A real-time foveated multiresolution system for low-bandwidth video communication

Wilson S. Geisler and Jeffrey S. Perry

Center for Vision and Image Sciences, University of Texas, Austin, TX 78712

ABSTRACT

Foveated imaging exploits the fact that the spatial resolution of the human visual system decreases dramatically away from the point of gaze. Because of this fact, large bandwidth savings are obtained by matching the resolution of the transmitted image to the fall-off in resolution of the human visual system. We have developed a foveated multiresolution pyramid (FMP) video coder/decoder which runs in real-time on a general purpose computer (i.e., a Pentium with the Windows 95/NT OS). The current system uses a foveated multiresolution pyramid to code each image into 5 or 6 regions of varying resolution. The user-controlled foveation point is obtained from a pointing device (e.g., a mouse or an eyetracker). Spatial edge artifacts between the regions created by the foveation are eliminated by raised-cosine blending across levels of the pyramid, and by "foveation point interpolation" within levels of the pyramid. Each level of the pyramid is then motion compensated, multiresolution pyramid coded, and thresholded/quantized based upon human contrast sensitivity as a function of spatial frequency and retinal eccentricity. The final lossless coding includes zero-tree coding. Optimal use of foveated imaging requires eye tracking; however, there are many useful applications which do not require eye tracking.

Key words: foveation, foveated imaging, multiresolution pyramid, video, motion compensation, zero-tree coding, human vision, eye tracking, video compression

1. INTRODUCTION

When a communication involves transmitting information that will ultimately be consumed by human observers, it is often possible to reduce transmission bandwidth requirements by exploiting the limitations of human perception. Specifically, bandwidth requirements can be lowered by transmitting only that information which the human sensory systems are capable of encoding and using. Four major human perceptual limitations have been exploited in the development of real-time video communication systems. First, the temporal contrast sensitivity of the human visual system declines at high frequencies creating a temporal resolution cutoff of approximately 60 Hz. Second, the spatial contrast sensitivity of the human visual system declines at high frequencies creating a spatial resolution cutoff of approximately 50 cycles per degree (cpd). Third, chromatic information is encoded in the human visual system by only three broad-band photoreceptors, with peak sensitivities at 440, 540 and 570 nm. Fourth, the chromatic spatial resolution of the human visual system is lower than the luminance spatial resolution by a factor of approximately two.

There is, however, a fifth major human perceptual limitation that has not been fully exploited. Namely, the spatial resolution of the human visual system declines dramatically and smoothly away from the point of fixation (direction of gaze) such that the resolution cutoff is reduced at a factor of two at 2.5 degrees from the point of fixation, and by a factor of ten at 20 degrees. In principle, large savings in transmission bandwidth can be obtained by matching the spatial resolution of the transmitted images to the fall off in spatial resolution of the human visual system.

Acceptance of foveated imaging as a useful image compression tool has been slow to develop because perceptually lossless (or nearly lossless) systems generally require tracking the position of the eye in real time, so that the high resolution region of the display can be kept aligned with the high resolution region of the eye (the fovea). Although eye tracking is practical in some applications, it is relatively expensive and complicated. However, a strong case can be made for the value of foveated imaging in a number of situations where eyetracking is not practical (see section **11**).

There have been attempts to use foveated imaging in low-bandwidth video communications. Early real-time systems used special purpose hardware, and created foveated images by increasing pixel-element size as a function of angular distance

(eccentricity) from the point of fixation.[1-5] More recently, Silsbee, Bovik & Chen[6] describe a foveated block pattern matching algorithm, which Barnett & Bovik[7] subsequently demonstrated has good real-time performance. Similarly, two years ago, Kortum & Geisler[8] described a real-time foveated imaging system that uses a general-purpose computer, and standard camera hardware. The system is able to foveate 8-bit, 256x256 images at around 18 frames/sec. However, all of these systems suffer from two important limitations: (1) the appearance of blocking artifacts and/or motion aliasing in the periphery with moderate degrees of foveation, and (2) the lack of a natural path for incorporating recent advances in multiresolution methods of image compression. To address these limitations, we have begun development of a real-time system for foveated imaging which is based upon multiresolution pyramid coding (see also, Chang & Yap.[9]) With multiresolution pyramid coding, an image is decomposed into a pyramid of 2D arrays of coefficients representing different spatial frequency bands. The first level of the pyramid contains the greatest number of coefficients and the highest spatial frequency band. Each successive level of the pyramid contains one fourth the number of coefficients of the previous level, and encodes the band of spatial frequencies centered at one half of the center spatial frequency of the previous level.

The foveated multiresolution pyramid (FMP) imaging system described here uses standard PCs, running Windows 95 or Windows NT, and does not require special purpose signal processing hardware.

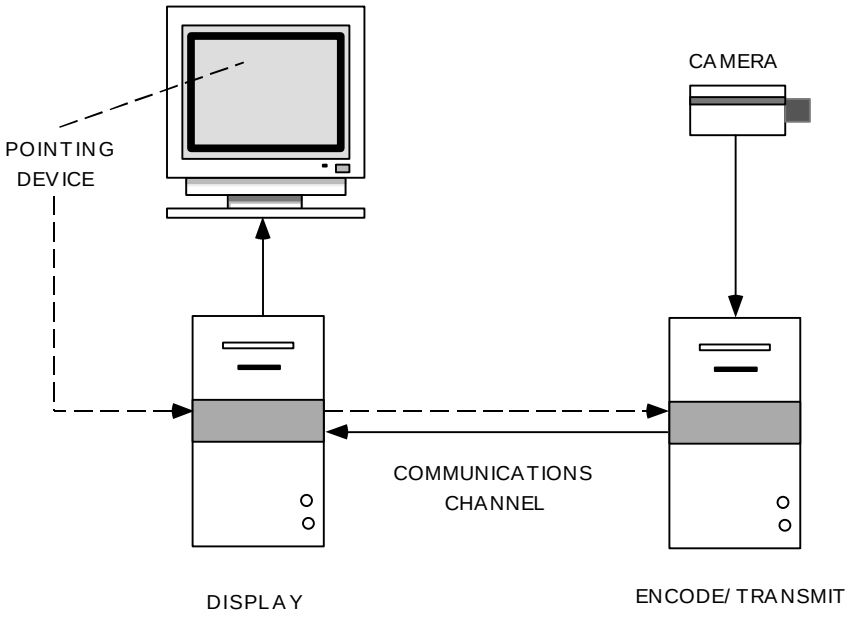

Figure 1. A foveated imaging system that is appropriate for tasks such as surveillance, teleoperation and telemedicine.

2. SYSTEM OVERVIEW

There are many potential applications of foveated imaging in real-time video communications. In some of these applications, such as surveillance, teleoperation and telemedicine, a user at one location controls the image data received from a camera at a remote location. The operation of a foveated imaging system in these applications is illustrated in Figure 1. First, the location of a foveation point is determined in real time (frame-by-frame) using some pointing device. The pointing device might be a mouse, a touch pad, or an eyetracker. The foveation point is the image location where the image will be displayed at highest resolution. Second, the coordinates of the foveation point are transmitted to the remote computer. Third, the remote computer captures a camera image. Fourth, the camera image is foveated; that is, the camera image is encoded so that the resolution of the image decreases away from the foveation point. The net result is that the degree of data compression increases with the distance from the foveation point. Fifth, the encoded image is transmitted to the local computer. Sixth, the received image is decoded and displayed on the video monitor such that the highest resolution region is centered at the foveation point. These six steps are repeated continuously in a closed loop.

A flow diagram illustrating the sequence of processing for the encoding and decoding of video image data in the FMP imaging system is given in Figure 2. Once the foveation point has been received at the remote computer, formulas based upon human psychophysical data are used to determine a foveation region for each level of the multiresolution pyramid. The foveation region is the set of pyramid elements in a level that will be further processed; no computations are done outside this region. Because spatial resolution decreases away from the fixation point, the foveation regions cover smaller fractions of the image at the lower levels of the pyramid.

An important advantage of implementing foveation in a multiresolution pyramid is that it is unnecessary to process pyramid coefficients outside the foveation region, in any given level. This makes the computation time of *every* step in the foveated codec substantially less than the computation time for a comparable non-foveated codec.

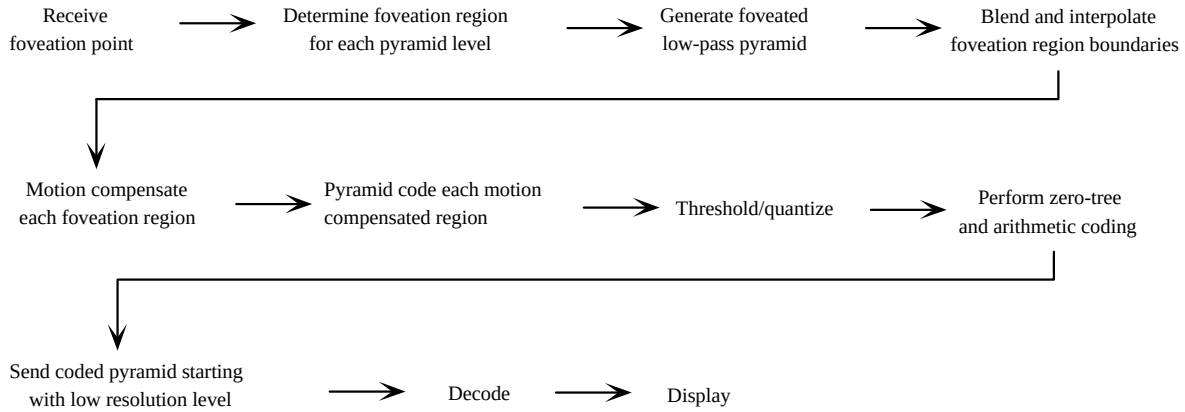

Figure 2. General flow diagram for the foveated multiresolution pyramid (FMP) imaging system.

The next step is to compute a foveated low-pass multiresolution pyramid. We use the "reduce" operation of a simple Laplacian pyramid,[10] and then select the foveated regions in each level for further processing. The next step consists of blending and foveation-point interpolation. Blending creates a smooth transition between levels of the pyramid at the boundaries created by foveation. Foveation-point interpolation incorporates the fact that a one pixel shift in the foveation point at a given level of the pyramid corresponds to fractions of a pixel shift at higher levels of the pyramid. Blending and foveation-point interpolation are important for producing smooth, artifact free foveation. The next step is to find local motion estimates for each foveation region in the pyramid, by comparing the current frame with the previous frame. We use a hierarchical block estimation method. The heirarchical motion estimation can make use of the same multiresolution pyramid used for foveation. The local motion estimates are then used to motion compensate each foveated region in the pyramid. (For a general review of motion compensation, see Tekalp.[11]) Motion compensation of each level of the pyramid is important in foveated imaging because it allows for faster processing; specifically, the compensation is only applied to the image data that will actually be transmitted. Next, each compensated foveation region is separately coded in a multiresolution pyramid. In the current version we use the Laplacian pyramid because of its excellent real-time performance; however, we expect useable real-time performace (and better compression) with a wavelet pyramid.[12, 13] The next step is to threshold and quantize the pyramid coefficients. A great deal of flexibility is available, but we have obtained good results using psychophysical measurements (contrast sensitivity data) as the basis for thresholding and quantizing as a function of both spatial frequency (level of the pyramid) and eccentricity (distance from the foveation point). The next step is lossless coding, which includes zero tree and arithmetic coding. Zero-tree coding exploits the fact that coefficients which are zero at a given level of the pyramid are likely to be superordinate to zeros in the lower levels, and thus it is often possible to code a whole "tree" of zeros with a special symbol.[14, 15] Following the lossless coding, the image data are transmitted beginning with the highest level of the pyramid (i.e., the lowest resolution data). Finally, the received data are decoded and displayed. We now describe each of these steps in more detail.

Note that two multiresolution pyramids are computed, one for foveation/motion-estimation, and another for final coding. Although this may seem inefficient, it is not. The simple, but fast, initial pyramid is sufficient for foveation and motion estimation. The foveation quickly strips away all of the image data that does not need to be processed further. The initial

pyramid also allows for very fast motion compensation, which (in our experience) must occur before final pyramid coding in order to be most effective. The more complex final pyramid coding is applied to the smallest amount of data possible.

3. FOVEATED LOW-PASS MULTIRESOLUTION PYRAMID

Our method of computing the foveated low-pass pyramid is illustrated in Figure 3. The first step is to perform a "reduce" operation like that used in a Laplacian pyramid.[10] The input image (level 1) is low-pass filtered and then down-sampled by a factor of two in both directions to obtain a lower resolution image (level 2) with one quarter the number of elements. This process of low-pass filtering and down sampling is repeated to obtain a sequence of successively lower resolution images; typically five or six resolution levels are computed, although only four are shown in Figure 3. From each of the levels we then select regions which define the amount of foveation. The inner solid squares in the upper row show the outer boundry of the foveation regions which are illustrated in the lower row. The inner dashed squares show the region in a level of the pyramid represented by the solid square in the previous level; they determine the inner boundaries of the foveation regions. In other words, the shaded regions indicate the image elements that will be processed further. In practice the inner boundaries are made a little smaller to allow blending between pyramid levels (see below). As can be inferred from this diagram, foveation can dramatically reduce both the amount of image data that must be coded and transmitted, and the total number of computations that must be performed.

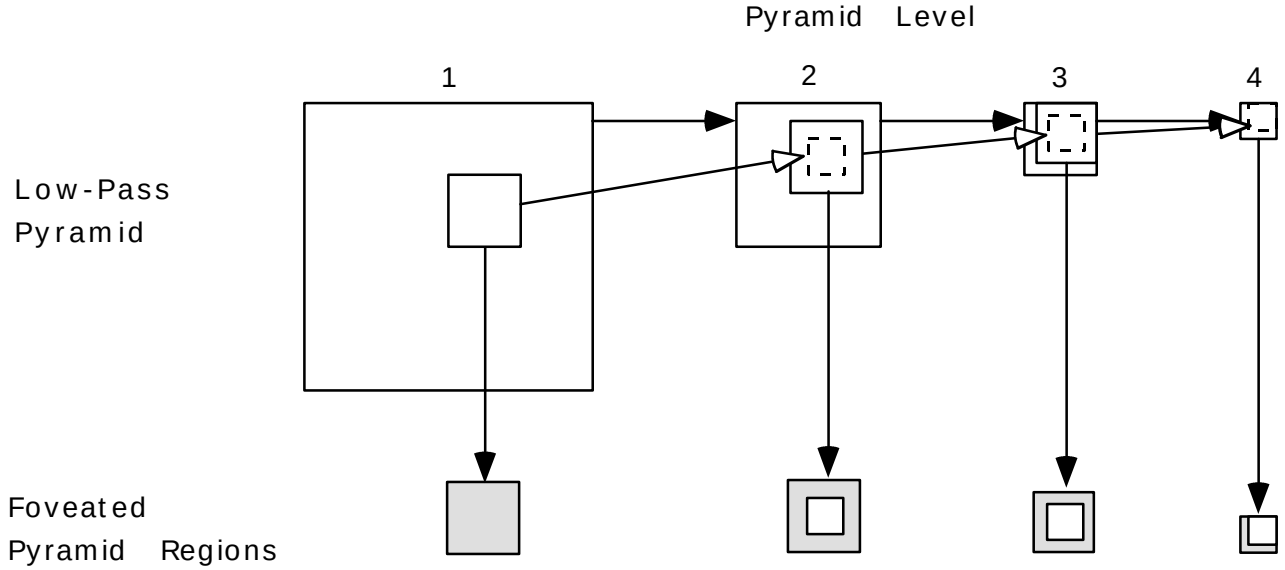

Figure 3. Schematic for the computation of a foveated low-pass multiresolution pyramid.

In the current system, the foveation regions are determined using the following contrast threshold formula, which is based upon human contrast sensitivity data measured as a function of spatial frequency and retinal eccentricity:

$$CT(f,e) = CT_0 \exp\left(af\,\frac{e+e_2}{e_2}\right) \qquad (1)$$

where f is spatial frequency (cycles per degree), e is the retinal eccentricity (degrees), CT_0 is the minimum contrast threshold, a is the spatial frequency decay constant, and e_2 is the half-resolution eccentricity. This formula was selected because of its simplicity and because it fits published contrast sensitivity data for small, briefly presented patches of grating, which are the most relevant contrast sensitivity data for predicting detectability under naturalistic viewing conditions. The solid curves in Figure 4 show the fit of equation (1) to the contrast sensitivity data (symbols connected by dashed lines) of Robson & Graham[16]. Equation (1) also provides an adequate fit to the data of Arnow & Geisler[17] and Banks et al.[18] (see the caption to Figure 4).

Equation (1) can be used to find the critical distance from the foveation point, e_c, beyond which a given spatial frequency will be invisible (below threshold) no matter what its contrast. Specifically, the critical eccentricity can be found by setting the left side of equation (1) to 1.0 (the maximum contrast) and solving for *e*:

$$e_c = \frac{e_2}{af} \ln\left(\frac{1}{CT_0}\right) - e_2 \qquad (2)$$

To apply equation (2), we convert into pixel units by taking into account viewing distance, and we set *f* to be the Nyquist frequency associated with each level of the pyramid (the highest frequency that can be reliably represented at that level). The resulting values of e_c (and the foveation point, x_0, y_0) define the foveation regions for each level of the pyramid.

Matching the foveation to the falloff in resolution of the human visual system with eccentricity makes optimal use of foveation, because it removes just that image information which cannot be resolved. However, in practice, we allow the user to control the degree of foveation by selecting the minimum contrast threshold, CT_0, and a minimum unfoveated radius, r_0. Raising CT_0 above the psychophysically measured value produces visible image degradation which is distributed across the visual field; however, there are many tasks where some distributed degradation will not reduce user performance. Similarly, there are tasks where it is important that the unfoveated region of the image not be less than some minimum size.

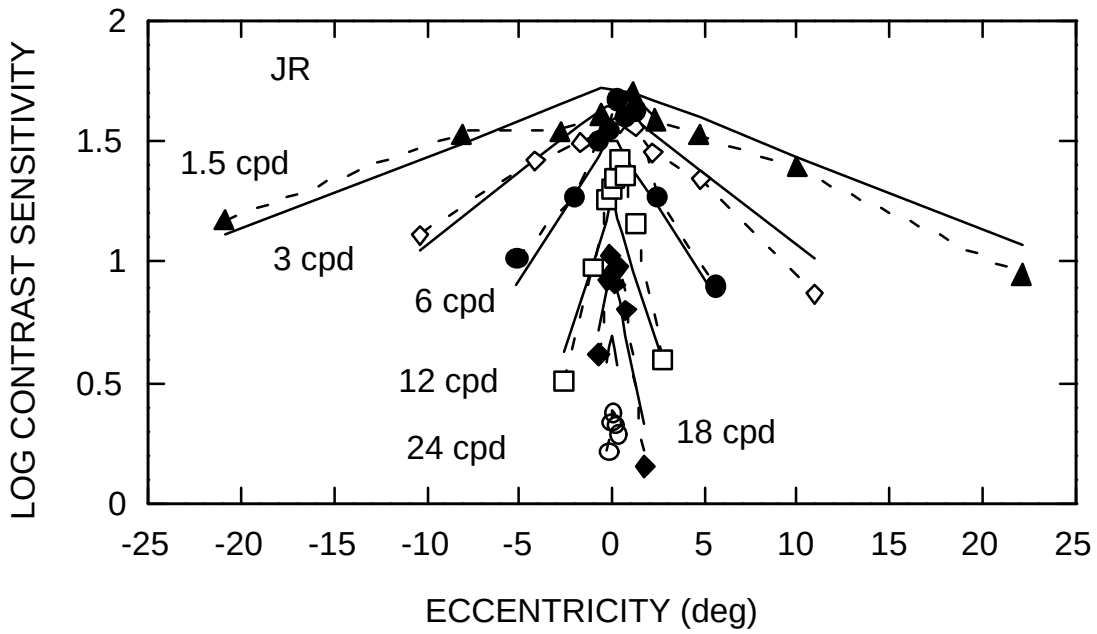

Figure 4. Contrast sensitivity (1/contrast threshold) for patches of sinusoidal grating as function of retinal eccentricity (degrees of visual angle), for a range of spatial frequencies. The symbols and connecting dashed lines are the measurements reported by Robson & Graham (1981); the solid curves are the predictions of equation (1). The best fitting parameter values (least squares fit in log units) are: $a = 0.106$, $e_2 = 2.3$, $CT_0 = 1/64$. The same parameters values for a and e_2 provide a good fit to the contrast sensitivity data of Arnow & Geisler[17] with $CT_0 = 1/75$, and an adequate fit to the data of Banks et al.[18] with $CT_0 = 1/76$.

4. BLENDING AND INTERPOLATION

In foveated multiresolution pyramids there can be visible boundaries at the edges of the foveation regions, where the spatial frequency content usually changes abruptly. These foveation boundary artifacts are most visible when there is image motion or movement of the foveation point. However, they can be minimized by applying a blending function, in our case a

raised cosine function, near the border of the foveation region at each level of the low-pass pyramid. Specifically, we multiply the outer edge of the foveation region by the following blending function:

$$b(x,y) = \begin{cases} 0.5\cos\left(\frac{p(e-e_c+w)}{w}\right)+0.5 & \text{if} \quad e_c - w < e < e_c \\ 1 & \text{if} \quad e \le e_c - w \\ 0 & \text{if} \quad e \ge e_c \end{cases} \tag{3}$$

where, $e = \sqrt{(x-x_0)^2 - (y-y_0)^2}$ and w is the width of the blending region. The inner edge is multiplied by a similar function, but with the width of the blending region set to $w/2$.

Another kind of artifact can arise when the foveation point is moved. The simplest method of foveating is to set the foveation region boundary to fall at the nearest element consistent with the value of e_c given by equation (2). However, the area of the image represented by an element increases by a factor of 4 at each level of the pyramid, and thus, for example, the foveation point must move a distance of 16 pixels in the image for the foveation boundary to move 1 element in the fifth level of the pyramid. As a result, when the foveation point is moved smoothly, the boundaries of the foveation regions in the reconstructed image jump abruptly by a distance that increases as the level of the pyramid increases. As might be expected, these jumps are most apparent for the boundaries in the higher levels of the pyramid. This problem can be effectively handled by interpolation at the foveation boundaries. Let, $x_0' + \Delta x, y_0' + \Delta y$ be the location of the foveation point, for some level of the pyramid, expressed in units of elements. In this notation, x_0' and y_0' are integers which represent the location of a whole element (the truncated coordinates of the foveation point), and Δx and Δy are fractions between 0 and 1 which represent offsets from the whole element. Now, let x_l, y_l and x_h, y_h be the lower left and upper right corners of the foveation region assuming a foveation point exactly at x_0', y_0'. To interpolate, we obtain the slightly larger foveation region, L(x,y), defined by x_l, y_l and $x_h + 1, y_h + 1$ and then modify the region at the boundary as follows:

$$\begin{aligned} L(x_l, y) &\leftarrow (1-\Delta x)L(x_l, y), \\ L(x, y_l) &\leftarrow (1-\Delta y)L(x, y_l), \\ L(x_h+1, y) &\leftarrow \Delta x L(x_h+1, y), \\ L(x, y_h+1) &\leftarrow \Delta y L(x, y_h+1) \end{aligned} \tag{4}$$

This procedure produces smooth apparent motion of the foveation region.

5. MOTION ESTIMATION AND COMPENSATION

Foveated multiresolution pyramids lend themselves readily to real-time estimation of local motion vectors from interframe comparisons. We use the hierarchical block matching method which is illustrated in Figure 5 to compute the motion vectors for each low-pass pyramid image. Specifically, we generate a low-pass multiresolution pyramid for each of the foveated subimages. These low-pass motion estimation pyramids do not have to be computed since they are contained in the previously computed foveation pyramid. The local motion for a block in the current frame (*j*) is estimated by finding the block of elements in the previous frame (*j-1*) that best matches the block in the current frame (the goodness of fit is taken to be the sum of the absolute differences of the element gray levels). The matching procedure begins at the highest (lowest resolution) level of the motion estimation pyramid and proceeds down level-by-level to the lowest (highest resolution) level.

Figure 5 illustrates the procedure for two successive levels of the motion estimation pyramid (for simplicity we illustrate a block size of 1 although we use a block size of 8 or 16). The shaded blocks in frame *j* show the blocks that are being matched; the shaded blocks in frame *j-1* show the blocks that are the closest match to the blocks in frame *j*. As indicated by the small solid squares in frame *j-1*, nine different matches are computed for each block. In this example, the nine blocks in

frame *j-1* are centered on the location corresponding to the block in frame *j*, and the upper left block provides the best match to the block in frame *j*.

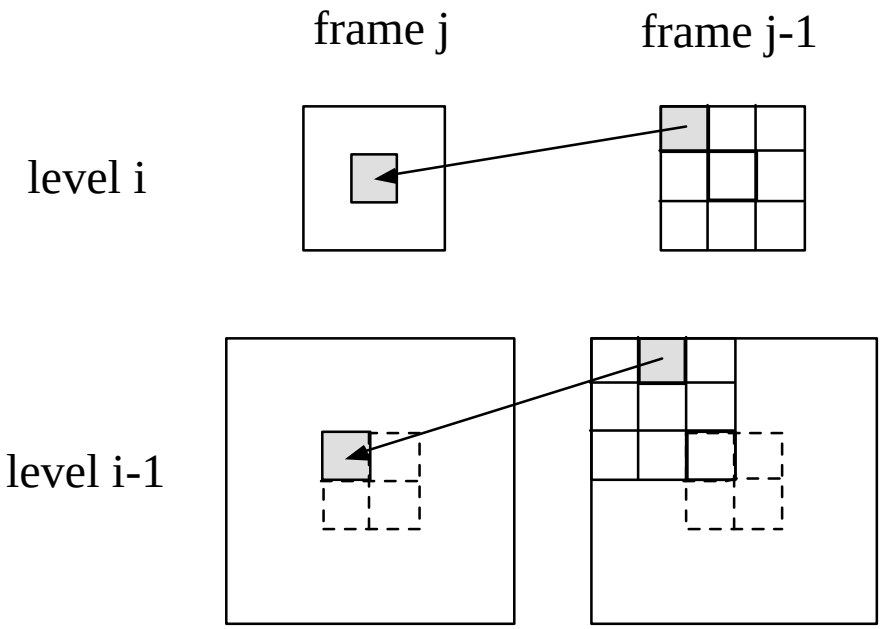

Figure 5. Hierarchical motion estimation in a multiresolution pyramid.

A useful aspect of the pyramid representation is that matches obtained at level i provide information that can constrain the search space for matches at level *i-1*. A block at level *i* corresponds to four blocks at level i-1. For example, in Figure 5, the shaded block in frame *j* at level i corresponds to the four blocks inside the dashed square in level *i-1*. Because the best match at level *i* was in the upper left direction, that direction is the most probable for a best-match (for any one of the four blocks at level *i-1*). Thus, the search space for the shaded block in frame j of level *i-1* is given by the 9 blocks indicated by solid lines in frame *j-1*. The best match for this block is 2 blocks up and 1 block over. This example, demonstrates how hierarchical matching is able to find matches over extended regions in the image, despite the ± 1 block search space at each level.

To describe the matching process more formally, let x_i, y_i be the coordinates (in units of elements) of a given block in level *i* of frame *j* which is to be matched against blocks in frame *j-1*, and let x_i^-, y_i^- be the coordinates of the block in level *i* of frame *j-1* that best matches the block at x_i, y_i. The values of x_i^-, y_i^- can be expressed in terms of the coordinates of the starting block for the search s_i^-, t_i^- and the offset, $\Delta x_i^-, \Delta y_i^-$, producing the best match:

$$\begin{aligned} x_i^- &= s_i^- + \Delta x_i^- & \Delta x_i^- &\in \{-1,0,1\} \\ y_i^- &= t_i^- + \Delta y_i^- & \Delta y_i^- &\in \{-1,0,1\} \end{aligned} \tag{5}$$

At the top level (*n*) of the motion estimation pyramid (where the motion estimation begins) the coordinates of the starting block are the same as those of the block being matched (as in level *i* of Figure 5):

$$\begin{aligned} s_n^- &= x_n \\ t_n^- &= y_n \end{aligned} \tag{6}$$

Below the top level of the pyramid the coordinates of the starting block are given by the following equations:

$$\begin{aligned} s^{-}_{i-1} &= x_{i-1} + 2x^{-}_{i} - \Delta x^{-}_{i} \qquad 2 \le i \le n \\ t^{-}_{i-1} &= y_{i-1} + 2y^{-}_{i} - \Delta y^{-}_{i} \qquad 2 \le i \le n \end{aligned} \tag{7}$$

where x_{i-1}, y_{i-1} are the coordinates of one of the four blocks which are daughters of the block with coordinates x_i, y_i.

The matching process can now be described precisely:

(1) For each block at level *n* in frame *j*, set the starting block for the search according to equation (6); then find the optimal values of the offset $\Delta x^{-}_{n}, \Delta y^{-}_{n}$; then substitute into equation (5) to obtain the coordinates x^{-}_{n}, y^{-}_{n} of the best matching block.

(2) For each block in the next lower level of frame *j*, use equation (7) to obtain the coordinates of the starting block; then find the optimal values of the offset; then substitute into equation (5) to obtain the coordinates of the best matching block.

(3) Repeat step (2) until all levels have be processed.

To obtain more precise motion estimates, we also provide the option of a second round of block matching using a ± 0.5 element step size. This second round of matching is carried out in the neighborhood of the block that gave the best match using the ± 1 element step size.

Motion compensation is performed using the motion estimates from the block matching procedure. Specifically, for each level of the pyramid, each block of pyramid coefficients in the current frame is subtracted (element by element) from the best matching block of the previous frame. To reduce the effects of motion estimation errors, the zero motion vector is also tested, and then selected if it provides better compensation.

6. BAND-PASS MULTIRESOLUTION CODING

Each of the motion-compensated foveation regions (see Figure 3) is coded using a multiresolution transformation (e.g., Laplacian pyramid, wavelet pyramid, discrete cosine transform). For the examples presented here, we used the Laplacian pyramid because of its good real-time performance. However, with moderate degrees of foveation, the foveation regions are small enough that useable real-time performance should be obtained with wavelet pyamids or with the discrete cosine transform.

7. THRESHOLDING AND QUANTIZATION

With foveated multiresolution pyramids, it is possible to obtain compression, with minimal loss of perceptual quality, by thresholding the transform coefficients on the basis of psychophysical data measured as a function of both the spatial frequency and the eccentricity. The thresholding function we use is essentially the same as equation (1):

$$d = T_{\max}\, CT_1 \exp\left(af \frac{e + e_2}{e_2} \right) \tag{8}$$

where d is the value of the threshold, $T_{\max}$ is the maximum absolute value of the transform coefficients, and the remaining constants and variables are the same as in equation (1). If a transform coefficient falls below the threshold then its value is set to zero:

$$\textit{if} \quad |T(x,y)| \le d \quad \textit{then} \quad T(x,y) = 0 \tag{9}$$

In applying equation (8), we allow the minimum contrast threshold parameter, CT_1, to be different from the value, CT_0, used to determine the sizes of the foveation regions.

To obtain further compression, we quantize the significant (non-zero) transform coefficients. In general, the higher resolution levels of the transformation can be quantized to a greater degree than the lower resolution levels, without

objectionable loss of image quality. Therefore, the number of quantization levels is set to a minimum value, $NQ_{\min}$, for the highest resolution coefficients, and is increased logarithmically, reaching a maximum value, $NQ_{\max}$, for the lowest resolution coeffients. With the Laplacian pyramid, we obtain better image quality with nonuniform quantization than with uniform quantization. Specifically, for each level of the pyramid, we bin the significant coefficients so that each bin contains approximately the same number of coefficients; the quantization value for each bin is taken to be the mean of the coefficient values in the bin.

8. ZERO-TREE AND ARITHMETIC CODING

Two forms of lossless coding are performed before data transmission. The first is a simple two-pass form of zero-tree coding. The first pass scans each level of the pyramid, in non-overlapping 2x2 blocks, beginning at the lowest level of the pyramid (the highest resolution). If all four elements in a block are zero, or are "zero root" symbols, then the parent element in the next higher level of the pyramid is checked; if the parent is also zero then the four elements are replaced with a "null" symbol indicating that they are not to be transmitted, and the parent is replaced with a zero root symbol. This process continues to the highest level of the pyramid. The second pass scans (in a fixed known order) all the elements starting at the highest level of the pyramid; all symbols except the null symbols are entered into the output data stream. Because, the scanning is in a fixed order, all of the zeros "under" a zero root symbol can be placed in their correct locations during reconstruction. The second and final lossless coding step is standard arithmetic coding.[19]

9. RECONSTRUCTION

Reconstruction proceeds by inversing the coding stages in the reverse order that they were applied.

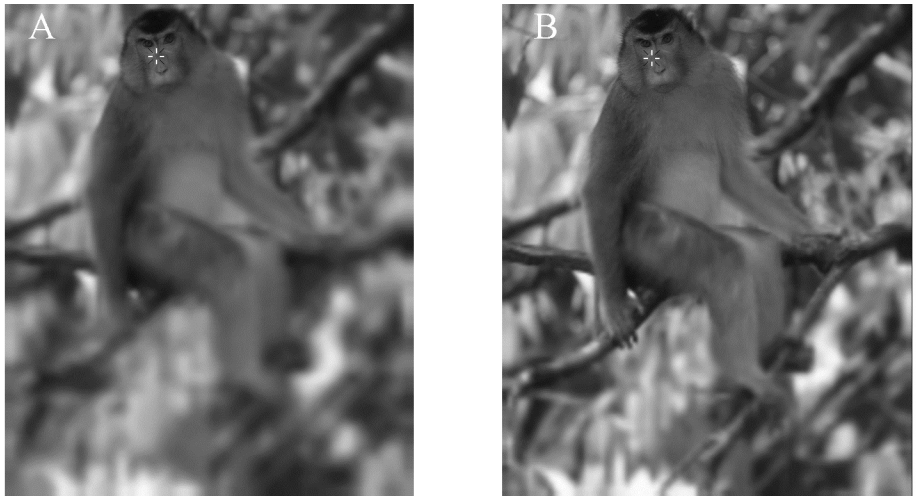

Figure 6. Foveated images (680 x 768) of a macaque monkey. The foveation point is indicated by the small plus/cross. A. Strong foveation resulting in a factor of 19 reduction in the number of pyramid elements (CT_0 = 0.25, $a = 0.1$, $e_2 = 2.3$, w = 10, r_0 = 2, deg/pixel = .046). B. Moderate foveation resulting in a factor of 5.5 reduction in the number of pyramid elements (CT_0 = 0.05, $a = 0.1$, $e_2 = 2.3$, w = 10, r_0 = 2, deg/pixel = .046).

10. SYSTEM PERFORMANCE

Our first real-time implementation demonstrates the following components of the full system: pointing device input, foveated low-pass pyramid coding, blending and interpolation, Laplacian pyramid coding, decoding, displaying. Figure 6 shows two example output images (680 x 768, 8-bit gray scale), obtained with a 3 x 3 kernel (for both the low-pass pyramid and the Laplacian pyramid),

$$K = \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix} \tag{10}$$

and a blending function width of 10 elements. The image on the left has been strongly foveated (factor of 19 reduction in the number of elements), and the one on the right has been moderately foveated (factor of 5.5 reduction). The small crosses indicate the foveation point. On a single 300 MHz Pentium Pro, 800 x 600 images are processed through all five components above at approximately 25 frames per second for the strong foveation, and approximately 20 frames per second for the moderate foveation. The frame rate is 50% higher for 640 x 480 images. Furthermore, these numbers underestimate performance for many applications (e.g., surveillance and teleoperation), because the coding, blending and interpolation would be done on a processor at the remote site, while the decoding and displaying would be done on another processor at the control site (see Figure 1). The first three components (pointing device input, foveated low-pass pyramid coding, blending and interpolation) could serve as a software preprocessor for a hardware MPEG coder.

Figure 7. "Claire." A. Uncompressed (6.3 b/pix) B. Compressed (0.043 b/pix, 36.7 dB) C. Foveated (0.020 b/pix)

Our second real-time implementation demonstrates all of the components in the full system. Although yet not fully optimized for real-time perfomance, we have obtained some preliminary results for three different video sequences: "Claire", "Mobile and Calender" and "Mall." The uncompressed entropy for of "Claire" is 6.3 bits/pixel. Figure 7A shows frame 22 of the uncompressed sequence. Figure 7B shows frame 22 of the compressed sequence (0.043 bits/pixel for I frame plus P frames, PSNR = 36.7 dB). Figure 7C shows frame 22 of the compressed and foveated sequence (0.020 bits/pixel). Foveation only adds a little to the total compression because the motion is primarily confined to the unfoveated region and because the image is small (360 x 288). The "Mobile and Calender" sequence better demonstrates of the value of foveation. Figure 8A shows frame 10 of the compressed sequence (0.08 bits/pixel for P frames, 28.1 dB). Figure 8B shows frame 10 of the compressed and foveated sequence with reduced thresholding and quantization so that the compression remains approximately the same (0.08 bits/pixel, 31.6 dB in the foveation region). This example demonstrates how foveation can be traded with quantization to dynamically allocate resolution to points of interest, without increasing bandwidth requirements (notice the greatly reduced number of artifacts in B near the foveation point on the ball). These 512 x 400 images have been clipped on the left to make the quantization artifacts in Figure 8A more visible in the reproduced images. The "Mall" sequence illustrates the value of foveation in applications such as surveillance or teleoperation. Figure 9 shows frame 10 of the compressed and foveated sequence (0.013 bits/pixel for the P frames, 29.3 dB in the foveated region). The unfoveated compressed sequence is not shown (0.092 bits/pixel, 29.3 dB). Foveation increased the compression by a factor of 7.

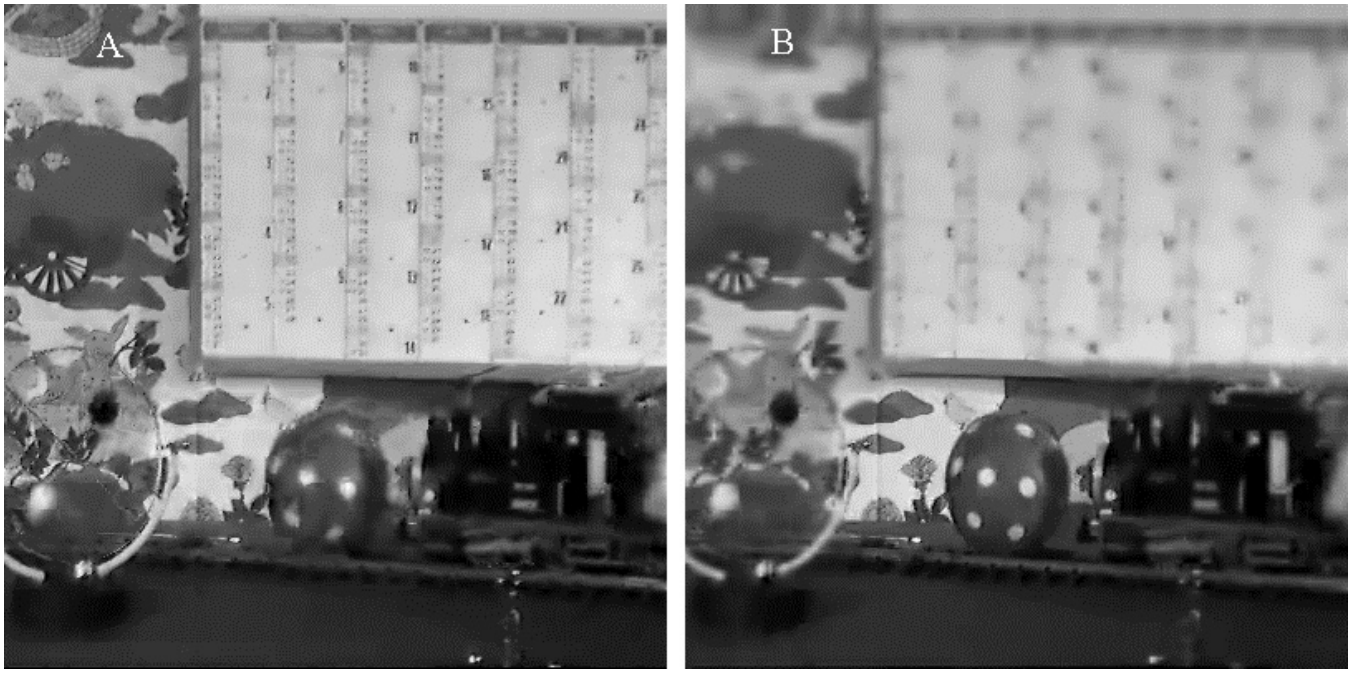

Figure 8. "Mobile and Calender." A. Compressed (0.08 b/pix, 28.1 dB) B. Foveated (0.08 b/pix, 31.6 dB in fovea)

We cannot yet report the speed performance of the full system because there are several inefficient steps with obvious remedies. Nonetheless, as it currently stands the encoding rate for foveated "Claire" (Figure 7C) is 19 frames/sec and the decoding rate is 94 frames/sec.

Figure 9. "Mall" Compressed--not shown (.092 b/pix, 29.3 dB), Foveated (0.013 b/pix),

11. APPLICATIONS

One of the more obvious applications of foveated imaging is in teleoperation, where there is often motion over extended image regions, and where bandwidth limitations are usually severe due to the need for wireless communication. For example, during teleoperation of a vehicle, high resolution information is required primarily in the heading direction for path planning and for avoidance of obstacles and hazards; but, relatively low resolution is sufficient in the periphery for judgments of

heading from optical flow, and for detection of incoming objects and/or vehicles. With the wide field of view usually desired for teleoperation, foveated imaging can have a truly dramatic affect on bandwidth transmission requirements.[8]

Other potential applications for foveated imaging are in surveillance, telemedicine and teleconferencing. In these applications, there are often localized regions of the video images that the user wants to inspect. Foveated imaging allows the user to dynamically allocate high spatial resolution to the regions of interest (see Figure 8).

Obviously, to make optimal use of foveated imaging, the resolution of the video information to be transmitted (e.g., the camera resolution) must be selected or created to exceed the bandwidth limitations of the communication channel when using the non-foveated codec at the desired image frame rate. Foveated imaging will then allow the user to access the high resolution video information that could not be accessed (at the desired frame rate) without foveated imaging.

The most elegant and seamless implementation of foveated imaging is with an eyetracker, which keeps the high-resolution region of the displayed image centered on the observer's line of sight. For example, Owl Displays Inc. (Austin, TX) is currently integrating the FMP imaging system into an elegant high resolution helmet mounted display system with a built in eyetracker. For moderate degrees of foveation in this system, the user cannot detect that the images are foveated.

On the other hand, foveation is often valuable even using simpler, less expensive and more robust pointing devices, such as a mouse or touch pad. For example, in teleoperation, directing the foveation point toward the heading direction will provide fine detail where it is most needed, but at the same time, provide a wide field of view. Although the foveation will sometimes be visible, the user will perform better than without foveation, given a fixed communication bandwidth. Similarly, in surveillance, telemedicine and teleconferencing it is often sufficient, for getting a particular task done, to direct the foveation point to regions of interest with a simple pointing device.

One way to think about value of foveated imaging is to consider being confronted with the choice of two nearly equivalent video communications systems. Both transmit information at the same bandwidth with equal resolution in a non-foveated mode, but one system gives the user the option of switching to a foveated mode where spatial resolution can be dynamically allocated to regions of interest without affecting frame rate. A little consideration leads one to the conclusion that there are many situations where this feature would very valuable in allowing the user to complete a task that would be difficult or impossible otherwise. This feature would be valuable even though the image degradation outside the foveation region might be visible, as it would be with strong foveation or with pointing devices other than an eyetracker.

12. CONCLUSION

This paper describes a foveated multiresolution pyramid (FMP) coder/decoder for low bandwidth video communications. The codec, although not yet fully honed, provides smooth foveation and good compression at useful frame rates on a general purpose computer (a Pentium running under Windows95/NT). The novel contributions include: (1) full integration of foveation into multiresolution pyramids, (2) the development of efficient pyramid, foveation, and motion estimation algorithms which make possible real-time operation on conventional computer hardware, (3) development of efficient methods for eliminating foveation artifacts, (4) the use of psychophysical contrast sensitivity data as function of spatial frequency and eccentricity to determine foveation regions, and to determine the thresholding and quantization. Our experience suggests that foveated imaging would be a useful feature in many video communications applications.

13. ACKNOWLEDGMENTS

This research was supported by AFOSR STTR grant F49620-94-C-0090 to WSG and to OWL Displays Inc., Austin TX, and by AFOSR URI grant F49620-93-1-0307. Larry Stern and Carl Creeger provided valuable technical assistance.

14. REFERENCES

1. C. M. Howard, "Display Characteristics of Example Light-Valve Projectors," Operations Training Division, Air Force Human Resources Laboratory, Williams AFB, AZ AFHRL-TP-88-44, 1989.
2. R. D. Juday and T. E. Fisher, "Geometric transformations for video compression and human teleoperator display," *SPIE Proceedings: Optical Pattern Recognition*, vol. 1053, pp. 116-123, 1989.

3. C. F. R. Weiman, "Video Compression Via Log Polar Mapping," *SPIE Proceedings : Real Time Image Processing II*, vol. 1295, pp. 266-277, 1990.
4. B. B. Benderson, R. S. Wallace, and E. L. Schwartz, "A miniature pan-tilt actuator: the spherical pointing motor," *IEEE Transactions Robotics and Automation*, vol. 10, pp. 298-308, 1994.
5. H. D. Warner, G. L. Serfoss, and D. C. Hubbard, "Effects of Area-of-Interest Display Characteristics on Visual Search Performance and Head Movements in Simulated Low-Level Flight," Armstrong Laboratory, Human Resources Directorate, Aircrew Training Division, Williams AFB, AZ. AL-TR-1993-0023, 1993.
6. P. L. Silsbee, A. C. Bovik, and D. Chen, "Visual pattern image sequence coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 291-301, 1993.
7. B. S. Barnett and A. C. Bovik, "Motion compensated visual pattern image sequence coding for full motion multisession videoconferencing on multimedia workstation," *Journal of Electronic Imaging*, vol. 5, pp. 129-143, 1996.
8. P. T. Kortum and W. S. Geisler, "Implementation of a foveated image-coding system for bandwidth reduction of video images," *SPIE Proceedings: Human Vision and Electronic Imaging*, vol. 2657, pp. 350-360, 1996.
9. E. Chang and C. K. Yap, "A wavelet approach to foveating images," *ACM Symposium on Computational Geometry*, vol. 13, pp. 397-399, 1997.
10. P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. COM-31, pp. 532-540, 1983.
11. A. M. Tekalp, *Digital Video Processing*. Upper Saddle River: Prentice Hall, 1995.
12. E. H. Adelson, E. Simoncelli, and R. Hingorani, "Othogonal pyramid transforms for image coding," *SPIE Proceedings: Visual Communications and Image Processing II*, vol. 845, pp. 50-58, 1987.
13. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavlet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205-220, 1992.
14. S. A. Martucci, I. Sodogar, T. Chiang, and Y. Zhang, "A zerotree wavelet video coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 109-118, 1997.
15. J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445-3462, 1993.
16. J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field.," *Vision Research*, vol. 21, pp. 409-418, 1981.
17. T. L. Arnow and W. S. Geisler, "Visual detection following retinal damage: Predictions of an inhomgeneous retino-cortical model," *SPIE Proceedings: Human Vision and Electronic Imaging*, vol. **2674**, pp. 119-130, 1996.
18. M. S. Banks, A. B. Sekuler, and S. J. Anderson, "Peripheral spatial vision: limits imposed by optics, photoreceptors, and receptor pooling," *Journal of the Optical Society of America*, vol. 8, pp. 1775-1787, 1991.
19. I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," *Communications of the ACM*, vol. 30, pp. 520-540, 1987.

Further author information -
W.S.G. (correspondence): Email: geisler@psy.utexas.edu; Telephone: 512-471-5380; Fax: 512-471-7356
J.S.P.: Email: jsp@mail.utexas.edu; Telephone: 512-471-3054; Fax: 512-471-7356