NOTE TO USERS

uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Kristopher Woodbeck
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

On Neural Processing in the Ventral and Dorsal Visual Pathways
Using the Programmable Graphics Processing Unit

TITRE DE LA THÈSE / TITLE OF THESIS

Eric Dubois
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Gerhard Roth
CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Yongyi Mao

Andrew Adler

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

On Neural Processing in the Ventral and Dorsal Visual Pathways Using the Programmable Graphics Processing Unit

by

Kris Woodbeck

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the MCS degree in
Computer Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-46508-0
Our file Notre référence
ISBN: 978-0-494-46508-0

Abstract

We describe a system of biological inspiration that represents both pathways of the primate visual cortex. Our model is applied to multi-class object recognition and the creation of disparity maps from stereo images. All processing is done using the programmable graphics processor; we show that the Graphics Processing Unit (GPU) is a very natural platform for modeling the highly parallel nature of the brain.

Each visual processing area in our model is closely based on the properties of the associated area within the brain. Our model covers areas V1 and V2, area V3 of the dorsal pathway and V4 of the ventral pathway of the primate visual cortex. Our model is able to programmatically tune its parameters to select the optimal cells with which to process any visual field. We define a biological feature descriptor that is appropriate for both multi-class object recognition and stereo disparity. We demonstrate that this feature descriptor is also able to match well under changes to rotation, scale and object pose.

Our model is tested on the Caltech 101 object dataset and the Middlebury stereo dataset, performing well in both cases. We show that a significant speedup is achieved by using the GPU for all neural computation. Our results strengthen the case for using both the GPU and biologically-motivated techniques in computer vision.

Acknowledgements

I would like to thank my advisors, Dr. Gerhard Roth and Dr. Eric Dubois, for giving me the freedom to explore an area of research to which I've always been drawn. Many would have been very skeptical when I started down this path, but their support gave me the opportunity to explore this intriguing topic.

Dr. Roth has always been incredibly supportive and ready with a wealth of advice about how best to proceed with my ideas. His eagerness to share and discuss anything and everything at all has made this period very enjoyable. Thank you.

Dr. Dubois has been very accessible for all administrative and funding-related aspects of my studies. His feedback has helped to guide me in many important ways.

I would like to thank the other students and professors in the NAVIRE project for their support and feedback.

I would also like to thank my family for listening to my seemingly incessant chatter about the inner workings of the brain and for supporting me in my decision to return to school. I would also like to thank my friends for making sure that I didn't forget to enjoy life to the fullest along with them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The human brain has a remarkable visual processing system known as the visual cortex. Due to the visual cortex's incredible dexterity and highly complex structure, computer vision systems have met with great difficulty in attempting to match its visual processing capabilities. The visual cortex outperforms computer vision systems by a wide margin at common tasks such as object recognition, object tracking and motion based processing. This has been compounded by the fact that common microprocessors do not provide an effective platform for modeling a system as highly parallel as the visual cortex. Recent advances in the Graphics Processing Unit (GPU), namely the development of the programmable graphics processor, have provided a parallel platform that is much more suitable for modeling the visual cortex. Neuroscience provides key insights into the inner workings and structure of the visual cortex and we leverage this biological knowledge to build a robust, biologically-inspired computer vision system using the programmable graphics processor.

1.1 Motivation

Computer vision systems have become quite adept at recognizing objects that they have already seen. Methods such as SIFT [56] have shown excellent results in recognizing objects whose orientation and lighting conditions do not change drastically. Many systems make significant efforts to maintain recognition capabilities that are invariant to changes in scale and rotation, but the larger problem of how the brain is able to perform so robustly has not yet been fully explored.

There is a significant body of neurobiological experiments and data related to the visual

cortex. To better equip a system for object recognition, it stands to reason that this data must be taken into account when designing computer vision systems. Biologically motivated systems, such as that by Serre et al. [40], have shown a significant step forward in both our understanding of the visual processes of the brain and its relation to object recognition systems. Our work aims to remove some of the limitations present in current biological models and to extend the biological model scope to cover a larger range of visual problems handled by the brain.

Our ultimate goal is to create a biologically based image feature descriptor that is suitable for indexing and querying a large number of objects in a geometrically, lighting and pose invariant manner. In this work, we describe feature descriptors that have a number of invariant properties and that are excellent for multi-class object recognition and stereo disparity problems.

1.2 Scope of the Thesis

This work serves to accurately model the biological processing done in the first levels of the visual cortex. We derive feature descriptors that perform well in multi-class object recognition on the Caltech 101 dataset [72]. The descriptors are also applied to the problem of disparity maps for stereo vision using the Middlebury dataset [75]. These problems were chosen because they represent functionality from both main pathways of the visual cortex. For a biological feature descriptor to be plausible, it must be applicable to all problems handled by the visual cortex. Our results are achieved by exploiting the parallel nature of the programmable graphics processor.

The tasks at hand are multi-class object recognition and the creation of disparity maps from stereo images. All images used in this work are converted to grayscale, due to the biological evidence supporting the fact that color perception occurs at much higher areas of the brain. Figure 1.1 shows a sample of images from the Caltech 101 dataset that is widely used to test object recognition systems. We also create disparity maps to model the process of integrating data from the left and right eyes.

Multi-class object recognition involves assigning a label to an image based on a limited number of training examples. The dataset used contains unsegmented images with objects generally aligned in a similar pose. Dealing with noise is crucial in this task, since many of the objects are shown in their natural surroundings. Some vision systems focus on segmented

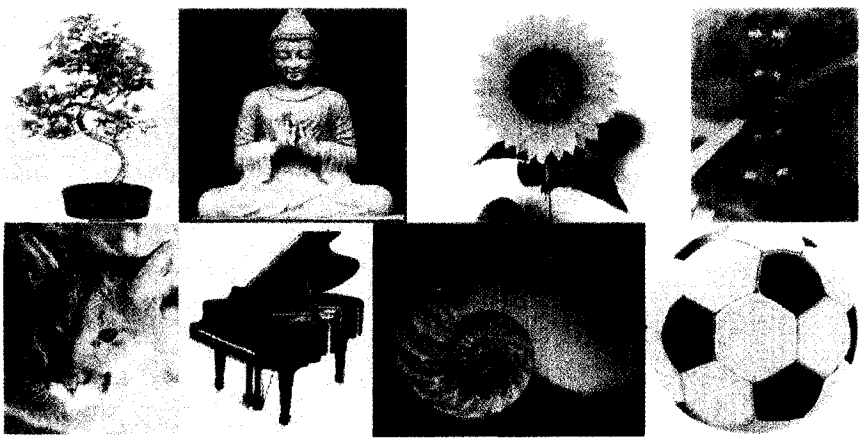


Figure 1.1: Sample of image from various categories

images, where the object in question has a bounding box showing its location. We focus on unsegmented images because they are analogous to processing a full visual field by the visual cortex itself. Object recognition is the primary responsibility of the ventral pathway of the human visual cortex.

We also apply our model to the problem of stereo vision where two images of an identical scene are taken from different positions and are used to compute the relative depth of each object within the scene. This task is analogous to how the human brain uses its left and right eyes to estimate the distance of a given object in the visual field. Disparity is one of the tasks handled by the dorsal pathway of the human visual cortex.

Our results from both problems strengthen the case for using both the graphics processor and biologically-motivated techniques in computer vision.

1.3 Contributions

The model presented in this thesis is a novel implementation which is similar in a number of ways to the Standard Model of the ventral stream of the primate visual cortex [39], which is itself an extension of the HMAX model of the primate visual cortex [35]. Our model has been expanded to support both pathways of the visual cortex and performs all visual processing entirely on the graphics processor. We have incorporated a number of other biologically-inspired properties, such as the definition of a feedback processing model, a parameter tuning model, illusory contour exploration, lateral inhibition and a feature selection mechanism using neural

activation.

We test these modifications by exploring feature invariance require for object recognition and give a method of selecting the best features to represent an entire class of objects. We perform multi-class object recognition on the Caltech dataset and stereo disparity on the Middlebury dataset. Our final system performs well in both tasks. We show that using the graphics processor demonstrates a considerable speedup over the common microprocessor.

1.4 Thesis Organization

This work describes a biologically-inspired model that processes the visual field in a manner similar to the initial areas of the visual cortex. This processing is accomplished with a series of functionally distinct visual processing modules that serve to model the corresponding areas of the visual cortex. Each area's implementation is designed to mimic the true neural properties of the corresponding area in the brain and is implemented entirely on the graphics processor.

The thesis is laid out as follows:

- Chapter 2 gives the a history of image feature descriptors, a history of work in modeling the visual cortex and describes the structure and biological properties of the visual cortex. It also gives a brief overview of our model and describes its use of programmable graphics processors.
- Chapter 3 describes the first area of the visual cortex and discusses the model's cellular tuning process which is designed to normalize the visual field prior to computing feature descriptors.
- Chapter 4 describes the second area of the visual cortex and describes the image feature descriptor computation process. It also provides an analysis of our feature descriptors with respect to scale, rotation and higher order invariance.
- Chapter 5 describes object recognition experiments, disparity experiments and examines the computational speedup achieved with the use of the GPU.
- Chapter 6 is the conclusion.

Chapter 2

Background and Literature Review

The most common way of recognizing objects relies on isolating the key salient regions within an image and using these regions to perfom classification. Salient regions are generally defined with a feature descriptor that uses key textural and geometric information to uniquely describe the region in question. Feature descriptors are generally for object recognition and matching similar regions within two images. Successful object recognition relies on the definition and extraction of feature descriptors with a number of invariant properties. Rotation, scale, changes to object pose, scene lighting, clutter and occlusion must all be taken into account. Generalizing feature descriptors so that they can be used to recognize any instance of a given object requires a high degree of robustness.

This chapter gives background on various types of feature descriptors used for object recognition, both of biological and non-biological inspiration. The overall structure of the visual cortex and the responsibilities of key areas within the visual cortex are reviewed. This information is given within the context of disparity estimation and object recognition. An overview of our mapping of this visual cortex model using the Graphics Processing Unit (GPU) is also given.

2.1 Image Feature Descriptors

Image feature descriptors are used to describe and match salient regions within an image. There has been a significant amount of research allocated towards developing non-biological feature descriptors, but computer vision has placed much less focus on how the problem is solved by one of the most robust object recognition systems known: the human visual cortex. This lack

of focus is due to a number of difficulties that must first be overcome. The main problem has been a dearth in the details of the structure and specific visual processing done in the human visual cortex. Compounding this problem is a shortage of techniques for dealing with the high computational requirements inherent in biological systems. Modeling the visual cortex is a computationally expensive task that common Single Instruction Single Data (SISD) microprocessors are ill-equipped to handle. Non biological feature descriptors are generally used in computer vision because they are much less computationally expensive than their biological counterparts.

2.1.1 Non Biological Feature Descriptors

There is a plethora of feature descriptors available to extract the salient regions of an image, most of which are not biologically based. Corner detectors, edge detectors and histogram based methods are among the most popular. Harris corners [58] use a sum of squared distances between image patches to isolate edge and corner regions. A number of related techniques have been derived from this algorithm, such as using multi-scale Harris corners in conjunction with a Gaussian image pyramid [59]. Another technique, called the Smallest Univalue Segment Assimilating Nucleus (SUSAN) operator, uses a mask centered around a nucleus to detect changes in brightness, corresponding to edge and corner regions [60].

Scale-Invariant Feature Transform (SIFT) is a popular blob detection algorithm that uses histograms over an image with a progressive Gaussian blur applied [56]. SIFT is widely used for a number of computer vision problems and performs well at recognizing objects that it has already seen, but SIFT alone does not perform well in generic object recognition tasks. It has been used in conjunction with a multi-resolution histogram feature clustering method and gives much better object recognition results than SIFT alone can achieve [81].

While histograms are popular due to their ease of computation, they are highly variant to changes in both lighting conditions and object pose. There is very little biological evidence to support the use of histograms as a primary factor in object recognition; the fact that color blindness does not impair object recognition in humans seems to contradict using a histogram-based approach for object recognition. While histograms are an appealing representation from a statistical viewpoint, one of the many problems they face is that techniques using histograms must deal with high dimensional feature descriptors. Dimensionality reduction is often used to deal with this problem.

2.1.1.1 Dimensionality Reduction

Algorithms with dimensional reduction properties are often used to improve the classification results of various image feature descriptors. These algorithms include Principal Component Analysis (PCA) [61], Multidimensional Scaling (MDS) [62], Linear Discriminant Analysis (LDA) [63] and others. SIFT feature descriptors initially have 128 dimensions; PCA is commonly applied in systems using SIFT keypoints and can lower descriptor dimensionality from 128 down to 36 or less [57]. Techniques involving dimensionality reduction are not used in our model due to the a lack of biological evidence.

In terms of object recognition, dimensionality reduction can have a number of negative side effects. When an object class with a small number of samples exists, methods such as PCA have the effect of minimizing the key class. This property is very unlike the brain: it is common that the class with a small number of instances is the most unique and recognizable class. For example, assume a user is attempting to search a database containing thousands of images of modern automobiles for a rare sports car. A unique entry such as this has a very limited set of instances. Methods such as PCA serve the purpose of maximizing the classifier's accuracy at the expense of rarer instances. This results in a lower performance for the key class.

In contrast, the human brain takes a completely different approach: a rare sports car is certainly the most easily recognized. Of course, this limitation in dimensional reduction systems can be partly overcome by assigning higher weights to the training samples. But one must realize that this is not always effective and is simply a manner of covering up the inherent problems in dimensionality reduction itself. The ideal object recognition system must be capable of effectively and implicitly handling high dimensionality. No dimensionality reduction techniques are used in this work due to the lack of biological evidence, as well as this dissimilarity between the properties of reduction algorithms and those of the brain.

2.1.2 Biological Feature Descriptors

Biological models of the visual cortex are generally less focused on the problem of formalizing a useful image feature descriptor; instead they focus on the overall problem of object recognition. Most models chose a prototype based approach where salient regions in an image are matched against a crafted database of feature descriptor prototypes. It is noteworthy that the advancement of visual cortex models corresponds directly with advancements in the technology used to monitor the brain, such as functional Magnetic Resonance Imaging (fMRI) and

other techniques.

2.1.2.1 The Neocognitron Model

The Neocognitron [33] is a biologically motivated multilayer neural network for handwritten character recognition. Its structure is shown in Figure 2.1. It models the arrangement of simple and complex cells [30] in the visual cortex by extracting local image features at lower stages. These are gradually merged into more complex features at higher stages. It has both supervised and unsupervised modes and is partially invariant to shift, rotation and other types of distortion. The Neocognitron was the first form of biologically motivated convolution network to use a series of convolutions to the input image and process the results in a hierarchical fashion.

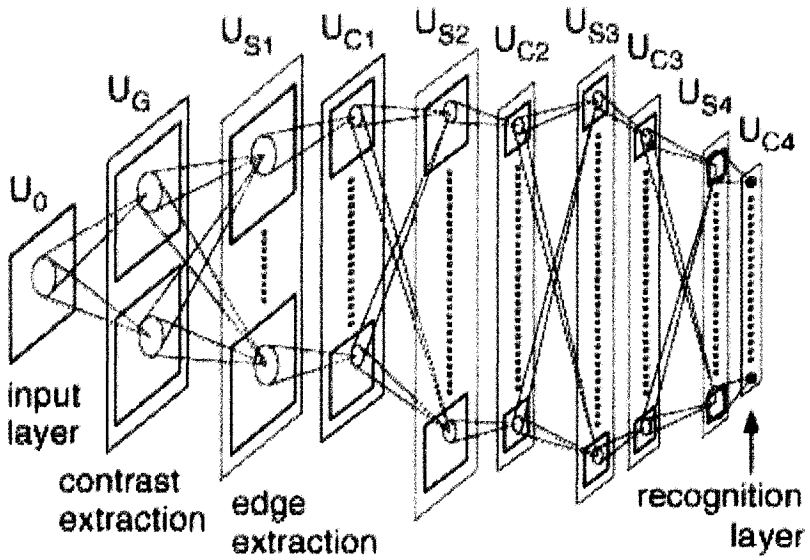


Figure 2.1: A typical architecture of the Neocognitron. Each layer is designated as contrast, simple, or complex cell layer. Taken from [34].

2.1.2.2 The HMAX Model / Standard Model

The HMAX model of the primate visual cortex [35] is a form of multi-layer convolution network, shown in Figure 2.2. This work explores the invariant properties of neurons in response to visual input. It is based on experiments with the macaque visual cortex where monkeys were trained on a restricted set of paperclip-like stimuli. When new views of the stimulus were presented, individual neurons were shown to activate in response to these novel views [36]. This supports the hypothesis that a small number of individual neurons are gradually tuned to

a geometrically invariant version of a given stimulus [38, 36].

HMAX is designed as a feedforward model of the visual cortex. It models a number of fairly well agreed upon facts about the visual cortex, namely:

- A hierarchical buildup of scale and position invariance, eventually leading to viewpoint invariance;
- A feedforward processing model;
- A gradual increase of cellular receptive field size;
- An increasing complexity of optical stimuli;
- Plasticity and learning at all stages in the visual cortex.

The complex cells of HMAX differ from previous models in that pooling occurs with a MAX operation instead of using a weighted sum. This is supported by physiological evidence from some cells of the visual cortex [31].

2.1.2.3 Standard Model 2.0

The HMAX model was further extended in the Standard Model 2.0 [39], shown in Figure 2.3. Like the HMAX model, it uses a feedforward-only model and is designed to model "rapid recognition" tasks where there is no opportunity for changes in attention or focus. It also makes attempts to model the neuroplasticity demonstrated at higher levels of the brain. The model is consistent with many of the cellular properties in the ventral stream of the visual cortex. It is limited by the use of static filterbanks, a static dictionary of hand-crafted features and an implementation using only SISD computational algorithms. It is of note that this model was able to significantly outperform SIFT features in generic object recognition tasks [41].

This model was extended and optimized by Mutch et al.[42]. Mutch outlined a method of improving classification results by optimizing the feature database. This is done by dropping features from the prototype database with low SVM weights. With these optimizations and feature pruning, the model was able to achieve even higher classification results.

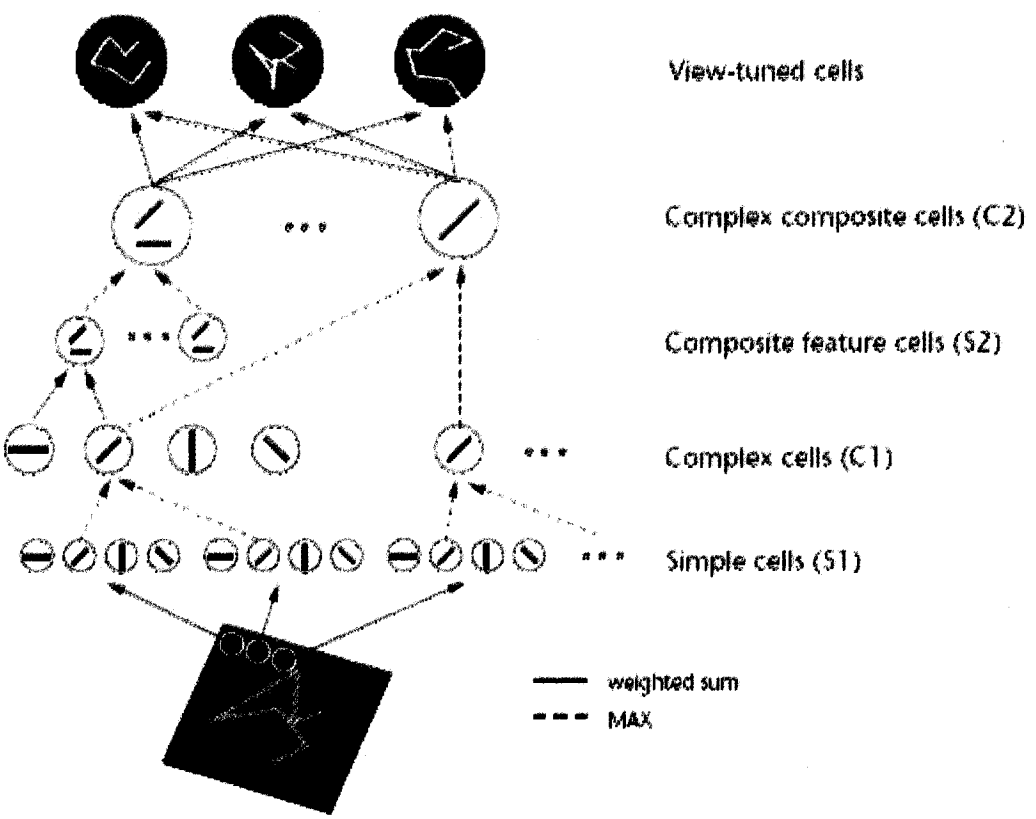


Figure 2.2: Overview of HMAX Model. Taken from [35].

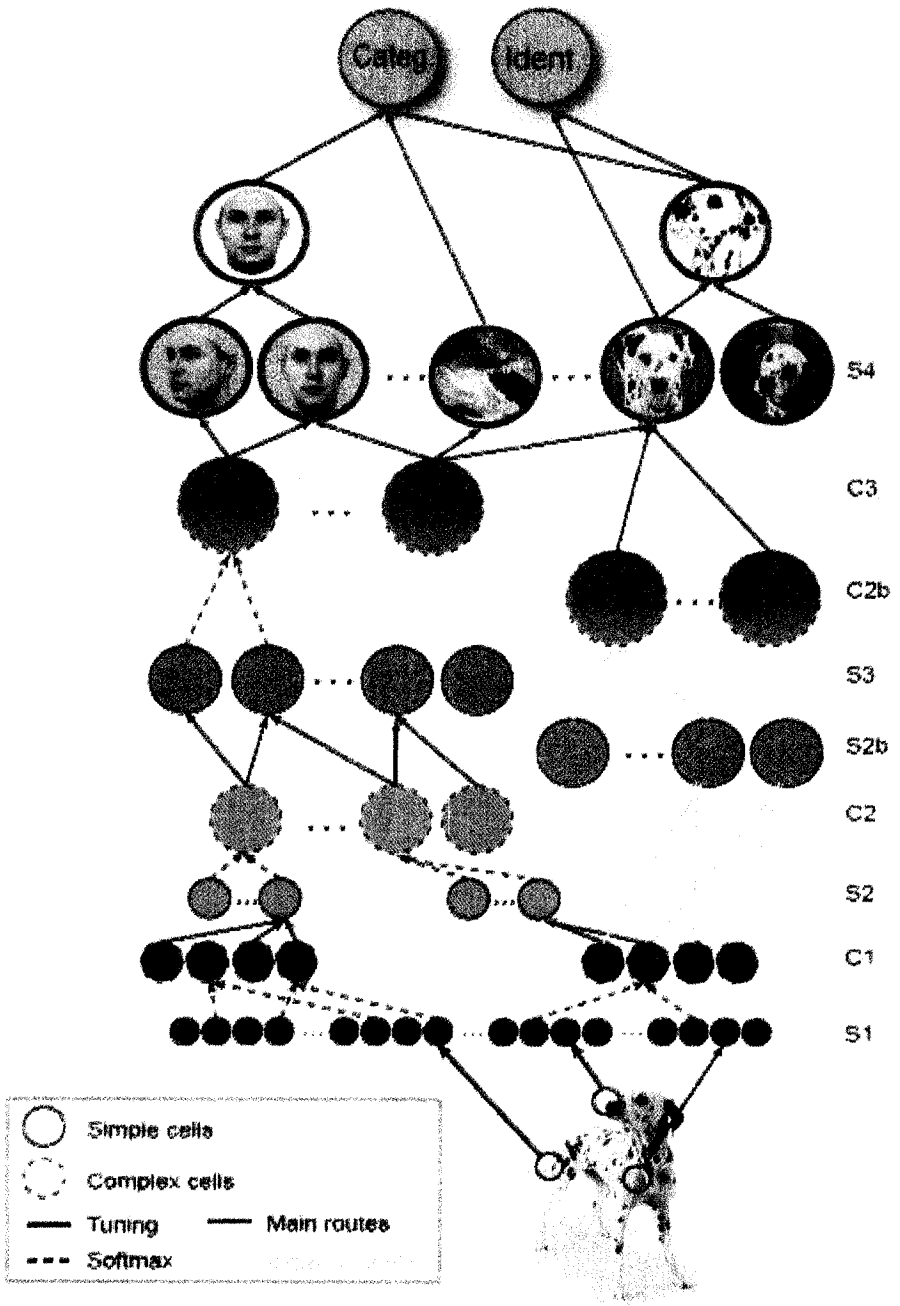


Figure 2.3: Overview of the Standard Model 2.0. Taken from [39].

2.1.2.4 The LISSOM Model

The Laterally Interconnected Self-Organizing Map (LISSOM) model [68] aims to replicate the development of the human visual cortex. It is based on the Self-Organizing Maps (SOM) architecture [66] and uses a growing network structure, sequential inputs and lateral connections to capture correlations between internal neural activations and visual input. The LISSOM model uses feedforward and lateral connections between neurons and performs its learning process with very general learning rules. It was originally designed to model only the first area of the visual cortex and did not lend itself well to working with natural image stimuli; many of these problems were addressed by later models, such as RF-LISSOM [69].

Visual cortex models all share a number of key properties. Simple and complex cells form the basis of all models, as well as a gradual increase in the size of these cellular receptive fields. A feedforward model is the most common approach. This is combined with a general increase in complexity of optical stimuli as the visual field progresses through a model. All of these properties are designed to work together in a manner that attempts to maximize neural plasticity.

2.2 Visual Processing in the Brain

The human visual cortex is a series of functionally distinct visual processing areas. Visual processing in the human brain begins at the eyes: they separate the current scene into a left and right visual field. The visual field traverses the optic chiasm, seen in Figure 2.4. In the optic chiasm, the visual field passes the Lateral Geniculate Nucleus (LGN), whose purpose is further explained in Section 2.3.1. The visual data continues on to the occipital lobe at the rear of the brain, where higher order visual processing begins at area V1. Areas V1 and above are formally known as the visual cortex. An overview of the human visual cortex is given by Grill-Spector et al. [21].

2.2.1 The Visual Cortex

The visual cortex is divided into unique, functionally distinct areas that are each responsible for very specific visual processing tasks. The cells in the lowest areas of the cortex are generally tuned to a single, preferred eye. The visual field is processed in a retinotopic fashion [29], meaning that adjacent areas in the visual field are processed by adjacent cells in the visual

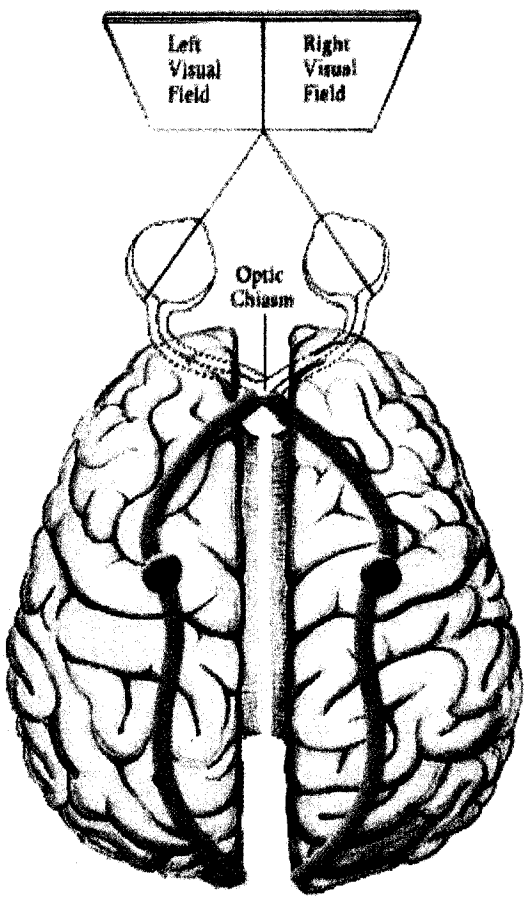


Figure 2.4: The human brain. The optic chiasm is shown in green. Areas of contact with the Lateral Geniculate Nuclei (LGN) shown in red. Modified from [3].

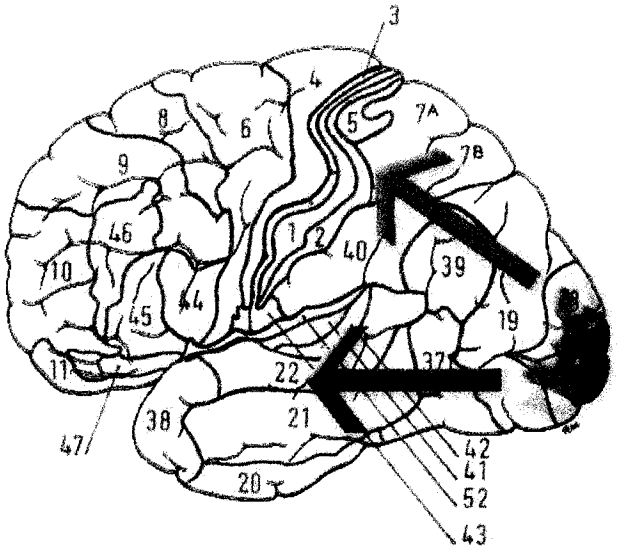


Figure 2.5: The human visual cortex. Approximate area of V1 shown in red, V2 in orange, V3 in yellow. The dorsal stream is shown in green, the ventral stream in blue. Modified from [3]; V-regions extrapolated from [43].

cortex. Each cell has a receptive field that corresponds to a particular region of the visual field; adjacent cells have overlapping receptive fields. The retinotopic visual field can be thought of as an array of neurons where each neuron is responsible for a subset of the visual field. This array of cells forms the visual field in its entirety. At the lowest levels, each area of the visual cortex has a complete map of the visual field. As the visual field is passed to the higher areas of the brain, it is modified by each successive area to take on a higher order of complexity, such as a complete geometrically projected visual field in the dorsal stream. The receptive fields of cells in the visual cortex change in a predictable manner and as a function of anatomical location [29]. One such change is a gradual increase in receptive field size in higher cortex areas.

As seen in Figure 2.5, the visual cortex has two common areas: V1 and V2. These are responsible for all shared, low level visual processing required for motion and recognition. Past V2, the visual cortex splits into two parallel processing pathways: a recognition (dorsal) pathway and a motion (ventral) pathway, commonly known as the "what pathway" and the "where pathway". These pathways are further explained in Section 2.3.4. Our work aims to biologically model all processing up to area V2, creating a biological feature descriptor. This feature descriptor is used for object recognition as done in the ventral pathway, similar to other

work in this field. The dorsal pathway is also explored in this model by computing disparity maps between the left and right eye. Disparity processing is done at an early stage in the dorsal pathway [47].

The human visual system is divided into key visual areas responsible for very specific tasks. For example, there are specific areas that are responsible for luminance based calculations; other areas are responsible for specific tasks such as invariant facial recognition. There is still a considerable gap in our comprehension of neural connectivity and communication protocols, but functional Magnetic Resonance Imaging (fMRI) provides an excellent tool to expand our understanding. While a single neuron or cell is not easily observed without neurosurgery, the activation of specific areas in response to crafted visual inputs is often measured via fMRI [12, 13, 25, 32, 31, 38, 48].

2.2.1.1 Color Processing

It is widely assumed that the human visual cortex does not perform significant color processing at its lowest levels. This is demonstrated by the cerebral achromatopsia disorder, where color processing is lost entirely due to damage at a high level in the visual cortex [2]. When a lesion is obtained at area V8 of the visual cortex, deep within the ventral stream, the subject can lose all color vision. This shows that the key area for color perception in the visual cortex is not crucial to most of its operation and that there is significant shape-based visual processing done prior to any color perception. Other forms of partial color blindness are relatively common, with approximately 7% of males in the United States having some form of color blindness [70]. There is no evidence to suggest that colorblindness hinders the subject's abilities in terms of object recognition; it is therefore necessary to question the use of color data techniques, such as color histograms, for object recognition in computer vision systems. Other visual cortex models also process only grayscale information at the lowest cortex levels [39, 35].

While most visual cortex models use grayscale images, it is known that cells at the V1 and V2 levels are responsible for some basic color processing [4, 2]. Since all higher complexity object and pattern recognition is done at later areas in the visual cortex, these cells are likely part of data routing within the brain, as indicated by Chatterjee [71]. A mechanism such as this may allow certain light frequency ranges to be processed by more primitive areas of the visual cortex, with help from the LGN. Of course, modeling this type of data routing is very species dependent and it would likely only be useful in a real-time system.

2.2.2 Simple and Complex Cells

Hubel and Wiesel found that processing in the primary areas of the visual cortex is divided into alternating groups of simple and complex cells [30]. There are several forms of simple and complex cells at various areas in the visual cortex. A simple cell is responsible for a specific computational task, the details of which depend largely on which area of the brain that cell is located. As an example, Gabor functions [1] provide an excellent model of V1 simple cells for many species of mammals, from cats to modern primates [20, 27]. A description and interactive model of simple cells can be found at [79]. Groups of simple cells are connected to complex cells. The task of complex cells is to selectively activate based on the responses within their group of simple cells. Complex cells are effectively responsible for operations that pool the output from simple cells.

All biological models use some form of simple and complex cell mapping, generally arranged into a feedforward model [39, 35, 42, 33]. Our model is arranged with feedforward connections, where each cell type is responsible for operations of varying complexity. All simple / complex cell pairs follow a computation / estimation process. The visual field produced by complex cells is the primary input used by higher areas of the model. This visual field is also the basis for feature descriptors used in our model.

Our model is separated into the biologically distinct V1 and V2 areas. Each area has both simple and complex cells whose properties are closely related to those of the corresponding cells in the primate visual cortex. While the first two areas of the cortex are the primary focus of this work, a significant effort has been made for elegant extension into higher areas of the brain. Keypoints from the lower levels are meant for extension into geometric-invariant keypoints, color sensitive cells and texture sensitive cells. Much of this work is focused on the end result of building a database of geometrically invariant feature descriptors.

2.3 Areas of the Visual Cortex

The lowest areas of the visual cortex, areas V1 and V2, are active for all forms of visual processing [21]. These areas are not directly responsible for complex tasks such as object recognition. Their output is instead used by higher areas that in turn perform more complex processing. Lower areas of the brain are responsible for processing that is common to both object recognition and motion-based tasks. Instant recognition, where no chance is given to

focus visual attention, is best described with a feedforward model [40].

The visual field begins at the eyes and then travels through the optic chiasm, shown in Figure 2.4 to area V1, shown in Figure 2.5. The visual field progresses through area V1, followed by area V2. Area V2 is the last shared area before the visual cortex splits into two distinct pathways, known as the ventral and dorsal pathways. Consequently, the feature descriptor output from V2 is designed to be compatible with the processing done in both visual pathways.

2.3.1 Lateral Geniculate Nucleus

The role of the Lateral Geniculate Nucleus (LGN) in visual processing is not well understood. As seen in Figure 2.4, it is located in a very distinct area from the rest of the visual cortex. Its main responsibilities appear to be in visual attention and other basic processes. The LGN is known to be responsible for cancelling out redundant information from the retina prior to the next level of processing in area V1 [22]. This redundancy removal is likened to a normalization process and as such, the LGN is not required for this model.

The choice to not use the LGN in our model is due primarily to its anatomical location; it is clearly not directly involved in recognition or motion processing. It is located in the thalamus region of the brain, which forms many crucial circuits believed to be involved in consciousness. The thalamus also plays a large role in sleep and wakefulness. The LGN connects both the visual and auditory systems and is the most logical route through which the auditory system could direct the visual system's attention. The close link to the LGN and consciousness indicates that it may play a role in basic physiological tasks such as controlling the retina, eyelids and other basic biological operations. None of these are required in our model.

2.3.2 Area V1

Cells in Area V1 have the smallest receptive fields and are activated in virtually every type of visual processing [21]. Cells at this level activate in response to a preferred eye, yet they retain the capability of being activated by either eye [37]. This is an instance of neuroplasticity, a common property of the cells in the brain. Their connectivity may allow the visual cortex's structure to optimize itself if an eye is damaged or lost. V1 is responsible for the simplest form of visual processing: V1 simple cells can be modeled accurately with local spatio-temporal filters [39, 42].

2.3.2.1 V1 Simple Cells

The simple cells used in our model correspond to the classical simple cells found in V1, as outlined by Hubel [30]. The receptive fields of simple cells are accurately modeled with a Gabor function, as defined in Equations 2.1, 2.2 and 2.3. The Gabor function is used to create a filter kernel that is convolved with the visual field from the retina. Examples of filter kernels can be seen in Figure 3.6 and the resulting visual field after the convolution operation can be seen in Figure 3.9. Simple cells are typically modeled using a static filterbank [39, 42].

$$F_{\lambda\theta\psi\sigma\gamma}(x,y) = \exp\left(-\frac{x_1^2+\gamma^2 y_1^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x_1}{\lambda}+\psi\right) \tag{2.1}$$

$$x_1 = x\cos\theta + y\sin\theta \tag{2.2}$$

$$y_1 = y\cos\theta - x\sin\theta \tag{2.3}$$

σ is the radius of the Gaussian and is directly related to λ, as defined by Equation 2.4. γ is the aspect ratio, ψ is the phase offset and λ is the period. An example of how modifying these parameters affects the Gabor kernel can be seen in Figure 3.6.

$$\frac{\sigma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{\ln 2}{2}}\frac{2^b+1}{2^b-1} \tag{2.4}$$

The ranges of values used can be found in Table 2.1. Unlike in previous models [39, 35], this model directly takes into account the bandwidth parameter b, as defined in Equation 2.4. This is due to the fact that initial experimental results show that bandwidth plays an important role in both the lower level and higher levels of the visual cortex. The bandwidth parameter is a crucial property of texture operations, such as modeling the grating cell operator [15]. This operator is a promising biologically motivated texture recognition technique that significantly outperforms other texture segmentation and classification methods [16]. We chose to use the bandwidth parameter in tuning, instead of σ, due to its relation to the grating cell operator.

2.3.2.2 Gabor Filter Parameter Ranges

It is known that the size of cellular receptive fields grow as the visual field progresses through the visual cortex. This corresponds to an increase in filter kernel size. The other parameters of the simple cells are given in Table 2.1. The parameters in Table 2.1 show the range of values in our system. While other models typically use a static set of parameters. our model has the

	Kernel	θ	ψ	b	γ	σ	λ
Our Model	$[5, 31]$	$\frac{0}{4}, \frac{\Pi}{4}, \frac{2\Pi}{4}, \frac{3\Pi}{4}$	$[-\frac{\Pi}{2}, \frac{\Pi}{2}]$	$[0.2, 3.0]$	$[0.0, 3.0]$	-	$[0.0, 6.0]$
Serre [40]	$[5, 35]$	$\frac{0}{4}, \frac{\Pi}{4}, \frac{2\Pi}{4}, \frac{3\Pi}{4}$	0	-	0.3	4.5	5.6
Mutch [42]	11	12 orient.	0	-	0.3	4.5	5.6

Table 2.1: Gabor kernel parameter ranges for various models

unique ability of being able to search a wide range of values for optimal feature extraction parameters.

The values used here are similar to the values commonly used in other biological models [35, 39, 42]. The filter parameters used for feature descriptor creation are selected with an iterative cellular tuning method, as outlined in Chapter 3. The set of all possible filter parameter creates a formidable search space. Certain properties of the cells are known to change from region to region in the cortex in a predictable manner. For instance, the ψ parameter is not frequently changed during feature extraction in area V1 and V2. But at higher areas of the visual cortex, highly textured inputs will more readily activate when ψ is modified, as shown by the grating cell operator.

2.3.2.3 Textures and Periodic Grating Cells

While outside the scope of our work, areas V1 and V2 have cells that respond to higher complexity inputs, such as those found in periodic grating cells [13, 15]. The corresponding grating cell operator provides an ideal basis for various texture based feature operations in Gabor space [14]. The grating cell operator's excellent performance at texture segmentation, along with its high biological plausibility, both support the fact that it is a key aspect in the visual system's processing. The grating cell operator is outside the scope of this model due to its computational expense and its lack of applicability to recognition or disparity. Yet, it is a crucial type of cell in the visual cortex and it remains highly compatible with this work. It would certainly stand to benefit from an implementation on the programmable graphics processor.

2.3.2.4 Complex Cells and Cortical Columns

The cortical column is a crucial structure in both the visual cortex and in our work. A cortical column is defined as a series of simple cells that are anatomically arranged into a column with cells differing only by their θ value, as shown in Figure 2.6. They are found in the visual cortex

[30], yet most models do not directly account for their presence [39, 42, 33, 35]. This is largely due to the fact that their responsibilities are not well known. Cortical columns are effectively a grouping of simple cells. Simple cells are connected to complex cells. Complex cells serve to aggregate the output of simple cells that are arranged into cortical columns. Complex cells are generally modeled with the use of the max operation and have the result of pooling a group of retinotopic simple cell columns. Patterns of activation of cortical columns within the visual field are of high interest for object recognition. These areas are often the regions within the visual field where the object's geometry changes or the disparity within a scene changes. The output from the complete collection of retinotopic cortical columns represents the entire visual field.

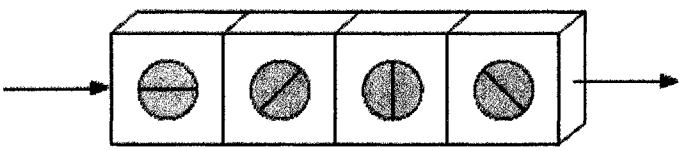


Figure 2.6: Simple cells in a cortical column. Each cell is shown as a box and has its preferred orientation encoded on the front. Visual data is processed in the direction of the arrow. Each column is tuned to a specific eye and left / right eye columns are alternated next to one another in the visual cortex.

Cortical columns are used within our model to define corner and edge components. Certain groups of cortical columns form corners, while other distinct groups of cortical columns form edges. The next area of the visual cortex, V2 is known to respond to luminosity and contrast ratios along object contours in the visual field. V2 serves the purpose of further processing V1's visual field by linking salient corners within an image to one another using a path-based edge detector.

2.3.3 Area V2

Area V2 has feedforward connections with V3 and V4, as well as feedback connections to area V1 [45]. Area V2 is known to have a complex response to contrast and luminance. Area V1 shows a linear contrast response, while area V2 shows a much higher contrast response than V1 [5]. Area V2 exhibits high gain at low contrast levels and has specialized thin stripe regions for processing higher contrast areas [4]. V2 is also known to respond strongly to the illusory contours of objects [54, 48]. Area V2 is clearly responsibility for a higher complexity feature

processing based on local luminosity and contrast ratios.

In area V2, the resulting visual field from area V1 is used to build higher order object contours, which form the basis for feature descriptors. Simple and complex cells are also present in this area, although their respective tasks change. V2 simple cells compute neighborhood luminosity values and complex cells again pool the responses from this operation. In this work, paths are used to define object contours. This models the properties of thin stripe regions present in Area V2 [4]. Paths are explored based on the filter response computed by Area V1, which represents these thin stripe regions. Local contrast ratios are used to provide simple feature ranking information, which is useful for matching features between eyes. It is important to note that luminosity values are more predictably normalized when compared with many techniques using color histograms.

Once the visual field has progressed past area V2, the visual cortex separates into the ventral and dorsal pathways. Area V2 is the highest shared area of these pathways. The pathways are separately responsible for very distinct, higher complexity processing such as motion field calculation, geometric projection and object recognition. Object contour paths form a basis for geometric projection, although geometry is not formally explored in our model. Geometric boundaries provide a basis for texture processing and recognition. Both geometry and texture are required in order to perform high level recognition tasks, such as face recognition.

2.3.4 Dorsal and Ventral Pathways

As seen in Figure 2.5, the visual cortex splits into two pathways: the dorsal pathway and the ventral pathway [28]. These higher areas are responsible for tasks such as local motion, disparity, geometric projection, global motion, object recognition and color processing. The dorsal stream is primarily responsible for time-based tasks, such as motion processing, and is known as the "where" stream [28]. The ventral stream is mainly responsible for object recognition and is known as the "what" stream [28]. The ventral stream is responsible for all forms of object recognition, including face recognition done in the specialized Fusiform Face Area (FFA) of the brain [32].

All areas of the visual cortex in this model, including the ventral and dorsal streams, are implemented using the graphics processor. Area V2 is responsible for creating feature descriptors that can be used in both the ventral and dorsal pathways. These descriptors are used to model

the functionality of the ventral pathway in a similar manner to previous models [39, 35, 42]. We also perform disparity estimation in the dorsal pathway to demonstrate both the robustness of our model and its feature descriptors.

2.3.4.1 Ventral Pathway: V4, V8

The ventral pathway is primarily associated with object recognition. It shows properties of highly parallel processing and is able to identify multiple objects in parallel within a cluttered scene [24]. It is connected to the limbic system, which is involved in both emotions and memory. The ventral pathway is also connected to the temporal lobe, which is responsible for long term memories. This is absolutely crucial in object recognition. The ventral stream consists of V1, V2, V4 and also connects to the temporal lobe through the posterior, central and anterior inferotemporal (PIT, CIT and AIT) areas. The latter regions have received much less study than areas V1 and V2; models of the cortex generally do not attempt to model these areas in a biological fashion. We simply note the presence of these areas and attempt to abstract their functionality with an equivalent simplified database of feature descriptors.

2.3.4.2 Dorsal Pathway: V3, V3a, V5

The dorsal pathway is responsible for motion tracking in the human visual cortex. Motion processing is an important property of the visual cortex and is known to occur in a local to global manner [21]. The highest areas of the brain are responsible for more complex motion-based tasks. Area V3 responds to local motion [49], area V3a is known to respond to disparity [48] and area V5 / hMT responds to global motion, also known as optical flow[51]. Disparity is the key attribute that defines object boundaries and the visual cortex is thought to carry this information through a retinotopically projected three dimensional map.

The visual cortex has areas that respond specifically to biological motion such as running, jumping, hand, eye, and mouth movements [21]. Specific areas also respond to the gender of the actor and can do so even in impoverished lighting conditions [23]. The retinotopic processing performed in area V2 must be conducive to both the computation of motion and disparity models in the dorsal stream and object recognition in the ventral stream. The dorsal stream is effectively the brain's analogue to real-time stereo video processing in computer vision.

2.3.5 Cellular Tuning

The cellular tuning used in our model is an iterative approach that is designed to model the arrangement and nature of simple and complex cells in the visual cortex. Simple cells are used to perform computationally expensive operations, such as applying Gabor filters. Complex cells are then used to compute a higher order function with the output of simple cells. The output of complex cells is then used to further tune the next series of simple cells for the following iteration. This is done using the MAX operation, which is known to be part of the complex cells computational process [31].

In our model, each set of simple and complex cells forms a tuning layer. By iteratively applying simple and complex cell processing phases, a series of tuning layers is produced. These tuning layers traverse the cellular search space and results in cellular parameters that are optimal for feature extraction in the current visual field. This iterative tuning approach is a key aspect of this work: it allows the system's parameters to adjust to any scene lighting conditions or geometric complexity. This is further outlined in Chapter 3 and Chapter 4.

2.3.6 Lesions in the Visual Cortex

Although not directly explored in this model, one of the crucial and unique properties of the brain is its ability to adapt to damage, such as lesions. This gives crucial insight into both its neuroplastic properties and the specific responsibilities of key regions. For instance, color vision is primarily handled by area V8 deep within the ventral pathway. This is known due to the condition called cerebral achromatopsia, resulting in color vision loss. This is correlated with damage in area V8, also known as the ventral occipitotemporal cortex [2]. It is also known that lesions in area V5 / hMT of the dorsal stream lead to a failure to perceive motion; this disorder is known as cerebral akinetopsia [25].

Understanding the effects of lesions such as these is crucial to achieving an accurate functional model of the visual cortex. Lesions at lower levels do not provide a good deal of information for visual cortex models, since they can significantly affect the extent and nature of the residual vision. This model uses lesions to justify key assumptions, such as how lesions in V8 indicate the location of color perception, allowing it to be discounted for lower levels. Existing visual cortex models can be thought of as suffering from both cerebral achromatopsia and cerebral akinetopsia. It is therefore important to study these disorders in order to further understand the nature of the brain and its limitations.

2.4 The Modern Programmable Graphics Processing Unit

The human brain is clearly very parallel in nature. It is able to deconstruct highly complex scenes containing countless objects and perform object recognition, all in real-time. The modern programmable Graphics Processing Unit (GPU) is an excellent platform for parallel computation. Its parallelism makes it a natural platform for modeling the visual cortex. The GPU is commonly abstracted as a stream processor and it is ideal for a number of high computation processing problems [64, 65]. The programmable GPU can be programmed with pixel and vertex shaders. These are programs that allow direct control over the graphics pipeline of the GPU. Pixel shaders are a type of program that executes on the graphics processor over every pixel within a texture. Vertex shaders are a related type of program that executes over a geometry. In our work, pixel shaders are written in the OpenGL Shader Language (GLSL), as defined in the OpenGL 2.1 specifications [87]. Pixel shaders are used to implement the simple and complex cellular operations described in Chapters 3 and 4. An introduction to GLSL programming for the GPU and a high level overview can be found at [80].

An example of a pixel shader can be found in Appendix I. Pixel shaders are executed separately on each pixel in the destination texture. This is the essence of Single Instruction, Multiple Data (SIMD) processing: identical instructions are executed on all pixels in parallel. The main responsibility of each pixel shader's execution is to output a single RGBA value for the current pixel. In doing this, it can perform any number of mathematical operations as well as sampling from other textures. In our model, the kernel of the Gabor filter is stored in a texture and the source image is stored in a separate texture. The shader in Appendix I shows how these two textures can be combined in order to perform a convolution operation. Note that each shader cannot read from the same pixel to which it writes and it can only write to a single pixel within the destination texture at a time, since execution of neighboring pixels generally occur in parallel to one another. A shader normally only writes to a single texture in a rendering pass, although modern GPUs do support writing to multiple textures in a pass.

In our model, each of the simple and complex cells in V1 and V2 exist as one or more pixel shader programs. The execution of a simple cell shader is always proceeded with the execution of its corresponding complex cell shaders. Figure 2.7 shows the general layout of simple and complex cell shader program execution. Simple cell shaders are tasked with the more computationally expensive operations, such as applying Gabor filters. Complex cells are responsible for pooling the responses from a cluster of simple cells. Note that with pixel shaders, a texture

library is used for all cellular input and output to and from the CPU. The actual shader implementation for all complex cells requires multiple rendering passes each with its own shader program. It is simplified in Figure 2.7 for brevity.

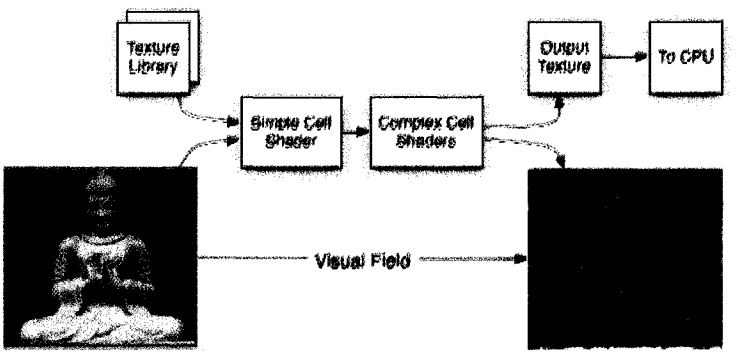


Figure 2.7: Execution of simple and complex cell pixel shaders

An optimization used in our GPU based approach is to place the four cortical column components of the visual field into a single RGBA texture. With this approach, the Gabor response at $\theta = 0$ is placed in the red channel, $\theta = \frac{\pi}{4}$ is placed in the green channel and so forth. The result of this process can be seen in Figure 3.2. This lets a single texture represent the entire visual field. More importantly, it allows the processing of all cortical columns to be done in a single render pass. Modern GPUs also support operations on multiple textures: a single render pass can write to multiple textures with a shader. This would be appropriate for performing more convolutions using a wider array of Gabor filter orientations. Using multiple textures would also allow the execution of a more complex operation, such as the grating-cell texture segmentation defined by Petkov [15]. Both of these approaches would effectively involve modeling a longer class of cortical column. This is useful in more complex recognition tasks, such as in face recognition where better results are shown when θ is allowed to have 8 or more orientations [17].

Chapter 3

Model Area V1: Cellular Tuning

This chapter gives an overview of our model's structure and fully describes the first area of our model, known anatomically as area V1. As in the visual cortex, area V1 in our model is responsible for some of the more computationally expensive operations of all areas in the visual cortex. Area V1 in our model is designed to operate well in a wide range of low lighting and cluttered environments. In order to achieve this type of invariance, the features used by higher visual cortex areas are derived from a visual field which factors out parameters such as lighting conditions and clutter. This provides a stable basis on which operations providing even higher order invariance can be performed, such as rotation and scale invariance. The role of V1 in our model is to bring invariant parameters into the visual field for use by all higher levels. This is achieved by adopting an iterative tuning approach to select the model's cellular parameters for each image, or visual field.

3.1 Model Overview

Figure 3.1 shows the overall design of our model and the states used in our model to process the visual field. The model begins with the retinal image traversing the optic chiasm to area V1, which we liken to the uploading of the current image to the graphics processor. Area V1 performs a tuning operation to select the best parameters to view the scene, as outlined later in this chapter. The visual field then progresses to area V2, where object contours and other properties are used to define feature descriptors, as outlined in Chapter 4. Feature descriptors are extracted at area V2 and are used for object recognition and disparity maps. All blue colored states in Figure 3.1 are executed directly on the graphics processor and each consist of a number of GPU pixel shaders. Once area V2 has completed execution, feature descriptors

are extracted and stored to disk. In our work, the ventral and dorsal streams are also modeled on the graphics processor; this is further explored in Chapter 5. This chapter serves to describe the processing done in the simple and complex cells within area V1 of our model.

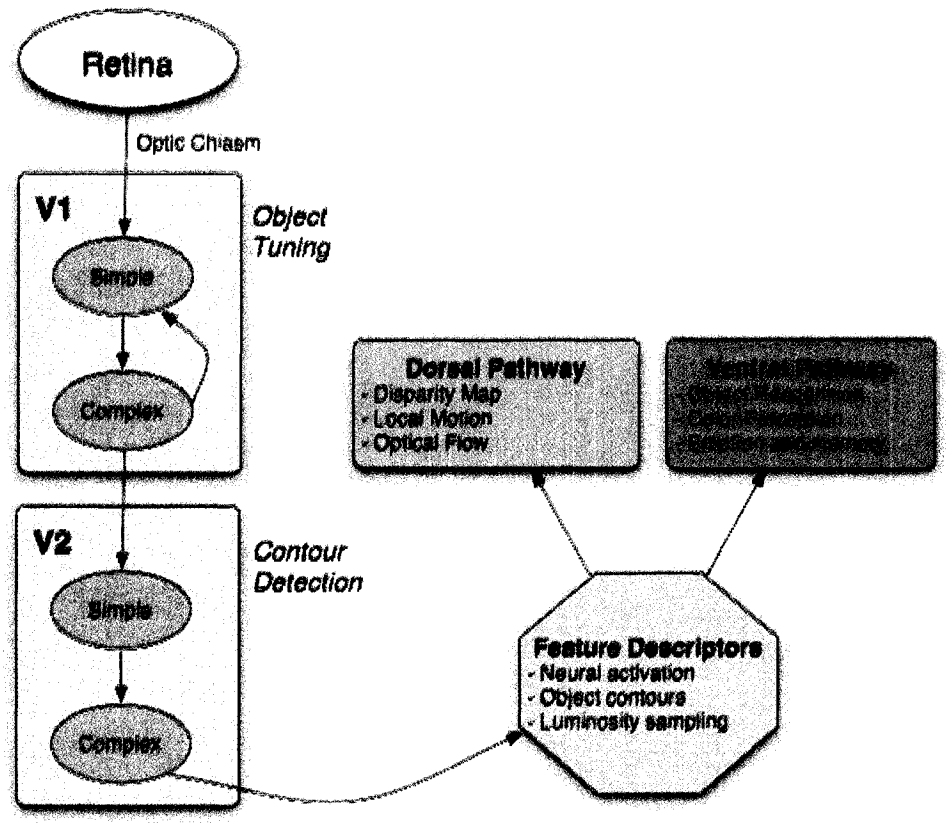


Figure 3.1: The processing states used in our model

3.2 Cells in Model Area V1

V1 is the first area of both our model and the visual cortex. Anatomically, it is directly connected to both the Lateral Geniculate Nucleus (LGN) and area V2, as seen in Figures 2.4 and 2.5. There are two main classes of cells at this level of the brain, simple and complex cells. In our model, simple cells act as spatio-temporal filters and complex cells are responsible for pooling their response. By alternating simple and complex cells, a tuning process allows for the efficient traversal of a large search space. This search determines the optimal cellular parameters with which to process the visual field.

3.2.1 V1 Simple Cells

Cells at the V1 level of the visual cortex have their responses primarily tuned to a single, preferred eye [37]. In our model, feature descriptors are constructed using a pooling operation over anatomically inspired regions of interest called cortical columns, seen in Figure 2.6. An image feature descriptor is a collection of inter-connected cortical columns within a particular retinotopic neighborhood. V1 simple cells, as outlined in Section 2.3.2.1, simply perform a convolution operation on the visual field. Each cortical column links a series of simple cells, effectively linking several convolution operations. Each complex cell responds to activations within groups of cortical columns.

In our model, the strength of a given cortical column's neural activation is a key property for feature descriptors. This cortical column activation is based on the response of cells in that retinotopic neighborhood of the eye's visual field. The edge test for a single simple cell at a particular retinotopic location (x, y) is $f_E(x, y)$, as shown in Equation 3.1. This activation is effectively an edge region that is combined over all orientations of Gabor responses to form a corner region. The activation of an entire cortical column is represented as a region where edge components overlap, shown in 3.2.

$$f_E(\theta, x, y) = \begin{cases} 1 \text{ if } C(\theta, x, y) > t_A \\ 0 \text{ otherwise} \end{cases} \tag{3.1}$$

$$k_E(x, y) = \sum_{\theta=0}^{\pi} f_E(\theta, x, y) \tag{3.2}$$

$C(\theta, x, y)$ corresponds to an entry in the cortical column with Gabor filter parameters $F_{\lambda\theta\psi\sigma\gamma}$, shown in Equation 2.1. t_A is the activation threshold and shows acceptable results when set to a value of 10% of the maximum value of $C(\theta, x, y)$. $f_E(\theta, x, y)$ activates when in the presence of an edge-based primitive at orientation θ. The response generated by $k_E(x, y)$ represents the neural activation of a single cortical column at retinotopic location (x, y). This is the simplest form of activation and serves to represent a columnar activation, similar to that found in [20, 26]. Each column is defined both at a specific retinotopic location (x, y) and over all possible values of θ in Gabor function $F_{\lambda\theta\psi\sigma\gamma}$, from Equation 2.1. The individual columnar activations are likened to a rudimentary edge detector; however, they are not themselves of particular interest at the V1 level. The crucial aspect here is patterns of columnar activation within a local neighborhood, or corners. Corners are defined in Equation 3.3.

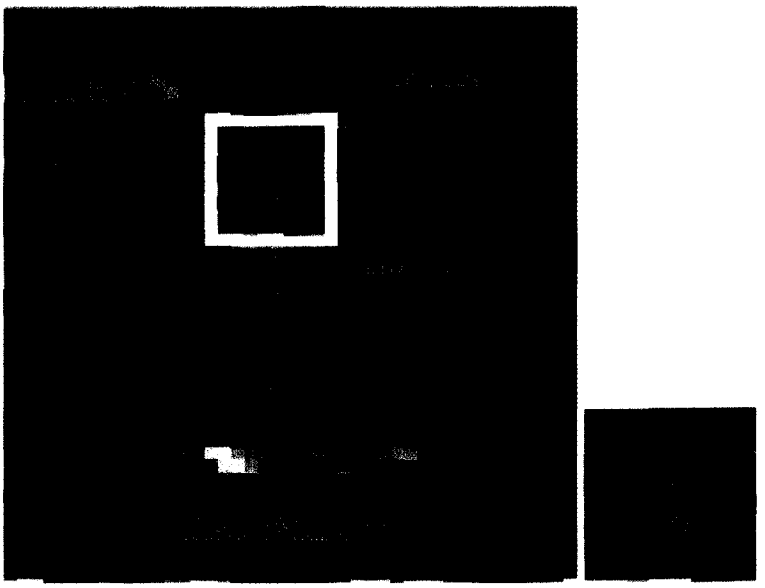


Figure 3.2: Cortical column activation in a given visual field. Both $\theta = 0$ and $\theta = \frac{\pi}{2}$ are shown as green, $\theta = \frac{\pi}{4}$ as blue and $\theta = \frac{3\pi}{2}$ as red. The white box in the left image shows region in the right image. All pixels shown as pink (overlap between the blue and red channels) have $k_E(x, y) > 1$; a subset of these have $k_C(x, y) > 0$. Original image shown in Figure 3.7.

$$k_C(x,y) = \begin{cases} \sum_{\theta=0}^{\pi} C(\theta, x, y) \text{ if } \left(\sum_{[i,j] \in N^L} k_E(x+i, y+j)\right) = t_C \\ 0 \text{ otherwise} \end{cases} \tag{3.3}$$

$k_C(x, y)$ reflects the neural activation of a corner component at location (x, y) in neighborhood N^L of the visual field. The summation neighborhood N^L is a union of all elements in an LxL neighborhood centered at (x, y). t_C is a threshold proportional to the size of N^L and shows good results when set to 30% of the maximal value for the given neighborhood. Experiments show that the size of L at this level of the cortex is best set to 3x3 or less. Like many cellular processes in the cortex, neighborhood regions become larger at higher levels of the cortex.

An example of the visual field produced by a retinotopic group of cortical column activations is given in Figure 3.2. Note that computing $k_E(x, y)$ is much less computationally expensive due to the fact that the cells within a column all share a common θ and (x, y). The computation $k_C(x, y)$ is more expensive due to the neighborhood traversal.

3.2.2 V1 Complex Cells

In order to obtain the optimal cellular parameters for viewing a given scene, complex cells are used to determine which simple cells will provide the highest number of feature points. Complex cells activate in response to the simple cells that have generated the highest ratio of corner to edge points over the entire retinal image. This is computed with Equations 3.4 and 3.5. Complex cells are pooled over a series of simple cells $j = 1$ to M. This pooling process happens repeatedly in an iterative fashion from $i = 1$ to N. Area V1 can be thought of as having length N and width M.

$$k_E^{i,j} = \sum_{(x,y)\in I^{i,j}} k_E(x,y) \tag{3.4}$$

$$k_C^{i,j} = \sum_{(x,y)\in I^{i,j}} k_C(x,y) \tag{3.5}$$

$I^{i,j}$ represents the visual field after it has been processed by the simple cells at level (i,j) in the cortex; the process by which $I^{i,j}$ is obtained is outlined in Section 3.2.3. The final value for the pooled cellular settings at level i, with Gabor settings $F^i_{\lambda\theta\psi\sigma\gamma}$, is obtained with Equation 3.6.

$$F^i_{\lambda\theta\psi\sigma\gamma} = \arg\max_{j=1}^{M} \left(\frac{k_C^{i,j}}{k_E^{i,j}} \right) \tag{3.6}$$

This achieves the result of selecting the simple cells with the highest ratio of corner to edge points from a collection of simple cells. This is the primary criteria used for tuning in our system, but it is of note that other criteria may function equally well for this task. The complex cell maximization process is used to dynamically select the best filter parameters according to the intrinsic properties within the visual field by using a tuning mechanism outlined in Section 3.2.3. Figure 3.4 shows sample responses from a number of simple cells connected to a given complex cell; their visual field comes from the source image in Figure 3.3.

In this model, complex cells are involved in finding simple cell settings that optimize the ratio of edge to corner responses. The entire tuning process produces cellular settings with the optimal ratio of corner and edge regions.

3.2.3 Iterative Cellular Tuning

We use alternating simple and complex cells to form a tuning process that traverses a large cellular search space. Algorithm 1 shows the process in detail. We use an iterative approach,

Figure 3.3: Tuning source image used for results in Figure 3.4, taken from [72].

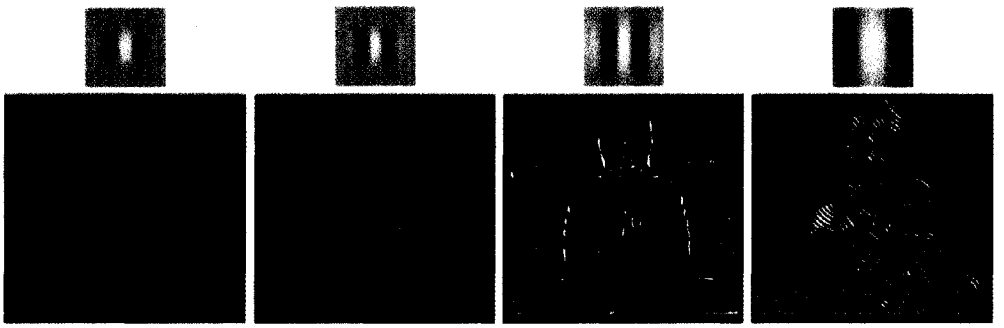


Figure 3.4: Sample responses from a single level of tuning. The lower images from left to right show samples from $F^{i1}_{\lambda\theta\psi\sigma\gamma}$ to $F^{iM}_{\lambda\theta\psi\sigma\gamma}$. The corresponding filter kernels are shown above the image. Green corresponds to the response at $\theta = 0$ and $\theta = \pi$, red corresponds to $\theta = \frac{\pi}{4}$ and blue corresponds to $\theta = \frac{3\pi}{4}$. The second tuning from the left is the winning node for this level.

tuning from levels $i = 1$ to N. The process begins at level 0 with a set of default cellular parameters. At each level i, a series of simple cells $j = 1$ to M are created by permuting the cellular settings from the winning cells at level $i - 1$. These new simple cells correspond to Gabor parameters $F^{i,j}_{\lambda\theta\psi\sigma\gamma}$. Each set of newly generated simple cells are applied to the visual field. All sets of simple cells are aggregated in the execution of the complex cells, which forms a single tuning level. Each tuning level has a winning set of simple cells selected; this is the winning cells for the level.

The winning cells for level i have parameters $F^{i}_{\lambda\theta\psi\sigma\gamma}$. The tuning process is applied for N iterations, which results in the final cellular parameters $F^{N}_{\lambda\theta\psi\sigma\gamma}$. These are the result of the tuning process in V1 using the current visual field. At this point, the visual field has completed

Algorithm 1 Tuning Algorithm

```
function tuneToVisualField(f) {
 cells = defaultCells()
 bestCells = cells
 // apply N levels
 for i = 1 to N {
  // test all cellular settings at the current level
  for j = 1 to M {
   // create and test a new F^{i,j}_{λθψσγ}
   newCells = cloneAndPermuteCells(cells,i,j)
   if cornerRatio(f,newCells) > cornerRatio(f,cells) {
    bestCells = newCells
   }
  }
  if bestCells == cells {
   // no better cells could be found, exit early
   return cells
  }
  else {
   // current iteration's best cells
   cells = bestCells
  }
 }
 // return optimal cells
 return cells
}
```

traversal of V1 and it is then passed to area V2. The maximization process from the complex cells, when applied iteratively, produces the ideal cells for processing the visual field.

Figure 3.5 shows the selection process for a single iteration and how the winning cells are used to generate a new set of cells for the next iteration. This shows the process from the inner loop in Algorithm 1 where each cellular setting is tested. Figure 3.6 shows an example of all filter kernels tested in a single iteration i.

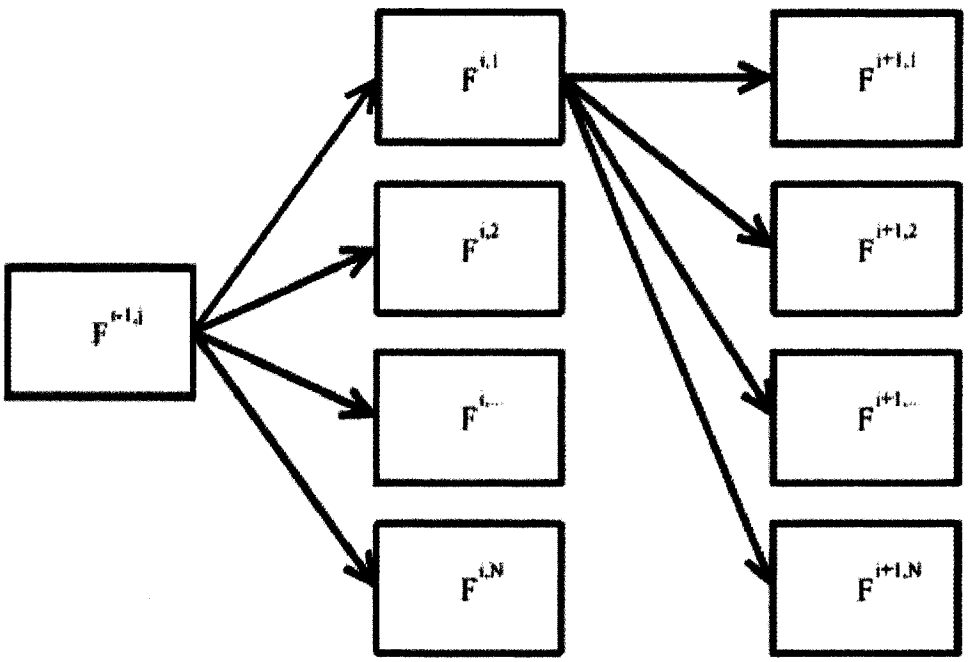


Figure 3.5: The process of generating a series of new responses from level $i-1$ to level i is shown. The winning node for level i is shown in red; the process is repeated for level $i+1$.

3.2.3.1 Cellular Search Space

The visual cortex has a wide range of cells with which it can process visual data; attempting to perform the computation of every class of cell in the visual cortex is not feasible. Using the cellular tuning model described here allows a substantial search space to be traversed with a much lower computation cost than the equivalent static filter bank. The process of selecting the optimal cellular parameters in an iterative fashion can be viewed as a form of hierarchical cellular selection and produces cellular parameters with lighting and rotation invariant properties which are explored in Chapter 4.

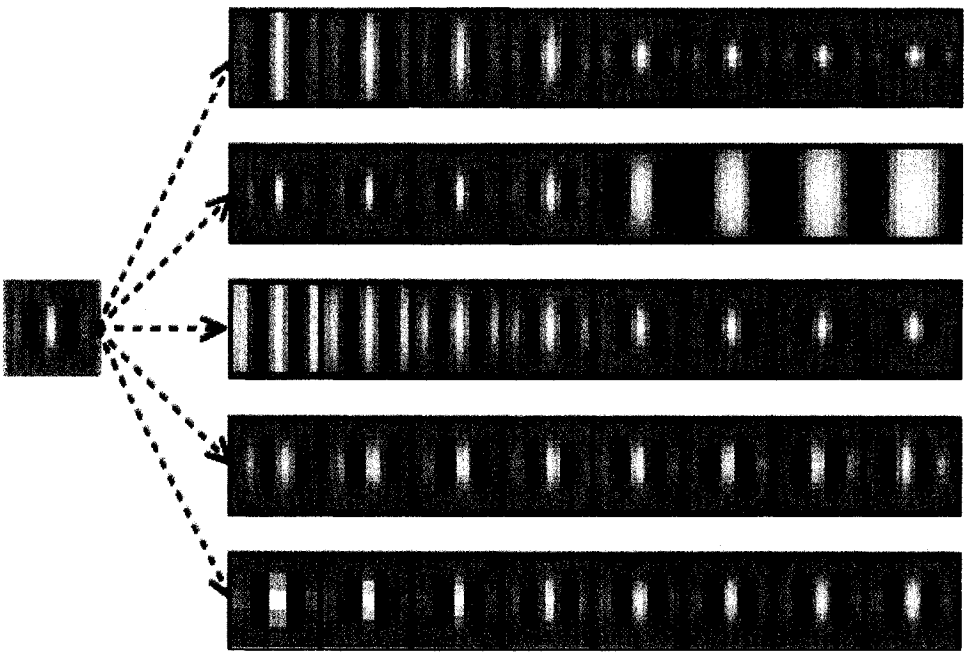


Figure 3.6: Gabor filter kernel search space.$F^{i}_{\lambda\theta\psi\sigma\gamma}$ is highlighted in green, each row contains filter kernels with a specific tuning parameter modified. The set of all kernels is made up of responses $F^{i1}_{\lambda\theta\psi\sigma\gamma}$ to $F^{iW}_{\lambda\theta\psi\sigma\gamma}$. The winning candidate for phase $F^{i+1}_{\lambda\theta\psi\sigma\gamma}$ is highlighted in red. This illustrates the filter search space for a single round of tuning. Level $i+1$ will create a completely unique set of filter kernels based on the value of $F^{i+1}_{\lambda\theta\psi\sigma\gamma}$

This iterative approach makes the connective properties of our complex cells to simple cells much more biologically plausible than other models. This tuning process closely mimics the exitatory influence that complex cells have on the receptive fields of simple cells, as found in [55]. Other cortex models use complex cells only for the purpose of pooling of simple cell responses and do not allow the same complex to simple cell connectivity [35, 39]. This tuning process is also much more conducive to creating feedback connections similar to those present in the visual cortex, unlike other models.

3.3 Attention-Based Processing

The visual cortex is primarily modeled as a feedforward network [39, 40, 44]. One of the core processing attributes of the visual cortex is its feedback connections [18, 44, 52]. These feedback connections allow higher areas of the brain to adjust properties of cells in lower visual cortex areas. This manifests itself as an attention-based tuning of the cells in the cortex,

which beginnings most predominantly at V2 [19]. This is likened to focusing ones attention on a particular object within the current scene. V1 also shows evidence of having feedback connections [52], but is much more limited in terms of its attentional tuning properties. When higher order process of the visual cortex requires further processing for a particular task such as face detection, our model supports the creation of a feedback connection to initiate further processing, as suggested in [18].

The tuning model presented here allows the dynamic selection of cells that are best suited for the processing present in the next area of the visual cortex. Feedback connections achieve the result of continuing the tuning process by repeating it on a specific subset of the retinal image. This is done when a higher level cortex area does not have sufficient coverage of a particular region. This is referred to as attentional processing, since it occurs when one focuses attention on a particular object. An example involving the use of a feedback loop from V2 to V1 is shown in Figure 4.8.

Through this feedback mechanism, the visual cortex is able to adjust its cells for various cluttered viewing conditions. Feedback loops are achieved in the model by providing further cell tuning functionality on sub-regions of the original image in a recursive manner. This localized processing provides a basis for the attentional tuning model that is designed to mimic the biological process of focusing one's attention on a particular object or region within a cluttered scene.

3.3.1 Tuning over Feedback Connections

Most models of the visual cortex use a static Gabor filterbank [39, 42, 35]. This approach is limited in some respects: one must select a predetermined number of filters and all corresponding filter parameters. All filters in the filterbank are generally applied for all images analyzed. The size of the filterbank reflects the amount of information that the model can extract. The relative volumes and structure of the components of the visual cortex varies quite highly from person to person [43] and species to species. Instead, the iterative process presented here allows our model to easily adjust its cover of the filter parameter search space; this allows it to scale to model species with more advanced or limited visual systems as need dictates. It is also much more highly flexible in that no assumptions are made about individual cell parameters or their particular quantizations.

Figure 3.7: Source Image for tuning results in Figure 3.8

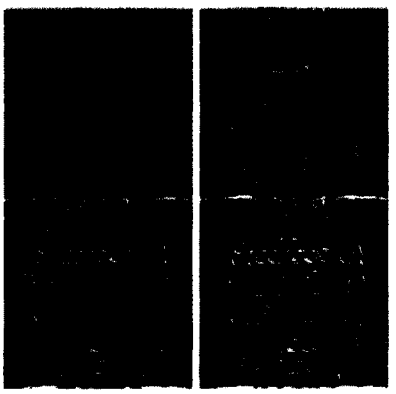


Figure 3.8: Retinotopic visual field after four iterations of cellular tuning on Figure 3.7

Our model primarily uses feedforward processing for computation. The tuning process can be extended with the use of feedback calls. This simply consist of higher order areas of the cortex calling the lower areas to perform operations on sub-regions of the original image to finer tune the results. This occurs when an existing feature is required at higher resolution than the initial tuning phase achieved. Feedback connections occur with V1, but are known to be much more widely used in V2. Feedback connections are simplified considerably with the use of the programmable graphics processor.

3.4 Higher Cortex Areas

Once the tuning process has completed the task of determining the most appropriate cells and has processed the visual field accordingly, V1's retinotopic visual field progresses to area V2. This second area is responsible for more complex feature tuning and extraction process. The V1 tuning process can select filter parameters that vary quite widely from scene to scene. Shown in figures 3.9 are some example results from each level in the tuning process. Figure 3.8 shows the visual field corresponding to source image 3.7 as it progresses through four iterations of the tuning process.

3.4.1 Interface to Area V2

Visual cortex area V2 serves to continue processing the visual field from the cellular tuning in area V1 by extracting feature descriptors. The main parameters connecting these two areas in the visual cortex are the properties of the tuned cells and the retinotopic visual field itself. V2 uses identical definitions of corner and edge points as V1, seen in Equations 3.2 and 3.3. Area V1 is known to be connected in a feedforward manner to V2, but is also known to have feedback loops from higher areas in the brain, such as V3 [45]. It is important to note this model's capacities in terms of modeling these feedback properties, since they obviously play a crucial role in attention and the human visual cortex. The primary connection type used in this work is feedforward, but feedback loops are explored in Chapter 4.

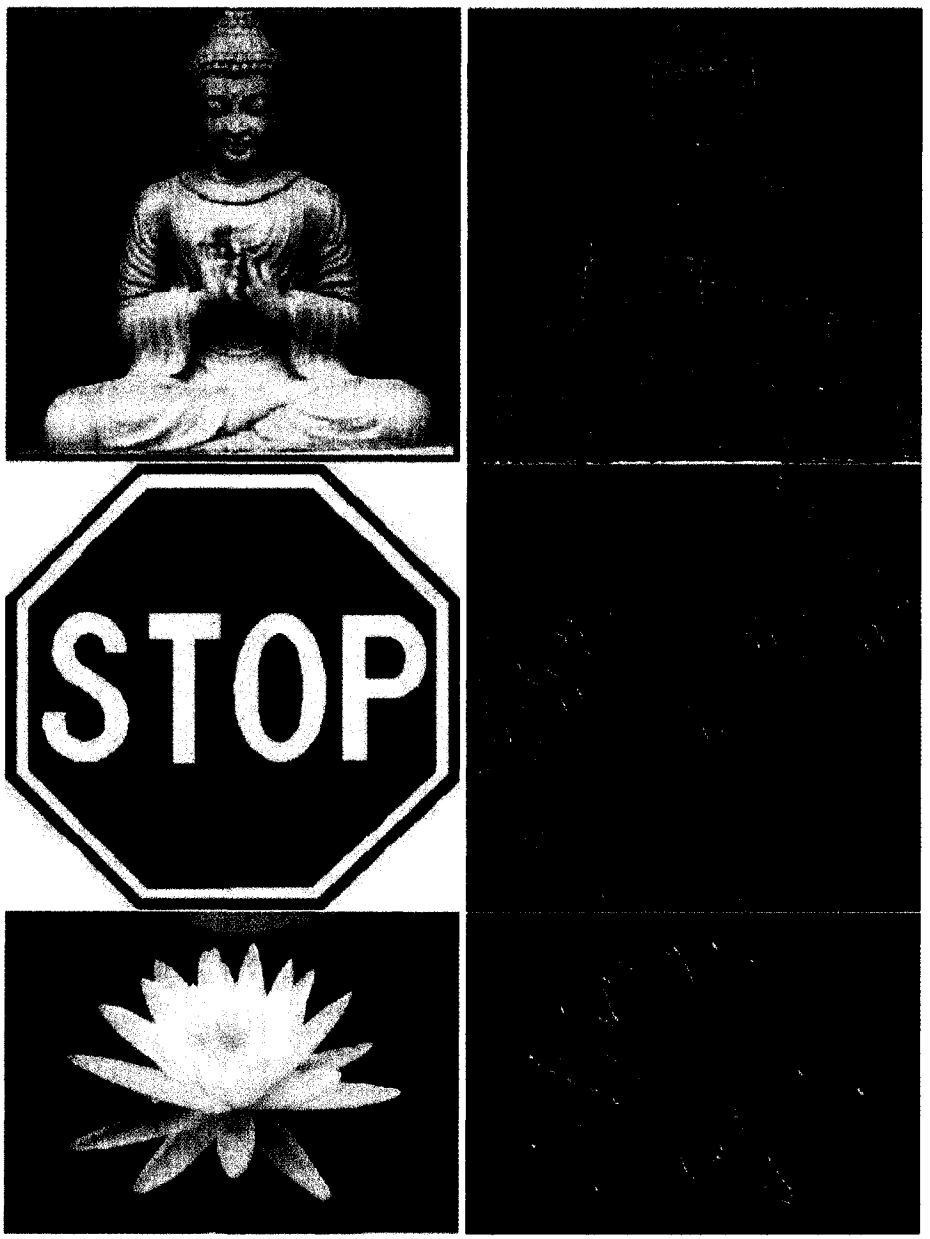


Figure 3.9: Resulting visual field from V1 for various objects

Chapter 4

Model Area V2: Feature Descriptors

This chapter describes the second area of our model, known anatomically as area V2. In a manner similar to the visual cortex, our model's area V2 is responsible for higher order processing of the visual field obtained from V1. Anatomically, much less is known about the function of V2 compared to V1. While V2 is as large as V1 in both macaque monkeys and humans, the number of studies using fMRI to explore area V1 outnumber area V2 on the order of 10:1 [7]. Cells in V2 have complex luminance and contrast ratio based activations. To model these complex activations, our feature descriptors rely on luminance-driven operations. Area V2 is also known to respond to the illusory contours of objects, implying that it is responsible for a form of object-defining feature descriptor. We chose a path-based approach to represent the illusory contours of objects within the visual field.

The feature descriptors in our model have the following components: a primary path orientation, path length, a luminance value sampled along the path, a retinotopic area of the visual field corresponding to the path and the neural strength of the cortical column of the corner from which the descriptor is derived. This information is used to form a feature descriptor similar in nature to the feature prototypes found in other models [39, 35, 42]. A line segment that estimates the primary path is the means used to provide rotation and scale invariance. It also provides an appropriate primitive for the contrast ratio calculation. The prominent role of contrast ratios in area V2 is well supported by biological evidence in the macaque monkey visual cortex [5, 6, 9].

4.1 V2 Cells

Certain primitive shapes have been identified as activating cells in area V2 [10]. The shapes that are of interest to this work are similar to the shapes from [10]. Corner regions with high neural activation are key areas of interest for path-based operations. The edge components of the visual field, while largely ignored in V1, are used to form higher order shape cues in V2 using illusory contours. These contours implicitly form the geometry of the objects within a scene and are used to mimic the biological evidence demonstrated in [10, 11, 12]. There is also strong evidence to suggest that V2 effectively sums the high contrast responses from V1 neurons in humans to form higher order shape cues [7].

4.1.1 V2 Simple Cells

Area V2 is subdivided in a similar manner to V1: simple cells have a calculatory role while complex cells perform a pooling operation over simple cell responses. Simple cells traverse a series of paths within a neighborhood; each path is proportional to the cell's receptive field size. Feature descriptors are defined around the most prominent path in a region. This most prominent path has luminance values sampled during its traversal. Each corner point in area V1, $k_C(x, y)$ from Equation 3.3, is a starting point for path traversal. There are two paths traversed for each value of θ. Simple cells compute these paths over all orientations in the visual field. This gives the final feature descriptor properties that are both rotation and scale invariant. Each path $P^{x_n,y_n}(x, y, \theta)$, defined in 4.1, begins at (x, y) and traverses along an edge where $f_E(\theta, x^i, y^i) = 1$ to stop at destination (x_n, y_n). The path continues only while the edge test is maintained: the path is terminated when an edge ends or when an edge connects to a neighboring corner region. The former situation is when $f_E(\theta, x^i, y^i) = 0$, the latter is when $f_E(\theta, x^i, y^i) > 1$.

$$P^{x_n,y_n}(x, y, \theta) = \bigcup_{(i,j)\in(x_\theta,y_\theta)} \{(x+i, y+j) \,:\, f_E(\theta, x+i, y+j) = 1\} \tag{4.1}$$

(x_θ, y_θ) is a set of path probe directions and are effectively a sampling mechanism. All directions are sampled and if none of them has the appropriate edge state, the path is terminated. This path traversal is computed for all $\theta = [0, 2\pi)$, leaving two sets of directions per Gabor orientation, namely a forward and backward path. Each path's direction (x_θ, y_θ) is implicitly calculated from the orientation of the corresponding θ parameter. For $\theta = 0$, the path sampling is $(x_\theta, y_\theta) = [(1, 0), (1, 1), (1, -1)]$ for the forward direction and $(x_\theta, y_\theta) = [(-1, 0), (-1, 1), (-1, -1)]$ for the backward direction. Path sampling can be seen

Algorithm 2 Path traversal

```
function traversePath(x,y,field,theta) {
 x1 = x
 y1 = y
 while distance(x,y,x1,y1) < maxDistance {
  if sample(x1,y1,directions[theta][0],field) == 1 {
   x1 += directions[theta][0].x
   y1 += directions[theta][0].y
  }
  else if sample(x1,y1,directions[theta][1],field) == 1 {
   x1 += directions[theta][1].x
   y1 += directions[theta][1].y
  }
  else if sample(x1,y1,directions[theta][2],field) == 1 {
   x1 += directions[theta][2].x
   y1 += directions[theta][2].y
  }
  else
   return x1,y1
 }
 return x1,y1
}
```

in Algorithm 2.

The traversal operation occurs over a maximum neighborhood size, which is also directly proportional to the receptive field size. This is receptive field size is equivalent to the tuned Gabor filter kernel size from V1. It is also true that $\forall k_E(x+i, y+j) = 1$ within the path, meaning that every entry in the path is an edge region from V1. The resulting paths from a traversal operation for a single corner is shown in Figure 4.1; the traversal operation shows all sampling paths for the given corner. Note that this path traversal operation, similar to V1 simple cells, is very computationally expensive and is only made practical with the use of the graphics processor. Algorithm 2 shows the path traversal for a single orientation; this operation is repeated twice for each Gabor orientation in the visual field.

Once all paths have been traversed, the longest path is selected as the primary path. This primary path is the key component used to dictate the feature descriptor's size and retinotopic location. Luminance values are sampled along the path as it is traversed. These values are averaged; this luminance value is another property of the winning path. The final winning path for the current cortical column is $P_{x,y}$, which is the longest path over all orientations. This is defined in Equation 4.2 and is used to modify the visual field accordingly. This visual field is then passed to V2 complex cells for the final pooling operation.

$$P_{x,y} = \arg\max_{\theta} P^{x_n,y_n}(x, y, \theta) \tag{4.2}$$

The collection of all $P_{x,y}$ values remains a retinotopic visual field and contains all information pertaining to the feature descriptor. While we currently only sample the luminosity values along the winning path, a more complex texture-based descriptor of the visual field surrounding the path could also be done using the biologically derived Grating cell operator [15]. Due to the fact that the corner regions defined in area V1 are relatively simple, an image can have anywhere between 5,000 and 200,000 corners and corresponding paths. There is a clear overabundance of information for any useful feature matching operation. To solve this, V2 complex cells are given the task of reducing the corner regions with the help of lateral inhibition.

4.1.2 V2 Complex Cells

Complex cells form a pooling operator using a neural mechanism known as lateral inhibition. Lateral inhibition is the process by which a neuron with the highest activation within a neighborhood inhibits the response of peer neurons [53]. This type of inhibition is also present in the retina [8] and is responsible for a number of interesting illusions, such as the scintillating grid illusion in Figure 4.2. The lateral inhibition mechanism is more complex for neurons in the brain, but works using a similar methodology. Each path from V2 simple cells forms a retinotopic descriptor with a given neural activation strength $k_C(x, y)$, from Equation 3.3. This neural activation is used for lateral inhibition. Paths are pooled by complex cells using their respective retinotopic locations and relative neural activation. Within a given neighborhood N, the winning path is found with Equation 4.3.

$$P_{i,j} = \arg\max_{(i,j)\in N^L} k_C(x + i, y + j) \tag{4.3}$$

(a) Original (b) Original to grayscale

(c) Path traversal (d) Path traversal in image space

Figure 4.1: Path traversal results. 8 separate paths are shown all stemming from the original feature descriptor located at the center of their (a) shows a tuning response from V1; (b) shows this converted to grayscale for simplicity; (c) shows all paths traversed by V2 Simple cells; and (d) shows these paths in the original image. The winning path for this feature, $P_{x,y}$, is the path traversing down the nose.

$P_{i,j}$ is the winning path for neighborhood N^L. The neighborhood's size is also directly proportional to the cell's receptive field size. $k_C(x+i, y+j)$ comes from Equation 3.3; it is the activation strength of the cortical column for a given corner. This has the effect of focusing on the corner with the highest activation from the neighborhood. This corner, along with the path results from V2 simple cells, form $P_{i,j}$. This is the basis for the feature descriptor extracted for that particular image region. This relatively simple lateral inhibition operation is a very common action used by various types of neurons throughout both the visual and non-visual areas of the brain. Its pooling properties are crucial for the cells in area V2 [53]. The inhibition operation works only within a limited neighborhood proportional to the receptive field of the cell. This is done to mimic the local-to-global processing theme of cells in the visual cortex.

4.1.3 Lateral Inhibition

Lateral inhibition is the key process used to extract feature descriptors with the highest level of complexity in a given retinotopic neighborhood. This complexity comes directly from the

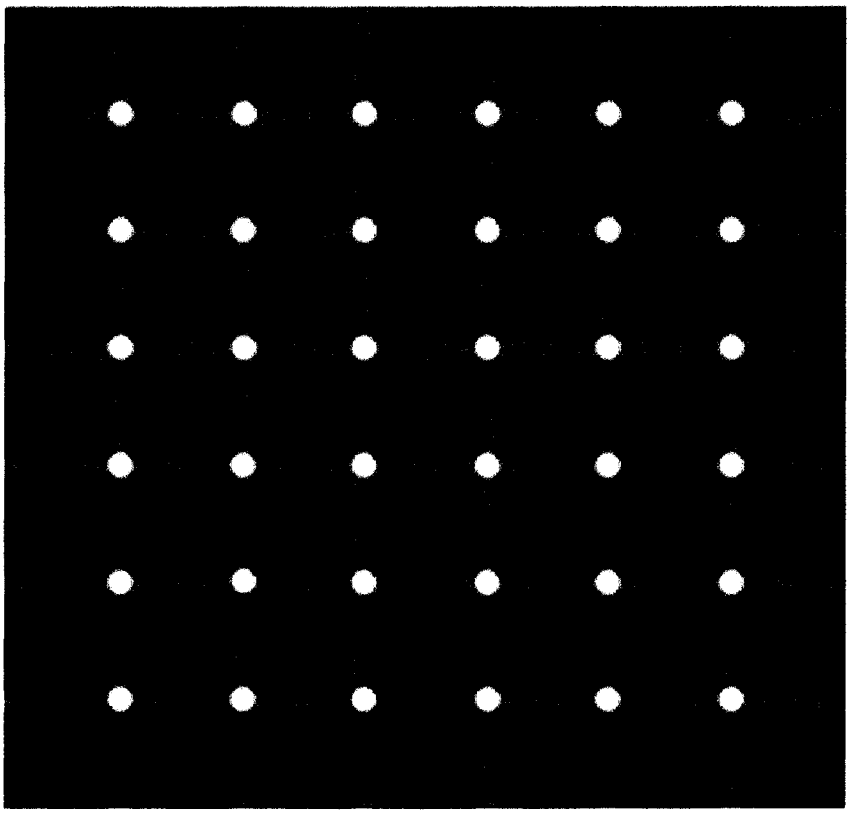


Figure 4.2: Scintillating grid illusion. This illusion illustrates lateral inhibition exhibited by the retina. In this illusion, the retinal lateral inhibition causes black dots to be perceived even though are not present in the image. Image from [83].

computation at level V1. This inherent complexity comes from both the geometric and textual properties of the object within the scene. Figure 4.3 shows the neighborhood of a visual field consisting of 25 neurons; 7 neurons within the visual field are active corner regions, each with a corresponding $k_C(x, y)$ neural activation. NC is the winning neuron and $N1$ to $N6$ are neurons whose responses will be inhibited by this winning neuron since they are within its inhibitory pool.

Each active (colored) neuron in Figure 4.3 corresponds to a corner point. Higher activation strengths are dictated by dark blue colors, lower activations are lighter blue colors, and white indicates no activation. The dotted lines show the inhibitory pool that NC will exhibit on its neighborhood. The activation values are dictated directly from the visual field computed at V1; they come specifically from the value of $k_C(x, y)$ from Equation 3.3.

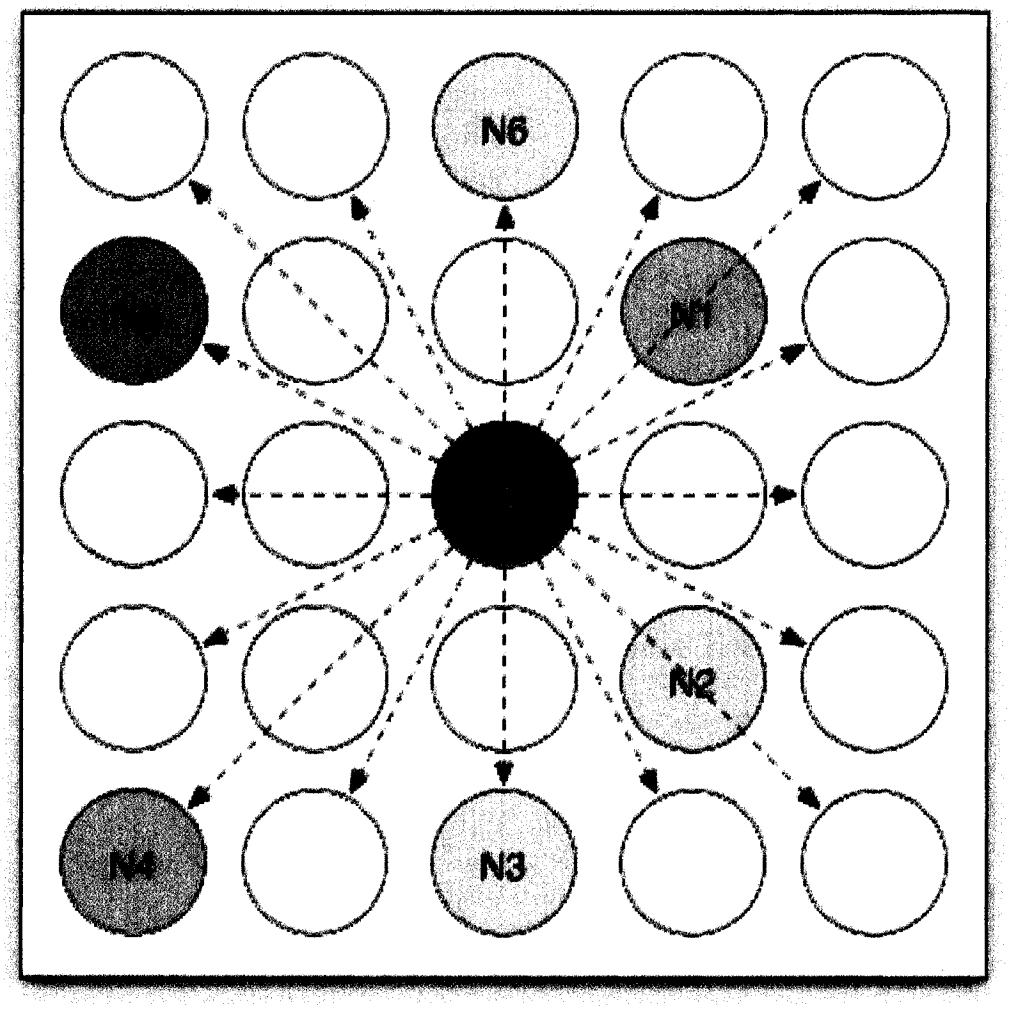


Figure 4.3: Lateral inhibition within a retinotopic neighborhood.

4.2 Feature Descriptors

Once lateral inhibition has been applied, the visual field consists of a series of features that together span all salient parts of the retinal image. The feature descriptors also have an intrinsic neural activation that allows fast feature matching to occur. At this point, feature descriptors can be extracted from the GPU for storage purposes. Each feature descriptor has a number of important properties:

- Feature descriptor location: (x, y);
- Activation strength of the feature corner's cortical column: $k_C(x, y)$;

- The primary feature path: $P_{i,j}$;
- Average luminosity values along the primary feature path;
- The feature descriptor data, represented as a quadrilateral patch from the visual field space whose size is proportional to the primary path $P_{i,j}$;
- Offsets of all other feature paths computed by V2 simple cells. This is currently not used in our model, although certainly would provide more robust matching capabilities

Note that the tuning process of V1 is necessary in order to calculate these values. We also explored the use of contrast ratios by sampling from the source image at end-points from each of the V2 simple cell paths, but this technique did not prove more effective than simply using the cortical column's activation strength.

In order to minimize texture bandwidth between the GPU and CPU, only the most significant features are extracted from the retinotopic field by using lateral inhibition. Feature importance is determined by the strength of the neural activation of the winning lateral inhibition node. This has the end result of providing a degree of rotation, scale and more complex invariance.

4.2.1 Feature Descriptor Invariance

The feature descriptors from V2 have many invariant properties due to their inherent structure. V1 simple cells, modeled with Gabor filters, are naturally invariant to small amounts of scene rotation. Our tests indicate that our feature descriptors are invariant to many types of distortion, such as rotation, scale and more complex changes. The use of multiple Gabor orientations allows for complete rotation invariance matching to be done, although this requires a somewhat more extensive search of the feature space. The human visual cortex is largely not rotation invariant, as suggested by the work done by Linden [76], where subjects are given goggles that rotate the visual field 180 degrees. The visual field remains inverted at the lowest levels, implying that recognizing an object in an irregular pose requires further processing by higher cortex levels. In order to demonstrate the invariant properties of our descriptors, we perform some simple tests.

Our task is not to perform a complete analysis of feature invariance. Instead, we simply demonstrate some basic properties of our feature descriptors that allow them to achieve the

performance outlined in the following chapter. Invariance is classified into three categories: rotation, scale and higher order invariance.

4.2.2 Rotation Invariance

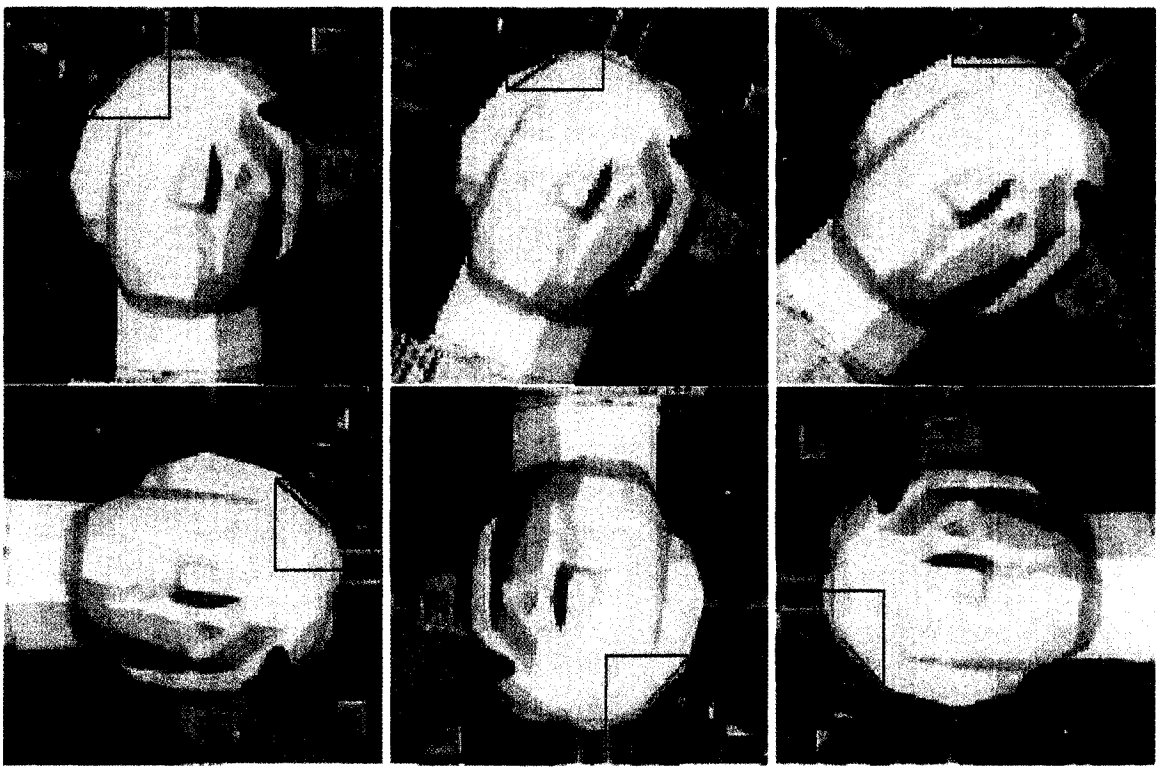


Figure 4.4: A test of feature invariance on a number of rotated images.

Figure 4.4 shows one of the more prominent features from the Tsukuba image matched across various global rotations of the image. Due to the fact that our feature descriptors traverse all orientations in their path calculation and all orientations are present for in our visual field, our features provide a high degree of rotation invariance.

One drawback of our method is that the shape of our feature descriptors is dictated by the width and height components from the offset of the primary path. This makes horizontal and vertical features more narrow than would otherwise be desired, as seen in the third example in Figure 4.4. This is overcome in matching by simply sampling all features at a common size instead of the size dictated by their primary path. We use a descriptor of size 32x32 in Figure 4.4. An ideal rotation invariant feature descriptor would sample the descriptor in a circular

manner and normalize the sampling according to the orientation of the primary feature path. This type of descriptor would also simplify scale invariance tests, but has the drawback of necessarily altering the data for the feature descriptor in order to normalize its rotation.

It is of note that the human visual cortex should not be considered to be completely rotation invariant. When subjects are given goggles that, for example, invert the visual field, it takes a length of time for the subject to adapt to these new visual settings and does not result in any changes to the lowest visual areas, such as V1 or V2 [83]. It is much more likely that recognition of an object at an unknown rotation involves much higher order search and processing than can be done at area V2 of the visual cortex. Inverting the visual field is a demonstration of neural plasticity: subjects learn to perform all of their tasks over the course of a number of days while the visual cortex rewires itself. Nonetheless, our feature descriptors use path traversal and lateral inhibition to gain properties that are conducive to rotation invariant recognition and matching.

4.2.3 Scale Invariance

Figure 4.5 shows one of the more prominent features from the Tsukuba image matched across various scales of the image. Note how the final lateral inhibition step manages to activate a similar region within each image regardless of the feature's scale. The feature descriptor size is normalized by scaling up the smaller features in order to perform the appropriate matching. Our method creates smaller feature descriptors for the smaller images, indicating that our descriptors are implicitly aware of scale.

Feature descriptor scale normalization is not done in the classification experiment in Chapter 5 due to the fact that the dataset used in that experiment tends to have a uniform object size, making scale invariance not essential for classification. While this example is in no means a complete test of the scale invariant properties of our method, it instead demonstrates how the path traversal and lateral inhibition tend to provide implicit feature descriptor scale invariance.

4.2.4 Higher Order Invariance

Figure 4.6 and Figure 4.7 show tests of the higher order invariance of the feature descriptors. We test our features on images from the Amsterdam Library of Images [74]. These sequences can be seen as a feature tracking exercise. To carry the experiment out, we select a feature

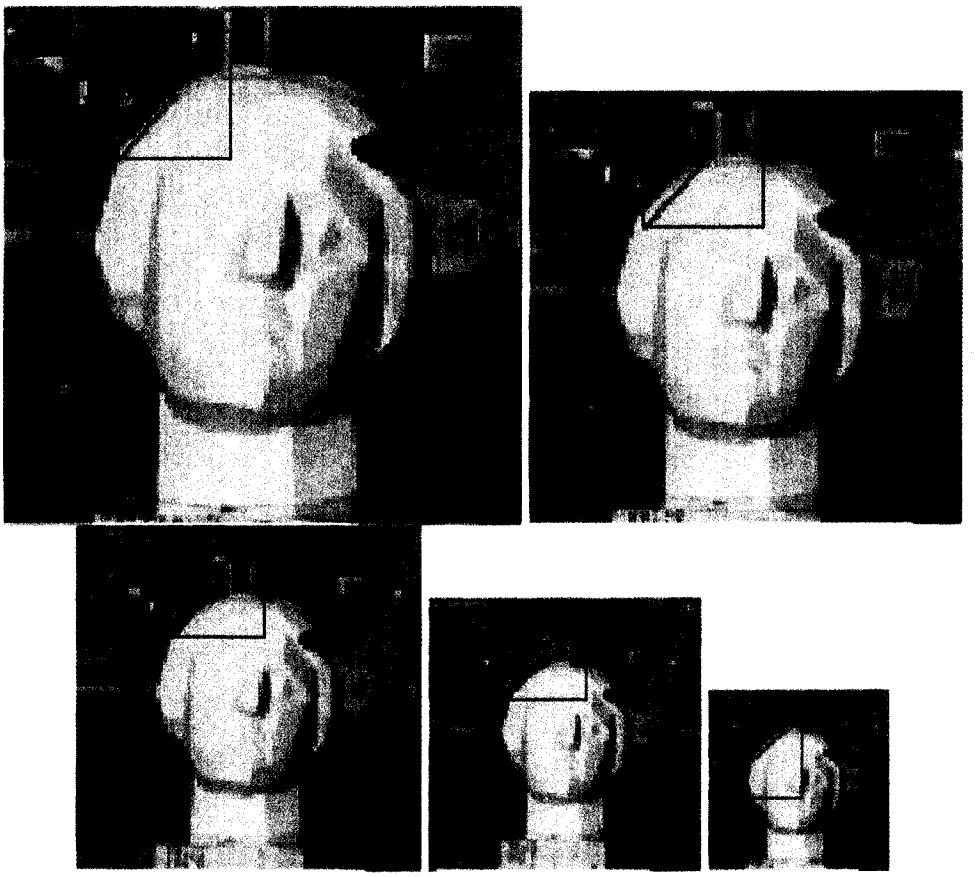


Figure 4.5: A test of feature invariance on a number of scaled images.

descriptor from one image in the sequence. The remaining images are processed and all features are matched against the query descriptors. Note that the feature descriptor comparison uses 32x32 descriptors for all matching purposes. The best matching feature descriptor is then highlighted in the sequence.

These results show that our feature descriptors are able to match well even under complex three dimensional changes to pose. The lateral inhibition process allows the more prominent feature within a region to remain the most prominent, regardless of changes to object pose. While these tests are relatively simple in comparison to real-world recognition, it serves to show that our descriptors possess a good basis for handling many forms of invariance. They are capable of handling more complex scene changes above and beyond simple rotation and scale.

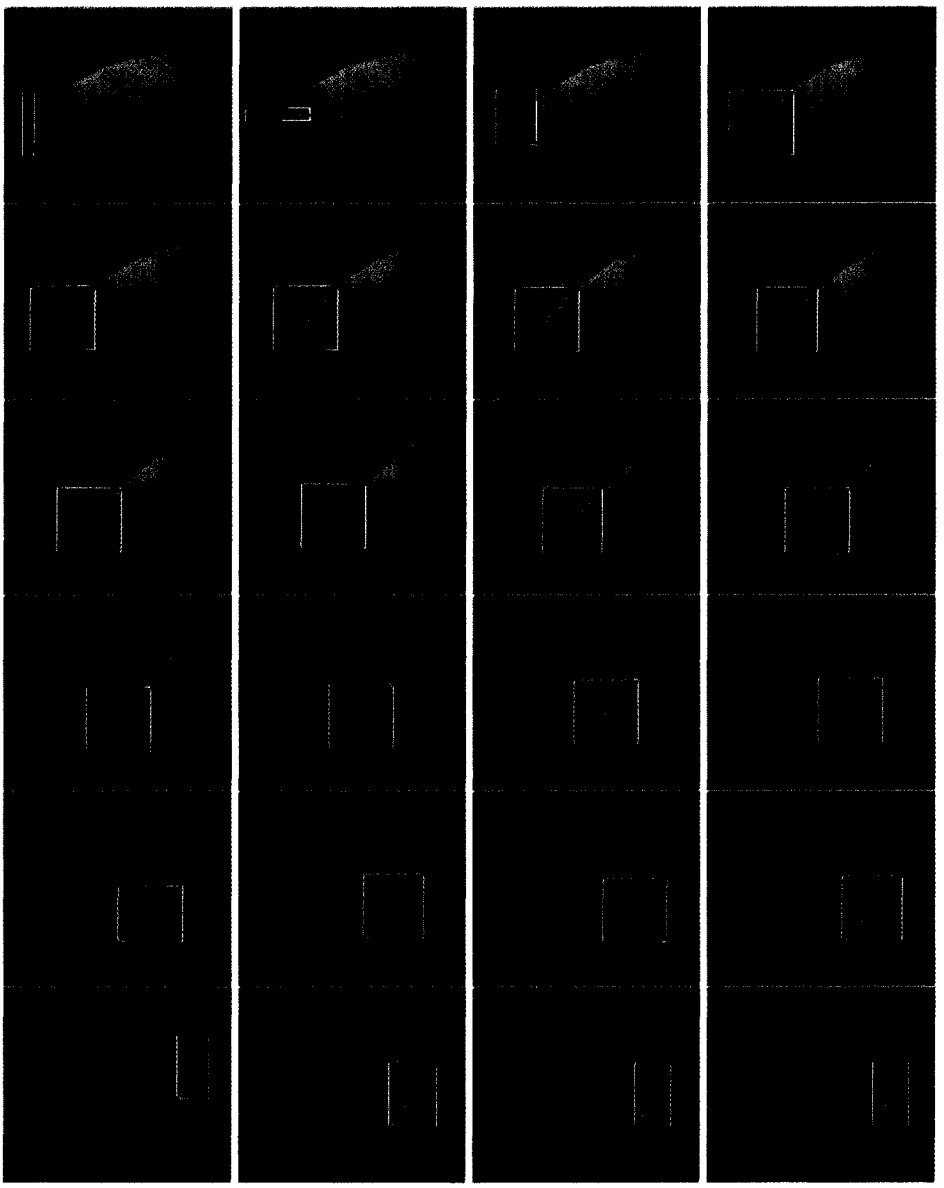


Figure 4.6: A prominent feature under complex scene transformation. The above sequence shows the most prominent feature from the scene, according to the cortical activation measure. It remains very stable under object rotation. The feature is highest activation in 10 of the 24 images (42%), second highest in 7 (29%), and fifth or better in 5 (21%). Sequence from [74].

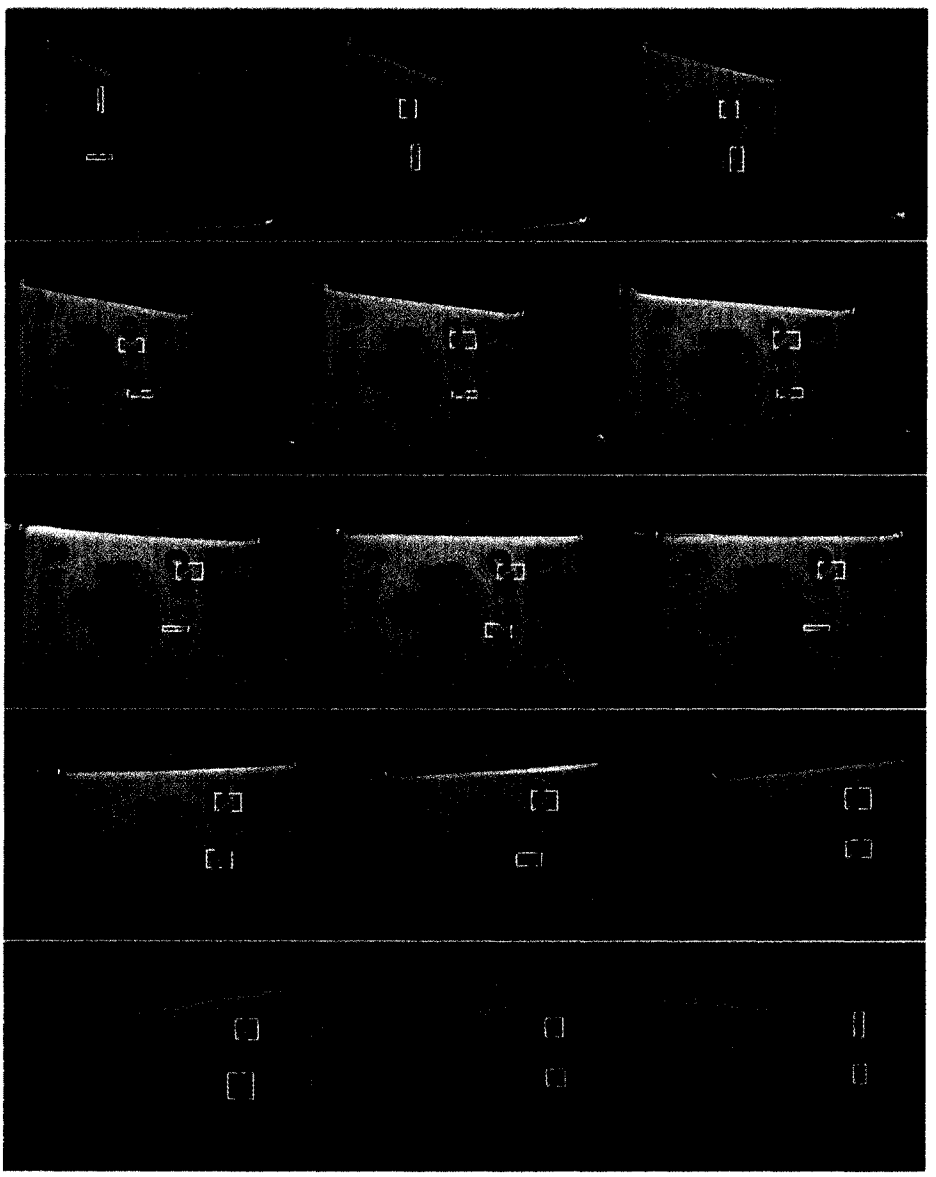


Figure 4.7: Two similar non-prominent features under complex scene transformation. This sequence shows the tracking for two similar features under object rotation. Sequence from [74].

4.3 Higher Cortex Areas

We have defined a feature descriptor with a number of invariant properties. Area V2 is not itself responsible for higher order visual functions; motion and recognition operations both occur at higher levels of the brain. Biological models must be capable of performing many of the tasks that the brain does well. Feedback loops are a known property of the visual cortex that allows higher levels of the cortex to modify the parameters of lower levels. This property has not been explored by other models and we briefly explore its functionality and implications here. We also discuss the interface V2 allows to the dorsal and ventral pathways.

4.3.1 V2 Feedback Loops to V1

In testing our model, it was discovered that paths can end abruptly when local contrast ratios change drastically. This is due to the fact that the tuning parameters selected by V1 may not tune well to all objects within a cluttered scene. In order to obtain a better resolution retinotopic visual field, the iterative tuning approach used in V1 is reapplied to a subset of the visual field with the use of a feedback loop. The region of the source image requiring further tuning is isolated and the feedback loop is applied to further process that particular region of the visual field. This simulates the process of focusing on an object in a scene. It simply uses the current cellular parameters and continues the tuning process described in Chapter 3 on the subregion in question. Figure 4.8 shows an example of a situation when a feedback loop is required and results from such an operation.

The exact cellular conditions that instigate an autonomous feedback connection are somewhat debatable; a completely automated mechanism serves as a primitive means for focusing our model's tuning on specific objects within a scene. While our model was able to classify objects with a slightly higher accuracy using feedback loops, they were not used in the final experiments due to the fact that they increase the running time by a considerable amount and did not appear to increase the classifier's accuracy. The process is likely to be more useful in the dorsal pathway for finding higher resolution object boundaries from disparity levels. Nevertheless, feedback loops are a crucial property of the visual cortex.

4.3.2 Area V3 and the Dorsal Pathway

The dorsal stream contains the first areas where cells are known to be tuned to responses within the visual fields of both eyes. Area V3 is the first area that can be considered fully eye

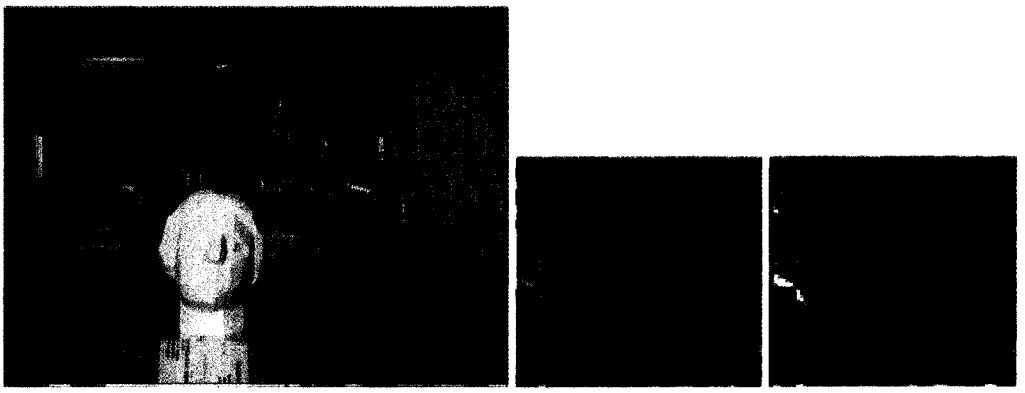


Figure 4.8: Results from tuning feedback connection. The leftmost image is the source Tsukuba image; the green box shows the region of focus where a feedback is used. This area is the region where V1 simple cells were unable to tune completely due to local image conditions. The middle image is the initial tuning results from V1 and the rightmost image is the results from a feedback loop to area V1. Images from [75].

invariant; the cells here are activated from stimulus in both eyes, instead of having a preferred eye as in V1 and V2. This eye invariance allows features to be matched between two visual fields from an identical scene. The dorsal pathway uses this information to compute disparity maps, for geometric computation, computing motion fields and optical flow. Although there is an abundance of algorithms to create accurate disparity maps between two captures of a scene, there has been much less research in biologically-inspired models that can efficiently create disparity maps and perform motion processing. This is crucial in finding optimal methods of encoding and indexing video. Computing disparity requires at least two views of a given scene. Results of our experiments in modeling the dorsal pathway using feature descriptors from V2 are given in Chapter 5.

4.3.3 Area V4 and the Ventral Pathway

Area V4 within the ventral pathway is where the cortex begins to specialize itself for object recognition. Thus, feature descriptors from V2 must also be capable of generalizing and recognizing a given class of object from a limited set of training examples. We have shown on a limited basis their invariant properties; the test of object recognition shows their true capabilities in terms of high level invariance and with respect to object recognition. The results of our ventral pathway experiments are also shown in Chapter 5.

Chapter 5

Experiments

In this chapter, we explore the robustness of our biological feature descriptors. We test their abilities with respect to object recognition and the creation of disparity maps. We also explore the speedup achieved by implementing our system on the programmable graphics processor.

5.1 Ventral Stream: Object Recognition

Our object recognition experiment is done using the Caltech 101 object dataset [72]. This dataset has become the unofficial benchmark for this task and contains 9197 images separated into 101 categories, including a background category. The dataset and ranks of the top results can obtained from [73]. We use a Support Vector Machine (SVM) classifier along with a set of feature descriptor prototypes for recognition.

5.1.1 Experiment Setup

In order to adapt our feature descriptors to be used with an SVM classifier for object recognition, we take a similar approach to other visual cortex models [39, 42]. A set of representative feature descriptors are extracted for each category in the dataset. Feature descriptors are extracted using the methods described in Chapter 4. The feature descriptors are ranked in comparison to one another and the top descriptors are used for classification. This set of top feature descriptors is effectively representative of the dataset itself. The selected features are known as prototypes. The prototype selection process is very computationally expensive and requires a graphics processor that is at least equivalent to the GeForce 8800. The prototype selection process is completely automated and does not require any specialized refining steps

that other models often use [42, 40].

A one-versus-all SVM classifier was used to carry out the experiments for the 101 object classes. For each class, N positive training images were randomly selected from the class's images; N was set to 5, 10, 15, 20, 30. All remaining images are placed in the test set. A further 50 random images were drawn from the background category for use as negative training examples. The SVM classifier used was based on libSVM [77].

5.1.1.1 Prototype Selection

A set of feature prototypes are selected for each object class, denoted C, in an automated manner. It is of note that our feature prototype selection process is very computationally expensive. The graphics processor is the main mechanism that allows this selection process to take place. Using the graphics processor, all prototypes were able to be selected from the dataset in under 4 hours. We performed some simple benchmarking which indicates that this feature prototype selection process would have taken several months using a cluster of CPUs.

Figure 5.1 shows a simplified set of features in a class containing 4 images that belong to the same category and the top 4 features per image. Algorithm 3 shows an example of the feature comparisons done to select the top feature prototypes from this category. There is a total of 16 features, namely features $a \ldots p$. From these, the top 4 features are isolated for use by the classifier.

Our prototype selection process results in a set containing the top N prototypes for each class C. We sample the prototypes from $S = 100$ images per class, or the entire set if there is less than 100 instances. Note that in Algorithm 3, we compare and rank every feature descriptor to every other feature descriptor in the class's set, giving our prototype extraction process a complexity of $O\left(CN^2S^2\right)$. Each class had the top N prototypes extracted; the prototypes from all classes were combined to form the entire prototype set. In our work $N = 64$, meaning that each class is represented by 64 prototypes. This makes this dataset's entire prototype set of size $C \times N = 6464$ features. Note that we also make use of a feature descriptor blacklist, many of which correspond to rotation artifacts within the dataset.

We note that this is a very brute-force selection process and would be difficult to scale

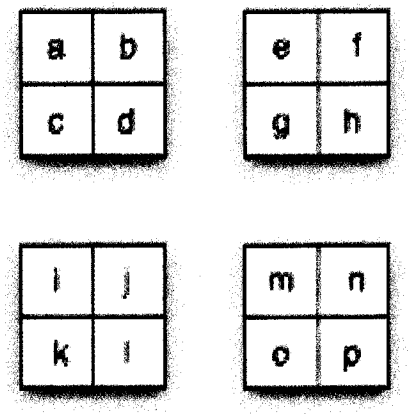


Figure 5.1: Example of 16 feature prototypes from 4 separate textures. These features are used as input to Algorithm 3.

beyond a few thousand classes. It is also important to note that SVMs, which are commonly used on this dataset, are not well suited for multi-class classification and have little biological evidence to support their use; we simply adopted our feature descriptors for use with SVM classifiers to be comparable with other such studies. This brute-force approach was used due to the fact that it is simple, computationally feasible using the graphics processor and highly effective for most categories in this dataset. A more biologically motivated classifier would be ideal for this classification, although an appropriate classifier has yet to be developed.

5.1.2 Results

Model	15 training images / cat.	30 training images / cat.
Serre et al.[39]	33	41
Holub et al. [78]	35	
Grauman et al. [81]	49	58
Mutch [42]	51	56
Zhang et al. [82]	**59**	**66**
Our Model	46	59

Table 5.1: Our results for the Caltech 101 object dataset along with results from previous studies. Scores shown are the average of 10 independent runs and are the average of the per-category classification results.

Rank	Category	Rate	σ		Rank	Category	Rate	σ
1	car_side	92.7	2.3		26	minaret	66.0	3.7
2	grand_piano	83.1	7.0		27	lotus	65.1	8.8
3	motorbikes	81.5	4.7		28	sunflower	63.9	5.3
4	dollar_bill	81.3	4.9		29	saxophone	63.0	12.2
5	trilobite	81.2	4.2		30	llama	62.3	5.1
6	crocodile	79.2	10.5		31	helicopter	62.2	5.0
7	hawksbill	77.7	7.3		32	beaver	62.0	7.2
8	dalmatian	77.3	4.0		33	binocular	61.5	12.1
9	pagoda	77.1	9.8		34	brontosaurus	61.3	13.5
10	cougar_face	76.2	7.9		35	camera	60.5	8.3
11	emu	75.9	7.5		36	euphonium	60.5	7.8
12	faces_easy	74.2	2.8		37	stop_sign	60.2	5.4
13	kangaroo	74.0	5.6		38	revolver	60.1	7.3
14	airplanes	73.8	5.2		39	ferry	59.8	7.4
15	lobster	73.1	10.0		40	windsor_chair	59.5	6.2
16	dragonfly	72.8	3.9		41	dolphin	59.2	4.2
17	wheelchair	71.6	6.4		42	elephant	59.1	6.8
18	flamingo_head	70.4	6.8		43	butterfly	59.0	6.4
19	electric_guitar	70.4	6.5		44	mayfly	58.3	8.3
20	leopards	70.2	5.5		45	crab	57.5	3.3
21	headphone	69.8	13.4		46	stapler	57.2	4.9
22	stegosaurus	68.4	6.0		47	accordion	56.8	9.8
23	buddha	68.3	7.3		48	hedgehog	56.7	12.8
24	crocodile_head	67.0	7.6		49	cannon	55.7	10.3
25	sea_horse	66.2	7.3		50	gramophone	55.7	8.3

Table 5.2: Results for the top 50 categories using 30 training images. Results are averaged over 10 runs. Only classes with at least 10 test images are shown.

Algorithm 3 Prototype selection process

```
function selectPrototypes(N,a...f) {
  // cmp() returns a score 0..1 based on feature similarity
  scores[a] = max(cmp(a,e),cmp(a,f),cmp(a,g),cmp(a,h))
  scores[a] += max(cmp(a,i),cmp(a,j),cmp(a,k),cmp(a,l))
  scores[a] += max(cmp(a,m),cmp(a,n),cmp(a,o),cmp(a,p))
  ...
  scores[p] = max(cmp(p,a),cmp(p,b),cmp(p,c),cmp(p,d))
  scores[p] += max(cmp(p,e),cmp(p,f),cmp(p,g),cmp(p,h))
  scores[p] += max(cmp(p,i),cmp(p,j),cmp(p,k),cmp(p,l))
  features = sort(scores)

  // the features with the top scores are returned
  return topNFeatures(N,features)
}
```

5.1.3 Analysis

The object recognition accuracy obtained with our approach achieves high classification. These results can be seen in Figure 5.2 and Table 5.1. Table 5.2 shows the accuracy of the top 50 categories along with their standard deviations. These results show that our model has excellent object recognition capabilities.

The accuracy obtained by our approach stems from two main sources: our feature descriptors and the prototype database. As explained in Chapter 4, our feature descriptors display a high degree of invariance. While our tests in this chapter show that our feature descriptors are suitable for recognizing a previously seen object placed in a new pose, our object recognition results here show that the descriptors are highly effective at capturing the very structural details of a given object. A specific feature descriptor is able to match different objects of the same class based on the structural and textural similarities between two given instances of the class. Furthermore, our lateral inhibition process is able to selectively activate feature descriptors on similar regions on other instances within the object category.

For example, within the kangaroo class, there are a number of prototype feature descriptors that are focused around the head and ears of a given kangaroo. Our lateral inhibition

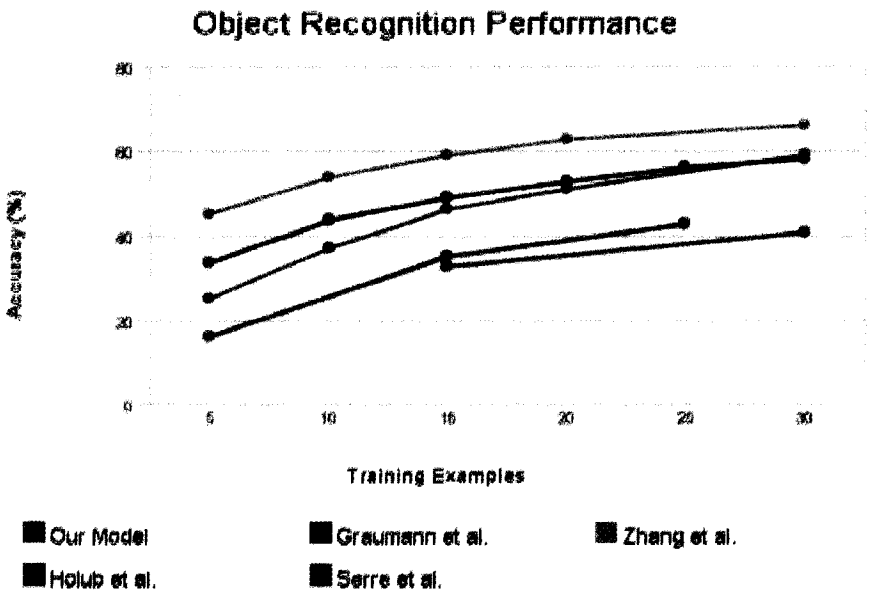


Figure 5.2: Our results for the Caltech 101 along with results from previous studies

process helps to center the destination feature on a similar region of the kangaroo head in each instance. The collection of "kangaroo head feature descriptors" has high activation on most instances from that category, but does not activate nearly as well for any other stimulus. This can be likened to a neuron that identifies kangaroo heads, although our process does require a number of similar such feature descriptors to provide the appropriate coverage. This object-specific selectivity occurs for nearly all of the object categories in this dataset.

The other key aspect of our model that affords this level of accuracy is our prototype database. Other visual cortex models have been very limited in terms of how they select prototypes for classification. It is of note that our model gives a very uniform recognition rate for most object classes. This is very likely due to the fact that our prototype selection process gives every object class a representative set of feature descriptors within the total prototype set. This method does not explicitly favor any given class, although certain classes were found to be less responsive to the prototype selection process. The primary factor that allowed us to select such a successful set of feature descriptor prototypes was the exhaustive search we performed on the dataset using the graphics processor. Since no other model has made use of such a vastly superior computation platform, it is of little surprise that our brute-force method is able to outperform other biological approaches.

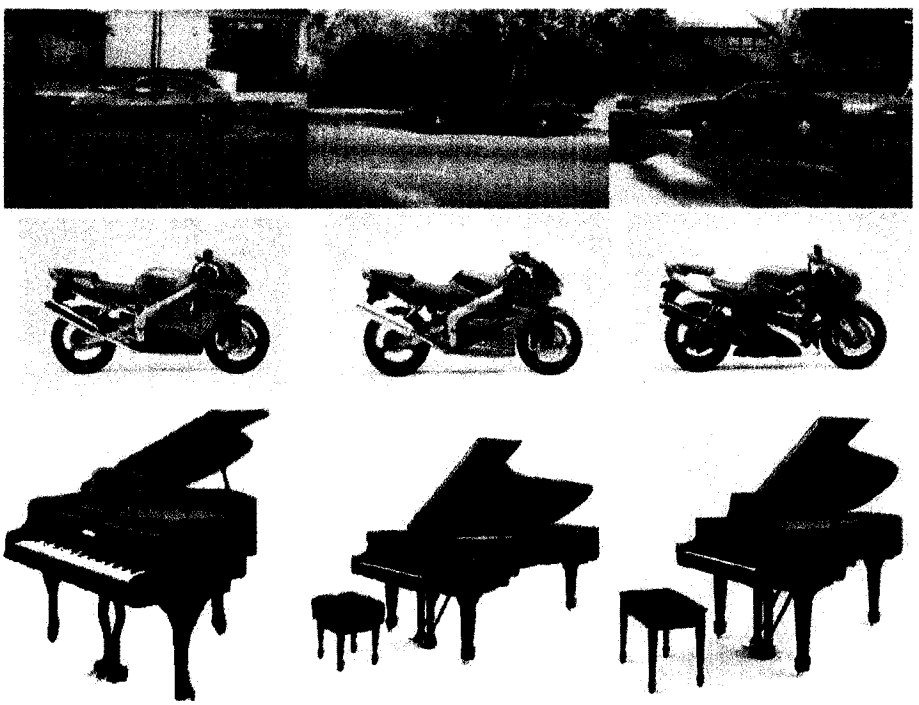


Figure 5.3: Samples from categories our system classifies more easily

Figure 5.3 shows examples of easy categories and Figure 5.4 shows examples of the more difficult categories.

The lower accuracy categories are ones that did not respond well to the feature descriptor prototype selection process. We tested the theory that our prototype selection process plays a significant role in the classifier's accuracy by performing another very simple experiment with the lowest accuracy category from above. For this, we hand-selected 64 feature prototypes. With hand-selected prototypes, this category's classification rate went from 38.1% to 92.3%. This confirms the work done by Mutch [42] which showed that an appropriate feature refining method can help to vastly improve classifier accuracy. This is clearly not a good solution to the problem, since hand-selecting prototypes is incredibly time consuming. But, it does serve to show that the method used to select feature descriptor prototypes plays very large role in classifier accuracy. This process seems to be nearly as important as the design of the feature descriptors themselves.

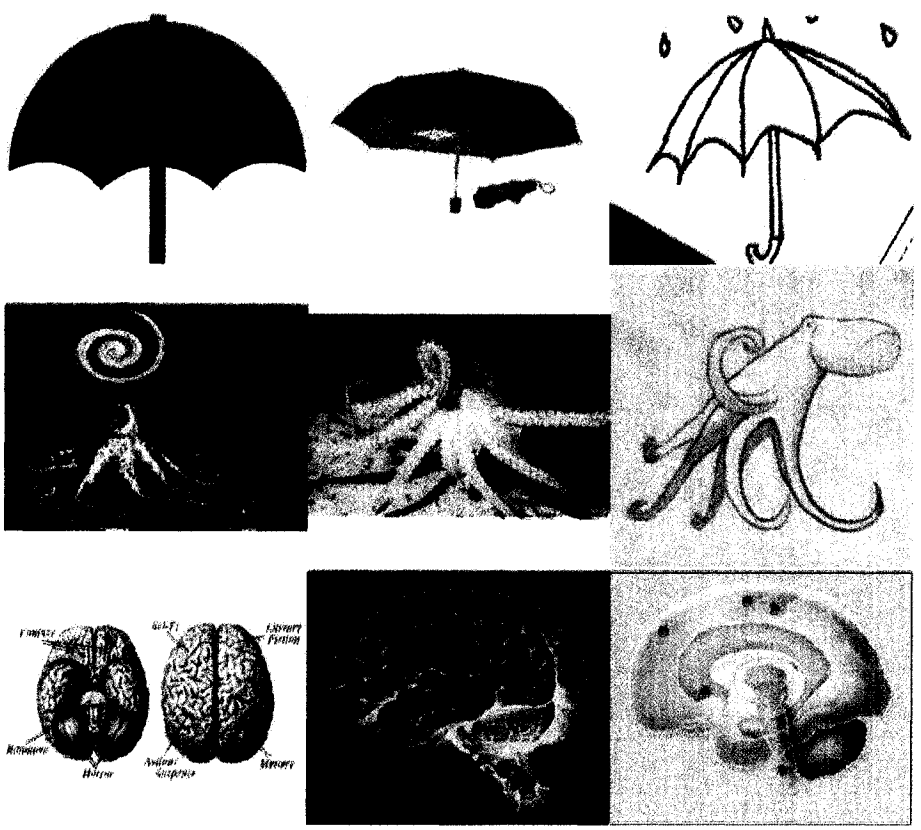


Figure 5.4: Samples from more difficult categories

5.2 Dorsal Stream: Disparity and Motion

An effective biological feature descriptor must be capable of modeling the functionality present in both pathways of the visual cortex. We apply our feature descriptors to creating disparity maps, which are one of the operations that occurs in the dorsal stream. The dorsal stream of the visual cortex is responsible for all motion based computation. Area V3a, one of the lower levels of the dorsal stream is thought to be responsible for creating a global disparity map using the input from both eyes [21]. This is the second area in the ventral stream that performs processing after V3, thus our approach is a hybrid between disparity in V3a and motion in V3.

The receptive fields of the cells within V3 are considerably larger than those in area V2 [46]. This means that visual acuity is diminished in comparison to V1 and V2, making this area of the visual cortex less able to analyze fine visual details with the visual field. While there are still unanswered questions as to the specific details of disparity calculation in the human visual cortex, the dorsal stream is widely thought to be responsible for global disparity. There is evidence to suggest that finer grain local disparity is done with the help of the ventral stream. Our work serves to model the global disparity present in the dorsal stream, which is closely related to motion processing. At higher levels in the dorsal stream, motion processing eventually becomes a visual field based on optical flow [51].

Our disparity map is a basic retinotopic visual field that is derived from the visual field generated by area V2. The disparity map in this area of the brain cannot be of high visual acuity; it can be likened to a hybrid between a global disparity field and a basic motion field. A motion field is outside the scope of our work, thus our results are presented as a disparity field. We model the visual fields from the left and right eye by using image sequences with dual captures taken from distinct viewpoints. We use the Middlebury stereo dataset [75].

5.2.1 Experiment Setup

A disparity map for the current visual field is effectively a manner of combining the visual fields from both the left and right eyes. The only parameters changed within the model is that the lateral inhibition is lowered to give a higher feature descriptor count. The disparity field is computed with the following steps:

1. Tune and extract all features from the left view.

2. Use the tuned parameters obtained from the left view to extract all features from the right view.

3. Match each left feature to the best corresponding right feature by comparing the feature descriptor patch. This matching is similar to the method used in the ventral stream, except that feature descriptors are larger.

4. Generate a displacement value between each left and right feature by calculating the feature's distance traveled.

5. Propagate the displacement along the primary feature descriptor path.

6. For each pixel that does not have a displacement value, assign it a value of the average three closest feature descriptor displacements.

7. Normalize the disparity map to give the final results.

We compare our results against the leading disparity map algorithms in the field. A bug in the GPU driver hindered our ability to perform disparity estimations on the remaining images in the dataset, thus only a subset of the typical images analyzed are used.

5.2.2 Feature Descriptors

The feature descriptors used for the dorsal stream are identical to the feature descriptors (prototypes patches) used in Section 5.1, with the exception of using a smaller lateral inhibition factor. Lowering the lateral inhibition has the effect of generating a larger number of feature descriptors for each image. It also increases feature descriptor overlap. Feature descriptor size was also set to a static value.

5.2.3 Results

5.2.4 Analysis

Our approach gives disparity maps with excellent continuity. The technique we use also generates disparity maps very quickly: the running time of our approach is less than 10 ms once features have been extracted. The use of a larger feature descriptor gives near-perfect disparity estimates within an object's boundaries, but gives poorer results at object borders. This is very similar to how motion processing occurs in the visual cortex. While our method does have

		Tsukuba			Venus		
Algorithm	Rank	nonocc	all	disc	nonocc	all	disc
Our Method	-	3.61	4.27	15.32	2.48	3.10	17.32
TensorVoting [84]	19.3	3.79	4.79	8.86	1.23	1.88	11.50
RealTimeGPU [85]	19.7	2.05	4.22	10.6	1.92	2.98	20.30
GenModel [86]	19.7	2.57	4.74	13.0	1.72	3.08	16.90

Table 5.3: Comparison of our ventral stream approach to other algorithms. Due to a GPU driver bug, we cannot evaluate the Teddy or Cone images.

problems at object edges, it is easily able to accurately discern object depths and keeps a correct general shape of each object.

Our results are very closely aligned to what is known about how data is processing in the dorsal stream of the visual cortex. The receptive fields of cells in the Middle Temporal motion area MT (V5) within the dorsal stream have receptive fields that are approximately 10 times larger than in area V1 [50]. Due to their exceptionally large receptive fields, they have very poor visual acuity. While they are exceptional at determining motion direction, these cells are poorly suited to recognizing an object.

Disparity maps in the human visual cortex have much better visual acuity than our results show, although the dorsal stream is thought to be responsible for lower acuity disparity that is more global in nature. A more accurate disparity map can be obtained with a multi-pass feature descriptor matching process. The use of square feature descriptors also makes it less able to handle objects that are thinner, such as the lamp arm in the Tsukuba images. Many of the leading disparity map techniques use a variant of the Belief Propagation algorithm. While there is no evidence to suggest that Belief Propagation is biologically plausible, using a hybrid approach would likely improve the results considerably.

While our initial results are promising, the dorsal stream is clearly much more complex than what is presented here. Our results can be likened to a hybrid of the motion and disparity calculation process within the visual field: it does not have a strong grasp of specific object contours, but does approximate motion direction very accurately.

We suggest that any truly biologically plausible feature descriptor extraction method or vi-

sual processing technique must necessarily show good results for both recognition and motion-based tasks. It is important to consider both pathways of the visual cortex when attempting to model its functionality. Clearly more work is required in this area.

5.3 GPU and CPU Timing Results

This section is designed to show the timing advantages of using the GPU for neural operations over the CPU. This is done by comparing the execution of V1 simple cells in the Single Instruction Multiple Data (SIMD) environment on the GPU to the Single Instruction Single Data (SISD) environment on the CPU. We also show the performance increase between two generations of graphics processors. It is of note that there is an overhead present in transferring any data to the GPU; these results to not take this overhead into account. Note that these results are limited only to the speedup achieved by the simple cells of cortex area V1. The other higher complexity areas in our model clearly benefit from the SIMD environment present on the GPU in a similar manner. The OpenGL 2.0 GLSL pixel shader source code for the GPU timing operations is given in Appendix I.

5.3.1 Experiment Setup

5.3.1.1 CPU versus GPU

We perform a number of experiments. Each experiment has an identical setup. On the GPU, the following steps are performed:

1. The filter kernels (sizes 7x7 to 31x31) and source images (sizes 512x512 to 4096x4096) are uploaded to the GPU.
2. The V1 simple cell pixel shader is applied; 4 filter kernel orientations done per pass.
3. This is done 10,000 times with each kernel and with each source image, for a total of 40,000 visual fields processed

Note that since the GPU executes in parallel with the CPU, it is important to call the appropriate flush command when timing operations are run on the GPU to ensure that the execution has been completed. On the CPU, the following steps are performed:

1. The identical V1 simple cell operation is applied on the CPU.

2. This is done 10 times at each of the 4 orientations for a total of 40 runs per source image and per filter kernel; the timing results are scaled up accordingly.

The GPU used for this benchmark was a GeForce 8800 GTS and the CPU was an AMD 64 X2 3800 at 2.2 GHz.

5.3.1.2 GPU versus GPU

Further timing is done between two separate Graphical Processing Units. For this timing, each GPU is given the task of performing the identical operation as above, but only on the 4096x4096 texture. The details of the processors used are as follows:

1. GeForce 8800 GTS: 96 shader pipelines, 1200 MHz shader clock speed, 1600 MHz memory clock speed.

2. GeForce 6800 GTS: 16 shader pipelines, 500 MHz shader clock speed, 1000 MHz memory clock speed.

Note that the vertex shader for this operation does not perform any tasks outside of its default requirements. The results from these timing experiment are shown in Figures 5.7, 5.8 and 5.9.

5.3.2 Results

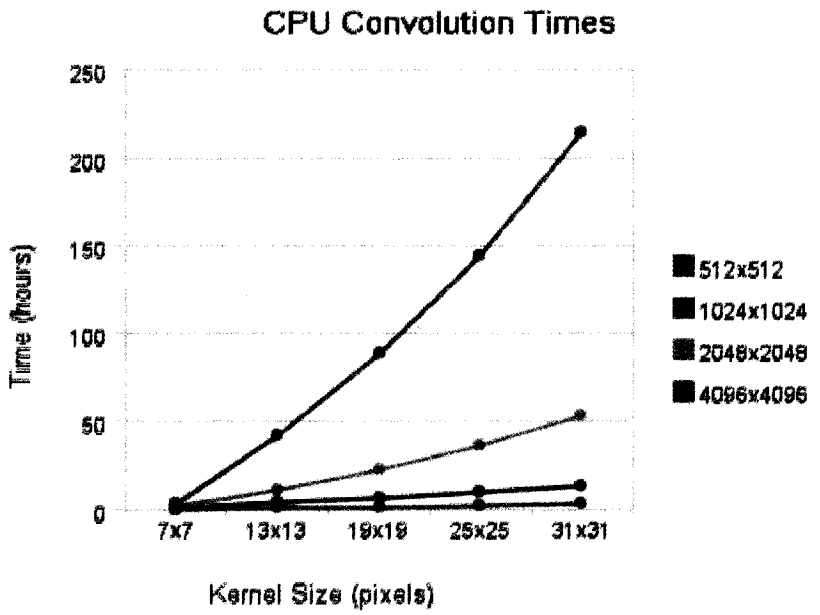


Figure 5.7: CPU Timing Results

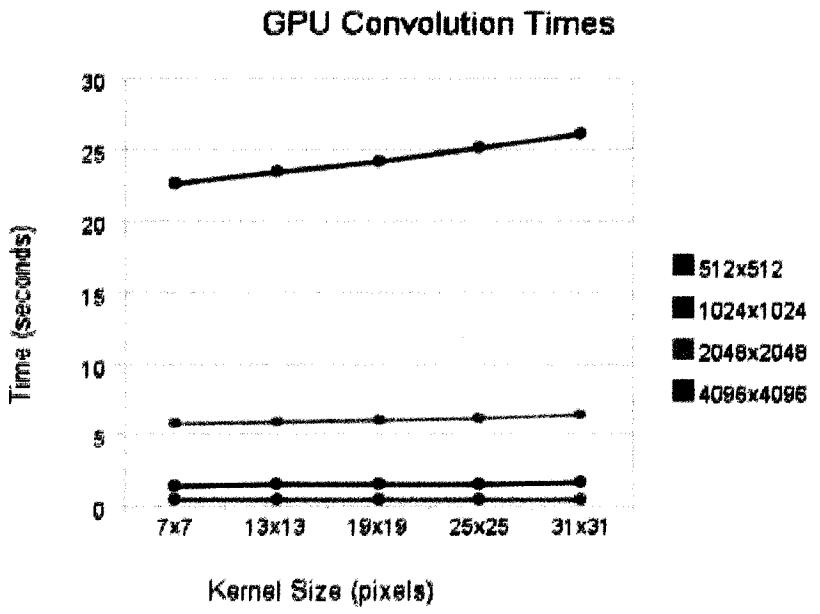


Figure 5.8: GPU Timing Results

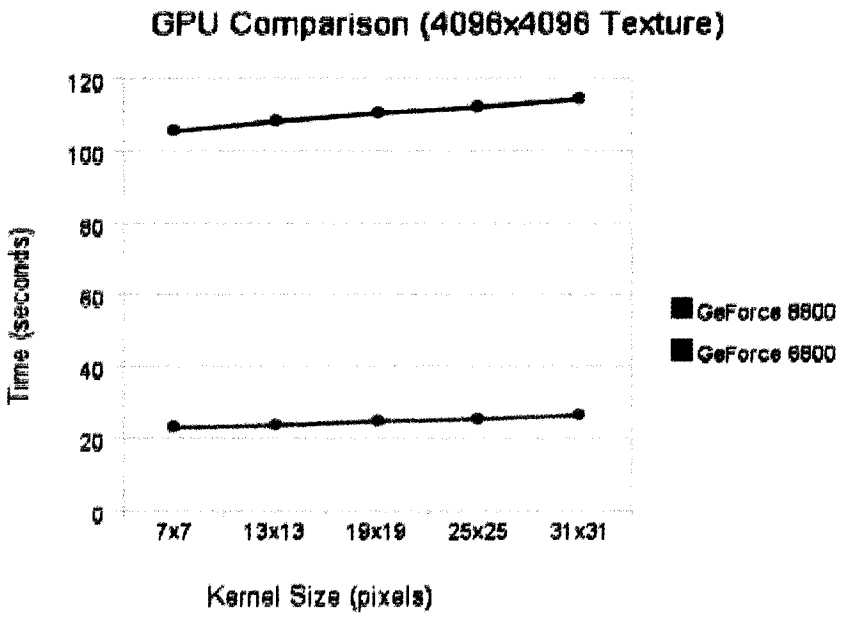


Figure 5.9: Comparison of GPU Timing Results

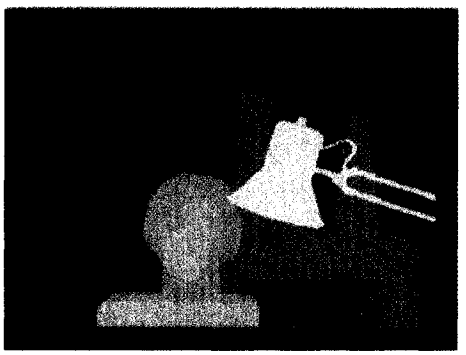
5.3.3 Analysis

Our timing results show that simple cells executed on the GPU are between 1,000 and 30,000 times faster than using the CPU. These results alone suggest that all biologically motivated vision research can be greatly benefited with the use of the graphics processor. The exceptional speedup that our technique gives has allowed us to perform very computationally intensive feature descriptor matching operations for object recognition. These operations would clearly have been computationally intractable using the CPU. We also note that modern graphics processors are accelerating at an unprecedented rate. The demonstrated speedup appear to contradict Moore's Law; although it is unlikely that this is the case. The processing speedup shown is likely due to changes in the structure and level of parallelism present in the modern GPU.

The difference in processing speed between the two graphics processors is noteworthy. This speedup is likely due to an increase in the number of stream processors between the two models. The GeForce 8800 used for these benchmarks has 96 stream processors, compared to 16 shader processors used in the GeForce 6800. The shader clock also increased from 500 MHz to 1200 MHz and the Memory clock increased from 1000 MHz to 1600 MHz. These factors all combine to give the appearance that the processing increment contradicts Moore's law; in fact, the per-processor speed increase and memory clock increase are still within the

appropriate margins. Theses two factors, combined with a 6-fold increase in stream processors clearly contributes to the speedup demonstrated.

(a)



(b)

(c)

Figure 5.5: Disparity Results for Tsukuba. The top image is the original, the middle image is the baseline and the bottom shows our result using 32x32 descriptors. Image from [75].

(a)

(b)

(c)

Figure 5.6: Disparity Results for Venus. The top image is the original, the middle image is the baseline and the bottom shows our result using 32x32 descriptors. Image from [75].

Chapter 6

Conclusion

We have presented a biologically-inspired method for processing visual data and extracting feature descriptors. Our model is implemented on the highly parallel programmable graphics processor. We have used our feature descriptors to model problems from both pathways in the human visual cortex. Our descriptors are extracted with a unique cellular tuning process that selects the most appropriate cells with which to process a given visual field.

We have shown that our biologically derived feature descriptors give high accuracy on a difficult object recognition dataset. We have shown that our feature descriptors are also suitable for computing disparity maps. Our disparity map creation process yields a result that is very close to what would be expected for the initial motion processing area in the brain. We have analyzed our feature descriptors in terms of invariance and have found that they are robust even under complex scene transformations.

Our model uses the programmable graphics processor, which affords a significant speedup for our biologically-inspired algorithms. Our feature extraction and comparison process is under 100 ms per image, showing a clear convergence towards real-time object recognition in computer systems. We have demonstrated that the parallelism present in the programmable graphics processor provides a much more natural platform for modeling the brain and is able to process images at a much higher speed than would otherwise be possible.

Our results show that the field of biological motivated research serves to gain a great deal with the use of programmable graphics processors. We have also demonstrated that computer vision stands to make significant advances by looking towards biology for inspiration. Our

results show that many algorithms in computer vision may gain by adapting algorithms to make use of the programmable graphics processors.

6.1 Future Work

Our work has a great deal of applicability in a number of fields, all of which have a biological equivalent within the human brain. Our object recognition results, combined with the high speeds we have achieved by using programmable graphics processors, is noteworthy. This provides an ideal basis for a large-scale image classification engine, although Support Vector Machines would not likely be the best choice for this type of problem. Instead, a biologically-derived method of indexing and storing images would be required. Ideally, it would mimic the brain by placing similar images in similar regions within its database structure. Suitable candidates for data storage involve techniques such as Self Organizing Maps [66] or Evolving Trees [67]. Our work is also highly suitable for any number of specific object recognition problems, such as face recognition and biometrics, most of which would require some manner of localization to be done.

Since our feature descriptors tend to activate very selectively and for specific objects, they would be an ideal candidate for a query-by-example (QBE) system that takes an image as input and ranks a series of images to give an ordered list of the most similar images. This has a number of applications, but could really only be made useful within the context of a large feature database.

An obvious extension of this work would be to include some form of color processing, although the details of color processing within the brain does require further fMRI study prior to implementation. Since we have demonstrated that color is not necessary for high accuracy object recognition, it is not entirely clear what use color data would have in our system, other than effectively being another form of meta-data.

Our work in disparity and motion makes an ideal framework for investigating a number of related problems. One such possibility is geometric projection for super-resolution, since there it is very likely that the brain does a primitive form of super-resolution when combining the visual fields from both eyes. This would involve creating higher accuracy local disparity maps, as in area V3a in the brain. Research into other motion-based processing may prove suitable for video encoding and indexing. Biologically derived video encoding techniques may pro-

vide an ideal basis for low bandwidth methods of coding video that minimize the loss in video quality. Biologically driven techniques may prove to minimize the perceptible loss in quality due to its inherent relationship with the human visual cortex.

Another highly suitable biological extension would be to implement texture classification capabilities from the grating cell operator [15, 14]. This would also serve to help in image segmentation when two views of a given object are not available. This would very likely prove useful for certain properties of object recognition and also has highly applicability to motion based processing. Our investigation of feedback loops within the brain would quite likely prove a crucial aspect in a number of problems such as motion, disparity, texture segmentation, super-resolution and video processing mentioned above.

In terms of further modeling more primitive areas of the brain, developing audio recognition on the graphics processor may prove to be an interesting possibility. This would involve modeling interactions between the visual and auditory regions of the brain at the thalamus level, which is a very primitive area in the brain. This approach would be suitable for robotics research and would be aimed at giving an autonomous system the ability to both see and hear its surroundings. Of course, the power requirements involved in having a mobile graphics processor would be a non-trivial challenge to overcome.

Chapter 7

Appendix I: V1 Simple Cell Shader

Below is the OpenGL 2.0 GLSL source code for the V1 simple cell pixel shader convolution operation. This represents the simple cells in area V1. Note that there are two input textures: the kernel texture and source texture. The source texture has been converted to grayscale prior to use. Note that this shader performs four convolutions per pass, each orientation is stored in a separate RGBA texture component in the destination.

```
uniform sampler2D sourceTex;
uniform sampler2D kernelTex;
uniform float width;
uniform float height;
uniform float kernelSize;
const float stepW = 1.0/width;
const float stepH = 1.0/height;
const float kernStep = 1.0/kernelSize;
const float halfKern = (kernelSize-1.0)/2.0;
void main(void)
{
  float i = 0.0;
  float j = 0.0;
  vec2 srcLoc = vec2(0.0);
  vec2 kernLoc = vec2(0.0);
  vec4 sum = vec4(0.0);
  vec4 kernelValue = vec4(0.0);
  vec4 sourceValue = vec4(0.0);
```

```
  if ( gl_TexCoord[0].s < (halfKern*stepW) ||
       gl_TexCoord[0].t < (halfKern*stepH) ||
       gl_TexCoord[0].s >= ((width-halfKern)*stepW) ||
       gl_TexCoord[0].t >= ((height-halfKern)*stepH) )
  {
    // skip edge pixel
  }
  else
  {
    // perform convolution
    for( i=-halfKern; i<=halfKern; i+=1.0 )
    {
      for( j=-halfKern; j<=halfKern; j+=1.0 )
      {
        srcLoc=gl_TexCoord[0].st + vec2(i*stepW,j*stepH);
        kernLoc=vec2( (i+halfKern) * kernStep,
                      (j+halfKern) * kernStep);
        sourceValue = texture2D(sourceTex, srcLoc);
        kernelValue = texture2D(kernelTex, kernLoc);
        sum += sourceValue.r * ((kernelValue*2.0)-1.0);
      }
    }
  }
  // output final value to current pixel
  gl_FragColor = vec4(sum.a,sum.r,sum.g,sum.b);
}
```

Bibliography

[1] D. Gabor. Theory of communication. Journal of IEE, 93:429-459, 1946.

[2] S. Zeki. A century of cerebral achromatopsia. Brain 1990;113:1721-77.

[3] P. L. Williams, L. H. Bannister, M. M. Berry, P. Collins, M. Dyson, J.E. Dussec, M. W. J. Ferguson. Gray's Anatomy. 38th ed, New York, Churchill Livingstone, 1995.

[4] H. D. Lu and A. W. Roe. Optical Imaging of Contrast Response in Macaque Monkey V1 and V2. Cerebral Cortex, 10.1093/cercor/bhl177, 2007.

[5] J. B. Levitt, D. C. Kiper and J. A. Movshon. Receptive fields and functional architecture of macaque V2. Journal of Neurophysiology, 71(6):2517-2542, 1994.

[6] H. D. Lu, M. Kraus and A. W. Roe. Optical Imaging of Contrast Response in functional domains in V1 and V2 of macaque visual cortex. Journal of Vision, 4(8):275, 2004.

[7] G. M. Boynton and J. Hegdé. Visual Cortex: The Continuing Puzzle of Area V2. Current Biology, Vol 14, 523-524, 2004.

[8] P. B. Cook and J. S. McReynolds. Lateral inhibition in the inner retina is important for spatial tuning of ganglion cells. Nature Neuroscience, 1(8):714-719, 1998.

[9] A. Anzai, D.C. Van Essen, X. Peng and J. Hegdé. Receptive field structure of monkey V2 neurons for encoding orientation contrast. Journal of Vision, Vol 2, 221a, 2002.

[10] J. Hegdé and D.C. Van Essen. Selectivity for complex shapes in primate visual area V2. Journal of Neuroscience, Vol 20, 61-66, 2000.

[11] J. Hegdé and D.C. Van Essen. Strategies of shape representation in macaque visual area V2. Visual Neuroscience, Vol 20, 313-328, 2003.

[12] M. Ito and H. Komatsu Representation of angles embedded within contour stimuli in area V2 of macaque monkeys. Journal of Neuroscience, Vol 24, 3313-3324, 2004.

[13] R. von der Heydt, E. Peterhans and M. Dürsteler. Periodic pattern-selective cells in monkey visual cortex. Journal of Neuroscience, 12:1416-1434, 1992.

[14] S.E. Grigorescu, N. Petkov and P. Kruizinga. Comparison of texture features based on Gabor filters. IEEE Trans. on Image Processing, 11:1160-1167, 2002.

[15] N. Petkov and P. Kruizinga. Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells. Biological Cybernetics, 76(2):83-96, 1997.

[16] P. Kruizinga and N. Petkov. Grating cell operator features for oriented texture segmentation. In proc. 14th International Conference on Pattern Recognition, 1010-1014, 1998.

[17] R. Lim, M. J. T. Reinders, and Thiang. Facial Landmark Detection using a Gabor Filter Representation and a Genetic Search Algorithm. In Proc, Seminar of Intelligent Technology and Its Applications, 2000.

[18] T. Ro, B. Breitmeyer, P. Burton, N. S. Singhal and D. Lane. Feedback Contributions to Visual Awareness in Human Occipital Cortex. Current Biology, 11:1038-1041, 2003.

[19] S.J. Luck, L. Chelazzi, S. A. Hillyard, and R. Desimone. Neural Mechanisms of Spatial Selective Attention in Areas V1, V2, and V4 of Macaque Visual Cortex. Journal of Neurophysiology, Vol 77, 1:24-42, 1997.

[20] J. P. Jones and L. A. Palmer. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysilogy, 58:1233-1258, 1987.

[21] K. Grill-Spector and R. Malach. The Human Visual Cortex. Annual Review of Neuroscience, 27:649-77, 2004.

[22] J. D. Forte, M. Hashemi-Nezhad, W. J. Dobbie, B. Dreher and P. R. Martin. Spatial coding and response redundancy in parallel visual pathways of the marmoset Callithrix jacchus. Visual Neuroscience, 22(4):479-491, 2005.

[23] E. Grossman, R. Blake. Brain areas active during visual perception of biological motion. Neuron, 35:1167-1175, 2000.

[24] G. A. Rousselet, S. J. Thorpe and M. Fabre-Thorpe. How parallel is visual processing in the ventral pathway? Trends in Cognitive Sciences, 8(8):363-370, 2004.

[25] A. T. Sack, A. Kohler, D. E. J. Linden, R. Goebel, and L. Muckli. The temporal characteristics of motion processing in hMT/V5+: Combining fMRI and neuronavigated TMS. NeuroImage, 29:1326-1335, 2006.

[26] C. R. Michael. Columnar organization of color cells in monkey's striate cortex. Journal of Neurophysiology, 46:587-604, 1981.

[27] D. L. Ringach. Spatial Structure and Symmetry of Simple-Cell Receptive Fields in Macaque Primary Visual Cortex. Journal of Neurophysiology, 88:455-463, 2002.

[28] M. Mishkin. L. G. Ungerleider, and K. A. Macko. Object vision and spatial vision: two cortical pathways. Trends Neuroscience, 6:414-417, 1983.

[29] M. G. P. Rosa. Visual maps in the adult primate cerebral cortex: some implications for brain development and evolution. Brazilian Journal of Medical and Biological Research, 35:1485-1498, 2002.

[30] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. Journal of Phyiology, 160:106-154, 1962.

[31] I. Lampl, D. Ferster, T. Poggio, and M. Riesenhuber. Intracellular measurements of spatial integration and the max operation in complex cells of the cat primary visual cortex. Journal of Neurophysiology, 92:2704-2713, 2004.

[32] N. Kanwisher, J. McDermott, and M. M. Chun. The Fusiform Face Area: A Module in Human Extrastriate Cortex Specialized for Face Perception. The Journal of Neuroscience, 17(11):4302-4311, 1997.

[33] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4):193-202, 1980.

[34] K. Fukushima. Neocognitron. Scholarpedia, p.7715, 2007.

[35] M. Riesenhuber, and T. Poggio. Hierarchical Models of Object Recognition in Cortex. Nature Neuroscience, 2:1019-1025, 1999.

[36] N. Logothetis, J. Pauls and T. Poggio. Shape representation in the inferior temporal cortex of monkeys. Current Biology, 5:552-563, 1995.

[37] N. C. Rust, O. Schwartz, J. A. Movshon and E. P. Simoncelli. Spatiotemporal Elements of Macaque V1 Receptive Fields. Neuron, 46(6):945-956, 2005.

[38] M. Booth and E. Rolls. View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex. Cerebral Cortex, 8:510-523, 1998.

[39] T. Serre, M. Kouh, C. Cadieu, U. Knoblich, G. Kreiman and T. Poggio. A theory of object recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex, CBCL Paper #259/AI Memo #2005-036, Massachusetts Institute of Technology, Cambridge, MA, December, 2005

[40] T. Serre, A. Oliva, and T. Poggio. Feedforward theories of visual cortex predict human performance in rapid categorization. Journal of Vision, Vol 6, 6:615, 2006.

[41] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robus Object Recognition with Cortex-Like Mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(3):411-426, 2007.

[42] J. Mutch and D. G. Lowe. Multiclass Object Recognition with Sparse, Localized Features. In CVPR, pp 11-18, New York, 2006.

[43] R. F. Dougherty, V. M. Koch, A. A. Brewer, B. Fischer, J. Modersitzki, and B. A. Wandell. Visual field representations and locations of visual areas V1/2/3 in human visual cortex. Journal of Vision, 3:586-598, 2003.

[44] E. M. Callaway. Feedforward, feedback and inhibitory connections in primate visual cortex. Neural Networks, Vol 17, 625-632, 2004.

[45] J. M. Hupe, A. C. James, B. R. Payne, S. G. Lomber, P. Girard and J. Bullier. Cortical feedback improves discrimination between figure and background by V1, V2 and V3 neurons. Nature, 394(6695):784-7, 1998.

[46] J. S. Baizer. Receptive field properties of V3 neurons in monkey. Investigative Ophthalmology and Visual Science, 23:87-95, 1982.

[47] P. Neri, H. Bridge and D. J. Heeger. Stereoscopic processing of absolute and relative disparity in human visual cortex.Stereoscopic processing of absolute and relative disparity in human visual cortex. Journal of Neurophysiology, 92(3):1880-1891, 2004.

[48] D. A. Stanley and N. Rubin. fMRI activation in response to illusory contours and salient region in the human lateral occipital complex. Neuron, 37:323-331, 2003.

[49] A. C. Huk and D. J. Heeger. Pattern-motion responses in human visual cortex. Nature Neuroscience, 5(1):72-75, 2001.

[50] N. M. Grzywacz and D.K. Merwine. Neural Basis of Motion Perception. The Encyclopedia of Cognitive Science, 3:86-98, 2003.

[51] M. C. Morrone, M. Tosetti, D. Montanaro, A. Fiorentini, G. Cioni, D. C. Burr. A cortical area that response specifically to optical flow. Nature Neuroscience, 3:1322-1328, 2000.

[52] A. Angelucci, and P. C. Bressloff. Contribution of feedforward, lateral and feedback connections to the classical receptive field center and extra-classical receptive field surround of primate V1 neurons. Progress in Brain Research, Vol. 154, 1:93-120, 2006.

[53] S. Corchs, and G. Deco. Large-scale Neural Model for Visual Attention: Integration of Experimental Single-cell and fMRI Data. Cerebral Cortex, Vol. 12, 339-348, 2002.

[54] E. Peterhans, R. von der Heydt. Subjective contours - bridging the gap between psychophysics and physiology. Trends Neuroscience, 14:112-19, 1991.

[55] G. M. Ghose, R. D. Freeman, and I. Ohzawa. Local intracortical connections in the cat's visual cortex: postnatal development and plasticity. Journal of Neurophysiology, 72(3):1290-1303, 1994.

[56] D. G. Lowe. Object recognition from local scale-invariant features. ICCV, 1150-1157, Corfu, Greece, 1999.

[57] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In Proc. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 506-513, 2004.

[58] C. G. Harris and M. Stephens. A combined corner and edge detector. In Proc. 4th Alvey Vision Conf., Manchester, 147-151, 1988.

[59] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In IEEE CVPR 2005, 1:510-517, 2005.

[60] S. M. Smith. A new class of corner finder. In Proc. 3rd British Machine Vision Conference, 139-148, 1992.

[61] I. T. Jolliffe. Principal Component Analysis. Springer Series in Statistics, New York, 1986.

[62] T. F. Cox and M. A. A. Cox. Multidimensional Scaling. Chapman & Hall, London, 1994.

[63] M. James. Classification Algorithms. Collins, London, England, 1985.

[64] Z. Fan, F. Qiu, A. Kaufman, and S. Yoakum-Stover. GPU Cluster for High Performance Computing. In Proc ACM/IEEE SC Conference, 47, 2004.

[65] M. Harris. Mapping Computational Concepts to GPUs. In GPU Gems 2. Addison-Wesley, 493-508, 2005.

[66] T. Kohonen. Self-Organizing Maps. Springer Series in Information Sciences, New York, 2001.

[67] J. Pikkanen, J. Iivarinen and E. Oja. The Evolving Tree - A Novel Self-Organizing Network for Data Analysis. Neural Processing Letters, 20(3):199-211, 2004.

[68] J. Sirosh and R. Miikkulainen. Cooperative Self-Organization of Afferent and Lateral Connections in Cortical Maps. Biological Cybernetics, 71:66-78, 1994.

[69] R. Miikkulainen, J. A. Bednar, Y. Choe, and J. Sirosh. Self-Organization in the Primary Visual Cortex: The RF-LISSOM ModelSelf-Organization, Plasticity, and Low-Level Visual Phenomena in a Laterally Connected Map Model. Psychology of Learning and Motivation, 36:257-308, 1997.

[70] G. Montgomery. Seeing, hearing and smelling the world. Howard Hughes Medical Institute, pp 15-24, 1995.

[71] S. Chatterjee and E. M. Callaway. Parallel colour-opponent pathways to primary visual cortex. Nature, 426:668-671, 2003.

[72] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision, 2004.

[73] A. D. Holub and G. Griffin. Caltech 101. Retrieved May 1 2007, from http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html

[74] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam library of object images. International Journal of Computer Vision, 61(1):103-112, 2005.

[75] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. International Journal of Computer Vision 47(1/2/3):7-42, 2002.

[76] D. E. Linden, U Kallenbach, A. Heinecke, W. Singer and R. Goebel. The myth of upright vision. A psychophysical and functional imaging study of adaptation to inverting spectacles. Perception, 28(4):469-481.

[77] C. C. Chang and C. J. Lin. LIBSVM: a library for support vector machines. May 2007. http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf

[78] A. D. Holub, M. Welling and P. Perona. Exploiting unlabelled data for hybrid object classification. In NIPS Workshop on Inter-Class Transfer, Whistler, B.C., 2005.

[79] S. Grigorescu and N. Petkov. 2-D Gabor Function. May 2007. http://www.cs.rug.nl/~imaging/simplecell.html

[80] M. Christen. Clockworkcoders Tutorials. May 2007. http://www.opengl.org/sdk/docs/tutorials/ClockworkCoders/

[81] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features. Technical Report MIT-CSAIL-TR-2006-020, March 2006.

[82] H. Zhang, A. Berg, M. Maire and J. Malik. SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In CVPR, 2006.

[83] M. Schrauf, B. Lingelbach and E. R Wist. The scintillating grid illusion. Vision Research, 37:1033-1038, 1997.

[84] P. Mordohai and G. Medioni. Stereo using monocular cues within the tensor voting framework. Pattern Analysis and Machine Intelligence, 28(6):968-982, 2006.

[85] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nistér. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. 3DPVT, 2006.

[86] C. Strecha, R. Fransens, and L. Van Gool. Combined depth and outlier estimation in multi-view stereo. CVPR, 2006.

[87] M. Segal and K. Akeley. The OpenGL Graphics System: A Specification (Version 2.1), 2006.