

Project Proposal

Bugra Onal

March 1, 2023

1 Intro

As a project idea, I wish to work on a streaming video histogram equalizer. As the name suggests, this is a histogram equalizer intended to work with videos. It will have the capability to receive 1 pixel a cycle and output 1 pixel a cycle.

Histogram equalization is a computer vision algorithm for adjusting the contrast of a grayscale image. In order to accomplish this a histogram of the pixel values will be counted and that histogram will be flattened. This will enable the image to use the full range of the color spectrum.

This application is commonly used in medical devices to enhance the image and make details more apparent.

2 Usage

For this project the histogram equalizer generator will be produced. This generator will have the following parameters:

1. Enable control signals [Bool]: When this option is enabled, a valid signal will be added to the input to control input flow.
2. Frame dimensions [Int, Int]: This option will control the dimensions of the frame. This will affect the number of cycles it takes to equalize a frame.
3. Pixel depth [Int]: This parameter will control the bit-width of the pixels. This will affect the number of words in the memory used to keep track of pixel occurrences.
4. Binning size [Int]: This parameter will determine the size of the bins and will therefore affect the memory size.
5. Histogram depth [Int]: Depth of each bin. Will affect the word size in the memory. This should depend on the image size and binning size, a maximum value can be inferred but will result in big memory words.

3 Architecture

In order for the streaming requirement to be met, one concession must be made. Every frame will be equalized based on the histogram from 2 frames ago. This will remove the requirement for storing the image in memory which will make this application easier to apply to larger sized videos.

The equalization has 4 main parts: The creation of the histogram, which will be referred to as the counting phase; the conversion from PDF (probability distribution function) to CDF (cumulative distribution function) which will be referred to as the accumulation phase; the calculation of the new pixel value, which will be referred to as the mapping phase; and writing back 0s to the memory which will be referred to as the emptying phase.

The intended method is to run these 3 phases in parallel, while the counting phase for the current frame is going on, the accumulation phase for the previous frame will be ongoing. At the same time the mapping phase for the current frame will use the CDF from 2 frames ago.

The mapping equation is as follows:

$$h(v) = \text{round}((cdf(v) - cdf_{min}) / ((M \times N) - cdf_{min}) \times (L - 1))$$

where:

- $h(v)$: The newly mapped value for pixel value v
- round : Function to round to nearest integer
- $cdf(v)$: CDF value for pixel value v
- cdf_{min} : The minimum value of the CDF
- M, N : The dimensions of the frame
- L : The maximum pixel value (ie. $2^{depth} - 1$)

Each of the phases will have their own memories where these memories will be swapped at the end of each phase. This means that the total memory space needed is $2^{depth} \times 4 \times 2^{hist\ depth}$.