Old Dominion University ODU Digital Commons

Electrical & Computer Engineering Theses & Disssertations

Electrical & Computer Engineering

Summer 2006

Learning as a Nonlinear Line of Attraction for Pattern Association, Classification and Recognition

Ming-Jung Seow Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/ece_etds Part of the <u>Artificial Intelligence and Robotics Commons</u>, and the <u>Electrical and Computer</u> <u>Engineering Commons</u>

Recommended Citation

Seow, Ming-Jung. "Learning as a Nonlinear Line of Attraction for Pattern Association, Classification and Recognition" (2006). Doctor of Philosophy (PhD), dissertation, Electrical/Computer Engineering, Old Dominion University, DOI: 10.25777/rtv9-tw07 https://digitalcommons.odu.edu/ece_etds/120

This Dissertation is brought to you for free and open access by the Electrical & Computer Engineering at ODU Digital Commons. It has been accepted for inclusion in Electrical & Computer Engineering Theses & Disssertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

LEARNING AS A NONLINEAR LINE OF ATTRACTION FOR PATTERN ASSOCIATION, CLASSIFICATION AND RECOGNITION

by

Ming-Jung Seow M.S., December 2002, Old Dominion University

A Dissertation Submitted to the Faculty of Old Dominion University in Partial Fulfillment of the Requirement for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY August 2006

Approved by:

Dr. Vijayan K. Asari (Director)

Dr. Stephen A. Zahorian (Member)

Dr. Lee A. Belfore II (Member)

Dr. Jessica R. Crouch (Member)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ABSTRACT

LEARNING AS A NONLINEAR LINE OF ATTRACTION FOR PATTERN ASSOCIATION, CLASSIFICATION AND RECOGNITION

Ming-Jung Seow Old Dominion University, 2006 Director: Dr. Vijayan K. Asari

The human brain memorizes information using a dynamical system made of interconnected neurons. Retrieval of information is achieved in an associative sense, starting from an arbitrary state that might be an encoded visual representation; the brain activity converges to a stable state in which the brain remembers. Associative memory can be modeled using a recurrent network, in which the stored memories are represented by the dynamics of the network convergence. Development of a mathematical model for learning a nonlinear line of attraction is presented in this dissertation, in contrast to the conventional recurrent neural network model in which the memory is stored in an attractive fixed point at discrete location in state space. A nonlinear line of attraction is the encapsulation of attractive fixed points scattered in state space as an attractive nonlinear line, describing patterns with similar characteristics as a family of patterns.

It is usually of prime imperative to guarantee the convergence of the dynamics of the recurrent network for associative learning and recall. We propose to alter this picture. That is, if the brain remembers by converging to the state representing familiar patterns, it should also diverge from such states when presented by an unknown encoded representation of a visual image. The conception of the dynamics of the nonlinear line attractor network to operate between stable and unstable states is the second contribution in this dissertation research. These criteria can be used to circumvent the plasticitystability dilemma by using the unstable state as an indicator to create a new line for an unfamiliar pattern. This novel learning strategy utilizes stability (convergence) and instability (divergence) criteria of the designed dynamics to induce self-organizing behavior. The self-organizing behavior of the nonlinear line attractor model can manifest complex dynamics in an unsupervised manner.

The third contribution of this dissertation is the introduction of the concept of manifold of color perception. Such a manifold representation can be learned using a novel recurrent neural network based learning algorithm. The concept of color manifold is utilized in this dissertation to solve challenging problems in image processing, computer vision and pattern recognition problems such as natural color representation, color balancing for nonlinear image enhancement, color constancy, and pattern classification.

The fourth contribution of this dissertation is the development of a nonlinear dimensionality reduction technique by embedding a set of related observations into a low-dimensional space utilizing the result attained by the learned memory matrices of the nonlinear line attractor network. One of the aims of biometrics research is to develop new techniques and/or algorithms for the automatic recognition of humans. The nonlinear dimensionality reduction technique is applied to model face in a set of similar face images—subjected to variations in pose, illumination and expression— as the manifold of facial perception.

Development of a system for affective states computation is also presented in this dissertation. This system is capable of extracting the user's mental state in real time using

a low cost computer. It is successfully interfaced with an advanced learning environment for human-computer interaction.

То

My Family

LIST OF TABLES	xi
LIST OF FIGURES	xii
CHAPTER	
1. INTRODUCTION	1
1.1. Underlying Fundamental Assumptions	3
1.2. Specific Objectives	4
1.3. Background Work	5
1.3.1. Hopfield Neural Network	5
1.3.2. Continuous Attractor	6
1.3.3. Principal Components Analysis	6
1.3.4. Backpropagation Learning	7
1.3.5. Support Vector Machines	8
1.3.6. Self-organizing Map	9
1.3.7. Adaptive Resonance Theory	9
1.3.8. Studies on the Human Brain	10
1.4. Dissertation Contributions	10
2. LEARNING ALGORITHMS	13
2.1. Ratio Rule Learning Algorithms	15
2.2. Nonlinear Line of Attraction	
2.3. Dynamics of the Networks	24
2.4. Simulation Results and Discussion	

TABLE OF CONTENTS

3.	STAE	BILITY OF THE ATTRACTOR NETWORKS
	3.1.	Stability of the Attractor Network
	3.2.	Divergence Dynamics of the Nonlinear Line of Attraction
	3.3.	Instability of the Divergence Dynamic
	3.4.	Simulation Results and Discussion35
4.	APPL	ICATION I: ASSOCIATIVE MEMORY
	4.1.	Ratio Rule and Homomorphic Filter for Enhancement of Digital Color Image40
		4.1.1. Problem Statement
		4.1.2. Color Image Enhancement Model41
		4.1.3. Results and Discussion46
	4.2.	Natural Color Representation using Ratio Learning Algorithm for Gray Level Enhancement of Digital Color Images
		4.2.1. Problem Statement
		4.2.2. Natural Color Representation
		4.2.3. Comparison
		4.2.3.1. Gain/Offset Correction
		4.2.3.2. Non-linear Transform
		4.2.3.3. Histogram Equalization57
		4.2.3.4. Homomorphic Filtering
		4.2.4. Results and Discussion
	4.3.	On Creating a Color Manifold for Color Characterization and Balancing using a Nonlinear Line of Attraction for Nonlinear Image Enhancement
		4.3.1. Problem Statement

		4.3.2.	Nonlinear I Tangent Fu	mage Enhancement using Hyperbolic inction
		4.3.3.	Simulation	Results and Discussion
	4.4.	Toward Memor	ls Represen y for Color	tation of Perceptual Color Manifold using Associative Constancy
		4.4.1.	Problem St	atement
		4.4.2.	Related Wo	orks
		4.4.3.	Color Mani	ifold and its Convergence Dynamics
		4.4.4.	Modeling the Results and	he Manifold of Color Perception and Simulation d Discussion87
5.	APPI	LICATIO	N II: PATT	ERN CLASSIFICATION97
	5.1. F	Problem S	tatement	
	5.2. 8	Simulation	n Results an	d Discussion97
6.	APPI	LICATIO	N III: DIMI	ENSIONALITY REDUCTION103
	6.1.	Probler	n Statemen	t
	6.2.	A New Lightin	Nonlinear l g Invariant	Dimensionality Reduction Technique for Pose and Face Recognition104
	6.3.	Preproc	cessing	
	6.4.	Invaria	nt Face Rec	ognition110
		6.4.1.	Feature Spa	ace Division111
		6.4.2.	Adaptive F	Seature Space Division113
		6.4.3.	Nonlinear	Dimensionality Reduction114
			6.4.3.1.	Skin Region Extraction117
			6.4.3.2.	Expression Invariant Face Recognition118
			6.4.3.3.	Facial Expression Classification119
			6.4.3.4.	Pose Invariant Face Recognition120

	6.5.	FRGC Experiments121
7.	SELF	-ORGANIZATION OF NONLINEAR LINE OF ATTRACTION124
	7.1.	Nonlinear Line Attractor Network Model126
	7.2.	Self-organizing Algorithm127
	7.3.	Dynamic of the Network130
		7.3.1. Pattern Classifier131
		7.3.2. Associative Memory131
	7.4.	Applications and Discussion131
8.	HUM	AN COMPUTER INTERACTION134
	8.1.	Perceptual User Interface
	8.2.	Affective Computing136
	8.3.	Face Acquisition138
		8.3.1. Integral Image140
		8.3.2. AdaBoost142
		8.3.3. Cascade of Filters143
		8.3.4. Experimental Results146
	8.4.	Facial Features Extraction148
		8.4.1 Active Appearance Model148
		8.4.1.1. Shape Formation150
		8.4.1.2. Texture Formation150
		8.4.1.3. Combined Model Formation151
		8.4.1.4. AAM Matching152
		8.4.2. Experimental Results153

.

8.5.	8.5. Fa	acial	Expression Re	cognition156
	8.:	5.1.	Extraction of t	he Facial Action157
	8.:	5.2.	Affective Cue	Computation160
			8.5.2.1.	Head Nod Detection161
			8.5.2.2.	Head Shake Detection162
			8.5.2.3.	Experimental Evaluation164
	8.:	5.3.	Affective State	es Computation166
			8.5.3.1.	Experimental Evaluation173
	9. CONCLU	ISIOI	N	
	REFERENCES			
	CURRICULUM	VITA	AE	

LIST OF TABLES

Table	Page
2.1.	Results of nonlinear line attractor network compared to methods in [59] and [10]
3.1.	Classification of colors using nonlinear line attractor networks
4.3.1.	Results of proposed method compared to methods in [78] and [61]78
8.1.	Measurement of the Head AUs with respect to the points in Fig. 8.11
8.2.	Measurement of the facial AUs with respect to the points in Fig. 8.11
8.3.	List of affective cues and their component AUs160
8.4.	Processing time of Kaliouby (source: [128])170
8.5.	Processing time of the proposed system173

LIST OF FIGURES

Figure

Page

2.1.	Block diagram of a linear associative memory15
2.2.	Associative memory model of a nervous system16
2.3.	Model of an output neuron
2.4.	Weight graph for ratio rule
2.5.	Weight graph for line attractor24
2.6.	Test images used in image reconstruction example
2.7.	Original image divided into 500 20-dimensional vectors27
2.8.	Images corrupted by 40% salt and pepper noise (top) and their reconstructions obtained by nonlinear line attractor networks (bottom)
2.9.	Lena image corrupted by 20% salt and pepper noise and reconstructed Lena image by [59] (left), by [10] (middle), and by nonlinear line attractor network (right)
2.10.	Filtered images obtained by nonlinear line attractor network (middle) and median filter (right)
3.1.	Families of color and corrupted color with Gaussian noise
3.2.	Number of iterations for the 768 pixels with zero mean μ and variance σ^2 between 0 and 16 to converge/diverge
3.3.	Two families of patterns
4.1.1.	Spectral sensitivity of the typical human visual system42
4.1.2.	Result of grayscale enhancement with and without Ratio learning algorithm
4.1.3.	Convergence characteristics48
4.1.4.	Families of color, corrupted color, and recalled color49

4.1.5.	Rms error with 10 families of colors with different variances51
4.2.1.	Color characterization, enhancement, and color balancing system56
4.2.2.	Color image enhancement (left) with (right) and without (middle) color characterization, enhancement, and color balancing60
4.3.1.	Color manifold62
4.3.2.	Memory of the attractor network64
4.3.3.	Re-tuning of the pixel67
4.3.4.	Result of Gray scale enhancement with and without color characterization and color balancing
4.3.5.	Convergence characteristic71
4.3.6.	Comparison of the proposed method with McCann99 Retinex and Multiscale Retinex
4.3.7.	Comparison of the proposed method with McCann99 Retinex and Multiscale Retinex
4.3.8.	Division of the histogram to five zones75
4.3.9.	Histograms of images in Figure 4.3.677
4.3.10.	Histograms of images in Figure 4.3.777
4.3.11.	Skin Segmentation Result using [57] with and without image enhancement
4.4.1.	Families of color, corrupted color, and recalled color85
4.4.2.	Number of iteration for the 5000 pixel with zero mean μ and variance σ^2 between 0 and 32 to converge to the nonlinear line of attraction
4.4.3.	The effect of varying σ in $S(s,q)$
4.4.4.	Manifold of color perception
4.4.5.	Memory of nonlinear line attractor network
4.4.6.	Color balancing algorithm
4.4.7.	Results of color correction94

4.4.8.	Results of color correction using MSRCR95
4.4.9.	Skin segmentation result96
5.1.	Skin distribution in <i>RGB</i> space98
5.2.	<i>RGB</i> weight graph representing 3-neuron linear attractor memory99
5.3.	Skin segmentation result
5.4.	Receiver operating characteristics curve of RGB space100
5.5.	Skin distribution in L*a*b space101
5.6.	L*a*b weight graph representing 3-neuron linear attractor memory102
5.7.	Receiver operating characteristics curve of <i>RGB</i> space L*a*b102
6.1.	Hyperbolic tangent function
6.2.	Lighting compensation110
6.3.	Distance function112
6.4.	Adaptive feature space division113
6.5.	Illustration of the efficiency of dimensionality reduction using the proposed method for extraction of human skin regions in various illumination conditions (indoor and outdoor)117
6.6.	Illustration of expression invariant face recognition118
6.7.	Illustration of the efficiency of dimensionality reduction using the proposed method for classification of human face expressions119
6.8.	Illustration of the efficiency of dimensionality reduction using the proposed method for pose invariant face recognition121
6.9.	Reconstruction of the face images by the proposed method122
6.10.	ROC of the first FRGC ver2 experiment
7.1.	Illustration of the self-organizing algorithm128
7.2.	Error histogram to find the threshold function129
7.3.	Self organization of nonlinear line attractor network
7.4.	Pattern reconstruction

7.5.	Divergence of a pattern	133
8.1.	Concept diagram of facial expression recognition system	137
8.2.	Haar features use for Viola Jones Detector	140
8.3.	Integral image	141
8.4.	Cascade of Filters	144
8.5.	Face detection results	147
8.6.	AAM Procedure	149
8.7.	AAM Example	149
8.8.	Visible imagery and near infrared imagery	153
8.9.	Sample face images used for training	154
8.10.	Hand annotated image	155
8.11.	Automatic face tracking using AAM	155
8.12.	Face model. 23 feature points track by AAM	157
8.13.	Dynamics of head shake motion	161
8.14.	Head nod detection module	162
8.15.	Head shake detection module	
8.16.	Head and head shake detection	164
8.17.	Computation of affective cues	165
8.18.	Agreement (source: [128])	167
8.19.	Disagreement (source: [128])	168
8.20.	Concentration (source: [128])	
8.21.	Interested (source: [128])	169
8.22.	Thinking (source: [128])	169
8.23.	Unsure (source: [128])	170
8.24.	Affective state computation	171

8.25.	Computation of thinking and unsure
8.26.	Computation of interested, concentration, and angry172
8.27.	Human computer interaction17
8.28.	Affective states computations

CHAPTER 1

INTRODUCTION

The human brain processes enormous volumes of high-dimensional data for everyday perception. To humans, a picture is worth a thousand words, but to a machine, it is just a seemingly random jumble of numbers. Though machines are very fast and efficient, they are far inferior to humans for everyday information processing. Algorithms that mimic the way the human brain computes and learns may be the solution. In this dissertation, we present a model for learning based on the observation that images of similar visual perceptions reside in a complex manifold in a low-dimensional image space.

Manifolds are fundamentals to perception. The perceived features are often highly structured and hidden in a complex set of relationships or high-dimensional abstractions. To model the pattern manifold, a novel learning algorithm using a recurrent neural network is presented. In most recurrent network models, patterns are represented as attractive fixed points at discrete locations in feature space. However, such discrete fixed-point attractors are often not suitable for representing patterns which have similar characteristics. To precisely characterize the similarity of many patterns, it is more appropriate to represent pattern association using a nonlinear line of attraction. The line attractor encapsulates the attractive fixed points scattered in the feature space as a line. Each fixed point on the line corresponds to a pattern with similar features. This nonlinear line thus constitutes a pattern manifold. Development of a mathematical model for a nonlinear line attractor that represents a pattern manifold in the feature space is presented in this dissertation.

The reference model for this work is IEEE Transactions on Neural Networks.

The brain memorizes information using a dynamical system made of interconnected neurons. Retrieval of information is accomplished in an associative sense. It starts from an arbitrary state that might be an encoded representation of a visual image and it converges to another state that is stable. The stable state is what the brain remembers. In designing a recurrent neural network, it is usually of prime importance to guarantee the convergence in the dynamics of the network. We propose to modify this picture: if the brain remembers by converging to the state representing familiar patterns, it should also diverge from such states when presented with an unknown encoded representation of a visual image. That is, the identification of an instability mode can be an indication that a presented pattern is far away from any stored pattern and therefore cannot be associated with current memories. These properties can be used to circumvent the plasticity-stability dilemma by using the fluctuating mode as an indicator to create new states. We propose to capture this behavior using a novel neural architecture and learning algorithm, in which the dynamical system performs self-organization utilizing a stability mode and an instability mode. Hence, in this dissertation, we developed a selforganizing line attractor, which is capable of generating new states in the feature space to learn and represent the unrecognized patterns. It is expected that this model can also create complex manifolds of any shape in a multidimensional feature space for pattern association, classification, and prediction.

Psychophysical findings indicate that perceptual tasks such as similarity judgment tend to be performed using a low-dimensional representation of sensory data. Low dimensionality is especially important for learning, as the number of examples required for attaining a given level of performance grows exponentially with the dimensionality of the underlying representation space. Because of this curse of dimensionality, categorization of the initial high-dimensionality sensory data must be reduced to a nontrivial computational process, which ideally should capture the intrinsic low-dimensional nature of families of visual pattern. Here we present a nonlinear dimensionality reduction technique using the self-organizing dynamics of the nonlinear line attractor network to embed a set of related observations into a low-dimensional space that preserves the intrinsic dimensionality and metric structure of the data. It is expected that extremely complex data in a high dimensional space can be transformed into a low dimensional space where efficient categorization of information is feasible.

1.1. Underlying Fundamental Assumptions

The investigation of this dissertation is guided by the following hypotheses:

- 1. Manifolds are fundamentals to learning. The perceived features are often highly structured and hidden in a complex set of relationships or high-dimensional abstractions embedded in the manifold [1, 2].
- 2. The exploitation of non-convergent dynamics by and within an appropriately embodied system is necessary and sufficient for producing general intelligent behavior [3].
- 3. The brain exploits the edge of chaos to achieve the properties of cognition. It is normal and desirable for a system to operate between stability and instability [4].
- 4. Low dimensionality is important for classification. Categorization in the initial high-dimensionality sensory data must be reduced to a nontrivial computational

process, which ideally should capture the intrinsic low-dimensional nature of families of visual patterns [5].

1.2. Specific Objectives

The specific objectives for this dissertation are:

Objective 1:

To develop a biologically inspired model for learning by representing memory as a nonlinear line of attraction, which is a pattern manifold in the feature space, and to analyze the stability of the nonlinear line attractor network.

Objective 2:

To develop an unsupervised learning algorithm using the concept of self-organization that is based on stability and instability criteria of the nonlinear line attractor, and to analyze the dynamical behavior of the self-organizing line attractor network as an associative memory and a pattern classifier.

Objective 3:

To develop a nonlinear dimensionality reduction technique by embedding a set of related observations into a low-dimensional space utilizing the result attained by the learned memory matrices of the nonlinear line attractor network.

Objective 4:

To apply the proposed algorithm to computer vision and pattern recognition applications such as skin region extraction, face recognition, face expression analysis, and human computer interaction.

1.3. Background Work

1.3.1. Hopfield Neural Network

An important advancement in the area of associative memory was made in the 1980s by John Hopfield, who developed the Hopfield network to help understand how biological memory works [6]. The use of a recurrent network as an associative memory is a classic approach to invariant object recognition [7]. The motivation for exploring recurrent architectures is in their potential for dealing with temporal behavior. Recurrent networks are capable of converging to a solution that satisfies many constraints, in a manner that seems to mimic human perception. For example, a human vision system appears to relax to an interpretation of an image that maximally satisfies a complex set of conflicting constraints [8]. In spite of the intuitive appeal and biological plausibility of this approach, it has largely been abandoned in practical applications. The main drawback of the Hopfield network is that, while motivated by biological memory, it still employs discrete state memory units with their inherent limitations [8]. Muezzinoglu et al. proposed a multi-valued associative memory to store fixed points into a complex valued multistate Hopfield network [10]. In this model of associative memory, memories are stored as attractive fixed points at discrete locations in feature space. Discrete attractors may not be appropriate for patterns with continuous variability like the images of a three-dimensional object from different viewpoints. When the instantiations of an object lie on a continuous pattern manifold, it is more appropriate to represent objects by attractive manifolds of fixed points, or continuous attractors [7].

1.3.2. Continuous Attractor

Several recent models of learning in cognitive neural systems (CNS) are referred to as continuous attractors. A continuous attractor has the advantage that it can potentially store all values of an analog input as a manifold. Seung et al. [7] devised a method to design a spiking neural network model that approximates a line attractor. Their model successfully replicates the main features of data recorded from persistent neurons that encode eye position in a goldfish. However, this research group also highlighted the difficulties involved in building a line attractor. Changing connection strengths between the neurons by as little as 1% produced unacceptable rapid drifts in the predicted eye position. In other words, even a slight tilt in the attractor leads the model to predict results that are incompatible with behavior in the fish. Such precision in construction appears to be biologically unrealistic.

1.3.3. Principal Components Analysis

Principal components analysis (PCA) is commonly used to perform dimensionality reduction by projecting data into a subspace spanned by the eigenvectors of the covariance matrix. The strength of PCA comes from its efficient computational mechanism. In computer vision applications, it has been used for the representation and recognition of faces [11] and recognition of 3D objects under varying poses [12]. However, dimensionality-reduction techniques are sensitive to image plane transformations. This approach will not work well for the following conditions:

High curvature of the manifold:

PCA finds a low-dimensional embedding of the data points that best preserves their variance as measured in the high-dimensional input space. However, many data sets contain essential nonlinear structures that are invisible to PCA.

Presence of small manifolds:

In many real-world problems, there is not only one global manifold but also a large number of manifolds that share additional information about the objects. A simple example is the manifold of transformations (view-point, position, lighting, etc) of 3D objects in 2D images. PCA is incapable of functioning in these complex situations.

1.3.4. Backpropagation Learning

Multilayer perceptrons have been applied successfully to solve many difficult and diverse problems by training them in a supervised manner with the backpropagation algorithm [13]. Backpropagation is probably the most widely applied neural network learning algorithm. Its popularity is related to its ability to form complex multidimensional mappings. However, a wide variety of cases exist for which backpropagation training is known to fail because of local minima issues [14]. The difficulty with backpropagation is that, once the architecture is chosen, no general method exists that can guarantee the gradient descent to converge to the global minima.

1.3.5. Support Vector Machines

Support vector machines (SVM) have been recently developed as a robust tool for classification and regression in noisy and complex domains [15]. SVMs can be used to extract valuable information from datasets and construct fast classification algorithms for massive data sets. The two key features of support vector machines are generalization theory, which leads to a principled way to choose a hypothesis and kernel functions, which introduce non-linearity in the hypothesis space without explicitly requiring a nonlinear algorithm [16]. SVM maps data points to a high-dimensional feature space, where a separating hyper-plane can be found. This mapping is performed implicitly using kernels. The separating hyper-plane is computed by maximizing the distance of the closest patterns [17-18]. The limitations of support vector machines include:

- Choice of the kernel function: Once the kernel is fixed, SVM classifiers have only one user chosen parameter (the error penalty function), but the kernel is an important part of support vector machine learning for the nonlinear transformation.
- Size problem: Training for very large dataset, which involve millions of support vectors, is an unsolved problem.
- Multi-class problem: Optimal design for multi-class SVM classifiers is still ongoing research.

1.3.6. Self-organizing Map

Kohonen's Self-Organizing Map (SOM) network is one of the most important network architectures developed during the 1980s [19]. The main function of SOM networks is to map the input data from an N-dimensional space to a lower dimensional space while maintaining the original topological relations. A limitation of Kohonen's network is the boundary effect of nodes near the edge of the network [20]. The boundary effect is responsible for retaining the undue influence of initial random weights assigned to the nodes of the network leading to ineffective topological representations.

1.3.7. Adaptive Resonance Theory

The concept of Adaptive Resonance Theory (ART) was introduced by Grossberg in 1976 [21] to solve the *plasticity/stability dilemma*. The first version of ART network, namely ART-1, proposed by Carpenter and Grossberg in 1987, was used to cluster binary data. Since then several variations of ART had been developed. The most important ones are ART-2 [22], an extension of ART-1 used to cluster analog data, ARTMAP [23], a supervised learning mechanism for binary data, and Fuzzy ARTMAP [24], a supervised learning algorithm for analog data. Many researchers around the world have proposed other types of ART networks [25-27]. The contribution of Adaptive Hamming Network [27] is that it realizes the inefficiency in ART algorithm and improves it by converting the searching problem to an optimization problem.

1.3.8. Studies on the Human Brain

On the basis of studies of the olfactory bulb of a rabbit, Freeman suggested that in the rest state, the dynamics of this neural cluster in the rabbit is chaotic [28]. Conversely, when a familiar scent is presented to the rabbit, the neural system rapidly simplified its behavior and the dynamics becomes more orderly. A chaotic system in general, and the chaos exhibited in the brain, often alternates in a seemingly random way between various areas of the phase space. The way the brain uses chaos to ensure continual access to previously learned patterns is to develop attractors for learned patterns. According to researchers, the background chaotic activity of the brain enables the system to jump rapidly from one attractor to another when presented with the appropriate input. That is, if the input does not send the system into one of the attractors, it is considered a novel input [29-33]. Some researchers have long speculated that chaotic processes have some fundamental roles in mental processes. It is indeed intriguing to think that dynamics of networks at the "edge of chaos" might be some source of creativity.

1.4. Dissertation Contributions

Low dimensionality is important for classification. The human brain processes enormous volumes of high-dimensional data for everyday perception. Psychophysical findings indicate that perceptual tasks such as similarity judgment are performed using a low-dimensional representation of the sensory data. Information related to similar data resides in a complex manifold, which can be visualized as a curved line in the feature space. This nonlinear line encapsulates attractive fixed points representing patterns with similar

characteristics. A model for learning a pattern manifold is presented in this dissertation based on the above observations.

In many real-world problems, there is not only one global manifold but also a large number of sub-manifolds that share additional information about the objects. We propose to capture this structure using a novel learning algorithm, in which the dynamical system performs self-organization utilizing a stability mode and an instability mode. These criteria can be used to circumvent the plasticity-stability dilemma by using the instability mode as an indicator to create a new line for an unfamiliar pattern. This selforganizing behavior of the nonlinear line attractor model helps to create complex manifolds in an unsupervised manner. The specific contributions of this dissertation research are:

- Development of a biologically inspired model for learning by representing memory as a nonlinear line of attraction, which is a pattern manifold in the feature space, and to analyze the stability of the nonlinear line attractor network.
- Development of an unsupervised learning algorithm using the concept of selforganization that is based on stability and instability criteria of the nonlinear line attractor, and to analyze the dynamical behavior of the network as an associative memory and a pattern classifier.
- Development of a nonlinear dimensionality reduction technique by embedding a set of related observations into a low-dimensional space utilizing the result attained by the learned memory matrices of the self-organizing nonlinear line attractor network.

• Use of the nonlinear dimensionality reduction technique to computer vision and pattern recognition applications such as skin region extraction, face recognition, expression analysis, and human computer interactions.

CHAPTER 2

LEARNING ALGORITHMS

During the past decades, artificial neural networks have been used in many different applications. Those applications include feature mapping in multilayer neural networks [34], optimization techniques using Hopfield neural networks [35], self-organized mapping by Kohonen's neural network [36], and pattern recognition and pattern association using Hopfield neural network [8, 35]. In this dissertation, we focus on the pattern classification, pattern association, and pattern prediction problem. Artificial neural networks used for pattern classification, pattern association, and pattern association, and pattern prediction problem.

- 1. An artificial neural network utilizing adaptive learning is capable of learning how to do tasks based on the data given for training.
- 2. An artificial neural network can process data in a parallel fashion. Hence, it is much faster than conventional techniques and makes it very suitable for real time applications.
- 3. It can store data in a distributed manner. Hence, it is very tolerant to noisy inputs.

One of the major goals of both biological neural networks modeling and artificial neural networks research is to discover better learning rules to yield networks that can learn more difficult tasks, such as tasks that the brain can handle [37, 38]. Most researchers now accept that animal learning involves changes of synaptic efficacy [39, 40]. Several researchers have proposed many abstract models. Connectionist models use

fairly simple mechanisms for both the neuron operation and the modification of synapses. They can be used to solve some difficult learning problems, including the problem of training hidden layers of the network when reinforcement or supervision is only available to some neurons [41, 42].

The ability to naturally implement adaptive categorization without the need of postulating an access to existing information from outside the system is one of the most important aspects of unsupervised learning algorithms for artificial neural networks [43, 44, 45]. Furthermore, in a recurrent architecture, those models are also able to categorize new exemplar pattern from previously learned categories, which makes it a very attractive field for research. An example of such a network is the Hopfield neural architecture, which was first introduced by J. J. Hopfield [6, 46]. This model, as any other neural network model, is completely specified by its architecture, its transmission rule and its learning rule.

Attractor networks such as the Hopfield network are used as auto-associative content addressable memories. The aim of such networks is to retrieve a previously learned pattern from an example which is similar, or a noisy version of one of the preciously trained patterns. To do this, the network associates each element of pattern with a binary neuron. These neurons are fully connected, and are updated asynchronously and in parallel. They are initialized with an input pattern, and the network activations converge to the closest learnt pattern. The most common learning rule for the Hopfield network is called the Hebb rule. Many attempts have been made to improve the Hebbian learning rule. Gorodnichy and Reznik used the pseudo-inverse learning rule [47] to increase the capacity of the network. Davey, Adam, and Hunt proposed the consequence

of diluting the weight of the standard Hopfield architecture associative memory [48] as an attempt to improve the associativity of a Hopfield network.

It is known that the bi-polar model of an artificial neural network does not give a good representation of the real biological neural network. In this chapter, we propose learning algorithms for recurrent neural networks suitable for multi-level pattern association. It is shown that the recurrent network can be trained for multi-value pattern association.

2.1. Ratio Rule Learning Algorithm

An associative memory may be classified as linear or nonlinear, depending on the model adopted for the neurons. In a linear case, the neurons act like a linear combiner. To be more specific, let the data vectors a and b denote the stimulus (input) and the response (output) of an associative memory, respectively. In a linear associative memory, the input-output relationship is described by:

$$b = Wa \tag{2.1}$$

where W is called the memory matrix. The matrix W specified the network connectivity of the associative memory. Fig. 2.1 shows the block diagram of a linear associative memory.



Figure 2.1: Block diagram of a linear associative memory.

In a nonlinear associative memory, on the other hand, we have an input-output relationship of the form:

$$b = f(W, a) \tag{2.2}$$

16

where, f(.,.) is a threshold function of the memory matrix and the input vector.

In a distributed memory, the basic issue of interest is the simultaneous or nearsimultaneous activities of many different neurons, which are the result of external or internal stimuli. The neural activities form a large spatial pattern inside the memory that contains information about the stimuli. The memory is therefore said to perform a distributed mapping that transforms an activity pattern in the input space into another activity pattern in the output space. We may illustrate some important properties of a distributed memory mapping by considering an idealized neural network that consists of two layers of neurons as shown in Fig. 2.2. Fig. 2.2 illustrates the case of a network that may be regarded as a model component of a nervous system [49, 50].



Figure 2.2: Associative memory model of a nervous system.

In the mathematical analysis to follow, the neural network shown in Fig. 2.2 is assumed to be linear, that is, each neuron acts as a linear combiner. To proceed with the analysis, suppose that an activity pattern a^s occurs in the input layer of the network and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

that an activity pattern b^s occurs simultaneously in the output layer. The issue we wish to consider here is that of learning from the association between a^s and b^s .

The pattern a^s and b^s represented by vectors, written in their expended forms as follows:

$$a^{s} = \left(a_{1}^{s}, a_{2}^{s}, \cdots, a_{N}^{s}\right)^{T}$$

$$(2.3)$$

and

$$b^{s} = (b_{1}^{s}, b_{2}^{s}, \cdots, b_{N}^{s})^{T}$$
 (2.4)

where the superscript T denotes transposition. Note that N is equal to the number of neurons and s is the index of a pattern. In any event, we may describe the different associations performed by the network by writing:

$$a^s \to b^s \quad s = 1, 2, \cdots, P$$
 (2.5)

where P is the number of training patterns. The activity pattern a^s acts as a stimulus that not only determines the storage location of information in the stimulus a^s , but also holds the key for its retrieval. Accordingly, a^s is referred to as a key pattern, and b^s is referred to as memorized pattern.

With the network of Fig. 2.2, the association vector a^s , and a memorized vector b^s described in eqn. (2.5), we may recast eqn. (2.1) as:

$$b^{s} = W^{s}a^{s} \quad s = 1, 2, \cdots, P$$
 (2.6)

where W^s is a weight matrix determined solely by the input-output pair (a^s, b^s) .

To develop a detailed description of the weight matrix W^s , consider Fig. 2.3, which shows a detailed arrangement of neuron *i* in the output layer.



Figure 2.3: Model of an output neuron.

The output b_i^s of neuron *i* due to the combined action of the elements of the key pattern a^s applied as stimulus to the input layer is given by:

$$b_i^s = \sum_{j=1}^N w_{ij}^s a_j^s$$
 for $i = 1, 2, \dots N$ (2.7)

Using matrix notation, we may express b_i^s in the equivalent form:

$$b_i^s = \left(w_{i1}^s w_{i2}^s \cdots, w_{iN}^s\right) \begin{pmatrix} a_1^s \\ a_2^s \\ \vdots \\ a_N^s \end{pmatrix} \quad \text{for } i = 1, 2, \cdots N$$

$$(2.8)$$

 b^s may then be expressed as:

$$\begin{pmatrix} b_{1}^{s} \\ b_{2}^{s} \\ \vdots \\ b_{N}^{s} \end{pmatrix} = \begin{pmatrix} w_{11}^{s} w_{12}^{s} \cdots w_{1N}^{s} \\ w_{21}^{s} w_{22}^{s} \cdots w_{2N}^{s} \\ \vdots & \vdots & \cdots & \vdots \\ w_{N1}^{s} w_{N2}^{s} \cdots w_{NN}^{s} \end{pmatrix} \begin{pmatrix} a_{1}^{s} \\ a_{2}^{s} \\ \vdots \\ a_{N}^{s} \end{pmatrix}$$
(2.9)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

19

Eqn. (2.9) is the expanded form of the matrix transformation or mapping described in eqn. (2.6). In particular, the $N \times N$ weight matrix W^s is defined by:

$$W^{s} = \begin{pmatrix} w_{11}^{s} & w_{12}^{s} & \cdots & w_{1N}^{s} \\ w_{21}^{s} & w_{22}^{s} & \cdots & w_{2N}^{s} \\ \vdots & \vdots & \cdots & \vdots \\ w_{N1}^{s} & w_{N2}^{s} & \cdots & w_{NN}^{s} \end{pmatrix}$$
(2.10)

The individual presentation of the s^{th} pattern described in eqn. (2.5) produces corresponding values of the individual matrix, namely, W^1 , W^2 , ..., W^p . Recognizing that the pattern association $a^s \rightarrow b^s$ is represented by the weight matrix W^s , we may define the $N \times N$ memory matrix that describes the summation of the weight matrices for the entire set of pattern associations as follows [51]:

$$W = \sum_{s=1}^{P} W^s \tag{2.11}$$

The memory matrix W defines the overall connectivity between the input and output layers of the associative memory. In effect, it represents the total experience gained by the memory as a result of the presentation of s^{th} input-output patterns. Stated in another way, the memory matrix W contains a piece of every input-output pair of activity patterns presented to the memory [50].

Now let us consider the fact that there exists a relation between each neuron and every other neuron, and that relationship is described by the ratio of two neurons:

$$W_{ij}^s = \frac{b_i^s}{a_j^s} \tag{2.12}$$
for input-output pair (a_j^s, b_i^s) and the resultant memory W_{ij}^s . The term $\frac{b_i^s}{a_j^s}$ represents the ratio between the state values of two neurons. This ratio will find the degree of similarity between two neurons. Therefore, W_{ij}^s may then be written as:

$$W^{s} = \begin{pmatrix} w_{11}^{s} & w_{12}^{s} \cdots & w_{1N}^{s} \\ w_{21}^{s} & w_{22}^{s} & \cdots & w_{2N}^{s} \\ \vdots & \vdots & \cdots & \vdots \\ w_{N1}^{s} & w_{N2}^{s} & \cdots & w_{NN}^{s} \end{pmatrix} = \begin{pmatrix} \frac{b_{1}^{s}}{a_{1}^{s}} & \frac{b_{1}^{s}}{a_{2}^{s}} & \cdots & \frac{b_{1}^{s}}{a_{N}^{s}} \\ \frac{b_{2}^{s}}{a_{1}^{s}} & \frac{b_{2}^{s}}{a_{2}^{s}} & \cdots & \frac{b_{2}^{s}}{a_{N}^{s}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{b_{N}^{s}}{a_{1}^{s}} & \frac{b_{N}^{s}}{a_{2}^{s}} \cdots & \frac{b_{N}^{s}}{a_{N}^{s}} \end{pmatrix}$$
(2.13)

To combine W^s to form a memory matrix W, we need to utilize statistical methods. The approach to this problem is to use the samples to estimate the unknown probabilities and probability density, and then use the resulting estimation as if they were the true values. To see how the maximum-likelihood method applies to a specific case, we can reasonably assume that the values of $\frac{b_i^s}{a_j^s}$ are drawn from a multivariate normal density with mean μ , although we do not know the exact value of these quantities. This knowledge simplifies the problem for estimating an unknown function as estimating the b^s

parameter μ . Let the log-likelihood of a single point $\frac{b_i^s}{a_j^s}$ be:

$$\ln\left[P(\frac{b_{i}^{s}}{a_{j}^{s}}|\mu)\right] = -\frac{1}{2}\ln\left(2\pi\right)^{d}|\Sigma| - \frac{1}{2}\left(\frac{b_{i}^{s}}{a_{j}^{s}} - \mu\right)^{T}\Sigma^{-1}\left(\frac{b_{i}^{s}}{a_{j}^{s}} - \mu\right)$$
(2.14)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where \sum is the covariance matrix. We usually find the value of μ that maximized $\ln \left[P(\frac{b_i^s}{a_j^s} | \mu) \right]$ by differentiating it by the component μ and setting the result to zero [52]:

$$\nabla \ln \left[P(\frac{b_i^s}{a_j^s} \mid \mu) \right] = \Sigma^{-1} \left(\frac{b_i^s}{a_j^s} - \mu \right).$$
(2.15)

The maximum likelihood estimation for μ can be obtained by:

$$\sum_{i=1}^{P} \Sigma^{-1} \left(\frac{b_i^s}{a_j^s} - \mu \right) = 0 \qquad \text{for } j = 1, 2, \dots N \tag{2.16}$$

Multiplying eqn. (2.16) on both left and right side by \sum and rearranging, we obtain the following maximum-likelihood estimation for μ .

$$\mu = \frac{1}{P} \sum_{i=1}^{P} \frac{b_i^s}{a_j^s}$$
(2.17)

Eqn. (2.17) shows that the maximum likelihood estimate for the unknown population mean is just the arithmetic average of the training samples. Fig. 2.4 shows the weight graph illustrating the concept of training based on eqn. (2.17). A weight graph is a graphical representation of the relationship between the i^{th} neuron and the j^{th} neuron for *P* patterns. Utilization of the weight graph may help visualize the behavior of one neuron pair.

Based on the above theory, the memory matrix for the *N* neuron network can be obtained as:

$$w_{ij} = \frac{1}{P} \sum_{s=1}^{P} \frac{x_i^s}{x_j^s} \qquad \text{for } 1 \le i, j \le N , \qquad (2.18)$$

where $x_m \in \{1, 256\}$ can be the magnitude of the input at m^{th} neuron. The learning algorithm can be interpreted as the mean ratio between two neurons.



Figure 2.4: Weight graph for ratio rule.

2.2. Nonlinear Line of Attraction

Fixed-point attractors may not be suitable for patterns which exhibit similar characteristics as described in the previous section and in [53]. As a consequence, it was suggested that memory could be represented as a continuous line of attraction in the state space [54-58]. Brody and Kepecs model a basic mechanism for graded persistent activity utilized attractor networks in [54]. Stringer et al. [55] and Seung [56] presented the utilization of continuous attractors for modeling biological processes.

In this section, a general learning model for a recurrent neural network capable of recalling memory vectors \square^N is proposed. The solution of the system is based on the concept of the nonlinear line attractor, in which the network encapsulates the fixed point attractors scattered in the state space as a nonlinear line. The performance of the network is tested by reconstructing noisy gray-scale images and is compared with two classical algorithms [59, 10].

$$x_i^s = \frac{1}{N} \sum_{j=1}^N \Lambda_i \left(x_j^s \right) \quad \text{for} \quad 1 \le i \le N$$
(2.19)

where $\Lambda_i(.)$ is defined by a k^{th} order nonlinear line as:

$$\Lambda_i\left(\Theta_j^s\right) = \sum_{m=0}^k w_{(m,j)}^s \left(\Theta_j^s\right)^m \quad \text{for } 1 \le i, j \le N$$
(2.20)

The m^{th} order term of the resultant memory w^s can be expressed as:

$$w_{m}^{s} = \begin{pmatrix} w_{(m,11)}^{s} & \dots & w_{(m,1N)}^{s} \\ \vdots & \ddots & \vdots \\ w_{(m,N1)}^{s} & \dots & w_{(m,NN)}^{s} \end{pmatrix} \text{ for } 0 \le m \le k$$
 (2.21)

Statistical methods can be utilized to combine w^s for *P* patterns to form a memory matrix *w*. The Least squares estimation approach in this problem can determine the best fit line where the error involved is the sum of squares of the differences between the expected outputs and the approximated outputs. Hence, the weight matrix that minimizes the total least squares error can be found as:

$$E_{ij}\left[w_{(0,ij)}, w_{(1,ij)}, \cdots, w_{(k,ij)}\right] = \sum_{s=1}^{P} \left[x_i^s - \Lambda_i\left(x_j^s\right)\right]^2 \text{ for } 1 \le i, j \le N$$
(2.22)

A necessary condition for the coefficients $w_{(0,ij)}$, $w_{(1,ij)}$, ..., $w_{(k,ij)}$, to minimize the total error E_{ij} is:

$$\frac{\partial E_{ij}}{\partial w_{(m,ij)}} = 0 \quad \text{for each } m = 0, 1, \cdots, k$$
(2.23)

The coefficients $w_{(0,ij)}$, $w_{(1,ij)}$, ..., $w_{(k,ij)}$ can hence be obtained by solving the above equations. Fig. 2.5 shows the weight graph illustrating the concept of training based on

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the above theory. A weight graph is a graphical representation of the relationship between the i^{th} neuron and the j^{th} neuron for P patterns. Utilization of the weight graph can help visualize the behavior of one neuron pair.



Figure 2.5: Weight graph for line attractor.

2.3. Dynamics of the Networks

The dynamics of the network is evolved in iteration t based on:

$$x_i(t+1) = x_i(t) + \alpha \frac{1}{N} \sum_{j=1}^N \Delta x_{ij}(t) \quad \text{for } 1 \le i, j \le N$$
(2.24)

where α is the update rate defined in the range: $0 < \alpha \leq 1$, and $\Delta x_{ij}(t)$ is the difference between the approximated output and the actual output obtained as:

$$\Delta x_{ij}(t) = \left[\Phi\left\{\Lambda_i\left[x_j(t)\right]\right\} - x_i(t)\right]$$
(2.25)

where $\Phi(.)$ is the activation function and it can be found by considering the distance between the approximated output and the actual output shown in the weight graph illustrated in Fig. 2.5. That is, in order to consider each pattern, we need to find the region where the threshold can encapsulate the pattern. Mathematically the activation function can be expressed as:

$$\Phi\left\{\Lambda_{i}\left[\Theta_{j}\left(t\right)\right]\right\} = \begin{cases} \Theta_{i}\left(t\right) & \text{if } \psi_{ij}^{-} \leq \left\{\Lambda_{i}\left[\Theta_{j}\left(t\right)\right] - \Theta_{i}\left(t\right)\right\} \leq \psi_{ij}^{+} \\ \Lambda_{i}\left[\Theta_{j}\left(t\right)\right] & \text{otherwise} \end{cases}$$
(2.26)

where

$$\psi_{ij}^{-} = \begin{cases} \psi_{(1,ij)}^{-} & \text{if } \theta \leq x_j < \frac{L}{\Omega} \\ \psi_{(2,ij)}^{-} & \text{if } \frac{L}{\Omega} \leq x_j < \frac{2L}{\Omega} \\ \vdots \\ \psi_{(\Omega,ij)}^{-} & \text{if } (\Omega-1)\frac{L}{\Omega} \leq x_j < L \end{cases}$$

$$(2.27)$$

and

$$\psi_{ij}^{+} = \begin{cases} \psi_{(1,ij)}^{+} & \text{if } 0 \leq x_{j} < \frac{L}{\Omega} \\ \psi_{(2,ij)}^{+} & \text{if } \frac{L}{\Omega} \leq x_{j} < \frac{2L}{\Omega} \\ \vdots \\ \psi_{(\Omega,ij)}^{+} & \text{if } (\Omega - 1) \frac{L}{\Omega} \leq x_{j} < L \end{cases}$$

$$(2.28)$$

where

$$\psi_{(l,j)} = \max \left\{ \min_{\forall s} \left\{ \begin{bmatrix} \Lambda_i(x_j^s) - x_i^s \end{bmatrix} \\ \text{in the region} \\ (l-1)\frac{L}{\Omega} \le x_j^s < l\frac{L}{\Omega} \end{bmatrix} \text{ for } 1 \le l \le \Omega, \quad 0 \right\}$$
(2.29)

and

$$\psi_{(l,ij)}^{+} = \min \left\{ \max_{\forall s} \begin{cases} \left[\Lambda_{i} \left(x_{j}^{s} \right) - x_{i}^{s} \right] \\ \text{in the region} \\ \left(l-1 \right) \frac{L}{\Omega} \leq x_{j}^{s} < l \frac{L}{\Omega} \end{cases} \quad \text{for } 1 \leq l \leq \Omega, \quad 0 \right\}$$
(2.30)

26

where L is the upper limit of x_j^s and Ω is the number of threshold regions such that $1 \le \Omega \le L$. The values $\psi_{(l,j)}^-$ and $\psi_{(l,j)}^+$ define the lower and upper limit around the nonlinear line of attraction. The *min* and *max* functions defined in eqns. (2.29) and (2.30) such that the threshold region will encapsulate the attractive curve line, as illustrated in Fig. 2.5.

2.4. Simulation Results and Discussion

To illustrate this associative property of the neural network, the proposed learning algorithm was applied to image reconstruction problems. Images were degraded with noise and the network was trained to reproduce the original images iteratively by converging to the learned stable states. Experiments were conducted on gray-scale versions of three well-known images, namely Baboon, Lena, and Camera-man shown in Fig. 2.6, to verify the effectiveness of the proposed learning algorithm. These images have good mixture of details, flat regions, shading, and texture.



Figure 2.6: Test images used in image reconstruction example.

For computational simplicity, the original images of size 512×512 were scaled to 100×100 . In addition, each image has been divided into 500 20-dimensional vectors by considering the sub-images of size 1×20 (column × rows) as illustrated in Fig. 2.7 and trained 500 20-neuron networks using the proposed algorithm.



Figure 2.7: Original image divided into 500 20-dimensional vectors.

The performance of the trained nonlinear line attractor network was tested using distorted versions of the training images obtained by adding 40% salt and pepper noise as shown in Fig. 2.8 (top). Each of these distorted images was divided as shown in Fig. 2.7 and was applied to their corresponding networks. After all the 500 networks converged to their lines of attraction, the output image was reconstructed as shown in Fig. 2.8 (bottom). It can be seen that the networks are capable of removing 40% salt and pepper noise on each image successfully. That is, none of these 500 networks diverged away from their nonlinear lines of attraction.

The recall capability of the nonlinear line attractor is compared with two other classical methods which are capable of performing the task of storing and recalling gray-scale images. Jankowski et al. [59] proposed a multi-valued associative memory in the form of a fully connected neural network composed of multi-state complex-valued neurons. This paper formulated the concept of complex-valued fully coupled neural



Figure 2.8: Images corrupted by 40% salt and pepper noise (top) and their reconstructions obtained by nonlinear line attractor networks (bottom).

network which computes associations based on phase-encoded information. Muezzinoglu et. al. also presented a method in [10] for storing and recalling *N*-dimensional integral memory vector as a fixed point into a complex-valued multi-state Hopfield network. In this experiment, the three images shown in Fig. 2.6 were used for training the networks presented in [59] and [10] similar to the fashion used for training the nonlinear line attractor network (Fig. 2.7). It can be seen from Fig. 2.9 (left) that the method in [59] is unable to reconstruct from the 20% distorted version of the images; instead it converged to a spurious state. Although the method in [10] is able to reconstruct from the 20% distorted version, it can be observed in Fig. 2.9 (middle) that the output image contains a lot of noise (grains). On the other hand, the nonlinear line attractor network is able to recall the image successfully as shown in Fig. 2.9 (right).



Figure 2.9: Lena image corrupted by 20% salt and pepper noise and reconstructed Lena image by [59] (left), by [10] (middle), and by nonlinear line attractor network (right).

Median filtering is known to be one of the most effective methods for filtering out salt and pepper noise. The performance of the nonlinear line network in filtering noisy images is significantly better compared to the performance of median filter (of size 3×3). This can be verified by the result shown in Fig. 2.10 (top and bottom), which illustrates the reconstruction of 20% salt and pepper corrupted images. Although the filtered image using median filter looks almost the same as the original, it becomes very blurry: the face of the man in the Camera-man image and the fur of the baboon in the Baboon image lose its high frequency components (details).



Figure 2.10: Filtered images obtained by the nonlinear line attractor network (middle) and median filter (right).

In table 2.1, the result of the nonlinear line attractor network is compared with the results of the classical approaches [59, 10]. It is seen that the proposed learning algorithm has better root mean square (rms) error compared to methods in [59] and [10]. These results are obtained by averaging the rms error between the original image and the reconstructed image of all 3 test images. The retrieval dynamics of the proposed learning algorithm is also compared with the classical approaches of recurrent neural network [59, 10]. Table 2.1 also provides the average number of iterations for the reconstruction of corrupted images of 10%, 20%, and 40% salt and pepper noise. It can be seen that the

proposed learning algorithm is 8.09 times faster than the method in [59] and 8.91 times faster than method in [10]. Fast retrieval dynamics of the nonlinear line attractor network is due to the fact that the line consists of many fixed points to which the pattern can converge.

	rms error (average)			Convergence
	10%	20%	40%	rate (average)
Method in [59]	12.4066	12.8103	13.5069	25.10 iterations
Method in [10]	1.3984	5.4935	7.4011	27.65 iterations
Nonlinear line attractor network	0.3827	3.3405	6.4140	3.10 iterations

Table 2.1: Results of nonlinear line attractor network compared to methods in [59] and [10].

CHAPTER 3

STABILITY OF THE ATTRACTOR NETWORKS

A novel neural network learning algorithm based on the concept of learning as a nonlinear line of attractor was presented in chapter 2. In this chapter, we examine the stability of the nonlinear line attractor network. Furthermore, a novel nonlinear line attractor network dynamics is designed, in which the network converges when the input pattern is close to one of the stored patterns.

3.1. Stability of the Attractor Network

Let the stability and associability of the linear and nonlinear line attractor network be examined by rewriting eqn. (2.24) as:

$$x_{i}(t+1) = \frac{1}{N} \left[x_{i}(t) + \alpha \Delta x_{ij}(t) \right] + \frac{1}{N} \sum_{\substack{k=1\\k \neq j}}^{N} \left[x_{i}(t) + \alpha \Delta x_{ik}(t) \right]$$

and considering eqns. (2.25) and (2.26) for the following three specific cases:

Case 1. If $\psi_{ij}^{-} \leq [\Lambda_i [x_j(t)] - x_i(t)] \leq \psi_{ij}^{+}$, then $\Delta x_{ij}(t) = [\Phi \{\Lambda_i [x_j(t)]\} - x_i(t)]$ $= x_i(t) - x_i(t) = 0$: no update is required. Case 2. If $[\Lambda_i [x_j(t)] - x_i(t)] < \psi_{ij}^{-}$, then $[\Lambda_i [x_j(t)] - x_i(t)] < 0$ since $\psi_{ij}^{-} \leq 0$. Consequently, $\Delta x_{ij}(t) = [\Phi \{\Lambda_i [x_j(t)]\} - x_i(t)] = \Lambda_i [x_j(t)] - x_i(t) < 0$. Furthermore, $x_i(t) > x_i(t) + \Delta x_{ij}(t)$ implies $\{\Lambda_i [x_j(t)] - x_i(t)\}$ $< \{\Lambda_i [x_j(t)] - [x_i(t) + \alpha \Delta x_{ij}(t)]\} = \{[\Lambda_i [x_j(t)] - x_i(t)] - x_i(t)\}$ $\leq \left[\psi_{ij}(1-\alpha)\right]$. The boundary of convergence is thus determined by α . That is,

when
$$\alpha = \left[1 - \frac{\psi_{ij}}{\Delta x_{ij}(t)}\right], x_i(t) + \alpha \Delta x_{ij}(t) = x_i(t) + \Delta x_{ij}(t) - \psi_{ij}$$
, there are two

conditions to be considered:

Case 2a. If
$$0 < \alpha < \left[1 - \frac{\psi_{ij}}{\Delta x_{ij}(t)}\right]$$
, then $x_i(t) + \alpha \Delta x_{ij}(t)$ is moving down

towards ψ_{ij}^- to the line of attraction.

Case 2b. If
$$\left[1 - \frac{\psi_{ij}}{\Delta x_{ij}(t)}\right] \le \alpha \le 1$$
, then $x_i(t) + \alpha \Delta x_{ij}(t)$ is moving down to

the region of convergence defined by

$$\psi_{ij}^{-} \leq \left[\Lambda_{i} \left[x_{j} \left(t \right) \right] - x_{i} \left(t \right) \right] \leq \psi_{ij}^{+}.$$

Case 3. If $\left[\Lambda_{i}(x_{j}(t))-x_{i}(t)\right] > \psi_{ij}^{+}$, then $\left[\Lambda_{i}[x_{j}(t)]-x_{i}(t)\right] > 0$ since $\psi_{ij}^{+} \ge 0$.

Consequently, $\Delta x_{ij}(t) = \left[\Phi\left\{\Lambda_i\left[x_j(t)\right]\right\} - x_i(t)\right] = \Lambda_i\left[x_j(t)\right] - x_i(t) > 0$.

Furthermore, $x_i(t) < x_i(t) + \Delta x_{ij}(t)$ implies $\left\{ \Delta_i \left[x_j(t) \right] - x_i(t) \right\}$

 $> \left\{ \Lambda_{i} \left[x_{j}(t) \right] - \left[x_{i}(t) + \alpha \Delta x_{ij}(t) \right] \right\}$ $= \left\{ \left[\Lambda_{i} \left[x_{j}(t) \right] - x_{i}(t) \right] \left[1 - \alpha \right] \right\} \ge \left[\psi_{ij}^{+}(1 - \alpha) \right].$ The boundary of convergence

is thus determined by α . That is, when $\alpha = \left[1 - \frac{\psi_{ij}^{+}}{\Delta x_{ij}(t)}\right]$,

 $x_i(t) + \alpha \Delta x_{ij}(t) = x_i(t) + \Delta x_{ij}(t) - \psi_{ij}^+$, there are two conditions to be considered:

Case 3a. If
$$0 < \alpha < \left[1 - \frac{\psi_{ij}^+}{\Delta x_{ij}(t)}\right]$$
, then $x_i(t) + \alpha \Delta x_{ij}(t)$ is moving up

towards ψ_{ii}^+ to the line of attraction.

Case 3b. If
$$\left[1 - \frac{\psi_{ij}^{+}}{\Delta x_{ij}(t)}\right] \le \alpha \le 1$$
, then $x_i(t) + \alpha \Delta x_{ij}(t)$ is moving up to the region of convergence defined by $\psi_{ij}^{-} \le \left[\Lambda_i \left[x_j(t)\right] - x_i(t)\right] \le \psi_{ij}^{+}$.

Based on these possible trajectories of the system, the system is said to be stable and is able to converge to the line of attraction and performs associative recall of multi-valued patterns.

3.2. Divergence Dynamics of the Nonlinear Line of Attraction

In designing a recurrent neural network, it is usually of prime importance to guarantee the convergence in the dynamics of the network. We propose to modify this picture: if the brain remembers by converging to the state representing familiar patterns, it should also diverge from such states when presented with an unknown encoded representation of a visual image. We propose to capture this behavior using the nonlinear line attractor network model. This model encapsulates attractive fixed points scattered in the state space representing patterns with similar characteristics as an attractive curved line. The dynamics of the nonlinear line attractor network is designed such that when the network is able to reach equilibrium (stable), the input is considered as one of the stored patterns. Conversely, when the network is unable to reach equilibrium (unstable), the input is

considered to be dissimilar to the stored patterns and therefore is considered as pattern of another class.

The dynamics of the network is evolved in iteration *t* based on:

$$x_{i}(t+1) = \frac{1}{N} \sum_{j=1}^{N} \Phi\left\{ \Lambda_{i} \left[x_{j}(t) \right] \right\} \text{ for } 1 \le i \le N$$

$$(3.1)$$

where $\Phi(.)$ is the activation function found in eqn. (2.26).

3.3. Instability of the Divergence Dynamics

In asymmetric networks there is no known Lyapunov function guaranteeing convergence to an attractor. Dynamic properties of trajectories in the state space of asymmetric networks can include chaotic and limit cycle behaviors. The symmetry of the synaptic connection matrix has been a constraint from the biological standpoint. Symmetry has been essential for the existence of a landscape picture for the dynamics of the network and asymmetry excluded such a landscape. It can be easily seen that the dynamics of the nonlinear line attractor network presented in eqn. (3.1) does not guarantee global stability. It can only guarantee convergence and stability if the input pattern is sufficiently close to of the trained pattern in its region of convergence described anv as: $\psi_{ij}^{-} \leq \left[\Lambda_{i}\left[x_{j}\left(t\right)\right] - x_{i}\left(t\right)\right] \leq \psi_{ij}^{+}, \forall i, j.$ In fact, Parisi pointed out in [60] that an attempt to implement a process of learning in symmetric artificial neural networks would encounter difficulties because every stimulus will quickly run into an attractor of the spin-glass phase which always accompanies the retrieval states. Consequently every stimulus will be perceived as a familiar pattern. In the next section, we will show that the nonlinear line attractor network presented is specifically designed to operate between stable and

unstable states for the purposed of pattern classification. That is, when the network is able to reach equilibrium (stable), the input is considered as one of the stored patterns (learned pattern). Conversely, if the network is unable to reach equilibrium (unstable), the input is considered to be dissimilar to the stored patterns and therefore is considered as pattern of another class (novel pattern).

3.4. Simulation Results and Discussion

The convergence and divergence dynamics of the recurrent neural network described in section 3.3 is examined by performing pattern classification tasks. The set-up of the first experiment is as follows: Gaussian noise of zero mean μ and variance σ^2 between 0 and 16 (0, 8, and 16 respectively) is added to a training set of 256 pixels drawn from a families of pattern (the color gray) as seen on Fig. 3.1.

The number of iterations for each pixel to converge or diverge is then recorded. Fig. 3.2 shows the result of this experiment. It can be seen that that the recurrent neural network converges in less than four iterations or it diverges. This experiment shows that the proposed neural network has very fast convergence dynamics and it identifies a pattern as another class if the pattern is far away ($\sigma^2 > 10$) from the training data.





Figure 3.2: Number of iterations for the 768 pixels with zero mean μ and variance σ^2 between 0 and 16 to converge/diverge.

Another experiment was also conducted where we have trained two families of color as seen on Fig. 3.3 with each having 256 instances of the pattern using two nonlinear line attractor networks, where each network stores each family.



(a) Families of yellow color



(b) Families of pink color

Figure 3.3: Two families of patterns.

Gaussian noise of zero mean μ and variance σ^2 =8 was then add to the training set to generate testing set. Table 3.1 shows the result of this experiment. We obtained 100% correct detection and less than four percent of miss-classification on the other class. This is due to the fact that these two families of patterns share some common features. For instance, both of these families have (0,0,0) as their red, green, and blue component (R,G,B), which can be observed in Fig. 3.3.

	Families of yellow	Families of pink
	converges (dataset)	converges (dataset)
Families of yellow	100%	2.734%
(Classifier)	(correct detection)	(false positive)
Families of pink	3.515%	100%
(Classifier)	(false positive)	(correct detection)

Table 3.1: Classification of colors using nonlinear line attractor networks.

CHAPTER 4

APPLICATION I: ASSOCIATIVE MEMORY

Humans access memory by its contents and not by where the information is stored in the neural pathways of the brain. We are able to access all aspects of the object (person, event, etc) description by remembering just one or two of the features. For instance, if given even a poor photograph of a person, we are good at reconstructing the person's face accurately. This is very different from a traditional computer where specific facts are located in specific places in computer memory. If only partial information is available about this location, the fact or memory cannot be retrieved at all.

One of the most important functions of our brain is recall from memory. It is difficult to imagine how we could function without both short and long term memory. That is, the absence of short term memory would render most tasks extremely difficult if not impossible. Life would be punctuated by a series of one time images with no logical connection between them. Equally, the absence of any means of long term memory would ensure that we could not learn by past experience. Indeed, many of our impressions depend on remembering our past history.

Our memories function as an associative memory. That is, a memory does not exist in some isolated fashion, located in a particular set of neurons. All memories are in some sense strings of memories. We remember someone in a variety of ways, by the color of their hair or eyes, by the shape of their nose, by their height, by the sound of their voice, or perhaps by the smell of a favorite perfume. Thus memories are stored in association with one another. These different sensory units lie in completely separate parts of the brain. In this chapter, we propose several novel applications using the learning algorithms presented in chapter 2. These applications include color image enhancement, natural color representation, and color constancy problems using associative memories.

4.1. Ratio Rule and Homomorphic Filter for Enhancement of Digital Color Image

A digital color image enhanced using a homomorphic filter provides good dynamic range compression, but it fails in color rendition. In this section, we propose to outfit natural color rendition to a digital color image that is enhanced by the homomorphic filter. The learning algorithm, named Ratio Rule, described in chapter 2, is applied to provide the natural color rendition. Ratio Rule learns to restore color in an image by representing each original pixels' relationships as a line of attraction. The dynamics of the Ratio network is then used as an associative memory for recalling the natural color characteristics of image pixels after homomorphic filtering. Several experiments on benchmark problems show that the new enhancement technique can generate natural color in the images from their original color information.

4.1.1. Problem Statement

High dynamic range is common in scenes that include both dark shadows and bright light sources. These scenes are difficult to represent because their dynamic range greatly exceeds the range of most electronic capturing devices. As a result, the dynamic range of the electronic image is compressed obscuring actual scenic details and/or colors. Image enhancement is a process to improve the appearance of the electronic image as perceived by humans [61, 62] or to render the image more suitable for machine analysis [63].

Homomorphic filtering is a well known and understood technique for image enhancement and/or correction. It simultaneously normalizes the brightness across an image and increases contrast. Although it is possible to enhance a digital color image by applying homomorphic filter to each red, green, and blue (*RGB*) component [61, 62, 64, 65, 66, 67], the resulting image may not be enhanced optimally. Image distortion will result if the relationships among the *RGB* components are not properly recombined after the homomorphic filters are applied to each component.

In this section, we propose a novel technique to provide simultaneous improvement of the dynamic range and color rendition of digital color images. The method is based on using an existence homomorphic processing system [66, 67] and the proposed Ratio Rule learning algorithm [53, 68]. The enhancement process based on our method will be described in sub-section 4.1.2. In sub-section 4.1.3, we will provide some results and discussions of the image enhancement process.

4.1.2. Color Image Enhancement Model

The human visual system can detect the range of light spectrum from 380 nm to 780 nm. Our perception of which color we are seeing is determined by what combination of ρ (red), γ (green), and β (blue) sensors are excited and by how much they are excited [69]. Fig. 4.1.1 shows the spectral sensitivity of the typical human visual system. Similarly, all digital color devices that handle the storage and reproduction of color images do so by storing *RGB* values. Digitally storing an image requires that it first be broken down into a grid of tiny pixels. Each pixel is sampled for the amount of red, green, and blue light present. A direct extension of the grayscale algorithms to the color domain can result in color shift that distorts the correct combination of $\rho\gamma\beta$. Therefore, the Ratio Rule is utilized as an associative memory to stores the original relationship of the *RGB* components of each pixel so that the correct combinations of *RGB* can be properly recalled after applying homomorphic filtering to each channel (red or green or blue) in the image.



Figure 4.1.1: Spectral sensitivity of the typical human visual system.

In the first stage, the relationship of the red, green, and blue components of a pixel is modeled as:

$$w_{(\iota,\kappa)}^{(x,y)} = \frac{1}{P} \sum_{s=1}^{P} \zeta_{(\iota) \ N(s)}^{(x,y)} \left[\zeta_{(\kappa) \ N(s)}^{(x,y)} \right]^{-1} \quad \text{for} \quad 1 \le \iota, \kappa \le n$$
(4.1.1)

where $w_{(i,\kappa)}^{(x,y)}$ represents the synaptic weight from the *i*th neuron to the κ th neuron located at (x, y) position of an image. The network topology is a fully connected recurrent neural architecture where each node represents each color component; in this case, n = 3corresponding to the *RGB* components in each pixel. The symbol $\zeta_{(i) N(s)}^{(x,y)}$ represents the *i*th color component (*R* or *G* or *B*) of the pixel at location (x, y) of an image. The notation *N* in the symbol $\zeta_{(i) N(s)}^{(x,y)}$ denote the corresponding neighbors of the pixel surrounding location (x, y), i.e. $\zeta_{(l) \ N(1)}^{(x,y)}$, $\zeta_{(l) \ N(2)}^{(x,y)}$, $\zeta_{(l) \ N(3)}^{(x,y)}$, ..., $\zeta_{(l) \ N(s)}^{(x,y)}$, with *P* pixels in the neighborhood. Eqn. (4.1.1) finds the degree of similarity between each neuron with other neurons. The Ratio Rule encapsulates the relationship of the pixel by describing its *RGB* components as a line of attraction: no matter how the pixel changes its value, the proportion between *R*, *G*, and *B* is always described in this network. The activation function of each neuron can be found by considering the distance between the approximated output $w_{(t,s)}^{(x,y)}\zeta_{(s)}^{(x,y)}$ and the actual output $\zeta_{(l)}^{(x,y)}$. Mathematically it can be expressed as:

$$\Phi\left\{w_{(\iota,\kappa)}^{(x,y)}\zeta_{(\kappa)}^{(x,y)}\right\} = \begin{cases} \zeta_{(\iota)}^{(x,y)} & \text{if } \psi_{\iota,\kappa}^{-} \leq \left\{\left[w_{(\iota,\kappa)}^{(x,y)}\zeta_{(\kappa)}^{(x,y)}\right] - \zeta_{(\iota)}^{(x,y)}\right\} \leq \psi_{\iota,\kappa}^{+} \\ w_{(\iota,\kappa)}^{(x,y)}\zeta_{(\kappa)}^{(x,y)} & \text{otherwise} \end{cases}$$
(4.1.2)

where

$$\psi_{\iota,\kappa}^{-} = \left\{ \psi_{(1,\iota,\kappa)}^{-}, \psi_{(2,\iota,\kappa)}^{-}, \cdots, \psi_{(\Omega,\iota,\kappa)}^{-} \right\} \quad \text{for} \quad 1 \le \iota, \kappa \le n$$

$$(4.1.3)$$

$$\psi_{\iota,\kappa}^{+} = \left\{ \psi_{(1,\iota,\kappa)}^{+}, \psi_{(2,\iota,\kappa)}^{+}, \cdots, \psi_{(\Omega,\iota,\kappa)}^{+} \right\} \quad \text{for} \quad 1 \le \iota, \kappa \le n$$

$$(4.1.4)$$

and

$$\psi_{(l,\iota,\kappa)}^{-} = \min_{\forall s} \begin{bmatrix} \left\{ \begin{bmatrix} w_{(l,\kappa)}^{(x,y)} \zeta_{(\kappa) \ N(s)}^{(x,y)} \end{bmatrix} - \zeta_{(l) \ N(s)}^{(x,y)} \right\}, \\ \text{in} \\ (l-1)\frac{L}{\Omega} \leq \zeta_{(\kappa) \ N(s)}^{(x,y)} < l\frac{L}{\Omega} \end{bmatrix} \text{ for } 1 \leq l \leq \Omega$$
(4.1.5)

$$\psi_{(l,\iota,\kappa)}^{+} = \max_{\forall s} \begin{bmatrix} \left\{ \begin{bmatrix} w_{(\iota,\kappa)}^{(x,y)} \zeta_{(\kappa) N(s)}^{(x,y)} \end{bmatrix} - \zeta_{(\iota) N(s)}^{(x,y)} \right\}, \\ \text{in} \\ (l-1)\frac{L}{\Omega} \leq \zeta_{(\kappa) N(s)}^{(x,y)} < l\frac{L}{\Omega} \end{bmatrix} \quad \text{for} \quad 1 \leq l \leq \Omega$$
(4.1.6)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

L is the number of levels. A typical digital color image has 256 discrete intensity units in each channel, therefore L = 255. Ω is the number of threshold regions around the line of attraction and $1 \le \Omega \le L$.

Stage two involves performing homomorphic filtering of each *RGB* channel of the image. The homomorphic filter is an illumination-reflectance model. An image, f(x, y), with its corresponding color channel c, viz the red channel, the green channel, or the blue channel, can be described as:

$$f_{c}(x, y) = i_{c}(x, y)r_{c}(x, y)$$
(4.1.7)

where $i_c(x, y)$ is the illumination component and $r_c(x, y)$ is the reflectance component of the image. The homomorphic system for image enhancement is based on the idea that the logarithmic operation separates the illumination $i_c(x, y)$ and reflectance $r_c(x, y)$ from the image. This model assumes that the illumination is the low frequency part of the image and the reflectance is the high frequency part of the image. This information makes it possible to treat these two components separately [67]. The image $f_c(x, y)$ is first disassociated to illumination $i_c(x, y)$ and reflectance $r_c(x, y)$ by:

$$g_{c}(x,y) = \ln\left[i_{c}(x,y)r_{c}(x,y)\right] = \ln\left[i_{c}(x,y)\right] + \ln\left[r_{c}(x,y)\right]$$
(4.1.8)

Then we find the Fourier transform of $\ln [i_c(x, y)]$ and $\ln [r_c(x, y)]$ by:

$$G_{c}\left(u,v\right) = I_{c}\left(u,v\right) + R_{c}\left(u,v\right)$$

$$(4.1.9)$$

where $G_c(u,v)$, $I_c(u,v)$, and $R_c(u,v)$ represent the Fourier transforms of $g_c(x,y)$, $\ln[i_c(x,y)]$, and $\ln[r_c(x,y)]$ respectively. Applying a high pass filter H(u,v) in the frequency domain to enhance changes in reflectance and suppressing changes in illumination gives:

$$\tilde{G}_{c}\left(u,v\right) = I_{c}\left(u,v\right)H\left(u,v\right) + R_{c}\left(u,v\right)H\left(u,v\right)$$
(4.1.10)

Taking the inverse Fourier transform yield

$$\tilde{g}_{c}(x,y) = \tilde{i}_{c}(x,y) + \tilde{r}_{c}(x,y)$$
(4.1.11)

whand $\tilde{\mathfrak{T}}_{-1}^{-1} \left[R_c^{(u,v)} H^{(u,v)} \right]^{d} \tilde{r}_c^{(x,y)}$ represent $\mathfrak{T}_{-1}^{-1} \left[\tilde{G}_c^{(u,v)} \right], \mathfrak{T}_{-1}^{-1} \left[I_c^{(u,v)} H^{(u,v)} \right]$, and $\mathfrak{T}_{-1}^{-1} \left[R_c^{(u,v)} H^{(u,v)} \right]$ respectively. Reversing the logarithm effect utilizing the exponential function:

$$\Upsilon_{c}(x,y) = \exp\left[\tilde{g}_{c}(x,y)\right] = \exp\left[\tilde{i}_{c}(x,y)\tilde{r}_{c}(x,y)\right]$$
(4.1.12)

where $\Upsilon_{c}(x, y)$ is the image after enhancement.

The relationships between the *RGB* channels of the image become distorted after application of homomorphic filtering in stage 2. Stage 3 involves adjusting the relationship between the *RGB* components using the memory matrices obtained using eqn. (4.1.1) and activation functions obtained using eqn. (4.1.2) for associative recall. The dynamics for recalling is computed iteratively t as:

$$\zeta_{(\iota)}^{(x,y)}(t+1) = \frac{1}{n} \sum_{\kappa=1}^{n} \left[\zeta_{(\iota)}^{(x,y)}(t) + \upsilon \Delta \zeta_{(\iota,\kappa)}^{(x,y)}(t) \right] \quad \text{for} \quad 1 \le \iota \le n$$
(4.1.13)

where v is the update rate ranging $0 \le v \le 1$ and $\Delta \zeta_{(\iota,\kappa)}^{(x,y)}(t)$ is calculated by:

$$\Delta \zeta_{(\iota,\kappa)}^{(x,y)}(t) = \left\{ \Phi \Big[w_{(\iota,\kappa)}^{(x,y)} \zeta_{(\kappa)}^{(x,y)}(t) \Big] - \zeta_{(\iota)}^{(x,y)}(t) \right\}$$
(4.1.14)

until the network becomes stable.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.1.3. Results and Discussion

Several experiments have been conducted to validate the performance of the proposed method. The criterion for evaluation of the enhancement algorithm is judged by the dynamic range compression and the color rendition of the image. The proposed method has been used to perform digital color image enhancement in more than 100 images. These images were selected for testing the dynamic range compression and color restoration properties of the proposed method. Fig. 4.1.2 shows a typical result of the enhancement. It can be seen that while the homomorphic filter provides excellent dynamic range compression, it lacks the color representation: the original color is corrupted after the enhancement. The result from the Ratio network provides excellent color representation to the scene. This experiment shows that it is necessary to apply the Ratio Rule for color restoration.

One major drawback of the iterative method is that it takes too long for the system (i.e. recurrent network) to converge to the solution. In this experiment, we examined the convergence dynamics of the Ratio network. The setup of the experiment was as follows: Gaussian noise of zero mean μ and standard deviation σ between 0 and 32 was added to 1000 pixels of training samples. The number of iterations for each pixel to converge was then recorded. Fig. 4.1.3 shows the result of this experiment in the form of a histogram. It shows that the Ratio network converged in one iteration most of the time. This experiment shows that the Ratio network has very fast convergence dynamics. This is a significant improvement compared to other recurrent networks [59, 10, 46], which take an average of more than ten iterations to converge to a fixed point [59, 10, 46] and which are unable to associate multi-valued patterns [46].



(a) Original image



(b) Homormorphic filtering



(c) Proposed method

Figure 4.1.2: Result of grayscale enhancement with and without Ratio learning algorithm.



Figure 4.1.3: Convergence characteristics.

Fig. 4.1.4 shows the convergence property of a family of color to illustrate the learning and recall capability of the Ratio memory. Instances of the purple color and their corresponding *RGB* combinations in the *RGB* space are illustrated in Fig. 4.1.4a. It can be seen that the purple color resided in the *RGB* color space as a line. The Ratio Rule can be applied to encapsulate the essence of purple as a line of attraction. That is, no matter how the pixel changes its value, the relationship between them is always described in the Ratio network. After training the Ratio network, distorted versions of the training pixels were obtained by adding Gaussian noise of zero mean $\mu = 0$ and variance $\sigma^2 = 500$ as shown in Fig. 4.1.4b. After the network converges to their line of attraction, the reconstructed pixels were obtained and are shown in Fig. 4.1.4c. It can be seen that networks are capable of correctly recalling the color successfully. That is, the network didn't diverge away from the line of attraction. Fig. 4.1.5 shows quantitative measures of the correct color recall when testing the Ratio network with 10 families of color, including the one shown in Fig. 4.1.4a. Each families of color is distorted with Gaussian



(b) Corrupted color pattern with Gaussian noise ($\mu = 0$ and $\sigma^2 = 500$)



(c) Recalled color

Figure 4.1.4: Families of color, corrupted color, and recalled color.

noise of zero mean $\mu = 0$ and variances $\sigma^2 = 100$, $\sigma^2 = 200$, $\sigma^2 = 300$, $\sigma^2 = 400$, and $\sigma^2 = 500$. It is shown consistently that the Ratio network is able to converge in less than 1 rms error. These results were obtained by comparing the actual color and the restored color (of the Gaussian noise). The residual rms error is due to the threshold function because the network has approximated the correct color in its stable state and prevents the color to jump from one state to another state. In addition, it can also be seen that the network converged in one iteration consistently. This experiment shows that the Ratio network can indeed be used to perform color restoration (or rendition).



Figure 4.1.5: Root mean square (rms) error with 10 families of colors with different variances.

4.2. Natural Color Representation using Ratio Learning Algorithm for Gray Level Enhancement of Digital Color Images

In this section, we proposed a general framework to apply gray level image enhancement algorithm to the color domain using single layer fully connected recurrent neural networks trained using the Ratio learning algorithm. The Ratio Rule learns to produce natural color rendition by representing the original pixels relationship of an image as a line of attraction in the state space. Its dynamic is then used for recalling the original color characteristics of image pixels by iteratively converging to the learned stable state after the gray level enhancement. Experiments show that the proposed general framework provide natural color rendition using Ratio Rule with existing gray level enhancement algorithms.

4.2.1. Problem Statement

The human eye has a remarkable dynamic range that enables it to detect subtle contrast variations and interpret scenes under a large variety of illumination conditions [70, 71]. Conversely, images captured using cameras usually lose their dynamic range since the images are digitized to a much narrower dynamic range. Physical limitations exist in the sensor arrays of image capturing devices, such as CCD and CMOS cameras. Often, the devices cannot represent scenes well that have both very bright and dark regions. The sensor cells are commonly compensated with the amount of saturation from bright regions, fading out the details in the darker regions. For instance, most images are digitized to 8 bit gray level for each red, green, and blue band [61]. Consequently, images captured in a scene which contains both very bright and very dark regions result in a trade off between enhancing the dark spot at the cost of saturating the bright spot or keeping

the bright spot at the cost of losing the dark spot. The well known method used for providing dynamic range compression is using Gain/offset correction, non-linear point transform, histogram equalization, and homomorphic filtering to the original image. However, these methods focused on enhancing gray level images and a direct extension of these processes into the color domain results in degraded images. Image distortion will result if the red, green, and blue components are not properly recombined after the gray level image enhancement algorithms are applied.

We propose a general framework to apply gray level image enhancement algorithm to the color domain using an associative memory to recall/restore the color relationship to produce images with natural color which exists in the original image. The associative memory we have used is a single layer fully connected recurrent neural network trained using the Ratio learning algorithm [72]. Ratio Rule learns to produce natural color rendition by representing the original pixels relationship of an image as a line of attraction in the state space. Its dynamic is then used for recalling the original color characteristics of image pixels by iteratively converging to the learned stable state after the gray level enhancement.

4.2.2. Natural Color Representation

Our method of improving color images using gray level enhancement algorithm can be described as three stages: color characterization, enhancement, and color balancing. In the first stage, the relationship of the red, green, and blue components of a pixel is modeled using Ratio Rule learning algorithm [72] as:

$$w_{(\iota,\kappa)}^{(x,y)} = \frac{1}{P} \sum_{s=1}^{P} \frac{\zeta_{(\iota) \ N(s)}^{(x,y)}}{\zeta_{(\kappa) \ N(s)}^{(x,y)}} \quad \text{for} \quad 1 \le \iota, \kappa \le n$$
(4.2.1)

where $w_{(\iota,x)}^{(x,y)}$ represents the synaptic weight from the ι^{th} neuron to the κ^{th} neuron located at (x, y) position of an image. n is 3 in this application representing RGB components in each pixel. The symbol $\zeta_{(\iota) N(x)}^{(x,y)}$ represents the ι^{th} color component (R or G or B) of the pixel at location (x, y) of an image. The notation N in the symbol $\zeta_{(\iota) N(x)}^{(x,y)}$ denoted the corresponding neighbors of the pixel surrounding location (x, y) with P pixels in the neighborhood. Eqn. (4.2.1) finds the degree of similarity between each neuron with other neurons. Ratio Rule encapsulates the relationship of the pixel by describing its RGBcomponents as a line of attraction: no matter how the pixel changes its value, the proportion between R, G, and B is always described in this network. The activation function of each neuron can be found by considering the distance between the approximated output $w_{(\iota,x)}^{(x,y)} \varepsilon_{(x,y)}^{(x,y)}$ and the actual output $\zeta_{(\iota)}^{(x,y)}$. Mathematically it can be expressed as:

$$\Phi\left\{w_{(\iota,\kappa)}^{(x,y)}\zeta_{(\kappa)}^{(x,y)}\right\} = \begin{cases} \zeta_{(\iota)}^{(x,y)} & \text{if } \psi_{\iota,\kappa}^{-} \leq \left\{\begin{bmatrix}w_{(\iota,\kappa)}^{(x,y)}\zeta_{(\kappa)}^{(x,y)}\\ -\zeta_{(\iota)}^{(x,y)}\end{bmatrix}\right\} \leq \psi_{\iota,\kappa}^{+} \\ w_{(\iota,\kappa)}^{(x,y)}\zeta_{(\kappa)}^{(x,y)} & \text{otherwise} \end{cases}$$
(4.2.2)

where

$$\psi_{i,\kappa}^{-} = \begin{cases} \psi_{(1,\iota,\kappa)}^{-} & \text{if } 0 \leq \zeta_{(\kappa)}^{(x,y)} < \frac{L}{\Omega} \\ \psi_{(2,\iota,\kappa)}^{-} & \text{if } \frac{L}{\Omega} \leq \zeta_{(\kappa)}^{(x,y)} < \frac{2L}{\Omega} \\ \vdots \\ \psi_{(\Omega,\iota,\kappa)}^{-} & \text{if } \frac{(\Omega-1)L}{\Omega} \leq \zeta_{(\kappa)}^{(x,y)} < L \end{cases}$$

$$(4.2.3)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

54

and

$$\psi_{\iota,\kappa}^{+} = \begin{cases} \psi_{(1,\iota,\kappa)}^{+} & \text{if } 0 \leq \zeta_{(\kappa)}^{(x,y)} < \frac{L}{\Omega} \\ \psi_{(2,\iota,\kappa)}^{+} & \text{if } \frac{L}{\Omega} \leq \zeta_{(\kappa)}^{(x,y)} < \frac{2L}{\Omega} \\ \vdots \\ \psi_{(\Omega,\iota,\kappa)}^{+} & \text{if } \frac{(\Omega-1)L}{\Omega} \leq \zeta_{(\kappa)}^{(x,y)} < L \end{cases}$$

$$(4.2.4)$$

where

$$\psi_{(l,\iota,\kappa)}^{-} = \min_{\forall s} \begin{bmatrix} \left\{ \left[w_{(\iota,\kappa)}^{(x,y)} \zeta_{(\kappa) \ N(s)}^{(x,y)} \right] - \zeta_{(\iota) \ N(s)}^{(x,y)} \right\}, \\ \text{in the region} \\ (l-1)\frac{L}{\Omega} \le \zeta_{(\kappa) \ N(s)}^{(x,y)} < l\frac{L}{\Omega} \end{bmatrix} \text{ for } 1 \le l \le \Omega$$

$$(4.2.5)$$

and

$$\psi_{(l,\iota,\kappa)}^{+} = \max_{\forall s} \begin{bmatrix} \left\{ \begin{bmatrix} w_{(\iota,y)}^{(x,y)} \zeta_{(\kappa) N(s)}^{(x,y)} \end{bmatrix} - \zeta_{(l) N(s)}^{(x,y)} \right\}, \\ \text{in the region} \\ (l-1)\frac{L}{\Omega} \leq \zeta_{(\kappa) N(s)}^{(x,y)} < l\frac{L}{\Omega} \end{bmatrix} \text{ for } 1 \leq l \leq \Omega$$
(4.2.6)

L is the number of levels. A typical digital color image has 256 discrete intensity unit in each channel, therefore L = 255. Ω is the number of threshold regions around the line of attraction and $1 \le \Omega \le L$.

Stage two involves performing dynamic range compression using transforms such as Gain/offset correction, non-linear point transform, histogram equalization, and homomorphic filtering to the each *RGB* channel of the original image. The output of such filters can be described by:

$$\Upsilon_{c}(x,y) = T\left[I_{c}(x,y)\right]$$
(4.2.7)

55

where $I_c(x, y)$ is the *c* channel of the image *I* at location (x, y), *T*[.] represents the transformation function., and γ is the image after enhancement.

The relationships between the *RGB* channels of the image become distorted after application of filtering in stage 2. Stage 3 involves adjusting the relationship between the *RGB* components using the memory matrices obtained using eqn. (4.2.1) and activation functions obtained using eqn. (4.2.2) for associative recall. The dynamics for recalling is computed iteratively t as:

$$\zeta_{(\iota)}^{(x,y)}(t+1) = \frac{1}{n} \sum_{\kappa=1}^{n} \left[\zeta_{(\iota)}^{(x,y)}(t) + \upsilon \Delta \zeta_{(\iota,\kappa)}^{(x,y)}(t) \right] \quad \text{for} \quad 1 \le \iota \le n$$
(4.2.8)

where v is the update rate ranging $0 < v \le 1$ and $\Delta \zeta_{(\iota,\kappa)}^{(x,y)}(t)$ is calculated by:

$$\Delta \zeta_{(\iota,\kappa)}^{(x,y)}(t) = \left\{ \Phi \left[w_{(\iota,\kappa)}^{(x,y)} \zeta_{(\kappa)}^{(x,y)}(t) \right] - \zeta_{(\iota)}^{(x,y)}(t) \right\}$$
(4.2.9)

until the network becomes stable. Fig. 4.2.1 illustrates our color image enhancement process.


Figure 4.2.1: Color characterization, enhancement, and color balancing system.

4.2.3. Comparison

We have compared the gain/offset correction, non-linear point transform, histogram equalization, and homomorphic filtering with and without our color image enhancement process: color characterization, enhancement, and color balancing.

4.2.3.1. Gain/Offset Correction

The gain/offset correction is a linear transformation which transforms the scene to completely fill the dynamic range of the display medium by stretching the dynamic range of the images. Gain/offset is one of the most common methods of enhancing an image.

For a display medium with the maximum dynamic range χ_{max} , the gain offset correction is describable as:

$$\Upsilon_{c}(x,y) = \frac{\chi_{\max}}{\tau_{\max} - \tau_{\min}} \Big[I_{c}(x,y) - \tau_{\min} \Big]$$
(4.2.10)

where τ_{\max} and τ_{\min} are $\max[I_c(x, y)]$ and $\min[I_c(x, y)]$ respectively.

4.2.3.2. Non-linear Transform

Non-linear transforms such as the gamma, the logarithm function, and the power-law function can be used for providing dynamic range compression to the original image. The output of such functions can be calculated as:

$$\Upsilon_{c}(x,y) = I_{c}(x,y)^{\gamma} \qquad (4.2.11)$$

$$\Upsilon_{c}(x, y) = a \log \left[I_{c}(x, y) \right]$$
(4.2.12)

$$\Upsilon_{c}(x,y) = aI_{c}(x,y)^{\theta} \qquad (4.2.13)$$

respectively. Where *a* is a scaling factor such that the maximum value of the enhanced image $\Upsilon_c(x, y)$ becomes 255. The basis number, γ and θ , is used depends on the desired degree of compression of the dynamic range.

4.2.3.3. Histogram Equalization

Histogram modeling techniques provide a sophisticated method for modifying the dynamic range and contrast of an image by altering that image such that its intensity histogram has a desired shape. Histogram equalization is a global technique that works well for a wide variety of images. This technique is based on the idea of remapping the histogram of the scene to a histogram that has a near-uniform distribution. Histogram equalization can be described by three simple operations:

1). Histogram formation.

2). New intensity values calculation for each intensity levels.

3). Replace the previous intensity values with the new intensity values.

This results in reassigning bright regions to darker values and dark regions to brighter values.

4.2.3.4. Homomorphic Filtering

The homomorphic system for image enhancement is based on the idea that the logarithmic operation separates $i_c(x, y)$ and $r_c(x, y)$ from the image. The homomorphic filter is an illumination-reflectance model. The model is described as:

$$I_{c}(x, y) = i_{c}(x, y)r_{c}(x, y)$$
(4.2.14)

where

 $I_c(x, y)$ is the image intensity function $i_c(x, y)$ is the illumination component

 $r_c(x, y)$ is the reflectance component

The model assumes that $i_c(x, y)$ is the primary contributor to the dynamic range of the image and the reflectance component $r_c(x, y)$ represents the details of an object. In another word, the model described in eqn. (4.2.14) assumes that the illumination is the low frequency part of the image and the reflectance is the high frequency part of the image. With these assumptions of a slower and a faster varying component, the low pass and high pass information make it possible to treat the two components separately.

4.2.4. Results and Discussion

The resulting images with gain/offset correction, non-linear transform, histogram equalization and homomorphic filtering are illustrated in Fig. 4.2.2. Shown on the first column is the original images, second column is the enhanced images without Ratio learning algorithm, and third column is the images with color balancing by Ratio Rule. The gain/offset correction did not contribute trivial enhancement for the type of nonuniform images where the min and max of the pixel intensities are close to 0 and 255 respectively. The image enhanced by gamma correction is shown in Fig. 4.2.2b (middle) with obvious color distortion. The natural color of the image is restored after color balancing. The image with histogram equalization is shown in Fig. 4.2.2c (middle) where the normal distribution of the pixel intensity results with pale appearance and overexposure of the image. However the learned relationship of the neural network restores back its natural color as illustrated in Fig. 4.2.2c (right). The image boosted by homomorphic filter brings out more detail hidden in the shadows. As it can be seen in Fig. 4.2.2d (middle) that the image appeared similar to the one with non-linear transform; it looks very vivid after restoring the ratio between *RGB* channels. The proposed algorithm works very well in general to retain the relationship of the color components presented in this category of distorted images.





(a) Gain/offset correction





(b) Non-linear transform









(c) Histogram equalization



(d) Homomorphic filtering





Figure 4.2.2: Color image enhancement (left) with (right) and without (middle) color characterization, enhancement, and color balancing.

4.3. On Creating a Color Manifold for Color Characterization and Balancing using a Nonlinear Line of Attraction for Nonlinear Image Enhancement

The human eye has a remarkable dynamic range that enables it to detect subtle contrast variations and interpret scenes under a large variety of illumination conditions. Conversely, images captured using a camera usually lose their dynamic range since the images are digitized to a much narrower dynamic range. For instance, most images are digitized to 8 bit gray level for each red, green, and blue band. Consequently, images captured in a scene in which both very bright and very dark regions are present result in a trade off between enhancing the dark spot at the cost of saturating the bright spot or keeping the bright spot at the cost of losing the dark spot. This is known as dynamic range problem.

The Retinex approach was introduced three decades ago as a simple but effective model of the human vision [73]. Its aim was to take into account the elements which influence our visual perception. The basic concept in the Retinex method is to separate the illumination and reflectance components of the image. It is assumed that the available luminance data are the products of illumination and reflectance. With the Retinex method, the dynamics of an image can be modified, and the visibility of details in areas characterized by different lighting conditions can be improved. Multi-scale Retinex method is a multi-scale embodiment of the Retinex algorithm which is suggested as a method to bridge the gap between what a camera sees and what a human sees [74-76]. It is able to provide image reproduction which are very similar to what the human viewer would have seen, whether they were present when the picture was taken [77].

4.3.1. Problem Statement

Image enhancement is a process by which acquired image data values are manipulated. Such manipulation is typically directed to improve the appearance of the data as perceived by humans, and/or to render the data to make it more suitable for machine analysis. Many image enhancement processes have focused on the improvement of graylevel images, but a direct extension of these processes into the color domain results in degraded or suboptimal images. Such image enhancement methods should be optimized for use in conjunction with color images, and should not merely constitute the reapplication of a gray-level method to the color domain. Although it is possible to enhance a color image by applying existing gray-level image enhancement algorithms to each color component (red, green, and blue) individually, the resulting image may not be enhanced optimally. Image distortion will result if the red, green, and blue components are not properly recombined after the monochromatic image enhancement algorithms are applied. That is, image distortion will also result if the red, green, and blue components are processed independently without considering correlation among the components.

4.3.2. Nonlinear Image Enhancement using Hyperbolic Tangent Function

The proposed method of improving color images consists of three stages: color characterization, enhancement, and color balancing. In the first step of the enhancement, the color manifold of each pixel is created by mapping that pixel into the color space along with a set of its neighbors using the nonlinear line attractor network. It is observed that the color of a particular pixel along with its neighborhood always forms a nonlinear pipeline in the feature space. Fig. 4.3.1 shows this phenomenon by illustrating the

62

instances of red, green, and blue components of a pixel at a particular image location along with its corresponding neighbors in the *RGB* color space.



Figure 4.3.1: Color manifold.

Fig. 4.3.2 shows the relationship between the red-green, red-blue, green-blue, etc components of a pixel using the nonlinear line attractor network. It can be seen that nonlinear line attractor network encapsulates the relationship of the pixels by describing its components (red, green, and blue) as a nonlinear transfer function. That is, no matter how the pixel changes its value, the relationship between them is always described in this function.



Figure 4.3.2: Memory of the attractor network.

Step two of the process involves utilizing the hyperbolic tangent functions for dynamic range compression. Hyperbolic tangent function is used for the reason of overcoming the natural loss in perceived lightness contrast that results when performing dynamic range compression. We have developed an enhancement strategy that will perform the range compression while maintaining the image details. The proposed solution is to develop the hyperbolic tangent functions that are tunable based on the statistical characteristics of the image. The proposed enhancement functions based on the hyperbolic tangent function will enhance the dark part of the image while preserving the light part of the image as described by:

$$R_{(x,y)}^{+} = \max\left\{255 \times \left[\frac{2}{1 + \exp\left(\frac{-R_{(x,y)}}{\varphi \mu_{\text{red}} + \gamma \psi_{(x,y)}^{\text{red}}}\right)} - 1\right], R_{(x,y)}\right\}$$
(4.3.1)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$G_{(x,y)}^{+} = \max\left\{255 \times \left[\frac{2}{1 + \exp\left(\frac{-G_{(x,y)}}{\varphi\mu_{\text{green}} + \gamma\psi_{(x,y)}^{\text{green}}}\right)^{-1}}\right], G_{(x,y)}\right\}$$
(4.3.2)
$$B_{(x,y)}^{+} = \max\left\{255 \times \left[\frac{2}{1 + \exp\left(\frac{2}{\varphi\mu_{\text{blue}} + \gamma\psi_{(x,y)}^{\text{blue}}}\right)^{-1}}\right], B_{(x,y)}\right\}$$
(4.3.3)

where $R_{(x,y)}^+$, $G_{(x,y)}^+$, and $B_{(x,y)}^+$ are the enhanced value of $R_{(x,y)}$, $G_{(x,y)}$, and $B_{(x,y)}$ respectively, "max" chooses the maximum of the processed pixel and the original pixel, φ controls the contribution of the global mean μ where $0 \le \varphi \le 1$, and γ controls the contribution from the local mean $\psi_{(x,y)}$ where $0 \le \gamma \le 1$. The reason for using the max function is to prevent the value of the pixel to decrease in luminance. This can happen when the global and/or the local mean of the image is very high and therefore the slope of the hyperbolic tangent function will be really low. The global means is the statistical mean of the images and it is expressed as:

$$\mu_{\rm red} = \sum_{x=1}^{N_1} \sum_{y=1}^{N_2} R_{(x,y)}$$
(4.3.4)

$$\mu_{\text{green}} = \sum_{x=1}^{N_1} \sum_{y=1}^{N_2} G_{(x,y)}$$
(4.3.5)

$$\mu_{\text{blue}} = \sum_{x=1}^{N_1} \sum_{y=1}^{N_2} B_{(x,y)}$$
(4.3.6)

where $N_1 \times N_2$ is the size of the image. The local mean of an image is defined by the relationship of the value of a particular pixel with respect to its neighbors. The local mean

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

of each pixel is calculated based on the center surrounded property of perceptual field and perceptual processes of human vision. The form of the surround function we used is Gaussian because it provides good dynamic range compression over a wide range of environment [76]. Consequently, the local mean of the image is calculated by:

$$\psi_{(x,y)}^{\text{red}} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} R_{(x+i-1,y+j-1)} H_{(i,j)}$$
(4.3.7)

$$\psi_{(x,y)}^{\text{green}} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} G_{(x+i-1,y+j-1)} H_{(i,j)}$$
(4.3.8)

$$\psi_{(x,y)}^{\text{blue}} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} B_{(x+i-1,y+j-1)} H_{(i,j)}$$
(4.3.9)

where $n_1 \times n_2$ is the size of the local region of the image and the filter *H* is defined by:

$$H_{(x,y)} = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right]$$
(4.3.10)

The size of this filter presents a trade off between the dynamic range compression and color rendition of the image. A smaller size filter will yield larger dynamic range compression but causes the image to lose its color. Conversely, a larger size filter will yield better color rendition but the shadow of the image will remain untouched.

Overall in the enhancement process, the μ and ψ parameters control the lightness of the image. Decreasing μ and ψ would increase the slope of the hyperbolic tangent function thereby increasing the brightness of the pixel. By controlling the contribution of μ and ψ using the parameters φ and γ , it is possible to tailor the amount of lighting for the overall enhancement. These four values $-\mu$, ψ , φ , and γ - together control the curvature of the hyperbolic tangent function. This means that when the processing image is dark, the μ and ψ would be small and therefore the curvature of

the hyperbolic tangent function will be steep and this will help the darker pixel to have brighter value. Conversely, when the image is bright, the hyperbolic tangent function will have a small slope and therefore the brighter pixel will either increase by a small amount or stay the same. The relationship of the red, green, and blue components of the pixel may be distorted by enhancing each channel of *RGB* components separately without considering the correlation-ship among themselves. Therefore in a color balancing process, the nonlinear line attractor network can be used to recall their original properties by converging the pixel to its original characteristic with the proper proportion of red, green, and blue components. The balancing process can be done by using eqn. (2.24) to (2.26). Fig. 4.3.3 illustrates the re-tuning of the relationships of the *RGB* components using a color manifold trained by a nonlinear line attractor network.



Figure 4.3.3: Re-tuning of the pixel.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

It can be observed that although the intensities of the *RGB* components are increased, the relationship is still perfectly preserved in the attractive nonlinear line. Consequently, the balancing process is able to represent the bright instance of a pixel with its darker instance of the pixel without corrupting the relationship of its *RGB* components.

4.3.3. Simulation Results and Discussion

Several experiments have been conducted to test and evaluate the performance of the color characterization, enhancement, and color balancing process which is based on the hyperbolic tangent function and the nonlinear line attractor network. The criterion for evaluation of the enhancement algorithm is judged by the dynamic range compression and the color rendition of the image.

Fig. 4.3.4(a) show examples of images that contain very uneven lighting conditions (i.e. extremely high dynamic range) with very dark shadow. These images were captured by the observer specifically to test the actual observed scene by the human and by the camera. From these images, we can see that the camera is unable to capture and represent the image using its limited dynamic range: a photograph of the same scene will either show the shadow as too dark, or the bright area as over-exposed. On the other hand, human can see details both in deep shadows and in nearby highly illuminated areas. The human eyes are able to adopt and accommodate these conditions. This is because humans can perform dynamic range compression to the scene. Human vision strongly compresses visual information across wide range of illumination. As a result, we can see the scenes perfectly without any problem. This experiment tests the color characterization, enhancement, and balancing algorithm for this uneven lighting condition. It can be seen



(a) Original Images



(b) Gray scale enhancement without color characterization and color balancing



(c) Color characterization, enhancement, and balancing

Figure 4.3.4: Result of Gray scale enhancement with and without color characterization and color balancing.

from Fig. 4.3.4(c) that our proposed enhancement algorithm can simultaneously provide good dynamic range compression and color rendition to the scene. The enhancement process is able to provide image reproductions which are very similar to what the human would have seen, whether they were present when the picture was taken. While the hyperbolic tangent function results in Fig. 4.3.4(b) provide good dynamic range compression, it lacks color rendition. Therefore, color characterization and color balancing process using the nonlinear line attractor network is necessary for graceful reproduction of the actual scene by transforming the scene to its natural color.

One major drawback of the iterative method is that it takes too long for the system (i.e. recurrent network) to converge to the solution. In this experiment, we examined the convergence dynamics of the nonlinear line attractor network. The setup of the experiment is as follows: ten images of size 800×600 are enhanced using the proposed color characterization, enhancement, and balancing process. The number of iterations for each pixel (4,800,000 pixels = $10 \times 800 \times 600$) to converge is recorded. Fig. 4.3.5 shows the result of this experiment in term of a probability distribution. It shows that the nonlinear line attractor network will most probably be converged in 3 iterations. It can also be calculated from the probability distribution that the nonlinear line attractor network takes an average of 3.7 iterations to converge. This is a significant improvement compared to other recurrent networks such as the Hopfield network and complex multivalued multi-state neural network [10] in terms of the rate of convergence. These networks take an average of 10 iterations or more to converge to a solution (for performing pattern association). This is due to the fact that nonlinear line attractor

network converges to a line of attraction whereas the Hopfield network converges to a point and it takes longer time (or many iterations) to converge to an attractive fixed point.



Figure 4.3.5: Convergence characteristics.

The proposed algorithm has also been tested on a set of high dynamic range color images and the results have been compared with two other methods, namely McCann Retinex [78] and multi-scale Retinex with color restoration (MSRCR) [61]. Retinex theory of color vision was originally developed by Land. Major conclusions about human color perception were drawn from the experiments performed by Land. Because Retinex theory is a simplified version of human color vision processing, it is adapted for computational image rendering models. Retinex based algorithms can be classified into three classes. The first class is path based algorithms where a new pixel value depends on the computation of ratios and products along paths in the image. The second class includes all algorithms that compute new pixel values depending on the recursive comparison of surrounding pixels. The third class includes the center/surround versions of Retinex. McCann Retinex belongs to the second class of the Retinex methods that computes the pixel values using a recurrent iterative formula. Rahman et al. proposed a different surround-based approach. Their method computes the enhanced image using a weighted sum of single-scale Retinex and a color restoration factor [61].

Several experiments have been conducted and some typical results of these comparisons (the proposed method, method in [67] and method in [61]) are shown in Fig. 4.3.6 and Fig. 4.3.7. It can be seen that the proposed algorithm outperforms the other two methods in dealing with high dynamic range images. That is, the proposed method provides good dynamic range compression and good color rendition. It is able to retrieve details in the shadow portion of the image while maintaining the light part of the image. On the other hand, method in [78] and method in [61] are unable to eliminate the shadow in the image. For example, the face in Fig. 4.3.6 and the glass in Fig. 4.3.7 still remain relatively dark after the enhancement. Moreover, method in [78] and method in [61] have also introduced the undesirable de-saturation effect on the image color. The chromaticity and hue of each pixel have changed. It is important to note that although with an appropriate scale (filter size), method [61] can reproduce arbitrary amount of dynamic range compression. The scale represents a trade off between dynamic range compression and color rendition. That is, a smaller scale can produce better dynamic range compression whereas a larger scale can produce better color rendition. However, if the scale is too small, it will remove all intensity difference in the image and produce an unnatural looking image. Nevertheless, applying the method in [61] with image with high dynamic range often gives poor result at sharp shadow edge. That is, although the shadow





(c) Method in [78]

(d) Method in [61]

Figure 4.3.6: Comparison of the proposed method with McCann99 Retinex and Multiscale Retinex.



Figure 4.3.7: Comparison of the proposed method with McCann99 Retinex and Multiscale Retinex.

region of the image is brightened, the edges itself become dark area between two light areas.

Image enhancement is the process by which we try to improve an image so that it looks subjectively better. Image enhancement quality is difficult to assess. There are no standard methods to specify how the enhanced image should look, but one can tell whether the image has been improved or not by looking at its histogram. Histogram provides a graphical representation of the tonal values for a given image. A pixel tonality is expressed as number between 0 and 255, in which 0 being pure black and 255 pure white. A histogram gives useful information about an image. Correct interpretation of the histogram can help perform visual judgment to an image. Histograms can be adopted as a tool for visual quality assessment of different image enhancement algorithms performed to an image. Histograms can be quite different depending on the content of the image.



Figure 4.3.8: Division of the histogram to five zones.

A histogram of a high key image with a majority of the content being very bright will produce a histogram that has most of the histogram graph located from the center to the right of center. A low key image with lots of dark and shadow areas will produce a histogram graph that is mostly center and left of center. Based on this observation, we can attempt to analyze and compare the results of different enhancement algorithms. First, let's divide the pixel tonality into five regions representing the dynamic range of the digital image and let's label each of these regions as Shadow, Dark, Medium, Light, and Bright as illustrated in Fig. 4.3.8. The following discussion will show the analysis of image quality comparison based on these five histogram regions. Fig. 4.3.9(a) and Fig. 4.3.10(a) show the histogram of the original images in Fig. 4.3.6(a) and Fig. 4.3.7(a) respectively. It can be observed that the histograms of these images fall mainly in the shadow and light regions. This is expected because the image contains scenes with very high dynamic range (the image contain both very bright elements and very dark elements). After performing an enhancement using the proposed algorithm, the histograms (Fig. 4.3.9(b) and Fig. 4.3.10(b)) shows good distribution of tones covering the five zones – from deep shadows on the left to some bright regions on the right.

These histograms fit well within the specific five regions of the dynamic range. As a result, the images shown in Fig. 4.3.6(b) and Fig. 4.3.7(b) have excellent dynamic range. On the other hand, histograms of method [78], as shown in Fig. 4.3.9(c) and Fig. 4.3.10(c), leave its light and highlight tones relatively stable while moving its shadows tones to the dark tones. Also notice that the medium zone of the histogram has fewer values compared to other regions. The enhancement algorithm proposed in [78] does not make good use of the dynamic range of the pixel tonality. Therefore, method in [78] is unable to provide good dynamic range to Fig. 4.3.9(d) and Fig. 4.3.10(d), redistributed its pixel tonality across different regions in the histogram. The distribution of the enhanced image (Fig. 4.3.9(d) and Fig. 4.3.10(d)) and the distribution of the original image



Figure 4.3.9: Histograms of images in Figure 4.3.6.



Figure 4.3.10: Histograms of images in Figure 4.3.7.

(Fig. 4.3.9(a) and Fig. 4.3.10(a)) contain similar features (the shape). Unfortunately, the redistribution is not enough to balance the histogram into the five regions, thus producing low dynamic range images as shown in Fig. 4.3.6(d) and Fig. 4.3.7(d).

Table 4.3.1 shows the timing results of the proposed color characterization, enhancement, and balancing process when compared with the method in [61] and the method in [78]. The results were obtained using Intel Pentium III 1.13 GHz processor with 384MB of memory. The results were obtained by averaging the time it takes to process 10 images for each algorithm. It can be seen that when processing images at the resolution of 320×240 , the proposed algorithm is about 4 times faster than the method in [78] and 9 times faster than the method in [61]. Moreover, when processing an image at the resolution of 640×480 , the proposed algorithm is 14 times faster than the method in [78] and 33 times faster than the method in [61].

	Proposed method	Method in [78]	Method in [61]
320 × 240	1.5940 sec	6.2970 sec	15.2568 sec
640 × 480	5.2531 sec	75.3270 sec	175.2348 sec

Table 4.3.1: Results of proposed method compared to methods in [78] and [61].

There is no general theory for determining the image enhancement quality quantitatively. Several literatures exist to provide image quality estimation [79]. When image enhancement techniques are used as preprocessing tools for other image processing techniques, then quantitative measures can determine which enhancement technique is better than the others. We have implemented [57] to perform human skin color segmentation. A nonlinear line attractor network was trained using a skin sample collected in our laboratory. Our aim is to test the skin segmentation results with and without the image enhancement algorithms. We have selected five images of each category of lighting conditions: normal lighting, uneven lighting, and extremely uneven light to test the abilities of the enhancement algorithm to improve the skin segmentation results. Fig. 4.3.11 shows the results of this experiment. It can be seen that in the normal lighting environments, enhancement improve the segmentation results slightly. Skin segmentation results are almost the same with or without the preprocessing step in the normal lighting condition. On the other hand, difficult lighting conditions introduced complexity into the image processing. Our proposed enhancement method has the highest ability to correct some of these difficulties to provide a more robust color based computer vision algorithm. It is important to note that the result provided in Fig. 4.3.11 is the percentage of true recognition given that there is 0% false recognition. It is possible to obtain a higher recognition rate on the expense of detecting non-skin as skin.



Figure 4.3.11: Skin segmentation result using [57] with and without image enhancement.

4.4. Towards Representation of Perceptual Color Manifold using Associative Memory for Color Constancy

In this section, we propose the concept of manifold of color perception through empirical observation that the center-surround properties of images in a perceptually similar environment define a manifold in the high dimensional space. Such a manifold representation can be learned using the novel recurrent neural network based learning algorithm. The dynamics of the proposed learning algorithm represents memory as a nonlinear line of attraction. The region of convergence around the nonlinear line is defined by the statistical characteristics of the training data. This learned manifold can then be used as a basis for color correction of the images having different color perception to the learned color perception. Experimental results show that the proposed recurrent neural network learning algorithm is capable of color balance lighting variations in images captured in different environments successfully.

4.4.1. Problem Statement

The majority of objects in our environments reflect incident light to our eyes. This means that the light that reaches our eyes from an object depends on both the surface and subsurface material characteristics of the object and on the spectral distribution of the incident light. It is well known that the spectral makeup of the light incident on objects in our environment is not constant. For example, the light from the sun varies in its spectral distribution depending on the time of day, weather conditions, reflection and filtering by forest, canopies, and so on. This represents the visual system with a fundamental problem: a constant object does not reflect a constant spectral distribution of light to the eyes [80]. Somehow this variation in incident light must take into account in rendering an estimate

80

of an object color. Humans seem to have no trouble for these effects when occurring in small or moderate amount. However, these effects can have significant impact on the digital images obtained with a digital camera or camcorder and hence introduce complexity into image processing algorithms. Addressing the variability of images due to these variations in incident light has been an important problem in machine vision. That is, color correction is an important preprocessing step for robust color based computer vision algorithms because the measured color of an object will change under different light source [81].

In this section, we propose the concept of manifold of color perception in which a given image will automatically converge to the trained color perception, allowing the color to be rectified. An image with Q pixels can be considered as a point in the Q-dimensional image space, and perceptual similarity in the images can be represented as features in the Z-dimensional feature space where $Z \ll Q$. Since perceptually similar lighting environments can be described by observing statistical parameters from images, it is reasonable to assume that the similar lighting variations can be represented using a color manifold [1, 82]. By describing perceptual similarity as a manifold, many instances of similar lighting variations can be regarded as a nonlinear line of attraction in the state space (or memory).

It is a complex task to learn the structure of the manifold in the high dimensional space [1, 83]. Therefore, we have applied statistical analysis on the images of same color perception to extract lower dimensional features for describing the manifold. We have investigated the center-surround information of the images in their prospective red, green, and blue channels and the way they varies in the same color perceptual environment. To

do this, we have developed a recurrent neural network named nonlinear line attractor network for modeling how the center-surround values of red, green, and blue channels of images varies jointly under typically similar perceptual environment. We have found that the center surround value of red, green, and blue channels of image is suitable for the purpose of describing the manifold of color perception [84]. Center-surround function is widely used in natural vision science to model the perceptual processes. After applying the nonlinear line attractor network, the manifold of color perception can be approximately considered as a curved line in the state space.

4.4.2. Related work

Color constancy is usually considered as the ability to perceive the same color in varying lighting conditions. Color constancy algorithms can be mainly categorized into two groups. The first group imposes some constraints on the scene (i.e. illumination) in order to remove some ambiguities from the ill-posed color consistency problem. The second group used statistical modeling to make an estimation of the illumination [81]. Many researchers formalize the problem of color constancy for a visual configuration typical of the first Land's experiments [74, 75, 85]. Land proposed in [86] that color of an object is determined by its three lightness values in the three receptor channels: short wavelength, middle wavelength, and long wavelength. He showed that the lightness of an object in any one channel does not vary when either the spatial or spectral properties of the illumination on the entire scene are changed. He attributed this constancy to the visual system's ability to estimate the relative amount of light an object reflects without being missed by its absolute brightness. The computed lightness of a surface by Land's Retinex

algorithm approximates the logarithm of its reflectance normalized by the geometric mean of the surrounding surface reflectance. The resultant triplet of lightness values in the three spectral channels then defines the color. Unfortunately, in order to recover constant colors, the algorithm must implicitly assume that the average reflectance is the same in each channel and for every scene. In another word, the Retinex algorithm relies on the gray world assumption. The gray world algorithm for color constancy assumes the average color of an image in some predefined value of gray [84, 86]. The gray world assumption allows lightness algorithms to compensate for temporal changes in the spectral energy distribution of the illuminant on a given scene but cannot distinguish such changes from skews in surface reflectance distributions between scenes. Funt and his colleague model color constancy by training a neural network to estimate the chromaticity of the illumination which allows a transformation of an image to another illuminant [87]. Many physiological experiments and analysis have shown that human visual system is grounded on the stimulation of cones by three fundamental ranges of frequencies. All experts agree in the assumption that cones have three different pigments that allow them to react differently to these three light frequency ranges [88]. Recent approaches on color perception assume that human visual system is able to build colors from relative contrast between regions in an image, rather than from absolute color stimulus [72].

In this section, we frame the problem of color constancy using a recurrent neural network model named, nonlinear line attractor network, that learns by observing how the center-surround values of these light frequencies: red, green, and blue channels of images change under common lighting changes. Such a model can be used to correct the images of different color perception to the learned color perception. The manifold of color perception can be trained in a completely unsupervised fashion that does not use prior information of lighting condition, surface reflection, and built-in knowledge of the physics of the image acquisition during data collection and modeling.

4.4.3. Color Manifold and its Convergence Dynamics

Fig. 4.4.1 shows the convergence property of a family of colors to illustrate the learning and recall capability of the nonlinear line attractor network. Instances of green color and their corresponding *RGB* combinations in the *RGB* space are illustrated in Fig. 4.4.1a. It can be seen that the green color resides in the *RGB* color space as a nonlinear line. The proposed recurrent neural network can be applied to encapsulate the essence of green as a nonlinear line of attraction. That is, no matter how the pixel changes its value, the relationship between them is always described in the neural network. After training the neural network, distorted versions of the training pixels are obtained by adding Gaussian noise of zero mean (μ =0) and variance σ^2 =32 as shown in Fig. 4.4.1b. After the network converged to the nonlinear line of attraction, the reconstructed pixels were obtained. This is illustrated in Fig. 4.4.1c. It can be seen that the network is capable of correctly recalling the color successfully. That is, the network did not diverge away from the nonlinear line of attraction. This experiment confirms the capability of the proposed neural network based technique to perform accurate color modeling and balancing.

The convergence dynamics of the recurrent neural network is also examined by performing pattern association tasks. The set-up of this experiment is as follows:





(b) Corrupted color pattern with Gaussian noise ($\mu = 0$ and $\sigma^2 = 32$)



(c) Recalled color

Figure 4.4.1: Families of color, corrupted color, and recalled color.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Gaussian noise of zero mean μ and standard deviation σ^2 between 0 and 32 is added to a training set of 5000 pixels. The number of iterations for each pixel to converge is then recorded. Fig. 4.4.2 shows the result of this experiment. It shows that the recurrent neural network converges in two iterations in a majority of the cases. This experiment shows that the proposed neural network has very fast convergence dynamics. This is a significant improvement compared to other recurrent networks [59, 46, 10], which take an average of more than 10 iterations to converge to a fixed point [59, 46, 10] and is unable to associate multi-valued patterns [46].



Figure 4.4.2: Number of iterations for the 5000 pixel with zero mean μ and variance σ^2 between 0 and 32 to converge to the nonlinear line of attraction.

4.4.4. Modeling the Manifold of Color Perception and Simulation Results and Discussion

A material with reflectance $\theta_i(\lambda)$ seen under an illuminant with spectrum $p_j(\lambda)$ can be rendered by calculating cone absorption for long L_{ij} , middle M_{ij} , and short S_{ij} wavelength sensitive cone [87]:

$$L_{ij} = \int p_j(\lambda) \theta_i(\lambda) L(\lambda) d\lambda \qquad (4.4.1)$$

$$M_{ij} = \int p_j(\lambda) \theta_i(\lambda) M(\lambda) d\lambda \qquad (4.4.2)$$

$$S_{ij} = \int p_j(\lambda) \theta_i(\lambda) S(\lambda) d\lambda \qquad (4.4.3)$$

Gray world model approximates cone absorptions corresponding to the illumination spectrum. They are obtained by taking the means of all the cone absorptions [89]

$$L_{c}(\hat{p}_{j}) = \sum_{i} \int p_{j}(\lambda) \theta_{i}(\lambda) L(\lambda) d\lambda \qquad (4.4.4)$$

$$M_{c}(\hat{p}_{j}) = \sum_{i} \int p_{j}(\lambda) \theta_{i}(\lambda) M(\lambda) d\lambda \qquad (4.4.5)$$

$$S_{c}(\hat{p}_{j}) = \sum_{i} \int p_{j}(\lambda) \theta_{i}(\lambda) S(\lambda) d\lambda \qquad (4.4.6)$$

where the subscripts c denotes the chromatic and the caps symbol $\hat{}$ denote the approximated quantities.

Experimental results indicate that the light striking rods and cones in the retina is not summed uniformly. Instead, the nerves that combine the signals from the rods or cones sum with center-surround opponency. Indeed, electrophysiological studies have shown that many receptive fields have a center-surround organization. Land's centersurround retinex demonstrated its ability for the color constancy properties. The general form of the center-surround retinex is similar to the difference of Gaussian function that is widely used in natural vision science to model both the receptive field of individual neurons and perceptual process.

Suppose we are given *L RGB* color images I_1 , I_2 , I_3 , ..., I_L , of similar color perception. Our approach is to model these observed pixels' statistical characteristics using the concept of manifold. We have applied the center-surround concept for each of the *RGB* channel of the image. The center-surround function we have chosen is of the Gaussian form because of its widespread use in natural and machine vision modeling and because of its distinctive regional processing of the pixels that provides excellent local information of a region. The surround function therefore is expressed as

$$\phi(q,r) = \tau e^{-(q^2 + r^2)/\sigma^2}$$
(4.4.7)

where σ is the standard deviation of the Gaussian distribution and τ is selected so that $\iint \phi(q,r) dq dr = 1$. σ is used to provide an estimation of the overall illumination in an image. For instance, in Fig. 4.4.3, $\sigma = 50$ gives illumination information of each object in the image. For example, it is still possible to tell the face of the person from the shirt which he is wearing. Conversely, $\sigma = 200$ gives an excellent overview of the global illumination across the entire image.



Figure 4.4.3: The effect of varying σ in S(s,q).

It is interesting to notice in the first row of the images presented in Fig. 4.4.3 that the illumination of the normal lighting image is not far away from the gray values when σ = 200. We want to utilize the gray world model to investigate this phenomena by studying the center-surround information of the *RGB* channel of the images and the way they varies with respect to each other in the same color perception environment. We found that the manifold of color perception does exist in such a *RGB* space. Fig. 4.4.4 shows the scatter plot of the red, green, and blue components of the center-surround information of images in the same lighting condition. These 4800,000 points consist of 10 images of size 800 × 600 collected in perceptually similar lighting environment. It can be seen from the scatter graph that similar lighting scenes indeed form a nonlinear line of attraction in the *RGB* space. This shows that it is possible to track, predict, and approximate the lighting in images mathematically using the dynamic of the nonlinear line attractor network, defining the manifold of color perception in the state space representation.



Figure 4.4.4: Manifold of color perception.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Furthermore, it is observed that when

$$w_{(m,ij)} = \begin{cases} 1 & \text{if } m = 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{for} \quad 1 \le i, j \le N \quad \text{and} \quad 0 \le m \le k$$

the proposed model is identical to the gray world model. The difference lies in the capability of the proposed model to approximating the lighting of images using perceptual manifold and center-surround concepts.

In the following experiment, our aim is to capture the structure of the manifold of color perception by observing images collected from the same color perception in an unsupervised learning fashion. To accomplish this, we have collected 250 images of size 64×64 using Logitech QuickCam Pro 4000 camera from 25 persons, each with 10 images, under a variety of lighting conditions. Our goal is to describe the color perception utilizing the concept of manifold and to converge all the images of different lighting conditions into the learned lighting condition. Fig. 4.4.5 shows the memory of the attractor network after training the network with the algorithm described in chapter 3 with eqn. (2.22) and (2.23) setting k=1. The order of the polynomial k presents a trade off between computational complexity and the accuracy in modeling the color manifold. The training set consists of 10 images of the same lighting perception from the same person. It can be visualized from Fig. 4.4.5 that the line attractor network forms a manifold in the \mathbb{R}^3 space.

The mapping of different color perception to the learned color perception can be performed using the following color balancing algorithm:

Step 1: Input image *I*.


Figure 4.4.5: Memory of nonlinear line attractor network.

Step 2: Calculate the center-surround information of *I* by convolving the image and the center-surround function $\Gamma = I * S$, where * is the convolution operator.

Step 3: Converge different color perception Γ to the trained color perception

 $\Theta(\Gamma)$ is obtained using eqn. (2.24) to eqn. (2.26).

Step 4: Correct image I using the following equation

$$I^{+} = I + \varepsilon \left| \Theta(\Gamma) - \Gamma \right| \quad \text{for} \quad 0 < \varepsilon \le 1$$

$$(4.4.8)$$

where I^{\dagger} is the color balanced image.

Fig. 4.4.6 gives illustration of these steps.

The color correction algorithm is applied to the database we have created and discussed in which the images are captured at different spectral properties. For instance, there are images in the databases that are captured in candlelight, incandescent light, fluorescent light, and sunlight. Since each person set contains 10 images, 10 images of the same person of the same color perception is used for training the nonlinear line attractor network. It is observed that all the images converge to the trained color



Figure 4.4.6: Color balancing algorithm.

perception using the proposed color balancing algorithm described in step 1 to step 4. These experiments were performed for different color perception and the test images always converge to the trained color perception in our database. Fig. 4.4.7 illustrates typical results of these experiments. The first row of the images is the reference color perception. It was used as a training set for modeling the manifold of color perception. Therefore, there were not any changes after processing the image with the proposed algorithm. Images of row 2, row 3, and row 4 convergence from candlelight, incandescent light, and daylight to the reference color perception.

Quantitative measures can be devised to identify which color balancing approach is better than the others by using the color balancing technique as a preprocessing tool for further image analysis. As a result, the following experiment tests the result of the



(a) Original images(b) Color balanced imagesFigure 4.4.7: Results of color correction.

proposed manifold of color perception in the application of skin color segmentation. The majority of images acquired today are colored. Consequently, skin color detection can be considered as an important preprocessing step for computer vision and pattern recognition applications since skin color can be an important source of information for discrimination. Four skin probability maps (SPM) [90] are trained for evaluating the skin segmentation results: 1. without preprocessing of the image; 2. with preprocessing of the image using the gray world model; 3. with preprocessing of the image using multi-scale Retinex with color restoration (MSRCR) [61, 91]; and 4. with preprocessing of the image using the proposed method. The theory of Retinex was proposed by Edward Land in modeling human based image processing algorithm that provides color constancy under

varying illumination conditions. Multi-scale Retinex is a multi-scale embodiment of the Retinex algorithm. Many works suggested that multi-scale Retinex as a method to bridge the gap between what a camera sees and what a human sees. Fig. 4.4.8 shows a typical example of MSRCR output. It can be seen that MSRCR is able to remove the color cast of the original image successfully. The training set for SPMs consist of five random selected skin samples of size 10×10 pixels around the face region from each of the 25 persons in the database. The testing images consist of all the 250 images in the database. It can be seen from Fig. 4.4.9 that the proposed method outperforms the other three methods. That is, the proposed method has the highest ability to remove some of the lighting complexities to provide improvement in skin segmentation.



(a) Original image

(b) MSRCR image

Figure 4.4.8: Results of color correction using MSRCR.



Figure 4.4.9: Skin segmentation results.

CHAPTER 5

APPLICATION II: PATTERN CLASSIFICATION

In this chapter, we perform pattern classification using the learning algorithm we have developed in chapter 2: Automatic skin color detection is studied extensively for human related recognition systems. We propose a novel approach for skin color modeling using the nonlinear line attractor network. The developed system is used for real time skin color detection.

5.1. Problem Statement

The first important step in recognizing faces in video sequences is to find where they are. The majority of images acquired today are color images. Consequently, skin color detection could be considered as an important preprocessing step for an automatic human face recognition system since skin color is an important source of information for discriminating faces from the background. It is observed that the image pixels corresponding to human skin colors form an elongated pipe in the three-dimensional *RGB* color space. Hence, it is possible to describe the skin color mathematically using a line attractor network.

5.2. Simulation Results and Discussion

The training skin samples set is collected from 10 individuals with each having 4 samples of size 10×10 pixels using a Sony EVI-D30 surveillance camera. These skin samples include skins of people belonging to different skin colors from different ethnicity. The

97

skin pixel distribution is plotted on a 3-D scatter plot and is illustrated in Fig. 5.1. It is observed from Fig. 5.1 that the data tends to bulge in the middle of the *RGB* space. Therefore, we have trained a line attractor network to learn the similarity of the skin characteristics but evenly divided the line of attraction into eight regions ($\Omega = 8$) for thresholding. As it can be seen in Fig. 5.2, the linear attractor network is able to classify the data into skin and non-skin regions more accurately using eight thresholding functions employing only the skin samples as training data.



Figure 5.1: Skin distribution in RGB space.

Fig. 5.3 shows an example of the skin color segmentation using the proposed method. It can be seen that the skin pixels converge to the line of attraction whereas the non-skin colors diverge away from the line.



Figure 5.2: RGB weight graph representing 3-neuron linear attractor memory.



Figure 5.3: Skin segmentation result.

For testing the performance of the proposed algorithm, we selected a set of ten images obtained using our Sony EVI-D30 surveillance camera. The selected images were chosen to span a different range of environmental conditions. People from various ethnicities and various skin tones were also represented in this test set. All these test images were hand-labeled to provide ground truth data for algorithm performance verification. Performance of the network for skin region segmentation is quantified by computing the receiver operating characteristics (ROC) curve, which shows the relationship between correct detections and false detections. It is shown in Fig. 5.4 that the performance of the skin classifier trained using the proposed learning algorithm outperformed skin probability map (SPM), which is a classical model for skin detection [90]. The SPM we used is a non-parametric skin modeling method that estimates skin color distribution from the training data. It constructed a skin probability map by assigning probability to each point in a $64 \times 64 \times 64$ discretized color space. After training, both the classifiers were tested with the same set of test images. Experiments have also been performed in the L*a*b space, which is a perceptually uniform color space, using the line attractor network. After converting the skin samples from the *RGB*



Figure 5.4: Receiver operating characteristics curve of RGB space.



Figure 5.5: Skin distribution in L*a*b space.

space to the L*a*b space, the skin pixel distribution is plotted on a 3-D scatter plot and is illustrated in Fig. 5.5. Notice that the scatter plot of the skin samples in the L*a*b space form a cluster in the 3-D space. However, the linear attractor network is able to encapsulate the skin samples in a line of attraction. Fig. 5.6 illustrates the divide and conquer modular approach using eight threshold regions in the line attractor network for forming a cluster in the L*a*b space. It can be seen in Fig. 5.7 that the skin classifier trained using the proposed learning algorithm outperforms SPM in the L*a*b color space.



Figure 5.6: L*a*b weight graph representing 3-neuron linear attractor memory.



Figure 5.7: Receiver operating characteristics curve of *RGB* space L*a*b.

102

CHAPTER 6

APPLICATION III: DIMENSIONALITY REDUCTION

We present a dimensionality reduction technique based on the observation that images of similar visual perception reside in a complex manifold in a low-dimensional image space. Manifolds are fundamental to perception. The perceived features are often highly structured and hidden in a complex set of relationships or high-dimensional abstractions. To model the pattern manifold, we utilize the learning algorithm developed in chapter 2.

6.1. Problem Statement

One of the goals of biometrics research is to develop new techniques and/or algorithms for the automatic recognition of humans. In this chapter, we propose the concept of a manifold of facial perception based on the observation that a perceived face in a set of similar face images—subjected to variations in pose, illumination and expression defines a manifold in the high dimensional space. Such a manifold representation can be learned from images in a database of similar facial feature characteristics. This learned manifold can then be used as a basis for facial recognition. Development of a mathematical model for a nonlinear line attractor that represents a pattern manifold in the feature space is presented in chapter 2 of this dissertation. The non-convex pattern manifolds in the feature space may be extremely complex and difficult to model. Therefore an adaptive divide and conquer modular approach which divides complex manifolds into smaller sub-manifolds in the feature space is also proposed for accurate modeling of the nonlinear line of attraction. A nonlinear dimensionality reduction technique using the learned matrices of the nonlinear line attractor network is then used to embed a set of related observations into a low-dimensional space that preserves the intrinsic dimensionality and metric structure of the data for fast and accurate face recognition. Results based on the proposed work from the first experiment of the Face Recognition Grand Challenge (FRGC) version 2 database have shown promising performance in improving the accuracy in the face recognition.

6.2. A New Nonlinear Dimensionality Reduction Technique for Pose and Lighting Invariant Face Recognition

Automatic recognition of faces is considered to be an important problem in computer vision and pattern analysis [92]. Researchers in computer vision and pattern recognition have worked on automatic techniques for recognizing human faces for the past 30 years. While there have been some successful systems, most face recognition systems fail when the facial pose is not fixed at a full frontal view, when the lighting is not controlled, when the facial expressions is varied, or any combination of the above.

Subspace based techniques have dominated the approaches in face recognition since the 1990s due to their simplicity and demonstration of excellent performance compared to others methods [93]. The majority of the subspace based approaches are based on the method described by Turk and Pentland [94], where the authors used the principal components to represent face images. The principle component analysis approach transforms a given set of face images into a smaller set of basis images using matrix decomposition techniques. Given a new face image, that image is projected into a reduced dimensional space spanned by these basis images, where the recognition is carried out by comparing the distances between this new face features to all other known faces features.

Indeed, psychophysical findings indicate that perceptual tasks such as similarity judgment tend to be performed using a low dimensional representation of the sensory data. Low dimensionality is especially important for learning, as the number of examples required for attaining a given level of performance grows exponentially with the dimensionality of the underlying representation space. Because of this curse of dimensionality, categorization of the initial high-dimensionality sensory data must be reduced to a nontrivial computational process, which ideally should capture the intrinsic low dimensional nature of families of visual pattern [95].

Principal components analysis is commonly used to perform dimensionality reduction by projecting the data into a subspace spanned by the eigenvectors of the covariance matrix. In computer vision applications, it has been used for the representation and recognition of faces [94], recognition of 3D objects under varying pose [96], and for representations of 3D range data [97]. Unfortunately, dimensionality reduction techniques are sensitive to image plane transformations. The principal components analysis approach will perform poorly under the following conditions:

- *High curvature of the manifold*: PCA finds a low-dimensional embedding of the data points that best preserves their variance as measured in the high-dimensional input space. However, many data sets contain essential nonlinear structures that are invisible to PCA.
- *Presence of small manifolds*: In many real-world problems, there is not only one global manifold but also a large number of manifolds that share

additional information about the objects. An example is the manifold of transformations of 3D objects in 2D images. PCA is incapable of functioning in these complex situations.

Manifolds are fundamental to learning. Information related to similar patterns residing in a complex manifold should be visualized as a *curved line* in the feature space. This nonlinear line encapsulates attractive fixed points representing patterns with similar characteristics. Some nonlinear techniques have been proposed to discover the nonlinear structure of the manifold, such as Isomap [98], LLE [99] and Laplacian Eigenmaps [100]. These nonlinear methods do yield impressive results on some benchmark artificial data sets. However, these methods are developed based on reconstruction and are not optimal for a classification viewpoint [93].

In this chapter, a mathematical model for a nonlinear line attractor that represents a pattern manifold in the feature space is introduced. The non-convex pattern manifolds in the feature space may be extremely complex and difficult to model. Consequently, an adaptive divide and conquer modular approach which divides complex manifolds into smaller sub-manifolds in the feature space is proposed. A nonlinear dimensionality reduction technique using the learned matrices of the nonlinear line attractor network is then used to embed a set of related observations into a low-dimensional space that preserves the intrinsic dimensionality for fast and accurate face recognition.

The chapter is organized as follows. We first describe lighting compensation in section 6.3. We then describe and discuss invariant face recognition in section 6.4. These include the nonlinear line attractor network model, the adaptive divide and conquer

modular approach, and the nonlinear dimensionality reduction technique. The FRGC experiment performed will be described in section 6.5.

6.3. Preprocessing

Electronic cameras based upon CCD detector arrays typically suffer from dynamic range compression problems. Dynamic range compression occurs when a scene contains both very bright and very dark elements. In such a case, one can either capture the bright region at the cost of losing the dark region, or the dark region at the cost of saturating the bright region. Addressing the variability of images due to these dynamic range problems has been an important problem in machine vision. These effects can have significant impact on the digital image obtained with digital cameras or camcorders and hence introduce complexity into computer vision and pattern recognition problems.

We have used a sigmoid transfer function for increasing the dynamic range of an image. A hyperbolic tangent function is used for the reason of overcoming the natural loss in perceived lightness contrast that results when performing dynamic range compression. We have developed an enhancement strategy that will perform the range compression while maintaining the image details. The proposed solution is to develop the hyperbolic tangent functions that are tunable based on the statistical characteristics of the image. That is, the functions will enhance the dark part of the image while preserving the light part of the image based on [101]:

$$\mathcal{G} = 255 \times \left(\frac{2}{1 + e^{-2\tau/\rho}} - 1\right)$$
 (6.1)

where τ is the pixel value, ρ is the statistics of the image, and \mathscr{G} is the enhanced pixel value. The parameter ρ controls the curvature of the hyperbolic tangent function. This

107

means that when the processing image is dark, ρ should be small and therefore the curvature of the hyperbolic tangent function will be steep and this will help the darker pixels to have brighter values. ρ can be expressed as [101]:

$$\rho = \left(255 - k\right) \left[\frac{\Upsilon}{255}\right]^{\gamma} + k \tag{6.2}$$

where Υ is the local mean of an image and k is the bias pixel intensity value. The local mean of each pixel is calculated based on the center surrounded property of a perceptual field and perceptual processes of human vision. The form of the surround function we used is Gaussian because it provides good dynamic range compression over a wide range of environments [71]. Consequently, the local mean of the image is calculated by:

$$\Upsilon_{x,y} = c e^{-(x^2 + y^2)/\sigma^2}$$
(6.3)

where σ is the standard deviation of the Gaussian distribution and *c* is selected so that $\int \int \Upsilon_{x,y} dx dy = 1$. The choice of σ presents a trade off between the dynamic range compression and color rendition of the image. A smaller σ will yield larger dynamic range compression but causes the image to lose its color. Conversely, a larger σ will yield better color rendition but the shadow of the image will remain constant.

Fig. 6.1 illustrates the variability of the hyperbolic tangent function based on equations (6.1) to (6.3). It can be observed that when the local mean of an image is small, the hyperbolic tangent function reshapes its curve towards the brighter pixel value to facilitate the rescaling of the range of the dark pixel to the brighter region. Conversely, when the local mean of an image is large, the hyperbolic tangent function compresses the brighter pixels to the darker region.



Figure 6.1: Hyperbolic tangent function.

The enhancement method has been tested on a set of high dynamic range color images. Fig. 6.2 shows typical results of the original and enhanced image pair. That is, Fig. 6.2b shows the enhanced result of the original image shown in Fig. 6.2a based on our method. From Fig. 6.2a, we can see that the camera is unable to capture and represent the image using its limited dynamic range. On the other hand, the enhancement algorithm is able to provide image reproduction which is very similar to what the human would have seen were he present when the image was taken; all the shadow in the image is removed.



(a) Original image



(b) Enhanced image

Figure 6.2: Lighting compensation.

6.4. Invariant Face Recognition

The principle component analysis based face recognition method is not effective under the conditions of varying pose and illumination. In this section, a novel face recognition method, based on the nonlinear line attractor network [102], is described. Our method for dimensionality reduction is similar to the principle component analysis. In the first step, weight matrices will be calculated based on the nonlinear line attractor network using the adaptive divide and conquer approach. Then, singular valued decomposition is used to decompose the learned matrices. The recognition is performed by comparing the distances between the low dimension representations of a face to all those known low dimensional representation of faces.

6.4.1. Feature Space Division

The nonlinear line attractor network model discussed in Chapter 3 can associate a pattern that can be described using a set of polynomial functions of degree k with their corresponding threshold regions. However, the polynomial functions and the threshold regions encapsulating the region of convergence cannot describe a region which contains a hole. That is, eqns (2.22) to (2.30) are designed for learning a convex region. We propose a divide and conquer modular approach to define a non-convex region in a feature space.

A feature space of size $N_1 \times N_2$ is divided into sub-spaces of smaller size, $n_1 \times n_2$, and a nonlinear line attractor network is created in each sub-space. We can model the division of the original space into sub-spaces using a distance based algorithm that can be expressed as [103]:

$$D_{i_1 i_2 j_1 j_2} = \begin{cases} 1 & \eta = 0 \\ 0 & \text{otherwise} \end{cases} \quad 1 \le i_1, j_1 \le N_1 \text{ and } 1 \le i_2, j_2 \le N_2 \tag{6.4}$$

where η is defined by:

$$\eta = \max\left\{ \left| \left\lfloor \frac{i_1 - 1}{n_1} \right\rfloor - \left\lfloor \frac{j_1 - 1}{n_1} \right\rfloor \right|, \left| \left\lfloor \frac{i_2 - 1}{n_2} \right\rfloor - \left\lfloor \frac{j_2 - 1}{n_2} \right\rfloor \right| \right\}$$
(6.5)

111

η describes the distance measure between the $(i_1, i_2)^{\text{th}}$ location and the $(j_1, j_2)^{\text{th}}$ location in the feature space. If $D_{i_1i_2j_1j_2}$ is equal to 1, then the $(i_1, i_2)^{\text{th}}$ location and the $(j_1, j_2)^{\text{th}}$ location belong to the same $(n_1 \times n_2)^{\text{th}}$ sub-space. Otherwise if $D_{i_1i_2j_1j_2}$ is equal to 0, they do not belong to the $(n_1 \times n_2)^{\text{th}}$ sub-space.



Figure 6.3: Distance function.

An example of eqn (6.5) is showed in Fig. 6.3. The distance between (i_1, i_2) and (j_1, j_2) is 2 since they are two blocks away from each other. As a result, $D_{i_i i_2 j_1 j_2}$ is 0 and they will not be used in the same nonlinear line attractor network. Conversely, the distance between (i_{21}, i_{22}) and (j_{21}, j_{22}) is 0 since they are in the same block. As a result, $D_{i_i i_2 j_1 j_2}$ is 1 and they will be used in the same nonlinear line attractor network.

6.4.2. Adaptive Feature Space Division

We can devise a strategy for deciding the minimum number of modules that is necessary to create the complex region by the divide and conquer modular approach presented in the previous section. It can be observed that the minimum number of required modules directly depends on the maximum number of sub-regions inside a region. This can be computed by observing the error histogram at different regions after creating the first manifold (green line) with a single network as illustrated in Fig. 6.4. The difference between the predicted output and the actual output of x_i^s is calculated as:

$$\varepsilon_{(ij,l)}^{s}\left(x_{j}^{s}, x_{i}^{s}\right) = \begin{bmatrix} \left\{\sum_{\nu=0}^{k} w_{(\nu,ij)}\left(x_{j}^{s}\right)^{\nu}\right\} - x_{i}^{s}, \text{ in the region of} \\ \left(l-1\right)\frac{L}{\Omega} \le x_{j}^{s} < l\frac{L}{\Omega} \end{bmatrix} \quad \text{for} \quad 1 \le l \le \Omega \quad (6.6)$$

The histograms of $\varepsilon_{(ij,l)}^s(x_j^s, x_i^s)$, $\forall s$ at l = 10, 20 and 95 are shown in Fig. 6.4.



Figure 6.4: Adaptive feature space division.

The network first creates a nonlinear line attractor based on eqns (2.22) to (2.23) (green line in Fig. 6.4a). The network then examines the histogram of error at all regions of l = 1, 2, ..., Ω , where Ω is the number of threshold regions. In Fig. 6.4b, the error histogram for l= 10 is uniform since that region contains a homogenous boundary as evident in Fig. 6.4a. In Fig. 6.4c, the error histogram for the region at l = 20 is disjointed, indicating that the region contains 2 boundaries. In Fig. 6.4d, the error histogram at l = 95 includes 10 disjointed regions since it contains 10 boundaries as can be seen in Fig. 6.4a. This is the maximum number of disjointed regions in this problem. Hence it chooses the number of modules as 10×10 for creating the spiral manifold shown in Fig. 6.4a using the divide and conquer modular approach.

6.4.3. Nonlinear Dimensionality Reduction

The "curse of dimensionality" refers to the fact that in the absence of simplifying assumptions, the sample size needed to estimate a function of several variables to a given degree of accuracy grows exponentially with the number of variables. The problem of dimensionality reduction is introduced as a way to overcome the curse of dimensionality when dealing with vector data in high-dimensional spaces. Many high-dimensional data sets in real-world applications can be modeled as sets of points or vectors lying close to a low-dimensional nonlinear manifold. Discovering the structure of the manifold from such a sample of data points represents a very challenging unsupervised learning problem. We propose a new algorithm for nonlinear dimensionality reduction using singular value decomposition (SVD) on the matrices learned by the adaptive feature space division technique. SVD provides the necessary mathematical tools for understanding an

important class of optimal dimensionality reduction mappings. Singular values represent important attributes of a matrix. Using the adaptive feature space division technique, the system produces r nonlinear line attractor networks, where the d^{th} network is represented by a memory matrix W^d . The output of the d^{th} network can be expressed as:

$$\begin{pmatrix} y_1^d \\ \vdots \\ y_n^d \end{pmatrix} = \begin{pmatrix} w_{k,11}^d & \dots & w_{k,1n}^d \\ \vdots & \ddots & \vdots \\ w_{k,n1}^d & \dots & w_{k,nn}^d \end{pmatrix} \begin{pmatrix} (x_1)^k \\ \vdots \\ (x_n)^k \end{pmatrix} + \dots + \begin{pmatrix} w_{0,11}^d & \dots & w_{0,1n}^d \\ \vdots & \ddots & \vdots \\ w_{0,n1}^d & \dots & w_{0,nn}^d \end{pmatrix} \begin{pmatrix} (x_1)^0 \\ \vdots \\ (x_n)^0 \end{pmatrix}$$
(6.7)

where y_i^d is the output at the *i*th neuron in the *d*th network (Note that *d* in this equation represents the network number, whereas *s* in eqn (2.21) represents the pattern number in a training set). The SVD of the *m*th order weight matrix w_m^d (where m = 0, 1, ..., k) of the *d*th network can be obtained by:

$$w_m^d = U_m^d \Sigma_m^d \left(V_m^d\right)^T \quad \text{for } 0 \le m \le k \text{ and } 1 \le d \le r$$
(6.8)

Where

$$U_m^d \in \mathbb{R}^{n \times n}, \quad V_m^d \in \mathbb{R}^{n \times n},$$

and

$$\Sigma_m^d = \operatorname{diag}\left(\sigma_1^{(d,m)}, \sigma_2^{(d,m)}, \cdots, \sigma_n^{(d,m)}\right) \in \mathbb{R}^{n \times n}$$

Such that

$$\sigma_1^{(d,m)} \ge \sigma_2^{(d,m)} \ge \cdots \ge \sigma_n^{(d,m)} \ge 0$$

Projection of the *n*-dimensional data to a *z*-dimensional subspace using a $z \times N$ submatrix obtained from the *V* matrix of the SVD yields a *z*-dimensional output Y_m , where *z* $\ll N$. The value *z* is chosen based on a trade-off between the performance of the classifier and computational speed requirements. This could be done by choosing a value of z based on the proportion of information covered by z number of eigenvalues that can be estimated by $\rho = \sum_{i=1}^{z} \sigma_i^{(d,m)} / \sum_{i=1}^{n} \sigma_i^{(d,m)}$ ($\rho \approx 0.9$ provides reasonable accuracy in the performance). Hence the z-dimensional vector obtained from the d^{th} network can be expressed as:

$$\begin{pmatrix} y_{1}^{d} \\ \vdots \\ y_{z}^{d} \end{pmatrix} = \begin{pmatrix} V_{k,11}^{d} & \dots & V_{k,1n}^{d} \\ \vdots & \ddots & \vdots \\ V_{k,21}^{d} & \dots & V_{k,2n}^{d} \end{pmatrix} \begin{pmatrix} (x_{1})^{k} \\ \vdots \\ (x_{n})^{k} \end{pmatrix} + \dots + \begin{pmatrix} V_{0,11}^{d} & \dots & V_{0,1n}^{d} \\ \vdots & \ddots & \vdots \\ V_{0,z1}^{d} & \dots & V_{0,2n}^{d} \end{pmatrix} \begin{pmatrix} (x_{1})^{0} \\ \vdots \\ (x_{n})^{0} \end{pmatrix}$$
(6.9)

Here we are able to represent *n*-dimensional data by r number of z-dimensional vectors, where r is the number of sub-manifolds needed to obtain the complex manifold by adaptive feature space division technique and z is the dimensionality of the projected subspace.

6.4.3.1. Skin Region Extraction

The color information (*Red, Green and Blue*) representing the skin pixels are overlapping in a 3D color space. By using the proposed nonlinear dimensionality reduction technique, we are able to transform the 3D information to a 2D space that yields clear separation of the skin and non-skin regions as it is illustrated in Fig. 6.5.



Figure 6.5: Illustration of the efficiency of dimensionality reduction using the proposed method for extraction of human skin regions in various illumination conditions (indoor and outdoor).

6.4.3.2. Expression Invariant Face Recognition

Fig. 6.6 illustrates the efficiency of dimensionality reduction using the proposed method for expression invariant face recognition on the CMU expression-variant database [104]. Each cluster in this illustration consists 75 points representing 75 images of each person in the 13 person database. The 4096 dimensional (64×64) face images are transformed onto a 2 dimensional space (z = 2) by the proposed dimensionality reduction technique using only one nonlinear line attractor network created by the adaptive feature space division method. It can be seen that the face feature manifolds created by the face images of each person are clustered at different locations in the subspace enabling expression invariant face recognition.



Figure 6.6: Illustration of expression invariant face recognition.

6.4.3.3. Facial Expression Classification

It is observed that the regions representing different emotions of a person are clearly separable by mapping the face images of the individuals to the 2D space (z = 2) using the proposed method. Fig. 6.7 shows clear separation of the facial expressions of person 3 in Fig. 6.7.



1st principal axis

Figure 6.7: Illustration of the efficiency of dimensionality reduction using the proposed method for classification of human face expressions.

6.4.3.4. Pose Invariant Face Recognition

Fig. 6.8 illustrates the efficiency of dimensionality reduction using the proposed method for pose invariant face recognition on the UMIST pose-variant database [105]. The projection of the face images to the 2 dimensional space by the proposed algorithm yields reasonable performance in pose-invariant face recognition. However, the 20 clusters representing the group of manifolds created by images of 20 individuals are not wellseparated in the 2 dimensional space as can be observed in the Fig. 6.8. 400 dimensional representation of the faces using the proposed dimensionality reduction method provides 0.9983% accuracy with 0% false positives. In the case of PCA, the recognition rate was 0.3% and false positive is 0.625%.



Figure 6.8: Illustration of the efficiency of dimensionality reduction using the proposed method for pose invariant face recognition.

6.5. FRGC Experiments

We have tested our non-linear dimensionality reduction technique for face recognition on the FRGC version 2 database [106]. Specifically, we have evaluated our algorithm on the first FRGC version 2 experiment. In this experiment, the system is trained on 12,776 separate recordings representing 636 individuals. The performance of the algorithm was evaluated using all 12,776 target recordings from the first experiment.

The recordings on which the recognition system were trained and tested were grayscale intensity images of size 32 by 37 pixels. These images contain only the subject's face; non-face regions are hidden by an elliptical mask. The cropping and masking process was performed using the partially-automated preprocessing facilities provided by the Biometric Experimentation Environment. This preprocessing technique uses a priori eye centers coordinates to locate and mask the face region geometrically. The technique is based on the algorithm by Moon and Phillips which was originally used to normalize face image for the FERET 96/97 studies. This partially-automated method is used in order to provide a more meaningful comparison to the BEE baseline recognition algorithms. No additional adjustment or normalization was performed on the recordings.

We have performed a test on the first FRGC version 2 experiment. The training process was performed by assembling two coefficient matrices, W_1 and W_2 using eqns (2.22) to (2.25) in section 2.2. Singular value decomposition is then performed on the W_1 and W_2 matrices using eqn (6.8). The resultant matrices V_{w1} and V_{w2} are used to project vectorized training recordings into the feature space using eqn (6.9). Fig. 6.9 shows the reconstruction of the face images from the test set of the FRGC experiment to verify the correctness of the encoding-decoding process using our proposed method. It can be seen that when z = n. The reconstruction of the face is close to perfect.



Figure 6.9: Reconstruction of the face images by the proposed method.

For validation, the recordings in the target set were projected into the feature space using the same V_{m} and V_{m_2} calculated during the training process. The feature vectors were classified by calculating the Euclidean distance between every combination of target image feature vectors. The classification was considered successful if a test feature vector and its nearest training feature vector were derived of recordings of the same person—matches resulting from identical images were naturally excluded. The success rate for the unmasked first FRGC version 2 experiment described above was 98.8% given 0% false positive. After applying FRGC mask III, which masks the results in such a way that target images from a given semester are only compared to target images from a different semester, the recognition rate becomes 38.7% at a 0.1% false accept rate. Fig. 6.10 shows the ROC of the first FRGC ver2 experiment. The results show that the algorithm needs fine tuning to compensate for the variations in the FRGC database. It is theorized that an appropriate transformation will be able to unfold the original face space to enable its representation with the nonlinear line attractor.



Figure 6.10: ROC of the first FRGC ver2 experiment.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER 7

SELF-ORGANIZATION OF NONLINEAR LINE OF ATTRACTION

The Human brain memorizes information using the dynamical system made of interconnected neurons. Retrieval of information is accomplished in associative sense; starting from an arbitrary state that might be an encoded representation of a visual image, the brain activity converges to another state that is stable and which is what the brain remembers. Associative memory can be modeled using a recurrent network, in which the stored memories are represented by the dynamics of the network convergence. In most models of associative memory, memories are stored as attractive fixed points at discrete locations in the state space. Fixed point attractor may not be suitable for patterns which exhibit similar characteristics. To precisely characterize the similarity of images and other perceptual stimuli, it would be more appropriate to represent the pattern association using a nonlinear line attractor network that encapsulates the attractive fixed points scattered in the state space with an attractive nonlinear line, where each fixed point corresponds to similar patterns [7, 102].

On the basis of studies of the olfactory bulb of an animal, Freeman suggested that in the rest state, the dynamics of the neural cluster in an animal is chaotic [28]. Conversely, when a familiar scent is presented to the animal, the neural system rapidly simplifies its behavior and the dynamics becomes more orderly. He found that when a rabbit's cells of smell are inactive, the electrical activity follows a loose chaotic pattern. But when active, this pattern explodes to a far more definite pattern of activity. According to other researchers [3], the background chaotic activity of the brain enables the system to jump rapidly from one attractor to the other when presented with the appropriate input. That is, if the input does not send the system into one of the attractors, it is considered a novel input [29]. Some researchers have long speculated that chaotic processes have some fundamental roles in mental processes [28].

The human brain has the ability to learn and memorize many new facts in a fashion that does not necessarily cause the existing ones to be forgotten. In order to design a truly intelligent machine, which is comparable to the human brain, it would be highly desirable to impart this ability to the mathematical models. Most learning paradigms tend to forget old information, if we attempt to store new patterns in an incremental fashion. That is, most learning techniques work only for static input environments. If a network is trained on a set of input vectors, it can classify the input environment correctly only if the input environment is not dynamic. In the presence of dynamically changing data, the accuracy of a network decreases rapidly because the fixed weights prevent the network from adapting to the changing environment. Thus, such networks are not plastic. To overcome this problem, the networks can be retrained on a new set of input vectors. The network will adapt to any changes in the input environment but this causes a rapid decrease of accuracy with which it categorizes the old inputs because the old information is lost. Thus, this algorithm is not stable. The above problem is called the stability-plasticity dilemma

In designing a recurrent neural network, it is usually of prime importance to guarantee the convergence of dynamics of the network. We propose to modify this picture. If the brain remembers by converging to the state representing familiar patterns, it should also diverge from such states when presented with an unknown encoded representation of a visual image. That is, the identification of an instability mode can be an indication that a presented pattern is far away from any stored pattern and therefore cannot be associated with current memories. These properties can be used to circumvent the plasticity-stability dilemma by using the fluctuating mode as an indicator to create new states. We propose to capture this behavior using a novel neural architecture and learning algorithm, in which the dynamical system performs self-organization utilizing a stability mode and an instability mode. This self-organizing behavior of the nonlinear line attractor model can help to create complex dynamics in an unsupervised manner.

7.1. Nonlinear Line Attractor Network Model

Information related to similar data resides in a pattern manifold, which can be visualized as a curved line in the state space. This nonlinear line encapsulates attractive fixed points representing patterns with similar characteristics. A simple model for learning a pattern manifold was presented and demonstrated in chapter 2 and our previous work [102, 57] based on these observations.

In asymmetric networks, there is no known Lyapunov function guaranteeing convergence to an attractor. Dynamic properties of trajectories in the state space of asymmetric networks can include chaotic and limit cycle behaviors. It is easily seen that the dynamics of the nonlinear line attractor network presented in [57] does not guarantee global stability. It can only guarantee convergence and stability if the input pattern is sufficiently close to any of the trained pattern in its region of convergence. The symmetry of the synaptic connection matrix has been a constraint from the biological standpoint. Symmetry has been essential for the existence of a landscape picture for the dynamics of the network and asymmetry excluded such a landscape. Parisi pointed out in [60] that an attempt to implement a process of learning in symmetric artificial neural networks would encounter difficulties because every stimulus will quickly run into an attractor of the spin-glass phase which always accompanies the retrieval states. Consequently every stimulus will be perceived as a familiar pattern. In the next section, we show that the nonlinear line attractor network presented in [57] is specifically designed to operate between stable and unstable states. That is, when the network is able to reach equilibrium (*stable*), the input is considered as one of the stored patterns. Conversely, if the network is unable to reach equilibrium (*unstable*), the input is considered as pattern of another class.

7.2. Self-Organization Algorithm

The proposed self-organizing nonlinear line attractor network is designed to solve the issue of stability-plasticity problem [22] in a dynamical environment. That is, it is designed to perform between stability and instability. It uses the instability mode (*divergence*) to self organize in real time and produces stable associations while learning input patterns beyond those originally stored. Fig. 7.1 shows a basic concept for the self-organizing line attractor network. The system creates the first module F_1 with the set of training data. A new module F_2 is created using the data rejected (*unlearned data*) by module F_1 . This process continues by successively creating new modules until all the data are stored (*learned*).


Solid lines represent the convergence of patterns

Dash lines represent the divergence of patterns

The circular regions can be represented by the nonlinear line attractor network.

Figure 7.1: Illustration of the self-organization algorithm.

The self-organizing nonlinear line attractor network operates by testing the stability of the nonlinear line attractor networks. If a set of the training input vectors is not stable, a new line attractor will be created using the unstable data in the previous nonlinear line attractors. This would leave the stable data in the previous line attractors undisturbed.

Self-organization algorithm: The various operating stages in the self-organizing nonlinear attractor can be summarized as:

- 1. Initialize r = 1.
- 2. Create a nonlinear line attractor network F_r according to equations (1) to (5) presented in [102].
- 3. Calculate the threshold function by the following steps:
 - a. Calculate the difference between the predicted output and the actual output for synaptic weights between the i^{th} node and the j^{th} node for *P* patterns.
 - b. Calculate the error histogram as in Fig. 7.2.

- c. Find the minimum and maximum error starting from 0 in a continuous region.
- d. If there are minimum or maximum as shown in Fig. 7.2a, choose those values as threshold values, otherwise if there is no minimum or maximum as in Fig. 7.2b, set the threshold values to zero.



- 4. Perform a test with all the trained patterns on F_r network structure.
- 5. Form the new input set with the unstable input set obtained in step 4.
- 6. Prepare to create a new network by modifying r; r = r + 1.
- 7. Repeat step 2 through 7 until there are no more unstable inputs in step 5

Fig. 7.3 is an illustration of the formation of lines of the response of the i^{th} neuron due to the excitations of j^{th} neuron by the self-organization algorithm for k = 1. When inputs as shown in Fig. 7.3a are applied to the self-organizing algorithm, F₁ is created. The line (*represented by the green line*) created has a decision boundary (*represented by the red lines*) as shown in Fig. 7.3b. The diverged pattern will become a new set of inputs to the self-organizing algorithm for creating a new network F₂. The second set of inputs will create a decision boundary as shown in Fig. 7.3c. As a consequence, the final result contains 2 lines that have a decision boundary as shown in Fig. 7.3d.



Figure 7.3: Self organization of nonlinear line attractor network.

7.3. Dynamics of the Network

The dynamics of the system based on the self-organizing algorithm and the nonlinear line attractor network is presented in this section. The system can be used in two modes of operations, namely associative memory and pattern classifier. The system operates as a pattern classifier evolving in iteration t according to eqns. (3.1) in chapter 3 and eqns. (2.26) to eqns. (2.28) in chapter 2 [57]. The network is designed to operate between stable and unstable states. That is, when the network is able to reach equilibrium (*stable*), the input is considered as one of the stored patterns. Conversely, if the network is unable to reach equilibrium (*unstable*), the input is considered to be dissimilar to the stored patterns and therefore is considered as a different class.

7.3.2. Associative memory

As an associative memory, the network evolves in iteration t according to eqns (2.24) to eqns. (2.28) in chapter 2 [102]. The stability and associability of the nonlinear line attractor is examined in chapter 3[102].

7.4. Applications and Discussion

Experiments were conducted with images of faces from the CMU face expression database described in [107]. This is a database of 975 face images of 13 people captured with different expressions. The example images were of size 64×64 , with gray scale ranging from 0 to 255. The self-organizing nonlinear line attractor network was trained using k = 1 on a specific person class, with the goal of learning complex pattern manifolds of expressions.

In the pattern association task, example face images were corrupted by zeroing the pixels inside a 25×25 patches chosen at random locations. Fig. 7.4 shows a few examples of this experiment. The original image is corrupted by removing part of the image. After applying it to the network, it can be seen that the missing part of the face is filled within three iterations. We have compared all the reconstructed facial expressions with the original facial expressions. There are no significant differences between the original versions and the reconstructed versions. Additional experiments were also performed to the original images and the network is able to retain these trained images. That is, the network converges in 1 iteration if the original trained images are not modified.



Figure 7.4: Pattern reconstruction.

In the pattern classification task, example images drawn from another person class were used. Fig. 7.5 shows an example of the divergence dynamics of a pattern. The dynamics of the network can be interpreted as follows. At the first six iterations, the network tries to converge to one of the closest learned pattern in its memory. Since the presented face image is drawn from another person class, the image diverges finally. We have trained 13 people with each 20 continuous expressions using 13 networks, where each network stores each person. We have got 100% accuracy when testing the full database. That is, the network converges familiar patterns and diverges dissimilar patters.



Figure 7.5: Divergence of a pattern.

CHAPTER 8

HUMAN COMPUTER INTERACTION

Charles Darwin was one of the first scientists to recognize that facial expression is one of the most powerful and immediate means for human beings to communicate their emotions, intentions, and opinions to each other [112]. The computer has been integrated into the daily part of our lives. Yet, when it comes to the world of computers, there are situations where the man-machine interaction could be improved by having machines capable of adapting to their users. For example, the term "human computer interaction" suggests a two-way exchange, with each participant aware of the other and responding appropriately. Computers may appear frequently rude and indifferent. This can be attributed to the fact that current computers are almost completely unaware of the actual state of the human user. The goal of this project is to contribute to the development of a human computer interaction (HCI) environment in which the computer detects and tracks the user's affective states, and initiates communications based on this knowledge, rather than simply responding to user commands.

8.1. Perceptual User Interface

Human-Computer Interaction (HCI) is a research area aimed at making the interaction with computer systems more effective, easier, safer and more seamless for the users. Desktop-based interfaces, also referred to as WIMP-based (Windows, Icons, Menus and Pointers) Graphical User Interfaces (GUIs), have been the dominant style of interaction since their introduction in the 80s when they replaced command line interfaces. WIMP interfaces enabled access to computers to more people by providing the user with a look and feel, visual representation and direct control using the mouse and keyboard. Nevertheless, they have some intrinsic deficiencies: they passively wait for the user to carry out tasks by means of mouse or keyboard and often restrict input to single nonoverlapping events. As the way we use, computers are becoming more pervasive, it is not clear how GUI-WIMP interfaces will accommodate for and scale to broader range of applications. Therefore, post-WIMP interaction techniques that go beyond the traditional desktop metaphor need to be considered [108].

In the scientific community, a shared belief is that the next step in the advancement of computing devices and user interfaces is not to simply make faster applications but also to add more interactivity, responsiveness and transparency to them. In the last decade much more effort has been directed towards building multi-modal, multi-media, multi-sensor user interfaces that emulate human-human communication with the overall long-term goal to transfer to computer interfaces natural means and expressive models of communication [109]. Cross-disciplinary approaches have begun developing user-oriented interfaces that support non-GUI interaction by synergistically combining several simultaneous input and/or output modalities, thus referred to as multimodal user interfaces. In particular, multimodal Perceptual User Interfaces (PUI) [110] have emerged as potential candidates for being the next interaction paradigm. There are two key features of PUI. First, they are highly interactive. Unlike traditional passive interfaces that wait for users to enter commands before taking any action, perceptual interfaces actively sense and perceive the world and take actions based on goals and knowledge at various levels. (Ideally, this is an "active" interface that uses

"passive," or non-intrusive, sensing.) Second, they are multimodal, making use of multiple perceptual modalities (e.g., sight, hearing, touch) in both directions: from the computer to the user, and from the user to the computer. Perceptual interfaces move beyond the limited modalities and channels available with a keyboard, mouse, and monitor, to take advantage of wider range of modalities, either sequentially or in parallel [111].

8.2. Affective Computing

Not all computers need to pay attention to emotions, or to have emotional abilities. Some machines are useful as rigid tools, and it is fine to keep them that way. However, there are situations where the human-machine interaction could be improved by having machines naturally adapt to their users. The Human-Computer Interaction (HCI) community is showing increasing interest in the integration of affective computing in their technology. Particular attention is being paid to research on emotion recognition, since computer systems should be able to recognize human emotions in order to interact with humans in a more adaptive and natural, human-centered way. Affective computing expands human-computer interaction by including emotional communication together with appropriate means of handling affective information.

Facial expressions are the facial changes in response to a person's internal emotional states, intentions, or social communications. Facial expression analysis has been an active research topic for behavioral scientists since the work of Darwin in 1872 [112-115]. Suwa et al. [116] presented an early attempt to automatically analyze facial expressions by tracking the motion of 20 identified spots on an image sequence in 1978.

After that, much progress has been made to build computer systems to help us understand and use this natural form of human communication [117].

Facial expression analysis includes both measurement of facial motion and recognition of expression. The general approach to automatic facial expression analysis consists of three steps (Fig. 8.1): face acquisition, facial feature extraction, and facial expression recognition.



Figure 8.1: Conceptual diagram of facial expression recognition system.

Face acquisition is a processing stage to automatically find the face region for the input images or sequences. It can be a detector to detect face for each frame or just detect face in the first frame and then track the face in the remainder of the video sequence.

After the face is located, the next step is to extract and represent the facial changes caused by facial expressions. In facial feature extraction for expression analysis, there are mainly two types of approaches: geometric feature-based methods and appearance-based methods. The geometric facial features present the shape and locations of facial components including but not limited to the mouth, eyes, brows, and nose. The facial components or facial feature points are extracted to form a feature vector that

represents the face geometry. With appearance-based methods, image filters, such as Gabor wavelets, are applied to either the whole-face or specific regions in a face image to extract a feature vector. Depending on the different facial feature extraction methods, the effects of in-plane head rotation and different scales of the faces can be eliminated by face normalization before the feature extraction or by feature representation before the step of expression recognition.

Facial expression recognition is the last stage of the automatic facial expression analysis systems. The facial changes can be identified as facial action units or prototypic emotional expressions. Depending on if the temporal information is used, the facial expression can be classified as frame-based or sequence-based.

8.3. Face Acquisition

Object detection, and in particular, face detection, is an important element of various computer vision areas, such as image retrieval, video surveillance, human-computer interaction etc. The goal is to find an object of a pre-defined class in a static image or video frame. Sometimes this task can be accomplished by extracting certain image features, such as edges, color regions, textures, contours, etc. and then using some heuristics to find configurations and/or combinations of those features specific to the object of interest. But for complex objects, such as human faces, it is hard to find features and heuristics that will handle the huge variety of instances of the object class (e.g., faces may be slightly rotated in all three directions; some people wear glasses; some have moustaches or beards; often one half of the face is in the light and the other is shadow,

etc.). For such objects, a statistical model (classifier) may be trained instead and then used to detect the objects.

Statistical model-based training takes multiple instances of the object class of interest, or "positive" samples, and multiple "negative" samples, i.e., images that do not contain objects of interest. Positive and negative samples together make a training set. During training, different features are extracted from the training samples and distinctive features that can be used to classify the object are selected. This information is "compressed" into the statistical model parameters. If the trained classifier does not detect an object (misses the object) or mistakenly detects the absent object (i.e., gives a false alarm), it is easy to make an adjustment by adding the corresponding positive or negative samples to the training set.

To build a system capable of automatically labeling features on the face it is first necessary to localize the face in the image. Recently, Viola and Jones [118] introduced an impressive face detection system capable of detecting faces in real-time with both high detection rate and very low false positive rates. The Viola-Jones detector consists of three parts. The first is an efficient method of encoding the image data known as an "integral image". This allows the sum of pixel responses within a given sub-rectangle of an image to be computed quickly and is vital to the speed of the Viola-Jones detector. The second element is the application of a boosting algorithm known as AdaBoost [119] to select appropriate features that can form a template to model human face variation. The third part is a cascade of classifiers that speeds up the search by quickly eliminating unlikely face regions. Viola et al claims a 15 frame per second rate on a 700 MHz Pentium III

when processing images with 384×200 dimensions. This speed is obtained by using a cascade of simple Haar-like features to progressively filter out non-face.



Figure 8.2: Haar features use for Viola Jones detector.

Using simple templates such as those in the above image (Fig. 8.2), the intensity ratios for sub-windows within the image can be calculated. For example, when template b is applied to the face, the value of this feature would be the sum of the pixel intensities in the white section over the sum of the intensities in the black section. Similarly, for more complex templates c and d, the value is the ratio of the sum of intensities in the white sections over those in the black. These sub-windows can be scaled to any size to find features over any sub-window within the image.

8.3.1. Integral Image

Since summing the pixel intensities many times over can be a slow undertaking, the Viola-Jones detector use the integral image representation to improve the speed. Let i(x, y) be the intensity value of an image at points x and y. We'll define the integral image to be:

$$ii(x, y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x', y')$$
 (8.1)

By using the following recurrences:

$$s(x, y) = s(x, y-1) + i(x, y)$$
(8.2)

$$ii(x, y) = ii(x-1, y) + s(x, y)$$
 (8.3)

and assuming that s(x,-1)=0 and ii(-1, y)=0, the integral image can be computed in a single pass over the image.



Figure 8.3: Integral image.

For example in Fig 8.3, the sum of intensities at region A would be the value of the integral image at point 1. The sum of intensities in region B would be the value of 2 minus those at point 1. For region D, the sum would be 4 + 1 - (2 + 3).

8.3.2. AdaBoost

Given a training set containing labeled examples of faces (positive) and non-faces (negative), a complex and robust classifier is built by multiple weak classifiers using a procedure called boosting, introduced by Freund and Schapire [119]. The boosted classifier is built iteratively as a weighted sum of weak classifiers:

$$h = sign\left[\alpha_1 h_1(x) + \alpha_2 h_2(x) + \dots + \alpha_n h_n(x)\right]$$
(8.4)

On each iteration, a new weak classifier h_i is trained and added to the sum. The smaller the error h_i gives on the training set, the larger is the coefficient α_i that is assigned to it. The weight of all the training samples is then updated, so that on the next iteration the roles of those samples that are misclassified by the already built *h* are emphasized.

Viola Jones detector assumes that out of the hundreds of thousands of possible features within a window, only a small number are necessary to form an effective strong classifier. Thus, the weak classifier is defined as follows:

$$h_{j}(x) = \begin{cases} 1 & \text{if } p_{j}f_{j} < p_{j}\theta_{j} \\ 0 & \text{otherwise} \end{cases}$$
(8.5)

Where x is the window in which we are searching for a face. Therefore, the weak classifier is comprised of the feature f_j , the threshold θ_j and a polarity p_j .

Given a set of example images $(x_1, y_1), \dots, (x_n, y_n)$ where y_i is 0 for negative examples, and 1 for positive examples, a strong classifier is created using the following procedures:

1. Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where *m* and *l* are the

number of negatives and positives respectively.

2. For t = 1, ..., T:

- 1. Make w_t the probability distribution: $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
- 2. For each feature f_j , train a classifier h_j . This implies selecting θ_j and p_j which produce the lowest error. The error for the classifier h_j wholly comprising of feature f_j , threshold θ_j , and polarity p_j (as mentioned previously), is $\varepsilon_j = \sum_i w_i |h_j(x_i) - y_i|$
- 3. Choose the classifier h_t with the lowest error ε_t
- 4. Update the weights of the examples for the next round $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$ where $e_i = 0$ if example x_i is correctly classified, otherwise

$$e_i = 1$$
, and $\beta_i = \frac{\varepsilon_i}{1 - \varepsilon_i}$.

After *T* iterations, the resulting strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{i=1}^{T} \alpha_i h_i(x) \ge \frac{1}{2} \sum_{i=1}^{T} \alpha_i \\ 0 & \text{otherwise} \end{cases}$$
(8.6)

where $\alpha_t = \log \frac{1}{\beta_t}$

8.3.3. Cascade of Filters

To increase the speed of the detector, it is best to remove as many non-face sub-windows from consideration as possible early on. The key insight is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances. Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates. Stages in the cascade are constructed by training classifiers using AdaBoost. Starting with a feature strong classifier, an effective face filter can be obtained by adjusting the strong classifier threshold to minimize false negatives. The initial AdaBoost threshold $\frac{1}{2}\sum_{t=1}^{T} \alpha_t$ is designed to yield a low error rate on the training data. A lower threshold yields higher detection rates and higher false positive rates. Based on performance measured using a validation training set, classifier can be adjusted to detect 100% of the faces with a more false positive rate (40%). Although the detection performance of the classifier is far from acceptable as an object detection system, the classifier can significantly reduce the number of sub-windows that need further processing with very few operations

The overall form of the detection process is that of a degenerative decision tree shown in Fig. 8.4. A positive result from the first classifier triggers the evaluation of a second classifier which has also been adjusted to achieve very high detection rates. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window.



Figure 8.4: Cascade of filters.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Algorithm for training a cascade of classifiers is as follow:

- User selects values for *f*, the maximum acceptable false positive rate per layer and *d*, the minimum acceptable detection rate per layer.
- 2. User selects target overall false positive rate F_{target} .
- 3. P = set of positive examples
- 4. N =set of negative examples

5.
$$F_0 = 1.0; D_0 = 1.0; i = 0$$

- 1. While $F_i > F_{target}$
- 2. *i*++
- 3. $n_i = 0; F_i = F_{i-1}$
 - 1. while $F_i > f \ge F_{i-1}$
 - 2. $n_i ++$
 - Use P and N to train a classifier with n_i features using
 AdaBoost
 - 4. Evaluate current cascaded classifier on validation set to determine F_i and D_i
 - 5. Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least $d \ge D_{i-1}$ (this also affects F_i)
 - 6. $N = \emptyset$
 - 7. If $F_i > F_{target}$ then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set *N*.

8.3.4. Experimental Results

We have implemented the full Viola-Jones detector to localize the face in the image. The output of the face detector is an image region containing the face. Figure 8.5 shows sample results of the Viola Jones detector. It can be seen that the face detector successfully detects faces across all sizes. We have also tried this implementation for video processing on a Pentium IV 2.8GHz machine with 512MB of memory; the face detector can process a 360 by 240 pixel image in about 0.019 seconds, which is equivalent of 52 frames per second.



Figure 8.5: Face detection results.

8.4. Facial Features Extraction

A fast and efficient face detection method has been devised in the previous section. It relies on the AdaBoost algorithm and a set of Haar Wavelet like features. A natural extension of this approach is to use the same technique to locate individual features within the face region. However, we find that there is insufficient local structure to reliably locate each feature in every image, and thus local models can give many false positive responses. In this section, we describe an algorithm capable of accurately and reliably detecting facial features on both high and low resolution image sets.

Active Appearance Models (AAM) was first described by Cootes, Edwards and Taylor in 1998 [120]. AAM is a statistical model describing an object's parameters. It combines both shape and texture, resulting in appearance. Shape is to be understood as the outlining contours of the object plus some inner edges corresponding to facial features. Appearance describes the texture of the object in a shape free space. The model becomes 'active' by being able to learn its statistical borders of representation in a one time training session. By learning a model from annotated images, one can prevent blind optimization in run time, which would slow the online process down. Instead, it is possible to optimize similarities in the model offline significantly speeding up later convergence in the underlying high dimensional space of the model.

8.4.1. Active Appearance Model

The Active Appearance Model (AAM) is a generalization of the widely used Active Shape Model approach, but uses all the information in the image region covered by the target object, rather than just that near modeled edges. An AAM contains a statistical

148

model of the shape and grey-level appearance of the object of interest which can generalize to almost any valid example. Matching to an image involves finding model parameters which minimize the difference between the image and a synthesized model example, projected into the image. The potentially large number of parameters makes this a difficult problem. The process is described in Fig. 8.6.



difference

in image



Iterate

Figure 8.6: AAM Procedure.

It's observed that displacing each model parameter from the correct value induces a particular pattern in the residuals. In a training phase, the AAM learns a linear model of the relationship between parameter displacements and the induced residuals. During search it measures the residuals and uses this model to correct the current parameters, leading to a better fit. A good overall match is obtained in a few iterations, even from poor starting estimates. Fig 8.7 shows frames from an AAM search for a new face, each starting with the mean model displaced from the true face centre.



Figure 8.7: AAM example.

149

The construction of the AAM and the matching procedure are introduced in this section.

8.4.1.1. Shape Formation

AAM handles planar shapes as a finite set of landmarks. The representation used for a single *n*-point shape is:

$$x = [x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n]^T$$
(8.7)

For dealing with redundancy in multivariate data – such as shapes – we performed dimensionality reduction on the weight matrix obtained by the nonlinear line attractor network. In this setting a shape of n points is thus considered one observation, x_i , in a 2n dimensional space.

The shape representation is essentially an eigen-analysis of the weight matrix of the nonlinear line attractor network of the shapes aligned with respect to position, scale and rotation, i.e. after a Procrustes analysis. New shape instances can thus be synthesized by deforming the mean shape, \bar{x} , using a linear combination, b_s , of the eigenvectors, Φ_s :

$$x = x + \Phi_s b_s \tag{8.8}$$

Thus, the points of the shape are transformed into a *modal representation* where modes are ordered according to the percentage of variation that they explain. To regularize our solution space and improve performance, modes are included until the cumulated variation explained by the model is above a certain threshold (e.g. 95%).

8.4.1.2. Texture Formation

Contrary to the prevalent understanding of the term texture in the computer vision community, the definition of texture is defined as "The pixel intensities across the object in question (if necessary after a suitable normalization)" in the context of AAM. For m samples over the object surface, the texture is represented as:

$$g = \left[g_1, g_2, \cdots, g_m\right]^T \tag{8.9}$$

In the shape case, the data acquisition is straightforward because the landmarks in the shape vector constitute the data itself. In the texture-case one needs a consistent method for collecting the texture information between the landmarks, i.e. an image sampling function needs to be established. This can be done in several ways. Here, a piece-wise affine warp based on the Delaunay triangulation of the mean shape is applied.

Following the warp from an actual shape to the mean shape, a normalization of the g-vector set is performed to avoid the influence from global linear changes in pixel intensities. Hereafter, the analysis is identical to that of the shapes. Hence, a compact representation is derived to deform the texture in a manner similar to what is observed in the training set:

$$g = \overline{g} + \Phi_g b_g \tag{8.10}$$

Where \overline{g} is the mean texture; Φ_g denotes the eigenvectors of the weight matrix of the nonlinear line attractor network and finally b_g is the set of texture deformation parameters.

8.4.1.3. Combined Model Formation

To remove correlation between shape and texture model parameters - and to make the model representation even more compact - a third dimensionality reduction is performed on the shape and texture of the training set, b to obtain the combined model parameters, c:

$$b = Qc \tag{8.11}$$

which are directly obtained due to the linear nature of the model:

$$b = \begin{bmatrix} W_s b_s \\ b_s \end{bmatrix} = \begin{bmatrix} W_s \Phi_s^T \left(x - \overline{x} \right) \\ \Phi_g^T \left(g - \overline{g} \right) \end{bmatrix}$$
(8.12)

A suitable weighting between pixel distances and pixel intensities is obtained through the diagonal matrix W_s .

Now, a complete model instance including shape, x and texture, g, is generated using the combined model parameters, c:

$$x = \overline{x} + \Phi_s W_s^{-1} Q_s c \tag{8.13}$$

$$g = \overline{g} + \Phi_g Q_g c \tag{8.14}$$

Regarding the compression of the model parameters, one should notice that the rank of Q will never exceed the number of examples in the training set.

8.4.1.4. AAM Matching

From the current estimate of the model parameters c_0 and the parameter derivatives for the model, texture and pose parameters (matrices R_c , $R_t \& R_p$ respectively), Cootes describes an iterative matching algorithm, consisting of the following steps:

- 1. Calculate the residual between target image and model patch $\delta g_0 = g_s g_m$.
- 2. Calculate the intensity error $E_0 = |\delta g_0|^2$
- 3. Using the pre-computed gradient matrices, determine the model parameter update $\delta c = R_c \delta g_0$, pose update $\delta p = R_p \delta g_0$ and texture update $\delta t = R_t \delta g_0$
- 4. Set k = 1 and determine a new estimate for the model parameters $c_1 = c_0 k\delta c$, pose parameters $p_1 = p_0 - k\delta p$ and texture parameters $t_1 = t_0 - k\delta t$
- 5. Calculate a new model based on c_1 , $p_1 \& t_1$,

- 6. Determine a new difference-vector and calculate error E_1 ,
- 7. If $E_1 < E_0$, select c_1 , $p_1 \& t_1$ as the new parameter vectors, else try k = 1.5, k = 0.5, k = 0.25, k = 0.125 etc and go to step 4.

Repeat until convergence (either using a fixed number of iterations, or until no improvement is achieved).

8.4.2. Experimental Results

To implement the AAM to perform tracking, we use a training set consisting of 1066 images of three different persons from different angles and with different facial expressions. To overcome the problem of color constancy and lighting variations in the environment, we have employed near infrared (NIR) imagery. The use of NIR imagery brings a new dimension for applications of invisible lights for computer vision and pattern recognition. This type of imaging hardware device not only provides appropriate active frontal lighting but also minimizes lightings from other sources, making it very suitable for facial feature tracking algorithm such as the AAM. Fig. 8.8 shows example images of a face illuminated by both frontal NIR and a side environmental light. It can be seen that the NIR provide constant illumination.



Figure 8.8: Visible imagery and near infrared imagery.

In order to create more examples and to enhance the tolerance to low resolution image, motion, and variations in intensity, a series of smoothing operations are applied to the initial set of examples. This step is to teach the system how to cope with situations where the AAM is fed with a weak, over smoothed or poorly-contrasted signal. Finally, the training set reached the number of 3198 face images. Some example of the training sample is shown in Fig. 8.9.



Figure 8.9: Sample face images used for training.

Each training image was subsequently annotated using 23 landmark points as shown in Fig. 8.10.



Figure 8.10: Hand annotated image.

The tracker performed successful tracking in 38 frames/sec at 320×240 resolution with minimum misalignment to the face feature for natural human head motion, which is typically ranges between 70° - 90° of downward pitch, 55° of upward pitch, 70° of turn, and 55° of tilt. Examples are given in Fig. 8.11.



Figure 8.11: Automatic face tracking using AAM.

8.5. Facial Expression Recognition

People mind read or infer mental states to others all the time, effortlessly, and mostly subconsciously. Mind reading allows us to make sense of other people's behavior, predict what they might do next, and how they might feel. The ability to mind read is essential to the social functions we have taken for granted. Roz Picard, a professor at MIT, believes that computers should also understand and exhibit emotion. She believed that the ability to detect and influence affective states in others is important in human communication and will be necessary for machines to interact effectively with humans. For example, Picard offers the following hypothetical example of a computerized piano tutor [121]:

"Imagine that you are seated with your computer tutor, and... it can also read your emotional state. In other words, it not only interprets your musical expression, but also your facial expression and perhaps other physical changes corresponding to your emotional feelings--maybe heart rate, breathing, blood pressure, muscular tightness, and posture. ... Given affect recognition, the computer tutor might gauge if it is maintaining your interest during the lesson, before you quit out of frustration and it is too late for it to try something different. ... If, however, it detects you are frustrated and making lots of errors, then it might slow things down and proffer encouraging feedback. ... The principles in the piano tutor scenario hold also for non-musical learning tasks--learning a software package, a new game, a foreign language, and more. The topic can vary, but the problem is the same: how should the computer adapt the pace and presentation to the user? How can it know when to provide encouraging feedback or to offer assistance? Certainly, the user should have the option to ask for this at any time; however, it has also been demonstrated that systems that proactively offer suggestions can provide a better learning experience." (p. 17)

Without the ability to recognize a person's emotional state, computers will remain at the most trivial levels of endeavor.

8.5.1. Extraction of the Facial Action

A number of studies have shown that the visual properties of facial expressions can be described by the movement of points belonging to facial features (114, 122, and 123). These feature points are typically located on the eyes and eyebrows for the upper face, the lips and nose for the lower face. Fig. 8.12 illustrates the 2D face model of the 23 feature points used throughout this chapter. By tracking these feature points over an image sequence and analyzing their displacements over multiple frames, a characteristic motion pattern for various action units (AUs) can be established.



Right outer eyebrow
 Right eyebrow center
 Right inner eyebrow
 Left outer eyebrow
 Left eyebrow center
 Left inner eyebrow
 Right outer eye corner
 Right upper eye center
 Right inner eye corner
 Right lower eye center
 Right lower eye center
 Right eye pupil
 Left outer eye corner

13. Left upper eye center
 14. Left inner eye corner
 15. Left lower eye center
 16. Left eye pupil
 17. Right nostril
 18. Left nostril
 19. Right mouth corner
 20. Upper lip center
 21. Left mouth corner
 22. Lower lip center
 23. Mouth center

Figure 8.12: Face model. 23 feature points track by AAM.

To understand the AU, one needs to first understand the Facial Action Coding System (FACS) system, which Ekman and Friesen developed in 1976 as an objective technique for measuring facial movements [124]. The FACS is a human observer based system designed to describe subtle changes in facial features. FACS consists of 44 action units, when people make faces – whether spontaneous expressions or deliberate contortions--they engage muscles around the eyes, mouth, nose and forehead. With FACS, Ekman and Friesen detailed which muscles move during which facial expressions.

Table 8.1 describes how the head AUs that are currently supported are measured. Table 8.2 describes the facial AUs. We have only chosen those AUs relevant to the facial expression on which we chose to focus.

Action	AU	Description	Lexamples.	SMeasurement	
Yaw	51	Turn left		$\frac{\operatorname{Dist}(P_7, P_9)}{\operatorname{Dist}(P_{14}, P_{12})} > h_t$	
	52	Turn right		$\frac{\operatorname{Dist}(P_{14}, P_{12})}{\operatorname{Dist}(P_{7}, P_{9})} > h_{t}$	
Pitch	53	Head up		Vertical displacement of $\left[\frac{1}{2}(P_{17}+P_{18})\right][t,t-1]$	
	54	Head down			
Roll	55	Tilt left		$\mathrm{Slope}(P_9, P_{14}) > h_a$	
	56	Tilt right		$\operatorname{Slope}(P_{14}, P_9) > h_a$	

Table 8.1: Measurement of the Head AUs with respect to the points in Fig. 8.11.

where Dist(A,B) is the distance between feature point A and feature point B, h_t is the threshold of the left eye and the right eye ratio, \vee is the logical OR operation, t is the

current video frame, Slope (A,B) is the slope between feature point A and feature point B, and h_a is the threshold of the slope of the left eye and the right eye.

ACION	ALL ALL	Descaphion -	Nample &	Ancasuroment second
Lips	12	Lip corner pull		$\frac{\text{Dist}(P_{19}, P_{21})[t]}{\text{Dist}(P_{19}, P_{21})[0]} > h_{l_c}$
	18	Lip pucker		$\frac{\text{Dist}(P_{19}, P_{21})[t]}{\text{Dist}(P_{19}, P_{21})[0]} < h_{l_p}$
Mouth	27	Mouth stretch		$\operatorname{Dist}\left\{ \begin{bmatrix} \frac{1}{2} (P_{19} + P_{21}) \\ P_{23} \end{bmatrix} > h_{m_{i}} \\ \begin{pmatrix} \frac{\operatorname{Dist} (P_{20}, P_{23})}{\operatorname{Dist} (P_{23}, P_{22})} > h_{m_{r}} \\ \vee \\ \frac{\operatorname{Dist} (P_{23}, P_{22})}{\operatorname{Dist} (P_{20}, P_{23})} > h_{m_{r}} \\ \end{pmatrix} \right\}$
Eyebrows	1 and 2	Eyebrow raise		$\frac{\frac{\text{Dist}(P_1, P_7)[t]}{\text{Dist}(P_1, P_7)[0]} > h_{e_r}}{\frac{\text{Dist}(P_3, P_9)[t]}{\text{Dist}(P_3, P_9)[0]} > h_{e_r}}$
	4	Eyebrow drop		$\frac{\frac{\text{Dist}(P_6, P_{14})[t]}{\text{Dist}(P_6, P_{14})[0]} > h_{e_d}}{\stackrel{\wedge}{\frac{\text{Dist}(P_4, P_{12})[t]}{\text{Dist}(P_4, P_{12})[0]}} > h_{e_d}}$

Table 8.2: Measurement of the facial AUs with respect to the points in Fig. 8.11.

where Dist(A, B)[t] is the distance between feature point A and feature point B at current frame, Dist(A, B)[0] is the distance between feature point A and feature point B at the initial frame, h_{l_c} is the threshold for the lip corner pull comparing the lip size of the initial frame and current frame, h_{l_p} is the threshold for the lip pucker comparing the lip size of the initial frame and current frame, h_{m_t} and h_{m_r} are the threshold for the mouth stretch, h_{e_r} is the threshold for the eyebrow raise comparing the distance of the eye and eyebrow of the initial frame and the current frame, and h_{e_d} is the threshold for the eyebrow drop comparing the distance of the eye and eyebrow of the initial frame and the current frame.

8.5.2. Affective Cue Computation

In the computational model of affective states, affective cues serve as an intermediate step between tracked AUs and inferred affective states. Table 8.3 lists the nine head and affective cues that are currently supported by our system and their underlying actions.

Affective Cue	Description				
Upod mod	Alternating head up (AU53) and head down (AU54)				
nead nod	actions				
Hood shake	Alternating head turn left (AU51) and head turn right				
	(AU52) actions				
Head tile	Tilt in one direction (AU55 or AU56)				
Head turn	Pose of turned head (sequence of AU51 or AU52)				
Lip pull	Lip corner pull (AU 12)				
Lip pucker	Lip pucker (AU18)				
Mouth open	Mouth stretch (AU27)				
Eyebrow raise	Eyebrow raise (AU1 + AU2)				
Eyebrow drop	Eyebrow drop (AU4)				

Table 8.3: List of affective cues and their component AUs.

A head nod or a head shake is composed of periodic motion that recurs at regular intervals. Fig 8.13 shows the temporal structure of a natural head shake, which is characterized by alternating head-turn-right, head-turn-left motion cycles. The cycle time is the interval of time during which a sequence of recurring motions is completed. The rest of the head and facial displays listed in Table 8.3 are characterized by the action unit they are associated with.



Figure 8.13: Dynamics of head shake motion.

8.5.2.1. Head Nod Detection

The y - displacement of the nose tip, which is the center of feature points P_{17} and P_{18} , is used to distinguish between upward (positive) and downward (negative) motions. The detected head motions are accumulated and are then processed by a head nod detector. The head nod detector parses through consecutive motions checking whether if they constitute a valid head nod gesture. This head nod state machine, shown in Fig. 8.14, is consists of eight states. A nod can start with an upward or a downward motion. If no such motion is found (i.e. leftward, rightward, or even combination of updown-leftright), the state machine returns to the check state. The head nod state machine looks for consecutive alternating head up, head down motions. A successful nod is returned when the minimum number of alternating motions is detected. In addition, a minimum threshold for purposeful head motion is imposed on the y displacements.



Figure 8.14: Head nod detection module.

8.5.2.2. Head Shake Detection

The x-displacement of the nose tip is used to distinguish between leftward (negative) and rightward (positive) motions. The detected head motions are accumulated and are then processed by a shake state machine. The state machine parses through consecutive

detector is shown in Figure 8.15.



Figure 8.15: Head shake detection module.
8.5.2.3. Experimental Evaluation

This level of the system has been tested at Old Dominion Vision Laboratory and the Center of Advanced Engineering Environments (CAEE). On both occasions the system ran successfully, detecting the affective cues in real time. Examples of the head nod and head shake detection are shown in Fig. 8.16(a) and Fig. 8.16(b) respectively.



(a) Frame 14 to frame 22 of a video illustrating head nod

Ne ™a	The r	i da i da i da i da i da i da i da i da
	đ teor	and the second sec

(b) Frame 200 to frame 209 of a short video illustrating head shake

Figure 8.16: Head nod and head shake detection.

Result of the head tile, head turn, lip pull, lip pucker, mouth open, eyebrow raise, and eyebrow drop is illustrated in Fig. 8.17.



(e) Eyebrow drops detection Figure 8.17: Computation of affective cues.

8.5.3. Affective States Computation

A person's mental state is not directly available to an observer. Instead it is communicated through nonverbal cues of the face of a person. Section 8.5.1 and 8.5.2 describe the recognition of the facial features and the affective cues from continuous video stream in real time. In this section, the output of the affective cues from the previous section will be used to compute the affective states of a person.

The process of reading the mind in the face is inherently uncertain. People can express the same mental state using different facial expressions. Moreover, the recognition of affective cues is in itself a noisy process. To account for this uncertainty, Kaliouby et al [125] use a multi-level representation of the video input, combined in a Bayesian inference framework, specifically the dynamic Bayesian networks, for developing their mind reading machine [125]. They have used the Mind Reading DVD [126], a computer-based guide to emotions, developed by a team of psychologists led by Simon Baron-Cohen at the Autism Research Centre in the University of Cambridge to train the statistical classifiers in their inference system. This DVD was originally designed to help individuals diagnosed along the autism spectrum recognize facial expressions of emotions.

The DVD is based on taxonomy of emotion by Baron-Cohen et al. [127] that covers a wide range of affective and cognitive mental states. The taxonomy lists 412 mental state concepts, each assigned to one (and only one) of 24 mental state classes. Out of the 24 classes, they have focus on the automated recognition of 6 classes that are particularly relevant in a human-computer interaction context, and that are not in the basic emotion set. The 6 classes are agreeing, concentrating, disagreeing, interested, thinking and unsure.

The parameters of the dynamic Bayesian networks developed by Kaliouby [128] are summarized in Fig. 8.18 - 8.23. They depict the conditional probability distribution tables and discriminative-power heuristic for each display (or affective cue) and mental state (or affective state) combination. Fig. 8.18 shows the results of parameter estimation for agreeing. The error bars depict the effect of the size and choice of training examples on the parameters. It can be seen that while the lip corner pull is the most likely display to occur, the head nod is the most discriminative.



Figure 8.18: Agreement (source: [128]).

Fig. 8.19 shows the results of parameter estimation for disagreeing. The head shake is the most likely display to occur and it is also the most discriminative.



Figure 8.19: Disagreement (source: [128]).

Fig. 8.20 shows the results of parameter estimation for concentrating. The presence of teeth is the most likely display to occur; the head tilt is the most discriminative.



Figure 8.20: Concentration (source: [128]).

Fig. 8.21 shows the results of parameter estimation for interested. The eyebrow raise, mouth open and presence of teeth are the most likely displays and the most discriminative.



Figure 8.21: Interested (source: [128]).

Fig. 8.22 shows the results of parameter estimation for thinking. The head tilt and the head turn are the most likely displays to occur and are also the most discriminative.



Figure 8.22: Thinking (source: [128]).

Fig. 8.23 shows the results of parameter estimation for unsure. While the head turn is the most likely display to occur, the absence of teeth and closed mouth are the most discriminative.



Figure 8.23: Unsure (source: [128]).

The mind reading machine developed by Kaliouby [128] was tested on an Intel Pentium IV, 3.4 GHz processor with 2 GB of memory. The processing time at each of the levels of the system is summarized in Table 8.4.

level	Tracking (wavelet based)	Action level / Action unit	Display-level / affective cue	Mental state-level / affective state	Total
Time (ms)	3.00	0.09	0.14	41.10	44.33

Table 8.4: Processing time of Kaliouby (source: [128]).

Kaliouby's system [128] basically abstracts raw video input into three levels:

- Actions are explicitly coded based on the facial feature point return from the FaceTracker
- Displays are recognized by Hidden Markov Models (HMMs)
- Mental states are assigned probabilities by DBNs

The facial feature tracker Kaliouby [128] used is named, FaceTracker, It is part of Nevenvision's commercial facial feature tracking [129] software development kit (SDK).

FaceTracker uses a generic face template to bootstrap the tracking process, initially locating the position of 22 facial landmarks. To track the motion of the points over a live or recorded video stream, the tracker uses a combination of Gabor wavelet image transformations and neural networks.

To improve the performance in terms of frames per second for mind reading, we proposed to capture Kaliouby's system using a binary decision tree structure composed of nine affective cues. Note that we do not claim our system to be better in recognizing user's mental states compared to the Kaliouby's mind reading system. Our system is merely an alternative (substitute) in which the trade off between the computation time and the accuracy of the inference is important. Therefore, in this dissertation work, we have encapsulated the affective states shown in Fig. 8.18 – 8.23 using a binary decision tree process shown in Fig. 8.24. The thinking, unsure, interested, concentration, and angry states are further broken down using the affective cues we have derived and it is shown in Fig 8.25 and 8.26.



Figure 8.24: Affective state computation.



Figure 8.25: Computation of thinking and unsure.



Figure 8.26: Computation of interested, concentration, and angry.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The affective states machined developed by us was tested on an Intel Pentium IV, 2.8 GHz processor with 512 MB of memory. The processing time at each of the levels of the system is summarized in Table 8.5.

Level	Face	Feature	Action	Affective	Affective	total
	detection	tracking	unit	cue	state	
Time (ms)	1.90	0.70	0.08	0.03	0.06	2.77

Table 8.5: Processing time of the proposed system.

This resulted in a frame rate of 36 frames per second. It is imperative to understand that the choice of the order of the binary decision tree seen on Fig. 8.24 - 8.26 is designed in such a way that it prioritized the affective states we desire. For example, in our application, we detect the agreement first and the disagreement second since these gestures play an important role in our advance human-computer interaction environment.

8.5.3.1. Experimental Evaluation

We have developed a working prototype of automatic facial expression analysis system that is specifically designed for intelligent HCI. This system is capable of running in real time, it require no user intervention in segmentation or any other forms of manual preprocessing. Furthermore, it has been successfully intergraded into the advance automatic learning system developed by the CAEE. The users of this system reported real time performance, high accuracy, and reliability of the system. Figure 8.27 shows the diagram which gives a brief description of the current CAEE's advance automatic learning system.



Figure 8.27: Human computer interaction.

Multimodal input in user interfaces has been implemented and integrated in CAEE's system as a means to obtain more efficient and robust computer human interaction. They have incorporated both speech recognition technology and affective cues and affective states developed by us to eliminate the used of keyboard and mouse as input for the advance human computer interaction system. Fusing head gestures with results from visual analysis of the environment provides rich vocabularies for human-machine communication because it renders the environment into an interface. Examples of the affective states computation are shown in Fig. 8.28 (a) to (g) respectively.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



(g) Angry detection Figure 8.28: Affective states computations.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

CHAPTER 9

CONCLUSION

We propose a nonlinear line attractor network based on the observation that images of similar visual perceptions reside in a complicated manifold in the low dimensional image space. That is, manifolds are fundamentals to perception. The perceived features are often highly structured and hidden in a complex set of relationships or high-dimensional abstractions. To model the manifold in the state space as an associative memory, we presented a novel learning algorithm using a recurrent neural network. Contrary to most models of the recurrent networks, in which the dynamics of the network can exhibit point attractors, the proposed network encapsulates the point attractors scattered in the state space as a nonlinear line of attraction, where the least squares estimation approach utilizing the interdependency between neurons defines the dynamics of the network. The regions of convergence around the attractors are defined based on the statistical characteristics of the input patterns. Experiments conducted on benchmark problems have shown that the proposed network outperforms other novel and classical approaches. Furthermore, unlike the conventional model in which the dynamics of the network is designed to be stable, the dynamics of this nonlinear line attractor network can also be designed to operate between stable and unstable states, where the stable states signify know pattern and the unstable states signify unknown pattern. Experiments conducted on several benchmark problems have shown that the network can learn and classify patterns successfully.

An extension of homomorphic filter was presented in section 4.1 of this dissertation. The new method improves digital images using a homomorphic system and the Ratio Rule learning algorithm. The proposed method provides simultaneous improvement of the dynamic range and color rendition to digital color images. Moreover, it opens the possibility for applying any gray level enhancement algorithm in the color domain. This research also demonstrates the use of an associative memory that encapsulates instances of a color for digital color image enhancement process.

Based on section 4.1, a general framework to apply gray level image enhancement algorithm to the color domain was proposed in section 4.2 of this dissertation. A single layer fully connected recurrent neural network trained using the Ratio learning algorithm was used as an associative memory to represent the original pixels relationship of an image as a line of attraction in the state space. Its dynamics is then used for recalling the original color characteristics to produce natural color rendition. We have described most of the common methods of enhancing an image and compared the results with and without our color image enhancement process. We have shown that our enhancement process can provide simultaneous improvement to the dynamic range and color rendition to digital color images by simply applying gray level image enhancement algorithm in color domain.

In section 4.3, we have presented a new technique for color image enhancement based on a nonlinear line attractor network. The proposed method involves color characterization, enhancement, and color balancing processes. It has been observed that the new approach outperforms many classical image enhancement algorithms. Furthermore, it was shown that the proposed method provides both good dynamic range compression and good color rendition compared to other classical methods of enhancement. In addition, we have introduced the use of histograms for performing visual quality assessment for different image enhancement algorithms. More importantly, this demonstrates the use of an associative memory that represents patterns as a manifold for a color image enhancement process.

In section 4.4, we proposed the concept of the manifold of color perception in which a given image will automatically converge to the trained color perception. To explore the structure of the manifold of color perception in a lower dimensional space, we have utilized the gray world model using the center-surround concept of human vision modeling. It is found that the proposed model is a generalization of the gray world model. A nonlinear line attractor network has been presented for modeling the proposed manifold. The recurrent neural network utilized the least squares estimation approach to approximate the relationship between the relative magnitudes of the output of a neuron with respect to the output of another neuron. Experiments conducted on several benchmark problems have shown that the proposed model can learn the manifold of color perception successfully.

In chapter 5, we have tested the unstable dynamics of the nonlinear line attractor network for pattern classification. The technique is based on a mathematical model described by the characteristics of the patterns under consideration. The method of leastsquares estimation has been utilized to approximate the relationship between the relative magnitudes of the output of a neuron with respect to the output of another neuron. The threshold function has been represented by the statistical features of the input patterns. Experiments conducted with the proposed learning algorithm on color images for human skin region extraction have shown that the network is capable of associating multiplevalued patterns.

A nonlinear dimensionality reduction technique was proposed in chapter 6 and used for invariant facial recognition. The current implementation of the line attractor network models the image pixels using a first-order polynomial. A cursory examination of the FRGC version 2 database suggests that a more accurate model could be achieved with at least a second-order polynomial. Initial evaluations showed that recognition accuracy could be improved by training and validating the system on recordings of larger size. Finally, we believe recognition rates for the fourth FRGC version 2 experiment could be improved using the image enhancement algorithm described in section 6.3 of this dissertation.

A novel learning strategy based on stability (*convergence*) and instability (*divergence*) criteria of a nonlinear line attractor network was presented in chapter 7 of this dissertation. These criteria can be used to circumvent the plasticity-stability dilemma by using the instability mode as an indicator to create a new line for an unfamiliar pattern. This self-organizing behavior of the nonlinear line attractor model helps to create complex dynamics in an unsupervised manner. The dynamics of the system can be used in two modes of operation, namely associative memory and pattern classifier. In the associative memory mode, the network is used for retrieving information. Conversely, in the classification mode, the network is used to discriminate information. The training of the nonlinear line attractor network is very fast since the only main computation is on a least squares fit of linear models. The network has very fast convergence dynamics due to the convergence to a line of attraction rather to several fixed points.

Finally in chapter 8, we have developed a system for affective states computation. Mental state inference is an active and challenging research topic in computer vision, impacting important applications such as human computer interaction. Future humanmachine interaction interfaces will need a perfect understanding of the person's behavior so that machines can learn from it and react accordingly. The research work in this chapter is aiming at achieving this vision by integrating the expression analysis module to an existing system for true human computer interface to enhance productivity and interactivity. The developed system is currently integrated into an advanced learning environment for natural human-computer interface. The users of the system we have developed in this chapter report real time performance with 90 percent accuracy.

The automatic recognition of human emotional and cognitive state is an extremely challenging problem and is largely unexplored. Bayesian networks are growing in popularity as the model of choice of many computer vision researchers for problems involving reasoning under uncertainty. The proposed future research work is to introduce a dynamic Bayesian network that can learn and infer a user human's emotional and cognitive state based on:

1. User state history

A state table will be updated and maintained to identity user characteristic.

2. Current facial expression

Facial expression analysis from our research work will be used to identify the user expression.

REFERENCES

- [1] H. S. Seung and D. D. Lee, "The manifold ways of perception," *Science*, vol. 290, pp.2268-2269, 2000.
- [2] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319-2323, 2000.
- [3] D. Harter and R. Kozma, "Nonconvergent dynamics and cognitive systems," *Cognitive Science*, 2003.
- [4] R. Kozma and W. J. Freeman, "Chaotic resonance: Methods and applications for robust classification of noisy and variable patterns," *International Journal of Bifurcation and Chaos*, vol.10, pp. 2307-2322, 2001.
- [5] S. Edelman and N. Intrator, "Learning as formation of low dimensional representation spaces," Proceedings of the 19th The Cognitive Science Meeting, J. Elman (ed.), pp. 199-204, 1997.
- [6] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. National Acad. Sci.*, vol. 79, pp. 2554-2558, 1982.
- [7] H. S. Seung, "Learning continuous attractors in recurrent networks," Advances in Neural Information Processing Systems, vol. 10, pp.654-60, 1998.
- [8] M. J. Seow and V. K. Asari, "High storage capacity architecture for pattern recognition using an array of Hopfield neural networks," *IEEE Computer Society Proceedings of 30th International Workshop on Applied Imagery and Pattern Recognition*, pp. 169-174, 2001.

- [9] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, "The capacity of the Hopfield associative memory," *IEEE Trans on Information Theory*, vol. 33 pp. 461-482, 1987.
- [10] M. K. Muezzinoglu, C. Guzelis, and J. M. Zurada, "A new design method for complex-valued multistate Hopfield associative memory," *IEEE Trans on Neural Networks*, vol. 14, no. 4, pp. 891- 899, 2003.
- [11] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol. 3, pp. 71-86, 1991.
- [12] H. Murase and S. K. Nayar, "Visual learning and recognition of 3D objects from appearance," *International Journal of Computer Vision*, vol. 14, pp.5-24, 1995.
- [13] D. E. Rumelhart and J. L. McClelland, "Parallel distributed processing: Explorations in the microstructure of cognition," vol. 1: Foundations, *MIT Press*, 1986.
- [14] Gori and Teai, "On the problem of local minima in backpropagation," *IEEE Trans* on Pattern Analysis and Machine Intelligence, vol.14, no.1, pp.76-86, 1992.
- [15] V. Vapnik, "The nature of statistical learning theory," Springer-Verlag, 1995.
- P. Bartlettand and J. S. Taylor, "Generalization performance of support vector machines and other pattern classifiers," in B. Schoelkopf, C. J. C. Burgfes, and A. J. Smola, editors, *Advances in Mernel Methods Support Vector Learning*, MIT Press, pp. 43-54, 1999.
- [17] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *IEEE Trans on Information Theory*, vol. 44, no. 2, pp. 525-536, 1998.

- [18] V. Blanz, B. Schoelkopf, H. Buelthoff, C. Burges, V. Vapnik, and T. Vetter, "Comparison of view-based object recognition algorithms using realistic 3D models," in C. von der Malsburg, W. von Seelen, J. C. Vorbrueggen, and B. Sendhoff, editors, *Artificial Neural Networks -- ICANN'96*, *Springer Lecture Notes in Computer Science*, vol. 1112, pp. 251-256, 1996.
- [19] T. Kohonen, "Self-organizing maps," Springer Series in Information Sciences, vol. 30, 1997.
- [20] M.Y. Kiang, U.R. Kulkarni, M. Goul, R.T. Chi, E. Turban, and A. Philippakis, "Improving the effectiveness of self-organizing map networks using a circular Kohonen layer," *International Conference on System Sciences*, vol. 5, pp. 521-526, 1997.
- [21] S. Grossberg, "Adaptive pattern classification and universal recoding: Parallel development and coding of neural feature detectors," *Biological Cybernetics*, vol. 23, pp. 121-134, 1976.
- [22] G. A. Carpenter and S. Grossberg, "ART2: Stable self-organization of category recognition codes for analog input patterns," *Applied Optics*, vol. 26, pp. 4919-4930, 1987.
- [23] G. A. Carpenter, S. Grossberg and J.H. Reynolds, "ARTMAP: Supervised realtime learning and classification of no-stationary data by a self-organizing neural network," *Neural Networks*, vol. 4, pp. 565-588, 1991.
- [24] G. A. Carpenter, S. Grossberg, N. Markuson, J. H. Reynolds and D.B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised

learning of analog multidimensional maps," *IEEE Trans on Neural Networks*, vol.3, no. 5, pp. 698-713, 1992.

- [25] C. A. Hung and S. F. Lin, "Adaptive hamming net: A fast-learning ART 1 model without searching," *Neural Networks*, vol. 8, no.4, pp. 605-618, 1995.
- [26] J. R. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps," *Neural networks*, vol.9, no. 5, pp. 881-897, 1996.
- [27] T. Kasuba, "Simplified Fuzzy ARTMAP," *AI Expert*, pp. 18-25, 1993.
- [28] W. J. Freeman, J. M. Barrie, "Analysis of spatial patterns of phase in neocortical gamma EEGs in rabbit," *Journal of Neurophysiology*, vol.84, pp.1266-1278, 2000.
- [29] Y. Yao and W. J. Freeman, "Model of biological pattern recognition with spatially chaotic dynamics," *Neural Networks*, vol. 3, pp. 153-170, 1990.
- [30] R. Kozma and W. J. Freeman, "Chaotic resonance: Methods and applications for robust classification of noisy and variable patterns," *International Journal of Bifurcation and Chaos*, vol. 10, pp. 2307-2322, 2001.
- [31] W. J. Freeman, R. Kozma, and P. Werbos, "Biocomplexity: Adaptive behavior in complex stochastic dynamical systems," *BioSystems*, no. 59, pp. 109-123, 2001.
- [32] W.J. Freeman, "The physiology of perception," *Scientific American*, vol. 264, pp.78-85, 1991.
- [33] W. J. Freeman, "Perception of time and causation through the kinesthesia of intentional action," *Cognitive Processing*, vol. 1, pp. 1, pp. 18-34, 2000.

- [34] D.E Rumelhart, G.E. Hinton, R.J. Williams, "Learning internal representations by error propagation," D. Rumelhart and J. McClelland, editors. *Parallel Data Processing, vol. 1*, M.I.T. Press, Cambridge, MA. pp. 318-362, 1986.
- [35] J.J. Hopfield and D.W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.
- [36] T. Kohonen, "Automatic formation of topological maps of patterns in a selforganizing system," Proceedings of 2SCIA: Second Conference on Image Analysis, pp. 214-220, 1981.
- [37] M.J. Seow and V.K. Asari, "A modular architecture for a recurrent neural network for character recognition," *International Conference on Artificial Neural Networks In Engineering (ANNIE 2002): Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Artificial Life and Data Mining, Rolla, Missouri*, vol. 12, pp. 21-26, 2002.
- [38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
- [39] J.H. Byrne, "Cellular analysis of associative learning," *Physiological Review*, vol. 67, pp. 329-439, 1987.
- [40] Y. Bengio, S. Bengio, and J. Cloutier, "Learning synaptic learning rules," In *Proceedings of the International Conference on Neural Networks for Computing*, pp. 969–974, 1991.

- [41] S. Becker and C.Y. Le, "Improving the convergence of back-propagation learning with second-order methods," *Proceeding of the 1988 Connectionist Summer School,* San Mateo, Morgan Kaufmann, pp. 29-37, 1989.
- [42] C.L. Scofield, and L.N. Cooper, "Development and properties of neural network," *Contemporary Physics*, vol. 26, pp. 125-145, 1985.
- [43] J.A. Anderson, J.W. Silverstein, S.A. Ritz and R.S. Jones, "Distinctive features, categorized perception, and probability learning: some application of a neural model," *Psychology Review*, vol. 84, pp.413-451, 1977.
- [44] J.A. Anderson and M. Mozer, "Categorization and selective Neurons," In G.E.
 Hilton, & J. A Anderson, *Parallel Models of Associative Memory*. Hillsdale, NJ:
 Erlbaum, pp. 213-236, 1981.
- [45] J.A. Anderson and G.L. Murphy, "Psychological Concept in a parallel System," In *Evolution, Games, and Learning*, New York: North Holland, 1986.
- [46] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy* of Sciences, USA, vol. 81, pp. 3088-3099, 1984.
- [47] D.O. Gorodnichy and A.M. Reznik, "Increasing attraction of pseudo-inverse autoassociative networks," *Neural Processing Letters*, vol. 5, no. 2, pp. 123-127, 1997.
- [48] N. Davey, R.G. Adams, S.P. Hunt, "High performance associative memory models and symmetric connections," *Proceedings of the International ICSC Congress on Intelligent Systems and Applications (ISA 2000): Symposium on Computational Intelligence (CI 2000)*, vol. 2, pp 326-331, 2000.

186

- [49] L.N. Cooper, "A possible organization of animal memory and learning," In Proceeding of Nobel Symposium on Collective Properties of Physical System (B. Lundquist and S. Lundquist, ed.), pp. 252-264, 1985.
- [50] S. Haykin, "Neural Network: A Comprehensive Foundation," Macmillan, New York, 1994.
- [51] D.O. Hebb, "The Organization of Behaviour," Wiley, New York, 1949.
- [52] R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern Classification," Wiley, 2000.
- [53] M.J. Seow and K.V. Asari, "Associative Memory Using Ratio Rule For Multi-Valued Pattern Association," *Proc. IEEE Int. Conf. Neural Networks*, vol. 4, pp. 2518-2522, 2003.
- [54] C.D. Brody and R.A. Kepecs, "Basic Mechanisms for Graded Persistent Activity: Discrete Attractors, Continuous Attractors, and Dynamic Representations," *Curr. Opin. Neurobiol.*, vol. 13, pp. 204-211, 2003.
- [55] S.M. Stringer, T.P. Trappenberg, E.T. Rolls, and I.E.T. de Araujo, "Self-Organizing Continuous Attractor Networks And Path Integration: One-Dimensional Models Of Head Direction Cells," *Network: Comput. Neural Syst.*, vol. 13, pp. 217-242, 2002.
- [56] H.S. Seung, "Continuous Attractors and Oculomotor Control," *Neural Networks*, vol. 11, pp. 1253-1258, 1998.
- [57] M.J. Seow and K.V. Asari, "Recurrent Network as a Nonlinear Line Attractor for Skin Color Association," Advances in Neural Networks - ISNN 2004, Springer-Verlag, LNCS 3174, pp. 870-875, 2004.

- [58] Z. Yi and K. K. Tan, "Multistability analysis of discrete recurrent neural networks with unsaturating piecewise linear transfer functions," *IEEE Trans. Neural Network*, vol. 15, no. 2, pp. 329 336, 2004.
- [59] S. Jankowski, A. Lozowski, and J.M. Zurada, "Complex-valued multi-state neural network associative memory," *IEEE Trans. Neural Network*, vol. 7, pp. 1491-1496, 1996.
- [60] G. Perisi, "Asymmetric neural networks and the process of learning," J. Phys. A19, L675, 1986.
- [61] D.J. Jobson, Z. Rahman, G.A. Woodell, "A multi-scale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 965-976, 1997.
- [62] D.J. Jobson, Z. Rahman, G.A. Woodell, "The statistics of visual representation," *Proc. SPIE* 4736, pp. 17-24, 2002.
- [63] M.J. Seow, D. Valaparla, V.K. Asari, "Neural network based skin color model for face detection," Proc. IEEE Int. Workshop on Applied Imagery and Pattern Recognition, pp. 141-145, 2003.
- [64] R.C. Gonzalez, "Digital Image Processing," Prentice Hall, 2002.
- [65] V. Caselles, J.L. Lisani, J.M. Morel, G. Sapiro, "Shape preserving local histogram modification," *IEEE Trans. Image Processing*, vol. 8, no. 2, pp. 220-230, 1999.
- [66] O.D. Faugeras, "Digital color image processing within the framework of a human visual model," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 27 no. 4, pp. 380-393, 1979.
- [67] T.G. Stockham, Jr., "Image processing in the context of a visual model," *Proc. of the IEEE*, vol. 60, no. 7, pp. 828-842, 1972.

- [68] M.J. Seow, K.V. Asari, "Homomorphic processing system and Ratio rule for color image enhancement," Proc. of the IEEE Int. Joint Conference on Neural Networks, no. 4, pp. 2507-2511, 2004.
- [69] P. Buser, M. Imbert, "Vision," MIT Press, Cambridge, MA, 1992.
- [70] H.R. Blackwell, "Contrast Thresholds of the Human Eye," *Journal of the Optical Society of America*, vol. 36, pp. 624–643, 1946.
- [71] D.J. Jobson, Z. Rahman, and G.A. Woodell, "Properties and Performance of a Center/Surround Retinex," *IEEE Trans on Image Processing*, vol 6, pp. 451-462, 1997.
- [72] M.J. Seow and K.V. Asari, "Ratio Rule and Homomorphic Filter for Enhancement of Digital Color Image," *Journal of Neurocomputing*, vol. 69, no. 7-9, pp. 954-958, 2006.
- [73] E. H. Land and J. McCann, "Lightness and Retinex theory," *Journal of the Optical Society of America*, vol. 61, pp. 1-11, 1971.
- [74] J. Jobson, Z. Rahman, and G.A. Woodell, "Retinex image processing: improved fidelity to direct visual observation," in Proc. of the IS&T/SID 4th Colour Imaging Conference: Colour Science, Systems and Applications, pp. 124-126, 1996.
- [75] Z. Rahman, J. Jobson, and G.A. Woodell, "A multiscale Retinex for colour rendition and dynamic range compression," in *Proc. of SPIE International Symposium on Optical Science, Engineering and Instrumentation, Applications of Digital Image Processing XIX*, SPIE 2825, pp. 183-191, 1996.

- [76] J. Jobson and G.A. Woodell, "Properties and performance of a center/surround Retinex: part 2. surround design," *NASA Technical Memorandum* 110199, 1995.
- [77] K. Barnard and B. Funt, "Investigations into multi-scale retinex," in *Colour Imaging: Vision*, pp. 9-17, John Wiley and Sons, 1999.
- [78] B. Funt, F. Ciurea, and J. McCann, "Retinex in Matlab," in Proc. of the IS&T/SID 8th Color Imaging Conference: Color Science, Systems and Applications, pp. 112-121, 2000.
- [79] D. Schilling and P. C. Cosman, "Image quality evaluation based on recognition times for fast image browsing applications," *IEEE Trans. Multimedia*, vol. 4, pp. 320–331, 2002.
- [80] V. Walsh, and J. Kulikowski, Perceptual Constancy: Why Things Look as They Do, Cambridge University Press, 1998.
- [81] C. RosenBerg, T. Minka, A. Ladsariya, "Bayesian Color Constancy with Non-Gaussian Model", *Proceeding of Neural Information Processing System*, 2003.
- [82] M.J. Seow and V.K. Asari. "Learning a perceptual manifold," Proceedings of the Ninth International Conference on Cognitive and Neural Systems – ICCNS, pp. 18, 2005.
- [83] Y. Chang, C. Hu, and M. Turk, "Manifold of Facial Expression", Proceeding IEEE International Workshop on Analysis and Modeling of Faces and Gestures, pp. 28-35, 2003.
- [84] G. Buchsbaum, "A spatial processor model for object color perception," *Journal of Franklin Institute*, pp. 1-26, 1980.

- [85] D. H. Brainard and B. A. Wandell, "Analysis of the Retinex Theory of Color Vision", Journal of the Optical Society of America A, pp. 1651-1661, 1986.
- [86] E. H. Land, "The Retinex", American Scientist 52, pp. 247-264, 1964,
- [87] B. Funt, V. Cardei, and K. Barnard, "Learning color constancy," Proceedings of the Fourth IS&T/SID Color Imaging Conference, pp.58-60, 1996.
- [88] V.C. Smith and J. Pokorny, "Spectral sensitivity of the foveal cone photopigments between 400 and 700 nm", *Vision Research*, pp. 161-171, 1975.
- [89] E.H. Land, "Recent advances in Retinex theory," Vision Research, pp. 7-21, 1986.
- [90] J.D. Brand, and J.S.D. Mason, "A skin probability map and its use in face detection," *Proceeding of International Conference in Image Processing*, pp. 1034-1037, 2001.
- [91] Z. Rahman, D.J. Jobson, and G.A. Woodell, "Retinex processing for automatic image enhancement," *Proceedings of SPIE, Human Vision and Electronic Imaging VII*, 4662, pp. 390-401, 2002.
- [92] D. Beymer, "Face recognition under varying poses," *IEEE Computer Vision and Pattern Recognition*, pp. 756-761, 1994.
- [93] Y. Wu, K. L. Chan, L. Wang, "Face recognition based on discriminative manifold learning," *17th International Conference on Pattern Recognition*, 2004.
- [94] M. Turk, and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive NeuroScience*, pp. 71-86, 1991.

- [95] S. Edelman, N. Intrator, "Learning as extraction of low-dimensional representations," *Perceptual Learning* R. Goldstone and D. Medin and P. Schyns (Eds) pp. 353-380, 1997.
- [96] H. Murase, and S. K. Nayar, "Visual learning and recognition of 3D objects from appearance," *International Journal of Computer Vision* 14, pp.5-24, 1995.
- [97] J. J. Atick, P. A. Griffin, and N. A. Redlich, "Statistical approach to shape-fromshading: deriving 3D face surfaces from single 2D images," *Neural Computation*, 1997.
- [98] J. B. Tenenbaum, V. de Silva, and john C. Langford, "A global geometric framework for nonlinear dimensional reduction," *Science*, vol. 290, 2000.
- [99] S. Roweis and L. K. Saul, "Nonlinear dimensional reduction by locally linear embedding," *Science*, vol. 290, 2001.
- [100] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," Advances in Neural Information Processing Systems, vol. 15, pp. 585-591, 2001.
- [101] M. J. Seow and K. V. Asari, "Color characterization and balancing by a line attractor for nonlinear image enhancement," *Neural Processing Letter*, vol. 22, no. 3, pp. 291-309, 2005.
- [102] M. J. Seow and K. V. Asari, "Associative memory using nonlinear line attractor network for multi-valued pattern association," Lecture Notes in Computer Science, Published by Springer-Verlag Heidelberg, *Advances in Neural Networks*, pp. 485-490, 2005.

- [103] M. J. Seow and K. V. Asari, "Learning using distance based training algorithm for pattern recognition," *Pattern Recognition Letters*, vol. 25, no. 2, pp. 189-196, 2004.
- [104] <u>http://amp.ece.cmu.edu/projects/FaceAuthentication/</u> (Accessed 2004)
- [105] <u>http://images.ee.umist.ac.uk/danny/database.html</u> (Accessed 2004)
- [106] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [107] X. Liu, T. Chen, B.V.K. Vijaya Kumar, "Face Authentication for Multiple Subjects using Eigenflow," *Pattern Recognition: special issue on Biometric*, vol. 36, pp. 313-328, 2003.
- [108] M. Turk and M. Kölsch, "Perceptual interfaces," In Gerard Medioni and Sing Bing Kang, editors, *Emerging Topics in Computer Vision*, chapter 9. Prentice Hall, 2004.
- [109] N.O. Bernsen, "Multimodality in language and speech systems. From theory to design support tool," In: Granström, B., House, D., and Karlsson, I. (Eds.):
 *Multimodality in Language and Speech Systems*Kluwer Academic Publ., pp. 93-148, 2002
- [110] <u>http://www.cs.ucsb.edu/conferences/PUI/index.html</u> (Accessed 2006)
- [111] Andrea Corradini, Manish Mehta, Niel-Ole Bernsen, Jean-Claude Martin, Sarkis Abrilian, "Multimodal Input Fusion in Human-Computer Interaction on the Example of the on-going NICE Project," *Proceedings of the NATO-ASI*

conference on Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management, August 18th-29th, Yerevan (Armenia), 2003

- [112] C. Darwin, "The Expression of Emotions in Man and Animals," John Murray, reprinted by University of Chicago Press, 1872.
- [113] P. Ekman, "The Argument and Evidence about Universals in Facial Expressions of Emotion," pp. 143–164. Wiley, New York, 1989.
- [114] P. Ekman and W. Friesen, "The Facial Action Coding System: A Technique for the Measurement of Facial Movement," Consulting Psychologists Press, San Francisco, 1978.
- [115] K. Scherer and P. Ekman, "Handbook of Methods in Nonverbal Behavior Research," Cambridge University Press, Cambridge, UK, 1982.
- [116] M. Suwa, N. Sugie, and K. Fujimora, "A preliminary note on pattern recognition of human emotional expression," In *International Joint Conference on Pattern Recognition*, pp. 408–410, 1978.
- [117] Yingli Tian, T. Kanade and J. F. Cohn, "Recognizing Action Units for Facial Expression Analysis," *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 2, 2001.
- [118] Paul A. Viola, Michael J. Jones, "Robust Real-Time Face Detection," International Journal of Computer Vision, vol. 57 no.2, pp. 137-154, 2004.
- [119] Yoav Freund and Robert E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55 no. 1 pp. 119–139, 1997.

- [120] T. F. Cootes, G. Edwards, and C. J. Taylor, "A comparative evaluation of active appearance model algorithms," *In BMVC 98. Proc.of the Ninth British Machine Vision Conf.*, vol. 2, pp. 680-689. Univ. Southampton, 1998.
- [121] Rosalind W. Picard, "Affective Computing," MIT Press. 1997.
- [122] Vicki Bruce, "Recognizing Faces," Hove, East Sussex: Lawrence Erlbaum, 1986.
- [123] Maja Pantic and Leon J.M. Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22 pp.1424–1445, 2000.
- [124] P. Ekman, & W.V. Friesen, "Measuring facial movement," *Environmental Psychology and Nonverbal Behavior*, vol. 1 no. 1, pp. 56-75, 1976.
- [125] R. El Kaliouby and P. Robinson, "Real-Time Inference of Complex Mental States from Facial Expressions and Head Gestures," In the IEEE International Workshop on Real Time Computer Vision for Human Computer Interaction at CVPR, pp. 154, 2004.
- [126] Simon Baron-Cohen, Ofer Golan, Sally Wheelwright, and Jacqueline J. Hill,"Mind Reading: The Interactive Guide to Emotions," London: Jessica KingsleyPublishers, 2004.
- [127] Simon Baron-Cohen, Ofer Golan, Sally Wheelwright, and Jacqueline Hill, "A New Taxonomy of Human Emotions," (under review), 2004.
- [128] Rana Ayman el Kaliouby, "Mind-reading machines: automated inference of complex mental states," Technical Report UCAM-CL-TR-636, 2005.
- [129] FaceTracker. Facial Feature Tracking SDK. Neven Vision, 2002.

CURRICULUM VITAE

Ming-Jung Seow

DOB: March 10, 1979. PhD Candidate. Department of Electrical and Computer Engineering. Old Dominion University. Email: mseow@odu.edu Homepage: http://www.lions.odu.edu/~mseow

EDUCATION

- Master of Science in Computer Engineering. Old Dominion University, Norfolk VA, USA. December 2002.
- Bachelor of Science in Computer Engineering with minor in Computer Science. Old Dominion University, Norfolk VA, USA. May 2001.

RESEARCH INTERESTS

- Artificial neural networks and machine learning.
- Computer vision and pattern recognition.
- Real time image processing and analysis.
- Natural language processing.
- Complexity and artificial life.

RESEARCH PUBLICATIONS

JOURNAL PAPERS

- 1. Ming Z. Zhang, **Ming-Jung Seow**, K. Vijayan Asari, "A high performance architecture for color image enhancement using a machine learning approach," nternational Journal of Computational Intelligence Research Special Issue on Neurocomputing and Applications, vol. 2, no. 1, pp. 41-48, 2006.
- 2. Ming-Jung Seow, K. Vijayan Asari, "Ratio rule and homomorphic filter for enhancement of digital color image," Journal of Neurocomputing, vol. 69, no. 7-9, pp. 954-958, March 2006.
- 3. **Ming-Jung Seow**, K. Vijayan Asari, "Recurrent neural network as a linear attractor for pattern association," IEEE Transactions on Neural Networks, vol. 17, no. 1, pp. 246-250, January 2006.
- 4. **Ming-Jung Seow**, K. Vijayan Asari, "Color characterization and balancing by a nonlinear line attractor network for image enhancement," Neural Processing Letters, vol. 22, no 3, pp. 291-307, December 2005.
- 5. Deepthi P. Valaparla, Li Tao, Ming-Jung Seow, K. Vijayan Asari, "Neural network model for image enhancement using histogram matching," Multimedia

Cyberscape Journal: Special Issue on Multimedia Data Processing and Compression, vol. 3, no. 4, pp. 12-18, November 2005.

- 6. **Ming-Jung Seow**, K. Vijayan Asari, "Ratio rule for recurrent neural network based associative recall of multilevel patterns," Journal of Neural, Parallel and Scientific Computations, vol. 12, pp. 545-566, December 2004.
- 7. Ming-Jung Seow, K. Vijayan Asari, "Learning using distance based training algorithm for pattern recognition," Pattern Recognition Letters, vol. 25, no. 2, pp. 189-196, January 2004.
- 8. **Ming-Jung Seow**, Hau T. Ngo, K. Vijayan Asari, "Systolic implementation of 2-D block based Hopfield neural network for efficient pattern association," Journal of Microprocessors and Microsystems, vol. 27, no. 8, pp. 359-366, August 2003.
- 9. **Ming-Jung Seow**, K. Vijayan Asari, "Modular architecture for Hopfield network and distance based training algorithm for pattern association," Journal of Neural, Parallel and Scientific Computations, vol. 10, no. 4, pp. 371-386, December 2002.

BOOK CHAPTERS

LECTURE NOTES IN COMPUTER SCIENCE (SPRINGER-VERLAG):

- 1. **Ming-Jung Seow,** K. Vijayan Asari, "Robust learning by self-organization of nonlinear lines of attractions," Lecture Notes in Computer Science, Published by Springer-Verlag Heidelberg, Advances in Neural Networks, 2006.
- Ming-Jung Seow, K. Vijayan Asari, "Associative memory using nonlinear line attractor network for multi-valued pattern association," Lecture Notes in Computer Science, Published by Springer-Verlag Heidelberg, Advances in Neural Networks, Edited by Jun Wang, Xiaofeng Liao, and Zhang Yi 3496, pp. 485-490, 2005.
- 3. **Ming-Jung Seow**, K. Vijayan Asari, "Recurrent network as a nonlinear line attractor for skin color association," Lecture Notes in Computer Science, Published by Springer-Verlag Heidelberg, Advances in Neural Networks, Edited by Fuliang Yin, Jun Wang and Chengan Guo, vol. 3173, pp. 870-875, 2004.

INTELLIGENT ENGINEERING THROUGH ARTIFICIAL NEURAL NETWORK (AMERICAN SOCIETY OF MECHANICAL ENGINEERS PRESS)

- Ming-Jung Seow, K. Vijayan Asari, "Nonlinear line attractor network for face expression association," Intelligent Engineering Systems Through Artificial Neural Networks, ASME Press Series (ISBN: 0-7918-0204-3), Proceedings of International Conference on Artificial Neural Networks In Engineering (ANNIE 2004): Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, and Artificial Life, Edited by C.H. Dagli, A.L. Buczak, M. J. Embrechts, and O. Ersoy, vol. 15, pp. 553-558, November, 2005.
- 2. Ming-Jung Seow, K. Vijayan Asari, "Real-time face detection in video sequences in complex background environment," Intelligent Engineering Systems

Through Artificial Neural Networks, ASME Press Series (ISBN: 0-7918-0228-0), Proceedings of International Conference on Artificial Neural Networks In Engineering (ANNIE 2004): Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Complex Systems and Artificial Life, Edited by C.H. Dagli, A.L. Buczak, D.L. Enke, M.J. Embrechts and O. Ersoy, vol. 14, pp. 573-578, 07-10, 2004.

- Ming-Jung Seow, Vijayan K. Asari, "A statistical neural network model for object detection," Intelligent Engineering Systems Through Artificial Neural Networks, ASME Press Series (ISBN: 0-7918-0228-0), Proceedings of International Conference on Artificial Neural Networks In Engineering (ANNIE 2003): Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Artificial Life and Data Mining, Edited by C.H. Dagli, A.L. Buczak, D.L. Enke, M.J. Embrechts and O. Ersoy, vol. 13, pp. 631-636, 2003.
- 4. Ming-Jung Seow, Vijayan K. Asari, "A modular architecture for a recurrent neural network for character recognition," Intelligent Engineering Systems Through Artificial Neural Networks, ASME Press Series (ISBN: 0-7918-0191-8), Proceedings of International Conference on Artificial Neural Networks In Engineering (ANNIE 2002): Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Artificial Life and Data Mining, Edited by C.H. Dagli, A.L. Buczak, J. Ghosh, M.J. Embrechts, O. Ersoy, and S.W. Kercel, USA, vol. 12, pp. 21-26, 2002.

REFEREED CONFERENCE PROCEEDINGS

- 1. **Ming-Jung Seow** and K. Vijayan Asari, "On the divergence dynamics of the nonlinear line of attraction," International Joint Conference on Neural Networks IJCNN 2006, Vancouver, BC, Canada, July 16-21, 2006 (accepted).
- 2. Ming Z. Zhang, **Ming-Jung Seow** and K. Vijayan Asari, "A hardware architecture for color image enhancement using a machine learning approach with adaptive parameterization," International Joint Conference on Neural Networks IJCNN 2006, Vancouver, BC, Canada, July 16-21, 2006 (accepted).
- 3. **Ming-Jung Seow,** Ming Zhang, K. Vijayan Asari, "Natural color representation using Ratio learning algorithm for enhancement of digital color images," 30th International Congress of Imaging Science ICIS'06, Rochester, New York, May 7-11, 2006 (accepted).
- 4. **Ming-Jung Seow**, K. Vijayan Asari, "Manifold of color perception: color constancy using a nonlinear line attractor," IS&T/SPIE Symposium on Electronic Imaging: Applications of Neural Networks and Machine Learning in Image Processing X, San Jose, CA, January 15-19, 2006.
- 5. Li Tao, Ming-Jung Seow, K. Vijayan Asari, "Nonlinear color image enhancement to improve face detection in complex lighting environment,"

IS&T/SPIE Symposium on Electronic Imaging: Algorithms and Systems V, San Jose, CA, January 15-19, 2006.

- Ming-Jung Seow, Richard Cortland Tompkins, K. Vijayan Asari, "A New Nonlinear Dimensionality Reduction Technique for Pose and Lighting Invariant Face Recognition," 1st IEEE Workshop on Face Recognition Grand Challenge Experiments FRGC 2005 in conjunction with the IEEE Conference on Computer Vision and Pattern Recognition CVPR 2005, San Diego, CA, June 20 25, 2005.
- Ming-Jung Seow, K. Vijayan Asari, "On using an associative memory for improving digital color images: Color characterization, enhancement, and color balancing," IEEE International Joint Conference on Neural Networks – IJCNN 2005, Montreal, Quebec, Canada, July 31-August 04, 2005.
- 8. Ming-Jung Seow, K. Vijayan Asari, "Learning a perceptual manifold," Ninth International Conference on Cognitive and Neural Systems ICCNS 2005, Boston, Massachusetts, May 18 21, p. 8, 2005.
- Ming-Jung Seow, Deepthi Valaparla, K. Vijayan Asari, "Analysis of X-ray images of airport baggage for explosive detection," SensorsGov Expo and Conference – SensorsGov-2004, Virginia Beach, VA, USA, September 13 - 15, 2004.
- Ming-Jung Seow, K. Vijayan Asari, "Homomorphic processing system and Ratio Rule for color image enhancement," Proceedings of the IEEE International Joint Conference on Neural Networks – IJCNN 2004, Budapest, Hungary, vol. 4, pp. 2507-2511, (ISSN: 1098-7576) July 25-29, 2004.
- Ming-Jung Seow, K. Vijayan Asari, "Neural network based human visual system modeling for visibility improvement in digital images," Eighth International Conference on Cognitive and Neural Systems – ICCNS 2004, Boston, Massachusetts, May 19 - 22, 2004.
- 12. Ming-Jung. Seow, Rajkiran Gottumukkal, Deepthi Valaparla, and K. Vijayan Asari, "A robust face recognition system for real time surveillance," Proceedings of the IEEE Computer Society International Conference on Information Technology: Coding and Computing ITCC 2004, Las Vegas, Nevada, vol. 1, pp. 631-636, April 5-7, 2004.
- Ming-Jung Seow, Deepthi Valaparla, Vijayan K. Asari, "Neural network based skin color model for face detection," IEEE International Workshop on Applied Imagery and Pattern Recognition, AIPR - 2003, Washington DC, USA, pp. 141-145, October 15 - 17, 2003.
- Ming-Jung Seow, Vijayan K. Asari, "Associative memory based on ratio learning for real time skin region extraction," IEEE International Workshop on Applied Imagery and Pattern Recognition, AIPR - 2003, Washington DC, USA, pp. 151-156, October 15 - 17, 2003.
- 15. Ming-Jung Seow, Deepthi Valaparla, K. Vijayan Asari, "L2-Norm approximation based learning in recurrent neural networks for expression
invariant face recognition," 2003 IEEE International Conference on Systems, Man & Cybernetics – SMC 2003, Washington, D.C., USA, pp. 3512-3517, October 5–8, 2003.

- Ming-Jung Seow, K. Vijayan Asari, "High performance associative memory with distance based training algorithm for character recognition," Proceedings of the IEEE International Joint Conference on Neural Networks – IJCNN 2003, Portland, Oregon, USA, pp. 1185-1188, July 20-24, 2003.
- Ming-Jung Seow, K. Vijayan Asari, "Associative memory using ratio rule for multi-valued pattern association," Proceedings of the IEEE International Joint Conference on Neural Networks – IJCNN 2003, Portland, Oregon, USA, pp. 2518-2522, July 20-24, 2003.
- Ming-Jung Seow, Hau T. Ngo, K. Vijayan Asari, "Systolic array implementation of block based Hopfield neural network for pattern association," Proceedings of the IEEE Computer Society Annual Symposium on VLSI - ISVLSI 2003, Tampa, Florida, USA, pp. 213-214, February 20-21, 2003.
- 19. Ming-Jung Seow, Vijayan K. Asari, "A generalized cellular neural network architecture for high storage capacity pattern recognition," Proceedings of Second International Workshop on Intelligent Systems Design and Applications (ISDA 2002), Atlanta, USA, pp. 63-68, August 07-08, 2002.
- 20. Ming-Jung Seow, Vijayan K. Asari, "A parallel VLSI architecture for real-time segmentation of images with complex background environment," Proceedings of 10th NASA Symposium on VLSI Design, Albuquerque, New Mexico, USA, pp. 8.2.1-8.2.7, March 20-21, 2002.
- Ming-Jung Seow, Vijayan K. Asari, "High storage capacity architecture for pattern recognition using an array of Hopfield neural networks," IEEE Computer Society Proceedings of 30th International Workshop on Applied Imagery and Pattern Recognition, AIPR - 2001, Washington DC, USA, pp. 169 - 174, October 10 - 12, 2001.

RESEARCH PROJECTS

1. SmartRoom – Natural Interaction (2006) National Aeronautics and Space Administration (NASA)

Involved in the SmartRoom project which allows users to collaborate in a shared space. The system "hears" users' voice commands and "sees" user's gestures Interactions are natural, more like human-to-human interactions.

2. Automated Mind-Reading for Intelligent Learning Hub (2005 - current) National Aeronautics and Space Administration (NASA)

Designed human-computer interaction (HCI) through facial expression analysis in video. For demonstration, visit project homepage: [http://www.lions.odu.edu/org/vlsi/demo/faceExpression.htm] 3. **Participated in the Face Recognition Grand Challenge (FRGC)** (2004 - 2005) National Institute of Standards and Technology

Participated in the FRGC V2 experiments. The results are published in IEEE International Conference on Computer Vision Pattern Recognition (CVPR) workshop.

4. **Pose and Lighting Invariant Face Detection Algorithm** (2004 - 2005) US Department of Defense: Technology Support Working Group

Developed a real-time facial recognition system that is invariant to expression, pose, and lighting variation. For demonstration, visit project homepage: Homepage: [http://www.lions.odu.edu/org/vlsi/demo/faceDetection.htm] Homepage: [http://www.lions.odu.edu/org/vlsi/demo/faceRecognition.htm] Homepage: [http://www.lions.odu.edu/org/vlsi/demo/skinSegmentation.htm]

5. Visibility Improvement in Color Images (2003 - 2004)

Vision Innovations Corporation and Virginia's Center for Innovative Technology (CIT)

Developed a real-time adaptive image enhancement algorithm for visibility improvement in digital color images (and video). The system is capable of processing a 320×240 resolution video stream at 30 fps using a Pentium IV 3GHz machine with 2GB memory. For demonstration, visit project homepage: Homepage:

[http://www.lions.odu.edu/org/vlsi/demo/imageEnhancement.htm] (Second video)

6. Explosive Detection from High Resolution X-ray Images (2003)

Through Homeland Security Research Group at ODU.

Involved in developing an automated explosives and contraband detection system. The system uses technology derived from elastic X-ray scatter approach and computer vision methodology to locate and characterized threatening object according to their unique diffraction profiles.

Obtained the Certificate of Completion from the Old Dominion University Environment Health and Safety Office for successfully completing the requirements for the course titled "Radiation Protection for the Analytical X-ray User", October 11, 2004.

 Self-organizing Nonlinear Line Attractor Network for Pattern Association, Classification, and Prediction (2003-2005)
US Department of Education
PhD GAANN Fellow

GAANN fellowships provide grant support to academic department and programs of institutions of higher education to support outstanding graduate students in key disciplines that will help to advance the United States' research enterprise. Recurrent Neural Networks and Algorithms for Reconstruction of Images from Noisy and/or Partial Data (2001-2002) US Department of Education MS GAANN Fellow

GAANN fellowships provide grant support to academic department and programs of institutions of higher education to support outstanding graduate students in key disciplines that will help to advance the United States' research enterprise.

PATENT FILED

1. Color Image Characterization, Enhancement and Balancing. Technology licensed to Vision Innovations Corporation

AWARDS AND HONORS

- Citation in "Who's Who in America," Marquis (eds.), 61th edition, 2007-2008.
- Citation in "Who's Who in Science and Engineering," Marquis (eds.), 9th edition, 2006-2007.
- Citation in "Who's Who in America," Marquis (eds.), 60th edition, 2006-2007.
- Outstanding Ph.D Research Award (2005), Old Dominion University Electrical and Computer Engineering Department.
- Nominated for Best Paper Award in ANNIE 2004 conference.
- Invited Session Chair for International Conference on Artificial Neural Networks In Engineering (ANNIE 2004)
- Best Research Project Award in Computational Biology, CS 895, Old Dominion University 2002.
- GAANN Fellow, PhD Research.
- GAANN Fellow, Masters Research.
- Summer Fellowship, Old Dominion University, Department of Electrical and Computer Engineering, 2001 (provided by department graduate director)
- Cum Laude, Old Dominion University 2001.
- The National Deans List.
- Dean's List, Old Dominion University.
- Member of Golden Key National Honor Society.
- Member of Eta Kappa Nu National Electrical Engineering Honor Society.

TEACHING EXPERIENCE

• Summer 2005, Teaching Assistant, ECE 458/558, Instrumentation (LabVIEW).

- Spring 2005, Teaching Assistant, ECE 458/558, Instrumentation (*LabVIEW*).
- Fall 2004, Teaching Assistant, ECE 481/581, Digital Signal Processing.
- Fall 2004, Project Mentor, ECE 486, "Homomorphic Processing System and Ratio Rule for Color Image Enhancement," Electrical Engineering Senior Design II.
- Spring 2004, Teaching Assistant, ECE 783/883, Digital Image Processing.
- Spring 2004, Tutor, ECE 783/883, "Tutorial on how to use MATLAB toolboxes," Digital Image Processing.
- Spring 2004, Teaching Assistant, ECE 489, "Develop, design, build and test a compression algorithm for 8-bit bitmap images based on Run Length Encoding (RLE) technique with microcontroller communication gateway and graphical user interface (GUI)," Computer Engineering Senior Design II.
- Fall 2003, Project Mentor, ECE 486, "Implementation of image enhancement algorithm based on Retinex (Retina + Cortex) image processing," Electrical Engineering Senior Design II.
- Spring 2003, Project Mentor, ECE 486, "Implementation of face recognition system using EigenFaces approach," Electrical Engineering Senior Design II.
- Fall 2002, Teaching Assistant, ECE 488, "Implementation of Adaptive Resonance Theory (ART) self organization map," Computer Engineering Senior Design I.
- Spring 2002, Project Mentor, ECE 486, "Implementation of face recognition System using Eigen Faces approach," Electrical Engineering Senior Design II.
- Fall 2001, Teaching Assistant, ECE 488, "Implementation of binary Hopfield neural network with an EPLD based coprocessor," Computer Engineering Senior Design I.

ECE 741	Formal Method in Computer System Design	A-
ECE 783	Digital Image Processing	A
ECE 881	Statistical Pattern Recognition	A
ECE 882	Digital Signal Processing II	A
CS 895	Computational Biology	A
ECE 845	Fault Tolerant Computing	A
ECE 895	Artificial Neural System	A
ECE 895	Advance Image Processing	A

COURSES (GPA: 3.96):

PROFESSIONAL MEMBERSHIP

- Student Member of Institute of Electrical and Electronics Engineers (IEEE).
- Student Member of IEEE Computational Intelligence Society.
- Student Member of IEEE Neural Networks Society.