University of Alberta

Stereo Image Compression using Vergence and Spatially Varying Sensing

by

Warren Wong ©

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

Department of Computing Science

Edmonton, Alberta
Fall 1996

ISBN 0-612-18335-1

University of Alberta

Library Release Form

Name of Author: Warren Wong

Title of Thesis: Stereo Image Compression using Vergence and Spatially Varying Sensing

Degree: Master of Science

Year this Degree Granted: 1996

Warren Wong

Warren Wong
16219 - 114 Street
Edmonton, Alberta
Canada, T5X 2L9

Date: OCT. 2, 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 **Stereo Image Compression using Vergence and Spatially Varying Sensing** submitted by Warren Wong in partial fulfillment of the requirements for the degree of **Master of Science**.

Dr. A. Basu

Z J Koles

Dr. Z. Koles

Xiaobo Li

Dr. X. Li

Date: 9/30/96

Abstract

Stereo image compression deals with compression of a pair of images with one image being a left view and the second being a right view. These pairs of images are commonly known as stereopairs and give better depth information on a scene than a single image can provide. Ideally, stereopairs are viewed on a head mounted display (HMD) to achieve the proper effect of depth perception. Since HMDs place images close to the eyes, one can attempt to take advantage of the human visual system to try to achieve better compression results. To enhance depth perception for the human visual system, the concept of vergence will be used to obtain stereopairs. To investigate the effects of non-uniform quality loss, the concept of spatially varying sensing will be used during compression. The lossy compression methods presented will also use the discrete cosine transform (DCT).

Acknowledgements

First of all, thanks to my supervisor, Anup Basu, for his guidance, ideas, free food, and other help throughout the course of my thesis work. Thanks also go to the following people:

- Jonathan Baldwin for helping me with the capturing of the stereopairs used in this thesis
- Kevin Wiebe for his help in using his variable resolution compression code
- Anne Nield for proofreading my thesis
- all the people that took the time to participate in my user survey, and
- all my supportive friends that kept life interesting.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Head mounted displays (HMDs) are excellent devices to view stereopairs. Already, there are HMDs available for personal computers and, as they become more affordable, there may be a demand for efficient transmission and/or storage of stereopairs. To reduce the amount of data that needs to be transferred or stored, some type of compression must be used. Currently, there are no standard stereopair compression schemes equivalent in popularity to a compression scheme such as the JPEG compression standard for single images. The lack of a standard is mainly due to the fact that there are not many HMDs in use with which to view stereopairs. This may change in the near future if HMDs for personal computers become more popular, so some exploration into the area of stereopair compression for HMDs needs to be done. In the cases where stereopair quality loss is unavoidable, due to space or bandwidth limitations, we would like the loss to occur in a way that least affects the appearance of the stereopair through the HMD. Uniform loss of quality works very well for single images viewed on a screen, but this may not be the case when viewing stereopairs with a HMD. The possibility of using existing hardware to assist the compression would be a good feature since there are compression cards available for personal computers, and some personal computer packages already include compression hardware such as MPEG (Motion Picture Experts Group) [11] boards. Parameters that control the amount of compression would also be desirable, since transmission speeds are different depending on the hardware and connection speed available. This thesis will present two methods of stereo image compression that allow control over the amount of compression and address the issue of quality loss that is less evident to the human visual system when viewed through a HMD.

1.2 Thesis Organization

Chapter Two will provide the necessary background information for understanding the reasoning behind the compression methods used in this thesis. This chapter includes explanations of

- the goals of stereo image compression
- vergence and spatially varying sensing,
- the discrete cosine transform and its usefulness in image compression, and
- different evaluation methods of compression.

Chapter Three will describe three stereo compression methods. The first method will be a base method used for comparison purposes against the other two compression methods that combine spatially varying sensing and the discrete cosine transform. Chapter Four will discuss issues involving the addition of color support to the image compression methods from Chapter Three. Chapter Five analyzes the performance of the three compression methods by presenting results of the compression for small, medium, and high compression ratios, along with results from a small user survey. Chapter Six presents some conclusions from the results of the three compression methods. Chapter Seven suggests some future work that may be attempted to extend the compression methods in different ways, or to improve their performance.

Chapter 2

Background Information

Image compression comes in two forms: lossless and lossy. Lossless compression compresses images so that the original image can be reconstructed exactly. Lossy compression compresses images so that the original image can be reconstructed approximately, with some loss in image quality. This thesis will be concerned with lossy compression.

The class of images being compressed are commonly called stereopairs. Stereopairs consist of a pair of images that correspond to a left view and a right view of an object or scene. Using single image lossy compression methods such as the JPEG (Joint Photographic Experts Group) baseline standard [14, 17, 24], one can compress the left and right images of a stereopair separately. Both compressed images would need approximately the same storage space; however, there are redundancies in a stereopair that can be used. In stereo image compression, the objective is to store the image in less space than that required if the left and right images were compressed separately.

2.1 Vergence and spatially varying sensing

A stereopair was defined as left and right views of an object or scene. Assuming that two cameras are used to create a stereopair, this definition allows for the possibility of choosing the orientation of the cameras. Camera geometry [18] will be the term used to describe the orientation of the cameras. Before describing camera geometries, a few terms must be defined. The center of a camera's lens will be referred to as the camera's focal center, and a plane parallel to the camera lens will be referred to as the image plane. The image captured by the camera is a rectangular portion of the image plane and is referred to as the image area. A ray extending from the focal center, perpendicular to the image plane, is referred to as the camera's focal ray. Figure 2.1 shows the top view of a commonly used camera geometry for stereopairs called the parallel axes geometry. In the parallel axes geometry, the left and right focal rays are

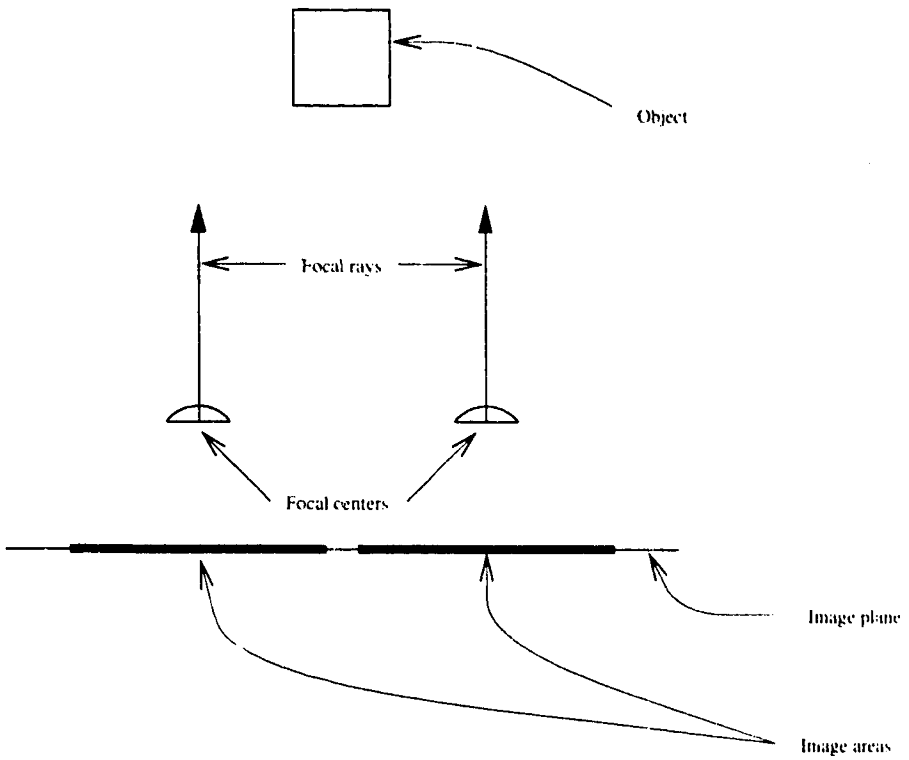


Figure 2.1: Parallel axes geometry

parallel, the left and right image planes are coplanar, and the bottom edge of the left and right image areas are collinear. Ideally, the two cameras should be at the same vertical distance from the ground. Stereopairs using the parallel axes geometry allow depth perception, but there may be some uncertainties in the depth estimation. The parallel axes geometry is a convenient camera geometry to use, but we would like to reduce the depth estimation error.

Cameras capture images, which are transformed into discrete pixels. With most cameras, these pixels are uniformly distributed over the camera's image area according to some industry standard. In order to place the object or scene into the discrete number of pixels, the projections of the 3D points must be approximated to the nearest pixel. The approximations generate what is called discretization error and, in stereopairs, discretization error leads to depth estimation error. Figure 2.2[1] illustrates this depth estimation error which increases as the distance from the cameras increases. As one can see, the error is not a simple function of distance since the diamonds have different sizes, and orientations. When estimating the depth of the point, the depth estimated can be any one of the four points defining the diamond, so reducing the size of the diamonds around the point of interest would be desirable.

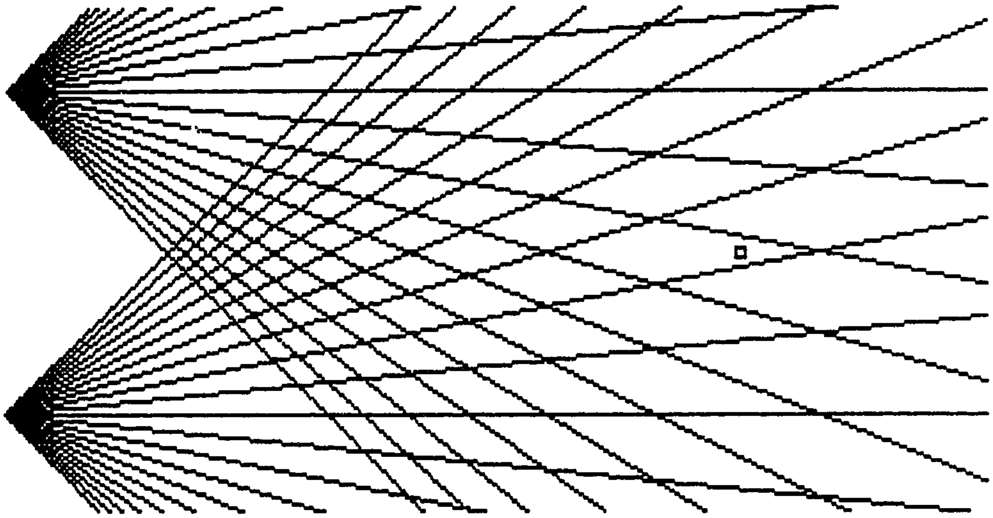


Figure 2.2: Depth estimation error with parallel axes geometry

Sahabi and Basu have used vergence and spatially varying sensing to reduce the depth estimation error [20]. Vergence involves the use of a camera geometry which will be referred to as the vergent axes geometry. In the vergent axes geometry, the left and right focal rays converge on the object of interest, and this creates two image planes with two image areas. The bottom edges of both image areas are no longer collinear, but there is the requirement that the bottom edges of the image areas are on the same plane. The angle that the left and right focal rays make to their corresponding parallel rays in the parallel axes geometry will be referred to as the vergence angle. Figure 2.3 illustrates the top view of the vergent axes geometry with vergence angle θ. Spatially varying sensing deals with the fact that eyes have high resolution in some areas and low resolution in other areas of the visual field. The area around a point called the fovea has high resolution, while the areas with low resolution are called the periphery. In the case of the human eye, the field of view is very large (close

[1] Figure taken from [20]

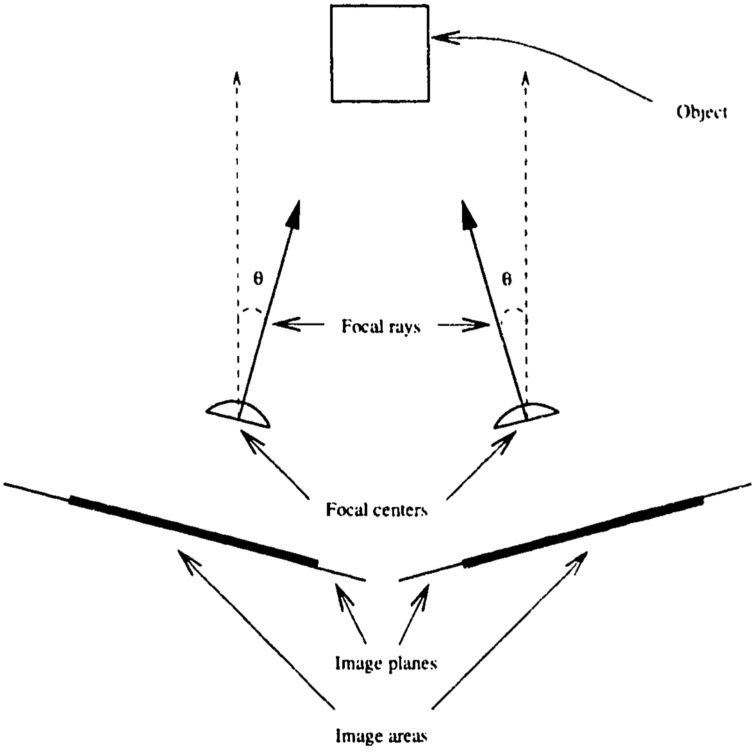


Figure 2.3: Vergent axes geometry with vergence angle θ

to 180°), but there is only one fovea in the center of the visual field. This means that the human eye can focus sharply on a limited area of interest while everything else appears blurred. The work done by Sahabi and Basu found that using cameras with high resolution in the center and nonlinearly decreasing resolution further away from the center reduces depth uncertainty when vergence is used. Figures 2.4 and 2.5[2] illustrate the difference between using vergence with a uniform resolution and vergence with their non-uniform resolution. With the higher resolution used in the region containing the point of interest (namely the center of both images), the depth uncertainty is significantly reduced. Using vergence and spatially varying sensing, the human visual system is modelled much better than using the parallel axes geometry and uniform resolution. When we look at a nearby object, our eyes do not simply look straight ahead as in the parallel axes geometry. Instead, the foci of both eyes converge on the object of interest to give us more accurate depth information. Figures 2.6 and 2.7 show examples of a stereopair taken with the parallel axes geometry and

[2] Figures taken from [20]

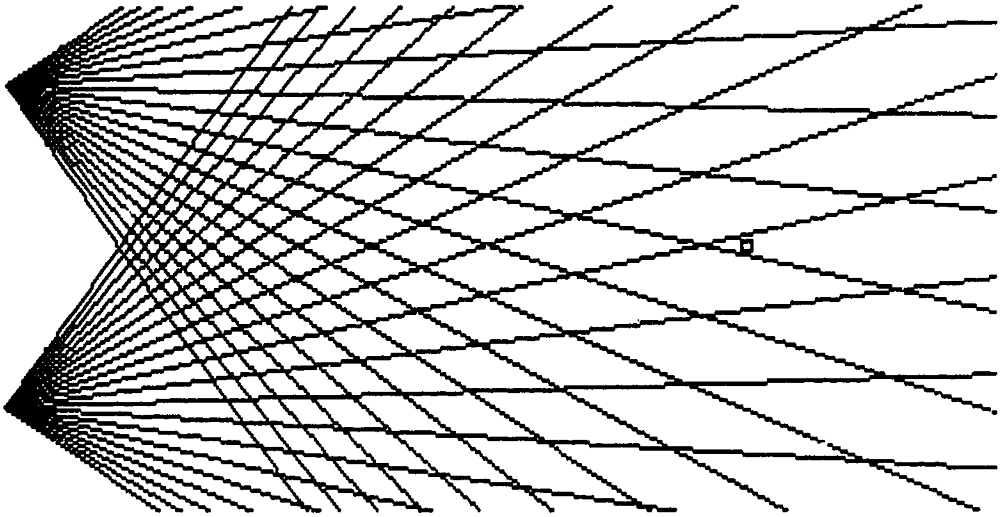


Figure 2.4: Depth uncertainty using uniform resolution

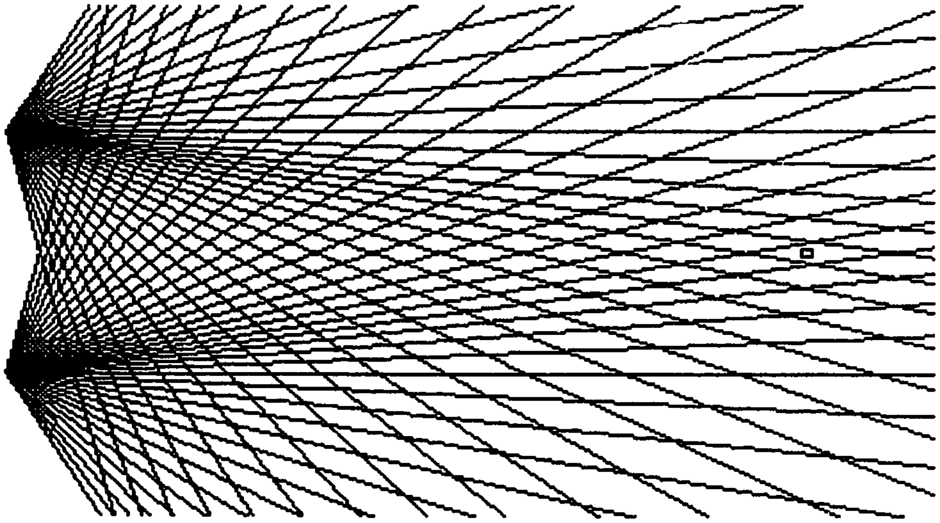


Figure 2.5: Depth uncertainty using non-uniform resolution

the vergent axes geometry respectively. The angle of vergence used in Figure 2.7 is 5°.

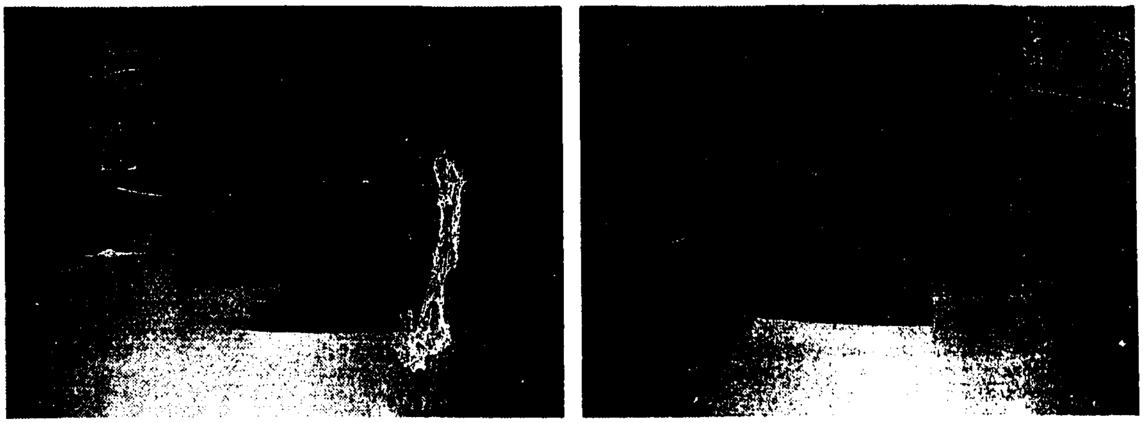


Figure 2.6: Parallel axes geometry stereopair

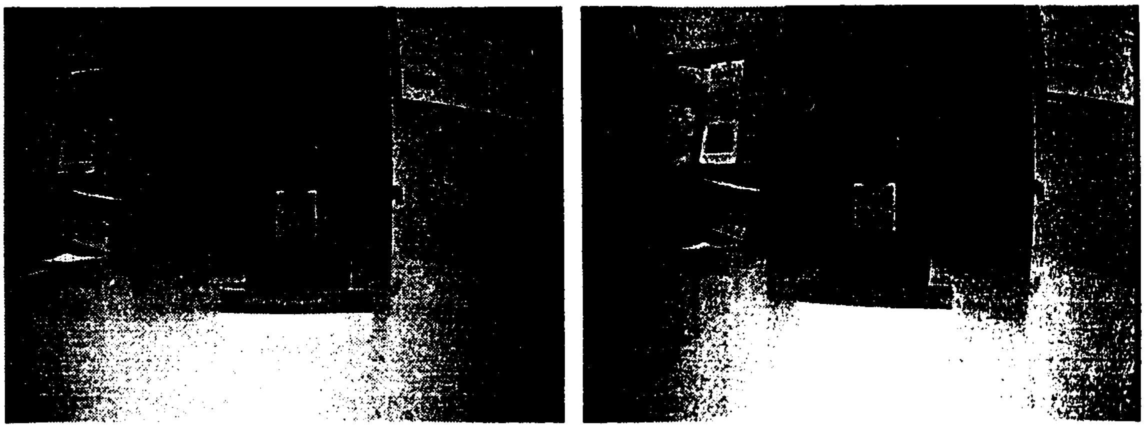


Figure 2.7: Vergent axes geometry stereopair

A transform called the variable resolution transform [1, 2] (from now on referred to as the VRT) was used to implement the idea of compression based on spatially varying sensing. The VRT has two parameters that control compression: one parameter controls the expected compression ratio; the other controls the resolution around the fovea. The first parameter will be referred to as scaling factor s, and the second parameter will be referred to as α. Using the polar coordinate system, with the center of an image being the origin, the transform maps a pixel with coordinates of (r, θ) to its new coordinates of (v, θ). The equation for v is given in Equation 2.1 and the inverse transform is given in Equation 2.2.

$$v = s\ln(\alpha r + 1) \tag{2.1}$$

$$r = \frac{\exp\left(\frac{v}{s}\right) - 1}{\alpha} \tag{2.2}$$

$$0 \leq \alpha \leq 1$$

The s value is calculated using the maximum distance a pixel can be from the fovea (r_{max}), and the maximum distance one would like this pixel to be from the fovea in the transformed image (v_{max}). r_{max} depends on where the fovea is located in the image. For example, if the fovea is located in the center of a MxN image, r_{max} would be $\sqrt{\frac{M}{2}^2 + \frac{N}{2}^2}$. v_{max} is defined by the user to specify how much compression is wanted. The closer v_{max} is to r_{max}, the less compression there is. Equation 2.3 gives the equation for s.

$$s = \frac{v_{max}}{\ln(\alpha r_{max} + 1)} \tag{2.3}$$

There is a small problem with the VRT in that the transformed image is not rectangular. When storage in a rectangular field is preferred – usually the case when working with computers – clipping or padding with unused pixels may be necessary. To overcome this problem, a modification of the VRT called the Cartesian variable resolution transform (CVRT) has been used in [1]. Advantages of the CVRT is that it is computationally less expensive than the VRT and it produces a rectangular transformed image. A disadvantage is that the CVRT is less accurate than the VRT with respect to the model of the human eye. The equations for the CVRT are given in Equations 2.4 to 2.11. The CVRT uses the Cartesian coordinate system rather than the polar coordinate system. The equations are basically the same as the VRT, but now there are different scaling factors for the x and y directions while α remains the same. Consider an image with the fovea located at (x_0, y_0). The distances from a pixel (x, y) to the fovea are defined in the x and y directions as dx and dy, respectively, given in Equations 2.4 and 2.5.

$$dx = x - x_0 \tag{2.4}$$

$$dy = y - y_0 \tag{2.5}$$

This pixel, which is a distance of dx and dy away from the fovea, is now moved to a distance of v_x and v_y away from the fovea. Equations 2.6 and 2.7 show how to compute v_x and v_y.

$$v_x = s_x \ln(\alpha\, dx + 1) \tag{2.6}$$

$$v_y = s_y \ln(\alpha\, dy + 1) \tag{2.7}$$

To reverse the transform, compute dx and dy from v_x and v_y using Equations 2.8 and 2.9.

$$dx = \frac{\exp\left(\frac{v_x}{s_x}\right) - 1}{\alpha} \tag{2.8}$$

$$dy = \frac{\exp\left(\frac{v_y}{s_y}\right) - 1}{\alpha} \tag{2.9}$$

The values of s_x and s_y are given in Equations 2.10 and 2.11. v_{xmax} and v_{ymax} have the same function as v_{max} in the VRT, and dx_{max} and dy_{max} have the same function as r_{max} in the VRT.

$$s_x = \frac{v_{xmax}}{\ln(\alpha\, dx_{max} + 1)} \tag{2.10}$$

$$s_y = \frac{v_{ymax}}{\ln(\alpha\, dy_{max} + 1)} \tag{2.11}$$

Pixels at the maximum x and y distance from the fovea in the original image are at the maximum possible x and y distance from the fovea in the transformed image just as with the VRT. Figure 2.8 shows the standard Lena image[3] that will be used to illustrate how the CVRT works. Selecting the fovea location to be approximately the center of the face, the image in Figure 2.9 is obtained after applying the CVRT. An α value of 0.1 and a compression ratio of 82% was used. The original file size is 65637 bytes and the compressed file size is 11554 bytes. Both images are in PGM format which basically stores the width and height of the image followed by the actual pixel values. Applying the regular VRT produces a circular "fish-eye lens" type image, but using the CVRT, it appears that the fish-eye image has been stretched at the corners. When reconstructing the image, many interpolation methods can be used, depending on what is needed. If processing speed is a factor then nearest neighbour interpolation is ideal, but the resulting image may be less appealing than if a more

[3] This image was obtained from the University of East Anglia Signal and Image Processing group's standard image page at http://www.sys.uea.ac.uk/research/resgroups/sip/images_ftp/index.html.

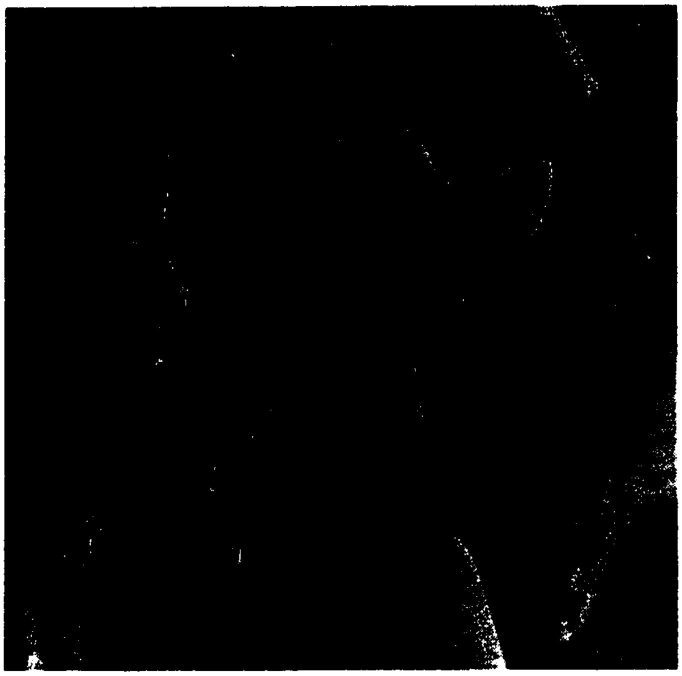


Figure 2.8: Original image

Figure 2.9: Compressed image ($\alpha = 0.1$, 82% compression)

complicated interpolation method is used. Reconstructing the Lena example image using bilinear interpolation produces the image shown in Figure 2.10. As one can see, the area of interest, namely the face, retains higher resolution than the remainder of the image. The result is a reconstructed image with the face remaining clear at approximately 82% compression.

Figure 2.10: Reconstructed image

2.2 The Discrete Cosine Transform

2.2.1 Choosing a good transform

An image can be considered as a two dimensional function with the values of the function being the image intensities at their respective Cartesian coordinates x and y. The image in this form is said to be in the spatial domain. Compression based on the VRT is an example of a compression method in the spatial domain. Some lossy compression methods make use of what is called transform coding [6]. In transform coding, the image is transformed from the spatial domain onto a set of "transform coefficients" using a reversible, linear transform. Some of these transform coefficients may be very small in magnitude and simply discarded, or some could be quantized without any noticeable loss in quality once the reverse mapping or transformation is applied. The compression occurs through the discarding and quantization of the transform coefficients. This ability to quantize and discard certain coefficients makes transform coding a good method of eliminating psychovisual redundancy from images.

The discrete cosine transform (from now on referred to as the DCT) is one of many transforms that are used in lossy image compression. Other transforms, including the discrete Fourier transform (DFT), the Walsh-Hadamard transform (WHT), and the Karhunen-Loève transform (KLT) can be used for image compression as well. Important deciding factors when choosing a transform are how well the transform "packs" information into a few coefficients, and how convenient the transform is to use. From the viewpoint of minimizing the MSE with the fewest transform coefficients, the KLT is known to be optimal [6, 19]. However, when convenience of use is considered, the KLT is data dependent involving the finding of eigenvectors and their corresponding eigenvalues. Because of the complexity of computing the KLT, it is rarely used in practice for image compression, but it is used to compare other transforms to see how close they come to being optimal. The WHT, DFT, and the DCT are all data independent so they have fixed basis images. The DFT and DCT fit into a class of transforms called "sinusoidal" transforms since their 1D basis functions resemble sine waves. Sinusoidal transforms are closer approximations to the KLT [12] than nonsinusoidal transforms for real world images because of the high inter-pixel correlations. The WHT is a nonsinusoidal transform, so it does not work as well as the DFT or DCT for the real world images being used. The DFT is a well known transform and is valuable for spectral analysis and filtering. However, it is not popular for transform coding of images mainly because of the possible discontinuities involved when transforming a set of samples from an image. These discontinuities can be seen for the 1D case in Figure 2.11. The DFT of the segment of 8 sample points is a transform domain representation of the segment being repeated. The discontinuities may produce spurious content in the high frequency components of the transform coefficients. For image coding, the spurious information in the transform coefficients combined with quantization may lead to visible boundaries between subimages with the boundary pixels possibly becoming the mean values of the pixels involved in the discontinuities. The DCT does not have the discontinuity problem that the DFT has

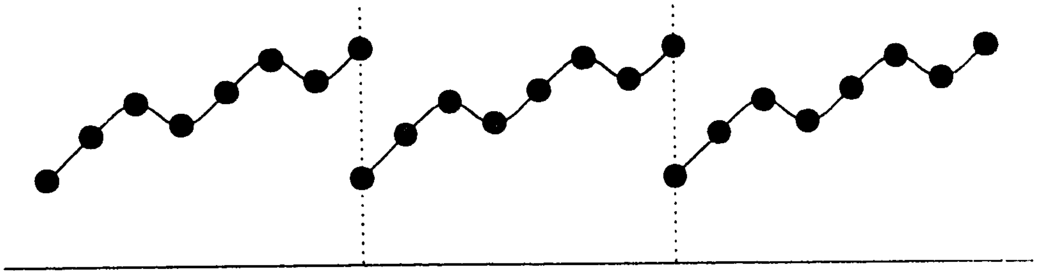


Figure 2.11: Discontinuities inherent in the DFT

since it makes the data to be transformed symmetric. This symmetry is illustrated in the 1D case in Figure 2.12. The segment is folded about the vertical axis, so when this new segment is repeated, there are no discontinuities. Because of the elimination of

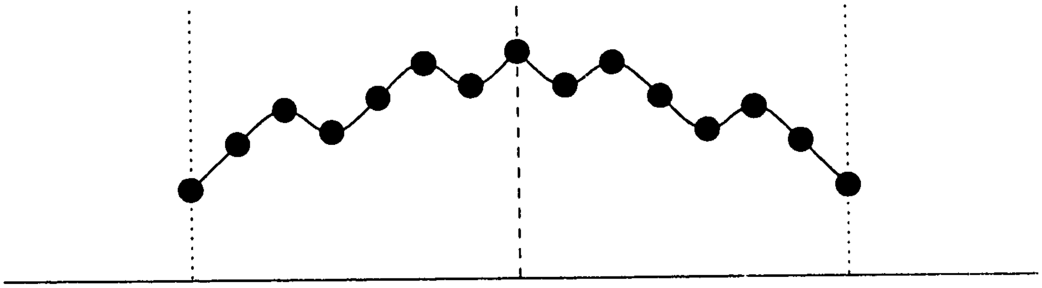


Figure 2.12: Elimination of discontinuities in the DCT

the discontinuities and the fact that the DCT involves only real numbers as opposed to complex numbers in the DFT, the DCT has been the choice of many people when using a transform for coding images. One of the more commonly known compression methods that use the DCT is the JPEG baseline compression method [14, 17, 24]. The DCT has also been used to perform stereo image compression, with good results [7, 8, 15]. Another deciding factor in using the DCT is the growing popularity of MPEG compression hardware in personal computers. MPEG also uses the DCT and it may be possible to take advantage of the hardware to quickly compute the DCT. This thesis will use the DCT as its method of transform coding since it is a good compromise between information packing efficiency and computational complexity.

2.2.2 Using the DCT

Consider an 8x8 block as a function f which depends on the two spatial dimensions x and y - both in the range $[0, 7]$. The forward DCT function C of an 8x8 block is defined over the parameters u and v as in Equation 2.12. Both u and v are also in the range $[0,7]$.

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\left(\frac{(2x+1)\pi u}{16}\right)\cos\left(\frac{(2y+1)\pi v}{16}\right) \tag{2.12}$$

$$\alpha(w) = \begin{cases} \frac{1}{\sqrt{8}} & , \quad w = 0 \\ \frac{1}{2} & , \quad w \neq 0 \end{cases}$$

The 64 sample points in the 8x8 block f are in the spatial domain. The DCT transforms the 64 sample points into 64 DCT coefficients in the 8x8 block C. Each coefficient can be thought of as a weight to one of 64 cosine basis images. The basis images are in the frequency domain with increasing "spatial frequency" as u and v increases. Figure 2.13 illustrates the cosine basis images graphically in the spatial domain with white being highest value and black being the lowest. The upper left block corresponds to the lowest spatial frequency of zero. It is one constant value because zero frequency means there is no change in the 8x8 block. Moving from the upper left to the lower right basis images yields more vertical zero-crossings while moving to the right, and more horizontal zero-crossings while moving to the bottom. When the bottom right basis image is reached, there is a maximum combination of horizontal and vertical zero-crossings corresponding to a high frequency of change in both the x and y directions in the spatial domain.

In the ideal case where all calculations are done with infinite precision, the 64 DCT coefficients can be used to reconstruct exactly the 64 sample points that were transformed. The inverse DCT for a block of size 8x8 is given in Equation 2.13.

$$f(x,y) = \sum_{u=0}^{7}\sum_{v=0}^{7} \alpha(u)\alpha(v)C(u,v)\cos\left(\frac{(2x+1)\pi u}{16}\right)\cos\left(\frac{(2y+1)\pi v}{16}\right) \tag{2.13}$$

$$\alpha(w) = \begin{cases} \frac{1}{\sqrt{8}} & , \quad w = 0 \\ \frac{1}{2} & , \quad w \neq 0 \end{cases}$$

One of the advantages of the DCT is that the forward and inverse transform are so similar. This is a useful feature of any transform in terms of understanding and implementation.

So far there has been no compression. What has been done is a transformation from the spatial domain to the frequency domain. While this transformation does no compression, it does compact the majority of the "signal energy" of the input block

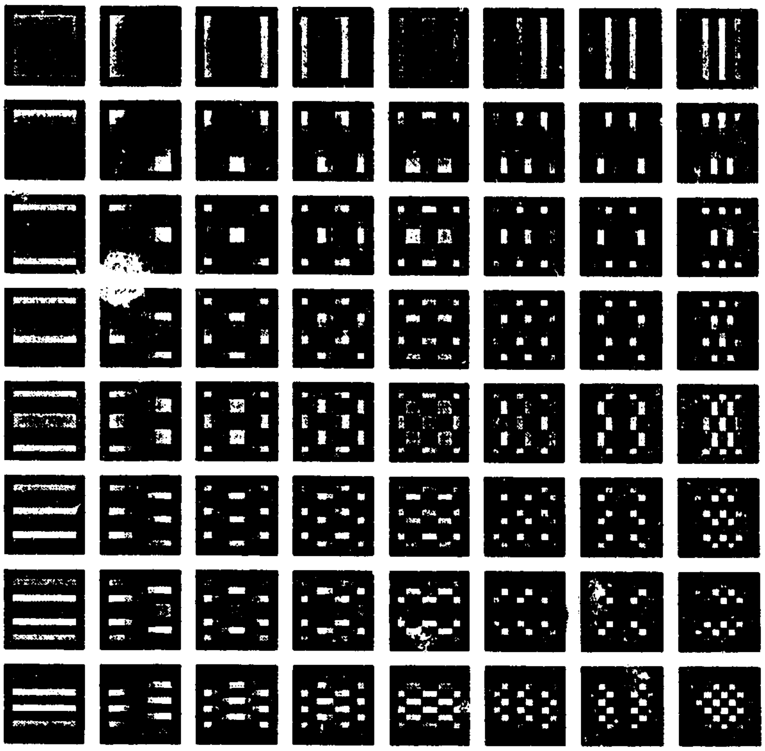


Figure 2.13: 8x8 DCT cosine basis images

into a few coefficients. Most computers have sufficient precision to reconstruct the 8x8 block back to its original state. The lossy part of the compression occurs when the DCT coefficients are quantized. Some of the coefficients can be quantized more coarsely than others. The human eye is less sensitive to high frequency changes, so referring to Figure 2.13 for our DCT basis images, the coefficients for the basis images near the lower right corner can be quantized more coarsely than for those near the upper left corner. The quantization usually results in a few non-zero coefficients for the upper left corner basis images, and the rest of the coefficients end up quantized to zero. The quantized coefficients can then be encoded using an encoding method such as Huffman coding or arithmetic coding in order to try to minimize the number of bits necessary to store the coefficients. The amount of quality loss depends on how heavily the DCT coefficients are quantized.

2.2.3 DCT Example

An example of compression using the DCT is given in [12]. The example comes from the JPEG baseline standard. Consider a typical 8x8 block of pixels such as the one in Figure 2.14(a). There are a possible 256 gray levels in this block. The process begins by first level shifting the pixels of the block by -2^7 or -128 gray levels. Now the values range from -128 to 127 instead of from 0 to 255 and the result of the shifting is in Figure 2.14(b). After the shifting the forward DCT (FDCT) is applied on the block of pixels, resulting in the coefficients in Figure 2.15(a). Using the JPEG recommended quantization matrix in Figure 2.15(b), the DCT coefficients are quantized to the set of coefficients in Figure 2.16. Note that most of the non-zero coefficients are in the upper left, while the lower right coefficients have all been quantized to zero. The large numbers of zeros can be taken advantage of when encoding the coefficients, and this encoding is where the majority of the compression takes place.

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

(a) Original 8x8 block

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

(b) Shifted 8x8 block

Figure 2.14: Typical 8x8 block of pixels

-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

(a) Results of FDCT

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

(b) Quantization matrix

Figure 2.15: FDCT results and its quantizer

When encoding the 8x8 block in the order seen in Figure 2.17(a), one could take advantage of the runs of zeros that occur, but because of the distribution of non-zero coefficients, the normal order is not the most efficient. To get longer runs of zeros, the JPEG baseline standard encodes the 8x8 block in the zig-zag order shown in Figure

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 2.16: Quantized DCT coefficients

2.17(b). In the example block of quantized coefficients, the zig-zag order would result in the following coefficients (with 0(n) denoting a run of n zeros):

```
-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0(2) -1 2 0(5) -1 -1 0(38)
```

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

(a) Normal

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

(b) Zig-zag

Figure 2.17: Normal and zig-zag orders

To reconstruct the new 8x8 block, the quantized coefficients are first unquantized resulting in the block in Figure 2.18(a). The inverse DCT (IDCT) is then applied which produces the block in Figure 2.18(b). Level shifting the values by 128 gives the reconstructed block in Figure 2.19. Note that most of the values in the reconstructed block do not match the original values in Figure 2.14(a), but the values should be close enough so that the difference would not be subjectively objectionable.

The DCT has become very popular in the field of image compression. Chips have been made to perform the DCT quickly and, in the absence of DCT specific hardware, the DCT can be computed in many ways. Computing the DCT via the DFT can be done if one has a fast DFT algorithm [5]. Computing the DCT can also be done via matrix multiplications [3]. A recursive method of the DCT also exists, as well as an

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a) Unquantized DCT coefficients

-70	-64	-61	-64	-69	-66	-58	-50
-72	-73	-61	-39	-30	-40	-54	-59
-68	-78	-58	-9	13	-12	-48	-64
-59	-77	-57	0	22	-13	-51	-60
-54	-75	-64	-23	-13	-44	-63	-56
-52	-71	-72	-54	-54	-71	-71	-54
-45	-59	-70	-68	-67	-67	-61	-50
-35	-47	-61	-66	-60	-48	-44	-44

(b) Results of IDCT

Figure 2.18: Unquantized coefficients and their IDCT

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

Figure 2.19: Reconstructed 8x8 block

implementation using several one dimensional DCTs. These and other methods of computing the DCT can be found in [19]. Depending on the available hardware or algorithms, the DCT can be quickly computed.

2.3 Evaluation of compression

When evaluating the results of image compression, there are two types of fidelity criteria that can be used. One type are called objective fidelity criteria. Objective fidelity criteria use numerical measures of how close the original image is to the reconstructed image. Two examples of objective fidelity criteria are the mean squared error (MSE) and the mean absolute error (MAE). Consider the original image to be a function f of the two coordinates x and y and the reconstructed image to be the function $\hat{f}$. Assuming the image has dimensions MxN, the MSE is defined in Equation 2.14 and the MAE in Equation 2.15.

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[f(x,y) - \hat{f}(x,y) \right]^2 \tag{2.14}$$

$$MAE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f(x,y) - \hat{f}(x,y)| \tag{2.15}$$

While these objective measures of error are easy and convenient to use, they do not accurately measure the appearance of the reconstructed image to the human eye. This is especially true for objective fidelity criteria biased on uniform quality since it has already been established that the human eye does not have uniform resolution over its field of view. Minimizing error measures such as the MSE and MAE are good for images that are needed for computer processing. However, many images will be viewed by humans so a different fidelity criterion is needed to assess how successful a compression method is. The second type of fidelity criteria are called subjective fidelity criteria. Subjective fidelity criteria are not as attractive as objective fidelity criteria, since they usually involve displaying the reconstructed image to test subjects who then provide a subjective evaluation of the image quality compared to the original image. An example of a subjective fidelity criterion is the subjective impairment scale shown in Figure 2.20. This point scale represents the degree of impairment of the original image the reconstructed image appears to have. The higher the points, the smaller perceptible difference there is between the original and the reconstructed image. After showing the images to the test subjects, an average is taken of the total points over the test subjects to obtain a mean opinion score (MOS). Other scales can be used to obtain a MOS as long as the same scale is used when comparing the MOS for different compression methods.

Opinion	Points
Not perceptible	7
Barely perceptible	6
Definitely perceptible but only slight impairment of the image	5
Impairment to the image but not objectionable	4
Somewhat objectionable	3
Definitely objectionable	2
Extremely objectionable	1

Figure 2.20: Subjective impairment scale

There are different ways to approach stereo image compression. One approach is to try to minimize some quantitative error measure, such as the MSE and MAE, between a reconstructed image and an original image. This approach places more importance on objective fidelity criteria and is used in [18, 21]. Another approach is to take advantage of the human visual system when compressing images, so that image quality is lost in a way that would be least objectionable to the human eye when the image is reconstructed. This approach places more importance on subjective fidelity criteria and is used in [7, 8, 12, 18]. In this thesis, the approach taken will be the one that takes advantage of the human visual system.

Chapter 3

Compression Schemes

Three stereopair compression methods will be discussed in this chapter. The first is a straightforward method of using the DCT to do stereo compression. The next two methods add spatially varying sensing in two different ways to improve the compression.

3.1 Uniform DCT

The uniform DCT is an intuitive approach to stereo compression using the DCT. It has been shown that an optimal way to compress stereopairs is to compress one side first, and then compress the other side using redundant information from the first side [18]. Using the left side or right side of the stereopair as the first side should not matter. The problem is in defining what is meant by redundant information. Since the DCT is done in blocks, it is logical to define redundant information as blocks that match. For the uniform DCT, an arbitrary decision was made to encode the left image first and encode the right image using matches to the left image. Figure 3.1 gives an overview of the compression part of the uniform DCT. The left and right images are first partitioned into 8x8 blocks. If the image does not have dimensions that are multiples of 8, the blocks are padded on the edges so that they are 8x8 blocks. Next, we shift the pixel values by -128 as in the JPEG baseline compression standard. The forward DCT is applied to all the blocks to obtain 8x8 blocks of DCT coefficients. These blocks of coefficients are then quantized using a scaled version of the quantization matrix shown in Figure 2.15(b). After the quantization of all the blocks, the left blocks are encoded, as is, by using the default Huffman code tables

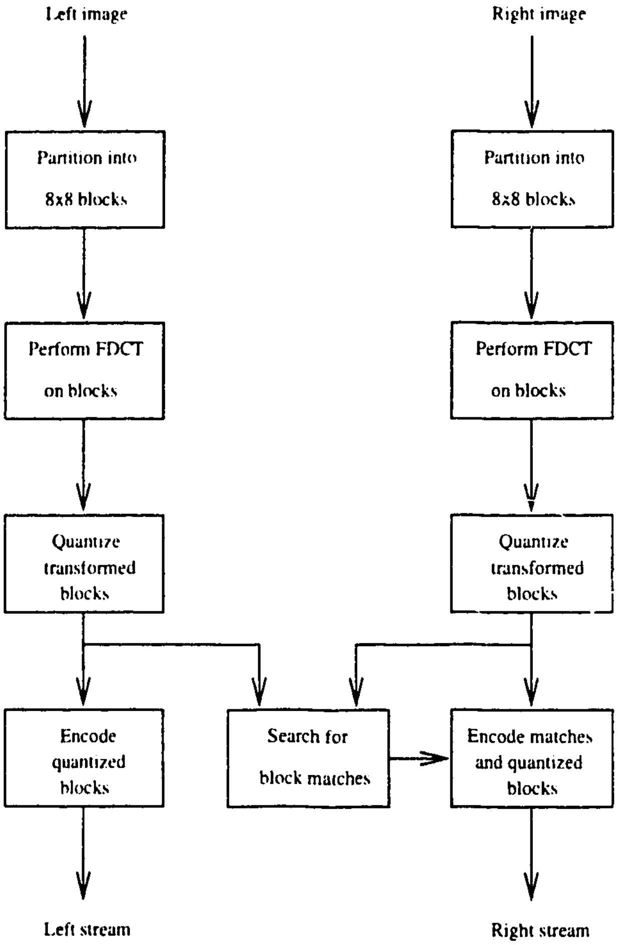

Left image
Right image
Partition into 8x8 blocks
Partition into 8x8 blocks
Perform FDCT on blocks
Perform FDCT on blocks
Quantize transformed blocks
Quantize transformed blocks
Encode quantized blocks
Search for block matches
Encode matches and quantized blocks
Left stream
Right stream

Figure 3.1: Uniform DCT compression scheme

for the JPEG baseline compression standard. For the right image, there is a category added to the Huffman tables to accommodate block matching. The right image blocks are encoded as matches to the left image blocks if they exist, otherwise they are encoded as they are, using the new Huffman code tables. To reduce the number of bits required to encode a match, the search area for the block matches must be restricted. Using trial stereopairs and several scaling factors for the quantization matrix, most matches were found to appear at either the same location in the left image or within a few blocks of the same location. The majority of the matching blocks came from within three horizontal blocks and appear either at the same row or one row above or below the block location in the left image. The search area is shown in Figure 3.2. In deciding what constitutes a match, the MAE is used as a quantitative measure of how large the difference between two blocks is. The difference is then compared to a threshold value and the blocks are said to match if the difference falls below the threshold.

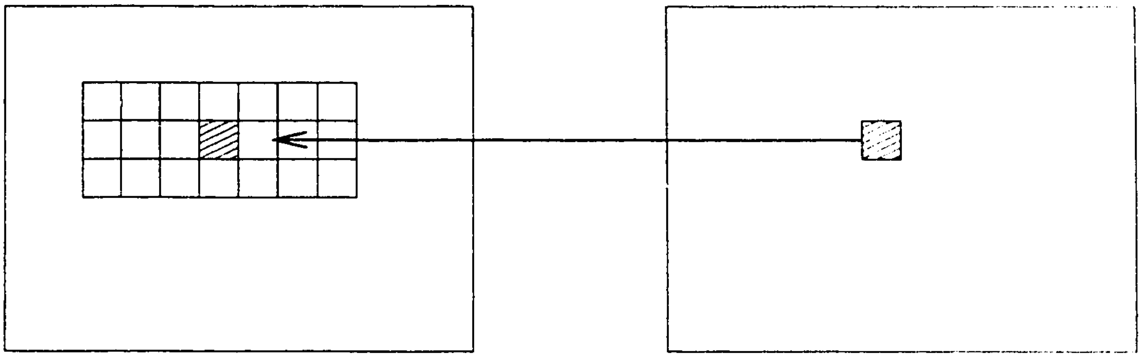


Figure 3.2: Search area in left image for right block

The left and right streams resulting from the encoding can be stored separately or combined into one stream. The uniform DCT decompression scheme is outlined in Figure 3.3. The left hand side (LHS) stream is first decoded into its 8x8 blocks. Then the right hand side (RHS) stream is decoded into its 8x8 blocks by either decoding an encoded block (as in the LHS stream) or by copying a block from the LHS blocks that matched with the RHS block when the image was compressed. Next, all the blocks are unquantized using the same scaling factor of the quantization matrix in Figure 2.15(b) that was used for compression. The inverse DCT is then performed on all the blocks, and the resulting pixel values are shifted by 128 to obtain the reconstructed image.

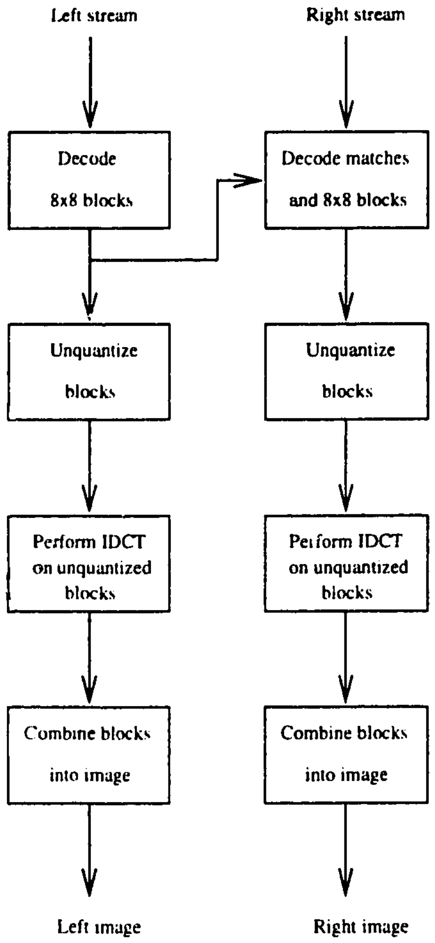

Left stream
Right stream
Decode 8x8 blocks
Decode matches and 8x8 blocks
Unquantize blocks
Unquantize blocks
Perform IDCT on unquantized blocks
Perform IDCT on unquantized blocks
Combine blocks into image
Combine blocks into image
Left image
Right image

Figure 3.3: Uniform DCT decompression scheme

There are two formal parameters given to the uniform DCT to control the image quality. One is the scaling factor used to scale the quantization matrix for the DCT coefficients found in Figure 2.15(b). Values for the scaling factor may be any number greater than or equal to zero. A value of 0.25 usually gives images that are almost indistinguishable from the original. A value of zero for the scaling factor is a special case where there will be no quantization of coefficients. A value of one usually gives subjectively acceptable reconstructed images, and values larger than three usually give reconstructed images containing noticeable quality loss. The other parameter given to the uniform DCT is the threshold value for block matching. The threshold is the largest value for the MAE that is acceptable for blocks to match and may be any integer greater than or equal to zero. A value of zero is generally used. but the and the amount of compression desired. The two parameters may be adjusted to achieve the desired compression ratio but, due to the nature of compression based on the DCT, an exact compression ratio cannot be specified. When high compression ratios are used, the quality loss occurs uniformly and when the reconstructed images are viewed on a HMD, the details of the object of interest in the image may appear blurred.

3.2 Spatially Varying DCT

One of the methods of combining spatially varying sensing and the DCT is to use the concept of spatially varying sensing while performing the quantization of the DCT coefficients. This combination will be referred to as the spatially varying DCT (SVDCT). The overviews of the SVDCT for compression and decompression look the same as for the uniform DCT in Figure 3.1 and Figure 3.3. The difference between the uniform DCT and the SVDCT is in the step where the DCT coefficients are quantized. The uniform DCT uses the same quantization matrix to uniformly quantize both images, but the SVDCT uses a different quantization matrix depending on the location of the block being quantized. The quantization matrix is modified so that the fovea in the stereopair retains its good quality while the periphery can afford to lose some quality. The stereopairs with vergence are convenient for use with spatially varying sensing because in both the left and right images, the fovea is located in the center.

Given a base quantization matrix, there must be some maximum limit on the quality loss moving from the fovea to the periphery. This limit will be in the form of a maximum scaling factor for the quantization matrix. Values for this limit can be any number greater than 1, but should not be very high if the quality of the image is to remain acceptable. The maximum limit on the scaling factor will be referred to as Q. At the fovea, the base quantization matrix is scaled by 1, so the matrix stays the same at the center of the image. At the extreme periphery, the base quantization matrix is scaled by the maximum scaling factor. As the blocks from the fovea to the extreme periphery are quantized, the quantization matrix should be scaled so that

VRT loses information exponentially moving from the fovea to the periphery, so it is logical to make the scaling factor approach Q exponentially. The scaling factor will be referred to as S and is dependent on the distance the block is from the center of the image, which will be denoted by x. The x value is normalized against the maximum distance a block can possibly be from the fovea. Given that the fovea is the center of a MxN image, the maximum distance is $\sqrt{\left(\frac{M}{2}\right)^2 + \left(\frac{N}{2}\right)^2}$. Since the VRT has expected savings and the α value as parameters to control the compression, a similar set of parameters should also be used for S to control the SVDCT compression. Along with the maximum scaling factor Q, two more parameters which will be referred to as n and C will be used to control the exponential loss of quality. The equation for S is shown in Equation 3.1 along with the restrictions on the various parameters.

$$S(x) = \exp\left((Cx)^n \ln(Q)\right) \tag{3.1}$$

$$C \geq 1$$

$$0 \leq x \leq 1$$

$$n \geq 1$$

$$Q \geq 1$$

The n parameter is used in a similar way to α in the VRT. A value of one gives a normal exponential curve and, as the value of n gets higher, an increasing area around the fovea will retain good quality. This matches the behaviour from higher values of α giving a more defined fovea in compression using the VRT. The second parameter C is called the curve compression factor. C is used to make the exponential loss of quality reach the maximum limit before the extreme periphery is reached. The higher the value of C, the faster the scaling factor S reaches Q. A value of one for C results in the scaling factor reaching Q exactly at the extreme periphery, but if a larger value is used then S reaches Q before the extreme periphery is reached. Since Q is our maximum scaling factor, in the cases where the calculation of S results in a larger value than Q, the value of Q will be used instead of the calculated S value. Thus a ceiling value is placed on S for the cases where the curve compression factor is greater than one. Examples of the effects of modifying n and C are shown in Figures exponential curve for the scaling factor as we move from the fovea ($x = 0$) to the extreme periphery ($x = 1$). Both n and C are 1 for this curve. In Figure 3.5, the value of n is 4, and this produces a low scaling factor for a longer distance from the fovea compared to the normal graph. In Figure 3.6, the value of n is set back to 1 and the value of C is set to 2. The resulting scaling factor reaches the maximum of five half way to the extreme periphery and remains at five until the extreme periphery is reached. Comparing Figure 3.4 to Figure 3.6, it is evident why the C parameter is called the curve compression factor. Figure 3.7 shows an example of the resulting scaling factor combining the use of n and C with $n = 3$ and $C = 1.5$.

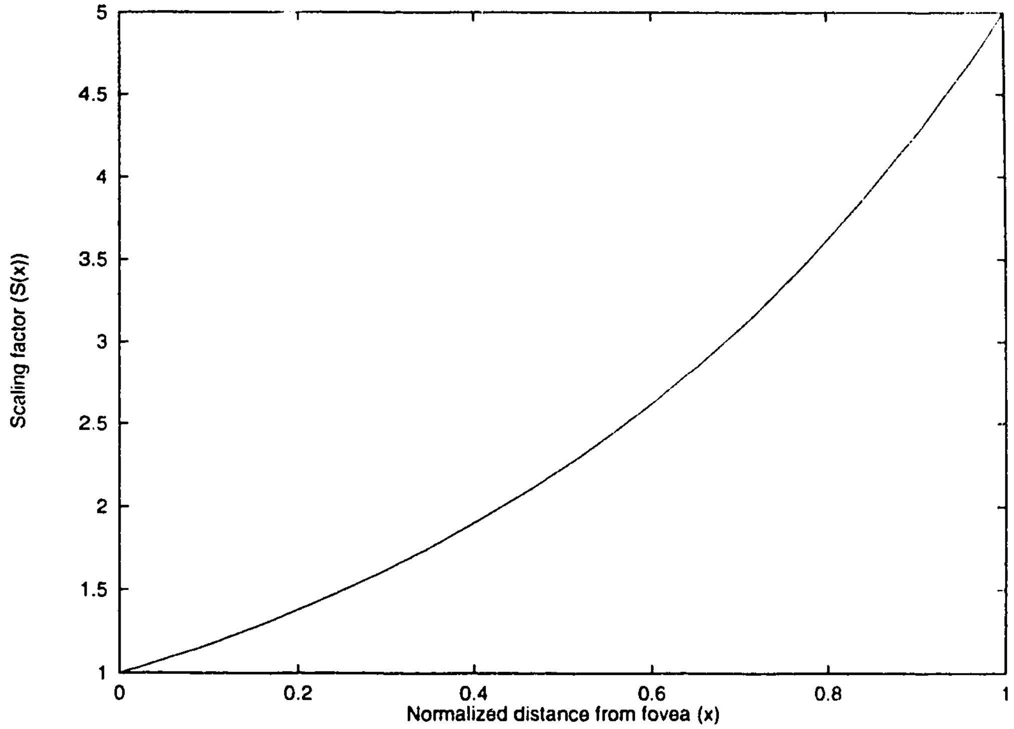


Figure 3.4: $S(x) = \exp(x \ln(5))$

In total, there are five formal parameters that must be given to the SVDCT compression method. The first parameter is the initial scaling factor of the quantization matrix in Figure 2.15(b). This first parameter gives us the base quantization matrix, so useful values may be anywhere from 0.25 to 1. Like the uniform DCT, the value of 0 for the scaling factor is the special case where no quantization of DCT coefficients is performed. The second parameter is the maximum scaling factor Q for

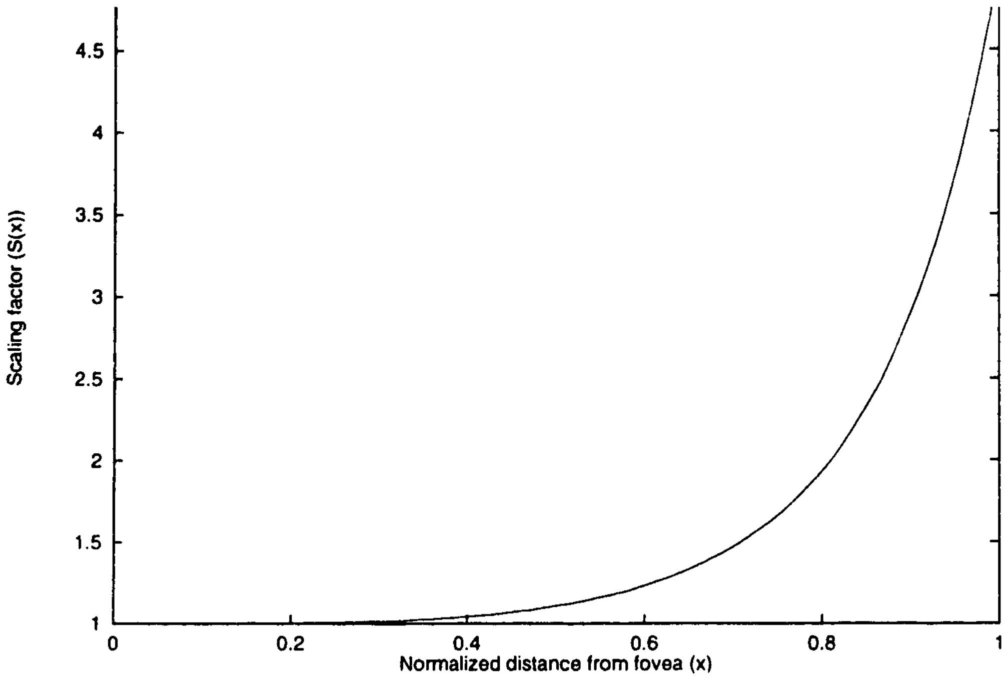


Figure 3.5: $S(x) = \exp(x^4 \ln(5))$

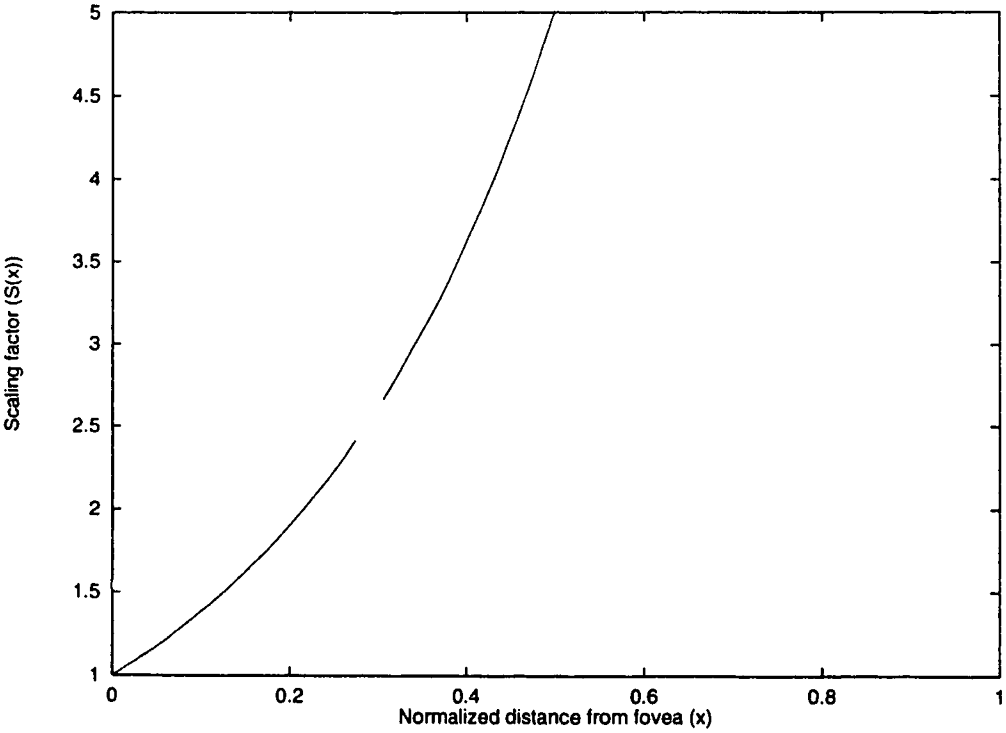


Figure 3.6: $S(x) = \exp((2x) \ln(5))$

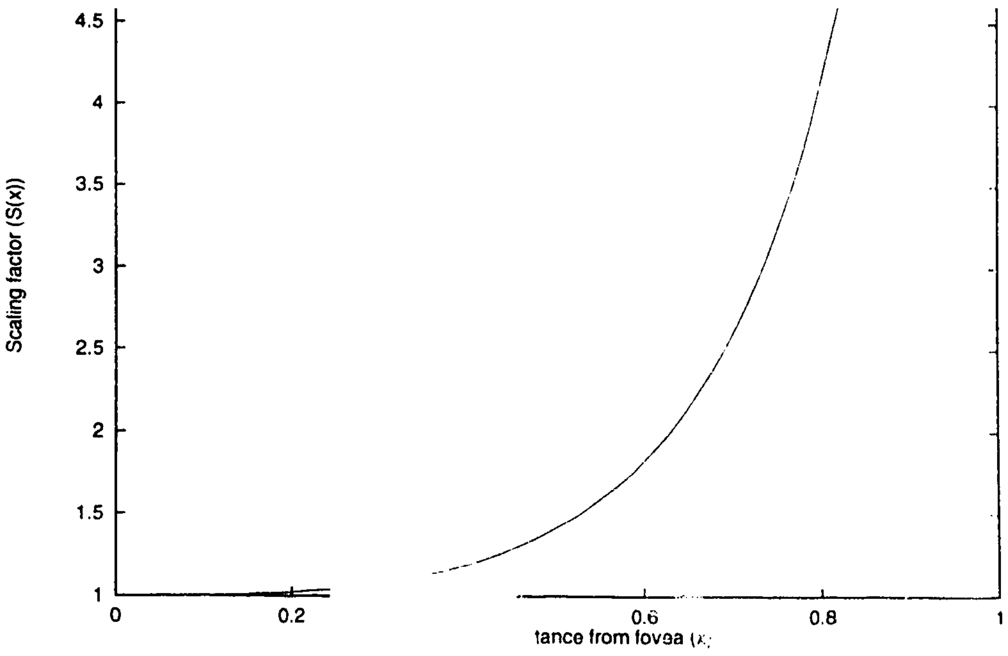


Figure 3.[illegible] [illegible] $\exp((1.5x)^3 \ln(5))$

the quantization matrix. The third and fourth parameters are the n and C values which were just described. Useful values for n range from 1 to 10, and those for C from 1 to 2. The final parameter is the threshold value for block matching. It is identical in function to the matching threshold used in the uniform DCT in deciding what constitutes a block match. The main objective of the SVDCT is to reduce the storage space necessary for the blocks in the periphery so that more space can be dedicated to keeping the quality high at and around the fovea. The storage savings occur because the quantization produces longer runs of zeros, which usually take less space to encode than a string of non-zero coefficients Additional storage savings may occur because the larger number of coefficients quantized to zero in the periphery can make block matching more successful. The extra storage space can then be used to enhance the quality of the blocks around the fovea, by either allowing the base quantization matrix to contain smaller values or by modifying the compression curve to produce a more defined fovea area. Stereopairs compressed with the SVDCT have better quality in the center of both images so the object of interest is clearer than the area surrounding the object of interest. This clarity is preferred when using a HMD.

so although the SVDCT requires a small amount of extra processing than does the uniform DCT, it produces stereopairs more suited for use with a HMD.

3.3 Spatially Varying Sensing Followed by DCT

Another way of combining spatially varying sensing with the DCT is to first perform the CVRT on the stereopair images and then use the uniform DCT to compress the resulting smaller images. This spatially varying sensing and the DCT will be referred to as the SVSDCT. Figure 3.8 shows an overview of the SVSDCT compression method. CVRT compression is used to preprocess the images given to the uniform DC . The uniform DCT is used instead of the SVDCT because the CVRT has already taken into account that the area around the fovea should take precedence over the periphery. There is no need to further enhance the area around the periphery in the preprocessed image with the SVDCT. Decompression using the SVSDCT is outlined in Figure 3.9. The left and right streams are first decompressed using the uniform

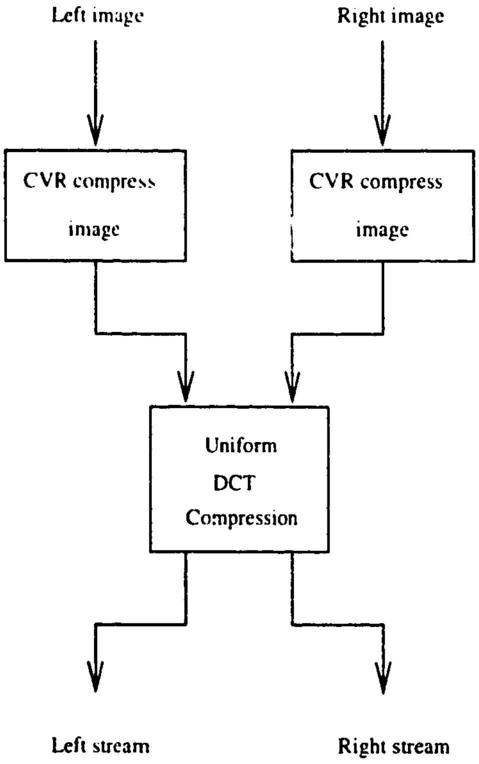


Figure 3.8: SVSDCT compression scheme

DCT, and the results are given to the CVRT decompression algorithm to reconstruct the left and right images.

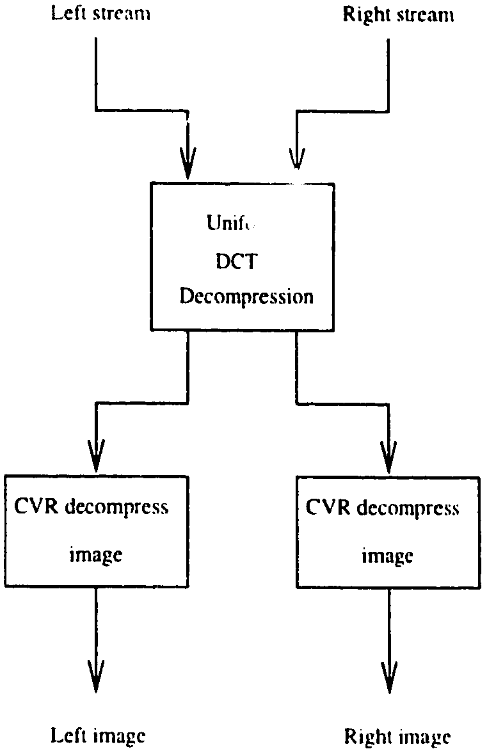


Figure 3.9: SVSDCT decompression scheme

The SVSDCT combines a spatial domain compression method (the CVRT) and a frequency domain compression method (the uniform DCT). The major advantage that the SVSDCT method has over the uniform DCT and the SVDCT is speed CVRT compression has linear complexity and produces a smaller image for the uniform DCT to compress. Assuming that there is no DCT hardware present, the smaller images significantly reduce the compression times of the uniform DCT. For decompression, the processing time for the uniform DCT portion is about the same as for compression, but the CVRT can use different types of interpolation to generate the unknown pixel values. If nearest neighbour interpolation is used then CVRT decompression is linear, making SVSDCT compression and decompression use relatively the same amount of processing time.

There are four formal parameters that must be supplied in order to use the SVSDCT compression method. The first two parameters are the expected savings and the

α value that are normally used in CVRT compression. The other two parameters are the scaling factor and the matching threshold used for the uniform DCT compression portion of the SVSDCT. The practical value for the scaling factor in this case, however, will not be the same as for the uniform DCT. Since there is already information being lost using CVRT compression, it is necessary to choose a scaling factor that would keep the images at a high quality. Practical values for the scaling factor for use with the SVSDCT would usually lie between 0.25 and 1. The same reasoning applies to the matching threshold. Remember that in the CVRT decompression, one pixel may contribute to many pixels around it. The matching threshold should be set very low so that block matches that are not exact will not become too obvious once the CVRT decompression is done. Stereopairs using the SVSDCT smooth out the blocky appearance of the uniform DCT and the SVDCT at high compression ratios and may be more appealing to the human eye.

Chapter 4

Color Support

Greyscale images are helpful to illustrate the quality loss when compression is high, but most images are in color. In this chapter we look at how to add color support to the greyscale compression methods. Color models (also known as color spaces) will first be discussed, and then the choice of color model and how we use it to compress color stereopairs will be presented.

4.1 Color Models

In order to display an image from a computer, the image is usually stored as triplets in the *RGB* color model. Each pixel in the image has a triplet of numbers corresponding to the amount of red, green, and blue present in the pixel. This color model is convenient for display devices, but it is not intuitive to think of colors in the *RGB* color model, nor is it ideal for compression purposes since each number in the triplets have equal importance to represent the pixel on the screen.

A more intuitive way to think of colors is in the *HSB* color model. The three letters stand for hue, saturation, and brightness. The *HSB* color model is also known as *HSV*. Like the *RGB* color model, the *HSB* color model has triplets of numbers representing the pixels in the image. The three numbers represent the hue, saturation, and intensity of a pixel. The *HSB* color model is a good one for artists since hue, saturation, and brightness are more closely related to a painter's palette than is the *RGB* color model. Operations such as darkening, brightening, and tinting are more easily described and, while the *HSB* color model is more intuitive, each value in a *HSB* triplet is equally important in representing a pixel's color.

A color model used by many color television broadcasts is YUV. It also goes by the name YIQ. The Y stands for luminance and the UV both stand for chrominance. As with the previous two color models, pixels in the YUV color model are triplets. The triplets represent a luminance value and two chrominance values for each pixel in the image. The luminance component of an image represents what the image would look like if it were in greyscale. The chrominance components add the color information to the image. The human eye detects differences in luminance, however, much more than differences in chrominance. Thus the Y value has more importance than either of the two chrominance values, making the YUV color model a good candidate for use with compression methods.

4.2 RGB versus YC_bC_r

If an image is left in the RGB color model, or converted to the HSB color model for compression, each component of the triplets would have to be weighted equally for the appropriate effect that the component has on the pixel value. Converting to the YUV color model would give the advantage of being able to reduce the quality of the chrominance components without noticeable loss in color quality to the human eye. However, there is another color model closely related to the YUV model called YC_bC_r, which is used in JPEG. The formulas for converting from the RGB to the YC_bC_r color model are given in Equations 4.1 to 4.3, and the formulas for converting back from the YC_bC_r to the RGB color model are given in Equations 4.4 to 4.6. All these equations assume that the RGB components have been normalized so that the lowest possible RGB value is 0 and the largest possible RGB value is 1. R, G, and B will stand for the red, green, and blue values, respectively, while Y, C_b, and C_r will stand for the luminance and the two chrominance values, respectively.

$$Y = 0.2989R + 0.5866G + 0.1145B \tag{4.1}$$

$$C_b = -0.1687R - 0.3312G + 0.5000B \tag{4.2}$$

$$C_r = 0.5000R - 0.4183G - 0.0816B \tag{4.3}$$

$$R = Y + 0.0000C_b + 1.4022C_r \tag{4.4}$$

$$G = Y - 0.3456C_b - 0.7145C_r \qquad (4.5)$$

$$B = Y + 1.7710C_b - 0.0000C_r \qquad (4.6)$$

The luminance component of the YUV and the YC_bC_r color models range from 0 to 1. The main advantage of the YC_bC_r color model over the YUV color model is that the chrominance values have a more convenient range of values. The chrominance values for the YC_bC_r range from -0.5 to 0.5 which, when denormalized, would be the preferred range of values over the denormalized YUV chrominance values when performing the DCT.

4.3 Modifications for Color Support

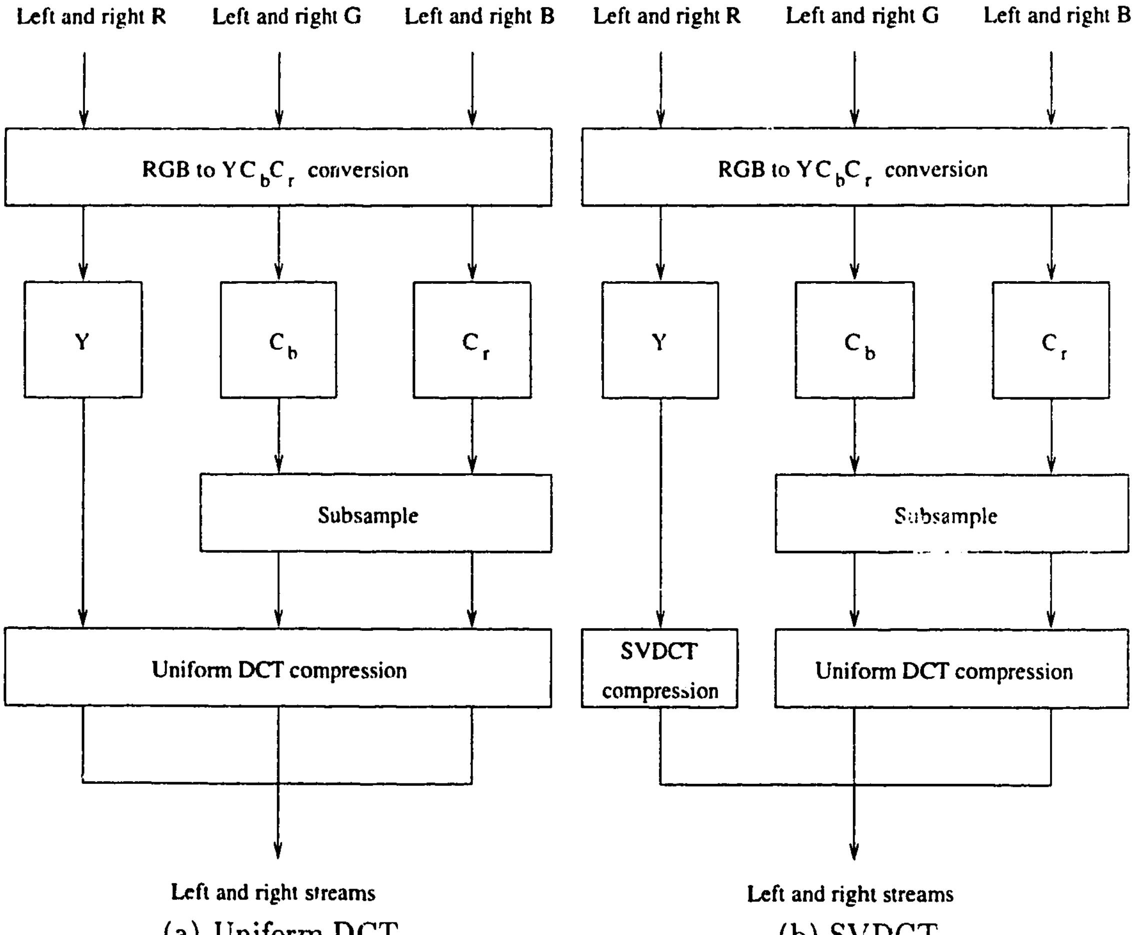
There are two modifications to be made when adding color support to all three greyscale compression methods. Both modifications are meant to take advantage of the fact that the chrominance values of the images can lose quality more than the luminance values without a major drop in perceivable color quality. One of the modifications is to quantize the chrominance images more coarsely than the luminance image. The quantization matrix used to quantize both the C_b and C_r components is shown in Figure 4.1, and the luminance or Y component is quantized as if it were a greyscale image using the matrix from Figure 2.15(b). Like the greyscale quantization matrix, the color quantization matrix also comes from the JPEG baseline compression standard. Comparing the two quantization matrices, it is clear that the chrominance values are quantized much more heavily than the luminance values. The second modification made to the chrominance portions of the image is to subsample the chrominance images by a factor of two. This modification also comes from the

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Figure 4.1: Quantization matrix for C_bC_r

JPEG baseline compression standard. The subsampling reduce. the space taken by each of the chrominance images to 25% of the original image, with subjectively little change in color quality.

Since the chrominance components are already quantized so heavily, the SVDCT will modify only the luminance quantization matrix during the compression. The color quantization matrix will remain constant for the SVDCT just as with the uniform DCT. For the SVSDCT, the luminance component will be treated as a greyscale image compressed with the SVSDCT. For the chrominance components, the VRT is performed first, and the resulting images are then subsampled by a factor of two before being compressed using the uniform DCT as if they were greyscale images. Figure 4.2 illustrates the compression process for both the uniform DCT and the SVDCT, and Figure 4.3 illustrates the compression process for the SVSDCT.



(a) Uniform DCT

(b) SVDCT

Figure 4.2: Color compression with the uniform DCT and SVDCT

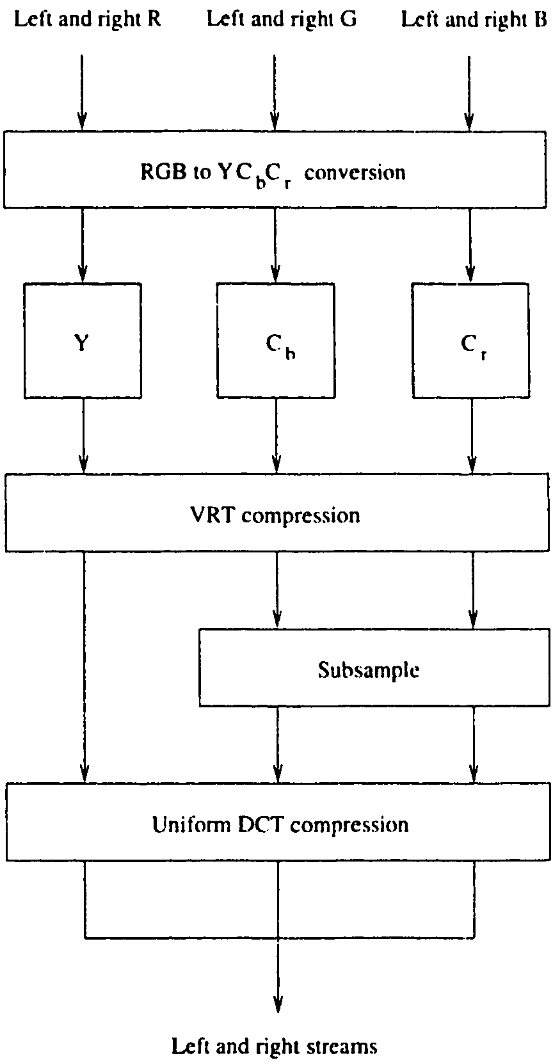
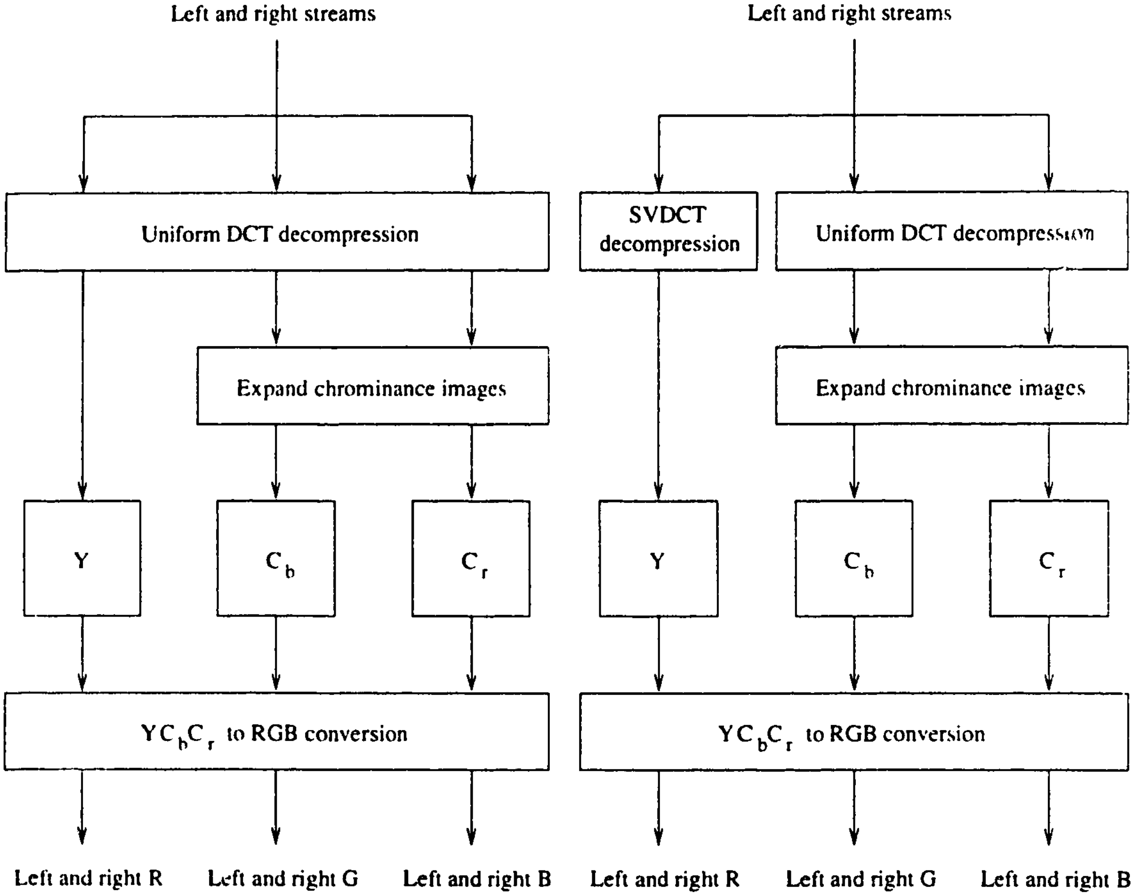


Figure 4.3: Color compression with the SVSDCT

Color decompression of stereopairs via the uniform DCT and SVDCT is shown in Figure 4.4, and decompression via the SVSDCT is shown in Figure 4.5. All three decompression methods are basically the reverse of their respective compression method. The portion of the Figures that may need clarification is the expanding of the chrominance images. When expanding the subsampled chrominance components, there is no fixed method of interpolation that is required. It is up to the programmer to decide which interpolation method to use, depending on the quality preferred. For the purposes of this thesis, nearest neighbour interpolation was used.



(a) Uniform DCT (b) SVDCT

Figure 4.4: Color decompression with the uniform DCT and SVDCT

The only extra formal parameter that is added for color support is the scaling factor of the color quantization matrix. The usual value of the color scaling factor is 1. If very high color quality is needed then the scaling factor may be reduced to around 0.25. Values larger than 1 may end up quantizing the chrominance components too

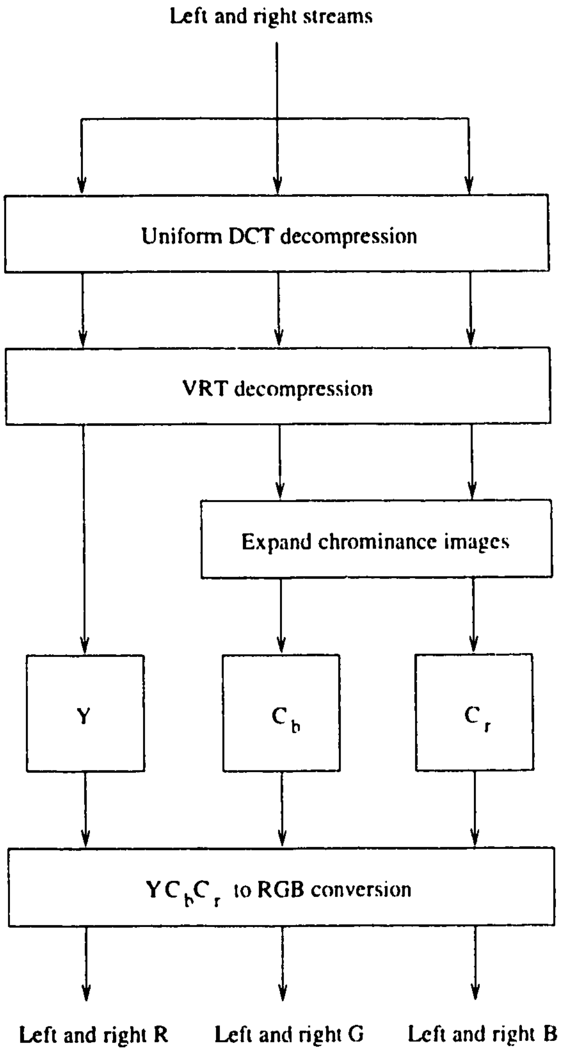


Figure 4.5: Color decompression with the SVSDCT

much and thus produce a stereopair that appears to be greyscale because much of the color information would be eliminated.

Chapter 5

Analysis of Compression

The uniform DCT, SVDCT, and SVSDCT will be analyzed according to their performance under low, medium, and high compression ratios. Results of the three compression methods will be shown, followed by results of subjective evaluation surveys for the images produced by the three compression methods.

5.1 Compression results for greyscale images

The original stereopair used for the greyscale image compression methods is shown in Figure 5.1. The left and right images have dimensions of 320x240 pixels each and are taken with a vergence angle of 5°. Each pixel is a value between 0 and 255, inclusive, with 0 being the darkest and 255 being the brightest. The left and right images are stored in PGM format, which is a simple format containing basically the width and height of the image followed by the pixel values

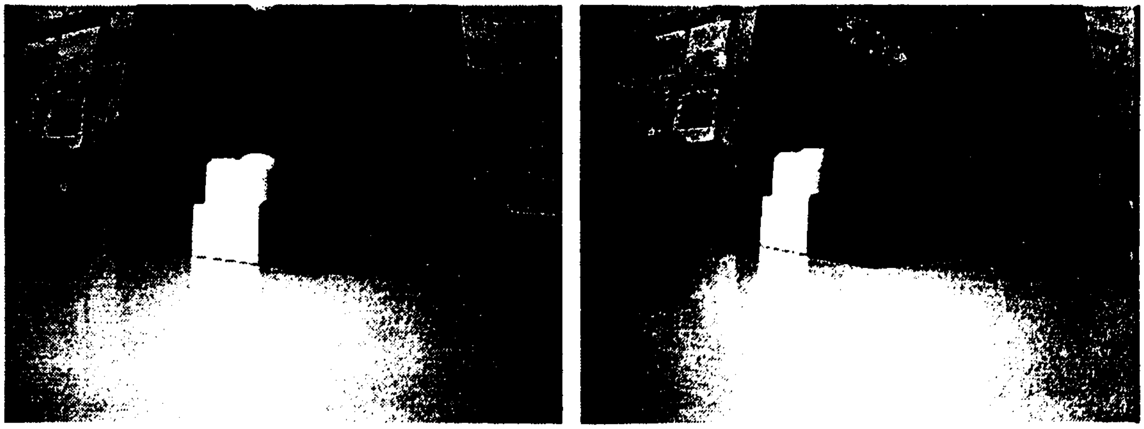


Figure 5.1: Original greyscale stereopair

5.1.1 Low compression ratios

For the low compression ratios, the reconstructed stereopairs should be practically indistinguishable from the original. For the highest image quality, the uniform DCT was applied with no quantization of the coefficients, so that the only reconstruction error was from the lack of precision of the floating point number operations. The matching threshold was set to zero so that every block match that would occur would be an exact match. Both original images have dimensions 320x240, which means each image contains 76800 pixels for a total of 153600 pixels per stereopair. Since the values of the pixels are between 0 and 255, one byte (8 bits) is required to represent one pixel and therefore 8 bits/pixel are needed to represent the stereopair (not including header information). When the uniform DCT is used to compress the stereopair, the amount of space taken by the two images totals to 86644 bytes (including header information), which averages out to approximately 4.51 bits/pixel. The compression ratio works out to be approximately 1.77:1. This result is a good illustration of the information packing ability of the DCT. The results of the compression are the same for all three compression methods, since the SVDCT and SVSDCT can both be made to behave like the uniform DCT. Because there was hardly any loss in quality, there were no block matches achieved for this particular stereopair. Figure 5.2 shows the reconstructed stereopair for comparison with the original in Figure 5.1.

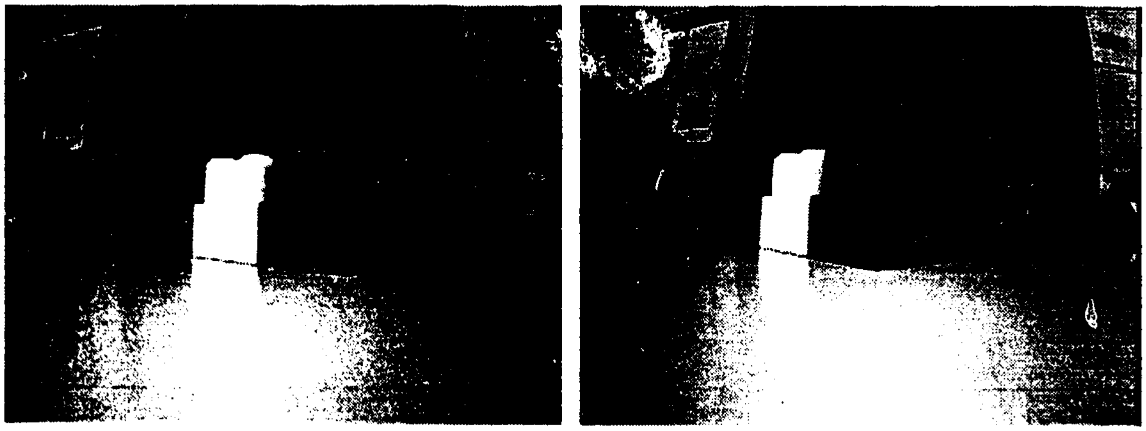


Figure 5.2: Low compression stereopair

5.1.2 Medium compression ratios

For the medium compression ratios, the reconstructed stereopairs should keep acceptable image quality, but not necessarily be indistinguishable from the original. The main goal of the medium compression ratios is to find a good compromise between good reconstructed image quality and small compressed file sizes.

The uniform DCT was considered first, and a value of 0.8 was used as the scaling factor, with zero as the matching threshold so that block matches would be exact. The resulting file size total of the left and right images was 9288 bytes, which averages to approximately 0.48 bits/pixel. The compression ratio is approximately 16.54:1, and the number of block matches found were 136. Figure 5.3 shows the reconstructed stereopair of the uniform DCT using the medium compression parameters.

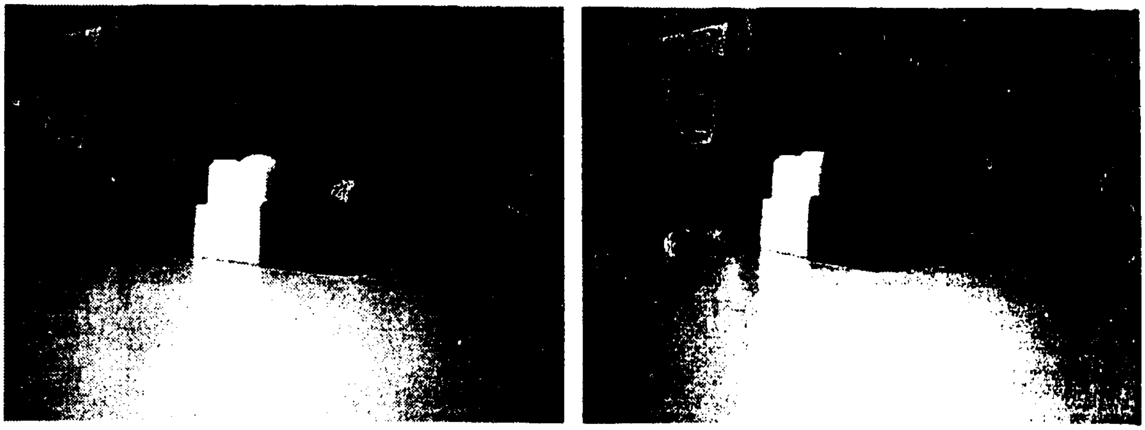


Figure 5.3: Medium compression stereopair from uniform DCT

The SVDCT was the second compression method considered. Parameters for the SVDCT were chosen to produce a compressed file size total which was very close to the resulting total in the uniform DCT. After some experimentation with different values, the parameters chosen were 0.5 for the base quantization matrix scaling factor, 3 for the value of n, 9 for the maximum quantization value, 1 for the CCF, and 0 for the matching threshold. The resulting file size total was 9280 bytes, averaging to approximately 0.48 bits/pixel as in the uniform DCT. The compression ratio for the SVDCT compression is approximately 16.55:1, and the number of block matches found were 95. The compression curve for the scaling factor of the base quantization matrix is shown in Figure 5.4, and the reconstructed stereopair using the SVDCT is shown in Figure 5.5.

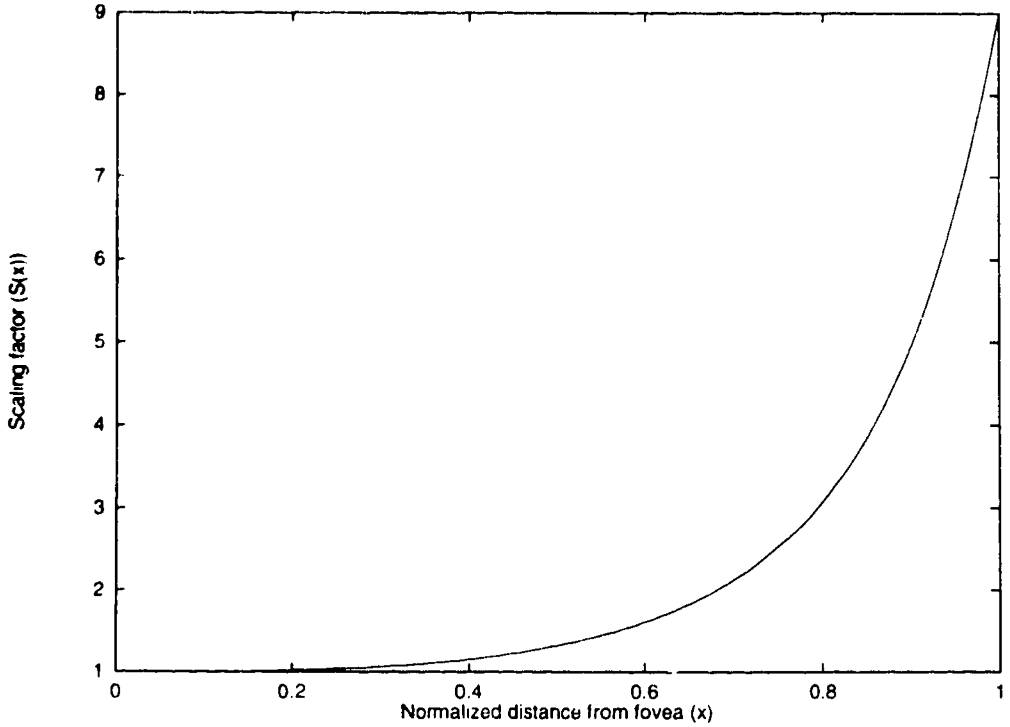


Figure 5.4: $S(x) = \exp(x^3 \ln(9))$

The SVSDCT was the final compression method considered. Like the SVDCT, the parameters chosen for the SVSDCT were selected so that the resulting file size total would be almost the same as for the uniform DCT. After experimenting with different values, the chosen parameters were 0.45 for the scaling factor of the quantization matrix, 75% for the expected compression, and 0.81 as the value of a. These parameter values resulted in a total file size of 9224 bytes, which works out to approximately 0.48 bits/pixel - the same as the uniform DCT and the SVDCT. The compression ratio is approximately 16.65:1, and the number of block matches achieved was 13. Figure 5.6 shows the reconstructed stereopair for the SVSDCT.

5.1.3 High compression ratios

For the high compression ratios, there is noticeable loss in image quality. The main goal here is to find the method which produces a reconstructed stereopair that is more appealing than the others.

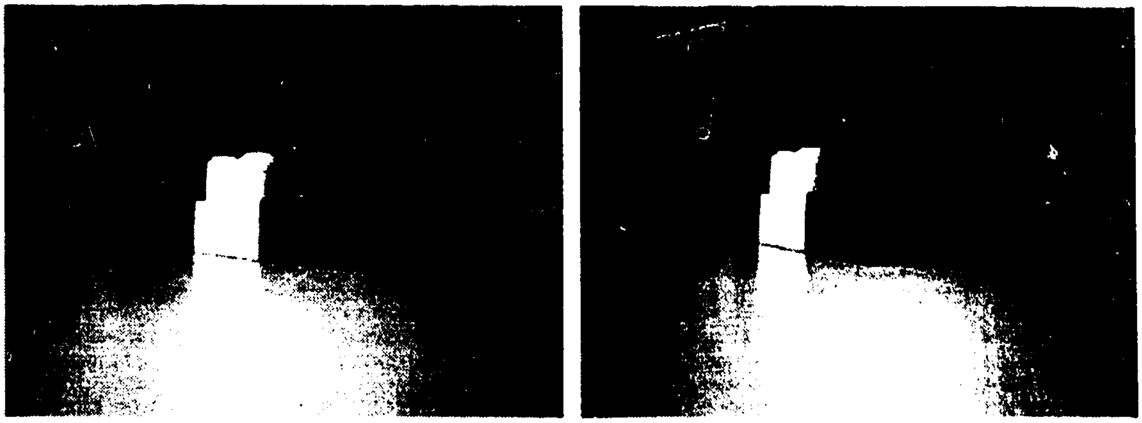


Figure 5.5: Medium compression stereopair from SVDCT

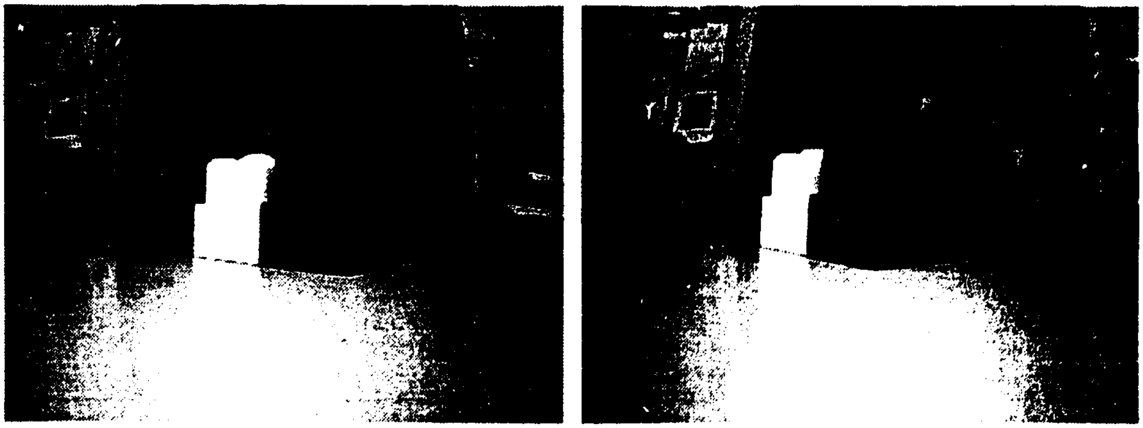


Figure 5.6: Medium compression stereopair from SVSDCT

As in the previous two sections, the uniform DCT was the first method considered. In order to illustrate how the uniform DCT loses quality, a value of 2.5 was used for the quantization matrix scaling factor, and a value of 0 was chosen for the matching threshold. These values produced a total file size of 4813 bytes, which averages to approximately 0.25 bits/pixel. The compression ratio is approximately 31.91:1, and the number of block matches found was 548. The reconstructed stereopair using the uniform DCT is shown in Figure 5.7. The whole image, including the object of interest, seems uniformly blocky in appearance.

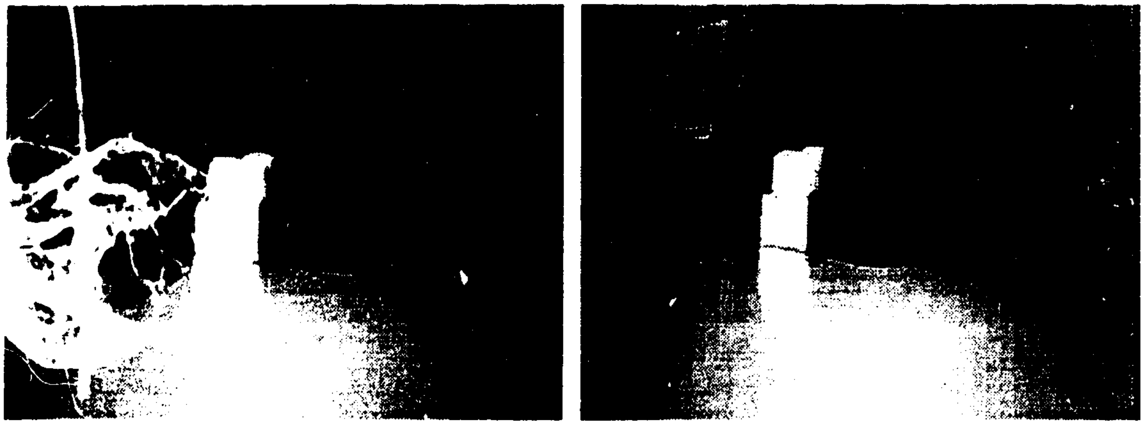


Figure 5.7: High compression stereopair from uniform DCT

Consistent with the previous section, the SVDCT was the next compression method to be considered. The parameter values were chosen to give a total file size very close to that of the uniform DCT. After experimenting with different values, the parameters chosen were 1 for the base quantization matrix scaling factor, 3 for the value of n, 6 for the maximum quantization value, 1.5 for the CCF, and 0 for the matching threshold. These parameters resulted in a total file size of 4875 bytes, which averages to approximately 0.25 bits/pixel, as in the uniform DCT. The compression ratio was approximately 31.51:1, and the number of block matches found was 530. The compression curve for the scaling factor of the base quantization matrix is shown in Figure 5.8, and the reconstructed stereopair using the SVDCT is shown in Figure 5.9. The object of interest is now clearer than the rest of the image which begins to get more blocky at the edges of the images.

The final compression method looked at was the SVSDCT. As before, the values

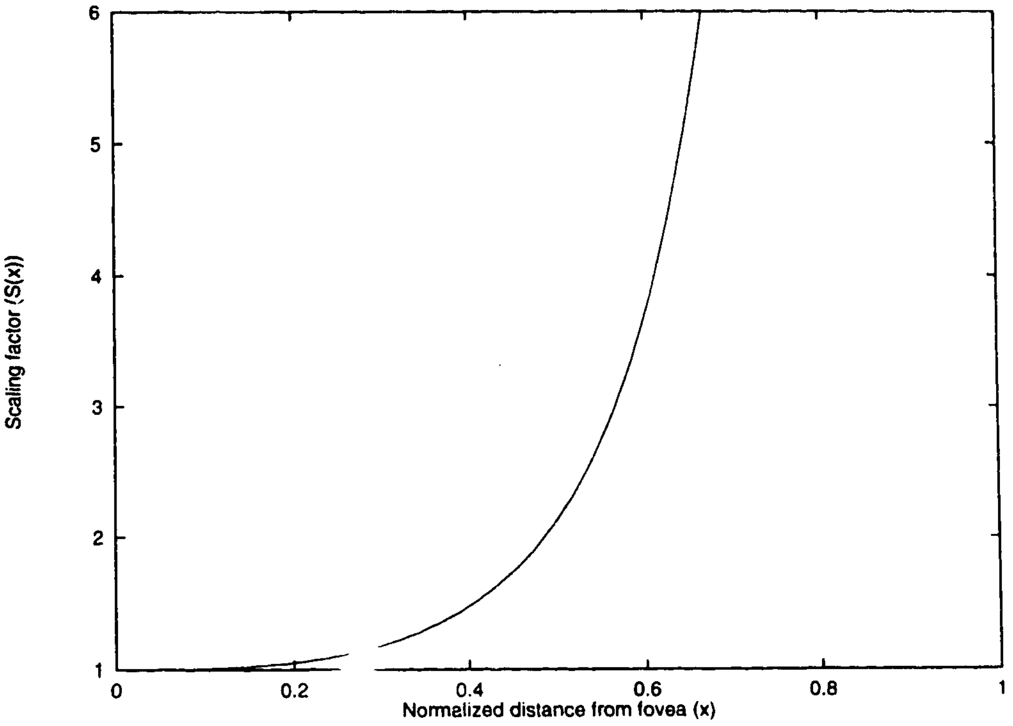


Figure 5.8: $S(x) = \exp((1.5x)^3 \ln(6))$

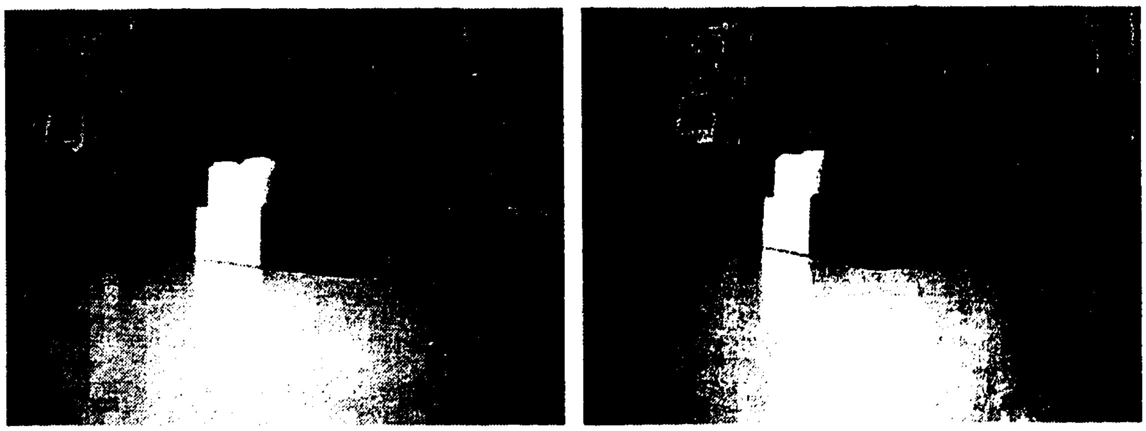


Figure 5.9: High compression stereopair from SVDCT

of the paramcters chosen for the SVSDCT were selected so that the resulting total file size would be very close to that of the uniform DCT and the SVDCT. After experimenting with different values, the parameters chosen were 0.5 for the scaling factor of the quantization matrix, 89% for the expected compression, 0.2 for the value of α, and 0 for the matching threshold. These values resulted in a total file size of 4796 bytes, which averages to approximately 0.25 bits/pixel, as in the uniform DCT and SVDCT. The compression ratio is approximately 32.03:1, and the number of block matches found was 11. The reconstructed stereopair using the SVSDCT is shown in Figure 5.10. The blocky appearance present in both the uniform DCT and the SVDCT appears to be smoothed out, with the object of interest at the center having better quality than the other areas of the image.

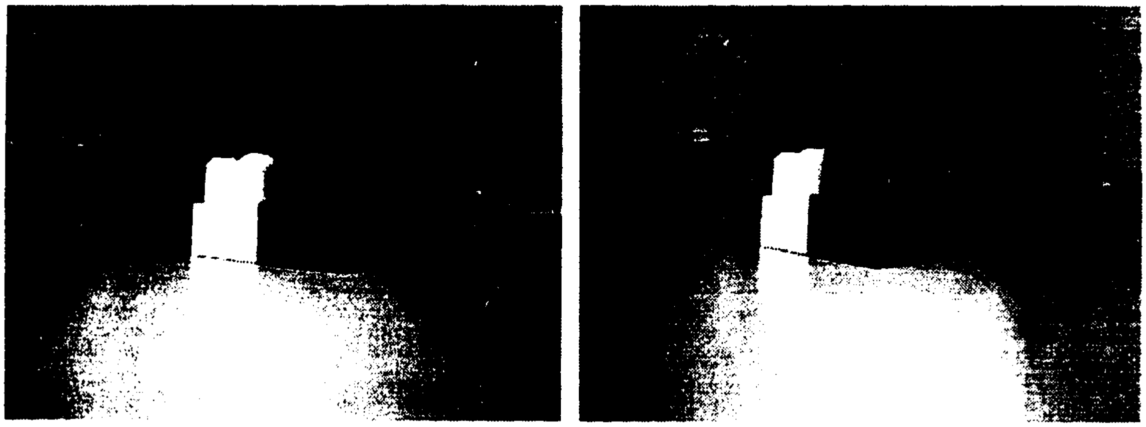


Figure 5.10: High compression stereopair from SVSDCT

5.2 Compression results for color images

For the color compression of stereopairs, the stereopair used also has left and right images with dimensions of 320x240 pixels. The vergence angle used was 5°, and the *RGB* components are all values between 0 and 255 inclusive. The format of the color images used was the PPM format which basically stores the width and height of the image followed by the *RGB* triplets representing the pixel colors.

5.2.1 Low compression ratios

As with the low compression ratio for the greyscale stereopair, we do not quantize the DCT coefficients. However, the loss in accuracy is no longer from the lack of precision in floating point operations only. Since we subsample the chrominance images before applying the DCT, there is reconstruction error in the expanding of the subsampled chrominance images. However, because the reconstruction error of the chrominance images has a lesser effect on the human eye, the reconstructed stereopair should still be indistinguishable from the original, from a subjective point of view. Both the left and right images have *RGB* components which range from 0 to 255, meaning each pixel needs three bytes for storage (8 bits per *RGB* component). In total, a color stereopair needs 460800 bytes to store the raw image information at 24 bits/pixel. When the uniform DCT is applied to the color stereopair, the resulting space needed to store the image is 121947 bytes. This averages out to approximately 6.35 bits/pixel. and gives a compression ratio of about 3.78:1. Because there was hardly any loss in quality, no block matches were found for this particular stereopair. The SVDCT and SVSDCT can be made to behave like the uniform DCT, so the compression results are the same for all three methods. Even though the chrominance images were subsampled, there was no noticeable loss in color quality or image quality from the original, just as there was no noticeable loss in quality for the low compression ratio and the greyscale stereopair.

5.2.2 Medium compression ratios

For the medium compression ratios, reconstructed stereopairs do not have the same quality as the original, but the important details of the image should be preserved. Acceptable quality loss with a good compression ratio is the main goal.

Using the uniform DCT, a value of 0.275 was used as the quantization matrix scaling factor for the luminance image, 1 was used as the quantization matrix scaling factor for the chrominance images, and 0 was used as the block matching threshold. The resulting storage needed for the compressed stereopair was 22191 bytes, which averages to approximately 1.16 bits/pixel. The compression ratio was approximately 20.77:1, and no block matches were found for the particular stereopair used.

For the SVDCT, a value of 0.2 was used as the base quantization matrix scaling factor for the luminance image, 1 was used as the quantization matrix scaling factor for the chrominance images, 3 was used as the value of n, 8 was used as the maximum quantization factor, 1 was used as the CCF, and 0 was used as the matching threshold. The resulting storage needed for the compressed stereopair was 22485 bytes, which averages to approximately 1.17 bits/pixel. The compression ratio was approximately 20.49:1, and again no block matches were found. The compression curve for the SVDCT medium compression parameters is shown in Figure 5.11.

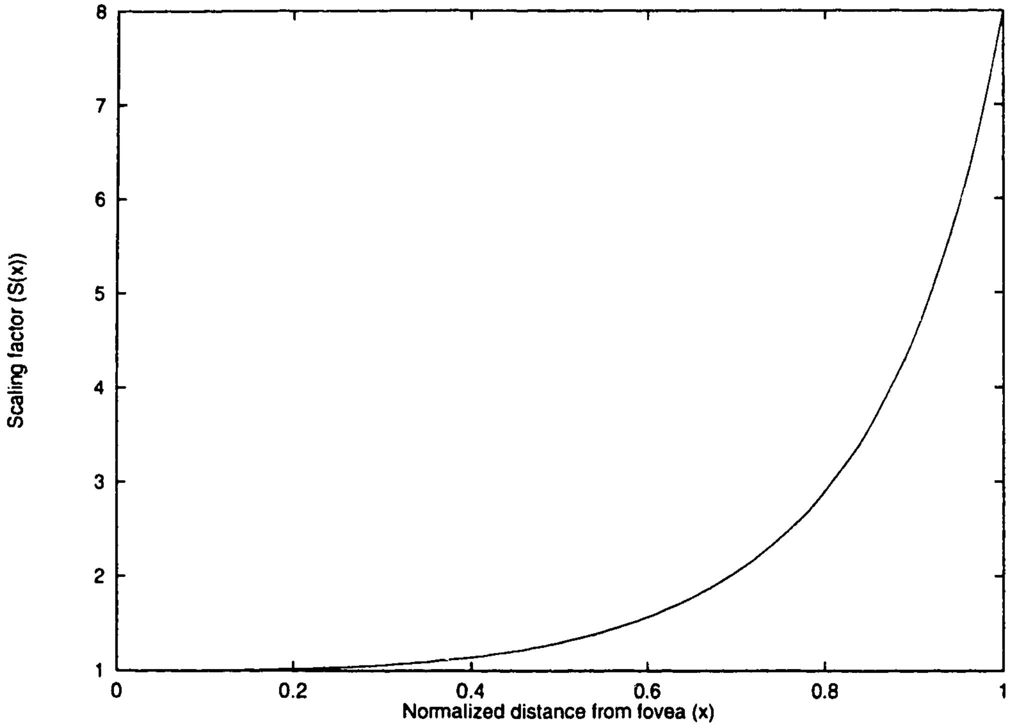


Figure 5.11: $S(x) = \exp(x^3 \ln(8))$

For the SVSDCT, a value of 0.2 was used as the quantization matrix scaling factor for the luminance image, 1 was used as the quantization matrix scaling factor for the chrominance images, 70% was used as the expected compression, 0.1 was used as the value for α, and 0 was used as the matching threshold. The resulting storage needed for the compressed stereopair was 22292 bytes, which averages to approximately 1.16 bits/pixel. The compression ratio was approximately 20.67:1, and no block matches were found.

The resulting images from all three compression methods used here produced images of the same quality as did the medium compression ratios for the greyscale images; however, the compression ratios are higher than that of the greyscale images because of the smaller size of the chrominance images.

5.2.3 High compression ratios

For the high compression ratios, there is a noticeable loss in image quality, so the goal here is to find the stereopair that is the most appealing, from the 3 compression methods.

For the uniform DCT, 2 was used as the quantization matrix scaling factor for the luminance image, 1 was used as the quantization matrix scaling factor for the chrominance images, and 0 was used for the matching threshold. The resulting storage needed for the compressed stereopair was 7192 bytes, which averages to approximately 0.37 bits/pixel. The compression ratio was approximately 64.07:1, and 388 block matches were found. The reconstructed stereopair displays the same type of uniform blockiness as its greyscale equivalent for high compression ratios using the uniform DCT.

For the SVDCT, 0.8 was used as the base quantization matrix scaling factor for the luminance image, 1 was used as the quantization matrix scaling factor for the chrominance images, 6 was used as the value for n, 6 was used as the maximum quantization factor, 1.9 was used as the CCF, and 0 was used as the matching threshold. The resulting storage needed for the compressed stereopair was 7130 bytes, which averages to approximately 0.37 bits/pixel. The compression ratio was approximately 64.63:1, and 423 block matches were found. The compression curve for the SVDCT high compression parameters is shown in Figure 5.12, and the reconstructed stereopair contains similar characteristics to the greyscale SVDCT at a high compression ratio, with the images being clear at the center and getting blockier at the edges.

For the SVSDCT, 0.5 was used as the quantization matrix scaling factor for the luminance image, 1 was used as the quantization matrix scaling factor for the chrominance images, 86% was the expected compression, 0.15 was used as the value for α, and 0 was used as the matching threshold. The resulting storage needed for the compressed stereopair was 7020 bytes, which averages to approximately 0.37 bits/pixel. The compression ratio was approximately 65:64:1, and 7 block matches were found.

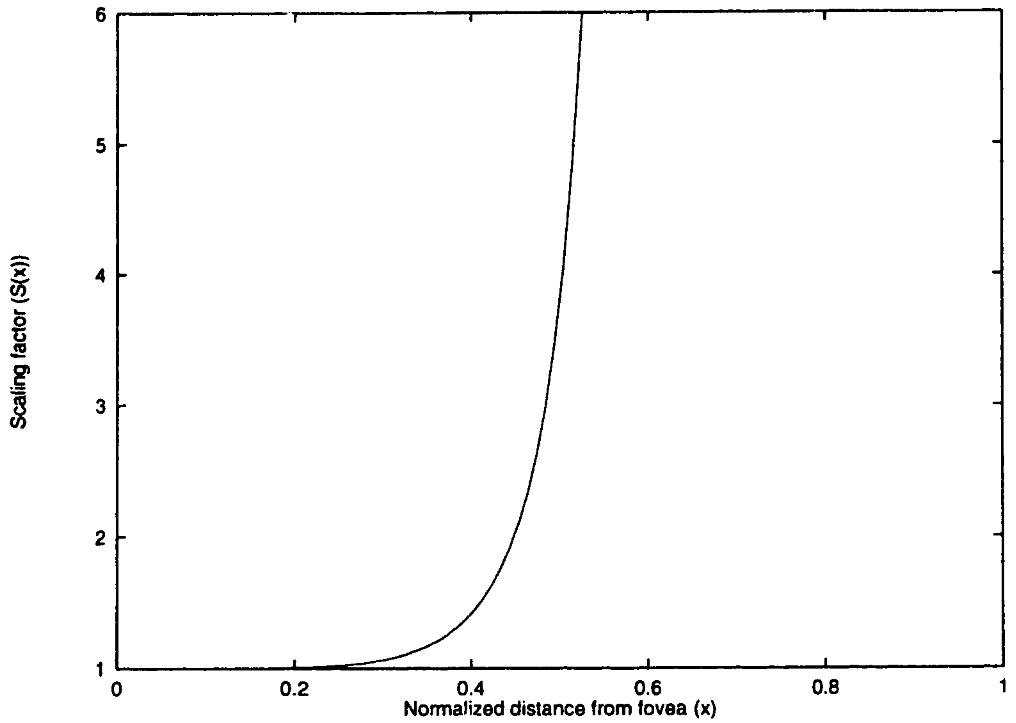


Figure 5.12: $S(x) = \exp((1.9x)^6 \ln(6))$

The reconstructed stereopair for the SVSDCT did not have the blocky appearance that the uniform DCT and the SVDCT had, which matches the behaviour of the SVSDCT for the greyscale image at a high compression ratio.

5.3 User survey results

The HMD used in the small user survey was the "I-glasses" from Virtual I/O. Since the stereopairs are taken with vergence, some of the subjects' eyes may not be able to adjust to a vergent stereopair. In these cases, the images are adjusted so that the images appear to have been taken with a parallel camera geometry. The adjusted stereopair is meant to simulate a parallel axes camera geometry where the left and right cameras have a wider field of view than the actual field of view of the cameras, and then the vergent stereopair is mapped onto the new image areas. A calibration stage was performed first with the parallel version of the vergent stereopair, as well as the vergent stereopair, shown to the subject to see which one he or she preferred to look at. Figure 5.13 illustrates the adjusted vergent stereopair for simulating a

Figure 5.13: Adjusted vergent stereopair for parallel stereopair

The survey was meant to test the quality of the reconstructed stereopairs on the HMD using the uniform DCT, SVDCT, and SVSDCT with different compression ratios. Questions on how the reconstructed stereopairs appeared compared to the original stereopair were asked in random order, but the same random order was used for every user surveyed. The users were also asked to give the order of preference for the medium and high compression ratio stereopairs, if they felt that there were advantages to some reconstructed images over others in identifying a possible defect in the main object in the stereopair. There were 19 people surveyed in total. The questions in the questionnaire are provided in Appendix A. The same questionnaire was used for both the greyscale and color stereopairs.

5.3.1 Greyscale

The first part of the survey dealt with the low compression ratio. The person was asked to rank the difference according to the subjective impairment scale in Figure 5.14. This is the same scale shown from the discussion of subjective fidelity criteria, but instead of the word "points", the word "rank" was chosen. The results are given in Figure 5.15 and the MOS (mean opinion score) from those results is 6.5. This high MOS supports the fact that the low compression ratio produces an image that is almost indistinguishable from the original.

The second part of the survey dealt with the medium compression ratios. The rankings for the reconstructed stereopairs from the uniform DCT, SVDCT, and SVSDCT are shown in Figure 5.16. The mean opinion scores for the three compression

Opinion	Rank
Not perceptible	7
Barely perceptible	6
Definitely perceptible but only slight impairment of the image	5
Impairment to the image but not objectionable	4
Somewhat objectionable	3
Definitely objectionable	2
Extremely objectionable	1

Figure 5.14: Subjective impairment scale used in survey

Rank	Occurrences
7	11
6	6
5	2
4	0
3	0
2	0
1	0

Figure 5.15: Low compression ratio scores

methods are 6.1 for the uniform DCT, 6.1 for the SVDCT, and 5.7 for the SVSDCT. Only 9 people thought the stereopairs needed an order of preference, and of those 9 people, 3 preferred the uniform DCT over the others, 4 preferred the SVDCT, and 2 preferred the SVSDCT. So, the survey results for the medium compression ratios seem to support the idea that the medium compression ratios do have some difference from the original, but not a very significant difference.

The third and final part of the survey dealt with the high compression ratios. The rankings for the uniform DCT, SVDCT, and SVSDCT are shown in Figure 5.17. The mean opinion scores for the three compression methods are 3.5 for the uniform DCT, 3.6 for the SVDCT, and 3.7 for the SVSDCT. This is consistent with the fact that the high compression ratios do produce images with a loss in image quality. There were 14 people that felt the stereopairs needed an order of preference according to which one would be most appealing to identify a possible defect in the object of interest. Of these 14 people, no one chose the stereopair from the uniform DCT over the others, 9 people chose the SVDCT stereopair, and 5 people chose the SVSDCT

Rank	Occurrences		
	Uniform DCT	SVDCT	SVSDCT
7	8	7	7
6	6	6	3
5	4	6	7
4	1	0	1
3	0	0	1
2	0	0	0
1	0	0	0

Figure 5.16: Medium compression ratio scores

stereopair. The preferences suggest that the compression methods using spatially varying sensing at high compression ratios produce more appealing images than the conventional uniform DCT method for a HMD.

Rank	Occurrences		
	Uniform DCT	SVDCT	SVSDCT
7	0	0	0
6	2	3	2
5	1	3	2
4	5	4	6
3	7	4	7
2	4	3	1
1	0	2	1

Figure 5.17: High compression ratio scores

5.3.2 Color

As in the greyscale survey, the first part of the color survey dealt with the low compression ratio. Each person was asked to rank the difference according to the subjective impairment scale in Figure 5.14. The results are given in Figure 5.18 and the MOS (mean opinion score) from those results is 6.8. This high MOS supports the fact that the low compression ratio produces an image that is almost indistinguishable from the original, just as in the greyscale survey.

The second part of the survey dealt with the high compression ratios. The rankings for the reconstructed stereopairs from the uniform DCT, SVDCT, and SVSDCT are

7	16
6	3
5	0
4	0
3	0
2	0
1	0

Figure 5.18: Low compression ratio scores

shown in Figure 5.19. The mean opinion scores for the three compression methods are 5.1 for the uniform DCT, 5.0 for the SVDCT, and 4.7 for the SVSDCT. There were 13 people that felt the stereopairs needed an order of preference. Of these 13 people, 2 chose the uniform DCT stereopair over the others, 7 people chose the SVDCT stereopair, and 4 people chose the SVSDCT stereopair. The preferences seem to suggest that the compression methods using spatially varying sensing produce a more appealing image for high compression ratios than the conventional uniform DCT method for the HMD.

Rank	Occurrences		
	Uniform DCT	SVDCT	SVSDCT
7	3	1	2
6	4	5	4
5	5	8	7
4	6	3	1
3	1	2	3
2	0	0	2
1	0	0	0

Figure 5.19: High compression ratio scores

The third and final part of the survey dealt with the medium compression ratios. The rankings for the uniform DCT, SVDCT, and SVSDCT are shown in Figure 5.20. The mean opinion scores for the three compression methods are 6.4 for the uniform DCT, 6.4 for the SVDCT, and 6.2 for the SVSDCT. Only 4 people thought the stereopairs needed an order of preference, and of those 4 people, 2 preferred the uniform DCT over the others, 1 preferred the SVDCT over the others, and 1 preferred seem to support the idea that the medium compression ratios do have some difference from the original, but not a significant difference.

Rank	Occurrences		
	Uniform DCT	SVDCT	SVSDCT
7	10	10	9
6	7	7	5
5	1	1	4
4	1	1	1
3	0	0	0
2	0	0	0
1	0	0	0

Figure 5.20: Medium compression ratio scores

The results of the survey suggest that there are no major differences between the uniform DCT, the SVDCT, and the SVSDCT for the low and medium compression ratios. For the high compression ratios, there does seem to be a difference in quality for the stereopairs. The SVDCT seems to be the most preferred in the case of high compression ratios. The SVSDCT is the second most preferred, followed by the uniform DCT which is least preferred.

Chapter 6

Conclusion

In this thesis, we try to enhance stereopair compression for use with a HMD in two ways:

- using a new type of stereopair called a vergent stereopair and
- integrating spatially varying sensing into the compression.

The first is meant to improve the depth information present in a stereopair. The second is intended to improve upon the conventional uniform loss of quality that many lossy compression methods use. The DCT is chosen as the transform for transform coding, and a straightforward stereopair compression method called the uniform DCT is described. To improve upon the uniform DCT, the concept of spatially varying sensing is used to create the SVDCT and the SVSDCT compression methods. All three compression methods have parameters that can be set to control the amount of compression in order to achieve low to high compression ratios. Compression of greyscale stereopairs was considered first, followed by compression of color stereopairs. Color stereopair compression used the YC_bC_r color model instead of the RGB color model to compress color information more efficiently. The results of a small user survey suggest that the SVDCT and SVSDCT compression methods are relatively the same, in terms of image quality, as the uniform DCT for low to medium compression ratios. When high compression ratios are used, the results of the survey suggest that the SVDCT and the SVSDCT produce better quality stereopairs than the uniform DCT. While uniform quality loss works well for lossy compression of single images viewed on a screen, spatially varying quality loss seems to work better than uniform quality loss for lossy compression of stereopairs viewed on a HMD. This improved compression performance was the intended result and could be a step in the right direction in creating a stereopair compression standard to prepare for the possible time when HMD devices become common.

Chapter 7

Future Work

Currently, the uniform DCT, SVDCT, and SVSDCT all have static Huffman code tables for all quantized coefficient blocks. For the higher compression ratios, it might be advantageous to allow for the specification of the Huffman code table used for a particular stereopair. There is a tradeoff between the Huffman code table size and the amount of storage space saved, but in some cases a custom Huffman code table yields a smaller compressed file size.

The MPEG standard for compression of a sequence of images is closely related to the three compression methods presented, since all use the DCT on 8x8 blocks, and all use block matching [19]. MPEG could be combined with any or all of the three compression methods presented to compress a sequence of stereopairs. With the growing popularity of MPEG hardware, it may be possible to use the DCT and block matching capabilities of the existing MPEG hardware in order to improve the performance of the stereo compression methods, and to implement a format for motion stereopairs.

There has been success using the wavelet transform to perform image compression. Wavelet compression is similar to compression using the DCT since both produce coefficients that are quantized and then encoded efficiently. The major difference between wavelet compression and DCT compression is the generation of the transform coefficients. The wavelet transform used in wavelet compression involves the selection of a wavelet function which has certain properties to try to efficiently pack image information into the fewest number of coefficients. Selection of a good wavelet function is not a trivial task, and a good discussion of the factors involved can be found in [10]. The wavelet function determines a set of basis functions or images which can be thought of as dilated and translated versions of the original wavelet function when in 1D. To illustrate the dilation and translation of the wavelet function, we will use a popular wavelet called the Haar wavelet in Figure 7.1. The first 8 basis functions in 1D are shown in Figure 7.2. In 2D, the wavelet transform produces a set of coefficients corresponding to basis images, and these coefficients can then be quantized for compression purposes. The wavelet transform can either be applied to the whole image or applied to subimages such as the 8x8 blocks used in this thesis. For the purposes of stereo image compression, the 8x8 blocks would most likely be used since it would

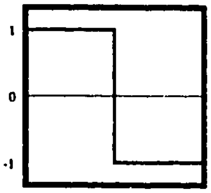


Figure 7.1: Haar wavelet

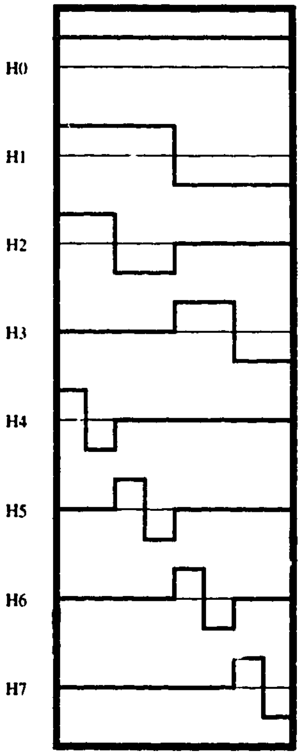


Figure 7.2: First 8 Haar bases

allow for block matching and for the spatially varying quantization of blocks. The current problem with using wavelet compression is that there is no wavelet function that is considered the "best wavelet". In the future, there may be a consensus as to which wavelet is generally best to use for most real world images. If a consensus is reached, then the option of using the wavelet transform instead of the DCT to generate coefficients may become more attractive. Using wavelet compression will have the disadvantage of not being able to take advantage of MPEG hardware but, depending on which wavelet is chosen, the processing time of the wavelet transform could be significantly less than that required of the DCT when there is no hardware assistance.

There is a property of the human visual system that has not been addressed using any of the three compression methods in this thesis. It is called the *singleness of vision property* [13] and is used in [7, 8]. Basically, this property suggests that when both eyes are given the two different images, only one side is perceived to have good resolution in an area while the same area in the other side is suppressed. Theoretically one side could have compression parameters to preserve much image quality, while the other side would have compression parameters that would eliminate much of the image quality, but still keep enough information for depth perception. Since the dominant eye is not on the same side for everyone and the degree of dominance is also not the same, it is difficult to apply the *singleness of vision property* to stereopair compression. For personal use, however, it may be useful to modify the stereopair compression methods to support different compression parameters for each side.

Bibliography

[1] 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, 1994.

[2] A. Basu, A. Sullivan, and K. J. Wiebe. Variable resolution teleconferencing. In *IEEE SMC Conference Proceedings*, France, Oct 1993.

[3] Jim Blinn. What's the deal with the DCT? *IEEE Computer Graphics and Applications*, pages 78–83, July 1993.

[4] David Bourgin. Color spaces FAQ. Available via anonymous ftp from rtfm.mit.edu as /pub/usenet/news.answers/graphics/colorspace-faq.

[5] Shing-Chow Chan and Ka-Leung Ho. Fast algorithms for computing the discrete cosine transform. *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, pages 185-190, March 1992.

[6] R. J. Clarke. *Transform Coding of Images.* Academic Press Inc. (London) Ltd., 1985.

[7] I. Dinstein, M. G. Kim, A. Henik, and J. Tzelgov. Compression of stereo images using subsampling and transform coding. *Optical Engineering*, 30(9):1359 1364, 1991.

[8] I. Dinstein, M. G. Kim, J. Tselgov, and A. Henik. Compression of stereo images and the evaluation of its effects on 3-d perception. *Applications of Digital Image Processing XII*, pages 522–530, 1989.

[9] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice.* Addison Wesley, second edition, 1990.

[10] Alain Fournier. *Wavelets and their applications in computer graphics.* Siggraph '95 Course notes available via anonymous ftp from ftp.cs.ubc.ca as /pub/local/bobl/wvlt/notes.ps.Z.

[11] Didier Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, April 1992.

[12] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley, 1992.

[13] Lloyd Kaufman. *Sight and Mind, an Introduction to Visual Perception.* Oxford University Press, 1974.

[14] Alain Léger, Takao Omachi, and Gregory K. Wallace. JPEG still picture compression algorithm. *Optical Engineering*, 30(7):947–954, 1991.

[15] W. Li and E. Salari. Efficient coding method for stereo image pairs. In *Proceedings of the SPIE*, volume 2094, pages 1470–1476, 1993.

[16] William Mendenhall, Dennis D. Wackerly, and Richard L. Schaeffer. *Mathematical Statistics with Applications.* PWS-KENT Publishing Company, fourth edition, 1990.

[17] W. B. Pennebaker and J. L. Mitchell. *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold, 1993.

[18] M. G. Perkins. Data compression of stereopairs. *IEEE Transactions on Communications*, 40:684–696, 1992.

[19] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications.* Academic Press, 1990.

[20] Hossein Sahabi and Anup Basu. Analysis of error in depth perception with vergence and spatially varying sensing. *Computer Vision and Image Understanding*, 63:447–461, May 1996.

[21] V. E. Seferidis and D. V. Papadimitriou. Improved disparity estimation in stereoscopic television. *Electronics Letters*, 29(9):782–783, 1993.

[22] M. W. Siegel, Priyan Gunatilake, Sriram Sethuraman, and A. G. Jordan. Compression of stereo image pairs and streams. In *Proceedings of the SPIE*, volume 2177, pages 258–268, 1994.

[23] Brandi Vidaković and Peter Müller. *Wavelets for Kids: A Tutorial Introduction*. Duke University, 1991. Available via anonymous ftp from ftp.isds.duke.edu as /pub/Users/brani/papers/wav4kidsA.ps.Z.

[24] Gregory K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1), February 1992.

[25] N. P. Walmsley, A. N. Skodras, and K. M. Curtis. A fast picture compression technique. *IEEE Transactions on Consumer Electronics*, 40(1):11–16, February 1994.

Appendix A

Questionnaire for the user survey

Instructions

It is probably best to read this entire thing before you put the headset on and begin. The images are viewed by pressing the key for the image you would like to view (for example, *o* for image O, *b* for image B, etc). When finished viewing an image, press escape or space bar to return to the prompt where you can select another image.

A.1 First section

The original stereopair is image O. Compare the original to image A and rank the difference between the two images according to the following table:

Opinion	Rank
Not perceptible	7
Barely perceptible	6
Definitely perceptible but only slight impairment of the image	5
Impairment to the image but not objectionable	4
Somewhat objectionable	3
Definitely objectionable	2
Extremely objectionable	1

The difference between the original and image A is ____________________.

A.2 Second section

For this section, you will be using the same original image and the three images B, C, and D. You will be ranking the difference between the original and the three images according to the table above. All the rankings may be the same. There is no requirement that the rankings be unique.

The difference between the original and image B is ________________________.
The difference between the original and image C is ________________________.
The difference between the original and image D is ________________________.

If you were to try to identify a defect in the main object in the image, would B, C, and D be equally appealing to identify the defect?

Yes **No**

If not, then list the order of preference of B, C, and D. ________________________

A.3 Third section

For this section, you will be using the original image O again and the three images E, F, and G. You will be ranking the difference between the original and the three images according to the table in the first section. All the rankings may be the same. There is no requirement that the rankings be unique.

The difference between the original and image E is ________________________.
The difference between the original and image F is ________________________.
The difference between the original and image G is ________________________.

If you were to try to identify a defect in the main object in the image, would E, F, and G be equally appealing to identify the defect?

Yes **No**

If not, then list the order of preference of E, F, and G. ________________________

quality loss for lossy compression of stereopairs viewed on a HMD. This improved compression performance was the intended result and could be a step in the right direction in creating a stereopair compression standard to prepare for the possible time when HMD devices become common.

Chapter 7

Future Work

Currently, the uniform DCT, SVDCT, and SVSDCT all have static Huffman code tables for all quantized coefficient blocks. For the higher compression ratios, it might be advantageous to allow for the specification of the Huffman code table used for a particular stereopair. There is a tradeoff between the Huffman code table size and the amount of storage space saved, but in some cases a custom Huffman code table yields a smaller compressed file size.

The MPEG standard for compression of a sequence of images is closely related to the three compression methods presented, since all use the DCT on 8x8 blocks, and all use block matching [19]. MPEG could be combined with any or all of the three compression methods presented to compress a sequence of stereopairs. With the growing popularity of MPEG hardware, it may be possible to use the DCT and block matching capabilities of the existing MPEG hardware in order to improve the performance of the stereo compression methods, and to implement a format for motion stereopairs.

There has been success using the wavelet transform to perform image compression. Wavelet compression is similar to compression using the DCT since both produce coefficients that are quantized and then encoded efficiently. The major difference between wavelet compression and DCT compression is the generation of the transform coefficients. The wavelet transform used in wavelet compression involves the selection of a wavelet function which has certain properties to try to efficiently pack image information into the fewest number of coefficients. Selection of a good wavelet function is not a trivial task, and a good discussion of the factors involved can be found in [10]. The of as dilated and translated versions of the original wavelet function when in 1D. To illustrate the dilation and translation of the wavelet function, we will use a popular wavelet called the Haar wavelet in Figure 7.1. The first 8 basis functions in 1D are shown in Figure 7.2. In 2D, the wavelet transform produces a set of coefficients corresponding to basis images, and these coefficients can then be quantized for compression purposes. The wavelet transform can either be applied to the whole image or applied to subimages such as the 8x8 blocks used in this thesis. For the purposes of stereo image compression, the 8x8 blocks would most likely be used since it would

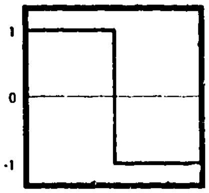


Figure 7.1: Haar wavelet

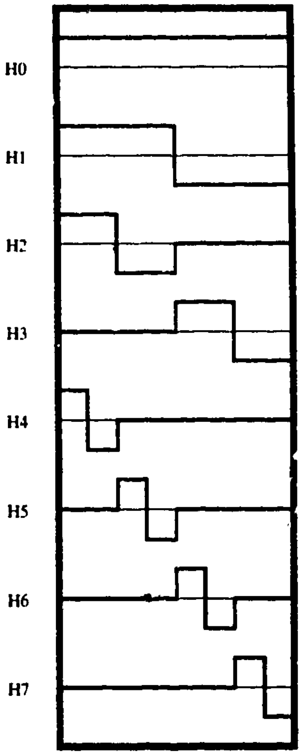


Figure 7.2: First 8 Haar bases

current problem with using wavelet compression is that there is no wavelet function that is considered the "best wavelet". In the future, there may be a consensus as to which wavelet is generally best to use for most real world images. If a consensus is reached, then the option of using the wavelet transform instead of the DCT to generate coefficients may become more attractive. Using wavelet compression will have the disadvantage of not being able to take advantage of MPEG hardware but, depending on which wavelet is chosen, the processing time of the wavelet transform could be significantly less than that required of the DCT when there is no hardware assistance.

There is a property of the human visual system that has not been addressed using any of the three compression methods in this thesis. It is called the *singleness of vision property* [13] and is used in [7, 8]. Basically, this property suggests that when both eyes are given the two different images, only one side is perceived to have good resolution in an area while the same area in the other side is suppressed. Theoretically one side could have compression parameters to preserve much image quality, while the other side would have compression parameters that would eliminate much of the image quality, but still keep enough information for depth perception. Since the dominant eye is not on the same side for everyone and the degree of dominance is also not the same, it is difficult to apply the *singleness of vision property* to stereopair compression. For personal use, however, it may be useful to modify the stereopair compression methods to support different compression parameters for each side.

Bibliography

[1] 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, 1994.

[2] A. Basu, A. Sullivan, and K. J. Wiebe. Variable resolution teleconferencing. In *IEEE SMC Conference Proceedings*, France, Oct 1993.

[3] Jim Blinn. What's the deal with the DCT? *IEEE Computer Graphics and Applications*, pages 78–83, July 1993.

[4] David Bourgin. Color spaces FAQ. Available via anonymous ftp from rtfm.mit.edu as /pub/usenet/news.answers/graphics/colorspace-faq.

[5] Shing-Chow Chan and Ka-Leung Ho. Fast algorithms for computing the discrete cosine transform. *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, pages 185–190, March 1992.

[6] R. J. Clarke. *Transform Coding of Images*. Academic Press Inc. (London) Ltd., 1985.

[7] I. Dinstein, M. G. Kim, A. Henik, and J. Tzelgov. Compression of stereo images using subsampling and transform coding. *Optical Engineering*, 30(9):1359 1364, 1991.

[8] I. Dinstein, M. G. Kim, J. Tselgov, and A. Henik. Compression of stereo images and the evaluation of its effects on 3-d perception. *Applications of Digital Image Processing XII*, pages 522–530, 1989.

[9] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison Wesley, second edition, 1990.

/pub/local/bobl/wvlt/notes.ps.Z.

[11] Didier Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, April 1992.

[12] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[13] Lloyd Kaufman. *Sight and Mind, an Introduction to Visual Perception*. Oxford University Press, 1974.

[14] Alain Léger, Takao Omachi, and Gregory K. Wallace. JPEG still picture compression algorithm. *Optical Engineering*, 30(7):947–954, 1991.

[15] W. Li and E. Salari. Efficient coding method for stereo image pairs. In *Proceedings of the SPIE*, volume 2094, pages 1470–1476, 1993.

[16] William Mendenhall, Dennis D. Wackerly, and Richard L. Schaeffer. *Mathematical Statistics with Applications*. PWS-KENT Publishing Company, fourth edition, 1990.

[17] W. B. Pennebaker and J. L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, 1993.

[18] M. G. Perkins. Data compression of stereopairs. *IEEE Transactions on Communications*, 40:684–696, 1992.

[19] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990.

[20] Hossein Sahabi and Anup Basu. Analysis of error in depth perception with vergence and spatially varying sensing. *Computer Vision and Image Understanding*, 63:447–461, May 1996.

[21] V. E. Seferidis and D. V. Papadimitriou. Improved disparity estimation in stereoscopic television. *Electronics Letters*, 29(9):782–783, 1993.

2177, pages 258–268, 1994.

[23] Brandi Vidaković and Peter Müller. *Wavelets for Kids: A Tutorial Introduction.* Duke University, 1991. Available via anonymous ftp from ftp.isds.duke.edu as /pub/Users/brani/papers/wav4kidsA.ps.Z.

[24] Gregory K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1), February 1992.

[25] N. P. Walmsley, A. N. Skodras, and K. M. Curtis. A fast picture compression technique. *IEEE Transactions on Consumer Electronics*, 40(1):11–16, February 1994.

Appendix A

Questionnaire for the user survey

Instructions

It is probably best to read this entire thing before you put the headset on and begin. The images are viewed by pressing the key for the image you would like to view (for example, *o* for image O, *b* for image B, etc). When finished viewing an image, press escape or space bar to return to the prompt where you can select another image.

A.1 First section

The original stereopair is image O. Compare the original to image A and rank the difference between the two images according to the following table:

Opinion	Rank
Not perceptible	7
Barely perceptible	6
Definitely perceptible but only slight impairment of the image	5
Impairment to the image but not objectionable	4
Somewhat objectionable	3
Definitely objectionable	2
Extremely objectionable	1

The difference between the original and image A is ________________________.

A.2 Second section

For this section, you will be using the same original image and the three images B, C, and D. You will be ranking the difference between the original and the three images according to the table above. All the rankings may be the same. There is no requirement that the rankings be unique.

If you were to try to identify a defect in the main object in the image, would B, C, and D be equally appealing to identify the defect?

Yes **No**

If not, then list the order of preference of B, C, and D. ______________________

A.3 Third section

For this section, you will be using the original image O again and the three images E, F, and G. You will be ranking the difference between the original and the three images according to the table in the first section. All the rankings may be the same. There is no requirement that the rankings be unique.

The difference between the original and image E is ______________________.
The difference between the original and image F is ______________________.
The difference between the original and image G is ______________________.

If you were to try to identify a defect in the main object in the image, would E, F, and G be equally appealing to identify the defect?

Yes **No**

If not, then list the order of preference of E, F, and G. ______________________

07

EN

7-0

FI

ND

5-9

IN

J

97