FPGA IMPLEMENTATION OF BILATERAL FILTER

Ratnaparkhi Madhura (MS Electrical Engineering)
Sharma Shivangi (MS Electrical Engineering)

Introduction

The fundamental operation of image processing and computer vision is filtering. Many filters in image processing are used to remove unwanted noise from the input image by averaging the image pixels present in a window. This processing is done based on the neighboring pixels with respect to the center pixel of the window. The gaussian filter takes the weighted average and the weights decreases as it goes far from the center pixel resulting in blurred image with linear low-pass filter. In this way the noise is averaged but the signal is preserved. However, the smoothing of image results in loss of small features of the image. Therefore, to prevent averaging across edges while still smoothing the image is achieved by using bilateral filter. The bilateral filter consists of two different gaussian filters, one based on the pixel coordinates and second based on the difference between the intensity pixel value of the image. The equation for Bilateral Filter is given as,

\[ B(u,v) = \frac{1}{2\pi\sigma_x\sigma_y} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x) g(y) \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) \]

The input to the line buffer is the 8-bit data representing gray scale image pixel. The MUX selects the rows from the fully pipelined line buffers to be convolved with the gaussian kernel and the range kernel based on the difference as mentioned. The look up table for convoluting the range gaussian according to the difference between the center pixel to the adjacent pixels within the window is in the range [0, 255]. The image controller instantiates the line buffer from the external world and output is given to the convolution module after concatenation. The output via the bilateral filter IP is displayed on the monitor using HDMI interface.

Design Implementation Details – Bilateral Filter IP

1. Input Image (Command)
2. Line Buffers
3. Image Controller
4. Convolution & Coefficient Selection
5. Output Image (Data in Parallel Format)

The stimulation shows the 8-bit input image data and at the positive clock the image data is filled in the line buffers only when the image data valid signal is high. Once the data has been processed the testbench includes a signal called sent size that shows the number of pixels bits being sent. The input is read in form of the binary data of the image file and the output of that image is written character by character into another file called filt and the 8-bit convoluted output is stored in the out data when ever the out data is valid. The out data valid and image data valid are the master and slave valid signals in order to communicate with the AXI stream TVVALID signal. When the image data valid signal goes low to logic 0 because of the new line buffer logic applied in the design. The integer count is reset to 0 again and the interrupt signal goes high to logic 1 in order to request for new set of input data. And as the new set of data has arrived the interrupt signal goes low to logic 0 and the image data valid signal goes high to logic 1. Furthermore, the sent data also stops at 800 because of the interrupt signal and restarts from 0 for new row of data pixels.

Analysis and Results

- Noisy input image
- Gaussian filter output \( \sigma_s = 16 \)
- Bilateral filter output \( \sigma_s = 16, \sigma_r = 0.2 \)

Due to the use of range sigma values (\( \sigma_r \)), bilateral filter gives more enhanced image as compared to traditional gaussian filter.

The quality of the output image gets enhanced as the kernel size increases.

The image processing performance is calculated in frames per seconds. The formula for calculating pixels per frame is given by:

\[ \text{Frames per second} = \frac{\text{Images per second}}{\text{Frames per second}} \]

Summary/Conclusions

This project implemented the bilateral filter using the brute force architecture and pipelining to improve the performance of overall design. The noise is removed from the image using the spatial gaussian filter and the edges are preserved using the range gaussian filter. The input image is read pixel by pixel in the rows parallelly depending on the window size and the AXI stream interface is used for streaming the input into the bilateral IP and then streaming the processed output data from the IP to the FPGA and is displayed on the monitor. The performance analysis of the designed IP shows that the signal to noise ratio of the image increases as the kernel size increases. Also, the accuracy with hardware stimulation decreases but the speed increases in comparison with software. This design can be incorporated with a camera for real time input data and the IP can be modified to convert the RGB image into gray scale image for real time image processing purposes.

Key References


Acknowledgments

This is to acknowledge the constant guidance and support from our project advisor Prof. Chang Choo. We would also like to thank project coordinator Prof. Morris Jones and Electrical Engineering Department for providing the platform for presenting our project.