Computing AHE Using Pixel-Planes

John D. Austin

October 25, 1984

1. Introduction

This report describes an algorithm that implements ~~a form of~~ Adaptive Histogram Equalization on the Pixel-Planes raster graphics engine. Two images computed using this algorithm have been found to be nearly identical to images computed with standard AHE.

There are two differences between this algorithm and standard AHE. First, the grey level mapping is determined directly by the rank of each pixel's intensity within its contextual region instead of a calculation from the histogram. Second, the contextual region around each pixel is used for the computation instead of an interpolation between a smaller number of contextual regions.

2. Description of Algorithm

The Adaptive Histogram Equalization processing technique defines a contextual region around each pixel. This is the local region that affects the grey level mapping for that pixel. Looking at the problem in reverse, there is also a second region of identical size around each pixel which covers all pixels whose contextual region is affected by the given pixel. This region will be called the unnamed region and is used in this implementation of AHE.

Each pixel location in the frame buffer will store the original image intensity (approximately 10-12 bits) and the rank of the pixel's intensity relative to the pixels in its contextual region (8-20 bits depending on image and region sizes).

The image data is first loaded into the frame buffer. Because there is no polygonal information in the image, each pixel is defined to be a polygon the same size as one pixel.

For each pixel in the image, the pixel intensity and the coefficients necessary to mask the unnamed region are transmitted to the engine. Pixels which lie inside the region compare their intensity to the intensity of the given pixel. If the intensity is greater than the given pixel, the rank counter stored in the pixel's memory is incremented. If the pixel lies outside the region it is disabled until the next pixel data is transmitted. This process is repeated for all pixels in the image.

After all regions and intensities have been transmitted, the rank of each pixel within its own contextual region has been computed in the rank counter. This value can then be scaled to the range 0-255 and output by the video controller for display.

3. Timing Estimates

Rough timing estimates by John Eyles and John Poulton indicate that AHE on a 512 x 512 image could be computed in less than 5 seconds. Each pixel represents one polygon, and all pixels must be written to the engine twice, which requires transmission of 256K polygons times something less than 10 microseconds per polygon, or 5 seconds. In addition, the scaling operation must be performed, but since it is done in all pixels simultaneously, would not considerably lengthen the estimated time.

4. Notes and Comments

One image was processed using this algorithm on the VAX, for two different contextual region sizes. No significant differences are present when compared to the images computed by the AHE program. There is better contrast in some areas of the image, probably because there is no interpolation. The intensities near the image edges are somewhat less in the Pixel Planes version because edge conditions are handled differently. A smaller, 64 x 64 image was computed using this algorithm, standard AHE, and uninterpolated AHE. The contextual regions are so small that the resulting images are not particularly useful, but the resulting images are nonetheless nearly identical.

Because this algorithm computes a rank, and not the actual histogram within the region, the intensity range of the processed image is equivalent to the number of pixels in each region. As long as this number is greater than approximately 256 (16 x 16 pixels per region), the processed image will be essentially the same as the actual AHE image. With smaller region sizes, there will be fewer intensity levels available in the image.

The scaling of the rank to a constant display scale is trivial if the contextual region size is a power of two, as the rank counter bits are shifted right or left the appropriate number of bits. If a real division is required, there is apparently a way to approximate division at the pixels that may be accurate enough. Alternatively, the rank data could be transmitted back to the host computer, and the rank scaled as the data is stored away. Other methods of scaling need to be considered.

The pixel planes engine handles arbitrarily shaped regions nearly as quickly as the square contextual regions, so pixel planes would allow regions to be defined arbitrarily. I'm not sure what this means in terms of other AHE alternatives.

A speed improvement could be gained if a method for loading image data directly into the frame buffer was developed. This could prove useful if other pixel oriented algorithms are developed.

Because the algorithm executes so quickly, one could imagine a system where the user has one input device which allows interactive control of the region size, and would therefore allow the best selection of the region size.