Society for Information Display, 30, 420-423, 1999

Variable-Resolution Displays for Visual Communication and Simulation (SID 99)

Wilson S. Geisler and Jeffrey S. Perry
Center for Vision and Image Sciences, University of Texas at Austin

Abstract

The spatial resolution of the human visual system decreases as a function of angular distance from the direction of gaze. This fact can be exploited in various applications to increase image compression, to increase image processing speed, and to decrease access time for image data. This paper describes a multiresolution pyramid method for creating variable resolution displays in real time using general purpose computers. The location of the high resolution region(s) can be dynamically controlled by the user with a pointing device (e.g., a mouse or an eye tracker) or by an algorithm. Our method has a number of advantages: high computational speed and efficiency, smooth artifact-free variable resolution, and compatibility with other image processing software/hardware. Applications to video communications (MPEG) and graphic simulation are described.

Introduction

The spatial resolution of the human visual system declines precipitously away from the point of gaze, and thus it is possible to decrease gradually the display resolution in the visual periphery with little effect on perceptual quality or visual performance. Thus, substantial savings in both bandwidth requirements and processing time can be obtained by matching the spatial resolution of displayed information to the fall off in spatial resolution of the human visual system. Variable resolution (foveated) displays are most effective when the direction of gaze is tracked so that the highest resolution region of the display can be kept aligned with the highest resolution region of the eye (the fovea). However, variable resolution displays are also useful in a number of applications where eye tracking is not practical.[1]

Previous real-time foveated imaging systems suffer from two limitations: the appearance of blocking artifacts and motion aliasing in the low resolution regions, and/or incompatibility with other imaging processing software and hardware.[2-6] The purpose of this paper is to describe and demonstrate a software library which can be used to create smooth artifact-free variable resolution images in real time on inexpensive platforms (e.g., PCs running the Windows95/98/NT OS). The method of encoding and decoding is designed to optimize compatibility with other image processing software or hardware.

Algorithm

The method of creating variable resolution images is illustrated in Figure 1. First, the input image is encoded as a "low-pass pyramid" with 2-6 levels, depending on the size of the image and degree of foveation. Specifically, the input image is low-pass filtered and down sampled by a factor of 2 in each direction to obtain the second level of the pyramid. The second level of the pyramid is then low-pass filtered and down sampled by a factor of 2 in each direction to obtain the third level of the pyramid. This process is repeated until all the desired levels are computed.

From each level of the low-pass pyramid an image region is selected. From level 1 a region is selected for the foveation (high-resolution) region; from level 2 a region is selected for the first ring around the foveation region; from level 3 a region is selected for the second ring around foveation region; and so on, for each level. This collection of selected image regions is called a *foveated pyramid*, and it defines the variable resolution image. The foveated pyramid is the output of the encoder.

A foveated pyramid is a collection of conventional images which can be processed subsequently one at a time. Or, instead, the images may all be packed into a single image and processed as a whole. The savings in bandwidth requirements and in subsequent processing time occur because the total number of pixel elements in a foveated pyramid is generally much less than the number in the original image.

Because a foveated pyramid is a collection of conventional images, it can serve as input to a wide variety of image processing software or hardware. For example, a foveated image sequence can easily be encoded, transmitted and decoded using MPEG software or hardware.

Following the other image processing, the foveated pyramid is decoded to obtain a smooth artifact free variable resolution image. Starting with the last level of the pyramid (the lowest resolution), the image is up-sampled, interpolated, and blended with the next highest resolution level. The resulting image is then up-sampled, interpolated, and blended with the next level, and so on, until the highest resolution image is finally blended to complete the foveated image.

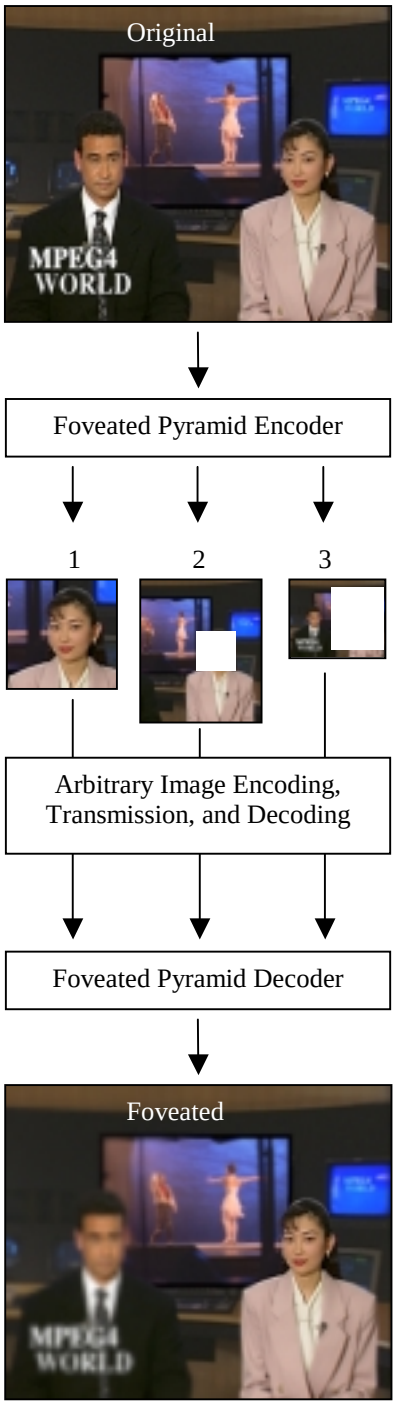

Figure 1. Foveated pyramid encode/ decode of a frame from the "News" video image sequence.

Software

A foveated imaging software package is available free of charge and may be downloaded from the CVIS foveated imaging website at http://fi.cvis.psy.utexas.edu. This package contains linkable libraries, documentation, example programs, and C/C++ source code examples that can assist developers and researchers in creating foveated imaging systems.

The foveated pyramid encoding and decoding process, along with other useful video and imaging operations, are implemented in the software package as C/C++ library functions. Key features of the library software include:

- Real-time foveated pyramid encoding and decoding with selectable filter kernels and arbitrary degrees of edge smoothing
- Real-time support for 8, 24, and 32 bit per pixel formats as well as support for custom pixel formats
- YUV/RGB conversion and YUV common interchange file format support (CIF, QCIF, SIF, ... etc.)
- Persistent image storage

The software is optimized to achieve high framerates while running on commonly available, inexpensive hardware platforms. Table 1 shows performance rates in frames per second of a foveated imaging application built with the library software.

Table 1 Software encode/decode performance of 640 x 480 video

Intel® Pentium® II MHz	8 bpp fps	24 bpp w/HW YUV conv fps	24 bpp w/ SW YUV conv fps
400	43	27	15
200	18	8	5.5

A fast personal computer can encode and decode 8 bit video at over 40 frames per second with a moderate amount of foveation. The same computer, equipped with a hardware YUV to RGB converter found on many video controllers, can encode and decode 24 bit video at almost 30 frames per second.

Applications

There are a number of potential applications for real-time foveated imaging. Two of the more important ones are in video communication and in 3D graphic simulation.

1. Video Communication

To examine the potential value of our foveated imaging algorithm and software for video communication we have used the foveated imaging encoder and decoder as a pre-processor and post-processor (respectively) for the MPEG H.263 codec. Several different video sequences have been tested with similar results. Here we report the results for the "News" sequence of which one frame is shown in Figure 1. The degree of foveation and smoothing were the same as illustrated in Figure 1. With this degree of foveation the number of pixels in the foveated pyramid was approximated 1/3 of that in the 352 x 288 image. The images in each foveated pyramid were copied into a single image (smaller than 352 x 288) prior to MPEG coding and decoding.

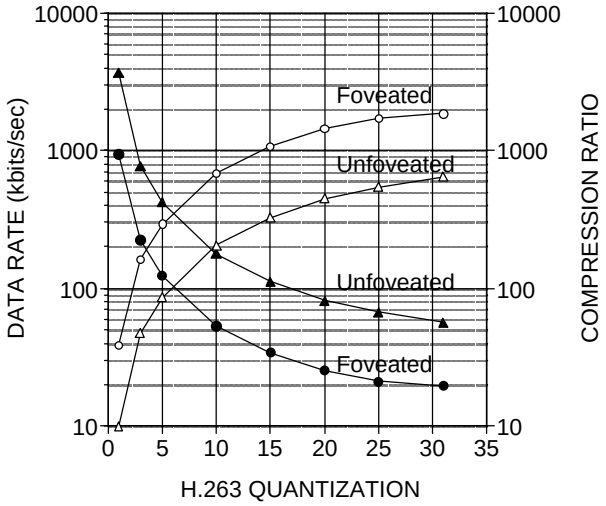

Figure 2. Data rates and compression ratios at 30 frame/sec for the "News" video image sequence.

The results of the test are shown in Figure 2. The vertical axis on the left (solid symbols) plots the data rate as a function of the level of H.263 quantization for a frame rate of 30 fps. The vertical axis on the right (open symbols) plots the compression ratio. As can be seen, the effect of foveation is roughly multiplicative—foveation reduces the data rate (or equivalently, increases the compression) by a factor of approximately 3, independent of the level of quantization. Thus, with foveation it is possible to preserve a wide field of view and high resolution in the region of interest, while reducing the bandwidth requirements by a factor of 3.

In addition to decreasing the bandwidth requirements, the foveation was also found to substantially increase the speed of the (software) MPEG encoding and decoding. This occurred because the images in the foveated pyramid were much smaller than the original image.

Rather than decreasing the data transmission rate, it is also possible to hold the transmission rate fixed and use foveation to increase the frame rate by a factor of 3. This is probably the more useful way to use foveation. A real time demonstration of this mode of operation is available at the CVIS foveated imaging website.

The degree of bandwidth savings for a given degree of foveation increases as the image size increases. This is illustrated in Figure 3, where the direction of gaze is located on the upper airplane in the 640 x 480 image region. The number of pixels in the foveated pyramid for this region is 1/6 of that in the original 640 x 480 region. If the display size is increased to 1024 x 768 or 1280 x 1024, the number of pixels becomes 1/11 or 1/16 of that in the original image. Thus, foveation is particular valuable in large display applications.

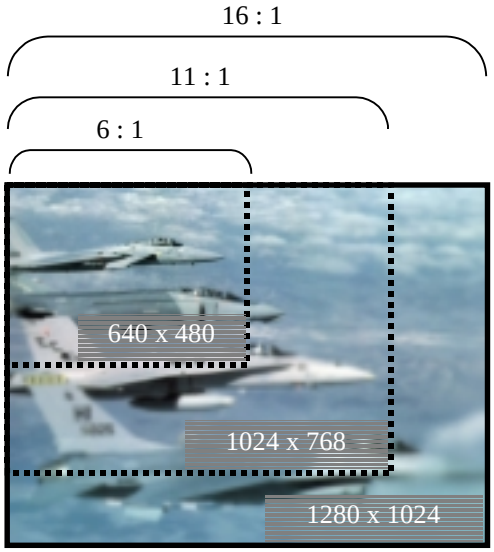

Figure 3. Compression ratios for various fields of view.

2. 2D and 3D Simulation

Two-dimensional and three-dimensional visual simulators can both benefit from foveated imaging techniques. For example, in 2D simulators, both images and geometric objects may be stored in a multiresolution fashion—the images as low-pass pyramids, and the geometric objects as collections of varying numbers of geometric primitives (e.g., lines or vertices). When objects are added to the scene, the detailed or high-resolution representation of an object is used when it is near the point of gaze. As the object

becomes further from the point of gaze, the less detailed or lower resolution representations are used. These multi-resolution objects are rendered to a foveated pyramid where object detail level is selected according to the level within the pyramid. The pyramid then serves as input to the foveated decoder to create foveated images (see Figure 1). Performance gain is achieved because access time (from, say, disk or CD-ROM) is much less for objects outside of the foveal region, so total access time is reduced. Furthermore, less processing time is required to translate, rotate, and scale lower resolution geometric objects, and less processing time is required to rotate, scale, and resample lower resolution images. The factor of improvement in access time and processing time can be as large as the compression ratio due to the foveation (e.g., 3:1 for the level of foveation and image size in Figure 1).

Three-dimensional visual simulators already commonly employ the use of multi-resolution pyramids, especially in real-time 3D rendering applications such as computer games. 3D objects are stored as lists of polygons with their associated color information and texture maps. In order to achieve better performance, some simulators represent the same object with varying levels of detail. Greater detail is represented with greater numbers of polygons and with higher resolution textures. The level of detail used to render the object is determined according to its distance from the viewer: objects that are further from the viewer are smaller due to perspective and, therefore, require less detail. By representing objects in this way, overall polygon and texture processing requirements are reduced without sacrificing the detail of objects that appear close up.

This same technique can also be used to select the detail level according to an object's distance from the point of gaze. Objects that are further from the point of gaze will have lower spatial resolutions, and therefore require less detail. Performance gain is seen because polygon and texture-map processing requirements are reduced for objects outside of the foveal region, even if the objects appear close to the viewer.

Even without the use of multi-detailed object representation, a 3D scene could be rendered to a foveated pyramid and used as input to the foveated decoder similar to the 2D situation discussed above. In this case, a performance gain will be seen when shading and texturing operations occur and when special effects like fog are applied. This is true because the total number of pixels being rendered in the foveated pyramid is generally much less than those that would be rendered in the unfoveated image.

Finally, a performance gain may also be seen in 3D visual simulators during area boundary anti-aliasing. Areas that are outside of the foveal region are implicitly anti-aliased by the foveated pyramid and therefore do not require additional anti-aliasing.

Acknowledgements

Supported by AFOSR grant F49620-94-C-0090 and NIH grant EY02688.

References

1. Geisler, W.S. and Perry, J.S. (1998) A real-time foveated multi-resolution system for low-bandwidth video communication In: B. Rogowitz and T. Pappas (Eds.), *Human Vision and Electronic Imaging, SPIE Proceedings* 3299, 294-305.
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. 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.
4. 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.
5. 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.
6. 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.