	Full Frame
Trial 1	[0.005 0.995 0.005]	[0.833 0.161 0.855]
Trial 2	[0.085 0.932 0.077]	[0.675 0.300 0.681]
Trial 3	[0.022 0.978 0.024]	[0.620 0.385 0.641]
Trial 4	[0.018 0.980 0.019]	[0.058 0.959 0.049]
Trial 5	[0.042 0.954 0.047]	[0.340 0.646 0.371]

compared to the full frame. Each of the three trials produced output vectors very close to [0 1 0] when the extracted region was processed, meaning that the model correctly determined that a car was present in the extracted region. When the full frame was processed instead of the extracted region, the results were close to the vector [1 0 1], meaning that the categorization model actually believed that no car was present in the input. These results clearly show that raw input from a camera is not optimal for the categorization model and by using the attention model, the categorization task is greatly improved.

Trials 4 and 5 yielded somewhat different, yet meaningful, results. In both trials, the extracted region and full frame both produced outputs close to the vector [0 1 0]. With both the extracted input and full frame input yielding similar results, it appears that the categorization model did not necessarily improve with the attention model. It should be noted, however, that the vectors produced for the extracted regions are much closer to [0 1 0] than the full frame and thus a tighter threshold can be used on the neural network output vector. In addition to producing a better vector for thresholding, the attention model also provides the location of the car in the scene. The categorization model, by itself, is only capable of recognizing that a car exists somewhere in the input.

6.4 Attention Model Benchmarks

To test the overall computational performance of the model, the model was executed on two different computers processing the same test video. The amount of time taken to process each frame of the test video was recorded and averaged. In addition to benchmarking the overall performance of the model, the computation of each feature map as well as the attention shifting module were benchmarked.

Table 11: Attention Model Benchmarks

Category	Computer 1	Computer 2
Processor	2x 1.4Ghz	2x 2.2Ghz
Intensity Contrast	0.49 sec	0.16 sec
Orientation	4.75 sec	1.77 sec
Color Difference	0.88 sec	0.34 sec
Motion	34.10 sec	12.46 sec
Attention	8.08 sec	3.38 sec
Overall Time	48.3 sec	18.11

The results from the benchmark tests seen in table 11 clearly show that the model does not operate in real time. For the model to operate in real time each frame has to be processed in less than 1/30 of a second, not including the processing time required for the operation of an attached recognition model. When run on relatively modern computers the overall performance of the model

is between 18 and 48 seconds per frame. The benchmarks also reveal that the largest bottlenecks in the system are the creation of the motion and orientation feature maps as well as the attention model itself. These results make sense since each of the slower modules require a large number of convolution operations which are very processor intensive. A number of enhancements are discussed in section 7.1 which can be used to improve the processing bottlenecks in the model, making it possible for the model to operate in real-time.

7 Future Work

The focus of attention model developed in this research conducts an unconscious search though a video, attempting to determine a new region of interest for each subsequent frame of video. The model by itself does not necessarily produce any stand alone or meaningful results, however, when enhanced or integrated with other systems, this model can be utilized in a variety of situations ranging from navigation to video surveillance.

7.1 Real Time Processing

One of the major drawbacks and hurdles when working with this model is that it is not a real-time system. As seen in the results, even when running on a modern computer, the processing time required for a single frame of data is significant. In order to be useful in navigation or surveillance systems, the processing speed of the model needs to be increased by an order of magnitude. While the goal of real-time processing may not seem attainable given the current speed of the model, a number of enhancements can be made which will drastically increase processing time.

7.1.1 Biologically Inspired Filters

The goal when developing this model was to construct it using as many biologically plausible mechanisms as possible. One of the main drawbacks of using biologically plausible filters is that they only operate on small regions of an input at a time, requiring the use of a convolution operation. The convolution operation is very time consuming and processor intensive, meaning that great speed gains can be attained by removing the biologically plausible filters which require those computationally expensive operations. One of the largest processing bottlenecks in the model is the creation of the motion feature map. To create the motion feature map, a segment of video must be convolved with a blink filter which requires one convolution per "time slice" of the blink filter. The result from the blink filter convolution is a set frames which must then be convolved with four different motion sensitive spatio-temporal filters each requiring a convolution for each "time slice" in the motion sensitive filter. The actual number of convolutions required to calculate a single frame of motion saliency is a function of the temporal depth of each of the filters. However, even when using filters of small temporal depth the number of convolutions necessary to create a motion feature map quickly rises to an unreasonable amount. A possible solution to the motion processing bottleneck is to replace the biologically plausible filters with a faster, non-biological, approach such as frame differencing. By taking the difference between two frames of video and using

some clever thresholding, it is possible to detect motion from frame to frame with high accuracy at very little computational cost.

Another processing bottleneck for the model is the local competition that occurs in feature maps for the purpose of bringing them into the same dynamic range. Itti and Koch developed the local competition combination strategy because it was more biologically plausible than global amplification, which is also the reason why it was used in this research. As with other biologically plausible methods, local competition requires the convolution of each feature map with a wide DoG filter which takes a significant amount of processing time. A simple way to remove the need for the convolution operation is to use the global amplification combination strategy explored by Itti and Koch. As described in section 4.2, the global amplification strategy does not require a convolution operator since it operates on and is a function of the entire input. Since the global amplification and local competition combination strategies have been shown to yield similar experimental results, global amplification can be used instead of local competition without drastically effecting the results of the attention model. By changing feature combination strategies, the model will gain a speed increase at the cost of it's biological plausibility. It is possible that other significant speed gains can be attained elsewhere in the model by replacing more of the biologically inspired filters with faster, non-biologically plausible, alternatives.

7.1.2 Parallel Computing

One of the main premises of Anne Treisman's Feature Integration Theory is that individual features are computed in parallel by the brain and later integrated to help focus cognitive resources on a specific area of the visual field. Instead of computing the individual feature maps in parallel, the model developed for this research computes each feature map iteratively. Beginning with the contrast feature maps, followed by the orientation, color difference, and finally the motion feature maps, the model was developed to be run on a single computer processing each feature map one at a time. By utilizing multiple computers, the model can be made faster and more biologically plausible. Each feature or sub-feature map can be calculated at the same time, each on a different computer and later integrated into the final saliency map. Every computer added to calculate a feature or sub-feature map should increase the overall performance of the entire model significantly.

7.1.3 Programming Language Optimization

The model created and benchmarked for this research was developed using the MATLAB computing environment. While the MATLAB computing environment provides many useful tools to assist in the creation of computer vision mod-

els, programs written in MATLAB are interpreted and thus not very memory efficient and do not execute very quickly in comparison to other programming languages available. Significant speed gains can be attained without sacrificing the biological plausibility of the model and without adding more computing resources by converting the model to a faster, compiled, language such as C++. Recreating the model in a different programming environment will be somewhat complex since the toolkits in MATLAB are not readily available in other computing environments and will have to be developed.

7.2 Recognition Model Integration

As seen in section the simulations and results section, the task of integrating this attention model with a object recognition model is fairly trivial. Using the raw image output of the attention model as an input to a recognition model effectively integrates both models, as is done for the testing of this model. In addition to using the raw image output from one model as the input to another, it is also possible that a recognition model can use the textual output from the attention model to extract the attended regions directly from a video stream. While this level of integration is easy to achieve and will produce adequate results, both models are still two separate entities operating independently of each other.

Depending on the object recognition model used, it is possible to achieve a higher level of integration between both models. The most basic premise of the attention model developed in this research is that low level features are extracted from an input and later used to create a saliency map to focus attention. Feature extraction happens to also be the basis of many object recognition models, such as the model developed by Thomas Serre and his colleagues at the Massachusettes Institute of Technology [14] or the categorization model build at the Rochester Institute of Technology. Briefly looked at in section 2.4, the Serre Model of object recognition simulates the inferior temporal cortex (IT) by recognizing complex objects using a series of simple and complex cells to build up invariance to position and size. However, the Serre model uses a series of Gabor filters to extract orientation features from an input (figure 38) which is not regarded as a task accomplished by the inferior temporal cortex.

Band Σ	1	2	3	4	5	6	7	8
filt. sizes s	7 & 9	11 & 13	15 & 17	19 & 21	23 & 25	27 & 29	31 & 33	35 & 37
σ	2.8 & 3.6	4.5 & 5.4	6.3 & 7.3	8.2 & 9.2	10.2 & 11.3	12.3 & 13.4	14.6 & 15.8	17.0 & 18.2
λ	3.5 & 4.6	5.6 & 6.8	7.9 & 9.1	10.3 & 11.5	12.7 & 14.1	15.4 & 16.8	18.2 & 19.7	21.2 & 22.8
grid size N^{Σ}	8	10	12	14	16	18	20	22
orient. θ	$0; \frac{\pi}{4}; \frac{\pi}{2}; \frac{3\pi}{4}$							
patch sizes n_i	$4 \times 4; 8 \times 8; 12 \times 12; 16 \times 16$ ($\times 4$ orientations)							

Figure 38: Parameters for the creation of Gabor filters used to extract features in the Serre and Poggio model of object recognition (Source: Serre05)

Since both the attention model developed for this research and the Serre model of recognition both utilize Gabor filters to extract orientation features from an input, it is possible that this information can be shared between the two models. If the attention model is used to help narrow down the input for the recognition model, the orientation based features extracted for the creation of the saliency map can be used by the object recognition model as well. By extracting the orientation features only once for both models the overall speed of the combined models will be increased. In addition to increasing the overall speed, the combined models also become more biologically plausible. The combined models will now more closely follow the visual stream since the recognition model no longer has to reproduce the work of earlier stages of the vision task to accomplish the recognition task. It is also possible that the Serre model, or any other feature based recognition model, can benefit from integration with the attention model because additional low level features, such as color differences, contrast, and motion, become available to assist in object recognition.

7.3 Pre-Cueing

The most successful feature combination strategy explored by Itti and Koch was the learned weights combination strategy (figure 16), where manual training was used to alter the weights of the feature maps used to create the saliency map. A similar strategy was used in this research for simulating the similarity preference rule of attentional shifting by slightly scaling each feature map according to the prominence of each individual feature in the previously attended region. A similar method of scaling feature maps before they are used to create the final saliency map can be used to implement the psychological concept of pre-cueing. Pre-cueing is used often in psychological experiments for influencing a persons response to a stimulus before the actual stimulus is presented. An example of this is when a person is directed to look to the right because something is supposed to happen there, or telling a person that they should look for something that is red.

The attention model developed here does not use the concept of pre-cueing because the search conducted is meant to be completely unconscious, however adding the ability to pre-cue the attention model is very plausible. Pre-cueing the model to respond to a certain feature, like red, can be accomplished in the same way similarity preference is accomplished. Each feature map can be scaled according to certain pre-cueing information in addition to the similarity preference information fed back from the previous frame. Pre-cueing the model to respond to a certain region of the frame can be accomplished in a similar way to proximity preference. Instead of using a wide negative difference of Gaussian filter to inhibit a large region of the screen as is done for proximity preference, a simple positive Gaussian filter can be used to excite a region of the screen instead.

Allowing the attention model to be pre-cued to a certain region of the screen or certain feature is useful if the situation in which the model will be used is known in advance. As an example, if the model is going to be used in the video surveillance of the entrance to a parking garage and the object recognition system used is designed to recognize license plates, it would make sense to pre-cue the attention model to the approximate location of license plates so that they are found faster. Since the camera is in a fixed location, and most cars are the same height, it is possible to pre-cue the model to the approximate location of the license plates on each car arriving at the parking garage.

7.4 Object Tracking

As regions are extracted by the attention model and identified by an object recognition model, it is possible that a "target" object will be found. For example, if the purpose of the combined models it to track cars as they come in and out of the scene, the goal is to first find a car and after finding it, keep tracking it until it leaves the scene. At present, the attention model will find a region of interest and then attempt to "move on" to the next region without regard to whether or not the object found was of actual interest. Using techniques similar to pre-cueing and similarity preference, it is possible to implement object tracking within the attention model. Once a target object is found by the recognition model, two changes need to be made to the attention model's shifting mechanism to allow for object tracking.

The first change necessary is to suspend the habituation of regions once a target object has been found. By stopping the habituation of a region once it has been determined to contain a target object, the probability that a shift will occur is greatly reduced since the salience of that region will remain relatively unchanged from frame to frame. However, it is still possible that the next frame will yield an attentional shift due to a new object entering the scene, changes in scene lighting, or changes in motion of the target object. To lower the probability of a shift occurring even after habituation has been suspended, the amount that feature maps are scaled due to similarity preference can be increased to higher levels than would normally be used for an unconscious search through the scene. Since the purpose of the model is to track an object once it has been identified, the purpose of similarity preference changes from making it more likely to focus on a similar region to making it very likely that the exact same region is selected. Instead of slightly modifying the weights of each feature map according to the feature composition of the previously selected region, the exact proportions of each feature in the previously selected region can be used instead, making the next saliency map specifically tuned to the previously detected object. While attentional shifts may still occur, a basic object tracking system can be easily implemented within the confines of the current attention model.

8 Summary and Conclusions

In summary, this research has explored the earliest stages of the human visual task. Focusing on the ventral or "what" stream, a biologically inspired computational model was built to simulate the low level feature detection which occurs in in the retina, LGN, and primary visual cortex as well as the attentional modulation which has been shown to occur in cortical areas V2 and V4 (figure 39). By simulating the attentional modulation that occurs in the later stages of the visual task, a high resolution input video that would normally be overly large and complex for a recognition model to accurately process can be reduced to a much smaller size containing only the interesting objects to be identified.

Low level feature maps (color, orientation, motion, contrast) are extracted for a single frame of video using a series of biologically plausible filters which, when combined, form a saliency map. Using the saliency map as a guide, a region of interest is found for a given frame. This region of interest can then be focused on by an external recognition model. Once a region has been focused on by the attention model, the model implements the concepts of habituation, dishabituation, proximity preference, and similarity preference to make an attentional shift likely to occur during the next frame of video. The attentional shifts that occur during each frame of video produce an unconscious search of the interesting objects in the video which can be used in conjunction with a recognition model to identify each of the interesting objects found.

In conclusion, the model accomplishes the task of selecting interesting regions quite well when given a complex input as shown by the simulations and results discussed in section 6. While not always yielding the most optimal region selections, the model produces fairly accurate results considering that it is never explicitly trained to handle any specific input. At present, the one major drawback of the model is the amount of raw computational power needed to process a single frame of video. As discussed in section 7.1, the computational performance of the model can be drastically improved in a number of ways from reducing the biological plausibility of the model to redesigning the model to run in parallel on multiple computers.

In addition to accomplishing the goal of building a model which can be used to reduce the complexity of an input for a recognition model, the model also shows great potential to expand beyond what it was initially designed for. By using information fed back from a recognition model, the attention model can potentially act as a object tracking system or by using the concepts of precueing, the model can be trained to find specific objects. A significant amount of time and scientific research has been dedicated to the late, object recognition, stages of the visual task. This thesis revolves around the idea that the both the early and late stages of the visual task can be unified into a cohesive whole in order to build a more accurate vision model.

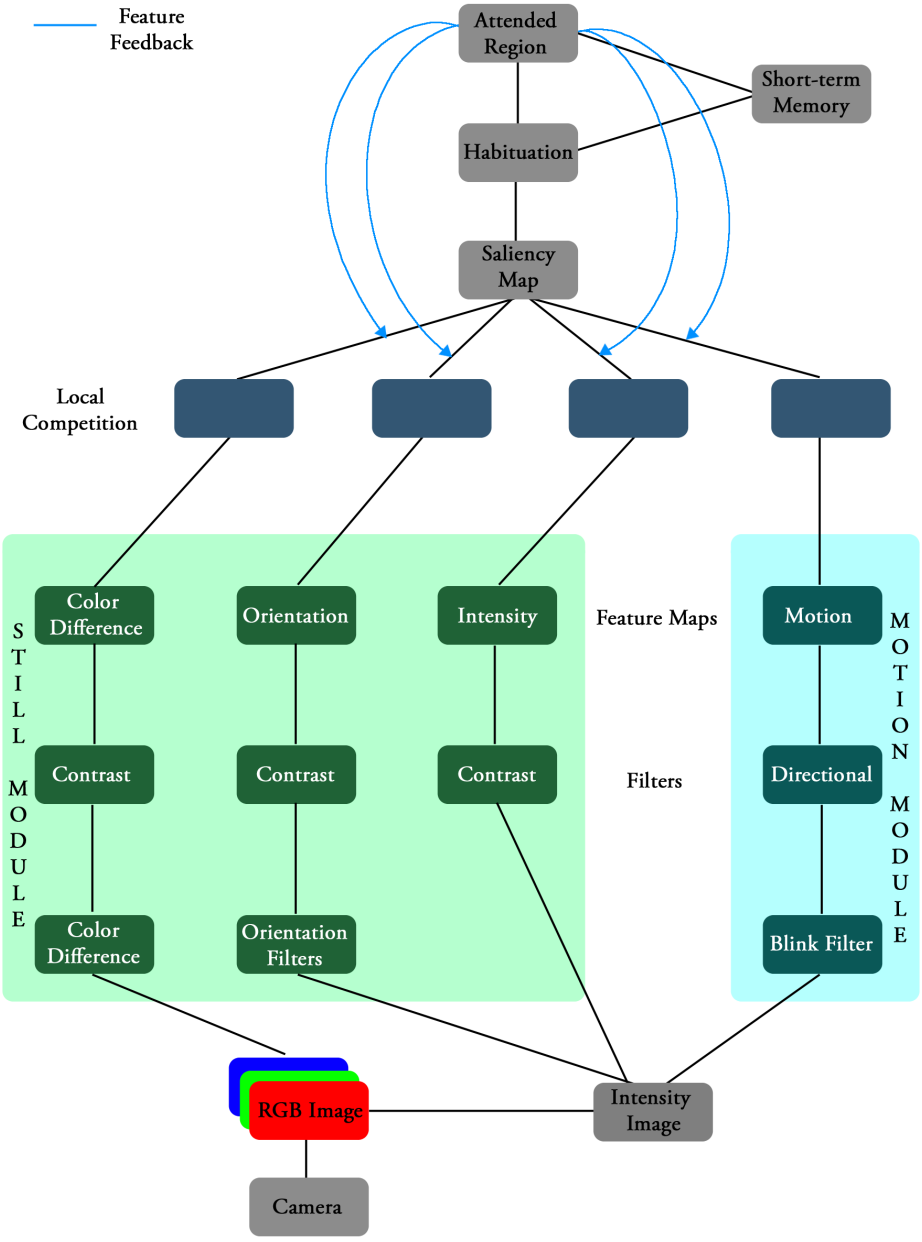

Figure 39: Diagram of the overall construction of the biologically inspired attention model built for this research

References

[1] Daugman, J. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters" Journal of the Optical Society of America Vol. 2 Num. 7 (1985)

[2] Enroth-Cugell, C. and Robson, J. "The Contrast Sensitivity of Retinal Ganglion Cells of the Cat" Journal of Physiology, V. 187 517-552 (1966)

[3] Gaborski, R. et al. "VENUS: A System for Novelty Detection in Video Streams with Learning" Department of Computer Science, Rochester Institute of Technology (2004)

[4] Gaborski, R. et al. "Biologically Inspired Object Categorization in Cluttered Scenes" Department of Computer Science, Rochester Institute of Technology (2007)

[5] Hubel, D. and Wiesel, T. "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex" Journal of Physiology (1962)

[6] Itti, L. and Koch, C. "Feature combination strategies for saliency-based visual attention systems" Journal of Electronic Imaging Vol. 10: 161-169 (2001)

[7] Itti, L. and Koch, C. "A saliency-based search mechanism for over and covert shifts of visual attention" Vision Research 40: 1489-1506 (2000)

[8] Koch, C. and Ullman, S. "Shift in selective visual attention: towards the underlying neural circuitry" Human Neurobiology (1985)

[9] Mishkin, M. and Ungerleider, LG. "Contribution of striate inputs to the visuospatial function of parieto-preoccipital cortex in monkeys" Behavioral Brain Research (1982)

[10] Moran, J. and Desimone, R. "Selective attention gates visual processing in the extrastriate cortex" Science (1985)

[11] Mullen, K. "The contrast sensitivity of human color vision to red-green and blue-yellow chromatic gratings" Journal of Physiology Vol. 359: 381-400 (1984)

[12] Rolls, E. and Deco, G. "Computational Neuroscience of Vision" Oxford University Press (2002)

[13] Rolls, E. and Stringer, S. "Invariant object recognition with trace learning and multiple stimuli present during training" Department of Experimental Psychology, Oxford University (2007)

[14] Serre, T. and Poggio, T. "Object Recognition with Features Inspired by Visual Cortext" Massachusettes Institute of Technology (2005)

[15] Treisman, A. "Perceptual Grouping and Attention in Visual Search for Features and for Objects" Journal of Psychology: Human Perception and Performance (1982)

[16] Treisman, A. and Gelade, G. "A feature-integration theory of attention" Cognitive Psychology (1980)

[17] Shaw, M. and Shaw. P "Optimal allocation of cognitive resources to spatial locations" Journal of Experimental Psychology: Human Perception and Performance 4: 586-598 (1977)

[18] Qiu, F. and Heydt, R. "Figure and Ground in the Visual Cortex: V2 Combines Stereoscopic Cues with Gestalt Rules" Neuron, Vol. 47 155-166 (2005)

[19] Young, R. and Lesperance, R. "The Gaussian Derivative model for spatial-temporal vision: I. Cortical model" Spatial Vision, Vol. 14, No. 3,4: 261-319 (2001)

[20] Young, R. and Lesperance, R. "The Gaussian Derivative model for spatial-temporal vision: II. Cortical data" Spatial Vision, Vol 14, No 3, 4: 321-389 (2001)