University of Alberta

Variable Resolution Vision: Biologically Motivated Foveal Compression and Prioritization

by

Kevin James Wiebe ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Department of Computing Science

Edmonton, Alberta
Fall 1996

ISBN 0-612-18132-4

University of Alberta

Library Release Form

Name of Author: Kevin James Wiebe

Title of Thesis: Variable Resolution Vision: Biologically Motivated Foveal Compression and Prioritization

Degree: Doctor of Philosophy

Year this Degree Granted: 1996

Kevin Wiebe

Kevin James Wiebe
#1103, 10883 Saskatchewan Drive
Edmonton, Alberta
Canada, T6E 4S6

Date: October 3 1996

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Variable Resolution Vision: Biologically Motivated Foveal Compression and Prioritization** submitted by Kevin James Wiebe in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

. . . .
Anup Basu

. . . .
Ze-Nian Li

. . . .
Werner Joerg

. . . .
Xiaobo Li

. . . . J. Harms . . . . .
Janelle Harms

Date: September 16, 1996

To my Designer
and
to Theresa, my inspiration.

Abstract

Computer Vision shares its interest in investigating animate behaviour and biological processes with other disciplines within the field of artificial intelligence (AI). Our research focuses on two aspects of biological vision and applies the knowledge gleaned from nature to appropriate computer vision situations. Both *variable resolution image compression* and *spatially variant image data prioritization* are present within animate visual systems and these concepts can be effectively transferred to enhance several computer applications.

In the area of digital image compression, new techniques are required to overcome significant storage and transmission problems in computer vision. A strong understanding of image characteristics enhances the effectiveness of compression and many other image processing operations. Traditional methods have maintained a constant resolution throughout an image. However, a survey of the various visual systems present in the animal kingdom demonstrates the potential of *Variable Resolution* (VR) compression methods. The author models several animate visual systems and outlines novel image compression techniques based on foveated vision. Interesting variations on the simple fovea are proposed, motivated by similar variations present in animate visual systems — specifically multiple, dynamic and weighted foveae, and visual streaks. Techniques for efficient modelling of fovea movement are also described.

The other topic discussed is the prioritization of image data. A fundamental drawback to increasingly popular ATM-based switching is the possibility of information loss with congestion. We demonstrate that with intelligent, fovea driven priority assignment of image data, we can reduce the negative impact of information loss over ATM networks. ATM standards allow a single bit to indicate high or low packet priority. To reduce the effect of this restriction we introduce the concept of *priority dithering.* Network multimedia multicast scenarios over heterogeneous link capacities where foveal prioritization would be of benefit are described. Included are network simulation results of this method, which demonstrate the advantages of priority dithered foveal prioritization over traditional methods.

Utilizing our knowledge of biological vision systems provide us with insights into new developments in the areas of image compression, video compression, image transmission, videophones, multimedia, teleconferencing, and telepresence. Original, substantive research is presented.

Acknowledgements

I would especially like to thank my wife, Theresa, for the patience she has shown me as I have worked towards this degree. Her willingness to move to the "bitter northern city" of Edmonton and sacrifice much in her career has not gone unnoticed. I could not have done without her constant support and encouragement. Our countless evenings of thesis related discussions not only helped clarify my ideas, but Theresa often provided substantial contributions of her own.

I would also like to thank my supervisor, Dr. Anup Basu, for the time he has spent discussing the topic with me. Several of my projects were inspired by our conversations. The concept of variable resolution sensing stemmed from work Anup had previously directed.

I am greatly indebted to Anne Nield for her careful editing and comments on the final thesis. Warren Wong and Ken Der provided many hours of fruitful discussions and debugging assistance. Ken had especially helpful input on packet prioritization in the ATM simulation. Dr. Janelle Harms also gave timely advice and provided helpful resources with respect to ATM network protocols.

Finally, I am particularly indebted to my family, especially my parents Lorenz and Vicki Wiebe. Their encouragement and support have been the primary influences in shaping and preparing me in all aspects of my life. Through them I have learned the importance of mental discipline, dedication and hard work. They have given me solid spiritual guidance and instilled in me a thirst for wisdom.

Kevin James Wiebe
September 16, 1996

Contents

List of Figures

Where there is no vision,
the people perish.
Proverbs 29:18

Chapter 1

Introduction

1.1 Motivation

Today, advances in technology are proceeding at an unprecedented rate. Computers have become commonplace in homes and offices around the globe, and are considered important tools for daily activities. One recent trend in modern computer applications is towards a greater usage of digital images. The advent of multimedia has posed many new problems for both the storage and transmission of image data. A typical colour image one half the size of a sheet of looseleaf (say, 600 square pixels), contains over 1 megabyte of information; the storage space necessary to save such images in memory would quickly become unmanageable. If transmitted over a typical 28,800 baud link, this single image would take over 37 seconds to arrive. Much research has been done to address these problems, and numerous digital image compression techniques have been proposed.

In many situations, such as videoconferencing, compression of some form is necessary. It reduces the bandwidth required for transmission, the memory required for storage, and the processing required for manipulation. However, one must be careful to take into account the purpose for the data when considering a method of compression. While some methods perform well in a specific domain, they may prove to be unsuitable for others.

Data compression techniques existed even before the days of computing science. However, initial schemes were designed mainly for the compression of text and/or programs, not for images [71]. Image compression requires new methods and ideas which may not apply to other data types. By researching the unique characteristics which certain images possess, we are able to develop compression schemes capable of exploiting them.

The transmission of these images may also occur over network scenarios where there is no guarantee of error free communication. Encoding data with a method that intelligently controls the nature of the data loss can minimize its detrimental effect. Again, such encoding schemes capitalize on known properties inherent within certain images.

Specifically, our research is based on the assumption that most scenes are not comprised of equally important areas. There is often at least one primary area of particular importance, while the remainder of the image is of less interest [113]. This primary area of interest is called the *fovea*, and more detailed information is required here; less detail is required in the areas not in the fovea, called the *periphery*.

Capitalizing on this knowledge for the purpose of image compression involves decreasing the spatial sampling rate of an image as one moves further from the location of the fovea, thereby spatially varying the resolution. Spatially prioritizing data in the same way can also assist in development of image encoding schemes suited to unreliable transmission scenarios.

In the past, *Variable Resolution* (VR) techniques have been successfully applied to tasks in many areas, such as stereo correspondence [97], two dimensional object recognition [81], the estimation of depth from motion [94], line detection [97], the evaluation of time-to-impact from optical flow [95], character thinning [59, 60], and linear motion estimation in robotic navigation and on assembly lines [97]. Its use in compressing images, image sequences[1], and in prioritization schemes however, has only recently been seriously explored [15, 16, 89].

We will demonstrate the advantageous characteristics of VR compression and encoding are that:

- VR compression methods, if designed with the use of look-up tables, require no complex calculations and therefore are extremely fast;

- VR transforms operate entirely in the spatial domain; any other compression

[1] Literature in this area may also refer to *image sequences* as *videos*. The former term will be used most frequently in this work.

scheme can be run on the compressed images for greater compression ratios;

- sampling can be adjusted to guarantee minimal compression ratios;
- similarly, sampling can be adjusted to achieve constant bandwidth requirements;
- by monitoring network loads or intelligently prioritizing data, bandwidth requirements can be varied to maximize network resource utilization.

1.2 Thesis Organization

This work[2] is concerned with the concept of the fovea, or region of interest. Methods of both image compression and data prioritization which incorporate this concept are shown. Biological vision is used as the inspiration.

Our work first considers the spatially nonuniform visual subsampling within animate vision and carries this idea into the area of image and video compression. Five common topographical isodensity features of biological retinae are of specific concern:

- **general characteristics** — single fovea, continuity, anisotropism, receptor boundary shape
- **multiple foveae** — weighted foveae
- **visual streaks**
- **optic discs**
- **multiple optic paths** — dynamic foveae

We also investigate spatially variant prioritization of image data in biological visual systems. These systems can be viewed as a network stretching from the eye's retina, along the optic nerve, into the occipital lobe at the posterior of the brain. The concepts illustrated by this natural prioritization of image sequence transmission are transferred to the domain of computer vision.

The goals of our research are to:

[2] Versions of several of the chapters in this thesis have been published in [15], [109], and [110].

- Study the differences in the visual systems of a number of dissimilar animals, from a variety of species and environments. Special attention will be given to the retina.

- Isolate unique characteristics of animate visual systems and link them, if possible, to their ecology; determine the purpose of retinal specializations.

- Develop simple, practical computational models of general structures or classes of these biological visual systems.

- Construct tools which assists in the modelling of both biological visual systems and their extensions; support the process of fitting computer visual models to their intended specific digital environments.

- Demonstrate applications which can benefit from a purposive active vision approach; match biological vision models to application purposes.

Chapter 2 outlines contributions that have been made to date, in the area of image compression. Several popular compression methods are described. This chapter also reviews general concerns with networked image transmission.

Chapter 3 contains a general description of image forming visual systems found in the animal kingdom, including naturally occurring image compression. This chapter also describes several features found within biological visual systems which effectively prioritize image data, providing spatially variant natural protection against information loss.

Chapter 4 is an overview of the main existing anthropomorphic computer vision implementations. While some transformations have tried to adhere strictly to the natural model, others have taken animate visual systems as merely a guideline. Distinctive features of each method are given.

Chapter 5 contains a description of the work of the author in developing unique VR compression methods and enhancing existing ones. Comparisons with respect to biological visual characteristics presented in Section 3.2 are made.

Chapter 6 addresses several issues relevant to the analysis of VR compression algorithms.

Chapter 7 outlines how the presented research impacts the areas of *image archiving*, *videophones*, *videoconferencing*, and *network transmissions*. Specific applications, tools, and prototypes are presented in these areas, using the techniques under discussion.

Chapter 8 proposes directions where the image compression research could be extended. Areas in which further ATM data transmission research could continue are also mentioned.

Chapter 9 concludes the material presented in this work. It summarizes the image compression material presented as well as the results of the ATM simulation.

Chapter 2

Image Compression and Transmission

As described in Section 1.1, the demand for efficient information handling is increasing rapidly. In many situations, the volume of information creates a tremendous burden on the system in use. It is therefore essential that the information be *compressed* — that is, the information must be encoded in some alternate format that takes up less space.

All compression methods are based on some assumptions of the data's characteristics. An accurate model of the data allows us to use that knowledge to make "shortcuts" in encoding the information. If, however, our models are not accurate, the method will not work well, or will not work at all. For example, a compression scheme that works well for compressing English text might not work as well for compressing French text. It is important to use the method best suited to the specific information being handled.

As researchers have identified more accurate models of the information we use, the number of different compression methods has increased. Also, technological advances in areas such as images and sound have added to the types of data we are using, further increasing the need for new compression methods.

In this chapter, we will review the most popular image compression methods currently in use. A general description of each is given. In Section 2.1 the two main categories of compression methods are defined. The motivation for, distinctive features of, and formulae for *variable resolution* compression methods are given in Chapters 4 and 5.

2.1 Lossless vs. Lossy Compression

Within the area of data compression, two main approaches classify all algorithms. *Lossless* compression, as its name implies, loses no information during compression. After decompression, the resulting data is identical to the original data in every respect. Many situations require such accuracy. When compressing something like a novel or detailed financial records, one can only be satisfied with a compression process that leaves the information intact; deviations from the original are unacceptable. Nor can discrepancies — no matter how minute — be tolerated in medical images, for instance, for reasons of safety. The accuracy lossless compression methods provide is obviously crucial in situations that may involve a matter of life and death.

Lossy compression methods, however do not provide the assurance of maintaining 100 percent accuracy of the original data; some data is usually maintained. Most lossy compression methods, however, have a degree of control over how much accuracy is lost. During the compression process, the user can usually indicate the degree of accuracy desired.

Although most lossy compression methods judiciously decide what information may be lost, it is natural to question any loss. What would be the advantage of lossy over lossless methods? The answer is that the former achieve much higher *compression ratios* than the latter. A compression ratio is a measurement of how well a method achieves its goal in reducing the size of the information. By removing or changing a small portion of the original data, lossy compression methods can reduce the size of the entire data collection much more effectively than if they are forced to maintain flawless accuracy. In situations where small imperfections are not crucial, lossy methods are preferable. For example, some image compression schemes do not maintain exact accuracy in very high frequencies of the image when assuming its purpose after decompression is to be viewed by humans. The reasoning is that the human visual system is not very sensitive to high frequencies, and will not perceive the small errors introduced.

2.2 Theoretical Considerations

When studying data compression, a natural question to pose is, "How much compression is possible?" What this question really addresses is the minimum number of bits necessary to encode the data. This limit can be calculated using the concept of *entropy*. Entropy is the amount of information contained in a string of data, and is calculated with the following formula:

$$\text{Number of bits} = -\sum_i p_i \log_2(p_i) \qquad (2.1)$$

The frequency (probability) of the occurrence of the value i is represented by p_i [89]. Clearly, the minimum number of bits necessary to encode the data stream can be calculated quite simply, once the probabilities of occurrence are established.

What is not so clear is how close to this theoretical minimum any compression method may come. We can approach the minimum by taking advantage of the knowledge we have of each value's probability, and by studying the type of data we are working with. Special codes can be assigned to each of the values, with shorter codes being given to values expected to occur more frequently. This approach is known as *entropy encoding*, and the more non-uniformly distributed the values' occurrences in the data are, the more effective it is. Written text, therefore, would compress quite well using entropy encoding schemes, while typical digital images would not. Sections 2.2.1 and 2.2.2 outline several methods based on entropy encoding.

2.2.1 Shannon-Fano / Huffman Encoding

One of the popular encoding schemes based on predictions of value frequency within data is *Huffman* encoding. It was developed in the early 1950's by D. A. Huffman. It assigns a unique binary code to each value. These binary codes are structured in such a way that, although they vary in length, no code is the prefix of another. Huffman codes are usually built with the use of binary trees. Each leaf on the tree represents one of the values, and the length of the binary code depends on the distance the leaf is from the root. Shorter codes are given to values which are expected to appear in the data more frequently, reducing the average number of bits per value necessary for encoding all the information [61].

Huffman encoding was an improvement on *Shannon-Fano* encoding, another minimum redundancy encoding scheme. This method was invented by C. Shannon and R. M. Fano, and operated by building a binary tree structure consisting of subdividing tables of values. Branching occurs to maintain a balance of frequencies, wherever possible, and one bit is added on to the code at each branching level. Again, more frequent symbols, having larger values, will be unable to be subdivided sooner, and thus stay closer to the root; they will have shorter codes.

It must be pointed out that these methods work well only if the values in the data do not all appear with roughly the same frequency. They must have *skewed* distributions. English text is a good example of data that compresses well under Huffman encoding. In most typical samples of English text, the letter 'a' appears more often than the letter 'z'. If the more frequent letters are given shorter codes than the less frequent ones, we can expect reasonable compression. The UNIX "pack" utility employs the Huffman encoding technique for file compression.

2.2.2 LZW Encoding

A more recent approach to encoding data is based on the observation that often patterns of values occur quite frequently. In English text, for example, the values 't', 'h', and 'e' often appear in this order. This pattern could be stored in a buffer and simply assigned one code, eliminating the need to encode each value separately. Such schemes are referred to as *Dictionary* compression methods, as they end up generating a look-up table — or dictionary — of patterns being encoded.

A. Lempel and J. Ziv first proposed this idea in 1977. Their method searched for repeating patterns in the data, using a one pass sliding window. Known as LZ77, this technique replaced the second (or greater) occurrence of any pattern found with a pointer to the first occurrence in the window. In 1978, they modified their technique, to produce LZ78. Here, a table of patterns is constructed. These patterns are extended wherever possible, and future occurrences of these patterns are replaced with appropriate references to the growing table. If new patterns were discovered, they were added to the table.

In 1984, T. Welch introduced still more refinements to the LZ77 and LZ78 methods [107]. One of the greatest improvements in his method was to initialize the dictionary with entries for each separate value before compression began. His technique, known as LZW, is widely used within several image file formats, the UNIX "compress" utility, and various archivers.

2.3 Single Image Compression

Values and pattern probabilities within image data typically have a fairly uniform distribution, and therefore do not compress well under methods based primarily on entropy encoding. There are, however, schemes which employ such methods. Common techniques concentrate on image continuity or eliminating details difficult to detect. Entropy encoding techniques are then used to further compress data structures or look-up tables used in the process.

Figure 2.1[1] compares a few of the most popular lossy single image compression schemes.

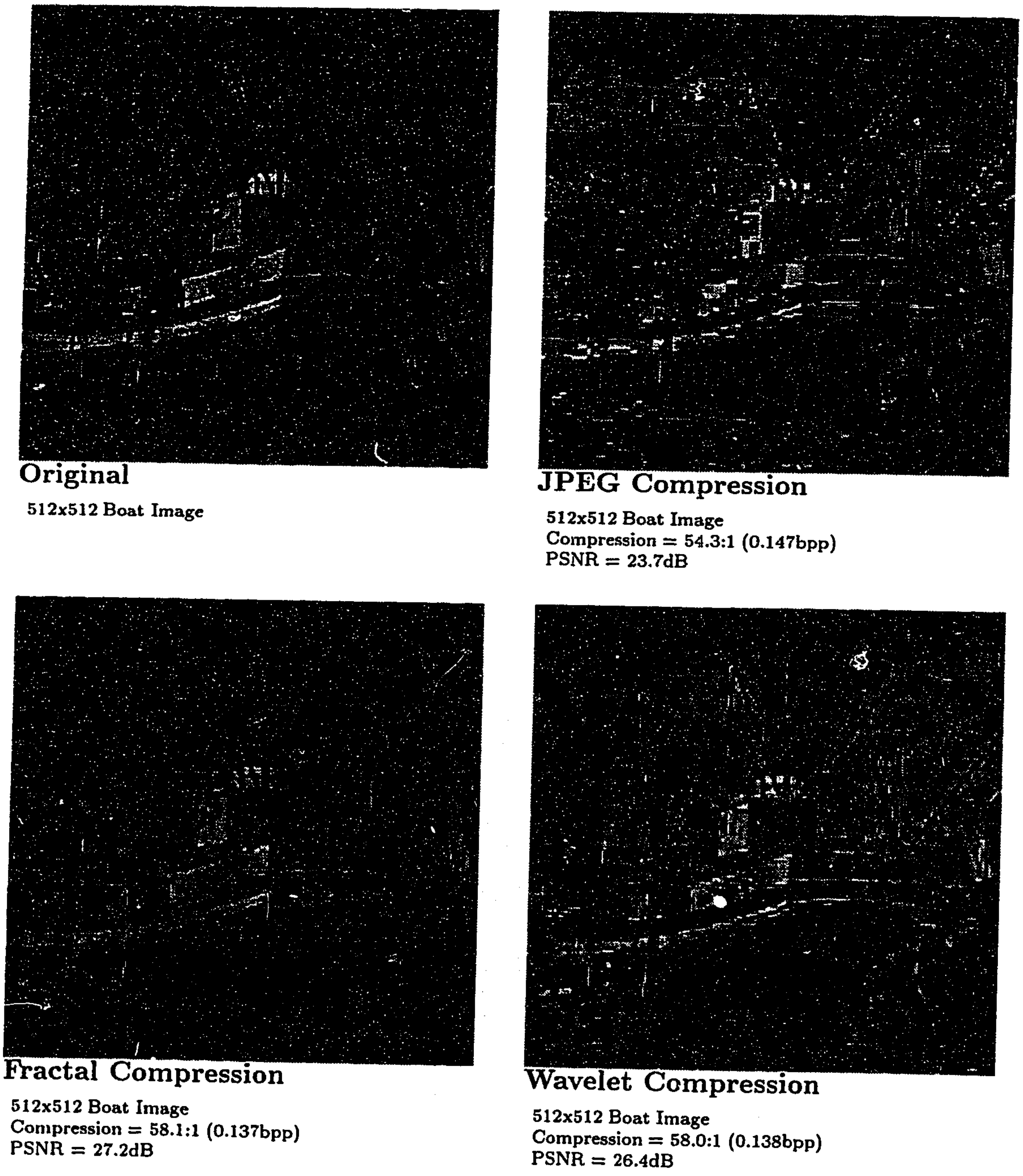
2.3.1 Graphic Interchange Format

The CompuServe organization has developed a lossless compression method which has enjoyed widespread popularity for many years — the Graphic Interchange Format (GIF) [23]. It preserves one byte of data for each pixel in the image. As it is based on a modification of the LZW dictionary compression method, it can be classified as an entropy encoding scheme. The colour map is quantized to 8 bits per pixel, or a maximum of 256 colours.

2.3.2 Tagged Interchange File Format

The Tagged Interchange File Format (TIFF) was designed by Microsoft and Aldus. It is meant to be a collection of other formats, an extendible superset of compression methods, including Huffman and LZW [61]. Depending on the type of data being compressed, alternate encoding methods can be used within the TIFF. A "tag" is put at the beginning of each compressed file to indicate which method is necessary for decompression.

[1] Reprinted from [32], page 681. Used with permission.



Original

512x512 Boat Image

JPEG Compression

512x512 Boat Image
Compression = 54.3:1 (0.147bpp)
PSNR = 23.7dB

Fractal Compression

512x512 Boat Image
Compression = 58.1:1 (0.137bpp)
PSNR = 27.2dB

Wavelet Compression

512x512 Boat Image
Compression = 58.0:1 (0.138bpp)
PSNR = 26.4dB

Figure 2.1: Example results of JPEG, Wavelet, and Fractal compression.

2.3.3 JPEG Compression

The International Standards Organization (ISO) and the Comité Consultatif International Télégraphique et Téléphonique (CCITT) joined forces and formed the Joint Photographic Expert Group (JPEG). They collaborated on a project investigating compression methods specifically designed for image data [5]. Their work between 1986 and 1991 resulted in the JPEG compression method.

JPEG, similar to TIFF, incorporates several compression techniques. The lossless method included in JPEG uses a predictive encoding scheme which estimates each pixel's value from the neighbouring pixels above and to the left of it. The errors in estimation are further compressed using Huffman encoding.

The most interesting compression method included in the JPEG standard is a lossy technique based on the two dimensional Discrete Cosine Transform (DCT) [104]. Like the famous Fourier Transform, the DCT is a lossless reversible function that transforms images to the frequency domain. The reason JPEG uses the DCT is that the image can be compressed differently when represented in the frequency domain, as opposed to the spatial domain. The formula of the DCT is given as follows [89]:

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) cos\frac{(2x+1)u\pi}{16} cos\frac{(2y+1)v\pi}{16} \tag{2.2}$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u,v=0 \\ 1 & \text{otherwise} \end{cases}$$

For reasons of efficiency, the image is first segmented into blocks of 64 pixels. Each of these blocks is then transformed to the frequency domain using the DCT. It has been previously noted that the human visual system is not as sensitive to the higher frequency information in images as it is to lower frequencies. Therefore, the JPEG method quantizes the frequency information non-uniformly, eliminating more of the high frequency information.

From here, the information is compressed using Huffman, predictive, and subsampling techniques. Specifically, pixel colours are represented in the luminosity-chrominance colour space, as opposed to the typical red-green-blue representation. This allows pixel intensity to be sampled more frequently than colour, as the human visual system is more sensitive to brightness than chrominance. Such techniques have proved to be extremely useful, and JPEG has gained worldwide acceptance as a viable standard in image compression. It must be noted, however, that since the JPEG techniques are based heavily on the characteristics of the human visual system, they may not perform adequately on images intended for machine use.

The main drawbacks to the JPEG algorithm are lack of speed and continuity. The DCT transform is computationally expensive. Also, although the compression ratios can be varied to balance image quality with actual compression, at high compression values the algorithm's segmentation interferes with final clarity. The 64 pixel blocks begin to appear, causing quite noticeable "blocky" artifacts.

2.3.4 Wavelet Compression

Wavelet compression interprets the image plane as a function of frequency versus time, as opposed to frequencies only — as in JPEG. Wavelets are a set of base functions of differing frequency upon which the image is built. There are several families of basis functions such as *Haar* and *Daubechies.* The wavelet bases are not discrete, but are symmetric wave pulses that decrease in amplitude continuously from their origin. The wavelet transform decomposes the phase plane and quantizes it in a similar method to JPEG [22] in that the original image (signal) can be reconstructed using the encoded temporal and frequency information.

Wavelet compression is well suited to some specific types of images, such as bilevel scanned text, and not suited to images with muted variations, textures, or soft edges. Unlike the DCT, wavelets involve only a convolution over the image, making them less mathematically complex and resulting in less computationally expensive implementations. *Fast Wavelet Transforms* exist in the same sense as fast DCT algorithms, making some wavelet compression methods even faster than JPEG. For example, the *Fast Fourier Transform* and *Fast Discrete Cosine Transform* have computational complexities of $O(n \log_2(n))$, while the *Fast Wavelet Transform* is of $O(n)$ [103]. One can also operate wavelet compression based on 64 bit blocks as JPEG does, but the problem of blocky artifacts remains.

2.3.5 Fractal Compression

Fractal compression uses complex fractal equations to achieve high compression ratios. Very complex equations are necessary to encode an image, with the most difficult task being to determine the values of the parameters in the equations. Numerous iterations within the algorithm are necessary to achieve reasonable image quality. This process is extremely time consuming — almost prohibitively so. The benefit, however, is that once the values are determined, decompression is relatively effortless [50].

Although it is still in its infancy, fractal compression has many promising aspects. The high compression ratios reported to date are indeed attractive. The ease of decompression is also of interest. More research is necessary on the compression side to speed up and completely automate the process. At present, some human guidance is often necessary during compression.

2.4 Image Sequence Compression

When one wishes to encode a movie or video clip, many of the standard image compression techniques can be used. There are some important differences between a set of unrelated images and a set of images taken in sequence. In the latter case, one expects only relatively small differences between consecutive images.

While some image sequence compression methods, such as *Motion JPEG* (M-JPEG), simply compress each image independently using standard image compression techniques [21], others take advantage of the temporal redundancies. Such considerations greatly increase compression ratios.

2.4.1 CCITT H.261 Compression

The compression of video to facilitate transmission over telephone lines was studied by the CCITT, and a standard, H.261, was developed for such a purpose [5]. Phone companies allocated bandwidth in ISDN channels of 64Kb/s each, which this standard took into account, and so it has more commonly become known as the "px64"[2] compression scheme.

[2] Pronounced "P times sixty-four."

The px64 method uses the DCT transform in much the same way as JPEG does in *intra* frames. However, it does not always compress each frame separately, but instead compresses only the change in the frequencies of each 64 pixel block in *inter* frames. That is, it compares successive frames and only records the difference, skewing the data towards zero and allowing the subsequent entropy encoding schemes to operate more effectively. The greater the similarity between frames the greater the skewing effect [62].

2.4.2 MPEG Compression

Like JPEG, the Motion Picture Expert Group (MPEG) was formed for the purpose of establishing a compression standard — this time for digital video image sequences [36, 53, 54]. While the px64 was designed to be decompressed in the forward direction only, MPEG sought to allow simple decompression in both directions. MPEG also addressed the issue of px64's vulnerability to transmission errors. Three different compression structures were developed, namely Intraframes (I), Predictive frames (P), and Bidirectional frames (B). These different frames are interleaved during transmission, with different ratios varying the quality and compression ratio.

Intraframes are compressed as independent images using a method similar to JPEG. Unlike px64, all errors introduced are not passed through the entire sequence, but stop once an I frame is encountered. P frames use predictive encoding schemes based on the previous I or P frame. B frames are designed to allow viewing of the frames in reverse order. They encode image data in 256 pixel blocks (*macroblocks*), compensating for pattern (area) motion within the scene.

2.5 Networked Scene Transmission

Many computer applications transmit images from one location to another, over a network. One example of an increasingly popular application based on foveated scenes is the videophone. The typical image sequence which is transmitted includes a human face, which is of greater interest to the viewer than the peripheral background.

An assumption in many traditional implementations is that the transmission of image data occurs over reliable, static bandwidth connections. However, in some network scenarios, such as with *asynchronous transfer mode* (ATM) technologies, this assumption does not hold [56]. Different network characteristics provide possibilities for optimizing the transmission of specific classes of image and video data. We will look to biological visual systems for ideas on how to address these issues. The three main concepts involved in scene transmission are data reduction, bandwidth allocation, and data prioritization.

2.5.1 Data Reduction

All data compression, in general, results in a fewer number of bytes encoding information. This is also true for image compression, including the compression techniques discussed earlier in Sections 2.3 and 2.4. If the information is being sent from site to site, this reduction in bytes translates to a necessity to transmit less actual data. Where network charges are calculated on the number of transmitted bytes, such a reduction could result in significant savings, depending on the compression ratio achieved. Therefore, image compression can be a significant concern.

2.5.2 Bandwidth Allocation

Data reduction prior to transmission can also effectively reduce the requirements for network bandwidth. Situations where greater demands are placed on networks already at capacity usually require the elimination of some of the traffic, or and upgrading of the network. Data compression is a much less costly solution. Where restrictions on bandwidth cannot be avoided, data compression allows a larger amount of information to be transmitted within the same number of bytes.

2.5.3 Packet Prioritization

When information is transmitted over a network, sometimes the data must be broken up into *packets* and sent separately. ATM networks operate on this principle. When the network gets congested, some of these packets must be dropped and are therefore not successfully transmitted. If there is a method of assigning priority levels to individual packets, usually the ones with the lowest priorities are lost first.

Chapter 3

Animate Visual Compression and Prioritization

In this chapter we outline the basic structure of the animate visual system. Most notably, the unique spatially variant qualities of the retina will be described. Other works describe this in greater detail [74].

An introduction to the physiology of animate vision is given in Section 3.1. It provides a description of foveal regions and saccadic movements and explains their important role in vision and perception.

As outlined in Section 1.1, many computer applications require some form of data compression in order to reduce the bandwidth required for transmission, the memory required for storage, and the processing required for manipulation.

For the same reason of optimal resource management, many biological visual systems reduce the amount of sensory data transmitted from the optical receptors to the brain. One of the data reduction methods employed in several stages during transmission is that of spatially varying the resolution. This concept deserves close attention for its possible application to various problems in computer vision.

The area assumed to be of greater interest, the *fovea*, requires more detail than the *periphery*. This effect can be achieved by decreasing the spatial sampling rate of the image as one moves further from the location of the fovea, thereby spatially varying the resolution. One cannot remove the periphery entirely, however, as it provides important general information such as motion, texture, and an overall context for the scene.

Clearly, VR methods cannot be used to compress all data, as frequently the area of primary interest cannot be determined in advance or quickly located automatically. In some domains, such as with medical images, the distortions caused by inaccurate foveal location may be unacceptable. However, there are some applications, such as videoconferencing, where the VR technique gives high compression ratios and acceptable image quality for relatively low computational cost.

3.1 Animate Vision

To say that the animate visual system is complex in design is a gross understatement. Scientists are still far from consensus as to exactly what occurs between the act of looking and the result of seeing. Much has been learned through experiments and dissections, but still more remains hidden. The sense of vision is naturally intertwined with cognition and self-awareness. One can quickly come to the conclusion that the secrets of vision and the secrets of how the mind works may have much in common [80]. Discoveries in one area may provide insights into the other. Our research begins with the area of vision that scientists have had the most success in understanding — the physical optic sensors. The human eye is probably the best understood in this respect.

The eye is an organ with the purpose of transmitting visual information about its surroundings to the brain. It is not necessary for the eye itself to understand the information, just to encode and relay it. Light entering the eye passes through an opening in the *iris* called the *pupil.* The curvature of the *cornea* and *lens* focus the light onto the back of the eyeball. The *retina* consists of a layer of special cells along the back of the eyeball that capture the projected image and transmit it to the brain via the *ganglion* cells and the *optic nerve* [74]. (See Appendix A for a diagram of the eye.)

Active Vision

The concept of *active vision* differs from *passive vision* in that tasks such as scanning, exploring, and searching are seen as essential to the perception process. Rather than merely conveying information to the brain about the surroundings that happen to present themselves, active visual systems exhibit purposive responses to the information they collect. Many researchers have begun to promote the view of animate perception as highly active [2, 3, 6–8, 10, 11, 70, 81, 90, 98]. With this approach, the concepts of a fovea and saccadic movements become increasingly relevant.

3.1.1 Retina

The retina is the layer of cells covering the inner surface of the back of the eye. These cells encode the two dimensional image projected onto it into a pulse repetition rate of neuron transmissions to the brain. The human retina is covered by approximately 125 million receptor cells of two types; *rods* and *cones* [55]. Rods operate best in low light and do not detect colour. Cones detect colour and do not operate well without high illumination [66].

As with most animals, in humans the majority of information received by these cells cannot be transmitted directly to the brain due to the small neural capacity of the optic nerve. Only about one million separate neurons connect the 125 million receptors in the eye to the brain.

3.1.2 Fovea

The 125 million rods and cones are not evenly distributed over the human retina. One small region, called the *fovea* [1], occupies approximately 2° out of the total visual range of 60° vertically and 180° horizontally. It is located near the optical axis and is mainly comprised of cones. The periphery is mainly comprised of rods. It is estimated that one hundred thousand receptors are present in the fovea and the remainder of the 125 million receptors are located in the periphery. The foveal area , or *macula lutea,* is used primarily to convey information about colour, shape, form, and perspective, while the periphery is used to detect motion, intensity, and context.

The outer receptors are also arranged so that their density decreases with distance from the fovea. The eye is directed by the brain to fixate on points of interest [114], so clearly the portion of the image projected on or near the fovea is assumed to be of more importance and therefore is sampled at a greater resolution. This decrease in

[1] In animals, the degrees of resource concentration in the foveal area of the retina differ. Biologists have loosely defined several terms for what we will continue to call a *fovea* for reasons of simplicity. Such an area is subjectively classified, from greater to lesser prominence, as a *fovea, foveola, area* (*area centralis*), or *region of high cell density* (RHCD). A *visual streak* is a less pronounced elliptical foveal area, usually horizontal, often connecting a temporal and a nasal fovea.

peripheral image sampling is just one way natural systems achieve variable resolution data compression.

A second method of reducing the amount of data transmitted to the brain involves the allocation of neural channels in the optic nerve. As previously mentioned, the transmission of data collected from the 125 million receptors in the human retina can only occur over the 1 million channels available to the brain (ganglion cells). The 100 000 foveal receptors are allocated channels at a ratio of 1:1, while the peripheral receptors are allocated the remaining 900 000. Data reduction does not occur in the fovea, while the compression ratio of the periphery at this stage is approximately 140:1. Roughly 10% of the optic nerve's capacity is utilized by the fovea, while the fovea operates on only .04% of the visual field.

Here again, biological visual systems have reduced the data transmitted to the brain with respect to the position of the fovea. The result is a spatially variable resolution visual sensor which has both a wide field of view and high local acuity. The clearest indication of the overall spatial compression that occurs before the data passes through the optic head is ganglion cell density. Therefore, topographical ganglion cell isodensity maps are of great interest in our research.

Saccadic Movements

The view through the structure of our eyes can be described as a frosty shower door with a tiny spot rubbed clear. If this is true, why does our vision not appear as such? The answer is that the spatially variable resolution is combined with *saccadic movements* to produce the illusion of a uniformly detailed field of view. Saccadic movements are the often jerky movements of the eye that position the fovea over the area of the scene we are most interested in viewing [19, 42, 113]. Our entire visual field appears in detail because the fovea is always placed exactly where we are "looking."

The classical problem of determining the focus of attention can be approached by categorizing gaze control mechanisms into three classes [83]. *Reflex* eye movements are involuntary reactions to sudden changes in scene or tracking smoothly moving objects[2]. *Task driven* eye movements depend on the actions being performed. As an example, one's eyes tend to alternate between focusing on the ground plane and the

[2] These smooth saccadic movements are called *optokinetic nystagmus*.

horizon while walking. Finally, *voluntary* eye movements are explicitly controlled by high level cognition processes, such as visual searching or recognition.

Active visual systems are therefore unique. They are comprised of both a variable resolution sensor and rapid, accurate foveal positioning controls [3, 6].

3.1.3 Phylogenetic Visual Characteristics

Although not all animals have foveal areas (e.g., goldfish), most advanced biological visual systems do have a non-uniform distribution of ganglion cells along their retinae. Some biological systems (e.g., human) employ a single fovea and a relatively isotropical ganglion cell decrease with respect to eccentricity. Other animals, such as the eagle possess a *visual streak*: a horizontally elongated band of high acuity (Figure 3.1 [48]). Just 5% of birds have no foveation, and 41% have at least two foveae [18, 68, 86].

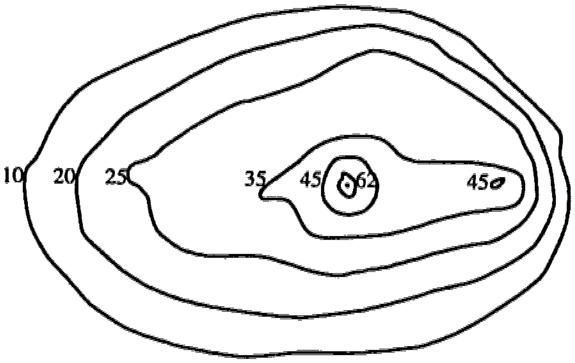


Figure 3.1: Ganglion cell isodensity map for the eagle. Dorsal direction is up and nasal direction is to the left of the page. Numbers indicate thousands of cells per mm^2. Notice the strong central and temporal foveae as well as a moderate horizontal visual streak.

The owl, sunbird, nuthatch, and blue jay are examples of monofoveate birds, although the position of the fovea with respect to the optic center of the eye differs. Procellariiform seabirds have been found to have a pronounced visual streak, usually with a central fovea [43]. Eagles, hawks, vultures, swallows, kingfishers, terns, bitterns, and hummingbirds are all bifoveate species with some form of horizontal visual streak, however the relative weights of their foveae differ.

Most canines have one well developed fovea in the temporal region of the retina with a visual streak extending in the nasal region. The topographical map of ganglion cell densities, therefore, includes a teardrop shaped visual streak. To date, all known studies on wolves have found that they have pronounced visual streaks, while most dogs have only moderate streaks. Studies included beagles, German shepherds, basset hounds, dobermanns, entlebuchers, as well as timber and Alaska wolves [73].

While cheetahs, leopards, and tree shrews have pronounced visual streaks, cats, mice, hedgehogs, and squirrels do not. Such differences can be explained once factors such as natural terrain are taken into account (see Section 3.2) [52, 67, 69, 92].

3.1.4 Visual Memory

Psychologists tell us that we build a mental representation of what we see — an internal world view. If we did not create an inner model, we would not be able to function; our surroundings would be a constant surprise. But how do we create it? We do not merely take one look at a scene and obtain all the necessary information from that; we are not capable of detailed perception over a large area. We instead, gather information about our environment over time. We use our eyes to scan our surroundings and slowly build a model. The more time we have to survey our surroundings, the more detailed our internal model will become. Scanning, then, is the method used in animate vision — a method whose application to computer vision is worthy of further study.

Some say that our visual system drives the internal representation, while others claim the relationship between the two is the reverse. Does what I look at always determine what I will see? Or are illusions more easily explained by the fact that what I see is what I expect to see, not necessarily what I am looking at? The truth probably lies somewhere in between. To be sure, one does not require visual signals free of noise on order to generate an effective mental model of the world. Anthropomorphically motivated computer applications, therefore, need not be immediately dismissed if an imperfect method of image capture is utilized.

3.2 Five Ecological Specializations

The unique visual systems of each animal can give researchers clues to the animal's interaction with its environment. Variations in habitat and lifestyle have a direct relation to the specialization of the visual system [1, 46, 88]. The study of an animal's retina can provide clues to its daily routine. Indeed, a variety of procellariiform seabirds who hunt for squid miles from shore, and who had never actually been seen feeding were studied in [43]. Close inspection of their foveae and visual streak, however, provided clues to the details of their hunting methods, which were later verified. Even species which are closely related may have widely differing retinal topographies; it appears that the individual's habitat is the determining factor [46]. While there is still some speculation about the purpose of some components of visual systems (the bird's pecten, for example), much of the specific differentiation has been linked to assisting the animal in some routine function.

There are five common topographical isodensity features of biological retinae that are of particular interest to our research:

- **general characteristics** — single fovea, continuity, anisotropism, receptor boundary shape
- **multiple foveae** — weighted foveae
- **visual streaks**
- **optic discs**
- **multiple optic paths** — dynamic foveae

3.2.1 General Characteristics

Some generalities can be drawn from surveying a wide variety of animals. In binocular vision there seems to be a tradeoff between a large field of view and acute foveae. Predators tend to have keen frontal vision with a strong temporal fovea and a limited posterior field of view; tracking a highly motile target is of greater importance than being on constant alert for other predators. Prey, on the other hand, tend to have wide fields of view with limited foveal capacity, in order to detect predators and navigate an evasion. A cat, for instance, has a 99° binocular field, a strong fovea, and 187° field of view. A rabbit has a 360° total field of view, but only a weak visual streak and a 24° binocular field.

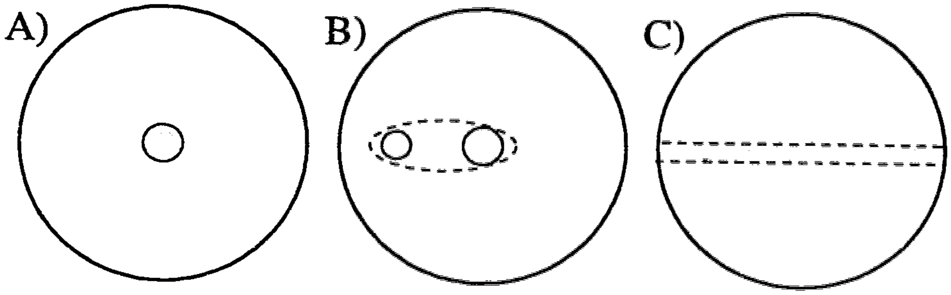


Figure 3.2: Stylized common retinal topographies. **A:** Ground feeders. **B:** Predators. **C:** Animals in wide open habitats. (Foveal areas appear shaded.)

While humans possess a single fovea, the majority of advanced visual systems are multifoveate. Three main classifications of retinal topographies can be made, as shown in Figure 3.2. Clearly only monofoveate systems (e.g., Figure 3.2 A) such as human eyes could be broadly classified as having isotropic ganglion isodensities. Anisotropic nonconformal cortical projections predominate, specifically suited to the environment or tasks.

Avian eyes with a single, usually central fovea, are the most common and are associated with ground feeding birds (e.g., pigeons). The relatively simple task of detecting and pecking static or slowly moving objects such as berries, seeds, and bugs requires a single area of accurate vision [63].

The natural shape of the eyeball is always spheroid, but the actual surface shape of the retina can vary. The border of the visual field in humans is elliptical, common to most animals. As well, most retinae are continuous at the back of the eye, with the possible exception of an optic disc. It is important to note these obvious characteristics of animate visual systems when adapting biological methods to the predominantly square, computing domain with possible mathematical singularities in the equations. This is discussed further in Chapter 4 and Section 5.1.

3.2.2 Multiple Foveae

Most avian predators (e.g., hawks, eagles) have more than one foveal area. The central or temporal fovea is used for binocular vision. Tracking rapidly moving or camouflaged prey requires this keen frontal binocular fixation on proximate objects. The nasal foveae are usually smaller and provide a wider panoramic view of the surroundings. These foveae are used for monocular foveation on surrounding objects such as trees during navigation. Fine detail is not as necessary here as in the temporal fovea. For example, a hummingbird will rely on its temporal fovea for catching insects, and its nasal fovea while eating nectar. A minor visual streak also assists navigation, as described below. The relative strength of the two foveal areas depends on the visual purpose. Typically carrion eating birds which pursue from the ground (e.g., vulture, condor, chimango) have weaker temporal foveae, while predators which capture live prey from flight or perches (e.g., eagle, hawk) have acute temporal foveae.

3.2.3 Visual Streaks

Finally, mammals or birds whose habitat is either in open spaces (e.g., leopard) or near water (e.g., puffin) often have a predominant visual streak and, occasionally, a poorly developed fovea. This can be linked to the greater importance the horizon has in the sensory ecology. Navigation with respect to the horizon and scanning for surface food both benefit from the visual streak. Eye or head movements are greatly reduced and a greater sensitivity to horizontal motion (e.g., prey) is gained. A strong visual streak is not typically found in animals that scurry among bushes (e.g., mouse, hedgehog, cat), because for them vertical vision is as important as horizontal. Most animals whose heads commonly assume a wide variety of orientations during daily activities (e.g., macaque) also exhibit an absence of a strong visual streak.

Even within similar configurations, different environments correspond to slightly differing retinal topographies. Examples of animals with visual streaks include herbivores (e.g., rabbit), carnivores (e.g., leopard), and ungulates (e.g., horse), whose habitats are fields or plains where the horizon is dominant. While most visual streaks are dorsal to the *optic disc* (blind spot), the rabbit's is ventral, due to its smaller size causing the horizon to be higher in its field of view. Some taller animals (e.g., cow, fallow deer) also have a weak vertical streak-like area rising upward from the fovea, called the *anakatabatic area* [46]. This area improves detail to the scene on the ground immediately in front of the animal.

A wide range of streak strength exists in the canine family, although it appears that wild species living in open terrain (e.g., wolf) have the expected strong visual streaks. Domesticated dogs, however, vary greatly, even within the same litter. It has been proposed that extensive breeding in the history of domesticated dogs has introduced a wide variety of visual streak expressions. If returned to the wild, the pressures of natural selection might eventually reduce streak variability [48].

3.2.4 Optic Discs

The arrangement of the layers of cells in vertebrate retinae requires the optic nerve to pass through the photoreceptor layer, creating an optic disc, or "blind spot." However, the shape and position of the optic disc is also often clearly related to the animal's ecology. Some prairie dogs and squirrels have thin elongated optic discs. Such specializations are for coping in the darker environment of diurnal activities and the need to reduce the risk of any object going undetected by falling entirely within the blind spot.

Cephalopods (e.g., octopuses, squids) have a different layering arrangement in their retinae which does not produce blind spots [74]. Certainly this specialization is the optimal solution to reducing the negative effect of operating with a blind area within the visual field.

3.2.5 Multiple Optic Paths

The purpose of multiple optic paths is most often related to the need for the visual system to operate well within two different media, air and water. The refraction of light is different for each, posing a problem. By using multiple optic paths, the eye can be structured such that one path works well in each medium. For example, the penguin has an egg shaped lens that focuses the light differently on two areas of the retina. When below water, the penguin can concentrate on the visual data received clearly on one part of the retina. When above water, it can concentrate on the other retinal area, now in focus. The dolphin similarly uses two areas of its retina — however, it directs the incoming light by appropriately constricting its pupil in a very irregular manner [1]. The bifoveate kingfisher bird also uses one fovea (temporal) while hunting under water and the other (nasal) while in the air.

Several fish (e.g., Atlantic flying fish, *Dialommus fuscus*, *Anableps anableps*) also require clear vision both in and out of water. Their eyes have divided retinae and separate corneal facets to compensate for their respective media. For *Anableps anableps*, the "top" retina can be used to scan for predatory birds in the air and the "bottom" retina can be used to navigate and search for food underwater.

Mice within the genus *mus* have a divided retina as well. Here the purpose is not to see through multiple media, but rather to take advantage of the generally constant but different colours of the sky (blue) and the ground (green). A predator is more easily seen against a rich blue or green background if the appropriate area of the retina has predominantly green or blue sensitive cones. To this end, the top and bottom areas of the retina are divided.

3.3 Biological Prioritization

The human visual system has a marvelous capacity of image compression, transmission, and processing. The system contains a natural information communications network, transmitting visual data in the form of electrical pulses from the retinae, along the optic nerve, to the occipital lobe of the brain. Like all computer networks, the biological system is prone to bandwidth limitations, damage, errors, and numerous malfunctions. Studies have shown that one method of coping with such constraints that is utilized by animate visual networks involves the concept of *data prioritization* with respect to the foveae. At many levels, the biological model prioritizes visual information provided by the fovea or macular area of the retina, as this information is assumed to be of primary importance.

3.3.1 Retinal Resilience

Research with rodents, goldfish, quail, and humans has demonstrated that peripheral photoreceptors in the retina are more susceptible to light damage when exposed to spatially homogeneous lighting conditions than those in foveal areas [91][3]. Recent studies suggest that aging retinae steadily lose peripheral cones, while foveal cones remain stable [38]. During adulthood, less sensitive rods, but not cones are lost in the fovea [26].

[3] It is believed the fovea is preserved because the macular pigment is a carotenoid antioxidant that decreases light-initiated lipid peroxidation, reducing the damaging effects of photochemical reactions [57].

3.3.2 Optic Nerve

The optic nerves connecting the retinal sensors and the occipital lobe can usually be viewed as a robust, static network. However, the human visual system has the ability to manage trauma intelligently and continue to function when this network is compromised.

The foveal data is transmitted through the core of the optic nerve, along the optic pathway. It is cushioned by the surrounding neurons which carry the peripheral data (see Figure 3.3). Many injuries or degenerative diseases which may damage the optic pathway would affect the periphery before reaching the fovea [55].

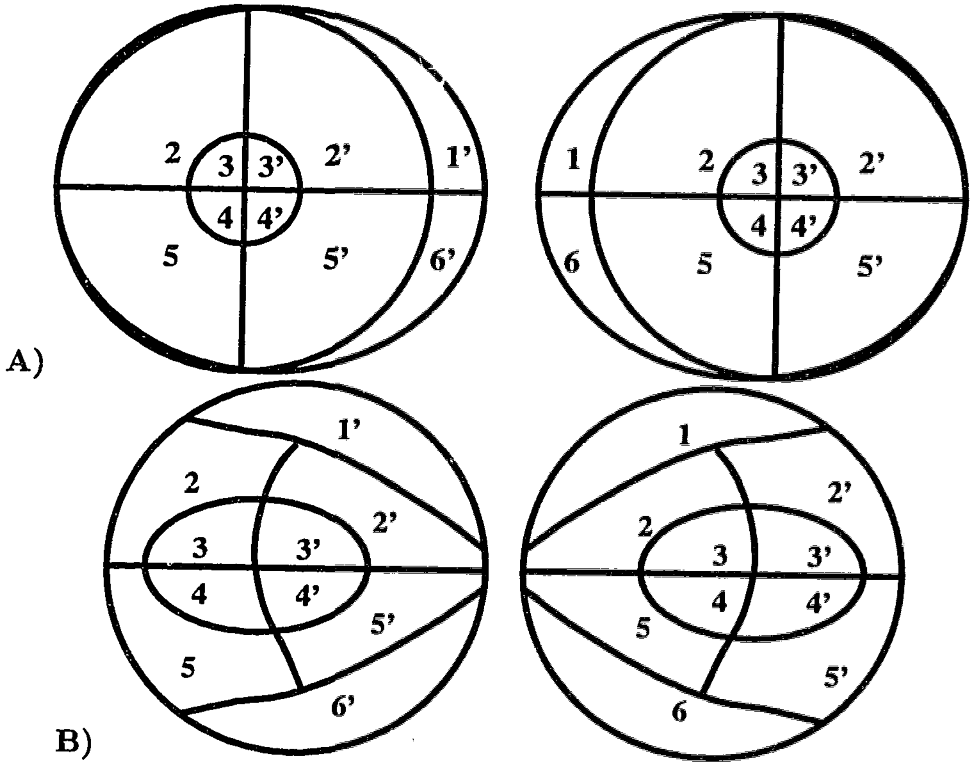


Figure 3.3: Arrangement of six areas of the visual field within the retinae (**A**) and the optic nerve (**B**). Cardinal numerals refer to the left half of the visual field, prime numerals to the right half (from [55] p. 24).

3.3.3 Blood Supply to the Occipital Lobe

Prioritization is also involved within the brain itself. The portion of the occipital lobe which processes foveal information has a dual blood supply, from both the middle cerebral and posterior cerebral arteries. The peripheral portions of the occipital lobe have only one nourishing artery. Therefore, in situations of head trauma, vascular disease, or temporary drop in blood pressure, the foveal information will often continue to be processed where the periphery will not be. This explains the common occurrence of gradual peripheral vision failure prior to fainting as well as the phenomenon of *macular sparing* seen in some stroke patients.

Therefore, in many traumatic situations where biological visual system networks experience gradual impairment, peripheral information is increasingly lost until only a small, blurred area remains in the center of the field of view. Human subjects relating experiences of gradual visual impairment often describe darkness "closing in" from the periphery. The retinae, optic nerve, and blood supply to the brain all contribute to this effect, known as *tunnel vision*. Total blindness is a possible final stage.

Chapter 4

Established Biological Modelling

The image compression schemes described in Section 2.3 treat the entire image uniformly, and we can infer from the information presented in Section 3.2 that this is not always necessary nor desirable. Studies have shown that animate vision has much higher resolution in the center of the visual field than in the periphery. In this chapter we describe several computer implementations of variable resolution algorithms. Although most of the methods are, to some degree, based on animate visual systems, the actual implementations have usually taken some simplifying liberties.

In Section 4.1.1, a review of the motivation for investigating animate vision is given. The cortical projection transform is discussed in Section 4.2, along with the formulae. Section 4.3 introduces the logarithmic approximation of visual systems. Section 4.4 presents a unique transform with varying resolution in only one dimension. The fish eye transform is presented in Section 4.5, and a modified version which produces square images is described in Section 4.6. It is this last method that is the starting point for our research.

4.1 Preliminaries

4.1.1 Anthropomorphic Benefits

Section 1.1 clearly outlined the need for image compression. Chapter 2 noted that effective compression methods spring from a solid understanding of the data under consideration. When working with computerized images, there are two main reasons we should look to animate visual systems for inspiration:

- Animate visual systems are the only natural systems that are able to perform complex manipulations and operations with detailed optic data. The way these systems function using images may provide important clues to the nature of visual data itself.
- Many of our image processing and compression methods will be used on data specifically intended for human viewing. Considering the user's visual system may allow us to tailor our algorithms, taking advantage of biological fixation and visual characteristics.

4.1.2 Pre-computational Modelling

The foveated aspect of human vision was recognized long before modern computers were invented. In fact, Johannes Vermeer van Delft, a painter, experimented with the *camera obscura* in the mid-seventeenth century in order to model the optics of the human eye. He joined several other Dutch painters and began the *Delft* school of painting, which sought optical accuracy within their work. Figure 4.1 is an example of Vermeer's work from the Louvre in Paris. It demonstrates the approximation the effect of the fovea on human vision. Vermeer intentionally painted finer detail around the regions of interest (hands, face) while purposely blurring the foreground, ball of string, and periphery.

The arrival of computers has empowered us to investigate and model biological visual systems with greater precision. JPEG and other schemes have utilized the fact that humans cannot detect changes in chrominance as well as they can detect changes in intensity. While such knowledge is useful, our research has concentrated primarily on the specific characteristics of spatial subsampling. Retinal layout and cortical projections provide us with the basic knowledge in this area.

At present, there are several computational models of some of the biological visual characteristics described in Section 3.1. However, until now, only the simplest of retinal topographies have been attempted. We will describe the methods of *Schwartz' Cortical Projection*, the *Log Polar* transform, the *Reciprocal Wedge* transform, the *Basic Variable Resolution* (BVR) transform, and the *Stretched Variable Resolution* (SVR) transform.

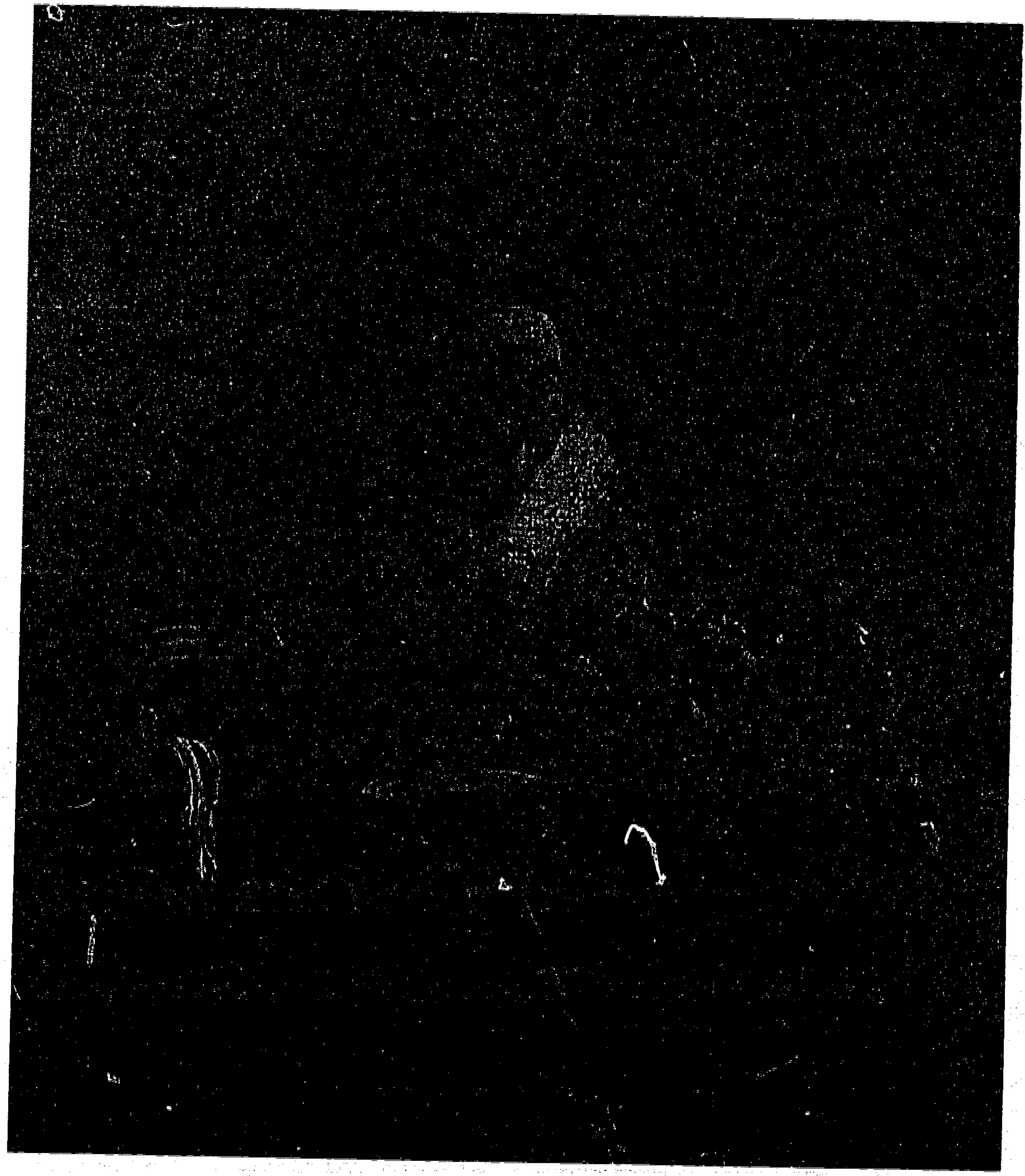


Figure 4.1: *The Lacemaker* by Johannes Vermeer van Delft, circa 1660. In pursuit of optical accuracy, he approximates the affect of the fovea on human vision.

4.1.3 Essential Characteristics

Keep in mind that when accurately modelling the animate visual system three main features appear vital. To some degree, all three are found in most biological visual systems:

- **Fovea:** While animate systems differ in the number, shape, or location of foveae, the information transmitted from the retina to the brain is usually of non-uniform spatial resolution.
- **Continuity:** Many biological visual systems maintain *image continuity* up to and sometimes including the projection of image data onto the cortex. Typically, while the true image may be transformed in many ways, information from adjacent areas in that image remain adjacent in the projected, compressed data.
- **Anisotropism:** While one might simplify the human visual compression into an isotropic transformation, closer inspection of the range in the animal kingdom indicates a wider variety of retinal topographies among other species. Indeed, the human eye is almost isotropic, but in general, the radial symmetry of an animal's eye closely reflects its habitat and function within daily activities. It is not isotropism itself that is important to achieve, but rather the match between topography and task.

4.1.4 Coordinate Systems

An image may be indexed using polar or Cartesian coordinate systems. A simple transformation indexes a point from one frame of reference to another. Formally, the mapping function from the Cartesian (x, y) coordinates to the log polar (r, θ) coordinates is given by

$$r = \sqrt{x^2 + y^2} \tag{4.1}$$

$$\theta = \tan^{-1}(y/x) \tag{4.2}$$

The decision as to which system to use depends on the actions being performed on the image. Image processing methods based on biological visual systems frequently lend themselves well to the use of polar coordinates. One of the reasons for this is the virtual isotropic nature of the retina found in the human eye.

4.2 Schwartz' Cortical Projection

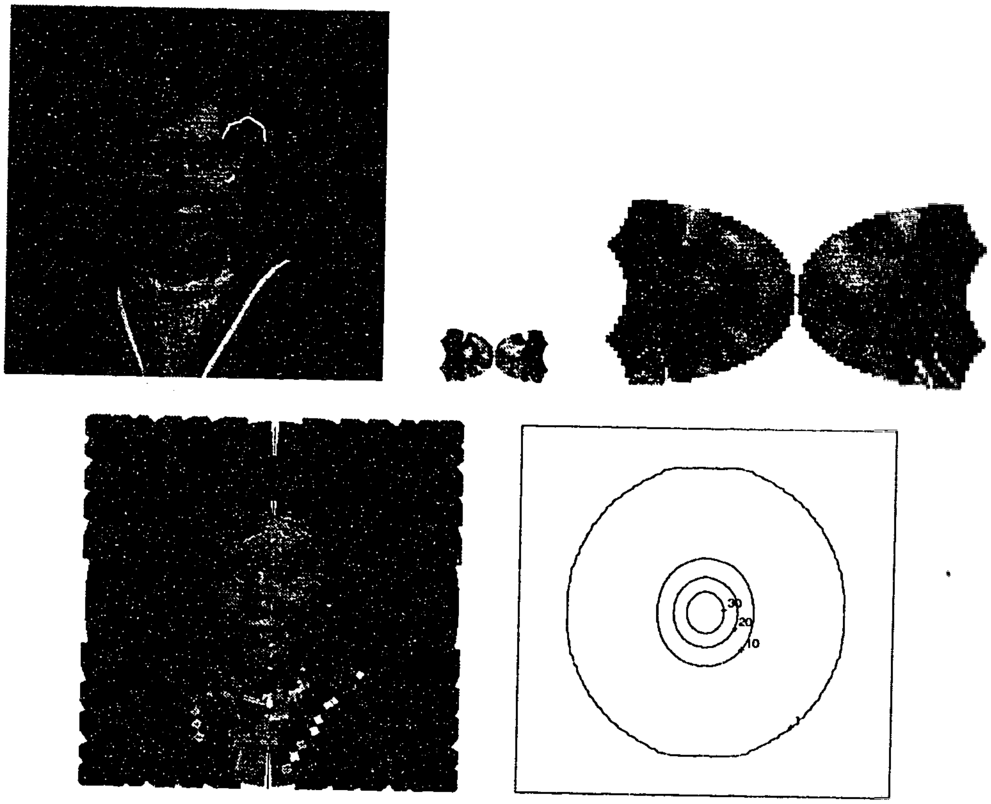


Figure 4.2: Schwartz' cortical projection:
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map. (The numbers give the sampling density in percent.)

Schwartz researched variable resolution models of vision. His research concluded that the retinal image mapped onto the visual cortex (occipital lobe) of the brain is approximated by a conformal log polar mapping [84, 85]. Figure 4.2 illustrates how this retinal mapping maintains a wide field of view, a single acute foveal area, significant data reduction, and reasonable resolution [82]. Schwartz preferred to express this transformation using complex variables, defined as

$$z = x + iy \tag{4.3}$$

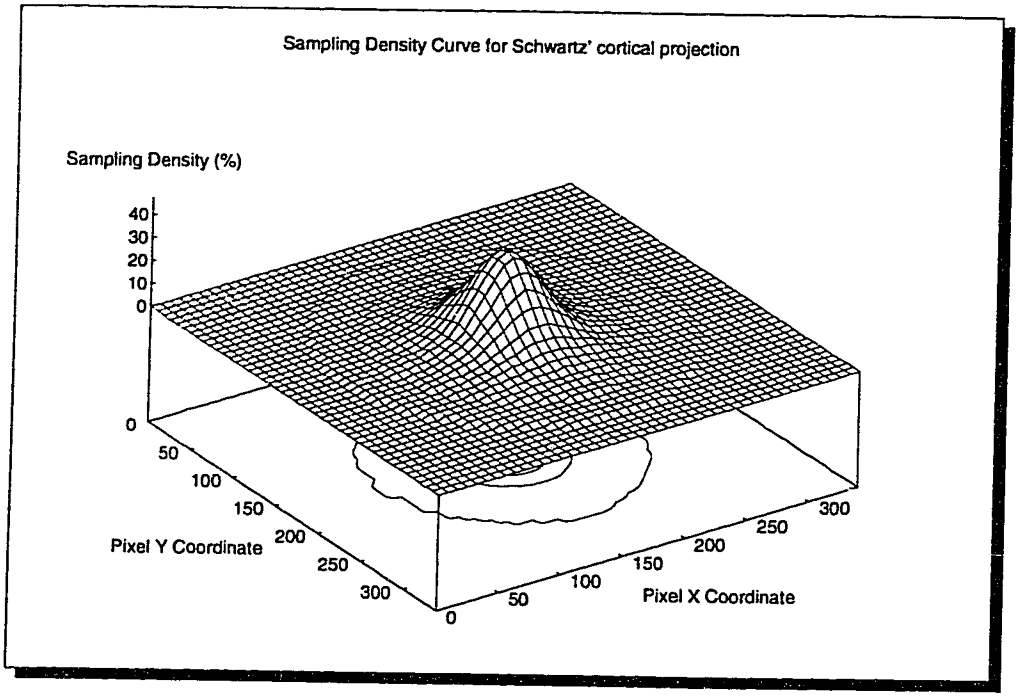


Figure 4.3: Schwartz' cortical projection: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 4.2.

$$w = x' + iy' \tag{4.4}$$

where z expresses the coordinates in the original image, and where w expresses the coordinates in the transformed image. Note that if initially polar coordinates are used, as defined in Equations 4.1 and 4.2, Equation 4.3 can also be defined as

$$z = re^{i\theta}. \tag{4.5}$$

With these complex variables, Schwartz' cortical projection (or "global retinotopic transformation") is expressed as

$$w = \ln(z + a) \tag{4.6}$$

where a is a shifting parameter that eliminates the singularity otherwise present at the origin. A practical disadvantage of Schwartz' cortical projection is that continuity along the vertical meridian is lost. Boundary checks for wraparound conditions must be made when traversing through an image; filtering and correlation calculations become more complex.

4.3 Log Polar Transform

The log polar transformation is another example of a system that is best expressed using polar coordinates (Figure 4.4).

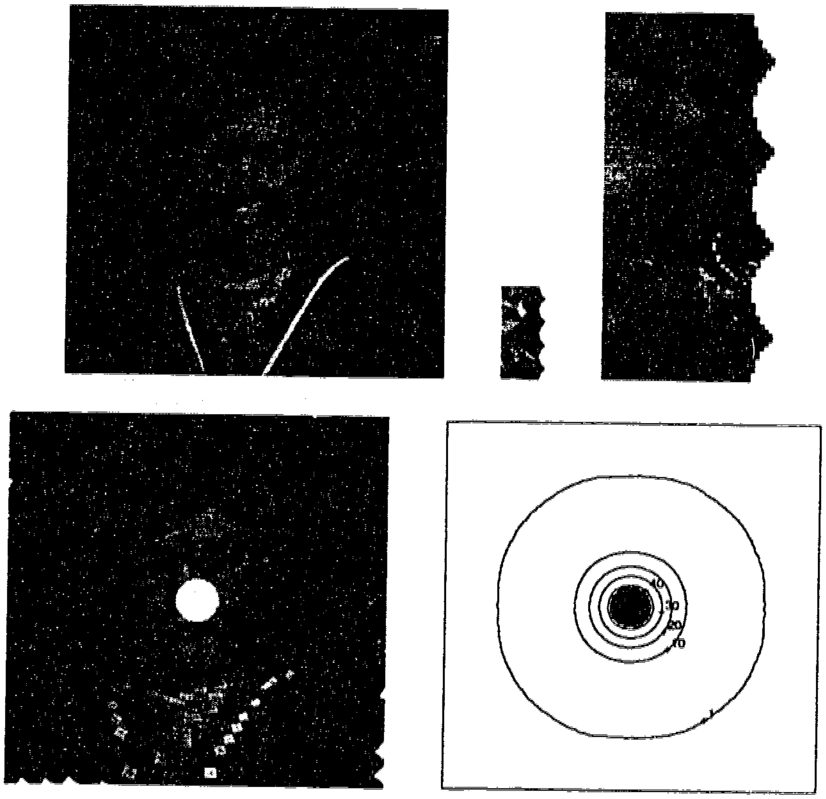


Figure 4.4: Log Polar transform:
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map (grey area is unsampled).

The log polar transform is a popular simplification of Schwartz' complex logarithmic mapping, and is defined as

$$r' = \ln r \tag{4.7}$$

$$\theta' = \theta \tag{4.8}$$

where r and θ are the coordinates in the original image, and where r' and θ' are the coordinates in the transformed image. It is shown in [106] that within this coordinate system scaling becomes a simple shift operation in the r' dimension, and rotation a simple shift operation in the θ' dimension. These properties make the log polar

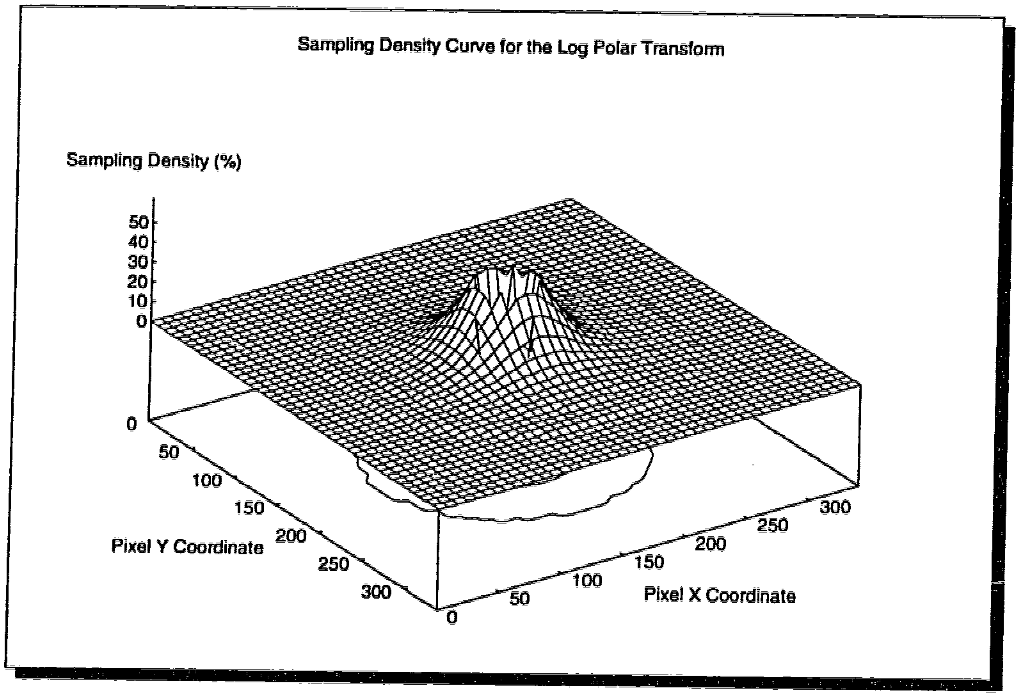


Figure 4.5: Log Polar transform: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 4.4.

transform an attractive method for two dimensional object recognition [81], the estimation of depth from motion [94], and the evaluation of time-to-impact from optical flow [95].

There are some major drawbacks to log polar systems used in practice, as addressed in [35, 96]. Here, in a rectangular array indexed by r and θ, continuity along the left half of the horizontal meridian is lost, similar to Schwartz' cortical projection. Boundary checks for wraparound conditions must be made when traversing through an image. As well, a singularity exists around the origin, usually requiring a small extra patch or a shift parameter [79, 85], and complicating any image manipulation process. It is often noted that, although it is the most important section of the image, the foveal patch is *not included* in many applications for the sake of simplicity [83].

Isotropic variable resolution sensors capable of capturing images in polar coordinates have also been developed, avoiding the use of mathematical transformations [58, 101]. During fabrication, however, such cameras require expensive variable sampling and circular sensors.

4.4 Reciprocal Wedge Transform

In [96], another space variant transform is proposed as an alternative to the log polar transform. While the log polar transform simplifies many centric transformations such as rotation and scaling about the origin, it complicates other linear functions into logarithmic sine curves [106]. Also, polynomial curves may change their degree after the log polar transform; they retain their degree using the *Reciprocal Wedge Transform* (RWT) (Figure 4.6).

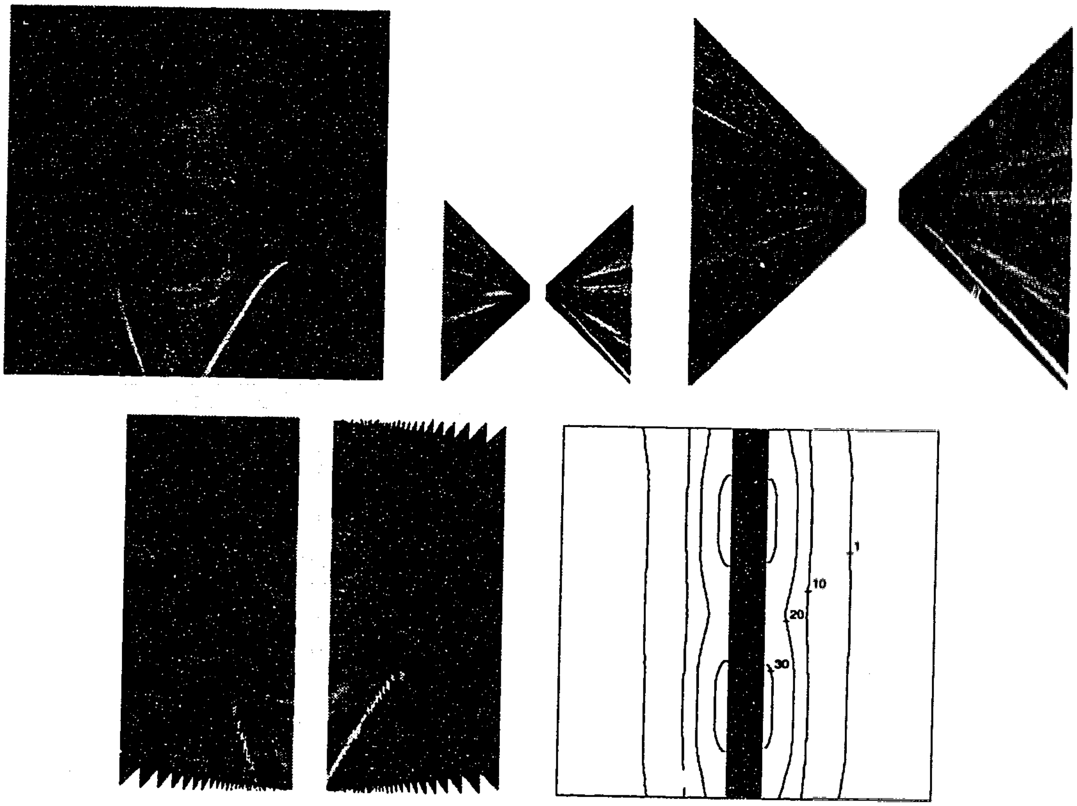


Figure 4.6: Reciprocal Wedge transform:
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map (grey area is unsampled).

Although the RWT is anisotropic, it preserves horizontal scaling, loses resolution predominantly in one dimension, and is better suited to working with linear movements like translations. This transform was not developed for use with image compression, but primarily for line detection, stereo correspondence, and linear motion estimation in robotic navigation and on assembly lines [97].

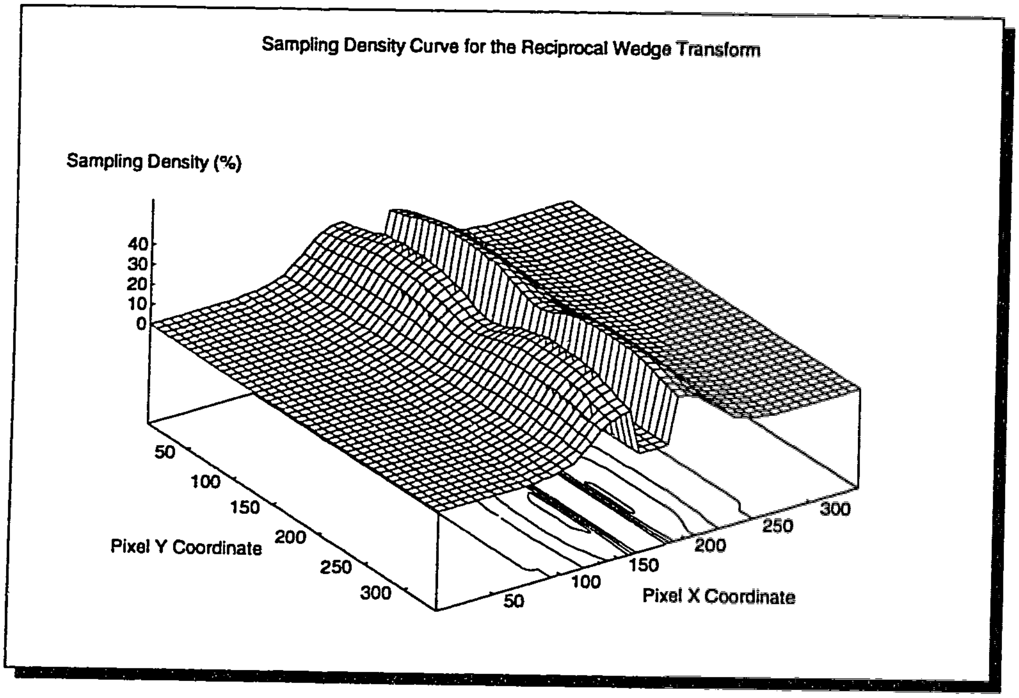


Figure 4.7: Reciprocal Wedge transform: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 4.6.

The RWT is defined as a mapping from (x, y) to (x', y') such that

$$x' = 1/x \tag{4.9}$$

$$y' = y/x \tag{4.10}$$

The RWT also is not continuous; the transformed image is divided along the vertical meridian. The singularity at $x = 0$ affects the center vertical strip intersecting the origin. The range of this strip is usually omitted from any processing [97], to simplify calculations. To operate on the entire image, either a center patch must be included, or a shift parameter a must be added to the equations 4.9 and 4.10 [1].

$$x' = 1/(x + a) \tag{4.11}$$

$$y' = y/(x + a) \tag{4.12}$$

In either case, the calculations become more complex.

[1] The effect of a is a shift of the Cartesian image along the horizontal axis.

4.5 Basic Variable Resolution Transform

The Basic Variable Resolution (BVR) transform is the starting point for much of our research, and is often referred to as the *Fish Eye Transform* (FET) (Figure 4.8).

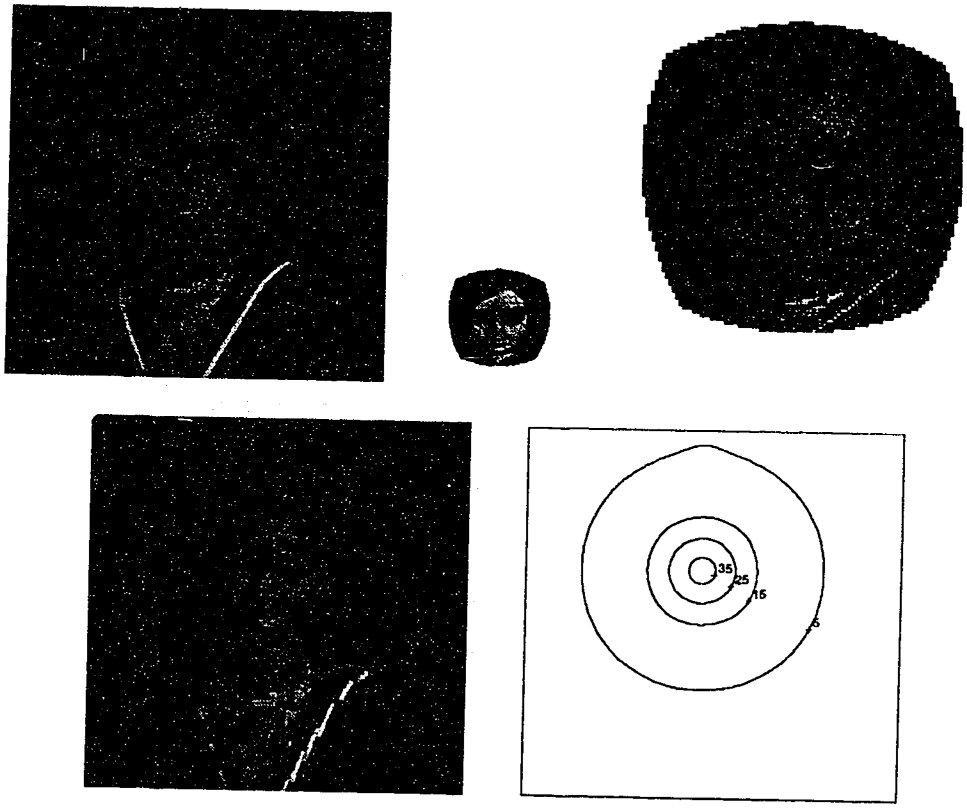


Figure 4.8: Basic Variable Resolution transform with fovea on the face. Central clarity is maintained while the edges become blurred.
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map.

This transform is a simplification of Schwartz' complex logarithmic mapping [12, 13]. It concentrates on providing greater detail around the fovea while maintaining continuity throughout the image. A shift value of 1 is added in the logarithm, avoiding a singularity at the origin. While the exact method outlined by Schwartz is clearly discontinuous across the vertical meridian, the BVR transform is not [14]. The algorithm is isotropic and based on polar coordinates. For our discussions, let (r, θ) represent the polar coordinates of the point (x, y) in the Cartesian domain, where they share the same origin at the fovea.

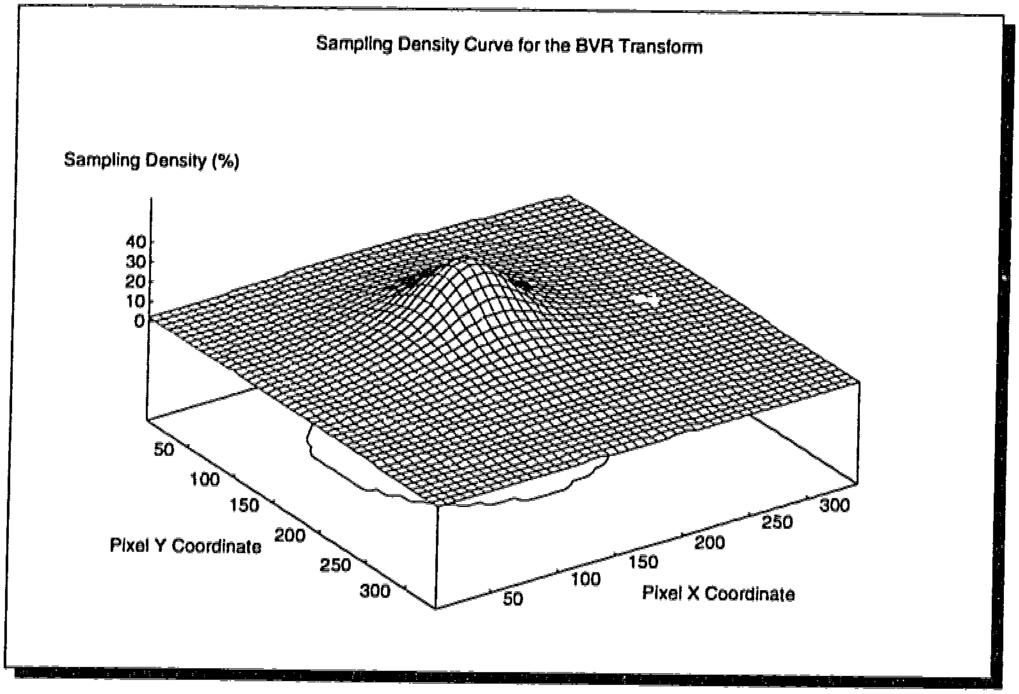


Figure 4.9: Basic Variable Resolution transform: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 4.8.

$$r = \sqrt{x^2 + y^2} \tag{4.13}$$

$$\theta = \tan^{-1}(\frac{y}{x}) \tag{4.14}$$

The BVR equations, then, transform the point (r, θ) to the polar coordinates (r', θ') using the equations:

$$r' = s \ln(r\alpha + 1) \tag{4.15}$$

$$\theta' = \theta \tag{4.16}$$

The Cartesian coordinates after the transform are therefore

$$x' = r' \cos \theta' \tag{4.17}$$

$$y' = r' \sin \theta' \tag{4.18}$$

To restate, the pixel value has moved from a distance of r to r' away from the fovea at its original angle. The value s is a scaling factor which controls the compression ratio, while α controls the distortion effect of the fovea. A high α value will cause the resolution of the periphery to drop substantially as compared to the foveal region (strong fovea). A low α value causes the resolution of the periphery to drop only slightly as one moves out from the fovea (weak fovea).

The reverse transformation is given from the following:

$$\begin{aligned}
r' &= \sqrt{x'^2 + y'^2} && (4.19)\\
\theta' &= \tan^{-1}(\frac{y'}{x'}) && (4.20)\\
r &= \frac{e^{\frac{r'}{s}} - 1}{\alpha} && (4.21)\\
\theta &= \theta' && (4.22)\\
x &= r\cos\theta && (4.23)\\
y &= r\sin\theta && (4.24)
\end{aligned}$$

An additional transform is presented in [60], called the *Polynomial Fish Eye Transform* (PFET). The purpose is the same as the FET, but PFET uses different equations for deriving r'. Here, $r' = F(r)$, where $F(r)$ is of the form

$$F(r) = a_0 + a_1 r + a_2 r^2 + \cdots + a_n r^n = \sum_{j=0}^{n} a_j r^j \qquad (4.25)$$

No simple inverse transform of the PFET exists.

The specific values of n and a_n can be changed to achieve different distortion and compression ratios. Good approximations to fish eye lenses were achieved with $n = 5$ [60]. While the exact effects of changing α and s in the FET were known, the effect of changing one value of a_n is unclear. To achieve similar effects,, many coefficients must be modified simultaneously. Under detailed analysis, however, the PFET does seem to provide better approximations of transformations done by wide angle or fish eye lenses.

4.6 Stretched Variable Resolution Transform

Notice that one of the features of the BVR transform is non-rectangular compressed images, similar to some of the other methods presented. We will address this in more

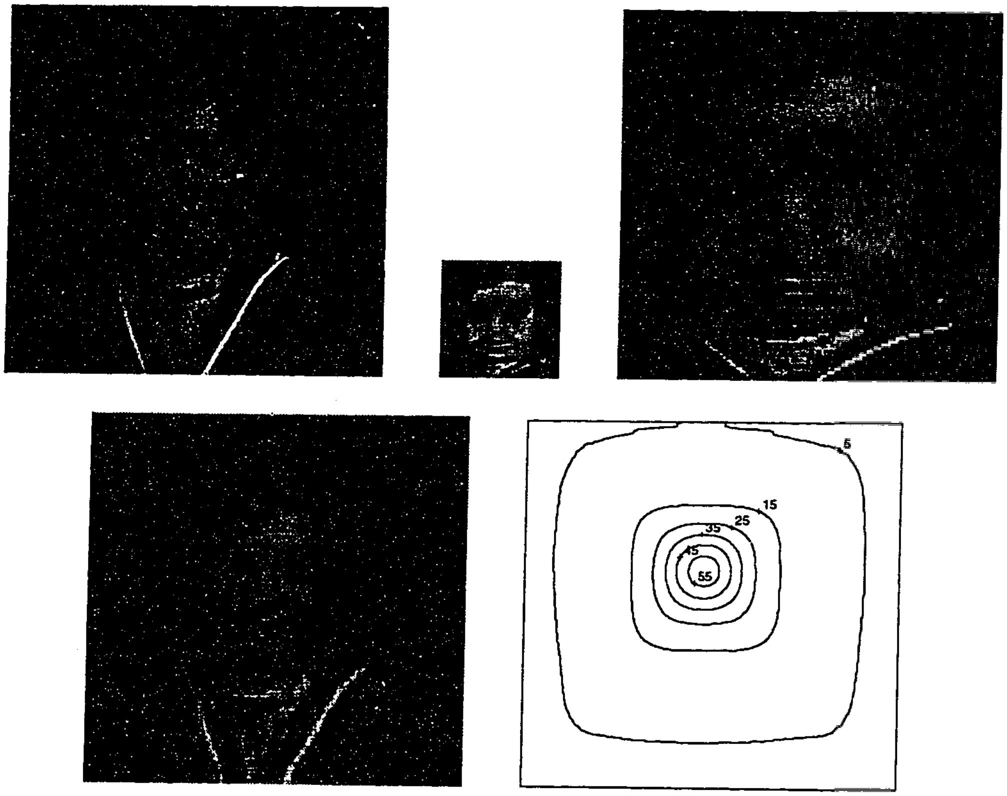


Figure 4.10: Stretched Variable Resolution transform with fovea on the face. Central clarity is maintained while the edges become blurred.
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map.

detail in Section 5.1

The *Stretched Variable Resolution* (SVR) transform, as presented in [16, 89] is based on the BVR transform, but produces rectangular images by using multiple scaling factors. The SVR transform varies s for each point, depending on its polar coordinate angle θ (See Equation 4.26).

For each angle we can calculate the maximum distance to the edge of the original image, $r_{\max}$, and the maximum distance in the compressed image, $r'_{\max}$ [16]. These will, obviously, depend on the location of the fovea (or origin). The desired compression ratio will determine the dimensions of the compressed image, setting $r'_{\max}$.

For each θ, then, the scaling factor becomes:

$$s = \frac{r'_{\max}}{\ln(\alpha r_{\max} + 1)} \tag{4.26}$$

This maps the entire original image to a rectangular compressed image for any

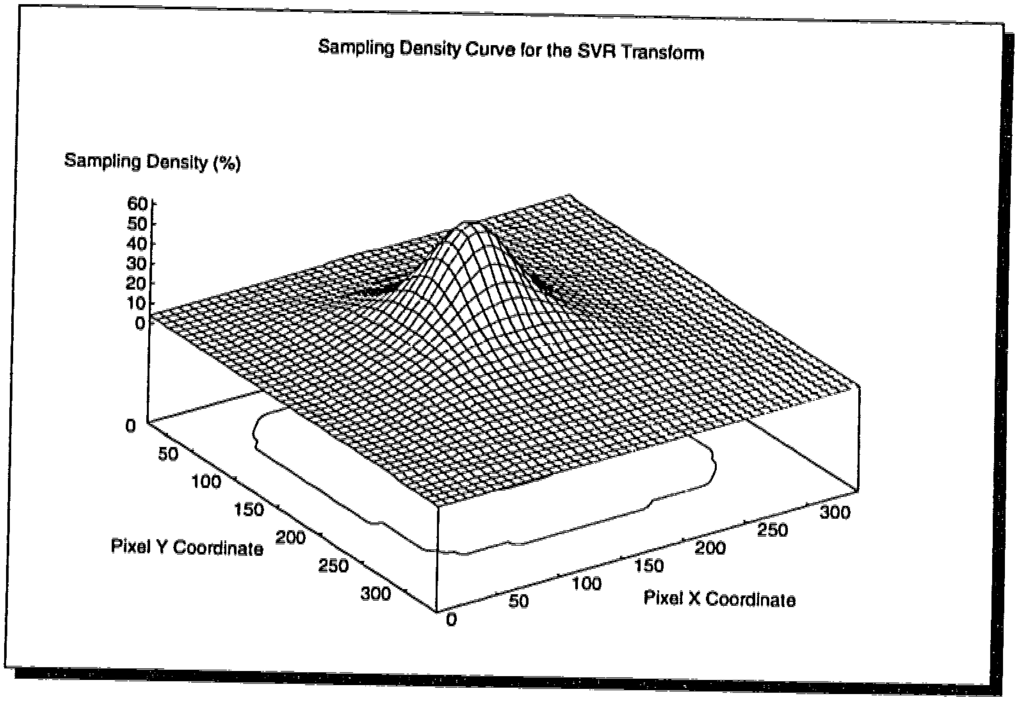


Figure 4.11: Stretched Variable Resolution transform: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 4.10.

value of α or any location of the fovea. The compressed image is effectively *stretched* to fill the entire rectangle. The SVR transform is not isotropic.

Figure 4.10 illustrates the SVR transform used in the compression of an image. This method does a reasonable job of maintaining the anthropomorphic foveal properties of the BVR formulae, but is relatively complex.

The affect of α within the SVR equations is similar to its affect within the BVR equations; it controls the distortion effect of the fovea. Figures 4.12 to 4.17 illustrate the affect of increasing α while maintaining a constant compression parameter value (here, at **95**). Notice how the higher α values will cause the resolution of the periphery to drop more rapidly as compared to the foveal region (strong fovea). The lower α values cause the resolution of the periphery to drop only slightly as one moves out from the fovea (weak fovea).

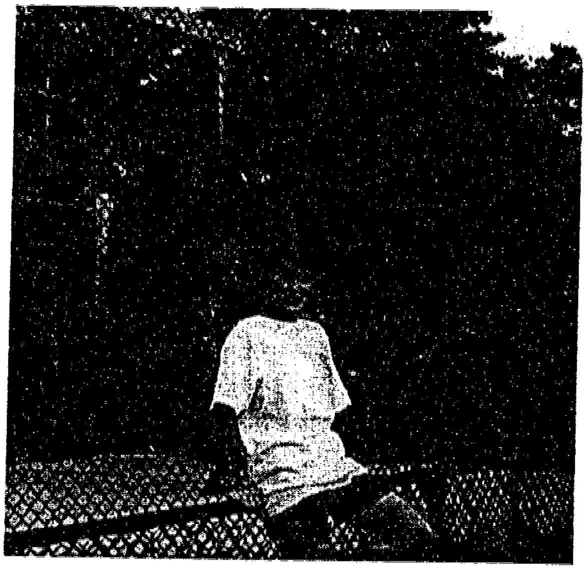
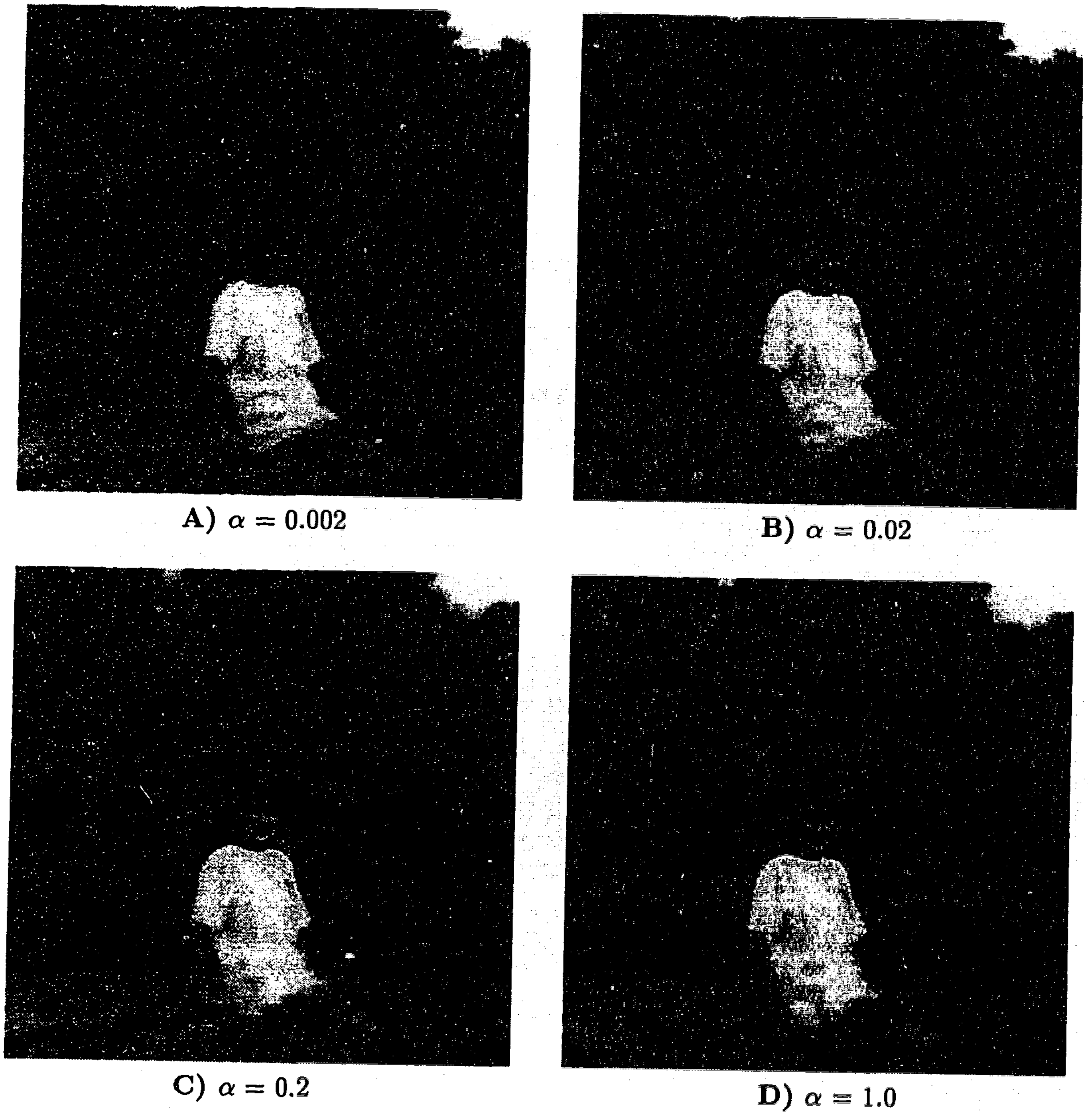


Figure 4.12: An original example image: *Zoo*



A) $\alpha = 0.002$

B) $\alpha = 0.02$

C) $\alpha = 0.2$

D) $\alpha = 1.0$

Figure 4.13: *Zoo* example with different α settings: Compression is constant at 95%. A single fovea on the face increases in strength with α, at the expense of the periphery.

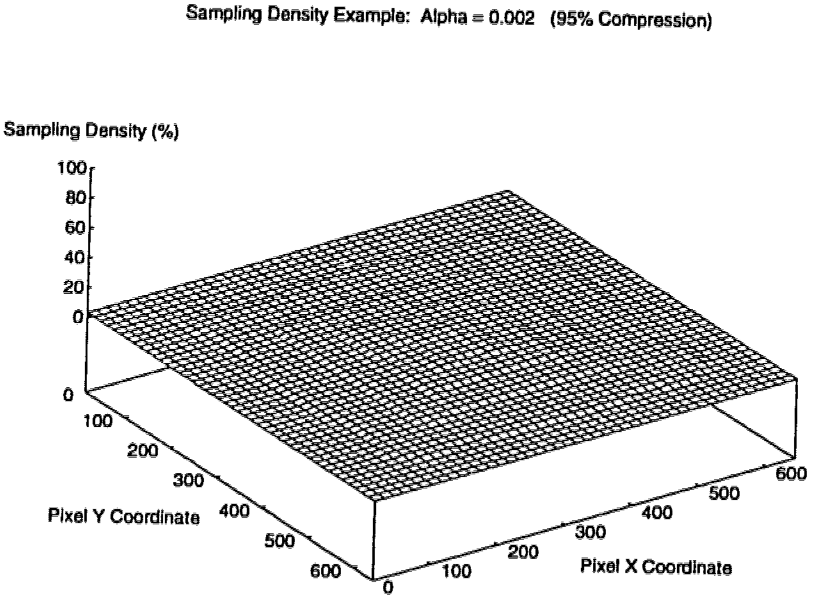


Figure 4.14: Sampling density map for $\alpha = 0.002$ (see Figure 4.13 **A**).

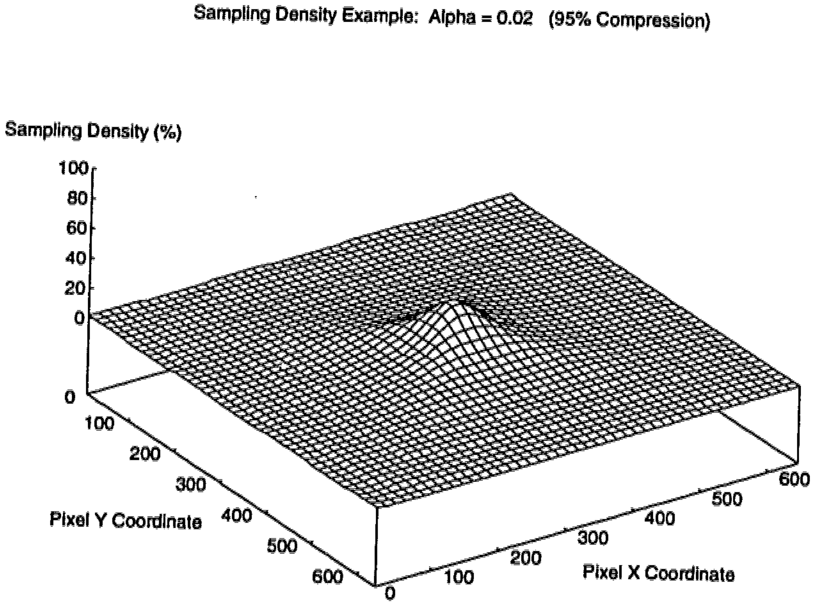


Figure 4.15: Sampling density map for $\alpha = 0.02$ (see Figure 4.13 **B**).

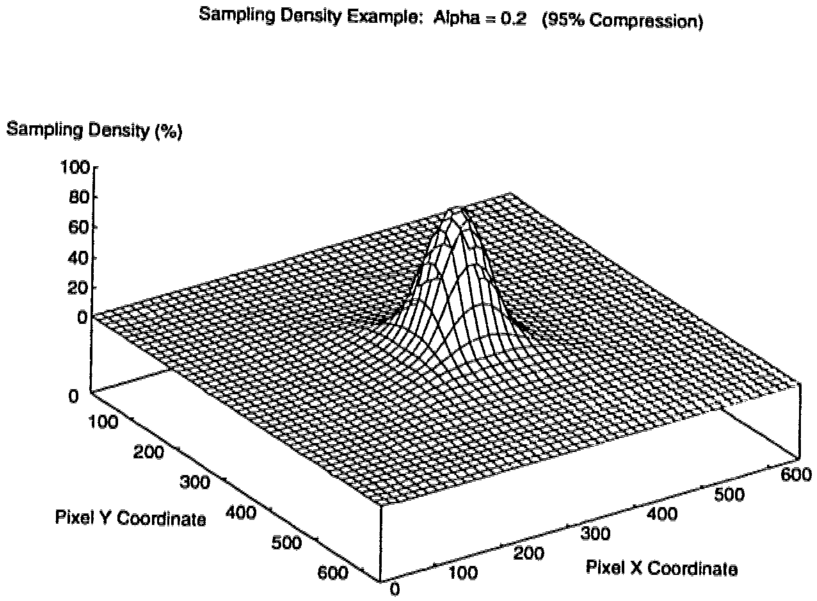


Figure 4.16: Sampling densiiy map for $\alpha = 0.2$ (see Figure 4.13 **C**).

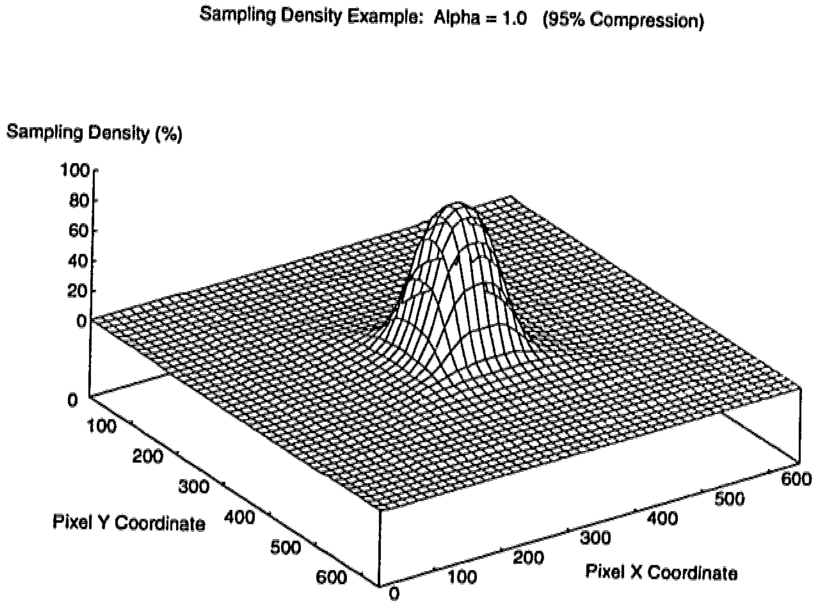


Figure 4.17: Sampling density map for $\alpha = 1.0$ (see Figure 4.13 **D**).

Chapter 5

Foveated Transform Enhancements

In this chapter, we introduce original doctoral thesis material. We indicate the areas in which work has been done, as well as indicate areas in which further research is warranted. Results, tools, and algorithms are also described. Chapter 7 outlines several applications based on the present research.

In Chapter 3 we saw the great variety of visual systems present in the animal kingdom. It is clear that while most share the feature of a fovea to some degree, there are many variations. The methods presented in Chapter 4 almost exclusively modelled monofoveate systems. If we wish to evaluate the comparative quality of these systems, we cannot do this without taking their environments and tasks into consideration. Biologically, each type of eye is equipped with the special features necessary for its purpose within its ecology. Computer vision systems also have unique tasks and environments. By modelling a simple fovea, multiple foveae, visual streaks, optic discs, and multiple optic paths we can build the tools necessary to adapt our computer systems properly for a much wider range of specific purposes. These five features will be addressed in that order.

The first issue we notice, as we move from the animate to the computing domains is the natural digital affinity towards rectangular image shapes.

5.1 Irregularly Shaped Images

While the biologically based image transforms introduced in Chapter 4 are generally accurate, all but the last two methods are discontinuous somewhere within the image. Also, except for SVR, the transforms do not produce both rectangular compressed and rectangular decompressed images from standard rectangular input. One great distinction betwеen the biologically based image transforms introduced in this section, and the previous ones, is the shape of the compressed image.

As an example, the BVR transform of a rectangular image is not a rectangular image, as illustrated in Figure 5.1. This problem is magnified when the fovea is not located near the center of the image, or when high α values are used.

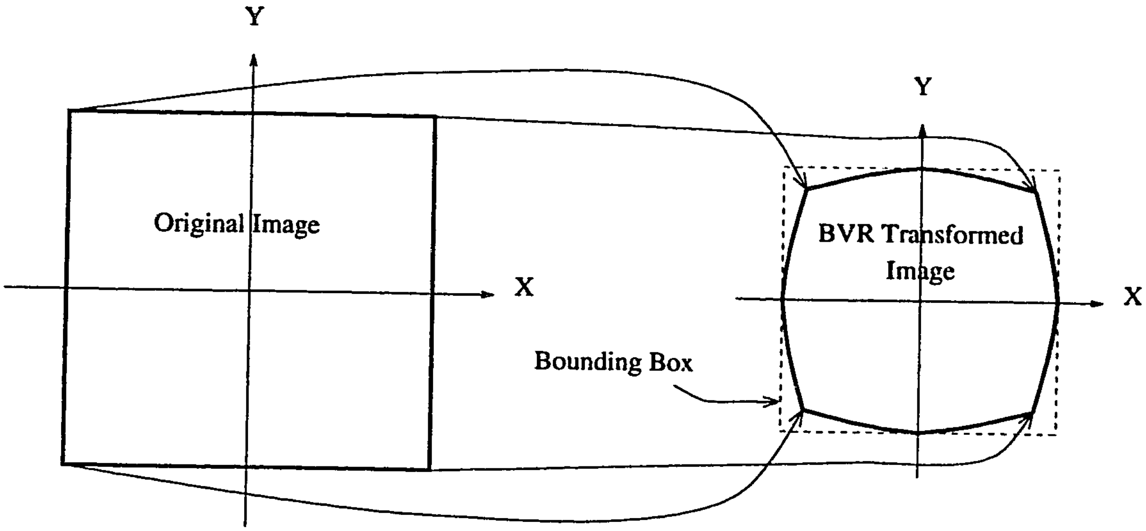


Figure 5.1: The BVR transformation of a rectangular image.

Currently, most computer applications that work with images expect them to be rectangular. It is often inconvenient to work with non-rectangular images. The transmission of irregularly shaped images involves also transmitting boundary information. In addition, the BVR compression method might be intended to preprocess an image being sent to another function. If this function (such as one of the standard image compression methods) requires rectangular images, we must find a remedy. Notice that Figures 4.2 to 4.8 all must be padded with white areas to fit into a rectangular array.

While padding the images solves the immediate problem of irregular shape, it increases the image's size with useless data, opposing the compression we wish to achieve (Figure 5.2). If we truncate (crop) the compressed image to fit a smaller rectangle, the decompressed image will no longer be rectangular, having lost complete parts of the image (Figure 5.3). This also is not acceptable. Alternatively, information

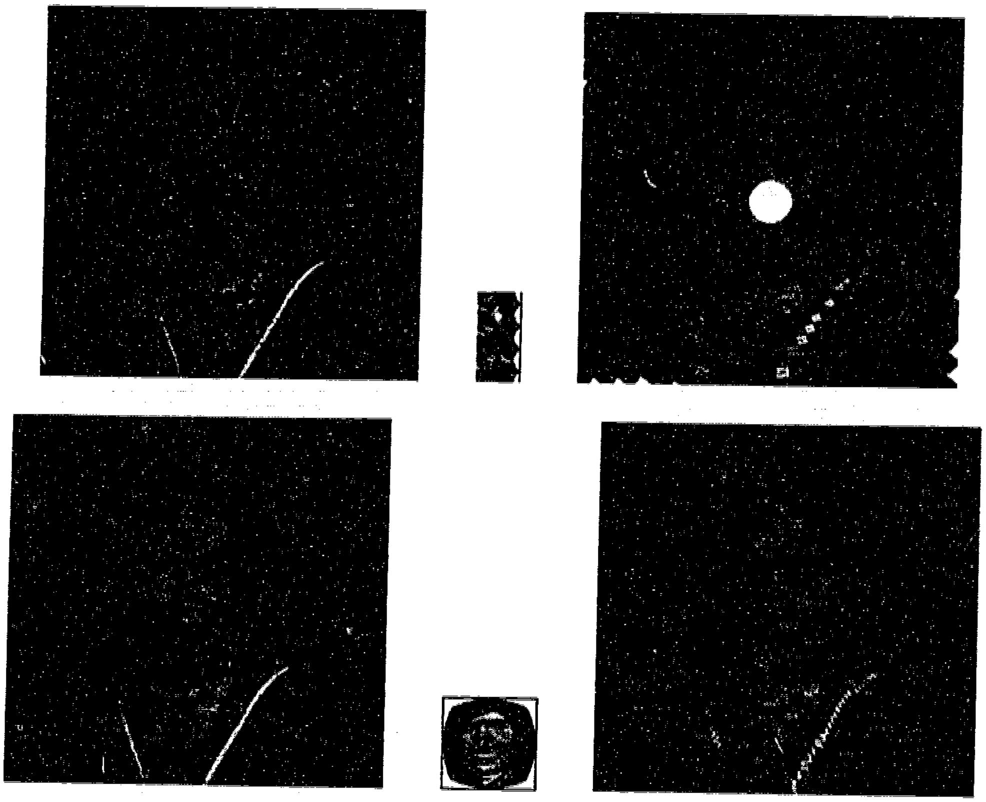


Figure 5.2: Using padding to handle irregularly shaped images.
Top row: Log Polar transform.
Bottom row: BVR transform.
Left to Right: Original, compressed, and decompressed images. Notice the white pixels (unused padding) around the edge of the compressed images.

about the shape of the image (the border) can be coded and transmitted along with the data itself, also increasing the amount of data needed to store or transmit the image. Thus, it is often inconvenient to work with non-rectangular images, and solutions involving padding with useless data or image cropping are often counter productive and not acceptable.

5.1.1 Isotropic and Conformal Mapping Properties

Certain mathematical transformations maintain the properties of isotropic and conformal mapping. Log polar is an example of this type of transform. The benefits of conformal mapping include simpler feature detection, tracking, and other mathematical image processing operations on the image which rely on the consistency of all angles within an image. Isotropic transforms also allow for simpler operations with

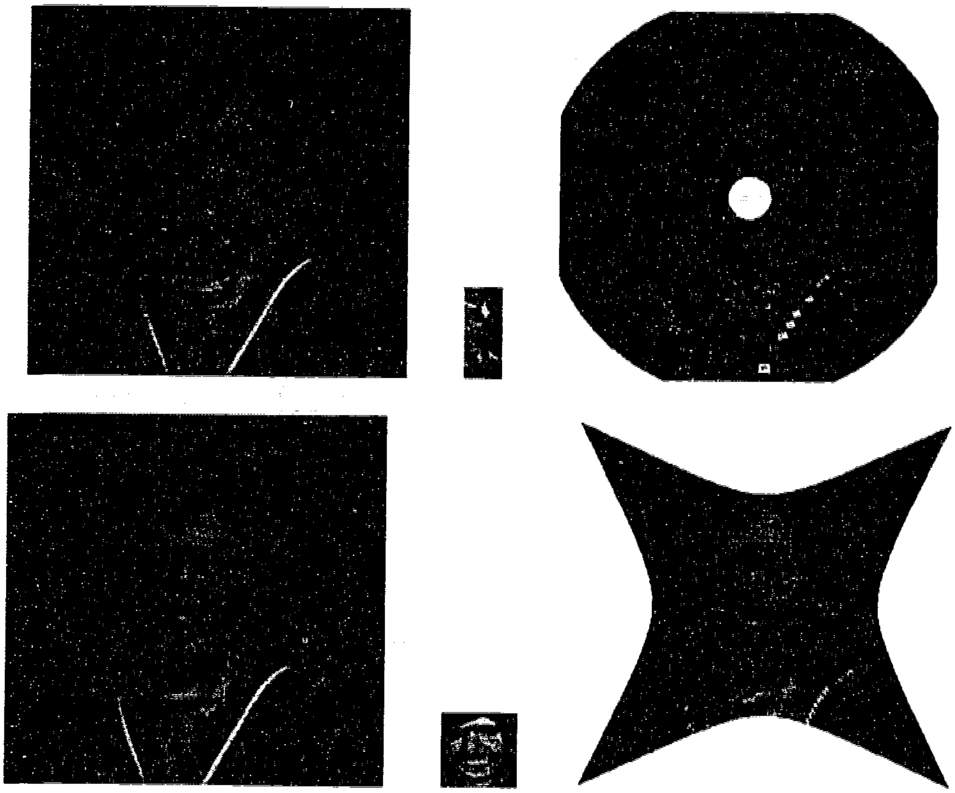


Figure 5.3: Using cropping to handle irregularly shaped images.
Top row: Log Polar transform.
Bottom row: BVR transform.
Left to Right: Original, compressed, and decompressed images. Note the lost information around the edge of the decompressed images.

object rotation and scaling. The human eye is also roughly modelled with isotropic properties.

These benefits, however, generally do not arise in image compression. Compression schemes work on any input image, regardless of its transformation history. Frequency domain based compression methods such as JPEG and MPEG are affected by image discontinuities which some transforms could produce, however.

The SVR transform, as presented, is not isotropic, nor is it a conformal mapping. Tracking simple object paths in the transformation space, therefore, may become more complex. The human eye is also less closely modelled with respect to isotropism. These drawbacks, while they must be identified, are a result of the effort to obtain rectangular compressed images, which is of great benefit. Certainly the ease with which this transform can be used in conjunction with standard compression methods outweighs the increased complexity of certain image processing tasks, especially when these tasks are not even attempted in the subsequent standard compression algorithms.

Anisotropism is the norm within the animal kingdom and therefore should not be considered inherently detrimental. It was noted in Section 4.1.3 that animate visual systems seem to match the possible non-conformal anisotropic topography of their retina to their habitat or some specific function. Biological anisotropism reflects the task and domain of the animal in the same way the anisotropism of SVR reflects the digital image domain and the prevalent necessity of rectangular images and Cartesian coordinate systems. Even in situations where strict isotropism might be considered preferable, the gain of rectangular compressed images far outweighs the minimal anisotropic nature of the SVR transform.

5.1.2 Stretched Variable Resolution Transform Solution

We have seen that with the BVR transform, the problem of irregularly shaped images is magnified when the fovea is not located near the center of the image, or when high α values are used.

One approach to solving the problem of irregularly shaped images used in the SVR algorithm was the use of multiple scaling factors (See Section 4.6).

The entire original image was mapped to a rectangular compressed image, stretching it out at the corners. Isotropic properties were not maintained, but the previously unused padded corners now held useful information.

The SVR method produces situations where particular pixels receive anomalous colour values when the popular bilinear interpolation method is used. The effect, although minimal, appears as spotty *noise*, mainly along the diagonals. Simply searching for the four nearest neighbours of sampled pixels that enclose an unsampled point does not always properly restrict the solution to the bilinear system of equations. The SVR method suffers from this difficulty, as it does not sample points in a regular pattern, requiring lower quality or more complex interpolation methods.

The SVR transform from [89] was promising and definitely worth closer inspection, such as exploring the many possible enhancements and adjustments to its original implementation, including the nearest neighbour search method, interpolation, and scaling factor arrays.

SVR Interpolation Methods

The research and analysis of SVR in [89] used straight pixel sampling — selecting one pixel to be representative of its nearest neighbours. However, improved picture quality may be obtained by other interpolation methods.

The original method simply worked outward from the point, looking for a neighbour. As it searched a line of pixels, it started at one end and worked towards the other end. One change made in our work was simply to start in the center of the line and search towards the ends. This does not give perfect results, and gives more accurate results in only some of the cases but even this small modification improves the signal to noise ratio quite significantly. More recently, a look-up table determining the exact order in which to search neighbouring pixels has been used by the author, but formal comparative analysis have not been done.

Several alternate interpolation methods were implemented and compared, including *bilinear* and *weighted average*. A visual inspection of the images resulting from the search enhancements and added interpolation methods showed a noticeable difference. It is clear from our results that the original methods used in [89] could be significantly improved upon with a little more computation.

Scaling Factor Arrays

Another important aspect of the SVR implementation used in [89] is the data structures used to hold scaling factors. Each point within the image requires a scaling factor, unique to the coordinate angle made with the fovea locations (See Equations 4.26). While this could be calculated separately for each pixel location, it is clear that the full range of scaling factors can be calculated for each angle in advance, and stored in array for quick access. This was previously accomplished using 4 separate arrays conceptually around the perimeter of the image.

This approach was successful for the simplest cases, but involved additional calculations to determine the correct array to access from the four options. An alternate approach implemented by the author generated a single, circular array, as depicted in Figure 5.4. This eliminated the boundary and indexing checks necessary in the original method.

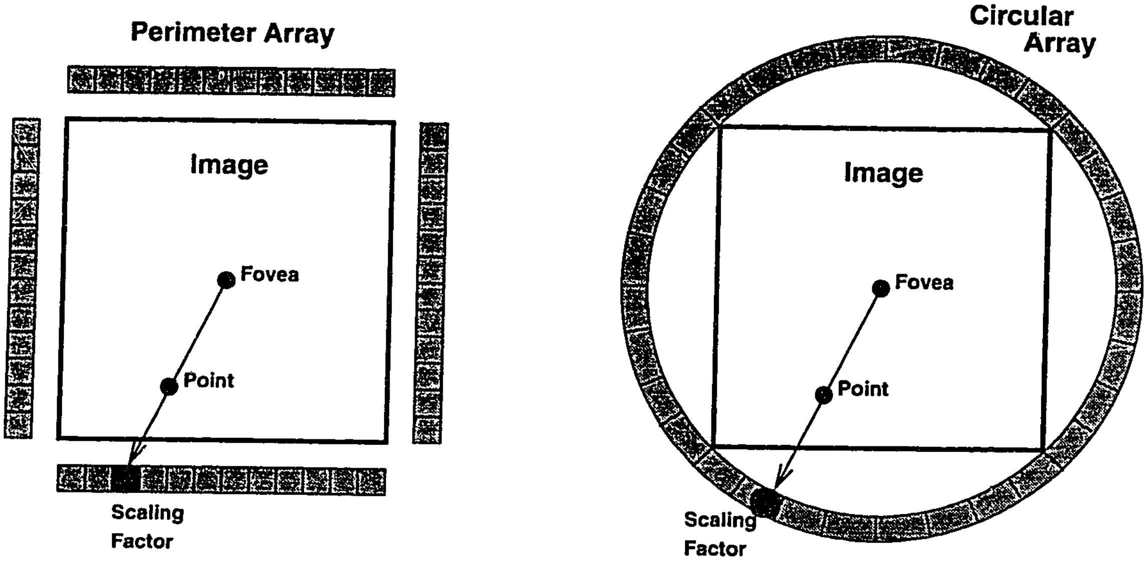


Figure 5.4: Perimeter and circular scaling factor arrays.

In addition to simplifying the step of calculating scaling factors, this enhancement also opened up the possibilities of alternate implementations of multiple foveae. For example, the image segment boundaries required to calculate the scaling factors necessary for *competitive multiple foveae*, (see Section 5.2.1), are not always horizontal or vertical, but could be located at any angle, any length, and any position within the image. The effort required to implement these scaling factors with perimeter arrays is prohibitively great, and vastly more cumbersome than necessary. Circular arrays allow one array per fovea to be calculated and accessed in an identical manner, regardless of the number and position of the foveae.

5.1.3 Cartesian Variable Resolution Transform Solution

The BVR model, as described in Section 4.5, produced irregular shaped compressed images. The SVR method addressed this issue, although in a relatively computationally complex manner. We have also seen how SVR's irregular sampling is problematic for bilinear interpolation. The complexity can be reduced with the *Cartesian Variable Resolution* transform (CVR). The CVR method does not suffer from noise with bilinear interpolation, as all points are sampled in a regular pattern. It also produces continuous rectangular compressed and decompressed images. Like the SVR method, CVR is non-conformal and anisotropic, accurately reflecting the domain of rectangular images in which it is designed to operate. The same reasoning with respect to isotropism and conformal mapping used in Section 5.1.1 applies here as well, as the gain of rectangular compressed images outweighs its possible disadvantages. The CVR method does a reasonable job of maintaining the anthropomorphic foveal properties of the original formulae, although it is simpler to compute.

The SVR approach to dealing with non-rectangular compressed images was to adapt the polar coordinate formulae. The CVR method greatly simplifies the formulae by isolating the vertical and horizontal components without ever operating with polar coordinates. Figure 5.5 demonstrates this approach.

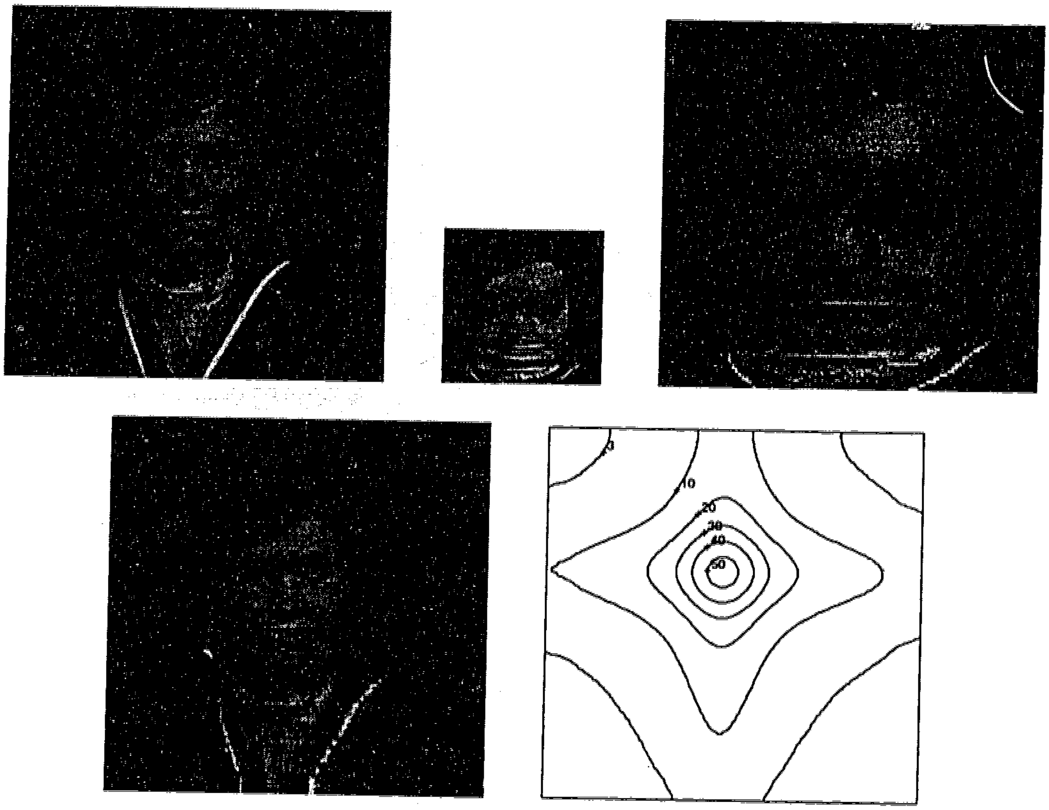


Figure 5.5: Cartesian Variable Resolution transform with fovea on the face. Central clarity is maintained while the edges become blurred.
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map.

For a given image with the fovea located at (x_0, y_0), for every pixel (x, y) in the original image, we define the distance from (x, y) in x and y directions as dx and dy

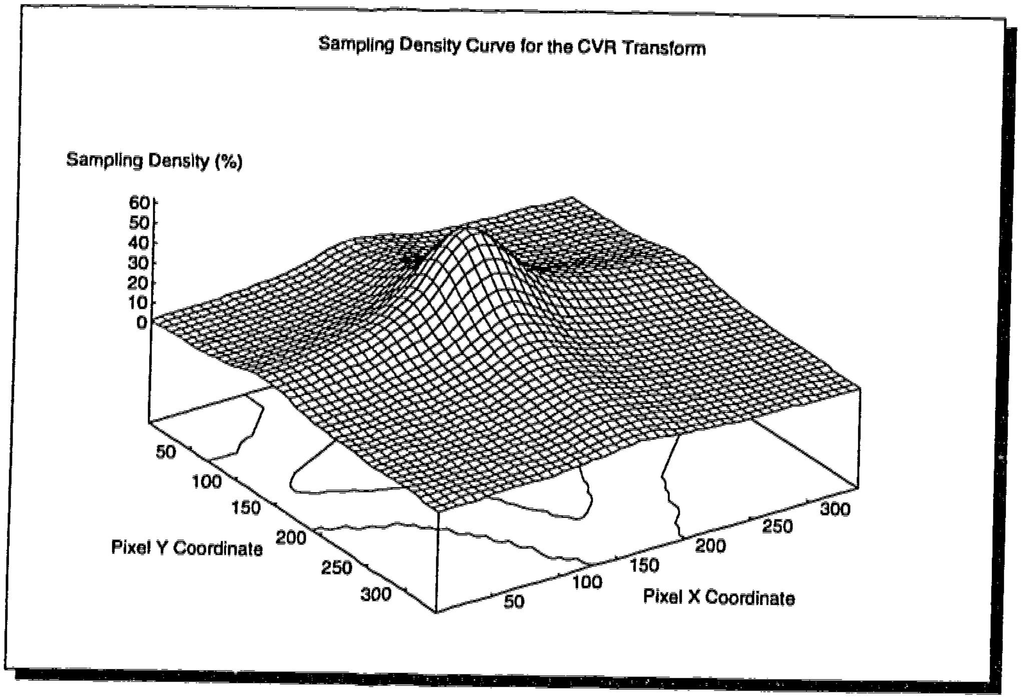


Figure 5.6: Cartesian Variable Resolution transform: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 5.5.

respectively, from the following equations:

$$dx = x - x_0 \tag{5.1}$$

$$dy = y - y_0 \tag{5.2}$$

Therefore, (x, y) is mapped to the point (x', y') where:

$$x' = x_0 + s_x \ln(\alpha\, dx + 1) \tag{5.3}$$

$$y' = y_0 + s_y \ln(\alpha\, dy + 1) \tag{5.4}$$

In other words, here a pixel is moved from dx and dy to dx' and dy' units away from the fovea in x and y directions, where

$$dx' = s_x \ln(\alpha\, dx + 1) \tag{5.5}$$

$$dy' = s_y \ln(\alpha\, dy + 1) \tag{5.6}$$

This transformation can be easily reversed, allowing dx and dy to be defined in terms of dx' and dy':

$$dx = \frac{e^{\frac{dx'}{sx}} - 1}{\alpha} \tag{5.7}$$

$$dy = \frac{e^{\frac{dy'}{s_y}} - 1}{\alpha} \tag{5.8}$$

As in the BVR model, the values s_x and s_y are scaling factors used to control the overall compression ratio. For each dimension we can calculate the maximum distance to the edge of the original image, dx_{max}, dy_{max}; and the maximum distance to the edge of the compressed image, dx'_{max}, dy'_{max}. For each dimension, then, the scaling factors become:

$$s_x = \frac{dx'_{\mathrm{max}}}{\ln(\alpha\, dx_{\mathrm{max}} + 1)} \tag{5.9}$$

$$s_y = \frac{dy'_{\mathrm{max}}}{\ln(\alpha\, dy_{\mathrm{max}} + 1)} \tag{5.10}$$

The CVR method can vary the scaling factors, both horizontally and vertically, depending on the position of the fovea and the compression ratio desired. (The compression ratio will determine the dimensions of the compressed image, setting dx'_{max} and dy'_{max}.)

The BVR transform of a rectangular image is not a rectangular image as illustrated in Figure 5.1. The exact position of the boundaries depends on the VR parameters and the location of the fovea. The CVR method addresses this issue, producing rectangular compressed images regardless of the location of the fovea.

5.2 Multiple Foveae

So far we have concentrated on transforms with one center of attention — one fovea. There may also be circumstances where there is more than one area of interest to the observer. This situation requires multiple foveae, where two or more regions are displayed with higher resolution than the remainder of the image.

When one introduces additional centers of attention, a decision must be made to either reduce the resolution around each fovea to compensate, or retain additional information for each additional fovea, thus reducing the compression ratio. The quality of an image then depends on the relative position of multiple foveae.

As previously noted, natural visual systems contain numerous examples of multiple foveae. Some include a visual streak, while others do not. One can clearly define the mathematical relationship between multiple foveae depending on the desired properties. Two distinct approaches are *cooperative* and *competitive* foveae. The former approach approximates the visual systems which include a visual streak, while the latter approximates divided retina such as *Anableps anableps'*. Both are extensions to the SVR formulae, and both are implemented using the circular scaling factor arrays, presented in Section 5.1.2.

5.2.1 Cooperative Multiple Foveae

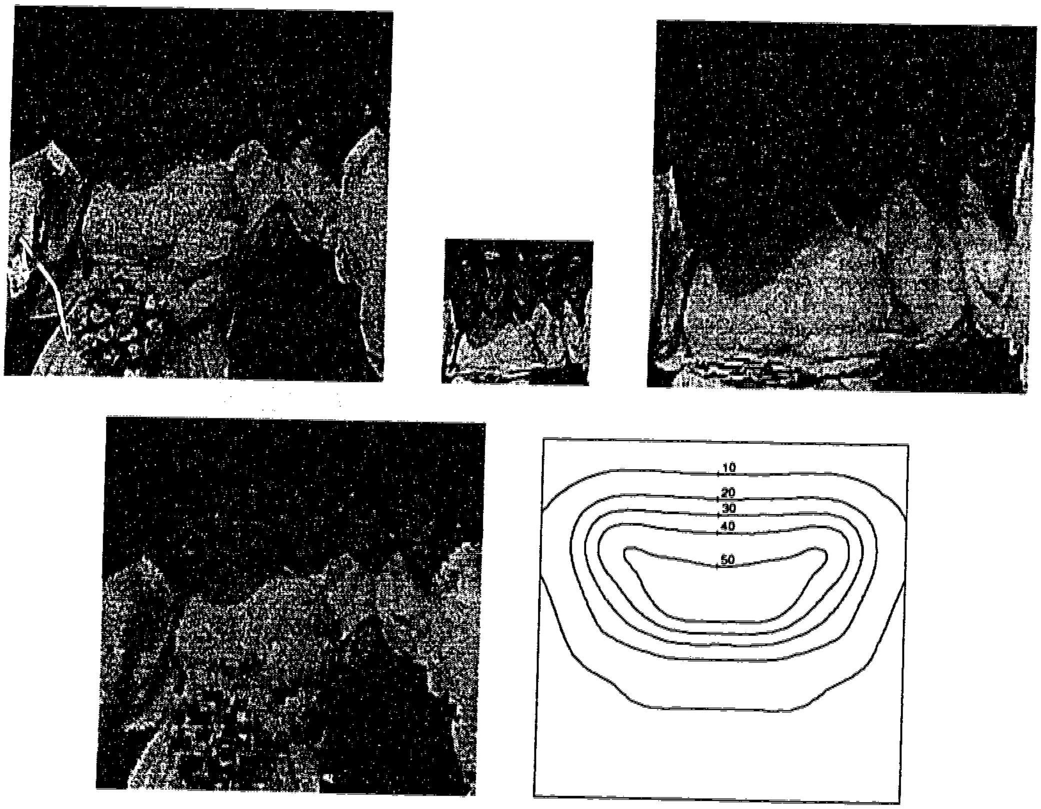


Figure 5.7: Cooperative foveae placed on the outer faces. Note the visual streak clarity across all faces.
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map.

A multifoveate method which works well calculates the possible locations of a point in the transformed image with respect to each fovea separately. The true location is then found by weighting the estimated positions according to the distance of the original point from the fovea. A higher weight is given to the location calculated

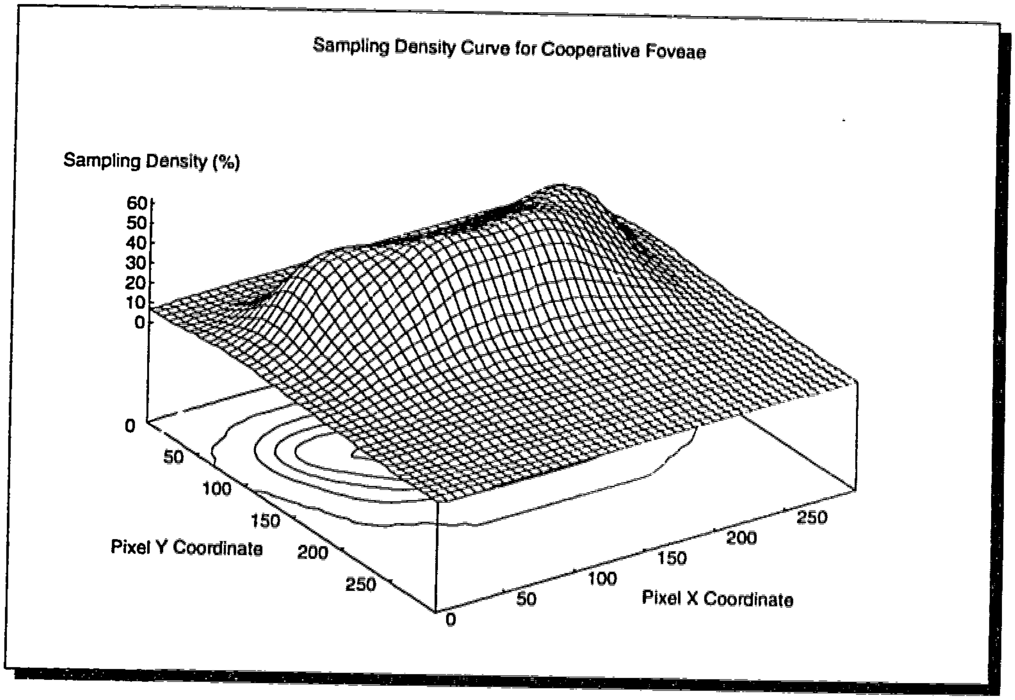


Figure 5.8: Cooperative fovea: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 5.7.

using the closer fovea. The formula for the actual location of each point is:

$$l_{\text{actual}} = \sum_{\forall \text{ foveae } i} (l_i \ \ (1 - d_i/w)) \tag{5.11}$$

where,

$$w = \sum_{\forall \text{ foveae } i} d_i \tag{5.12}$$

and where l_i represents the coordinates of a point calculated using fovea i, and d_i represents the distance to fovea i. The method is termed *cooperative* because all fovea contribute to the calculation of the position of a point in the transformed image.

A unique property of cooperative foveae is the existence of *visual streaks* between them. The area of highest quality in the scene will not only be at the foveae, but also in the area between the foveae. Figure 5.7 generates a banana shaped visual streak to cover all the faces in the image.

If two of several foveae lie on exactly the same location, the transformed image will not always be the same as if only one fovea had existed at that spot. Each contributes to the final positioning of each point in the transformation, no matter where it lies.

Only when all foveae reside at the same location will they exhibit identical behaviour to a single fovea.

5.2.2 Competitive Multiple Foveae

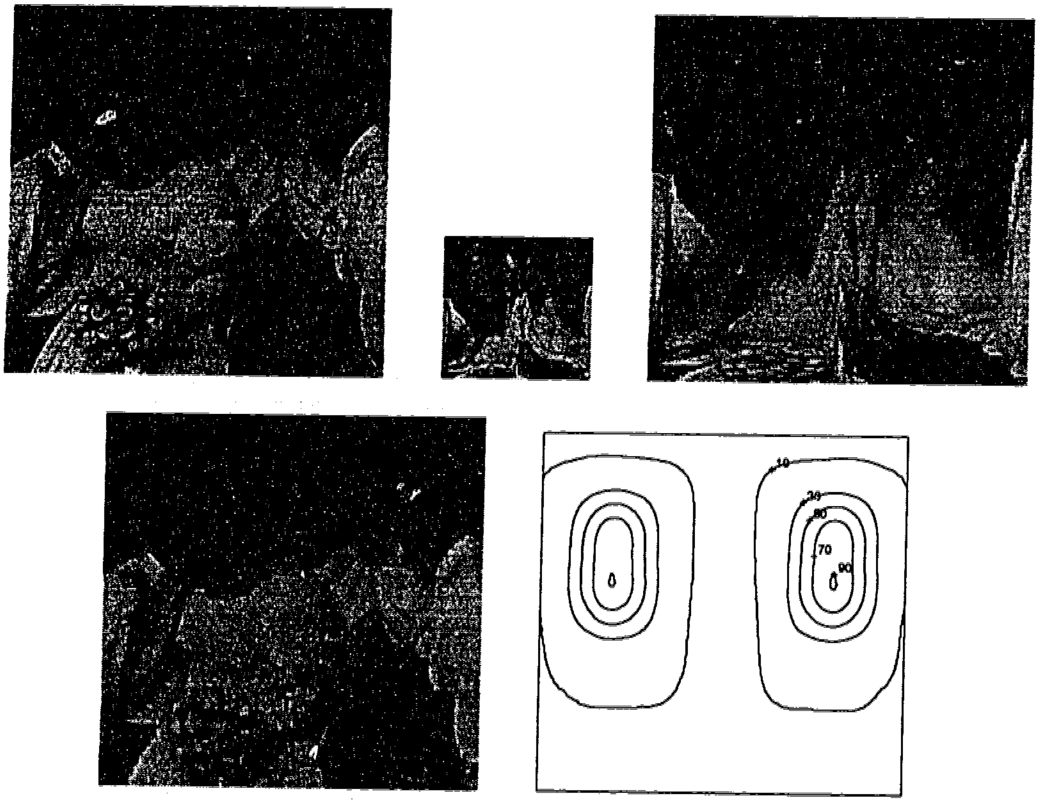


Figure 5.9: Competitive foveae placed on the outer faces. Note the foveal clarity on the outer faces only.
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map.

The definition of *competitive* foveae comes from the fact that all foveae compete to calculate the location of each point in the transformed image. The fovea which is closest to any point in the original image will be the one that determines its transformed position.

$$l_{\text{actual}} = \{l_i : \forall \text{ foveae } j,\ \ j \neq i,\ \ d_i < d_j\} \tag{5.13}$$

Again, l_i represents the coordinates of a point calculated using fovea i, and d_i represents the distance to fovea i.

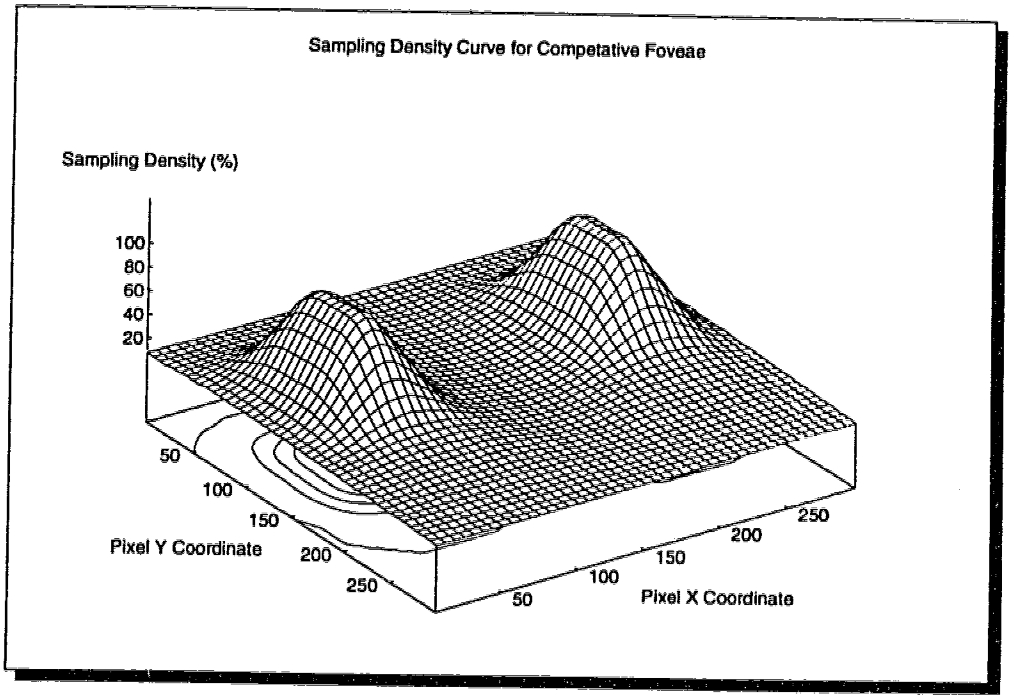


Figure 5.10: Competitive foveae: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 5.9.

Unlike cooperative foveae, when scaling factors are computed to guarantee a rectangular compressed image, the maximum distances used in the formula are not the edges of the image. Instead, a simple Voronoi tessellation is generated around the foveae and the maximum edge of each voronoi area is used. Figure 5.9 demonstrates two foveae covering the outer faces in the image.

The *visual streaks* do not appear between competitive foveae, as the image is essentially broken up into separate regions, each with only one fovea contributing to the compression. This can closely resemble the divided retina of animals with multiple optic paths. There is no noticeable transition between regions, as the boundary is equidistant from the contributing foveae, by its definition.

Here, the quality of an image not only depends on the proximity to a fovea, but also on the relative position of all the foveae. If the total compression ratio is set to a given constant, then the overall quality of the image will increase as two foveae move closer. This effect is more pronounced in the periphery of the scene. If two foveae are centered on the same location, then they act exactly as a single fovea would in that position. Common monofoveate visual systems can therefore be easily modelled with this method.

5.2.3 Weighted Multiple Foveae

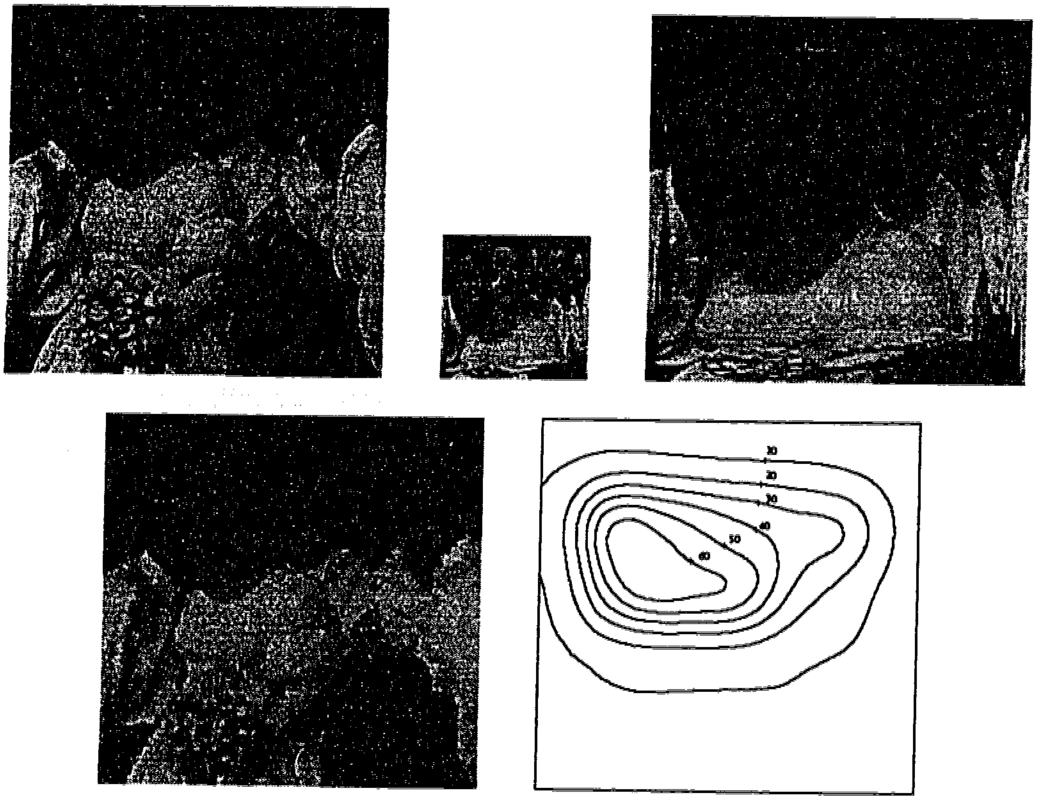


Figure 5.11: Weighted Cooperative foveae placed on the outer faces. The left fovea is weighted three times stronger than the right fovea.
Top row: Original; compressed (to scale); compressed (magnified).
Bottom row: Decompressed; topographical isosampling map.

In either competitive or cooperative multiple foveae systems, the effect of each fovea might not be equal. Most animals that have multiple foveae have one primary and one or more secondary foveae. To achieve this within our current systems, it is necessary to weight the distances to the foveae mathematically, according to their relative dominance. In these cases, the calculation of d_i is no longer simply the distance of a point from fovea i, but rather the distance divided by the weight assigned to that fovea. (A higher weight indicates a stronger fovea.)

$$d_i = \frac{\text{distance to fovea } i}{\text{relative weight of fovea } i} \tag{5.14}$$

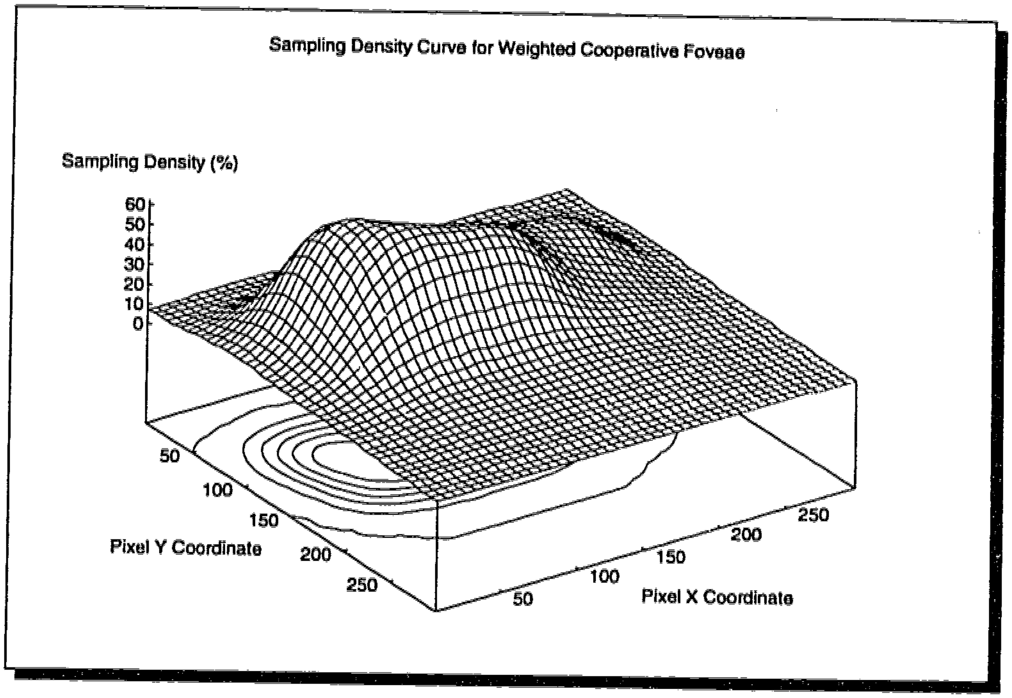


Figure 5.12: Weighted Cooperative foveae: This sampling density map is the three dimensional representation of the topographical isosampling map shown in Figure 5.11.

The calculations using d_i, such as Equation 5.12, may remain unchanged.

Figure 5.11 illustrates two weighted cooperative foveae. A ratio of 3:1 was used with these foveae. The left fovea is clearer than the right, although both foveae maintain a higher resolution along the faces to some degree.

5.3 Visual Streaks

As noted in Section 5.2.1, one of the desired effects of the cooperative SVR foveae method is the modelling of visual streaks. With the proper positioning of multiple foveae in this technique, streaks of various lengths, positions, and strengths can be generated. A larger exponent in the formula that calculates pixel distances from the foveae will create increasingly weaker visual streaks. If a visual streak is located between differently weighted foveae, it will result in a teardrop shape, as desired.

From the transforms described in Chapter 4, only one of the methods not proposed by the authors exhibits some characteristics of a biological visual streak (See Figure 4.6). The Reciprocal Wedge Transform [96, 97] retains clarity along the vertical meridian of the image. A singularity exists along this line, therefore a shift factor or a patch must be included to avoid the loss of central information. While this transform could be rotated to resemble biological horizontal visual streaks more closely, there would still be some aspects difficult to model. Biological systems do not have a discontinuity along the visual streak and usually do not extend across the entire field of view.

5.4 Optic Discs

Section 3.2.4 described some of the variations in optic disc size, shape, and location within the animal kingdom. Some are dorsal to the fovea; some are ventral. In some cases blind spots are circular, while others are elongated. Such variation is motivated by the need to reduce the risk of any object going undetected by falling entirely within the blind spot.

In the same vein, we do not wish to eliminate any part of the images we work with. Figure 5.3 shows how cropping compressed images creates undesirable "blind spots" around the edges of the decompressed images. As well, it was noted that singularities in the transformation formulae of the RWT and Log Polar transforms without shift parameters produce particularly disturbing blind spots directly over the fovea (Figures 4.6 and 4.4). Certainly, this is the most inappropriate location for an optic disc. Solutions to these problems which involve the use of a separate, smaller image to patch the missing areas add complexity.

Sections 5.1.2 and 5.1.3 clearly demonstrate that our transforms do not include blind spots when appropriate shift parameters are used. Like cephalopod eyes, there are no areas in the image to which the transforms are blind.

5.5 Multiple Optic Paths

As described in Section 3.2.5, there are two fundamental ways of modelling a visual system with a divided retina, based on whether or not the different sections of the retina are used concurrently.

The somewhat strange looking "four eyed fish" (*Anableps anableps*) actually has only two eyes, with *divided* retina. It swims with half the eye above the water and half below. The optics of the eye allow a clear projection of images onto both sections of the retina in this situation. In other words, both centers of attention, below and above the water, are of interest at the same time. The competitive SVR foveae method, as described in Section 5.2.2, models the concurrent use of multiple optic paths. Different sections of the image are influenced by a single fovea, or area, at a time.

A simpler transform can be used in situations where an animal alternates attention among multiple optic paths according to the circumstances. Animals such as penguins, dolphins, and kingfishers are capable of clear sight both above and below water, but not at the same time. They achieve this by shifting their focus of attention to the portion of the retina which is receiving a clear image at that time. When their surrounding medium changes, their concentration also changes to the other section of the retina. In this way, they are conceptually using only one part of the retina, or fovea, at a time while ignoring the other. This can be modelled by a transform with a single fovea and a method of changing that fovea's location at will.

While the animals described here typically have two predetermined locations to which they shift their attention, in our computing domain we do not have such restrictions. There is no limit to the number of locations, nor must we predetermine these locations as we move through a sequence of images.

5.5.1 Dynamic Foveae

When used to compress several images in a continuous sequence, the position of the fovea need not be the same for each image. If the positions do differ, the fovea will appear to move when the images are viewed in sequence, and may be called a *dynamic fovea.* Animals will move their eyes or shift their attention among several foveae continuously to track objects of primary importance [19, 42].

One obvious place this capability would be useful in computer vision is in a videophone application. The location of a fovea could follow the person's face, keeping it as the clearest part of the image. As the individual moved through the scene, so, too, would the fovea.

Section 2.4 described important characteristics image sequences possess that could assist in compression. While any of the single image compression schemes, such as

SVR, could be applied to each image in the sequence separately, they would not take advantage of temporal redundancies that exist. As well, SVR in particular includes significant overhead of look-up table calculations based on the location of the fovea (e.g., Equation 4.26). If the location of the fovea changes between each image in the sequence, the number of recalculations required would be unacceptable. This was the approach used in [89]. For a stationary point of interest this approach seemed sufficient, but no research was done on moving foveae. More accurately modelling multiple optic paths or developing an image sequence compression algorithm incorporating the saccadic movements of animate visual systems would necessitate a method of implementing a *dynamic fovea* with little or no overhead calculations between each frame.

To this end, it is worth noting that the CVR method is similar to the SVR method in that variable scaling factors can be used. The CVR method can adjust the scaling factors, both horizontally and vertically, depending on the position of the fovea (See Equations 5.9 and 5.10). The look-up tables (LUT) constructed using this method will then be dependent on the position of the fovea, but the resulting compressed images will be of constant size. If, however, the scaling factors used vertically and horizontally are computed *independent* of the position of the fovea, the compressed images will vary in size, depending on the fovea location. The advantage of the latter method is that to generate rectangular images, the tables need not be recomputed each time the fovea changes location. This is certainly important when implementing dynamic foveae. Figure 5.13 demonstrates how the same LUT is used with several different fovea locations; no additional tables are necessary.

In order to compute the look-up table once, independently of the location of the fovea, we must take into consideration all possible coordinates. By creating a LUT four times as big as the image size, we can align the fovea's coordinates on the image, wherever it is, with the center of the LUT and still have valid LUT entries for each pixel in the image. Figure 5.14 depicts the effect of movement of a fovea from location F1 to F2.

We can also use the fact that the LUT will be symmetric to reduce the necessary calculations. We require the entries in only one quadrant (e.g., top right) as the other quadrants are symmetric over the axes. All the points in the image lie in the first

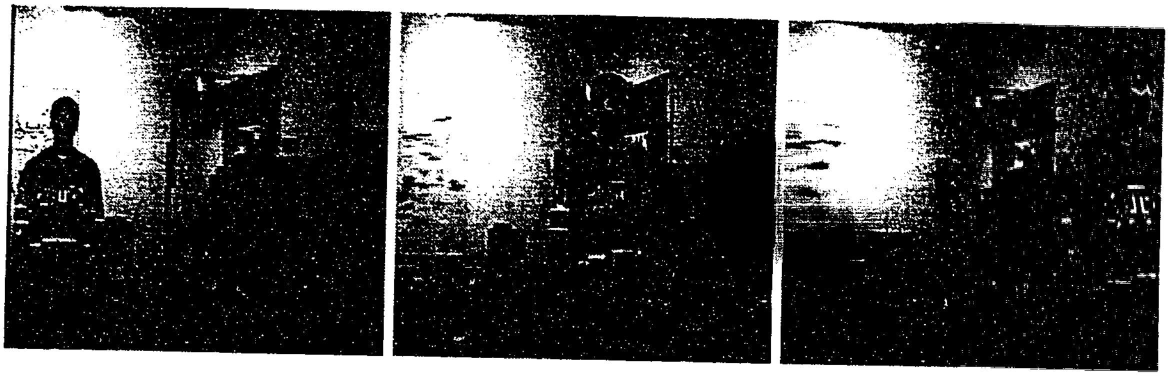


Figure 5.13: The CVR method used with a moving fovea. (Approximately 91% compression.) Notice how the face remains clear as it moves through the scene.

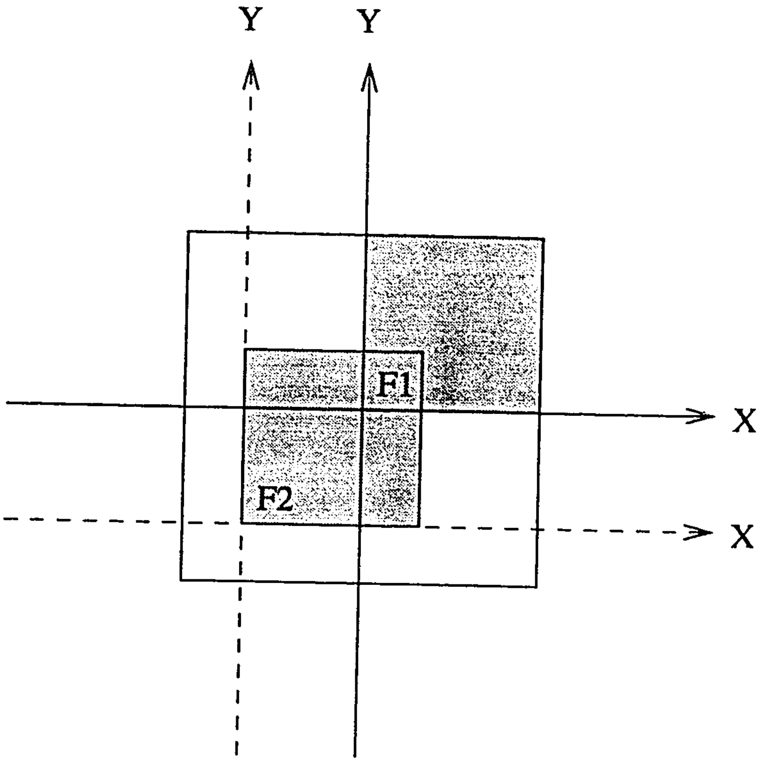


Figure 5.14: Foveal movement within a look-up table.

quadrant when the fovea is in the left bottom corner of the image (fovea location F1). In this case the number of entries in the LUT is equal to the size of the image. For the fovea location F2, only some of the points of the image lie directly in the first quadrant. In this case we have a problem with the points in the other quadrants. Their entries can, however, be obtained by *folding* the image over both axes and referring to their respective positions in the top left corner.

Not only is the CVR transform symmetric, but the sampling pattern is also very regular. This allows an even greater memory savings in addition to the folding. Only one linear array 1/2 the length of the LUT is required to store values necessary for computing all the LUT entries, as both horizontal and vertical components are also symmetrical.

To compress the image, the relative coordinates of the pixels in the image are obtained. Then the transform of the points $(x_{max}, 0)$ and $(0, y_{max})$ can be computed to obtain the dimension of the compressed image.

It is clear that to decompress any of the frames using this implementation of the CVR method, one needs to know only the location of the fovea and to have calculated the look-up tables ahead of time, once, for all frames.

5. ,2 Scene Construction

Sections 3.1.2 and 3.1.4, outline the methods the human visual system uses to assimilate information gathered over time. An inner model of the world, or of a scene, is slowly built up in our minds as our eyes scan our surroundings. This method of scanning is linked intrinsically to the presence and use of a fovea. The fovea is placed over areas in the scene where more detail is required. The eye's movements compensate for the relatively small number of neurons carrying information to the brain.

Image sequences, or videos, contain temporal information, while single images do not. It is possible that the human visual system's ability to gather information over time can be used as a model for VR scene reconstruction. Compression based on the use of a fovea must determine where that fovea should be located. Taking the method of scanning into consideration, the fovea could change its location within the scene over time.

It is obvious, however, that some type of memory is required for humans to remain oriented during rapid eye movements. Each moment of visual sensing is not merely processed in isolation, but with respect to that which has been previously seen. In the same way, it would not be appropriate for an image sequence compression method utilizing a scanning fovea to deal with each image independently. Such an approach would result in the area of clarity within the scene shifting, which would cause confusion if such movements were irregular or over great distances.

It would be more suitable, then, to decompress the image based on a series of previous images. Clearly, the information received from the most recent image would be used in its entirety, but there would be areas in the scene where little detail would be provided. In these areas, the scene could be reconstructed using previous images. One of the recent earlier images may have had a fovea placed on a different area. Therefore, by appealing to these previous images we can gather detailed information in the areas where such information in the current image is lacking. Without testing the implementations, it is unclear as to exactly which method of combining the data over time is the best one. Weighting the data with respect to its age may suffice, as this seems most natural. The older the information, the less useful it may be.

Scanning patterns also would determine the final quality of the system. Random placement of the fovea does not seem intelligent. Placing the fovea predominantly in the area of interest with small saccadic movements, as in humans, would help spread individually sampled pixels, even in the periphery. Infrequent larger saccades to border regions, also modelled after humans, would provide the occasional detail necessary for the periphery to remain clear at all times.

It is obvious that the more time we have to survey our surroundings, the more detailed our internal model of the world will become. In the same way, the more time this computer application is given to view a scene, without it changing, the more accurate the construction will be. The effect of a rapidly changing scene, however, is peripheral inaccuracies. When the above VR methods, such as CVR, are employed, movement within the area indicated as being of interest will be accurately reflected, while ghosting will appear around rapid movement along the periphery. However, for applications such as videophones, such performance may be acceptable.

5.6 A Tool for Modelling Visual Systems

As new types and uses of digital information rise in popularity, research must be directed to specifically tailor digital image processing and compression methods. By examining the specific characteristics of the visual data being used, we can incorporate the biological image compression models previously described to suit a specific purpose. For example, Figure 5.15 shows how certain settings of *cooperative SVR* foveae can closely model the retinal topography of the eagle, if necessary.

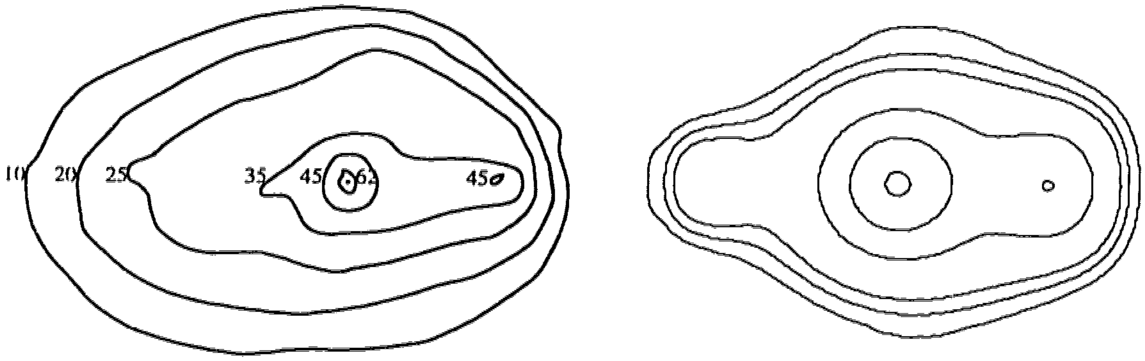
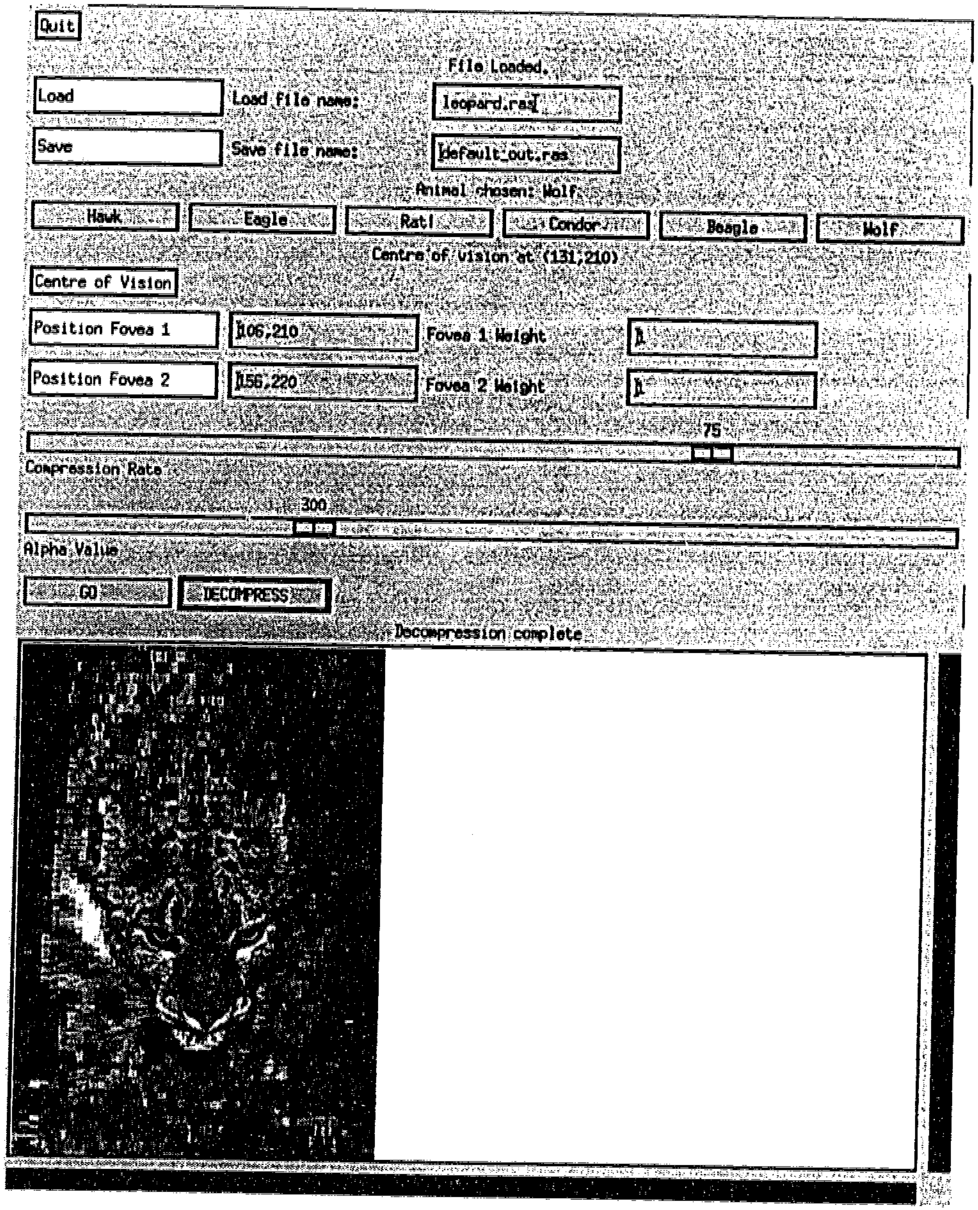


Figure 5.15: **Left:** Ganglion cell isodensity map for the eagle. (See Figure 3.1.) **Right:** Topographical isosampling map with settings approximating that of the eagle.

A tool developed to aid in this task is shown in Figure 5.16. Default settings for modelling the visual systems of several animals can be used, or the user can specify the settings manually. Arbitrary isosampling maps, not taken from biology, can also be modelled and tested easily with this tool.



File Loaded.

Load file name: leopard.ras

Save file name: default_out.ras

Animal chosen: Wolf

Centre of vision at (131,210)

Position Fovea 1: 106,210 Fovea 1 Weight: 1

Position Fovea 2: 156,220 Fovea 2 Weight: 1

75

Compression Rate

300

Alpha Value

Decompression complete

Figure 5.16: An animate retina modelling tool.

Chapter 6

VR Image Compression Analysis Issues

Two important issues must be considered when assessing the merits of VR compression methods. First, Section 1.1 indicated that the VR compression techniques under discussion operate entirely within the spatial domain, allowing them to be used *in addition to* other popular compression methods not focussed primarily in this domain. Comparisons between VR compression methods alone and methods such as JPEG or Wavelet do not do justice to the advantage which can be gained by our research. Used properly, VR should be judged for its merit in assisting other image compression schemes. In this light, VR hybrid compression is a promising development.

Second, the results of image compression are often analyzed using a mathematical quality measure, such as the *signal to noise ratio* or the *mean squared error* metrics. Upon reflection it appears that these measures, while accurate in reflecting human visual quality assessment in many general cases, are not as useful with the class of images with which we are concerned: foveated scenes. A well suited error metric is necessary for mathematical analysis to be meaningful.

6.1 Hybrids

Today, the most popular image compression techniques (e.g., GIF, TIFF, JPEG, Wavelet, Fractal, px64, MPEG) concentrate on eliminating details difficult for the human eye to detect, or on taking advantage of properties such as image continuity. Entropy encoding techniques or *dictionary* tables are then used in the process of compressing the resulting data structures further. These compression schemes treat the entire image uniformly, whereas we have seen that this is not always necessary or desirable [21–23]. The successful use of variable resolution models in animate vision indicates that some situations may support a much higher resolution in the center of the visual field than in the periphery, or other variations. Combining two compression methods which attempt to reduce separate redundancies can outperform either method alone. The most popular compression methods typically work outside the spatial domain, making it difficult to create hybrids between themselves. VR compression does operate within the spatial domain. A hybrid with any other compression method is often possible without any modification *at all* to the second method, if VR is used as a preprocessing function on the image.

For example, JPEG, MPEG, and other schemes have utilized the fact that humans cannot detect changes in chrominance as well as they can detect changes in intensity. While such knowledge is useful, our research has concentrated primarily on the specific characteristics of spatial subsampling. Retinal layout and cortical projections provide us with the basic information in this area. Figures 6.1 and 6.2 show how much quality can be retained at high compression ratios if both of these redundancies are taken into account, together.

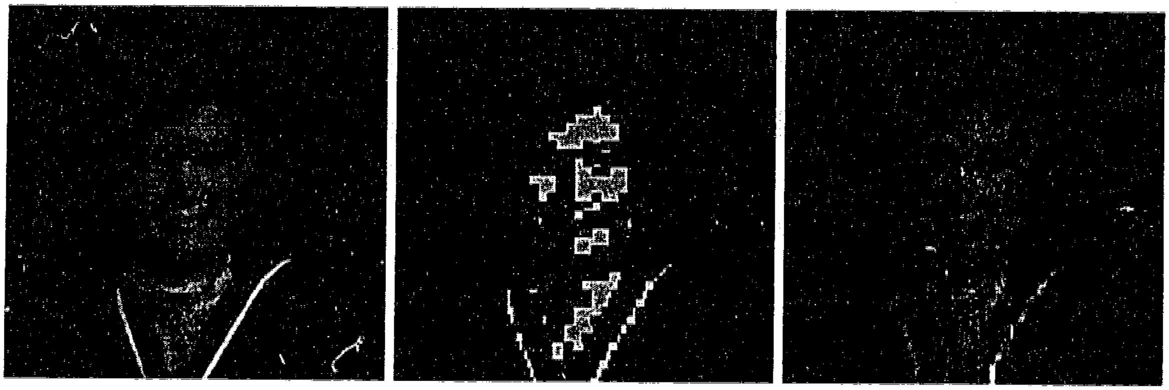


Figure 6.1: SVR/JPEG: Taking advantage of two compression methods which operate on reducing separate data redundancies. **A)** The original image (115738 bytes). **B)** After JPEG compression only (1997 bytes). **C)** After both SVR and JPEG compression (1992 bytes).

With similar reasoning, other combinations between different VR and popular compression methods also achieve improved performance. Figure 6.3 demonstrates

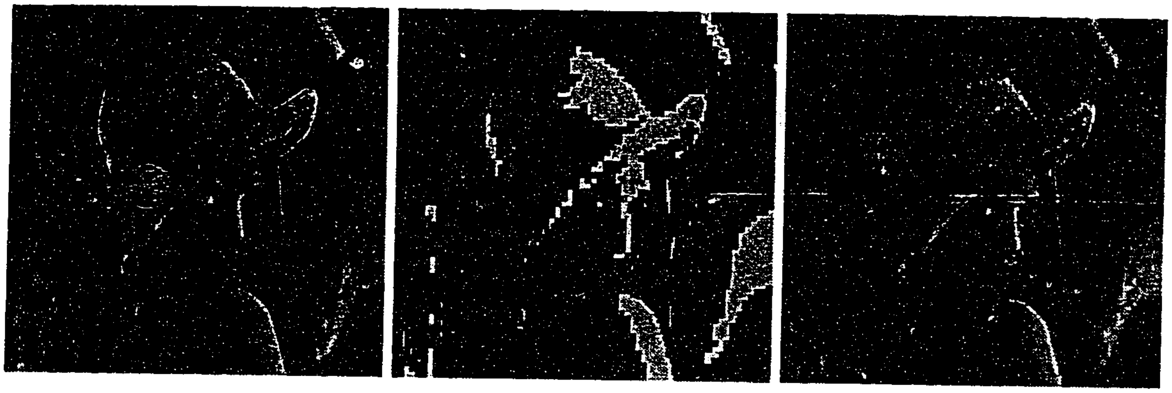


Figure 6.2: A second example of SVR/JPEG hybrid performance. **A)** The original image (262184 bytes). **B)** After JPEG compression only (3798 bytes). **C)** After both SVR and JPEG compression (3776 bytes).

the use of CVR with Fractal compression, while Figure 6.4 gives an example of CVR/Wavelet hybrid compression results. To varying degrees, the clarity of the face is higher in the images compressed with hybrid methods.

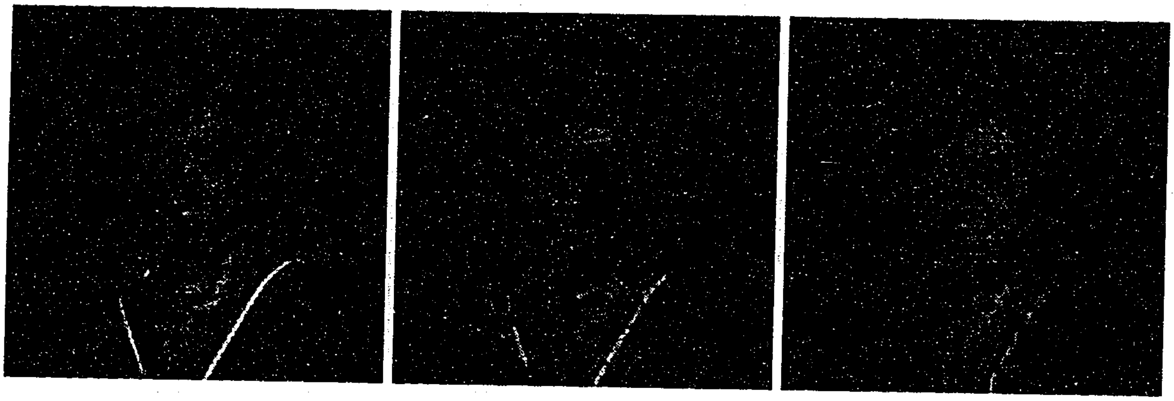


Figure 6.3: An example of CVR/Fractal hybrid performance. **A)** The original image (65717 bytes). **B)** After Wavelet compression only (1290 bytes). **C)** After both CVR and Wavelet compression (1287 bytes).

6.2 Quality

When designing new compression methods, it is essential to assess their relative quality and merit. In this section, different areas of comparison are discussed, and performance measures are evaluated on their suitability to our domain of images.

With reference to the domain of digital images, it is important to note that pixels are discrete elements. Thus, when reducing the size of an image via the VR transform,

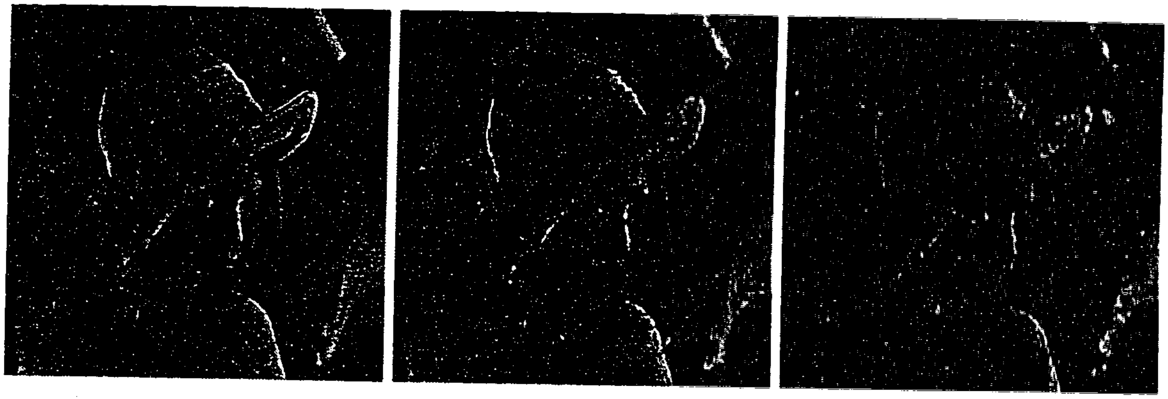


Figure 6.4: An example of CVR/Wavelet hybrid performance. **A)** The original image (262184 bytes). **B)** After Wavelet compression only (2103 bytes). **C)** After both CVR and Wavelet compression (2237 bytes).

one VR pixel may represent several pixels in the original image. Several interpolation methods have been tested for use with decompression. *Bilinear Interpolation* or *Lagrange Interpolation* have proved to be best suited for our purposes and have been used throughout our research.

In the area of image quality, many standard measures attempt to quantify the degradation process objectively. While most research has relied on mathematical standards, clearly there are some cases where such measures do not accurately reflect actual visual assessment. Some images might have only a slight mathematical difference in quality, but the difference in (subjective) visual quality is large. (See Figure 2.1 for an example of several images with similar signal to noise ratios that clearly do not appear similar in quality.) Also, images with a large mathematical difference can sometimes be nearly indistinguishable when visually compared. The problem is that the noise caused by digital compression schemes is non-statistical, so it cannot be determined in the same way as noise caused by analogue systems (e.g., Gaussian noise).

For this reason, the author feels that the best way of evaluating the performance of a digital compression method is by looking at the final decompressed image and comparing it to the original. One must visually compare sample images from several compression schemes in order to reach a satisfactory conclusion. This may be done by the individual researcher or, more rigorously, by a group of test persons [20].

6.2.1 Standard Quality Measures

In attempting to make objective comparisons of lossy image transmission schemes, most research has relied on mathematical standards. While subjective personal visual inspection of the images might be preferable, we often do not have that luxury. Numerical quality measures are more practical. When choosing a measure of quality, there are several standards we must consider [40].

There is a family of quality standards in data compression that measures the error in L^p norms. It is argued in [29] that the flexibility offered by varying p can allow the choice of an L^p norm that matches the Contrast Sensitivity Threshold curve of the human visual system for high frequencies (where one introduces the most error). It is concluded in [29] that $p = 1$ is better than $p = 2$ for natural images.

Other common names for error measures are listed below. Their equations are based on an M by N image, where a pixel in the original image is indicated by $f(x, y)$, and a pixel in the decompressed image is indicated by $f'(x, y)$:

- The mean absolute error (MAE) is also known as the L^1 (or Lp-1) norm.

$$e_{abs} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f'(x,y) - f(x,y)|}{MN} \tag{6.1}$$

- The mean square error (MSE) is also known as the Euclidean error, and the L^2 (or Lp-2) norm.

$$e_{ms} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f'(x,y) - f(x,y))^2 \tag{6.2}$$

- The root mean square error (RMSE) is more sensitive to large errors in single pixels than the MAE measure.

$$e_{rms} = \sqrt{e_{ms}} \tag{6.3}$$

- Normalized Mean Square Error (NMSE)

$$e_{nms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f'(x,y) - f(x,y))^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)^2} \tag{6.4}$$

- Peak Mean Square Error (PMSE)

$$e_{pms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f'(x,y) - f(x,y))^2}{255^2 \, MN} \tag{6.5}$$

- The signal to noise ratio (SNR), as a function of the MSE, is probably the most popular quality measure.

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f'^2(x,y)}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f'(x,y) - f(x,y))^2} \tag{6.6}$$

or

$$SNR_{nms} = -10 \, log_{10}(e_{nms}) \tag{6.7}$$

- The root signal to noise ratio (RSNR) can be calculated from

$$RSNR_{ms} = \sqrt{SNR_{ms}} \tag{6.8}$$

- Peak Signal to Noise Ratio (PNSR)

$$PSNR_{pms} = -10 \, log_{10}(e_{pms}) \tag{6.9}$$

or

$$RSNR_{rms} = \frac{255}{e_{rms}} \tag{6.10}$$

6.2.2 A Foveated Quality Measure: VRMAE

There is one main problem in using any of the standard error measures in evaluating VR based techniques. While these measures weight the pixel errors equally, regardless of location within the image, this does not accurately reflect how foveated scenes are perceived by humans. Figure 6.5 illustrates the maximum spatial resolution of the human retina afforded by ganglion cell density [4, 9, 25, 46]. Notice how the slight naso-temporal asymmetry occurs only in the periphery and does not appear in balanced binocular vision.

For this reason of non-uniform acuity in the human visual system, new quality measures are necessary where centers of focus are known. Our research deals with a restricted domain of digital images, specifically those with identifiable areas of greater interest, therefore, performance measures suitable to foveated scenes must be used.

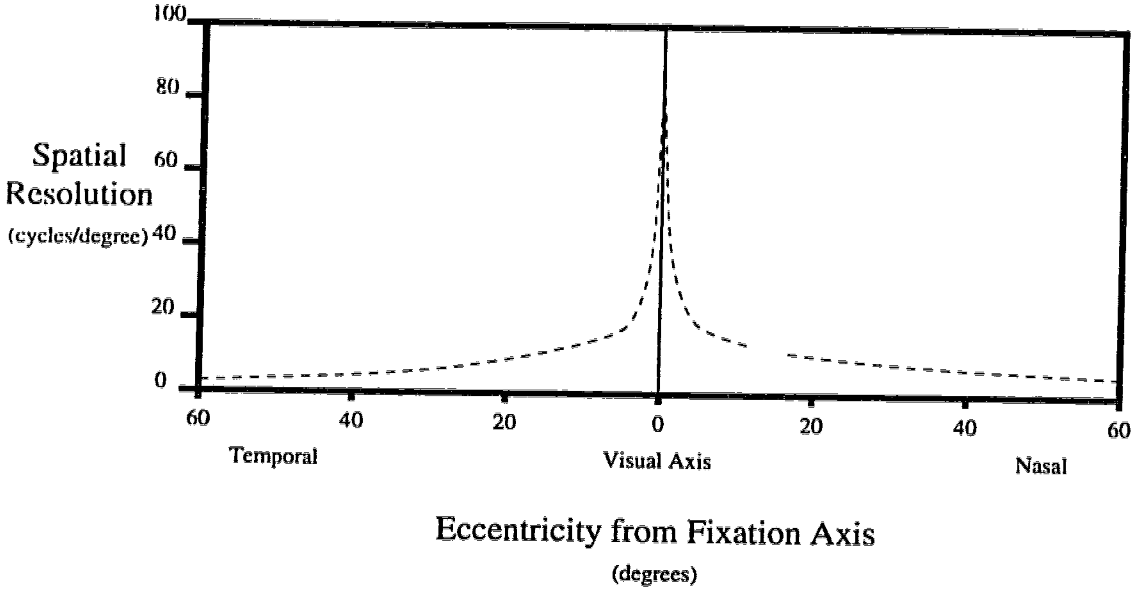


Figure 6.5: Maximum spatial resolution of the human retina afforded by ganglion cell density as a function of retinal eccentricity along the horizontal meridian. Note the nasal blind spot at approximately 15 degrees.

If it is assumed that the foveated error measure we propose is used appropriately, these scenes will have to include foveae locations and strengths which indicate the primary areas of interest within the image. Image quality and pixel accuracy around the centers of interest should have a higher weight than those in the periphery. (If the foveae are not positioned in the correct locations, it is not proper to discredit the VR transform on final image quality, but rather to grade the part of the procedure that located the foveal positions.) Note that for the error measure, there is no need for anisotropic properties as there was with the issue of irregularly shaped images in Section 5.1. True estimation of human visual perception dictates the standard VR equations presented in [16] are sufficient for pixel error weighting. In this light, a foveated quality measure has been developed as an extension of the standard *Mean Absolute Error* (MAE) quality measure.

The error at each pixel is weighted according to its distance from the closest fovea. This weight is in the range of [0, 1], 1 being exactly on a fovea, and 0 being at the maximum distance any pixel is from its nearest fovea. The equations are based on an M by N image, where a pixel in the original image is denoted by $f(x, y)$, and a pixel in the reconstructed image is denoted by $f'(x, y)$:

$$e_{VR} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|f'(x,y)-f(x,y)|w_{xy}}{MN} \tag{6.11}$$

$$w_{xy} = 1 - s\ln(d_{xy}\alpha + 1) \tag{6.12}$$

$$s = \frac{1}{\ln(\alpha d_{\max} + 1)} \tag{6.13}$$

Here d_{xy} is the shortest distance between (x,y) and its nearest fovea, and $d_{\max}$ is the maximum value of d_{xy}, $\forall\ (x,y)$. The value s is a scaling factor, while α controls the distortion effect of the fovea. A high α value indicates a relatively strong foveal region and a low α value means that a fovea is only of slightly greater significance than its periphery. We assume that each fovea location we are supplied with also comes with its corresponding α value. For example, Figure 6.6 shows the weights used in a situation where a single fovea is placed in the center of an image, with an α value of 0.2. Lower height in the graph indicate a lower weight for the pixel. Notice the similarity in shape between Figures 6.5 and 6.6. The steepness of the peak is controlled by the α value.

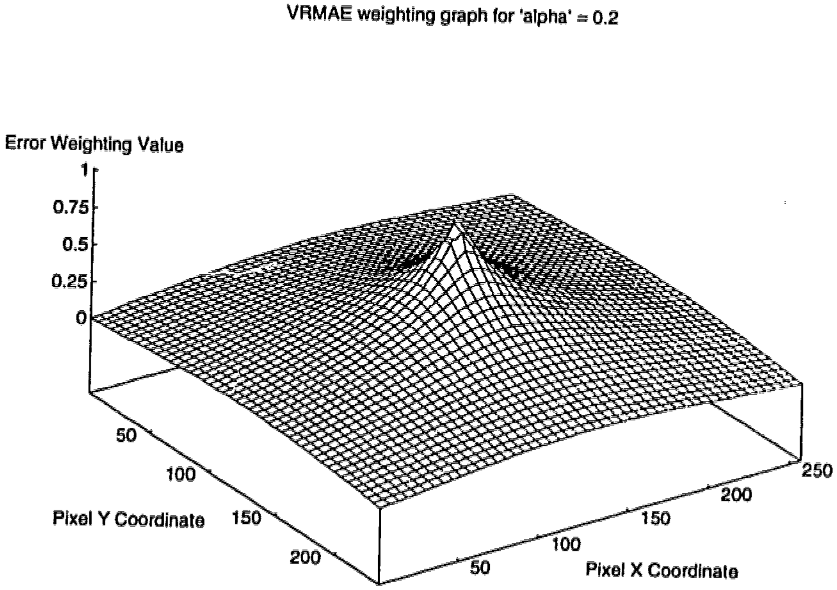


Figure 6.6: A three dimensional representation of VRMAE weights with $\alpha = 0.2$.

6.2.3 Quality Analysis

Experimental statistics comparing the hybrid compression schemes being developed to existing methods are presented in this section. The parameters involved in each experiment are numerous, making comprehensive comparisons infeasible. Factors to be considered when comparing the traditional and novel VR techniques include:

- The exact compression methods used.
- The type of machine used in the compression.
- The size of the test image.
- The content of the test image.
- The number of foveae and their locations.
- The interpolation methods used.
- The error measurements used.
- The timing measurements used.
- The **Q** values used in JPEG and their rate of change.
- The **compression** and α values used in VR and their rate of change.
- The combination of compression parameter values used in hybrid methods, and their rates of change.

We have decided to narrow our analysis to what we feel are acceptable typical values for most of these factors. Our comparisons are between the three SVR, JPEG, and SVR/JPEG methods, run on a Solaris workstation. The test image is a 256x240 pixel standard facial image with a moderately detailed office background (see Figure 6.7). One fovea was specified in the center of the face. Bilinear interpolation was chosen for image reconstruction and the VRMAE metric was used to measure image quality. The speed of the processes are gauged in CPU seconds.

For the series of compression experiments, the JPEG **Q** value was varied between its maximum and minimum values, 100 and 1. For the SVR series, α was maintained

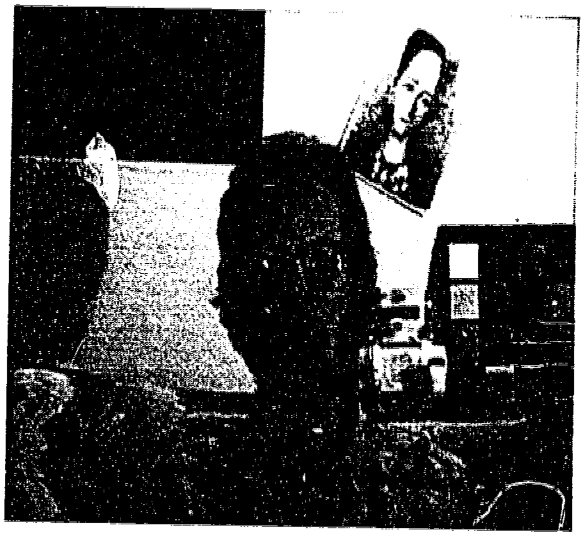


Figure 6.7: The original facial image used in analysis.

at a constant value of 0.2 (see Figure 4.13) while the compression value was varied between 0 and 100. The question of how to vary these parameters simultaneously in testing the SVR/JPEG hybrid method requires closer inspection of the effects of all combinations.

The Variation of SVR/JPEG Hybrid Compression Parameters

When using the SVR/JPEG hybrid compression method, both the JPEG **Q** and the SVR compression value must be set between 0 and 100. (JPEG's parameter operates in the reverse direction from SVR's compression parameter because the **Q** parameter is actually intended to control the image quality - in opposition to compression.) The implications of the combinations of both compression parameter settings become clearer with graphical representation.

Figure 6.8 shows the gradual increase in compression ratio for the lower compression values of both SVR and JPEG and a more rapid increase in compression ratios to the extreme. This rapid increase is more evident along the SVR axis.

Figure 6.9 shows the increase in image error over the same full range of the combination of the two compression parameters. The shape of the surface is generally comparable to Figure 6.8, but one can clearly notice the steeper slope of the surface along the SVR axis at low compression ratios.

In order to choose a combination of SVR/JPEG compression parameters wisely,

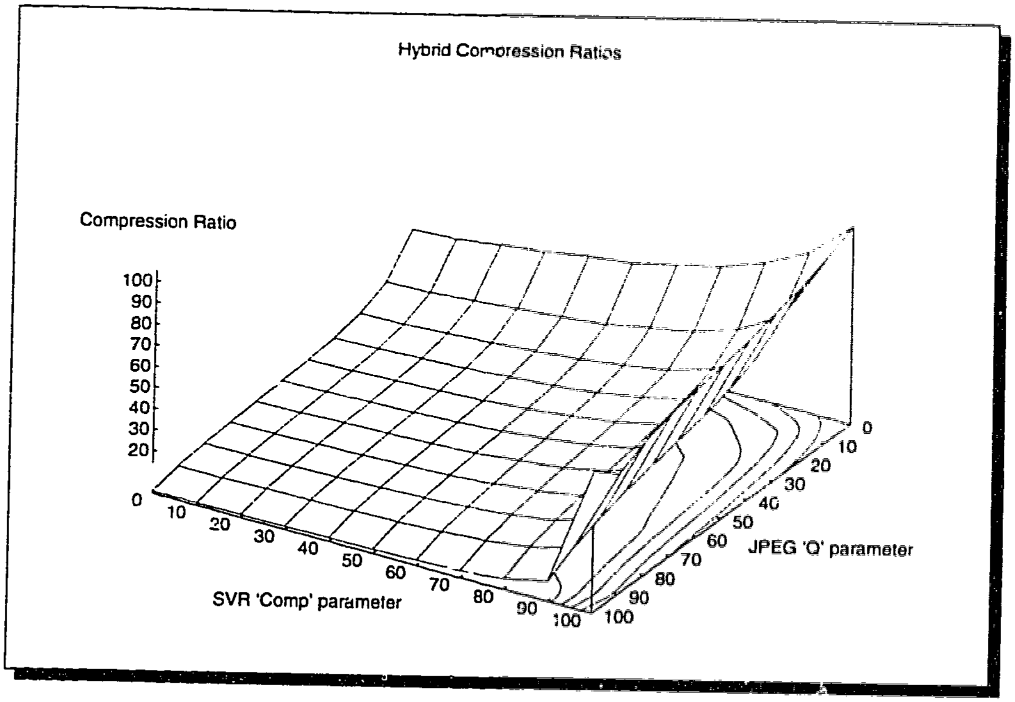


Figure 6.8: A graphical representation of compression ratio versus both SVR/JPEG hybrid compression parameters.

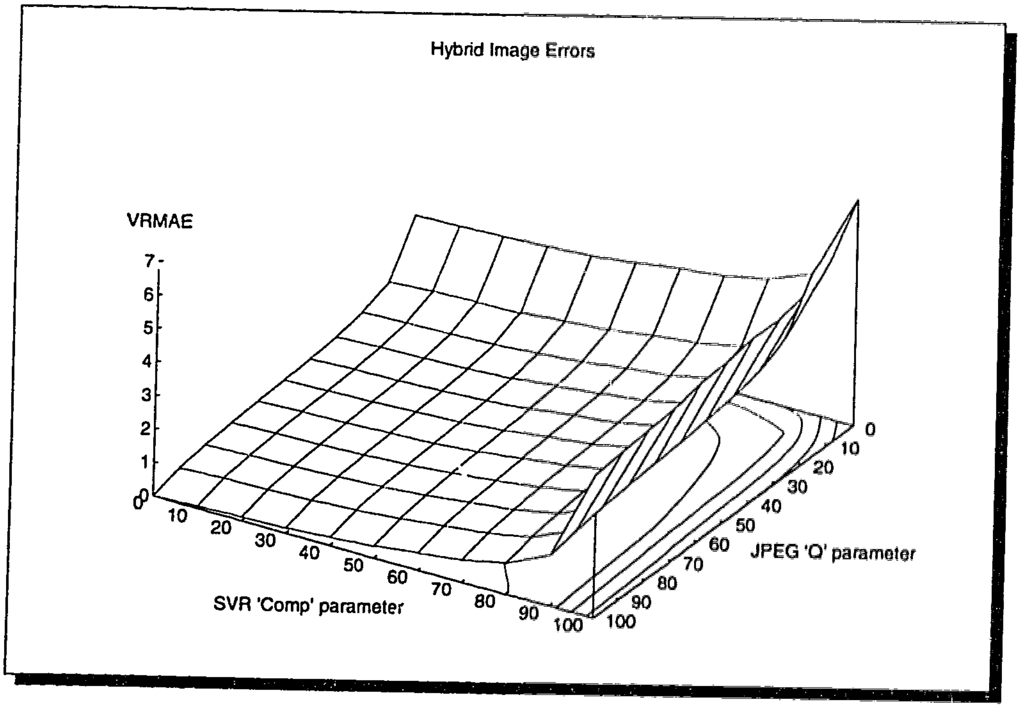


Figure 6.9: A graphical representation of image error (VRMAE) versus both SVR/JPEG hybrid compression parameters.

one would be looking for as little image error as possible with increasing compression ratios. Figure 6.10 combines the information in Figures 6.8 and 6.9 to represent the level of image error (VRMAE) per compression ratio achieved for all parameter settings. In this respect, a lower value is more desirable, as a lower VRMAE error is incurred per compression ratio. An optimal path from the lowest to highest compression settings (from corner to corner in Figure 6.10) would follow the lowest possible height. Note how the large wave in the foreground was anticipated when the steeper slope in that area was noted in Figure 6.8 when compared to Figure 6.9.

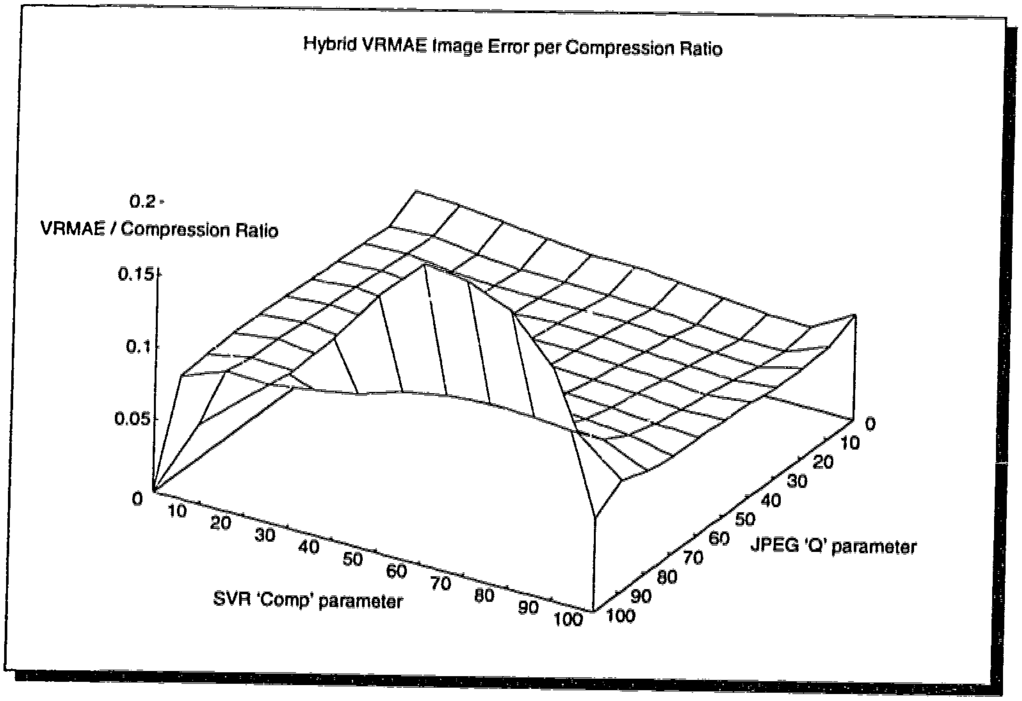


Figure 6.10: A graphical representation of VRMAE per compression ratio versus both SVR/JPEG hybrid compression parameters.

The graphs clearly indicate that a nearly diagonal path of compression parameter assignments approximates the optimal combinations. The SVR compression value must be adjusted at only a slightly accelerated rate in comparison to JPEG's Q value to achieve the best combination over their entire ranges. For our experiments the SVR/JPEG hybrid values were varied in this exact manner, at a constant rate — SVR compression between 0 and 100, and Q between 100 and 10[1].

[1] The far right side of Figure 6.11 also includes the values of the most extreme compression settings of both methods in the hybrid, to give completeness to the graph.

SVR/JPEG Experimental Results

The experiments were completed with the previously outlined parameters. The resulting compression ratios and VRMAE errors can be seen in Figure 6.11. Before further analysis is done on the data as presented, certain clarifications must be made with respect to the quality measure used (VRMAE).

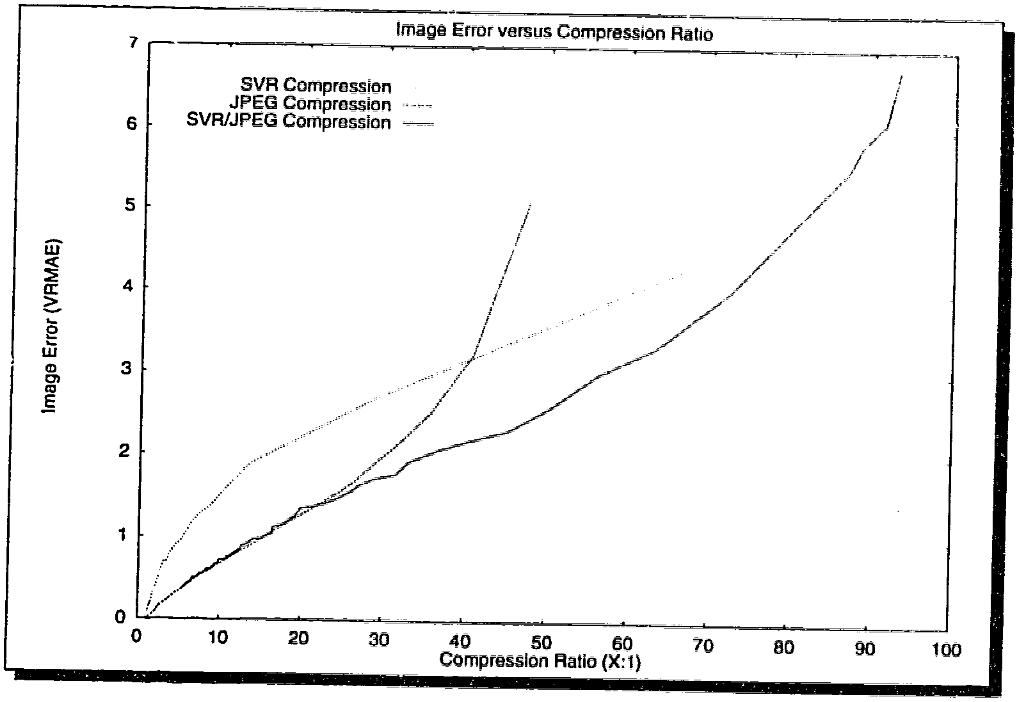


Figure 6.11: Three different image compression techniques, compared in a graph.

First, while error measurements are mathematically correct at each point in the graph, it becomes meaningless to discuss results with a clearly unacceptable error level. For example, Figure 6.12 shows an image with an error value of more than 6. Upon close visual inspection of the images produced in the experiment, it is clear that images with a VRMAE value of greater than 2.5 are well beyond any practical value.

Second, images with extremely low error values are practically indistinguishable from the original, and hence also of little interest in comparison results. The errors that do appear in these images are minute, and relatively small differences in VRMAE values do not indicate a significant visible difference in image quality. Images with a VRMAE number below 1 have extremely high accuracy (See Figure 6.13).

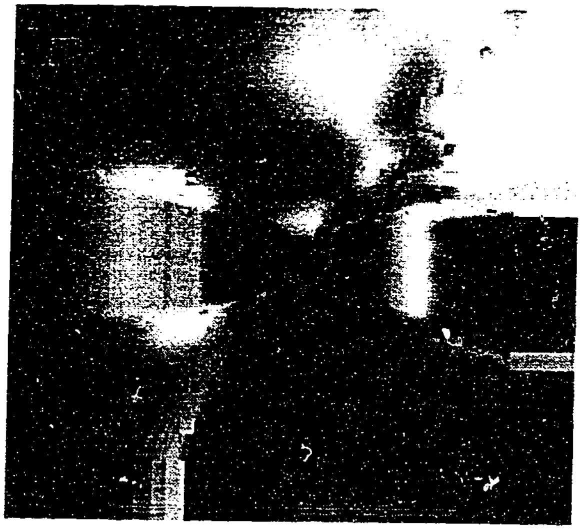


Figure 6.12: Image compressed beyond any reasonable usefulness. (VRMAE = 6.77)

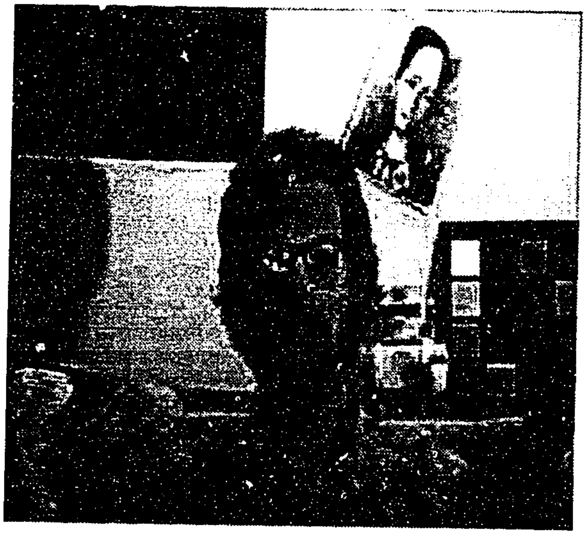


Figure 6.13: Image compressed with few visible errors. (VRMAE = 0.79)

Third, while the VRMAE does detect individual pixel errors, it is not sensitive to particularly disturbing blocky artifacts, present in some JPEG images. As well, while the VRMAE does weight errors occurring near the fovea higher than errors in the periphery, in some cases the VRMAE value is not as close to the subjective visual quality as in other cases. For our experiments, both these situations occur between the moderate VRMAE values of approximately 1 to 1.4. This means that the graph actually shows the SVR/JPEG compression as performing nearly equivalent to JPEG compression, where such does not appear to be the case upon visual inspection.

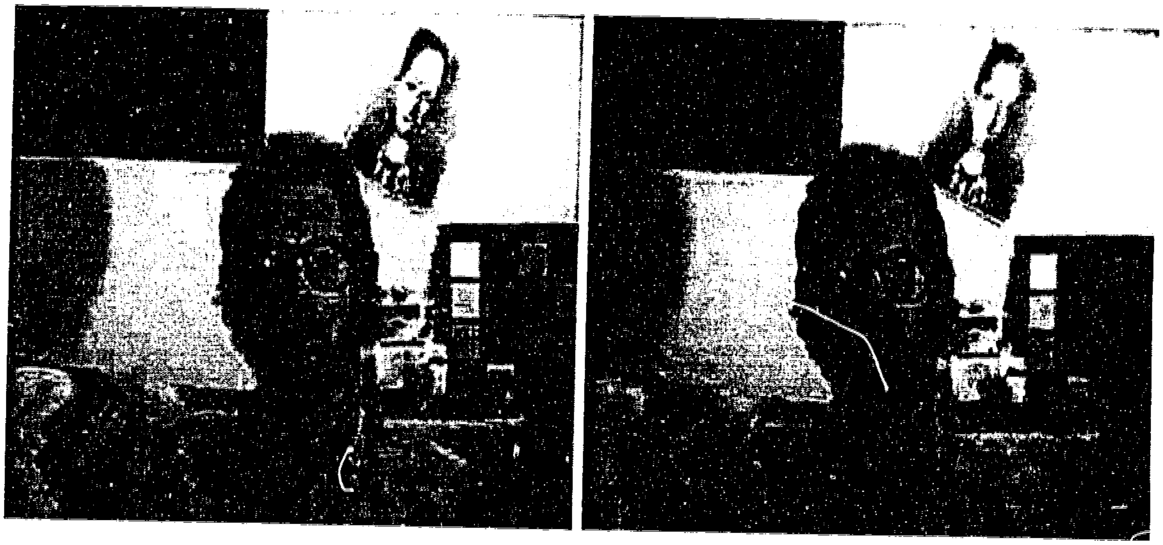


Figure 6.14: Comparing images with similar moderate VRMAE values. (VRMAE approximately 1.2) **Left:** JPEG. **Right:** SVR/JPEG.

Moderate VRMAE Values (see Figure 6.14)

Compression Method	**JPEG**	**SVR/JPEG**	
JPEG **Q** value	11	36	
SVR **compression** value	N/A	63	
SVR α value	N/A	0.2	
Original Image Size (bytes)	61523	61523	
Compressed Image Size (bytes)	3269	3358	
Compression Ratio	18.82:1	18.32:1	
VRMAE (error value)	**1.22**	**1.20**	
Interpolation Method	N/A	**Bilinear**	**Nearest**
SVR CompLUT Time (CPU seconds)	N/A	1.20	1.20
SVR DecompLUT Time	N/A	13.20	11.03
SVR Compression Time	N/A	0.00	0.00
JPEG Compression Time	0.08	0.04	0.04
JPEG Decompression Time	0.06	0.03	0.03
SVR Decompression Time	N/A	0.19	0.04
Total Comp/Decomp Time	**0.14**	**0.26**	**0.11**

Figure 6.15: Statistics for the example in Figure 6.14.

Instead, SVR/JPEG images appear to be of superior quality to JPEG images with similar VRMAE values. For example, Figure 6.14 shows two images with VRMAE values of approximately 1.2, but clearly the greater number of peripheral errors in the SVR/JPEG image are not as disturbing as the foveal errors and blocky artifacts in the JPEG image.

After the limitations of the strictly mathematical analysis of the experimental data have been taken into account, we can begin our meaningful analysis by limiting our graph to the range of practical VRMAE values: 1.2 to 2.5. This range is illustrated in greater detail in Figure 6.16.

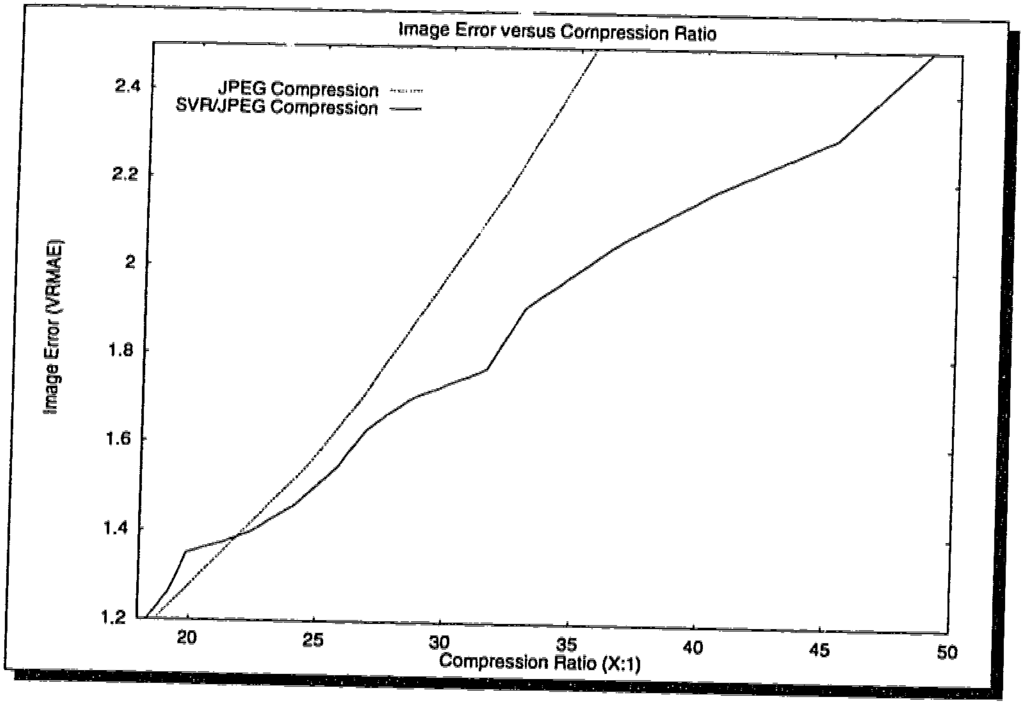


Figure 6.16: JPEG and SVR/JPEG image compression techniques, compared in a graph covering practical value scenarios.

Several observations can now be drawn from the gathered data. In terms of quality, SVR alone is inferior, as it does not even appear on the graph in this compression ratio range. This result was anticipated in Section 6.1. It also appears that the SVR/JPEG method increasingly outperforms JPEG as the compression ratio demanded increases. Some specific examples, shown visually, underline this point.

Figure 6.17 shows JPEG and SVR/JPEG images at approximately the same compression ratio of 35:1. The hybrid compression scheme is clearly superior in this range of high compression. One can conclude that if some application demands high compression, higher image quality can be achieved using the SVR/JPEG hybrid versus JPEG compression.

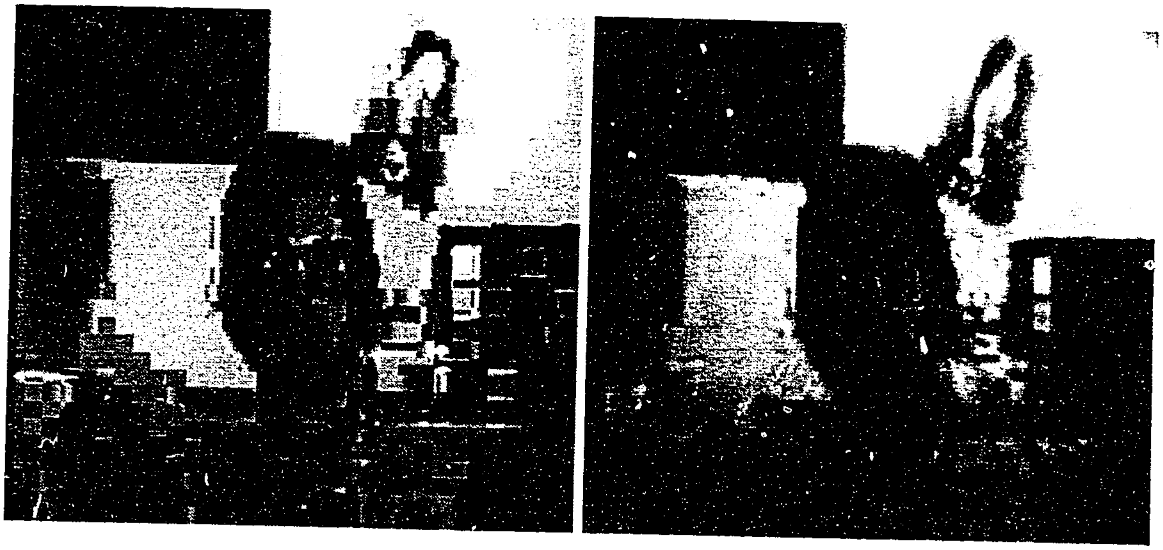


Figure 6.17: Comparing images with similar high compression ratios (35:1). **Left:** JPEG. **Right:** SVR/JPEG.

High Compression (see Figure 6.17)

Compression Method	JPEG	SVR/JPEG	
JPEG **Q** value	3	20	
SVR **compression** value	N/A	85	
SVR α value	N/A	0.2	
Original Image Size (bytes)	61523	61523	
Compressed Image Size (bytes)	1718	1676	
Compression Ratio	**35.81**:1	**36.71**:1	
VRMAE (error value)	2.52	2.06	
Interpolation Method	N/A	**Bilinear**	**Nearest**
SVR CompLUT Time (CPU seconds)	N/A	0.46	0.46
SVR DecompLUT Time	N/A	15.88	12.84
SVR Compression Time	N/A	0.00	0.00
JPEG Compression Time	0.08	0.03	0.03
JPEG Decompression Time	0.05	0.01	0.01
SVR Decompression Time	N/A	0.21	0.03
Total Comp/Decomp Time	**0.13**	**0.25**	**0.07**

Figure 6.18: Statistics for the example in Figure 6.17.

Figure 6.19 shows JPEG and SVR/JPEG images at approximately the same high VRMAE values of 2.2. While both images do appear to be pushing the limit of reasonable image errors, their errors do not appear in the same areas. JPEG has

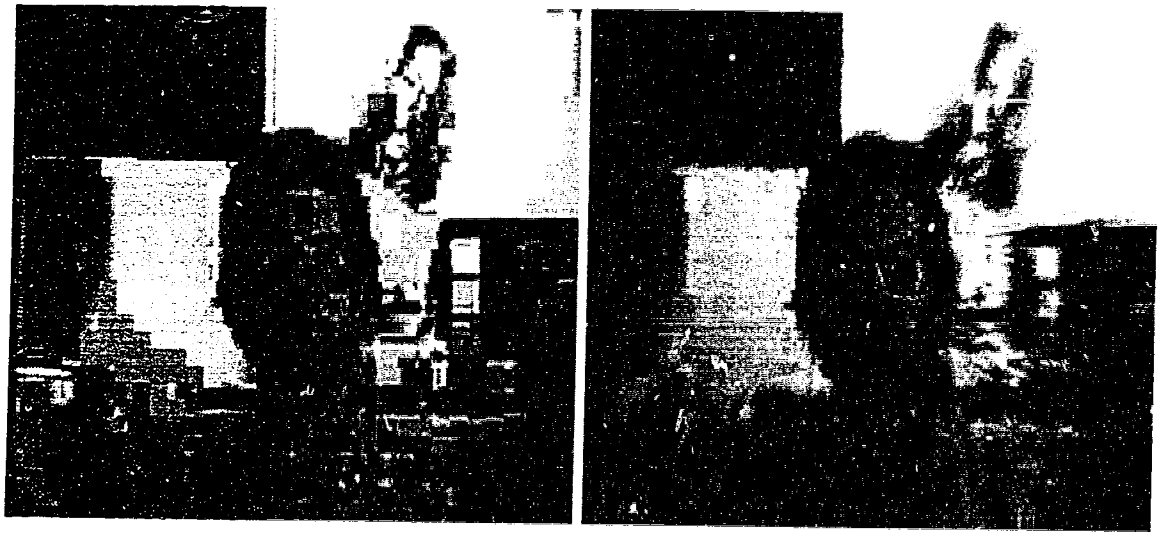


Figure 6.19: Comparing images with similar high VRMAE values (2.2). **Left:** JPEG. **Right:** SVR/JPEG.

High VRMAE Values (see Figure 6.19)

Compression Method	**JPEG**	**SVR/JPEG**	
JPEG **Q** value	4	19	
SVR **compression** value	N/A	87	
SVR α value	N/A	0.2	
Original Image Size (bytes)	61523	61523	
Compressed Image Size (bytes)	1911	1518	
Compression Ratio	32.19:1	40.53:1	
VRMAE (error value)	**2.17**	**2.18**	
Interpolation Method	N/A	**Bilinear**	**Nearest**
SVR CompLUT Time (CPU seconds)	N/A	0.44	0.44
SVR DecompLUT Time	N/A	17.24	13.02
SVR Compression Time	N/A	0.00	0.00
JPEG Compression Time	0.08	0.02	0.02
JPEG Decompression Time	0.05	0.01	0.01
SVR Decompression Time	N/A	0.22	0.03
Total Comp/Decomp Time	**0.13**	**0.25**	**0.06**

Figure 6.20: Statistics for the example in Figure 6.19.

maintained a higher image quality around the periphery at the expense of the fovea. The opposite is true for the SVR/JPEG method. A clearer face (with much clearer eyes and nose) is achieved at the cost of most of the peripheral details. Compared to JPEG, the hybrid method has also reduced the size of the compressed image by almost 1/4 - increasing the compression ratio from 32:1 to 41:1. One can conclude that if an application is operated at the maximum acceptable image error level, the SVR/JPEG hybrid can achieve significantly better compression ratios than JPEG compression.

6.3 Speed

When measuring the time a compression or decompression method takes in completing a task, it is often beneficial to time different parts of the process separately. Some computations must be made separately for each image processed, while others must be made only once, if a series of images are being handled. Our VR techniques require the use of a LUT, both for compression and decompression. These LUTs only need to be computed once for a series of images, and are therefore timed separately.

The optimization of code also plays a large role in its execution time. The standard JPEG code used has been in existence for some time and has presumably been optimized to some detail. The SVR code used in our research, however, has not been optimized for timing performance. There are many areas where the code has not been designed for efficiency, particularly in the implementation of interpolation methods.

The interpolation methods required by SVR decompression also vary in execution time. Bilinear interpolation is more complex than Nearest Neighbour interpolation, taking more time to execute. Therefore one must take into account what interpolation method is being used when analyzing SVR timing statistics.

6.3.1 Speed Analysis

The VR transforms are extremely fast. Typical compression times (measured in CPU seconds on a Solaris workstation) for the compression algorithm on images used as examples in this work were in the range of 0.00 to 0.01 seconds — often too small to even register any time at all. In comparison, JPEG is a relatively slow algorithm. Typical JPEG compression times fell in the range of 0.5 to 1.3 seconds. (Other implementations may be faster, but often require additional special hardware.)

Because the VR compression schemes result in an intermediate image still in the spatial domain, other algorithms such as JPEG can be run on these as well, compressing the image even further. In addition to the speed of VR itself, when used in a hybrid it can substantially reduce the amount of time required for secondary steps. (A typical value: compression with SVR by 70% can reduce the total processing time taken by more than 1/2.)

Figures 6.15, 6.18, and 6.20, provide us with example execution times for several JPEG and CVR/JPEG compression scenarios. Figure 6.21 graphically illustrates the timing values contained in Figure 6.18.

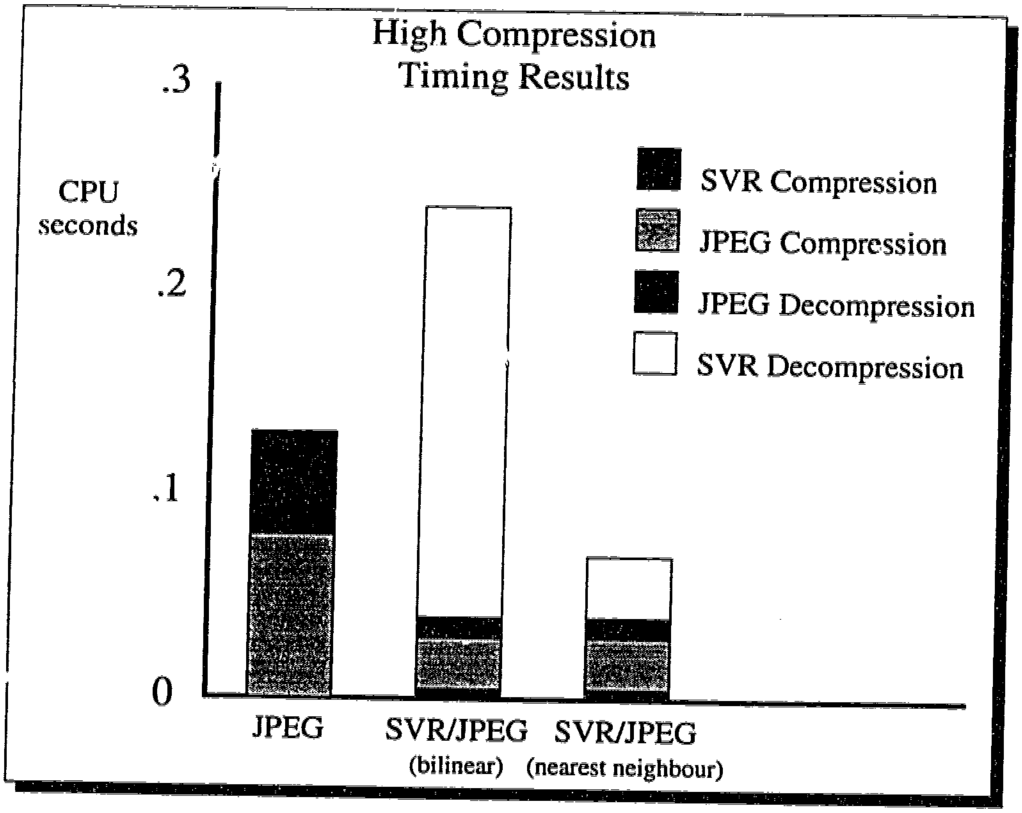


Figure 6.21: Timing results for the high compression example of Figure 6.18.

The first conclusion one can draw from the timing data is that the time required to build the LUTs for SVR compression are much larger than any of the times for other computations. If a situation would require frequent rebuilding of LUTs, SVR/JPEG compression is much slower than JPEG compression. However, if we ignore the LUT computation time, as we can when compressing a large number of images at once, or compressing an image sequence, JPEG may not always be the fastest method.

If compression time were the only factor, the SVR/JPEG algorithm is clearly superior in these circumstances. If decompression is also important, the time taken by the interpolation method largely determines the speed. Figure 6.21 clearly shows how using the simpler nearest neighbour method of interpolation instead of the more complex bilinear method reduces the necessary decompression time. A tradeoff between image quality and decompression time exists when considering interpolation methods. Application designers must decide where priorities lie when deciding on which compression method, hybrid ratio, and interpolation method to use.

Chapter 7

Applications

As observed previously, multimedia systems often rely heavily on the use of digital images. Image compression is of significant concern. VR compression techniques are suitable for any scene where perfect image quality is not required and points of primary interest can be determined. VR methods do not seem appropriate for unpredictable, rapidly changing scenes (like unstructured live television).

In this chapter, several practical applications will be outlined. Section 7.1 illustrates how VR technology can benefit archival systems. Section 7.2 describes a video compression and decompression tool using a hybrid compression method. In Section 7.3, a videophone utilizing VR compression is proposed. A general VR teleconferencing system is described in Section 7.4. Finally, Section 7.5 describes an application of foveal based data prioritization for ATM network transmissions and an implemented simulation and Section 7.6 outlines some generalized advantages of foveal prioritization.

7.1 Mug Shot Database: *FaceBase*

Many organizations maintain an image archival system consisting of people's faces (*mug shots*). Some are used for security reasons, while others are merely kept as part of personnel files. These mug shots must be stored in a compressed format so as not to consume excessive resources. While the background may not be necessary, merely storing the information on the face may not be an aesthetically feasible solution. Variable resolution compression allows the entire image to be stored. The face remains of high quality, while the lower quality periphery still provides the overall context.

7.1.1 Hybrids: SVR/PGM and SVR/JPEG

In the process of developing a complete mug shot database system called *FaceBase*, new encoding standards utilizing VR techniques were developed. Hybrids between SVR and both *Portable Graymap* (PGM) and JPEG file formats were designed and implemented.

A complete library of file access functions for both hybrids was written. Implemented in C code, simple function calls both read and write the desired file format. The SVR/PGM functions were written from scratch, while the SVR/JPEG functions incorporated some software supplied by the official Independent JPEG Group[1].

These new formats tried to remain as close to the standard format as possible, so as to work seamlessly with all existing image processing software. This was in fact accomplished by encoding the required SVR parameters within the annotation features already present in the standard. Within bounding comment markers, the parameters necessary for the SVR part of the hybrid are:

- compression value,
- α value,
- decompressed size: rows, columns,
- exponent power (exponent for visual streak strength),
- number of foveae,
- position of each fovea: x, y,
- SVR method (cooperative or competitive).

Appendix B contains an example of the exact file header syntax of the SVR/PGM format (called *VR3*). The SVR/JPEG header format is identical in structure, but utilizes the JPEG **COM** markers in binary, which do not lend themselves well to being viewed directly.

[1] While the actual compression parts of the JPEG code was not altered, the interface was extensively modified to, among other things, correctly preserve COM markers according to official JPEG specifications.

7.1.2 An Image Database Interface Tool

A complete Image Database Tool was implemented, using the SVR technologies[2]. Figure 7.1 shows the main window of the application and Figure 7.2 shows several of the other control windows.

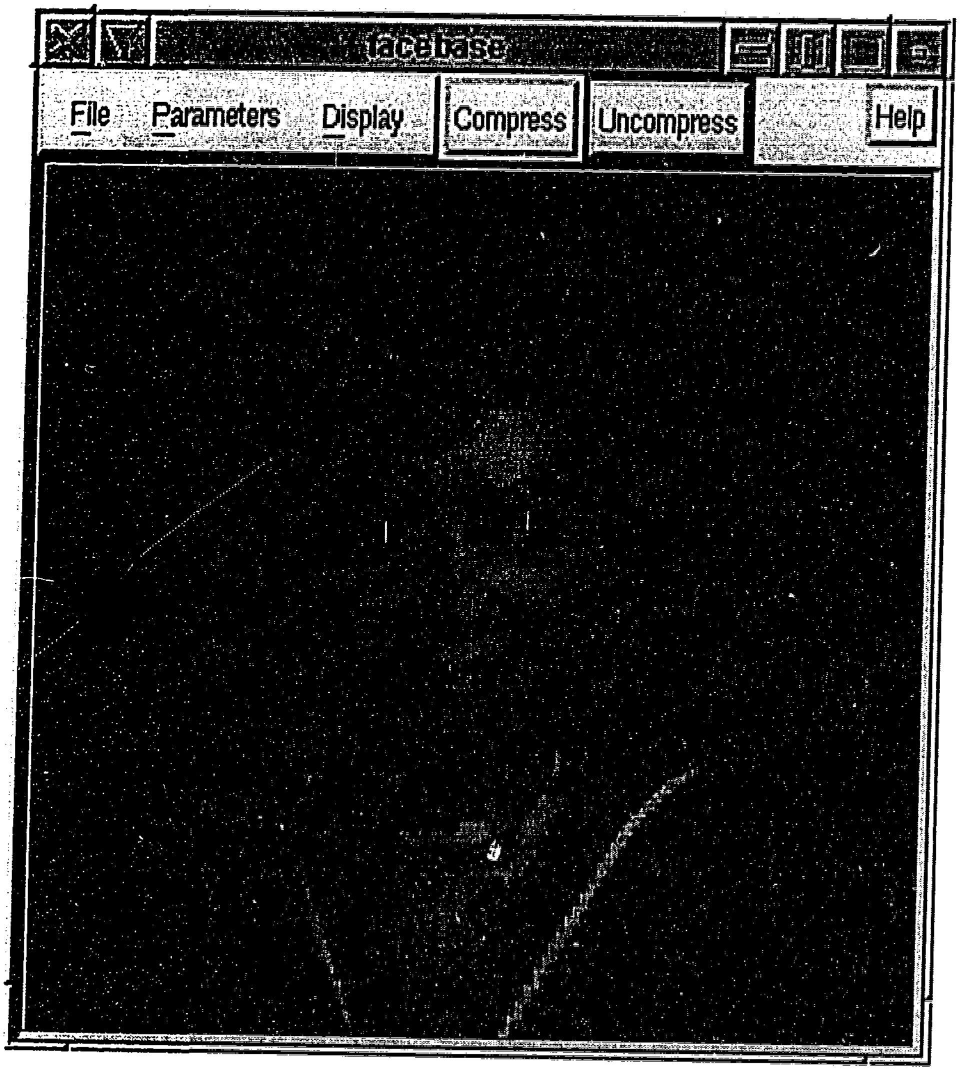
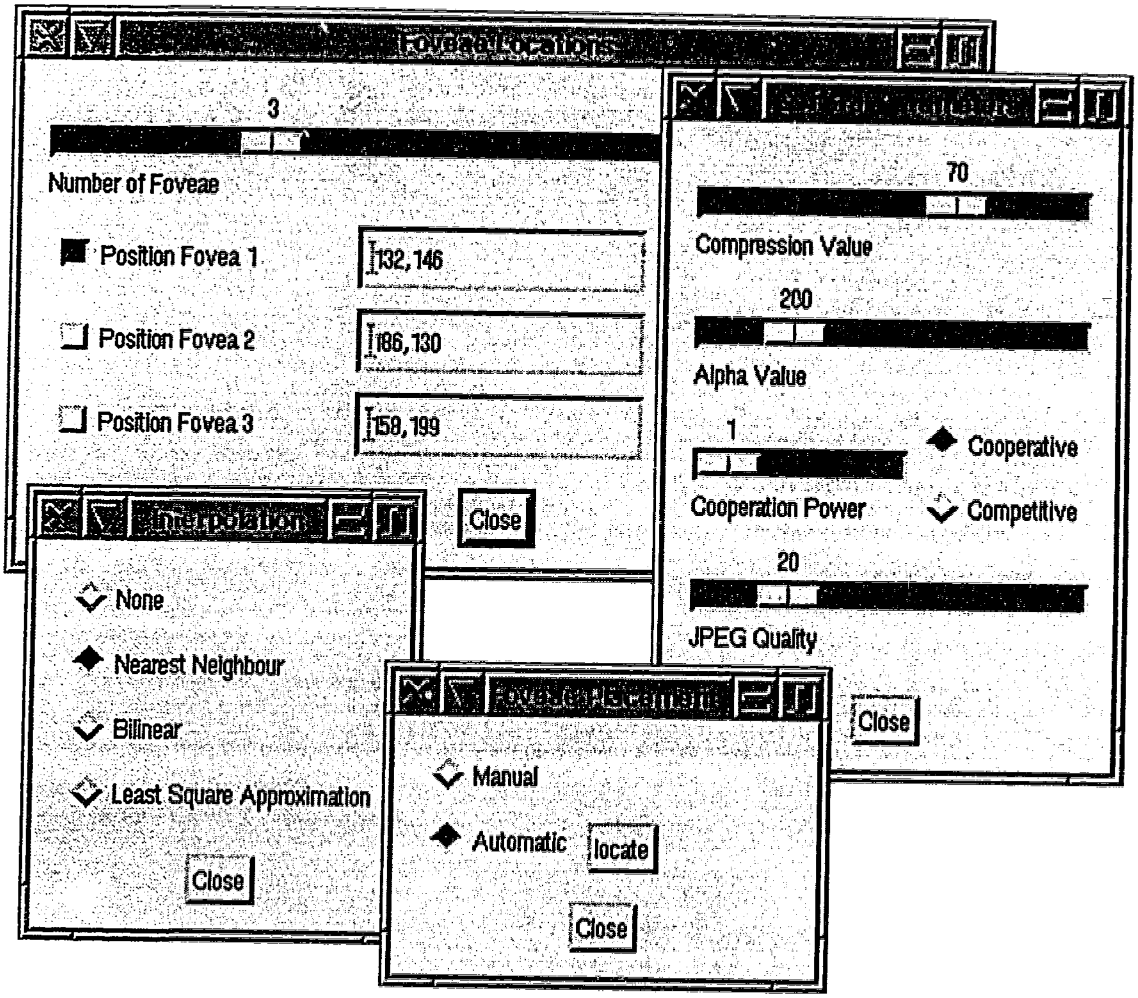


Figure 7.1: Example: Main *FaceBase* application window. (Cross-hatches indicate the three foveae locations.)

Many significant features included in this tool are listed below.

[2] The majority of the work on this application was completed by the author. However, some work on the user interface and implementation of *powers of cooperation* was provided by Kirill Richine. Anup Basu also directed Kirill in helping to interface this code with Gloria Chow's feature detecting code. As mentioned in Section 5.1.2, some of the SVR code originated from Allan Sullivan's original SVR code, but was significantly modified.



Foveae Locations

Number of Foveae: 3

☒ Position Fovea 1: 132,146

□ Position Fovea 2: 186,130

□ Position Fovea 3: 158,199

Compression Value: 70

Alpha Value: 200

Cooperation Power: 1

◆ Cooperative

◇ Competitive

JPEG Quality: 20

Interpolation

◇ None

◆ Nearest Neighbour

◇ Bilinear

◇ Least Square Approximation

Foveae Placement

◇ Manual

◆ Automatic

Figure 7.2: Example: control windows for the *FaceBase* application.

- *FaceBase* utilizes a simple, easy to use Motif user interface.
- Many of the SVR alternatives are provided as options. Both *cooperative* and *competitive* foveae are available, with all parameters easily modified and displayed. The JPEG Q parameter is also available.
- Up to 10 foveae are supported.
- Multiple interpolation methods are available.
- Fovea positioning can be done manually either by clicking a mouse directly on the image or entering exact coordinates with the keyboard.

- Fovea positioning can be done automatically, using feature detection routines written by Gloria Chow. With this method, three foveae are automatically placed on the left eye, right eye, and the mouth, if possible.

- The display window can be toggled between the original, compressed, and decompressed images.

- The location of the fovea can be displayed, if desired, as cross-hatches over the image. Both the compressed and decompressed images can have the fovea positions marked.

- Multiple file formats are supported. These include JPEG, PPM, PGM, BMP (bitmap), Targa, GIF, OS2, as well as both SVR/PGM and SVR/JPEG formats. Both reading and writing of files is supported, although colour images are converted to greyscale when they are loaded. File type is also automatically detected when loading. Directory listing, filtering, and file searching features are available both when loading and saving images. Compressed and decompressed images can both be loaded and saved.

- Scrollbars automatically appear for use with large images.

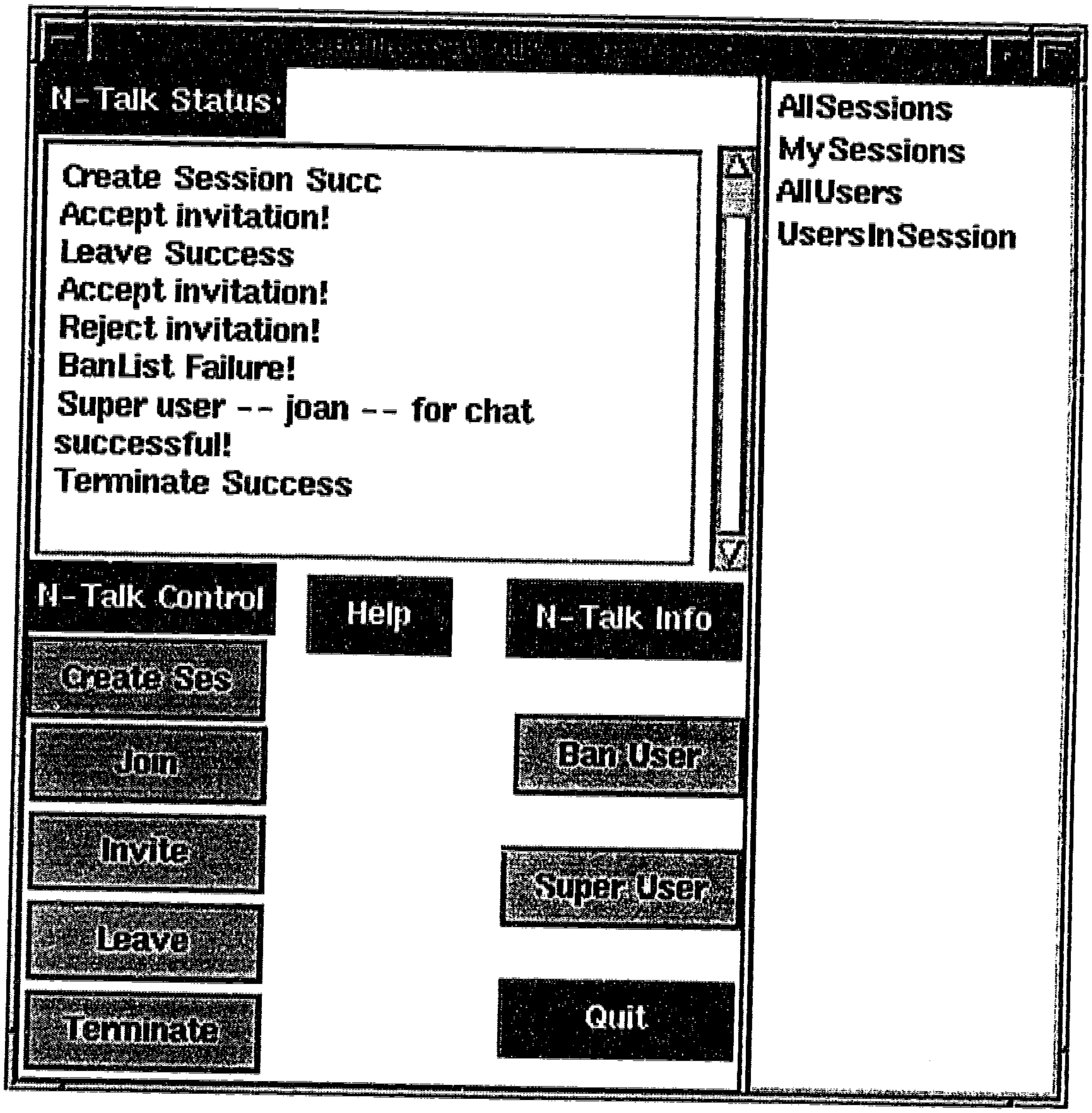
- Several Help windows are available for assisting users.

- Command line options allow a series of compressions to be carried out on one image, in batch mode. The range and progression of JPEG Q values and SVR Comp values can be specified on the command line as well.

With this tool, maintenance of an image database becomes less difficult. Directories of files in the hybrid formats can be quickly viewed, updated, and modified. The faces of all faculty, staff, and students in the Computing Science Department at the University of Alberta were placed in a Face Database using this tool.

7.1.3 User Identification in Collaborative Work Environments

The author's research included working on an application which provides a real-time collaborative work environment for multiple users [75] (See Figure 7.3). One aspect of this project is a central Face Database. Each user who connects to the application can request the images of any or all of the other participants (See Figure 7.4). Also, when discussion groups are established, the image of the speaker is placed on the side of each window (See Figure 7.5).



Create Session Succ
Accept invitation!
Leave Success
Accept invitation!
Reject invitation!
BanList Failure!
Super user -- joan -- for chat successful!
Terminate Success

Figure 7.3: The main control window from the Collaborative Work Environment application.

The database is maintained with the *FaceBase* tool. The server and client programs of the collaborative application use the library of functions to access and decompress the SVR/JPEG images. Extreme compression is required to reduce the bandwidth and time required to transmit the facial images from the server to the re-

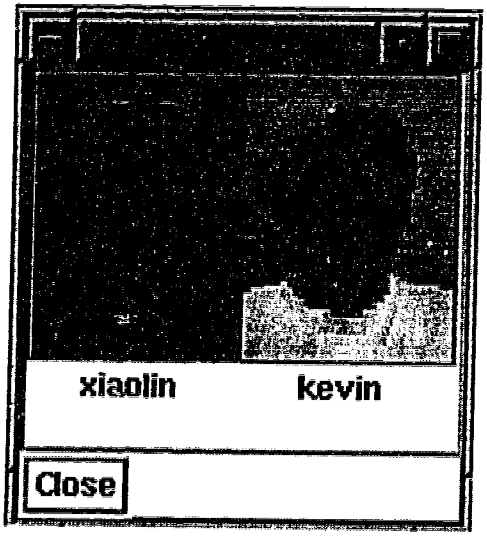


Figure 7.4: Multiple face display window from the Collaborative Work Environment application. (Note: Approximately 20:1 compression ratio achieved. Nearest neighbour interpolation is used.)

mote clients. The images are transmitted in the compressed format and decompressed "on the fly" at the client's location.

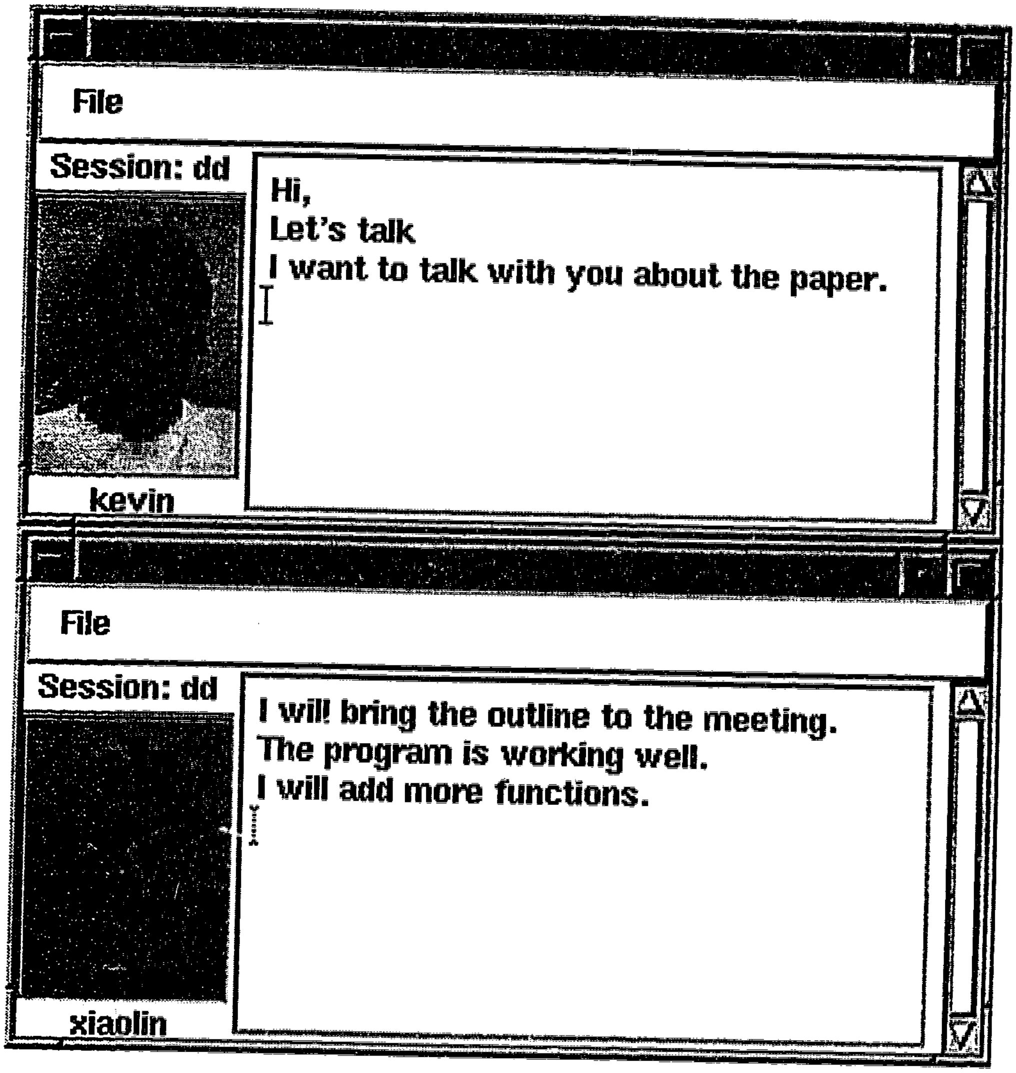
7.2 Video Compression: CVR/MPEG Hybrid

Video compression takes a sequence of still images and compresses it into a single file. Video decompression then takes this compressed file and reconstructs the sequence of images. The results produced by the video decompression can be saved in separate image files, but it is most commonly displayed directly to the user.

Section 2.4.2 outlined one of the currently most popular video compression standards, MPEG. Figure 7.6 **(Top)** illustrates how MPEG takes the input image sequence along with the required compression parameters to produce a compressed video file. Figure 7.6 **(Bottom)** outlines the process of decompressing an MPEG file[3].

In a method similar to the SVR/PGM and SVR/JPEG hybrids presented in Section 7.1.1, the CVR compression techniques were combined with MPEG to produce the hybrid CVR/MPEG video compression format. Figure 7.7 **(Top)** illustrates how

[3] Software to implement both these functions were obtained from University of California at Berkeley, with permission.



Hi,
Let's talk
I want to talk with you about the paper.

I will bring the outline to the meeting.
The program is working well.
I will add more functions.

Figure 7.5: The talk window from the Collaborative Work Environment application. (Note: Approximately 20:1 compression ratio achieved. Bilinear interpolation is used.)

the input image sequence is passed first through the CVR routines before being further compressed by the MPEG algorithm. Figure 7.7 (**Bottom**) outlines the process of decompressing a CVR/MPEG file.

The software used to implement MPEG was not altered. Notice, however, that the CVR parameters are not encoded in a comment in the header, as they were with the single image hybrids presented in Section 7.1.1. The CVR parameters are instead included in a separate file and paired with the compressed video file. The necessary parameters include the decompressed image dimensions as well as the compression

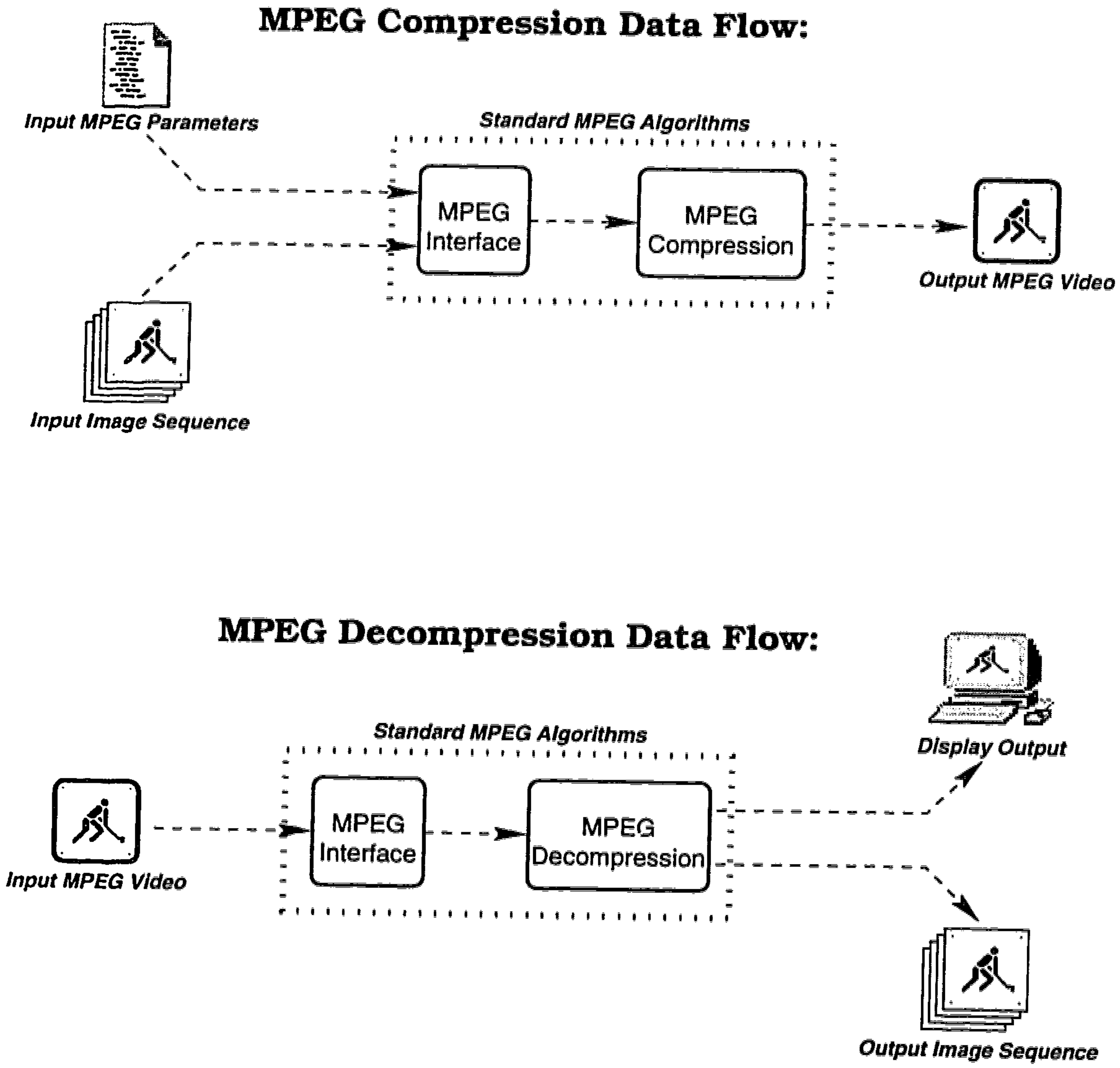
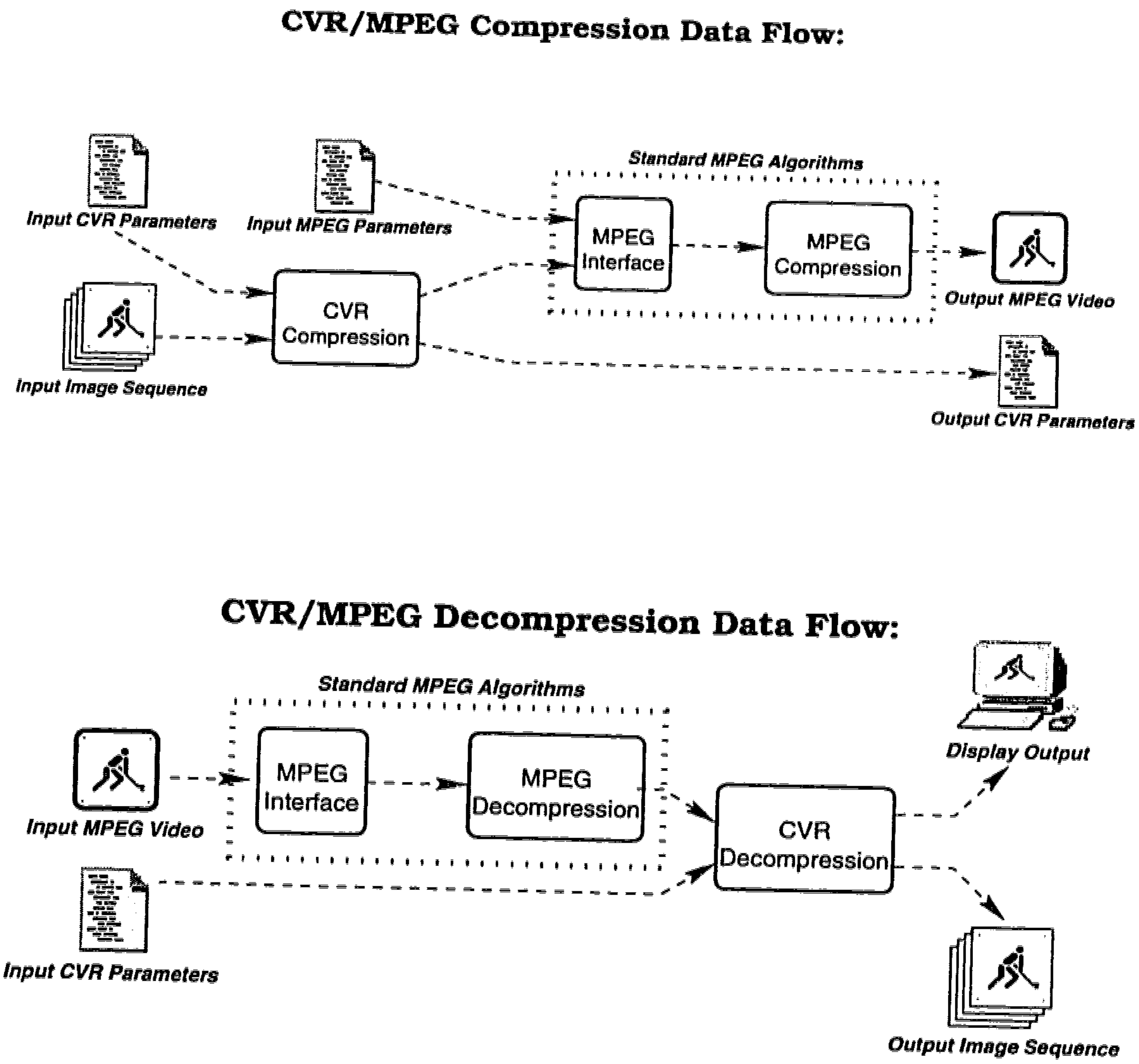


Figure 7.6: **Top:** MPEG video compression. **Bottom:** MPEG video decompression.


CVR/MPEG Compression Data Flow:

CVR/MPEG Decompression Data Flow:

Figure 7.7: **Top:** CVR/MPEG video compression. **Bottom:** CVR/MPEG video decompression.

parameter, α, and the fovea location values for each image. The last four values need only be included if they change from the previous image, so there will be at least one set of them, and at most one for each frame. This file may typically be only a few bytes long.

Recall that the CVR compression method was designed to operate efficiently with dynamic foveae. To assist in utilizing this capability, numerous automatic fovea locating features were implemented. These options, as well as the general compression parameters can be included in the standard MPEG parameter file (see Appendix C). The options include:

- Using CVR compression: if this option is not chosen, the CVR steps will be bypassed and standard MPEG compression will be done. Also, in order to fulfill MPEG's input size requirements, we must either *pad* or *crop* the images to the correct size of a multiple of 16 pixels for each dimension. The necessary line in the input parameter file to choose cropping is:

  ```
  USE_CVR      CROP
  ```

  The necessary line in the input parameter file to choose padding is:

  ```
  USE_CVR      PAD
  ```

- Setting the compression value: for example, the necessary line in the input parameter file to set the compression value to 95 is:

  ```
  CVR_COMP     95
  ```

- Setting the α value: for example, the necessary line in the input parameter file to set α to 200 is:

  ```
  CVR_ALPHA    200
  ```

- Setting a *static* fovea: the user can specify that every image will have the same fovea location. For example, the necessary line in the input parameter file to set the static fovea to (176, 144) is:

```
CVR_FOVEA   STATIC 176 144
```

- Centering a *dynamic* fovea: the user can specify that the fovea be in the center of every image. For example, the necessary line in the input parameter file to center a dynamic fovea is:

```
CVR_FOVEA   DYNAMIC CENTER
```

- Setting a *dynamic* fovea on a human face: the user can specify that the fovea track a human face through the entire sequence. The face detection algorithms used here are the same as those used in Section 7.1.2 and require a *size* parameter which indicates the size of face to search for. For example, the necessary line in the input parameter file to position a dynamic fovea on a face in each image with the scale value at 300 is:

```
CVR_FOVEA   DYNAMIC FACE 300
```

- Setting a *dynamic* fovea on movement within the scene: the user can specify that the fovea track movement through the entire sequence. The fovea is placed in the center of the first image, and thereafter a motion detection algorithm written by the author is used. This algorithm simply performs an image subtraction followed by a thresholding. The center of mass is calculated on the resulting pixels of motion. This algorithm requires a threshold parameter which sets the minimum greyscale change that would classify as motion, and a scale parameter which indicates the level of detail to search. For example, the necessary line in the input parameter file to position a dynamic fovea on an area of motion in each image with the threshold value at 30 and the scale value at 4 is:

```
CVR_FOVEA   DYNAMIC MOTION 30 4
```

- Setting a *dynamic* fovea with user defined locations: the user can specify the exact location of the fovea in each image, through the entire sequence, if desired. In this case the user must provide a text file with all the locations listed sequentially. For example, the necessary line in the input parameter file to position a dynamic fovea on the locations specified in file "movie.fov" is:

  ```
  CVR_FOVEA   DYNAMIC FILE movie.fov
  ```

The current CVR/MPEG programs were designed in such a way so the fovea location functions can easily be extended in the future. More features of this nature can be added at any time with no changes to the compression code being necessary [47, 112].

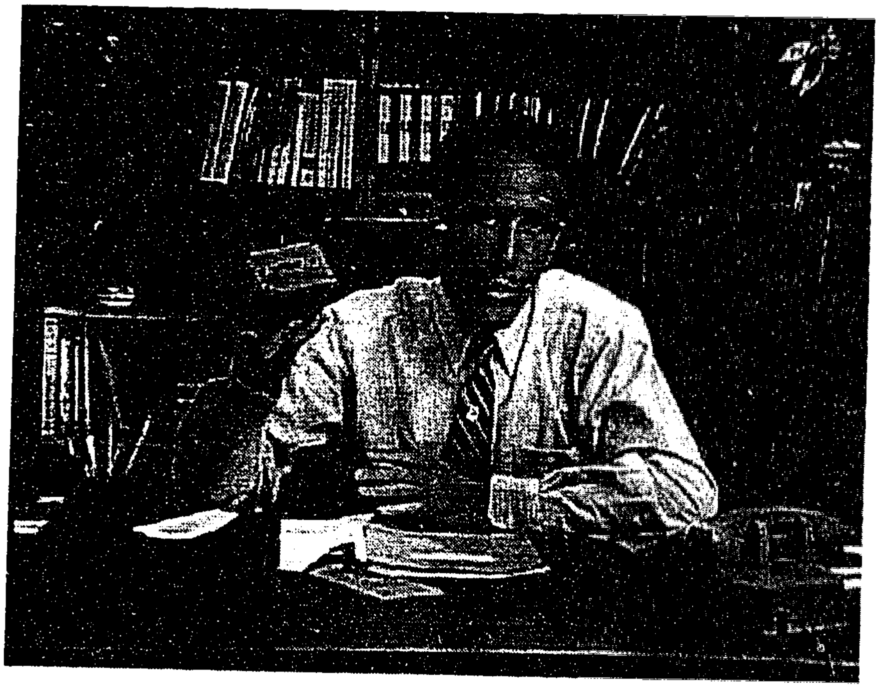


Figure 7.8: Example image from the original *Salesman* video.

One of the image sequences used to test this program was the widely available "salesman" video. Figure 7.8 is an example of one of the images in this sequence. Figure 7.9 demonstrates MPEG compression of this sequence at a high compression ratio

(70:1). Figure 7.10 demonstrates the CVR/MPEG compression method used on the same sequence with the same resulting compression ratio. Clearly the CVR/MPEG hybrid outperforms MPEG alone in this example.

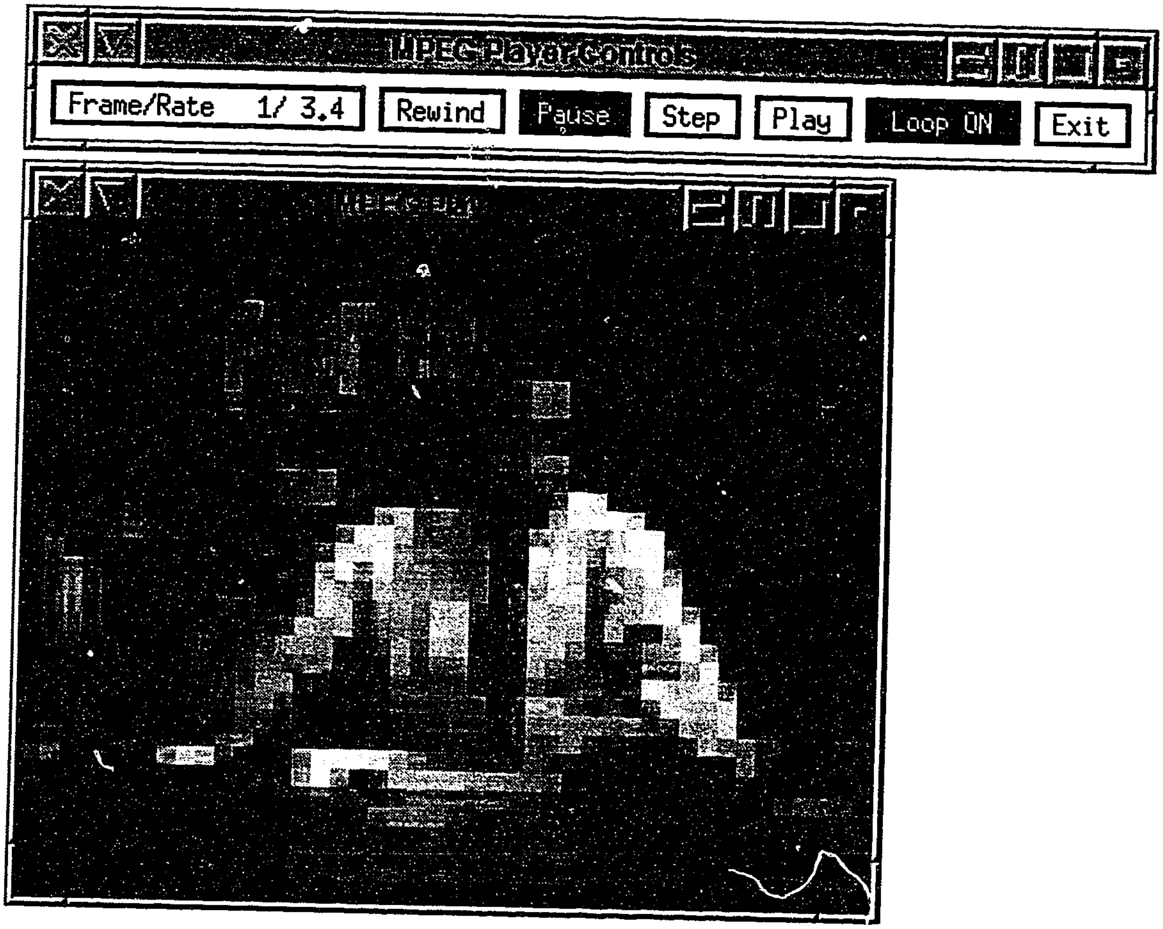


Figure 7.9: Example image from the MPEG compressed *Salesman* video. (Compression ratio is 70:1.)

7.3 Videophone

Videophone applications require the real time compression and transmission of images. They are prime applications for VR compression schemes for several reasons:

- Videophones require fast compression of images
- High image quality is not a high priority
- The typical *talking head* scene provides an ideal fovea location

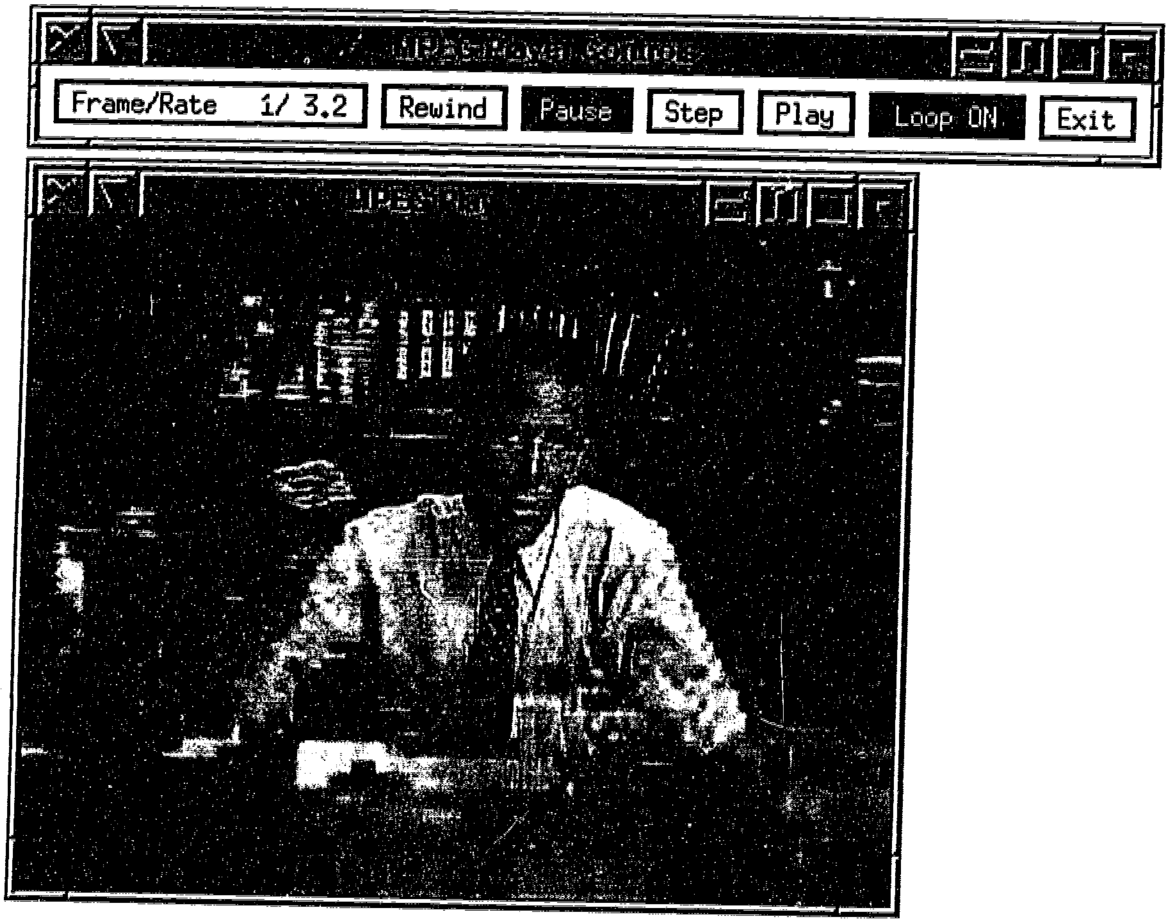


Figure 7.10: Example image from the CVR/MPEG compressed *Salesman* video. (Compression ratio is 70:1.)

- VR methods can provide constant compression, suitable for transmission of images over channels with a fixed bandwidth.

A tradeoff between the refresh rate and image quality can easily be achieved by varying the compression parameters.

To date we have not implemented a complete videophone application utilizing VR compression methods. While tools suitable for real time audio and video compression exist, they were not available to the author. The videophone, however, can be viewed as an application with a subset of the features present in a general videoconferencing tool as outlined in the following section. Section 7.2 demonstrated the technical validity of incorporating the CVR algorithm with standard MPEG video compression.

7.4 Videoconferencing

As with the use of videophones, teleconferencing involves the compression and transmission of images. Unlike videophones, the scenes may not always contain simple head shots, but are usually situated in structured environments. A typical view may be of several people seated and/or a speaker in front of an overhead or blackboard. In such a situation, a fovea cannot be placed in the center of the image automatically, but may require the user to place multiple foveae where appropriate, possibly indicated by pointing with a mouse, or automated in some way. Areas of interest could include the faces of people in the scene, the blackboard area, and so on. Some foveae could be static (as with seated individuals) while other dynamic foveae could use existing tracking algorithms to follow the motion of an individual through a scene (as with a speaker moving back and forth).

7.4.1 A Prototype Videoconferencing System

Research done on the SVR compression method in the past has included a rudimentary prototype of a videoconferencing system [89]. The author's research has enhanced this implementation significantly. The system can function in a generic environment with little or no additional hardware. Such a system is inexpensive, easy to upgrade and maintain, and portable across many systems.

Much of the original prototype code was rewritten, modularized, and optimized. The source code has also been rewritten to be portable between several platforms, such as Sun 3 workstations, Sun 4 SPARCstations, Solaris workstations, and SGI workstations. Support for additional camera systems (e.g., IndyCam) was added. Both XView and Motif interfaces were added. The number of grey levels being handled was doubled, from 128 to 256 levels. Improvements to the multiple foveae features and interpolation methods were made. The *dynamic foveae* feature was added, using the new CVR transform. Capabilities for automatic feature tracking were added, and saccadic, self-directed foveal movement functions were implemented.

The current videoconferencing prototype is able to transmit greyscale images from an image *server* to a *display* or *viewer* process. The server process is responsible for capturing, compressing, and transmitting the image. The display process accepts compressed images, decompresses them and displays them on a screen. Figure 7.11 illustrates the videophone organization.

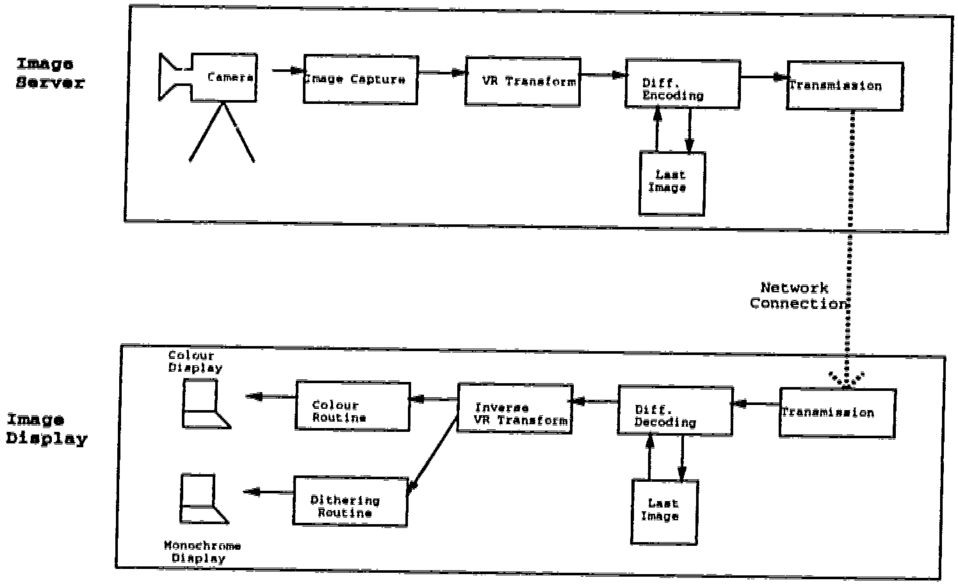


Figure 7.11: Diagram of a videoconferencing prototype.

In addition to the CVR encoding, additional compression is provided by an intraframe difference encoding routine. The difference between pixels in successive frames is found (most pixels will not change if the image is static) and only changed pixel values are transmitted, in Run Length Encoded (RLE) format.

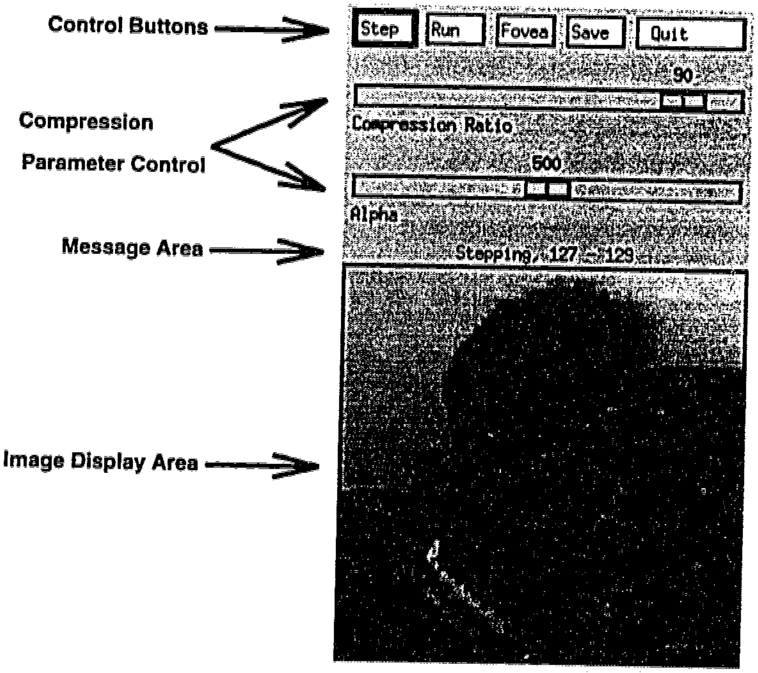
Currently, frame rates on the order of 8 - 15 frames per second have been achieved across an Ethernet, up from 1.5 - 2 frames per second in the original prototype presented in [89]. With the compression values obtained while maintaining reasonable image quality(up to 98% with interframe encoding) the network can easily handle much higher frame rates while sustaining acceptable quality. Especially worth noting is that the system is based entirely on software; thus, no special compression hardware is required.

The videoconferencing component operates on the same principle as the biological systems researched — however, multiple foveae can be placed at the user's discretion. Dynamic foveae have been implemented, and algorithms can be provided which automatically track any person or other moving object in a scene, as chosen by the observer. Current options include stationary foveae, random saccadic movements around the original location, and simple path following. Face and eye tracking, and moving object tracking are also possibilities. Several programs which track objects or features within an image sequence already exist [47, 112]. Such programs could be used effectively to control the movement of a fovea automatically. Buttons which provide a real-time choice between these and possibly other automatic foveal controls can easily be added, but at present these choices must be made at the time the prototype is compiled.

The videoconferencing system can be cheaply and easily run over an existing network with no additional hardware cost other than camera equipment.

7.4.2 Details

The user interface has been rewritten from *XView* to *Motif* based. This was done for reasons of aesthetics and portability. Several tools have been added to the main window, allowing greater user control.



Compression Ratio 90

Alpha 600

Stepping, 127 - 129

Figure 7.12: Example videoconference window.

Single frames can be saved to a file, if requested by pressing the *Save* button. The system can also be run step by step, if the *Step* button is pressed. This will send only one image at a time, until another is similarly requested. *Run* will resume normal operation. Several foveae can be selected using the mouse, after *Fovea* is selected. Compression and α parameters are associated with sliders.

Feature Tracking

One might also find situations where an option to track features is required. For the hearing impaired, or those users who wish to read lips, it would be convenient to have a fovea located on the mouth in the image. Rather than having the user constantly move the fovea whenever the person being viewed changes position, a fovea could place itself over the mouth automatically, regardless of absolute position within the scene. This feature could also be extended to hands, if sign language was being used.

In both of these examples, when compared to typical videophone usage, a significant difference in the tradeoff between quality and speed would be required. The sliders controlling the compression and α values already allow for this flexibility. A user wishing to read lips may require an extremely high frame rate, and would concentrate almost entirely on the mouth. This user can then increase the compression ratio dramatically, reducing the bandwidth to increase the frame rate. The α value can also be greatly increased, making the fovea more prominent and causing the majority of quality loss to be taken along the periphery. The mouth would maintain an acceptable degree of clarity.

These specific tracking features have not been added to the prototype to date, but all that is necessary is the actual feature tracking code. An interface for self-directed dynamic foveae exists and has been successfully tested with simple foveal placement algorithms.

7.5 Foveated Scene Transmission over ATM Networks

7.5.1 Motivation

It seems natural to research data prioritization within computer image encoding schemes for scenes with spatially non-uniform importance. As with biological visual systems, congested networks could result in peripheral information being lost gradually, until total network failure results in 'blindness'. ATM networks allow data cell prioritization to control such loss.

An experiment was conducted with our anthropomorphically motivated teleconferencing schemes over simulated extreme network conditions. The study of the effects of serious packet loss provided insight into the viability of these methods in various levels of traffic congestion. If a videophone implementation similar to the prototype introduced in Section 7.4.1 e..iployed a VR compression technique and packet prioritization as outlined in Section 7.5.4, the author would expect scene degradation similar to that which may be experienced by a failing human visual system. Our simulation results provide support to this expectation.

Although mimicking biological visual systems perfectly may not always be desirable or optimal, nature provides us with the only objective standard outside computing with which we can gauge our success. Experimental simulation results might also prompt new theories on how the brain prioritizes visual data, thereby adding to our understanding of ourselves.

7.5.2 Goals

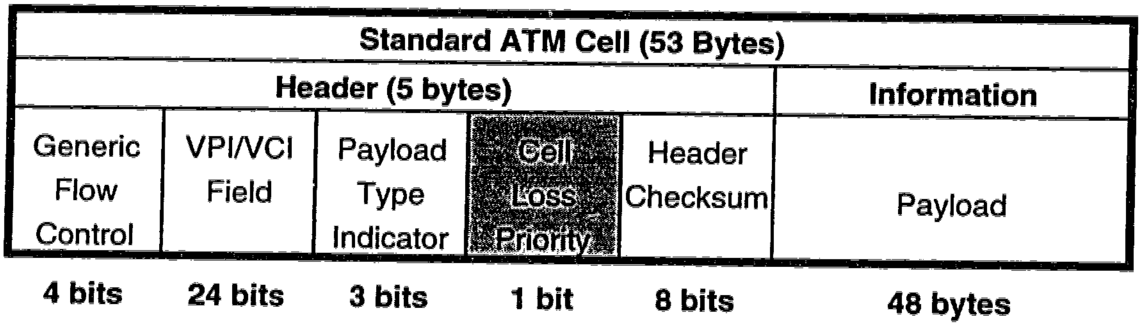
A goal in the transmission of image or video data is to reduce the bandwidth required while preserving reception quality. Data compression results in a smaller number of bytes encoding information, but may degrade image quality. As previously mentioned, there exists a large class of scenes (still images or video sequences) that are composed of particular areas of primary interest. While the remainder of the scene is necessary for reasons such as texture, context, aesthetics, or motion detection, its importance is significantly less than that of the primary area. Each scene is coupled with *foveae*, which mathematically delineate the positions and relative interest of each area. The VR single image compression techniques that have been presented in Chapters 5 and 6 appear to be successful in balancing required bandwidth and image quality for the transmission of foveated scenes [16] (see Figures 6.1 – 6.4).

The assumptions with VR compression, however, are that the transmission of image data occurred over reliable, static bandwidth connections. Data loss was not considered. Also, the parameters of the compression schemes must be adjusted prior to transmission in order to match the available or target bandwidth requirements. Where network charges are calculated on the number of bytes transmitted, any data reduction could result in significant dollar-cost savings, depending on the compression ratio achieved.

In some network scenarios, such as with ATM technologies, existing VR techniques may not be the optimal solution. By taking advantage of ATM network characteristics, we can improve the encoding scheme for certain classes of foveated image and video data.

7.5.3 Asynchronous Transfer Mode Technology

ATM networks are expected by many to become the most popular medium over which to transmit real-time multimedia data. Rather than a continuous stream, ATM is based on fixed packet sizes of data, or *cells.* Each cell is 53 bytes, 5 of which hold header information. Figure 7.13 shows the format of a standard ATM cell [33].



Standard ATM Cell (53 Bytes)					
Header (5 bytes)					Information
Generic Flow Control	VPI/VCI Field	Payload Type Indicator	Cell Loss Priority	Header Checksum	Payload
4 bits	24 bits	3 bits	1 bit	8 bits	48 bytes

Figure 7.13: The format of a standard ATM cell.

ATM networks require that connections be established prior to transmission. The users negotiate with the network and agree on the properties of the connection, including quality of service (QoS), bandwidth limits, and service class. QoS dictates the level of acceptable data loss, and the bandwidth limits bound the data transmission rate. The costs incurred and the traffic patterns produced by each service class are different [102].

Service Classes

In *guaranteed* service classes, such as *constant-bit-rate* (CBR) and *variable-bit-rate* (VBR), the network explicitly guarantees a QoS or does not establish the connection. *Available-bit-rate* (ABR) is a *best-effort* service class in which no such guarantee exists

[87].

The VR compression and transmission approach to foveated scenes is best suited to the most expensive class, CBR. Here the peak bandwidth required for transmission is requested and the entire collection of data is transmitted with low-loss probability. The bandwidth reserved for the connection must be paid for, whether or not it is fully used.

However, the VR approach could conceivably also be used with VBR class connections. There would be no guarantee that the entire image would be successfully transmitted. Only the bandwidth used would be paid for, but the risk of data loss would increase. During bursty periods, some information could be dropped by the network. The VR methods presented in Chapter 5 do not address this possibility, and therefore such data loss within the scene would occur spatially independent of the fovea [77].

The ABR scheme is the least expensive of the three, but there is an even greater risk of data loss, again randomly throughout the scene. Here one is taking advantage of otherwise unused bandwidth on the network, deferring to higher service class connections in times of congestion. The network dictates the bandwidth allocated to ABR connections, based on the current traffic load. VR compression schemes are quite inflexible in dynamically adjusting their bandwidth requirements. It is computationally expensive to adjust the compression parameters in real-time continually, based on the network load. Delays in adjustments complicate the process and can result in data loss. It is also unclear if all current rate, credit, and intelligent congestion control schemes proposed in the ATM Forum would forward the required network load status above the ATM switch network layer, to the application itself at the user layer. A flexible encoding scheme which can operate within these constraints is required.

Multicasting

While VR compression techniques can be used in broadcast or multicast applications, they are not particularly suited to heterogeneous link bandwidth scenarios. The inflexibility of these approaches would force one of several sub-optimal solutions:

- Broadcast separate streams over each group of links with the same bandwidth limit to give the best quality at each destination.
- Broadcast one stream over all links at the lowest capacity. The quality on the large bandwidth connections would unnecessarily suffer.
- Broadcast one stream over all links at the highest capacity. The connections with the narrowest bandwidths would lose information randomly within the scene.

A simple encoding scheme which is flexible enough to respond to bandwidth requirements dynamically and reduce quality intelligently during transmission is required. An appeal to biological visual systems provides insight into a satisfactory solution.

The VR compression schemes discussed in Chapter 5 have one characteristic that may be useful in optimizing performance within bandwidth restrictions. By varying the compression parameter and α, the bandwidth needs of the VR methods presented earlier will change. Bandwidth requirements also depend on whether static or dynamic foveae are implemented. It is possible that these choices can be made automatic, based on network feedback. Such applications would automatically avoid network congestion as well as make use of unused bandwidth in times of minimal traffic. They could be flexible enough to run on platforms where there are either constant or variable bandwidth allocation schemes.

7.5.4 Cell Priority

When information is transmitted over an ATM network, it must be broken up into *cells*, or *packets* of data, which are sent separately. When the network gets congested, some of these cells must be dropped and are therefore not successfully transmitted. The network may employ *policing* to prioritize data cells. A common method, *leaky bucket*, assigns priority levels to individual cells based on traffic patterns, negotiated connection service classes, and network congestion [17, 76]. Under heavy traffic load, the cells with the lowest priorities are lost first [49].

If several users have bursty traffic and are using the same ATM network with the VBR connection class, statistical multiplexing is often implemented. That is, a user is permitted to transmit data beyond the average (or sustainable) bandwidth limits negotiated for its connection during the periods where other users are not fully utilizing their assigned bandwidth. This traffic is then regulated with the use of a policing function setting the *Cell Loss Priority* (CLP) bit within each data cell. This bit marks each cell as either high or low priority. If some data must be lost, low priority cells are dropped before high priority cells [64].

Figures 7.14 and 7.15 are examples of image transmissions using 2 priority levels, suffering from the loss of 83% and virtually all low priority data, respectively. (Only 7% of the image data is high priority.) Specifically note the quality difference between the three final images in both figures, even though they have exactly the same amount of successfully transmitted data.

Typically, each user can transmit at whatever rate they wish, but the network will only set a portion of the data cells as high priority. Those cells which exceed the agreed bandwidth limit will be indiscriminately set to low priority and will be dropped if the network is congested (see Figures 7.14 B and 7.15 A).

When transmitting foveated images, it is possible to package data into spatially logical groups. That is, the images can be divided into sections, and each section transmitted as a single cell. In this situation, the priority of a cell can be determined by its relative distance from the fovea. The closer the cell's data is to the fovea, the more important the information is, and the higher the priority it should be assigned.

Depending on the policing strategy used, one can take advantage of statistical multiplexing and the CLP bit within the ATM networks to increase the quality of transmitted images. If one sets the CLP bit prior to transmission, restricting the number of high priority cells to the limit allowed under the VBR connection agreement, scene degradation similar to that which may be experienced by a failing human visual system in traumatic situations can be expected (see Figure 7.15 B).

7.5.5 Priority Dithering

One would expect that prioritizing data based on the distance from the fovea should result in gradually increasing peripheral quality loss during network congestion. However, when using only two priority levels, a circular region exists around the fovea, separating the high and low priority areas. Figure 7.14 B shows that for a moderate

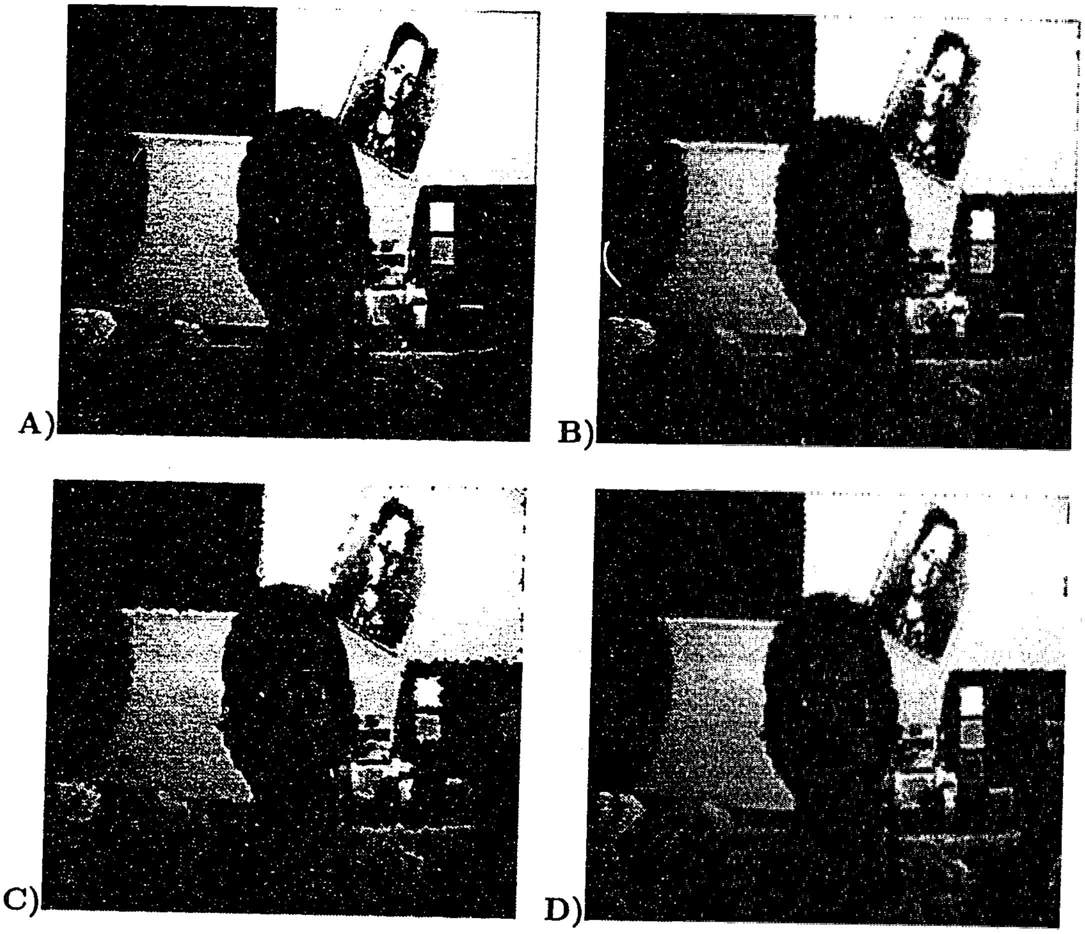


Figure 7.14: **A)** Original image. **B)** All pixels initially set to high priority, adjusted by the ATM policing function. **C)** Foveal based pixel prioritization. **D)** Dithered pixel priorities.

loss of low priority data (83%), prioritization based directly on the distance from the fovea is the superior method. The periphery is not completely lost and the circular threshold does not become apparent. However, Figure 7.15 B demonstrates that upon extreme loss of low priority data (100%), the quality of the scene in the periphery deteriorates completely, revealing the threshold as a circular artifact.

To retain the gradual quality loss even during the most extreme network congestion, one must utilize a greater number of priority levels. An increase of CLP bits to 5 would give us 32 priority levels, each level having a different probability of being dropped under network stress. By allocating the priority of the data based on the distance from the fovea, each outer *ring* of data would gradually deteriorate faster than the inner rings. The more levels available, the more gradual the deterioration

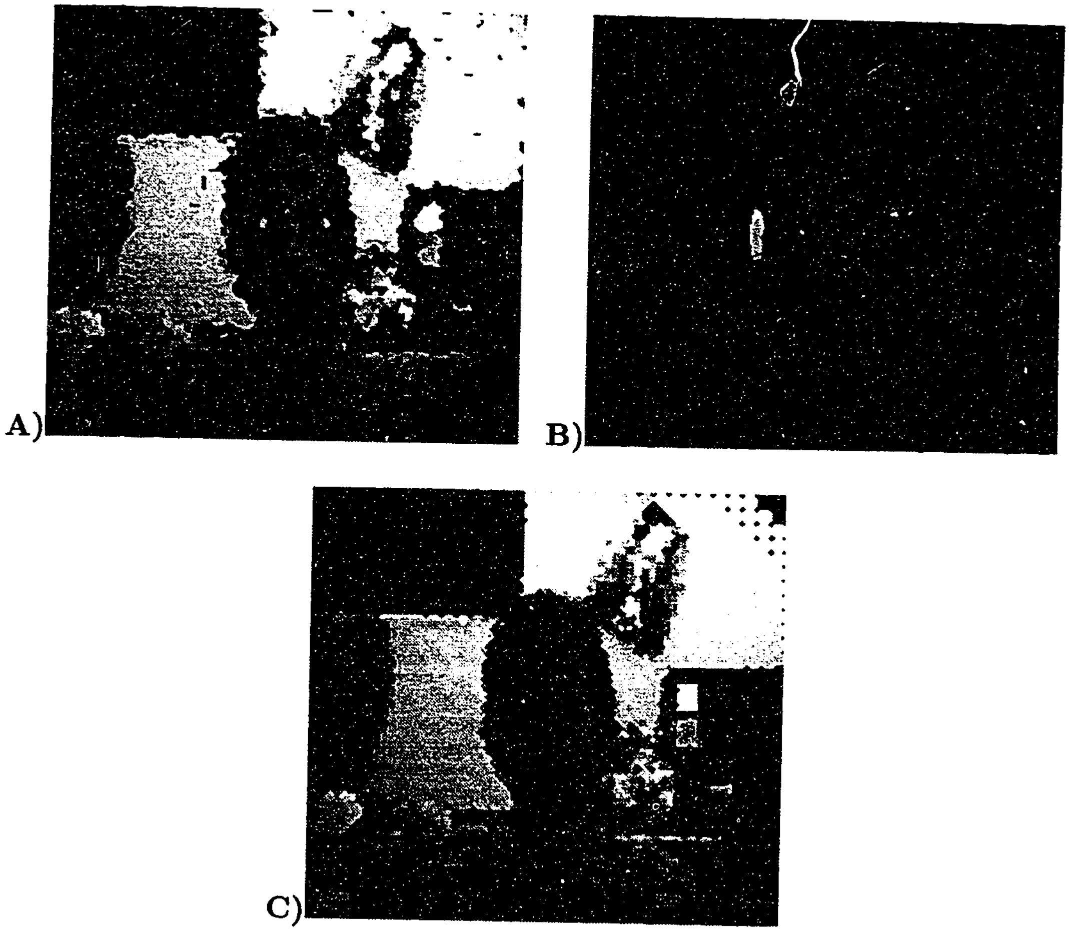


Figure 7.15: **A)** All pixels initially set to high priority, adjusted by the ATM policing function. **B)** Foveal based pixel prioritization. **C)** Dithered pixel priorities.

towards the fovea, therefore modelling the human experience more closely. Total peripheral loss could thus be further delayed.

However, current ATM standards provide one CLP bit, or two priority levels. One can *simulate* multiple priorities with only one bit with the use of *dithering* [100]. The advantage of priority dithering over foveal prioritization which is *not* dithered can be seen in Figures 7.15 C and 7.15 B, respectively. In both cases virtually all low priority cells were lost in transmission. Observe that while the periphery has deteriorated in both cases, it is not completely lost with dithering — as it is when dithering is not used. While dithering does not produce the best results for moderate loss of low priority data (Figure 7.14), it is clearly superior with extreme levels of low priority data loss (Figure 7.15).

7.5.6 Network Simulation

The quality differences between scenes transmitted with foveated and dithered priority settings can be demonstrated with a relatively simple ATM network simulation. Our simulation consisted of sending 255 facial images through the network, with one fovea placed over the face. These scenes included various conditions of illumination, scale, background, and head orientation[4] (See Figure 7.16).

Figure 7.16: Three examples of faces used in the ATM network simulation.

Setup

The simulation was written in C++ and gawk, and run on a Solaris workstation. The test sequences of facial scenes simulated encoding, prioritization, and transmission

[4] The faces were obtained with permission from the Massachusetts Institute of Technology (MIT) face database.

with the VBR connection class through the ATM network with identical simulated background traffic. Several competing background VBR video traffic streams were realistically approximated using a *Multiple Markov Poisson Process* (MMPP) traffic generator [44]. The mean duration of a busy period of background traffic was 250 microseconds, while the mean duration of a calm period was 1000 microseconds.

The simulated ATM network contained one switch with an outgoing capacity of 45 Mb/s and a packet transmission time of 8.985731 microseconds per packet. The incoming traffic shared 100 cell buffers when not occupied by reserved bandwidth. Images for the test sequences were transmitted at 30 frames per second.

Three prioritization schemes were compared — the ATM policing function automatically adjusts the priority levels of each stream using the *leaky bucket* method:

(i) All cells are initially set to high priority; no prioritization based on the fovea location.

(ii) Initial binary prioritization based solely on the fovea location.

(iii) Initial binary dithered foveal prioritization, simulating multiple priority levels.

Encoding ensured that inter-cell and intra-cell pixel order within cells of the same priority level were randomized to avoid consecutive sections of the image being lost from either a single lost cell, or consecutive cell loss during bursty network congestion. Simple look-up tables were used to map pixel locations within the image to locations within ATM cells. Each ATM cell contained sequencing and error checking information along with the image data.

Standard *leaky bucket* policing and congestion control was used within the ATM network. The policing function considered the user's preset priority (if any) for each incoming packet and only incremented its counter on high priority packets. The policing function reduces the priority of the high priority packets when the rate of incoming cells exceeds the sustainable bandwidth limit agreed upon at connection setup time.

Interpolation[5] was used to reconstruct the images from the cells that successfully

[5] A Gaussian filter was used.

passed through the ATM network. After a warmup period, the average cell loss and image quality were calculated for each level of network congestion. The quality of each image was calculated using the error measure developed for foveated scenes.

Foveated Quality Measure

In the area of image quality, many standard measures attempt to quantify the degradation process objectively. Here our research deals with a restricted domain of digital images, specifically those with identifiable areas of greater interest. Performance measures suitable to foveated scenes must therefore be used. The foveated quality measure VRMAE, described in Section 6.2.2, was used for our analysis here.

Analysis

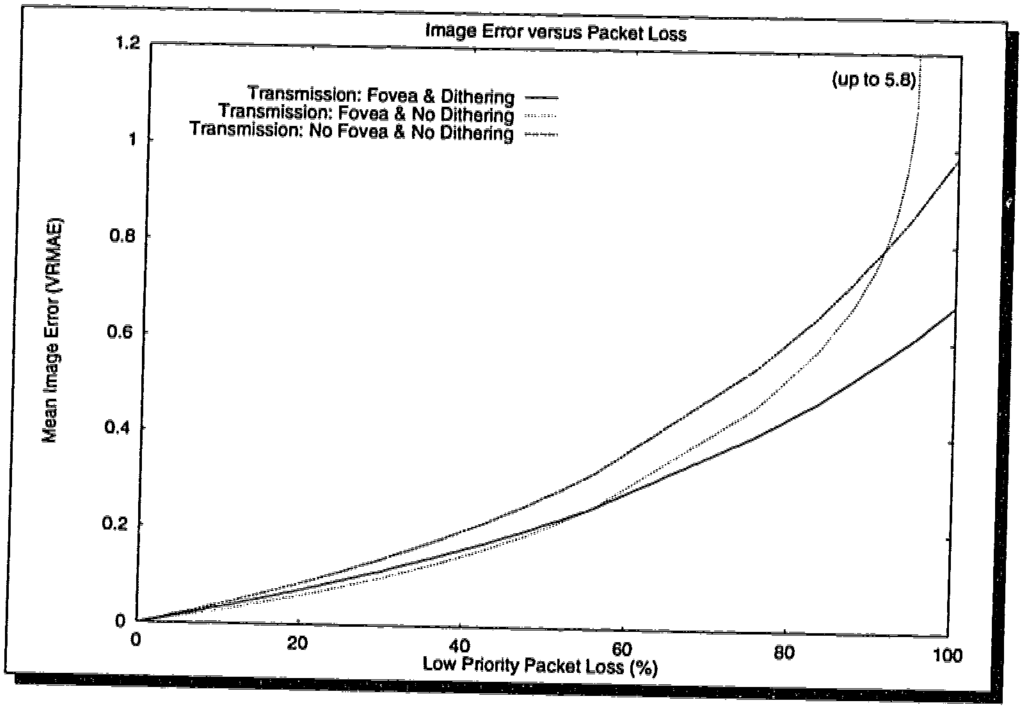


Figure 7.17: ATM simulation results.

Figure 7.17 shows the performance of three prioritization methods during increasing cell loss caused by increasing network loads. Initially, under light congestion and moderate cell loss, both foveated prioritization schemes outperform the standard encoding method. The dithering method does have a slightly higher error than non-dithering, as some cell loss occurs closer to the fovea. In this case, non-dithering produces the best results.

However, under extreme congestion and cell loss, the total deterioration of the periphery in the non-dithered scheme becomes unacceptable, both visually and as expressed by the VRMAE metric. The gradual peripheral quality loss without total degradation, in the dithered approach, consistently produced better visual results than the standard method that did not consider foveae. Clearly, in this case, dithering greatly improves image quality, and is the superior method.

7.6 Foveal Prioritization Advantages

Foveated scenes, such as typical videoconferencing transmissions, possess qualities which can be exploited advantageously. Chapter 3 examined biological systems to provide insight into how nature compresses image data based on the presence of foveae. ATM cell prioritization gives us the opportunity to model biological foveated scene prioritization. Our simulations show that by prioritizing and dithering encoded image data based on distance from a fovea, increased network congestion will cause peripheral deterioration as opposed to spatially independent noise. Two communications scenarios demonstrate the benefit of this technique.

7.6.1 Frugal Service Classes

The transmission of information over ATM connections requires a QoS contract, including a specified service class. While the CBR service class is the most reliable, and thus will result in the highest quality image, it is also the most expensive. *Foveal prioritization* concentrates cell loss to the scene periphery, thus increasing the perceived image quality using lower quality and less expensive service classes such as VBR and ABR. A user may find that an acceptable QoS on a cheaper service class is not possible without such prioritization.

Furthermore, complex dynamic encoding techniques are not necessary when using the ABR service class. Instead of altering the encoding method to reflect the current bandwidth limit available to the connection, foveal prioritization will operate effectively, without adjustments, under all conditions. The policing filter at the entrance to the ATM network will adjust the bandwidth automatically, taking the cell priorities set by the user into account [37].

7.6.2 Multicasting over Heterogeneous Link Speeds

Simultaneously transmitting images to multiple destinations over ATM may include scenarios where all communication links do not have equal bandwidth capabilities. It is certainly not desirable to encode and transmit the information separately along each link speed. It may also not be practical to either reduce the single stream to that of the lowest link capacity or to transmit at a higher rate, causing spatially independent scene deterioration along the lowest capacity connections. *Video Gateways* have been proposed to address this problem [39, 93].

Video Gateways

Video Gateways are elements within the transport level of a network that monitor video signals being transmitted through that switch. As the video signal enters the network switch, the video gateway interprets this signal and transmits a (possibly) altered video stream. *Rate-reduction* video gateways compress the incoming video stream to a lower bandwidth requirement before retransmission.

In the process of reducing video signal bandwidth, three subsampling techniques are available to video gateways. *Spatial subsampling* reduces the dimensions of each frame of the video. *Temporal subsampling* reduces the frame rate of the video by dropping selected frames altogether. *Amplitudinal selection* reduces the quality of each frame.

As well, video gateways can operate at two different levels. *Pel level* gateways decode the incoming video stream entirely, up to the pixel (or *pel* level. At this level many methods of subsampling and compression can be utilized as the video signal is encoded to lower bandwidth requirements. *Transform level* gateways decode the incoming video signal to the transform level only, and further compress transform coefficients. Only where codec standards allow (i.e. H.261 and MPEG), the concept of frames can exist at the transform level, allowing spatial and temporal subsampling at the transform level [27, 28, 99].

All of the above decoding and encoding allows a network with video gateways to control the stream bandwidth dynamically at each branch or node within the network, addressing the multicasting problem at hand. However, such solutions have significant drawbacks, not the least of which is switch complexity. Each gateway must contain the necessary information to intelligently monitor available bandwidth as well as all video decoding and encoding programs. Delays are introduced at each gateway, depending at which level the video stream is operated on, the number of cells which need to be buffered to complete the decoding, and the complexity of the compression methods used [31, 34].

Foveal Prioritization

Foveal prioritization, on the other hand, allows a single, simple encoding to be transmitted to all destinations without the need for additional video gateway hardware. Each switch need not know the encoding method used within the video stream, but simply the priority of each individual packet. Upon entering an ATM network or link with increased bandwidth constraints, the policing functions already present at that switch will automatically reduce the bandwidth of the stream, taking the cells' preset priorities into account. While slower links will obviously have lower quality images, they will be of a higher quality than that possible without foveal prioritization and dithering, because the majority of information loss will occur away from the fovea. The simple encoding scheme produces a single outgoing multicast stream for all destinations and would work seamlessly with multiple link speeds and without the added delay of video gateway technology.

Drawbacks to foveal prioritization stem primarily from the complexity of compression methods available. Pel level video gateways do not have requirements on their compression methods to include spatially equivalent data within the same ATM cell, while prioritization methods based on fovea location do rely on such encodings. In some cases the higher compression ratios achieved by complex video gateways may outweigh the delay and expense they incur. However, with a good knowledge of the network traffic characteristics, a wise choice of prioritization parameters (such as ratio of high priority cells and dithering rate) may achieve an acceptable level of quality loss at all multicast destinations.

While our ATM simulations prioritized facial image data without advanced compression techniques, their utilization is possible. JPEG and MPEG are good examples of popular compression techniques which operate locally, in 64 bit blocks. The image transformation and coefficient quantization all occur within these areas, and therefore the possibility of encoding them within the same ATM cell exists. In this way data prioritization based on the location of a fovea is possible. While wavelet compression in general cannot be combined easily with foveal prioritization methods, special wavelet techniques that operate on 64 bit blocks do exist, which could take advantage of known foveae locations. One could label this foveal based prioritization of compressed video as a *foveal layering* technique, however such layers would actually constitute a continuum of enhancement data when dithering is incorporated into the encoding scheme, as outlined in Section 7.5.5.

Chapter 8

Future Work

Our work, while covering many areas, reveals still more issues that may spawn further fruitful research.

8.1 Implementation Optimization

Section 5.1.2 described areas where previous SVR image compression code was enhanced. Search algorithms, interpolation methods, and internal data structures were optimized for speed or image quality. Preliminary comparisons indicate new methods are indeed promising, but do not prove general application. A wider range of sample images is necessary in order to be able to draw any significant conclusions. As well, bilinear interpolation is not the only method one can apply during decompression. Specifically, the speckle artifacts which arise from irregular sampling in the SVR method do not impact the signal to noise ratio, but are clearly visible. Lagrange, Gaussian filtering, and weighted average interpolation methods will not be affected by irregular sampling, and their use with the SVR transform might also be investigated.

Indeed, all of the code used in applications, simulations, and experiments could be further optimized. While most code was written for efficiency, time was not taken specifically for thorough inspection and optimization.

8.1.1 Videophone Prototype Performance

For example, more effort is currently underway to improve the performance of the videophone prototype. While the speed has greatly increased with the enhancement, replacement, or optimizing of algorithm implementations, more attention could be specifically directed to this area. Also, tradeoffs between speed and quality should be taken into account when making decisions on interpolation methods to be used.

8.2 Colour Images

Our research has mainly been with greyscale images. The majority of our work could be almost directly applied to colour images as well. For instance, the image compression techniques could divide an image into its red, green, and blue colour planes and each compressed separately with the methods as currently presented. However, it may be more advantageous to break the image into its luminance and chrominance colour planes and compress these with different parameters. More work might be done in intelligently controlling the respective compression ratios, having a greater reduction in the chrominance data as compared to the reduction of luminance data. Also, greater α values in the chrominance planes would more closely reflect biological vision systems. In animate retinae, cone density (colour sensitivity) decreases with eccentricity from the fovea much more rapidly than rod density (luminance sensitivity).

8.3 Single Image Hybrids

Sections 6.1 and 7.1.1 outline research involving hybrid compression schemes using the SVR transform, PGM, and JPEG methods. Examples of CVR hybrids with CVR and Wavelet and Fractal are shown in Figures 6.3 and 6.4, but are not analyzed in the detail SVR/JPEG is, nor are formal hybrid format specifications proposed. This analysis could be done, not just with these hybrids, but numerous other compression hybrids could also be investigated. Not only would hybrids with other popular compression methods require future investigation, but single image hybrids with other combinations of VR compression schemes might also yield interesting results. CVR, for example, could easily be added to the SVR/JPEG hybrid image format outlined in Section 7.1.1. This format already allows for easy extensions to the existing header syntax. Perhaps the *FaceBase* tool described in Section 7.1.2 might also benefit from the added option of CVR, Fractal, or Wavelet based hybrids.

8.4 Image Sequence Hybrids

Section 7.2 reviewed the implementation of the CVR/MPEG hybrid compression scheme for image sequences. This program could be extended to compress colour images, which it does not do at present. As well, the CVR formulae need not be kept entirely separate from the MPEG section. By integrating these to methods at a lower level, redundancy is reduced, speed could be gained, and greater quality could be achieved. The use of other VR methods such as SVR could be investigated here as well.

8.5 Scaling Factors

Throughout our research, the implementation of all SVR and CVR based transforms calculated the scaling factors necessary based on a single compression factor provided by the user. This factor was used to calculate the size and shape of the compressed image (See Equations 4.26, 5.9, and 5.10). This single factor was quite inflexible in terms of allowing the user to generate compressed images with a different rectangular shape than the original image. The compressed images always have the same width to height ratio as the original image.

More freedom could be gained by dividing the compression factor into horizontal and vertical components. This would allow different scaling factors to produce different compression ratios in either direction. Or, the user could supply a parameter that directly influenced the scaling factors, and the two new compression parameters could be generated internally with respect to this value. Theoretically, there is nothing mathematically complex hindering such an improvement.

8.6 Error Measures

Only one VR error measure was presented in this work. While this method appears superior to the one used in [89], no formal comparison was done. Other error measures, possibly based on a Gaussian curve as opposed to the logarithmic function might provide promising metrics.

8.7 Temporal Integration

Some research was done in the area of modelling of saccadic eye movements. Further study could analyze more complex models and attempt to integrate these into the existing videoconferencing application presented in Section 7.4. Temporal scene integration with foveae is an area that also appears to have promise.

8.8 Automatic Arbitrary Isosampling Map Modelling

Note that the tool in Figure 5.16 allows one to model arbitrary visual features, such as animate retinae, manually. This could possibly be automated to some degree. For example, given Figure 3.1, the optimum parameters for a multiple cooperative foveae transform could conceivably be mathematically determined. This would eliminate the need for trial and error model fitting. The process could also be extended beyond animate models to specific situations arising from computer vision applications such as videoconferencing or assembly line robot tasks.

8.9 ATM Network Implementation

One obvious extension to this research is to implement an actual application which utilizes foveal priority dithering of image data over an ATM network. Access to an ATM switch and network with a videophone prototype would be necessary.

As well, the current ATM simulation used a single static fovea. Future work could generalize this approach and incorporate *multiple* and *dynamic foveae*, where appropriate. Our research did not use any compression in the simulation, but combinations of both prioritization and compression could also be examined.

Chapter 9

Conclusion

Two related issues in the area of image processing and transmission are reducing the size of the data and dealing with uncontrollable data loss. Both are significant issues impacting many multimedia applications. It is important that research focus on the techniques designed specifically to handle new types of digital information. The use of images and image sequences is becoming more widespread, and biologically based image processing schemes seem promising.

A survey of biological visual systems introduced us to the concept of the *fovea*. This concept is found in the methods used by nature to compress image data as well as in those used to cope with data loss over internal animate optic networks. These methods have been modelled, both in general, and specifically focusing on features such as *anisotropism*, *multiple foveae*, *visual streaks*, *optic discs*, and *dynamic foveae*.

There are two main reasons why we may be inspired by animate visual systems:

- Animate visual systems are the only natural systems that are able to perform complex manipulations and operations with detailed optic data. The way they function, using images, may provide important clues to the nature of the visual data itself.

- Many of our image processing and compression methods will be used on data specifically intended for human viewing. Considering the user's visual system may allow us to tailor our algorithms, taking advantage of biological fixation and visual characteristics.

If we are open to the lessons taught to us while observing and modelling biological systems, we will find ourselves in a better position to address the specific requirements of computer vision in the future. As interest in foveated systems grows, it would be wise not to restrict ourselves to the concept of a single specialized fovea. Rather, we should avail ourselves of the plethora of possibilities expressed by the animal kingdom.

Unique approaches to image compression, utilizing the ideas of foveae and spatially variant sensors, have been considered. Original, substantive research in this area has been outlined. It has been shown that the area of primary interest, or fovea, can be favoured at the expense of the periphery of an image.

The benefits of VR approaches were demonstrated in several settings. Applications for a *mug shot database* and *videoconferencing system* were successfully implemented, clearly showing the benefit of these techniques. Novel hybrid compression standards were developed for improved performance in the specific domain of foveated scene and video compression. An ATM packet prioritization method was simulated, and the analysis showed that VR prioritization schemes provide improved results.

Variable Resolution (VR) algorithms modelled after biological systems have several advantages over traditional compression methods. Looking strictly at implementation issues, a variety of advantageous characteristics of VR compression include increased speed, hybrid possibilities, guaranteed minimum compression ratios, and network bandwidth adaptation capabilities:

- The VR compression methods, if designed with the use of look-up tables, require no complex calculations and, therefore, are extremely fast.
- The VR transforms operate entirely in the spatial domain; any other compression scheme can be run on the compressed images for greater compression ratios.
- Sampling can be adjusted to guarantee minimal compression ratios.
- Similarly, sampling can be adjusted to achieve constant bandwidth requirements.
- By monitoring network loads or by prioritizing data, bandwidth requirements can be varied to maximize network utilization.

- In all cases where cell loss will occur due to network congestion or bandwidth limitations, scene degradation is similar to that which may be experienced by the human visual system.

By modelling natural systems we have developed several transforms and algorithms with great potential to enhance computer vision in general and image compression specifically. Clearly, our models cannot be used to compress all data, as frequently the area of primary interest cannot be determined in advance (as with medical images). Thus, the distortions caused in such cases may be unacceptable. However, there are some applications, such as videoconferencing, where our techniques give superior compression ratios and acceptable image quality for relatively low computational cost.

The rate at which VR methods it can compress images, especially on machines with limited processing speed, and the high quality present in the foveal region, make it ideal for multimedia applications. Our experience with the *FaceBase* tool and videophone prototype supports this belief. A significant conclusion from the latter is that acceptable frame rates can be expected over either local or wide area networks without the need of additional hardware.

Some ideas on remaining areas of future research were outlined in Chapter 8 and included expanding the research to involve colour images and a greater number of popular established compression methods. As researchers continue to unlock the myriad of mysteries present in our own biological visual system, more new opportunities will unfold. One quickly begins to feel that this aspect of Computer Vision, as in many of the fields in Artificial Intelligence, is largely a profound exercise in plagiarism of the Designer.

Bibliography

[1] M. A. Ali. *Sensory Ecology: Review and Perspectives*, pages 226–257, 274–277, 502–515. Plenum Press, New York, 1978.

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

[3] J. Y. Aloimonos. Purposive and qualitative active vision. In *Proceedings of Image Understanding Workshop*, pages 816–828, 1990.

[4] Stephen J. Anderson, Kathy T. Mullen, and Robert F. Hess. Human peripheral spatial resolution for achromatic and chromatic stimuli: limits imposed by optical and retinal factors. *Journal of Physiology*, 442:47–64, 1991.

[5] Peng Ang, Peter Ruetz, and David Auld. Video compression makes big gains. *IEEE Spectrum*, 28(10):16–19, October 1991.

[6] R. Bajcsy. Active perception vs. passive perception. In *Proceedings of the Third IEEE Workshop on Computer Vision*, pages 55–59, 1985.

[7] R. Bajcsy. Active perception. *Proceedings of IEEE*, 76(8):996–1005, 1988.

[8] D.H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.

[9] Martin S. Banks, Allison B. Sekuler, and Stephen J. Anderson. Peripheral spatial vision: limits imposed by optics, photoreceptors, and receptor pooling. *Journal of the Optical Society of America*, 8(11):1775–1787, November 1991.

[10] A. Basu. Active calibration: Alternative strategy and analysis. *CVPR*, 93:495–500, 1993.

[11] A. Basu. Active calibration of cameras: Theory and implementation. *IEEE Transactions on systems, man, and cybernetics*, 25(2):456–465, February 1995.

[12] A. Basu and X. Li. A framework for variable resolution vision. In *Advances in Computing Information - ICCI 91*, pages 721–732, Ottawa, Canada, May 1991.

[13] A. Basu and S. Licardie. Modeling fish-eye lenses. In *IEEE IROS Conference*, Yokohama, Japan, July 1993.

[14] A. Basu and S. Licardie. Alternative models for fish-eye lenses. In *Pattern Recognition Letters*, pages 433–441, April 1995.

[15] A. Basu, A. Sullivan, and Kevin James Wiebe. Variable resolution teleconferencing. In *IEEE SMC Conference Proceedings*, France, October 1993.

[16] A. Basu and K. J. Wiebe. Videoconferencing using spatially varying sensing with multiple and moving foveae. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pages 30–34, Jerusalem, Israel, October 1994.

[17] Flaminio Borgonovo and Luigi Fratta. Policing in ATM networks: An alternative approach. *Proceedings of the 7th ITC Specialists Seminar*, 1990. Session 10.2, Morristown, New Jersey.

[18] E. T. Burtt. *The Senses of Animals*, pages 98–103. Wykeham Publications (London) LTD, 1974.

[19] R.H.S. Carpenter. *Movements of the Eyes.* Pion, London, 1977.

[20] CCIR Recommendation 500. Method for the subjective assessment of the quality of television pictures.

[21] Greg Cockroft and Leo Hourvitz. NeXTstep: Putting JPEG to multiple uses. *Communications of the ACM*, 34(4):45, April 1991.

[22] R. Coifman, Y. Meyner, Stevnen Quakne, and M. Victorh Wicnknerhauser. Signal processing and compression with wavelet packets. In *Proceedings of the IEEE.* Toulouse, 1992.

[23] CompuServe. *Graphics Interchange Format.* Springer-Verlag, first edition, 1987.

[24] Nils Conradi and Johan Sjöstand. A morphometric and steriologic analysis of ganglion cells of the central human retina. *Graefe's Archive for the Clinical and Experimental Ophthalmology*, 231:169–174, 1993.

[25] C. A. Curcio and K. A. Allen. Topography of ganglion cells in human retina. *Journal of Comparative Neurology*, 300:5–25, 1990.

[26] Christine A. Curcio *et al.* Aging of the human photoreceptor mosaic: Evidence for selective vulnerablility of rods in central retina. *Investigative Ophthalmology and Visual Science*, 34(12):3278–3296, November 1993.

[27] B. DeCleene and H. Sorenson. Prioritized subband coding of video for packet-switched networks. In *Proceedings of the 51st Midwest Symposium on Circuits and Systems*, August 1992.

[28] Philippe Delsarte, Benoit Macq, and Dirk T. M. Slock. Efficient multiresolution signal coding via a signal-adapted perfect reconstruction filter pyramid. *ICASSP '91*, pages 2633–3638, May 1991.

[29] R. DeVore, B. Jawerth, and B. Lucier. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*, 38(2):719–746, March 1992.

[30] David DiLorento Jr. and Christopher Cox *et al.* The influences of age, retinal topography, and gender on retinal degeneration in the fischer 344 rat. *Brain Research*, 647:181–191, 1994.

[31] H. Erikson. MBONE: The multicast backbone. In *Proceedings of INET '93*, pages F3.1–3.4, San Francisco, California, August 1993.

[32] Y. Fisher, D. Rogovin, and T. P. Shen. A comparison of fractal methods with DCT and wavelets. In *'94 San Diego SPIE Conference Proceedings.* SPIE, 1992.

[33] The ATM Forum. *ATM User-Network Interface Specification Version 3.0.* PTR Prentice Hall, third edition, September 1993.

[34] Ron Frederick. *Experiences with real-time software video compression.* Xerox Parc, July 1994. available online at ftp://parc-ftp.xerox.com/net-research/nv.

[35] Brian V. Funt. Conformal transplantation of lightness to varying resolution sensors. In *Proceedings of IEEE CVPR 93*, pages 563–569, 1993.

[36] Didier Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[37] G. Gallassi, G. Rigolio, and L. Fratta. ATM: bandwidth assignment and bandwidth enforcement policies. *IEEE Globecom 89*, pages 1788–1793, 1989.

[38] H. Gao and J. G. Hollyfield. Aging of the human retina. *Investigative Ophthalmology and Visual Science*, 33:1–17, 1992.

[39] Pawel Gburzynski and Shankar Gopalkrishnan. A layered video coding algorithm for multimedia applications over ATM. In *33rd Annual Allerton Conference on Communication, Control, and Computing*, October 1995.

[40] Rafael Gonzalez and Paul Wintz. *Digital Image Processing.* Addison Wesley, second edition, 1987.

[41] R. L. Gregory. *Eye and Brain: the psychology of seeing*, pages 57–63, 112–123. World University Library, 1981.

[42] S. S. Hacisalihzade, J. S. Allen, and L. W. Stark. Computer analysis of eye movements. *Computer methods and programs in biomedicine*, 40:181–187, 1993.

[43] B. P. Hayes and M. de L. Brooke. Retinal ganglion cell distribution and behaviour in procellariiform seabirds. *Vision Research*, 30(9):1277–1289, 1990.

[44] Harry Heffes and David M. Lucantoni. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on selected areas in communications*, SAC-4(6):856–867, September 1986.

[45] Joy Hirsch and Christine A. Curcio. The spatial resolution capacity of human foveal retina. *Vision Research*, 29(9):1095–1101, 1989.

[46] A. Hughes. The topography of vision in mammals of contrasting life style: Comparative optics and retinal organisation. In *Handbook of Sensory Physiology, Volume VII/5: The Visual System of Vertebrates*, pages 697–756, Springer Verlag, Berlin, 1977.

[47] Daniel P. Huttenlocher, Jae J. Noh, and William J. Rucklidge. Tracking nonrigid objects in complex scenes. CUCS TR 92-1320, Cornell University, Department of Computing Science, 1992.

[48] O. Inzunza and Hermes Bravo *et al.* Topography and morphology of retinal ganglion cells in *falconiforms*: A study on predatory and carrion-eating birds. *The Anatomical Record*, 229:271–277, 1991.

[49] A. Iwata, N. Mori, and C. Ikeda. ATM connection and traffic management schemes for multimedia internetworking. *Communications of the ACM*, 38(2):72–89, February 1995.

[50] Arnaud Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1):18–30, January 1992.

[51] David G. Jones, Richard C. Van Sluyters, and Katheryn M. Murphy. A computational model for the overall pattern of ocular dominance. *The Journal of Neuroscience*, 11(12):3794–3808, December 1991.

[52] S. S. Easter Jr. Retinal growth in foveated teleosts: Nasotemporal asymmetry keeps the fovea in temporal retina. *The Journal of Neuroscience*, 12(6):2381–2392, June 1992.

[53] Ronald Jurgen. Digital video. *IEEE Spectrum*, 29(3):24–30, March 1992.

[54] Ronald Jurgen and William Schreiber. All-digital tv's promise/problems. *IEEE Spectrum*, 28(4):28–30,71–73, April 1991.

[55] Matthew Kabrisky. *A Proposed Model for Visual Information Processing in the Human Brain*, pages 15–30. University of Illinois Press, first edition, 1966.

[56] B. G. Kim and P. Wang. ATM network: Goals and challenges. *Communications of the ACM*, 38(2):39–44, February 1995.

[57] K. Kirshfeld. Carotenoid pigments: Their possible role in protecting against photooxidation in eyes and photoreceptor cells. *Proceedings of the Royal Society of London*, 216:71–85, 1981.

[58] G. Kreider and J. Van der Spiegel, *et al.* The design and characterization of a space variant CCD sensor. In *SPIE Vol. 1381 Intelligent Robots and Computer Vision IX: Algorithms and Techniques*, Boston, 1990.

[59] X. Li. and A. Basu. Variable resolution character thinning. *Pattern Recognition Letters*, 12(4):241–248, 1991.

[60] Sergio Licarde. Variable resolution vergence control matching. Master's thesis, University of Alberta, 1993.

[61] C. Lindley. *Practical Image Processing in C.* Addison-Wesley, 1991.

[62] M. Liou. Overview of the p*64 kbit/s video coding standard. *Communications of the ACM*, 34(4):59–63, April 1991.

[63] J. N. Lythgoe. *The Ecology of Vision*, pages 6–15, 64–69, 92–95, 146–151. Oxford University Press, 1979.

[64] B. A. Makrucki. A study of source traffic management and buffer allocation in ATM networks. *Proceedings of the 7th ITC Specialists Seminar*, 1990. Session 5.5, Morristown, New Jersey.

[65] L. R. Marotte. Location of retinal ganglion cells contributing to the early imprecision in the retinotopic order of the developing projection to the superior colliculus of the wallaby (*marcropus eugenii*). *The Journal of Comparative Neurology*, 331:1–13, 1993.

[66] S. Panda-Jonas MD, J. B. Jonas MD, M. Jakobczyk, and U. Schneider MD. Retinal photoreceptor count, retinal surface area, and optic disk size in normal human eyes. *Opthalmology*, 101(3):519–523, 1994.

[67] W. H. Merigan and L. M. Katz. Spatial resolution across the macaque retina. *Vision Research*, 30(7):985–991, 1990.

[68] David B. Meyer. The avian eye and its adaptions. In *Handbook of Sensory Physiology, Volume VII/5: The Visual System of Vertebrates*, pages 549–611, Springer Verlag, Berlin, 1977.

[69] Brigette Müller and Leo Peichl. Horizontal cells in the cone-dominated tree shrew retina: Morphology, photoreceptor contacts, and topographical distribution. *The Journal of Neuroscience*, 13(8):3628–3646, August 1993.

[70] D. Murray and A. Basu. Motion tracking with an active camera. *T-PAMI*, 16:449–459, 1994.

[71] Mark Nelson. *The Data Compression Book.* M and T Books, 1992.

[72] Keith Oatley. *Brain Mechanisms and Mind*, pages 75–99. E. P. Duntton and Co., Netherlands, 1972.

[73] L. Peichl. Topography of ganglion cells in the dog and wolf retina. *The Journal of Comparative Neurology*, 324:603–620, 1992.

[74] William K. Purves and Gordon H. Orians. *Life: The Science of Biology*, pages 680–688. Sinauer Associates, Inc., 1982.

[75] Xiaolin Qiu. A study in real-time collaboration systems. Master's thesis, University of Alberta, June 1996.

[76] Erwin P. Rathgeb. Policing mechanisms for ATM networks - modelling and performance comparison. *Proceedings of the 7th ITC Specialists Seminar*, 1990. Session 10.1, Morristown, New Jersey.

[77] Erwin P. Rathgeb. Policing of realistic VBR video traffic in an ATM network. *International Journal of Digital and Analog Communication Systems*, 6:213–226, 1993.

[78] Andreas Reichenbach and Mathias Zeigert *et al.* Development of the rabbit retina. *Developmental Brain Research*, 79:72–84, 1994.

[79] A. S. Rojer and E. L. Schwartz. Design considerations for a space-variant visual sensor with complex-logarithmic geometry. In *Proc. 10th International Conference on Pattern Recognition, volume II*, pages 278–285, Atlantic City, 1990.

[80] Oliver Sacks. *An Anthropologist on Mars.* Vintage Canada, 1996.

[81] G. Sandini and P. Dario. Active vision based on space-variant sensing. In *Proc. 5th Int. Symp. on Robotics Research*, pages 75–83, Tokyo, 1990.

[82] G. Sandini and V. Tagliasco. An anthropomorphic retina-like structure for scene analysis. *Computer Graphics and Image Processing*, 14:365–372, 1980.

[83] G. Sandini and M. Tistarelli. Vision and space-variant sensing. In *ECCV-94 Workshop on Natural and Artificial Visual Sensors*, pages 398–425, Osquldа's vag 6, Stockholm, Sweden, May 1994.

[84] Eric L. Schwartz. Spatial mapping in the primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194, 1977.

[85] Eric L. Schwartz. Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding. *Vision Research*, 20:645–669, 1980.

[86] S. Sinclair. *How Animals See*, pages 2, 50, 89–94, 98–103. Library of Congress, 1985.

[87] Kai-Yeung Siu and Hong-Yi Tzeng. Intelligent congestion control for ABR service in ATM networks. *Computer Communication Review*, 24(5):81–106, October 1995.

[88] J. Stone. *Parallel Processing in the Visual System*. Plenum Press, New York, 1983.

[89] A. Sullivan. Variable resolution image compression. Master's thesis, University of Alberta, Dept. of Computing Science, 1993.

[90] M. J. Swain and M. Stricker (eds.). Promising directions in active vision. *International Journal of Computer Vision*, 11(2):109–126, 1993.

[91] S. M. Sykes, W. G. Robinson, M. Waxler, and T. Kuwabara. Damage to the monkey retina by broad-spectrum fluorescent light. *Investigative Ophthalmology and Visual Science*, 20:425–434, 1981.

[92] Á. Szél and G. Csorba *et al.* Different patterns of retinal cone topography in two genera of rodents, *mus* and *apodemus*. *Cell and Tissue Research*, pages 143–150, 1994.

[93] Kannan Thiruvengadam. Scalability in media-on-demand systems. Master's thesis, University of Alberta, June 1995.

[94] M. Tistarelli and G. Sandini. Estimation of depth from motion using an anthropomorphic visual sensor. *Image and Vision Computing*, 8(4):271–278, 1990.

[95] M. Tistarelli and G. Sandini. On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(4):401–410, 1993.

[96] F. Tong and Z.N. Li. The reciprocal-wedge transform for space-variant sensing. In *Proc. Int. Conf. on Computer Vision (ICCV '93)*, pages 330–334, 1993.

[97] F. Tong and Z.N. Li. Reciprocal-wedge transform in motion stereo. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1060–1065, San Diego, 1994.

[98] J. K. Tsotsos. On the relative complexity of active vs. passive visual search. *International Journal of Computer Vision*, 7(2):127–141, 1992.

[99] Thierry Turletti and J. C. Bolot. Issues with multicast video distribution in heterogeneous packet networks. In *Proceedings of the 6th International Workshop on Packet Video*, pages F3.1–3.4, Protland, Oregon, September 1994.

[100] Robert Ulichney. *Digital Halftoning*. The MIT Press, second edition, 1988.

[101] J. Van der Spiegel, *et al.* A foveated retina-like sensor using CCD technology. In C. Mead and M. Ismail, editors, *Analog VLSI Implementation of Neural Systems*, pages 189–211. Kluwer, 1989.

[102] Ronald J. Vetter. ATM concepts, architectures, and protocols. *Communications of the ACM*, 38(2):30–38, February 1995.

[103] Brandi Vidaković and Peter Müller. *Wavelets for Kids: A Tutorial Introduction.* Duke University, 1991. available online at ftp://ftp.isds.duke.edu/pub/Users/brani/papers/wav4kidsA.ps.Z.

[104] Gregory Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):31–44, April 1991.

[105] Heinz Wässle, Ulrike Grünert, and Jürgen Röhrenbeck *et al.* Cortical magnification factor and the ganglion cell density of the primate retina. *Nature*, 34:643, October 1989.

[106] C.F.R. Weiman and G. Chaikin. Logarithmic spiral grids for image processing and display. *Computer Graphics and Image Processing*, 11:197–226, 1979.

[107] Terry Welch. A technique for high performance data compression. *IEEE Computer*, 17(6):645–669, June 1984.

[108] Kevin James Wiebe. Video Presentation: spatially varying sensing for multimedia teleconferencing. In *AI/GI/VI '94*, Banff, Alberta, Canada, May 1994.

[109] Kevin James Wiebe and A. Basu. Improving image and video transmission quality over ATM with foveal prioritization and priority dithering. In *Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, August 1996. ICPR.

[110] Kevin James Wiebe and A. Basu. Modelling ecologically specialized biological visual systems. *Pattern Recognition Journal*, 1996. to appear.

[111] Kevin James Wiebe and V. Nehru. Poster: variable resolution image compression. In *Research Revelations '94*, University of Alberta, March 1994.

[112] John Woodfill. *Motion Vision and Tracking for Robots in Dynamic, Unstructured Environments.* PhD thesis, Stanford University, 1992.

[113] A.L. Yarbus. *Eye Movements and Vision.* Plenum, New York, 1967.

[114] B. L. Zuber. *Models of Oculomotor Behaviour and Control.* CRC Press, Boca Raton, Florida, 1981.

Appendix A

Diagram of the Human Eye

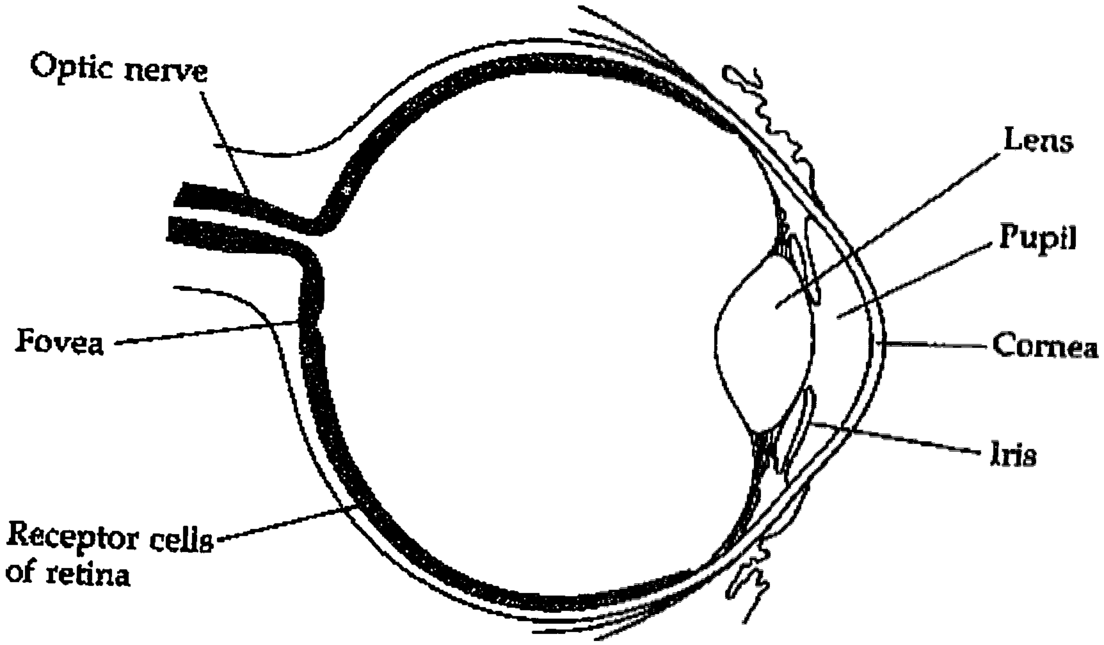


Figure A.1: Diagram of the human eye.

Diagram A.1[1] is of a horizontal section through the eyeball, to detail the primary components and to show position of derivatives of its three primary layers.

[1] This figure was reprinted from [74] with permission.

Appendix B

SVR/PGM Image Header Format

Below is a sample header from an image in the hybrid SVR/PGM image format.

```
P5
# VR3 Image by Kevin Wiebe.
# Compression Parameters:
# VR3_begin
# c 70 (compression value)
# a 200 (alpha value)
# s 340 340 (decompressed size: rows, columns)
# p 1 (combining power)
# n 3 (number of foveae)
# f 132 146 (position of fovea 1: x y)
# f 186 130 (position of fovea 2: x y)
# f 158 199 (position of fovea 3: x y)
# m 2 (method: cooperative)
# VR3_end
186 186
255
```

Appendix C

CVR/MPEG Parameter File Format

Below is a sample parameter file for compressing an image sequence in the hybrid CVR/MPEG format.

```
# salesman parameter file

PATTERN      IBBPBBPBB
OUTPUT       salesman.mpg

BASE_FILE_FORMAT    PNM
INPUT_CONVERT    rasttopnm * | pgmnorm | pnmgamma 2
GOP_SIZE     30
SLICES_PER_FRAME  1

INPUT_DIR    /usr/jasper/grad/kevin/VC/MPEG/salesman

INPUT
gs* [000-299]
END_INPUT

PIXEL        HALF
RANGE        10

PSEARCH_ALG LOGARITHMIC
BSEARCH_ALG CROSS2

IQSCALE      8
PQSCALE      10
BQSCALE      25

USE_CVR      PAD
CVR_COMP     95
CVR_ALPHA    200
# Here is the current choice for foveae locations:
CVR_FOVEA    DYNAMIC FACE 200

# Here are some other choices for foveae locations:
#    His Face
```

```
#CVR_FOVEA   STATIC 208 104
#    His Object
#CVR_FOVEA   STATIC 124 131
#CVR_FOVEA   DYNAMIC MOTION 30 4
#CVR_FOVEA   DYNAMIC CENTER
#CVR_FOVEA   DYNAMIC FILE cvrsale.fov

FORCE_ENCODE_LAST_FRAME

REFERENCE_FRAME DECODED
# Default string has the date in it, bad for tests!
USER_DATA  /dev/null
```