COL 215

IMAGE FILTER

Submitted By: Sourav Bansal (2018CS50421)
Manav Bansal (2018CS50410)

Introduction

Convolution has been widely used in computer vision and image processing, including object recognition and image machining. Image filtering is used in pre-processing to eliminate useless details and noise from the image. These operations when done on hardware can be done parallelly which can reduce the run time considerably.The project has been implemented in VHDL and synthesized on the FPGA board.

Aim of the Project

- Designing an Image Filter that uses a 3X3 Sliding Window.
- Downloading the image file to be filtered using a serial Receiver from a file into the memory.
- Apply filtering operations on the image which includes Smoothening Filter and a Sharpening Filter and integrating a switch to change it.
- Uploading the filtered Image back into another file through Transmitter.

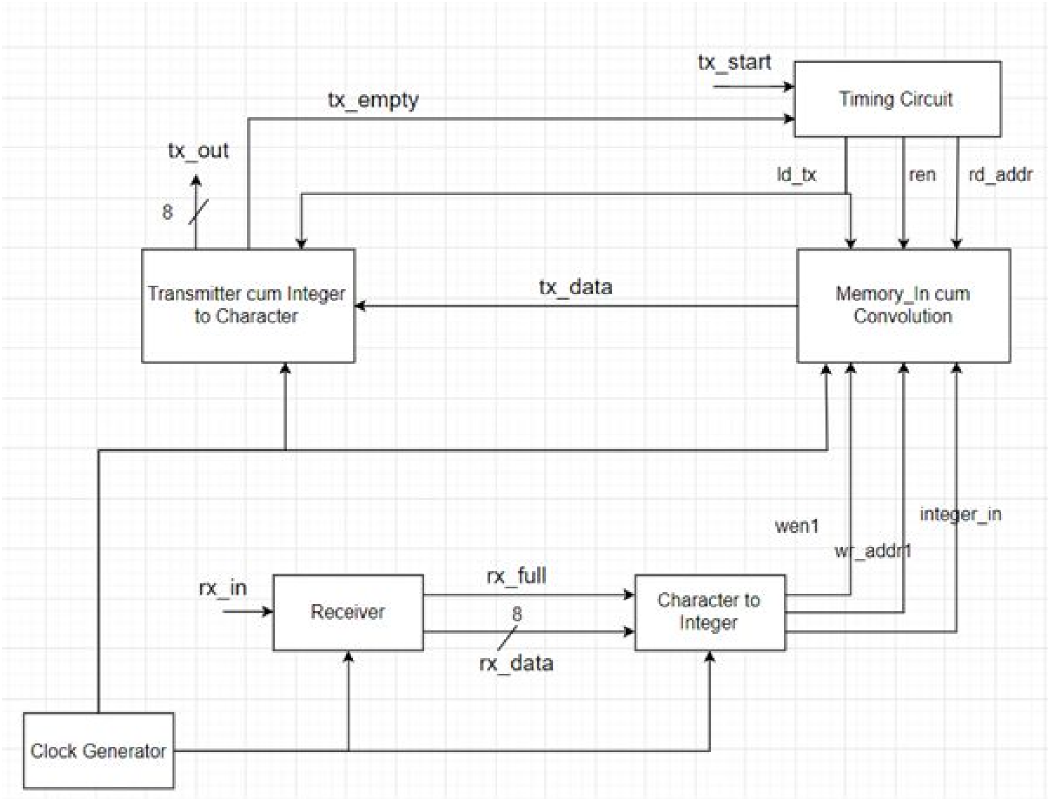
Design

Input: Image file in pgm format inputted through gtkterm into our board. A switch for changing the mode has been provided with ‘1’ for smoothing mode and ‘0’ for sharpening mode. A push button is used to signal the start of image filtering operations.

Output: The pixel values of the image after convolution are outputted in the gtkterm which form the pgm image. The final image has (n-2) X (m-2) size if the size of the initial image is n X m.

Modules: We have implemented the following modules for the project. All these are listed below:

1. **Receiver:** This module receives the pixel values (each digit as a character) from the file through gtkterm serially. When a single digit value has been received, the signal rx_full is set to’1’ and the bit vector (rx_data) containing the digit value (ASCII Code) is sent to next module for processing.
2. **Character to Integer:** It takes the bit vector from receiver which are the digits of the pixel values. This module inputs the digits till it receives a character whose ASCII code does not lie in between 48 to 58 (those corresponding to digits). When character other than digits is received, it initiates a calculation to calculate the pixel value. Simultaneously, a signal write enable is enabled and the integer to be written is sent as integer_in. The write address is sent as an integer.



3. **Memory_In cum convolution:** This module involves the storage and the convolution process. The integers received are stored for convolution. When the tx_start signal is received from the push button, the convolution process is started. The convoluted signal as per the algorithm provided iterated over the pixel values using the sliding window. When the mode is '1' the sliding window corresponding to smoothening is used while that of sharpening is used when mode is set to '0'. The writing operation is enabled by the write enable signal wen1 whose address is received as wr_addr1. The read is enabled with a signal ld_tx which starts reading, convolutes and outputs the data as an integer tx_data.

   The sliding windows are:

Smoothening filter

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

Sharpening filter

$$\begin{bmatrix} -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \\ -\frac{1}{9} & 1\frac{8}{9} & -\frac{1}{9} \\ -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \end{bmatrix}$$

4. **Transmitter cum Integer to character:** It takes the integer as an input from the memory cum convolution module and transmits them bit by bit to the gtkterm terminal. When a pixel value is received from the memory modules, this module performs successive divisions and transmitter the dividend as “digit+48” which is the ASCII code for the corresponding digits. When the pixel value has been transmitted, a space is transmitted to differentiate different pixels.

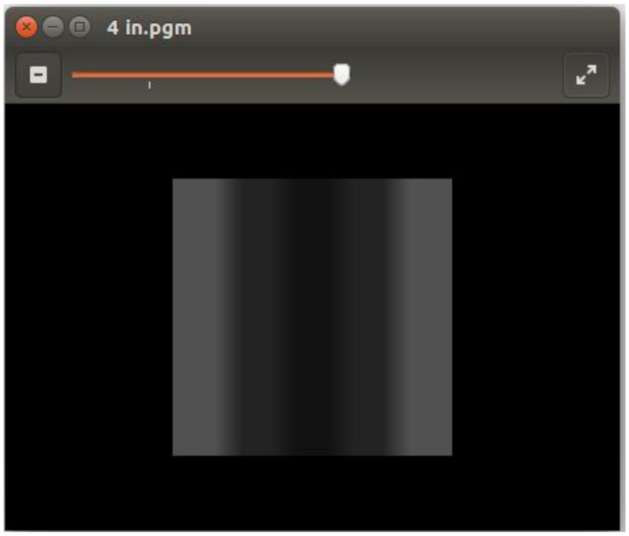
   The internal working of the transmitter is driver by three extra states apart from those being used in UART. The tx_empty is set to ‘0’ after the pixel value which may contain 1,2 or 3 digits has been transmitted completely which initiates the convolution of the next pixel value.

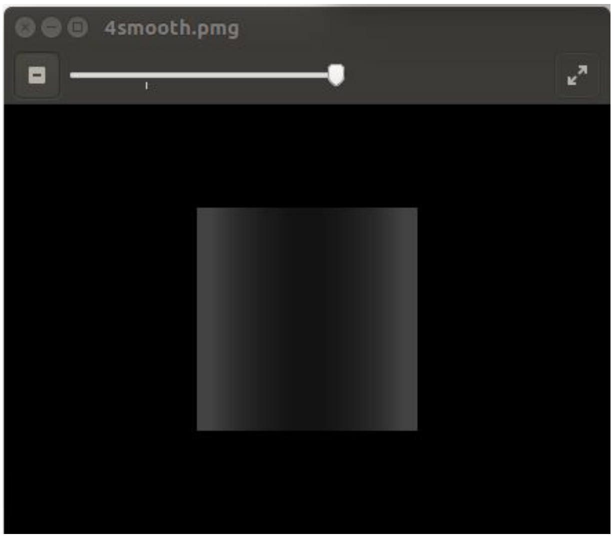
5. **Timing Circuit:** This controls the communication between different modules. The write address starts from 1. The read address is initiated from 3 so as to skip the convolution for the height and width integers. The write address is incremented by 1 in each cycle of writing, however, the read address is incremented by 3 when the pixel to be read is second last in the row else by 1. This ensures that the convolution runs only for the legitimate pixel values.

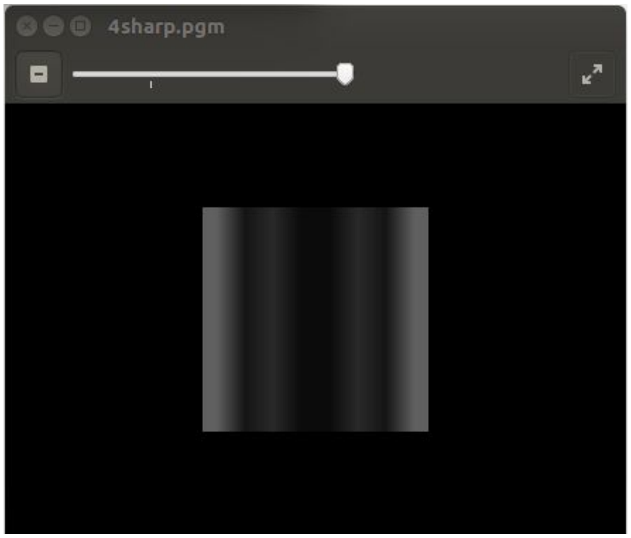
Whenever tx_start is pressed the timing circuit starts the convolution setting ld_tx to 1 so as to initiate the convolution and transmission of that integer. The circuit waits till the tx_empty signal is 1 and then starts the next cycle.

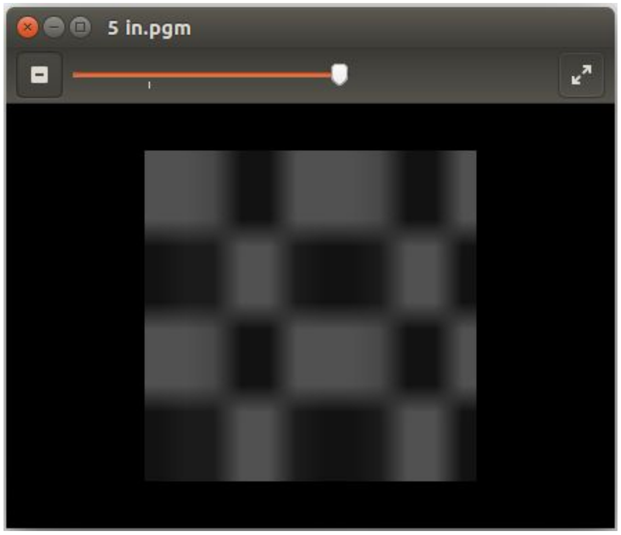
Test Cases

Outputs and inputs are the following:

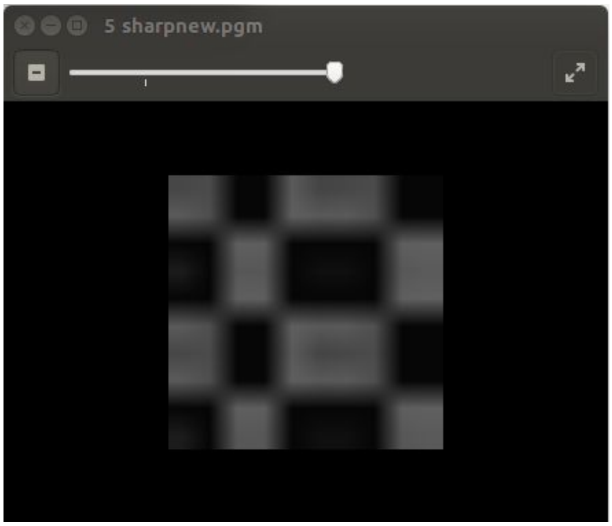




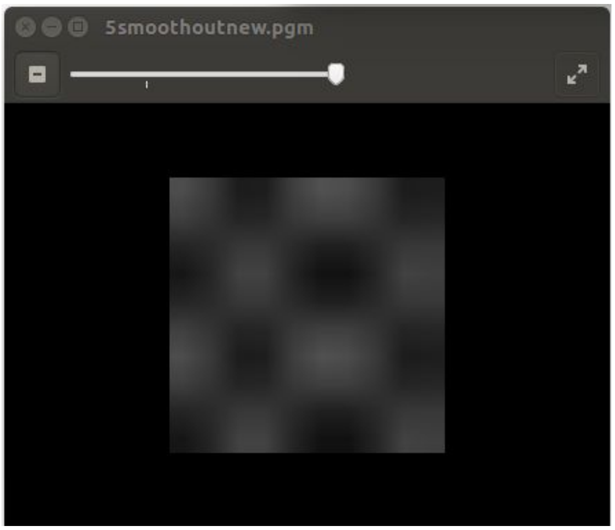




5 in.pgm



5 sharpnew.pgm



5smoothoutnew.pgm

Highlights of the project:

- The Receiver modules read the original image as set of character, extracts height, width and the pixel values as integers from it. The whole process has been done without using external python or C programs
- The Transmission module transmits the image in the original format in the form of pixel values. The integer is split into the three corresponding digits and transmitted

Real Applications of the project:

- Can be used by Investigation departments to apply image filters to extract information about a person.
- Can be used to improve image quality to allow better Optical Character Recognition on images to extract text.
- Can be used in general photo editing tools.