Videoconferencing using Spatially Varying Sensing with Multiple and Moving Foveae

Anup Basu, Kevin James Wiebe

Videoconferencing using Spatially Varying Sensing with Multiple and Moving Foveae *

Anup Basu and Kevin James Wiebe

Department of Computing Science, University of Alberta
Edmonton, Alberta, Canada T6G 2H1

Abstract

A new method for videoconferencing using the concept of spatially varying sensing is introduced. Various techniques are discussed for combining information obtained from multiple points-of-interest (foveae) in an image. A fovea can be used to follow a moving object of interest as well. A fast videoconferencing prototype for desktop computers is also described.

1 Introduction

The roots of data compression lie in entropy encoding techniques. These schemes range from minimum redundance codes (such as Huffman encoding) to dictionary coding methods (LZ77, LZ78 and LZW). The ability of any entropy encoding technique to compress data depends on the presence of patterns (repeating bytes or skewed frequency distributions) within the input data stream. Unfortunately, digitized computer images often lack such patterns, which limits the effectiveness of entropy encoding techniques in image compression. A number of different methods have been applied to these difficulties with varying degrees of success. They include vector quantizing, wavelets and fractal based compression.

Much attention has been given to compression schemes based on the Discrete Cosine Transform (DCT) [7]. The DCT is able to convert images from the spatial to the frequency domain. Once converted, high frequencies (ones which the human eye cannot detect) can be eliminated. An advantage is that compression and quality can be traded against each other. If a small image is required, additional high frequencies can be eliminated at a corresponding decrease in quality. The DCT is the basis of a lossy compression scheme designed for still image compression, and developed by the Joint Photographic Expert Group (JPEG). JPEG allows quality to be exchanged for compression, usually by using some quality value.

One of the prime applications for motion pictures has been teleconferencing, where individuals can communicate with each other visually over networks as easily as using a telephone [3, 4]. Early digital **videoconferencing** systems were often limited in their capabilities [1]. However, similar systems, often allowing the integration of voice, video, and text, have recently begun to make their appearance on desk top workstations and PCs within 'multimedia' systems.

In many images, there often exists one or more areas which are of greater interest than the remainder of the picture. In such an area (fovea) more detail is required. The outer regions (periphery) are often of secondary importance, and thus less detail is required. This decrease in detail within the periphery may be achieved by varying the spatial resolution. The concept of variable resolution (VR) has been applied to many different areas, but its application to image compression is relatively new [2]. There are a number of distinct advantages of variable resolution compression:

- By controlling the sampling, guaranteed compression ratios can be obtained
- The VR transform can be achieved using a look-up table, thus VR compression can be performed quickly
- Images compressed using the VR transform can be further compressed using other methods

Clearly, VR cannot be used to compress all data, since there are many cases where the area of primary interest cannot be determined in advance (such as medical images), and the distortions caused may be unacceptable. However, there are some applications (such as videoconferencing) where this technique can be used effectively and gives high compression ratios for relatively low computational cost.

2 The VR Transform

Studies have shown that animate vision has much higher resolution in the center of the visual field than in the periphery. Swartz developed several mathematical models of vision systems based on this variable resolution concept [6]. More recently, cameras capable of capturing images using log-polar coordinates directly have been developed [5], bypassing the need for mathematical transformations altogether in some

*This work is supported in part by the Canadian Natural Sciences and Engineering Research Council (NSERC). Kevin Wiebe was supported by an NSERC Ph.D. scholarship and the Ralph Steinhauer fellowship. A patent, based on this work, is pending.

cases. Results presented here use a simplified mathematical model (the "Fish-eye" transform) to implement the VR transform.

2.1 The Basic Variable Resolution Model

The VR transform we use has two parameters which affect the resulting image: the expected savings (compression), and alpha (α) which controls the distortion at the edges of the image with respect to the fovea. A high α value gives a sharply defined fovea with a poorly defined periphery; a small α value makes the fovea and periphery closer in resolution.

Under the VR transform, a pixel with polar coordinates (r, θ) is mapped to (vr, θ) where

$$vr = \ln(r * \alpha + 1) * sf \tag{1}$$

In other words, the pixel is moved from r to vr units away from the fovea. This transformation is easily reversed, allowing r to be defined in terms of vr.

$$r = \frac{\exp(vr/sf) - 1}{\alpha} \tag{2}$$

The value sf is a scaling factor used to control the overall compression ratio. It is calculated so that points at the maximum distance from the fovea in the original image are at the maximum possible distance in the VR image:

$$sf = \frac{vr_{max}}{\ln(r_{max} * \alpha + 1)} \tag{3}$$

It must be noted that pixels are discrete elements. Thus, when reducing the size of the image via the VR transform, one VR pixel may represent several pixels in the original image. Several interpolation methods have been tested for use with decompression. "Bilinear Interpolation" has proved to be best suited for our purposes and is used throughout our research, except where noted.

2.2 Extending the Model

There is one difficulty with the basic variable resolution model and its application to image compression. Under the VR transform, images are not rectangular. If storage in a rectangular field is necessary, areas of the image must be clipped off, or the image will have unused pixels in the corners. The problem is magnified with high α values and when the fovea is not located in the center of an image.

One approach to solving this problem uses multiple scaling factors, each scaling factor dependent on the angle θ in polar coordinates ("Modified Variable Resolution"). Each angle θ has its own maximum distances to the image edge, and thus its own scaling factor. The transformed image can now be mapped to a rectangle with full space utilization, regardless of the location of the fovea. This method does a reasonable job of maintaining the isotropic properties of the original formulae, but is relatively complex.

Another approach to dealing with non-rectangular compressed images is to greatly simplify the formulae by isolating the vertical and horizontal components

Figure 1: Original image.

Modified  Cartesian

Figure 2: Compressed image; fovea near center. Compression is 90%.

("Cartesian Variable Resolution (CVR)"). Figures 1-3 demonstrate the alternative approaches. Computational complexity is significantly decreased for the CVR method — however, isotropic accuracy is reduced as well. For a given image with the fovea located at (x_0, y_0), for every pixel (x, y) in the original image, we define the distance from (x, y) in x and y directions as dx and dy respectively, from the following equations:

$$dx = x - x_0 \tag{4}$$

$$dy = y - y_0 \tag{5}$$

So, (x, y) is mapped to point (x_1, y_1) where:

$$x_1 = x_0 + \ln(dx * \alpha + 1) * sf_x \tag{6}$$

$$y_1 = y_0 + \ln(dy * \alpha + 1) * sf_y \tag{7}$$

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

$$dvx = \ln(dx * \alpha + 1) * sf_x \tag{8}$$

$$dvy = \ln(dx * \alpha + 1) * sf_y \tag{9}$$

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

$$dx = \frac{\exp(dvx/sf_x) - 1}{\alpha} \tag{10}$$

Figure 3: Uncompressed image (CVR Method)

$$dy = \frac{\exp(dvy/sf_y) - 1}{\alpha} \tag{11}$$

As in the basic VR model, the values sf_x and sf_y are scaling factors used to control the overall compression ratio. The scaling factor is calculated so that points at the maximum x or y distance from the fovea in the original image are at the given maximum possible x and y distance in the compressed image:

$$sf_x = \frac{dvx_{max}}{\ln(dx_{max} * \alpha + 1)} \tag{12}$$

$$sf_x = \frac{dvy_{max}}{\ln(dy_{max} * \alpha + 1)} \tag{13}$$

As with the Modified VR method, different scaling factors can be used. The CVR method can vary the scaling factors, both horizontally and vertically, depending on the position of the fovea. The lookup tables constructed using this method will thus be dependent on the position of fovea, but the resulting compressed images will be of constant size. If, however, the scaling factors used vertically and horizontally are computed independently from the position of the fovea, the compressed images will vary in size, depending on the fovea location. The advantage to this method is that the tables need not be recomputed each time the fovea changes location. This is important when implementing *moving foveae.*

2.3 Moving Fovea

When used to compress several images in a continuous sequence, the position of the foveae need not be the same for each image. If they do differ, when the images are viewed in sequence the fovea will appear to move, and may be called a *moving fovea.* One obvious place such a fovea would be useful is in a videophone application. The location of a fovea could follow a person's mouth, keeping that part of the image most clear. As the individual moves through the scene, so too would the fovea. The system must respond to the changes in the location of the fovea in real-time. Figure 4 depicts the effect of movement of the fovea. For the fovea location at F2, the results of the transform of only some of the points lying in the first quadrant can be obtained from the look-up table (LUT) when the fovea is at the center (F1). In this case we have a problem with the other points in that quadrant. Thus the best structure for LUT is one that contains the results of the transform for all the points lying in the first quadrant, when the fovea is at the bottom left corner of the image. In this case the number of entries in the LUT is equal to the size of the image. For any location of the fovea, the relative coordinates of the pixels in the image are obtained. Then the transform of the points $(x_{max}, 0)$ and $(0, y_{max})$ is computed to obtain the dimension of the VR image bounding box. Using this information the LUT is searched and the required information for performing the VR transform is obtained. Note that the VR transform for other quadrants can be related to the LUT for the first quadrant, thus it is enough to store information only for the first quadrant.

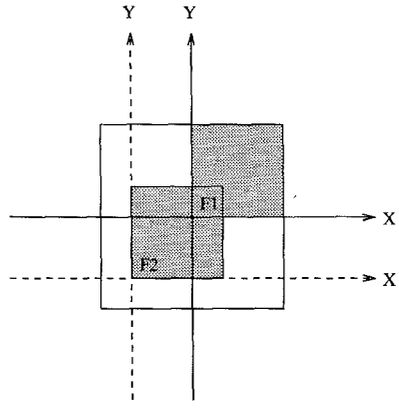

Figure 4: Movement of the fovea.

2.4 Multiple Foveae

There may be circumstances where there is more than one area of interest to the observer. This situation requires multiple foveae, where two or more regions are displayed with higher resolution than the remainder of the image.

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

There is no unique way of defining multiple foveae. However, one can clearly define the relationship depending on the desired properties. Two distinct approaches are *cooperative* and *competitive* foveae.

2.4.1 Cooperative Fovea

Figure 5: Sample image before compression.

A method which works well for two fovea situations calculates the location of a point in the transformed image with respect to each fovea separately. The true location is then found by weighting the two estimated points according to the distance of the original point from the fovea. A higher weight is given to the location calculated using the closer fovea:

$$l_{actual} = l_1 * \frac{d_2}{d_1 + d_2} + l_2 * \frac{d_1}{d_1 + d_2} \qquad (14)$$

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

This scheme is not limited to two foveae. A greater number of foveae simply involves computing a weighted average of a larger number of contributing points. This method is termed *cooperative* in that all foveae work together to calculate the position of a point in the transformed image.

A unique property of cooperative foveae is the existence of "ghost foveae" between them. If only two foveae exist, then the area of highest quality in the scene will not only be at these foveae, but also along the line connecting the foveae. Multiple fovea systems will also contain areas of high quality between foveae, whether spots or curves. If two foveae lie on exactly the same location, the transformed image will not always be the same as if only one fovea had existed there. Each fovea, no matter where it lies, contributes to the final positioning of each point in the transformation.

In general, though, the areas outside of all foveae remain a constant quality, based only on the proximity to the foveae. Quality is independent of relative fovea position in that as long as the weighted average of all the foveae remains the same, the positions of the foveae do not make a difference to periphery quality. In a scene where multiple moving cooperative foveae exist, although "ghost foveae" may appear between them, the periphery will remain a constant quality throughout.

Figure 6: Cooperative Foveae placed on the outer faces; 80% compression.

Figure 7: Competitive Foveae placed on the outer faces; 80% compression.

2.4.2 Competitive Foveae

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

$$l_{actual} = \{l_i : \forall \ foveae \ \ j, j \neq i, d_i < d_j\} \qquad (15)$$

where l_i represents the coordinates of a point calculated using fovea i, and d_i represents the distance to fovea i. Unlike cooperative foveae, when scaling factors are computed to guarantee a rectangular compressed image, the maximum distances used in the formula are not the edges of the image. Instead, a simple voronoi tesselation is generated around the foveae and the maximum edge of each voronoi area is used.

The "ghost foveae" do not appear between competitive foveae, as the image is essentially broken up into separate regions, each with only one fovea contributing to the compression. There is no noticeable transition between regions, as the boundary is equidistant from the contributing foveae.

Here, the quality of an image not only depends on the proximity to a fovea, but also on the relative position of all the foveae. If the total compression ratio is set to a given constant, then the overall quality of the image will increase as two foveae move closer. This effect is more pronounced in the periphery of the scene. If two foveae are centered on the same location, then they act exactly as a single fovea would in that position. Figures 5–7 illustrate the concepts discussed above.

3 A videophone prototype

The videophone component is able to provide transmission of grey scale images from an image **server** to a **display** or **viewer** process. The server process is responsible for capturing, compressing and transmitting the image. The display process accepts images, uncompresses and displays them on a screen. Figure 8 shows a sample videophone organization.

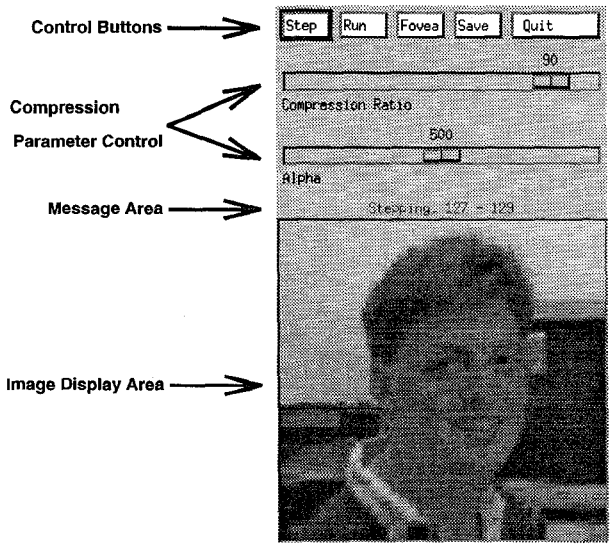

Figure 8: Sample videophone window.

In addition to variable resolution, an interframe difference encoding routine provides further compression. The difference between pixels in successive frames is found (most pixels will not change if the image is static) and only changed pixels are transmitted.

Without using any special purpose hardware or DSP chipsets, frame rates in the order of 10 to 12 frames per second have been achieved on the SGI Indy stations across an Ethernet. In comparison, JPEG can produce only around 1 frame per second. With the compression values obtained (up to 98 % with interframe encoding) the network can easily handle much higher frame rates.

The teleconferencing component operates on the same principle as the videophone, except that it does not have one fixed fovea at the center. Instead, multiple foveae can be placed at the user's discretion. Work is currently under way to implement moving foveae which automatically track any person or other moving object in a scene, as chosen by the observer.

4 Conclusion

Variable resolution has some major advantages as a teleconferencing technique. The rate at which it can compress images, especially on machines with limited processing speed, and the high quality present in the foveal region, make it ideal for the multimedia market. The experience with the multimedia prototype supports this belief. Most importantly, frame rates can be maintained without the need for additional hardware. Using this approach an organization can implement a teleconferencing system on a local or wide area network with very little hardware cost.

References

[1] D. Anastassiou, M. Brown, H. Jones, J. Mitchell, W. Pennebaker, and K. Pennington. Series/1 based videoconferencing system. *IBM Systems Journal*, 22(1/2):97–110, 1983.

[2] A. Basu, A. Sullivan, and K. J. Wiebe. Variable resolution teleconferencing. In *IEEE SMC Conference Proceedings*, pages 170–175. IEEE, October 1993.

[3] Alan Borning and Michael Travers. Two approaches to casual interaction over computer and video networks. In S. Robertson, G. Olson, and J. Olson, editors, *CHI '91 Conference Proceedings*, pages 13–19. ACM, 1991.

[4] Mon-Song Chen, Zon-Yin Shae, Dilip Kandlur, Tsipora Barzilai, and Harrick Vin. A multimedia desktop collaboration system. In *Globecom 92*, pages 739–746. IEEE, 1992.

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

[6] Eric Schwartz. Computational anatomy and functional architecture of striate cortex: a spatial mapping approach to perceptual coding. *Vision Research*, 30:645–669, 1980.

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