Project Proposal: Implementation of Median Filter on Z -Board and AWS

Anushka Swarup, Medini J. Aradhya, Liming Xu

Project description:

The Median Filter is a nonlinear digital filtering technique for removal of noise from images. It works particularly well for certain types of random noises like the salt and pepper noise. This method is often used as a preprocessing step for edge detection algorithms. It is advantageous as this method better preserves the edges in certain instances.

Implementation of this filter using the AWS F1 FPGA Acceleration Platform. The amazon web services cloud platform is compatible with existing FPGA algorithms hence makes it a versatile open source platform for accelerated computing.

What is the basic algorithm?

The Median filters comes under the category of Order-Statistic filters. These filters are nonlinear in nature and their response is based on ordering the pixels contained in the region encompassed by the filter. One application of Median filters is in smoothing of images. It is achieved by replacing the value of the center pixel by the median of the intensity values of the neighborhood of that pixel. Median filters are efficient in noise reduction for salt-and-pepper noise.

The basic algorithm follows that the median, j, of a set of values is such that half the values in the set are less than or equal to j and the other half are greater than or equal to j. For example, in a 3x3 neighborhood the median is the 5th largest value. Thus, the main aim of median filters is to force pixels to be more like their neighbors. In order to perform median filtering a window of an arbitrary size is made to move over the entire image. At every iteration, we sort the values of the pixels inside the window, determine their median, and assign the median value to the pixel at the center of the window in the filtered image.

Why is this application amenable to FPGA-based reconfigurable computing?

Median filtering is mainly used in digital image processing. It is preferred since it preserves edes while getting rid of noise by taking the median within the window selected and sliding the window over the entire image/signal.Since these processes are repetitive and simple to carry out implementation on an FPGA would provide appreciable speedup. Adding to this the parallelism functionality of the FPGA would allow for efficient resource utilization.The datapath would define the speed of the system as discussed subsequently. This when compared to the software implementation if the same should provide considerable improvement. Lastly, the reconfigurability of the system allows for varied window types to be implemented within the same hardware platform not taking away from the speed-up.

Proposed approach and preliminary design:

Are you designing/implementing this application “from scratch” or are you leveraging existing works from others: e.g., research papers, algorithms, software implementation of the algorithm, etc.

1) Software: For the software implementation of the Median filter we will be leveraging an existing work. A C++ implementation of the median filter for digital image processing will be used. The output from this software code as well as the time for execution will be analysed with the hardware implementation of the median filter.

2) Hardware: research papers?
   The controller and memory buffer design would be similar to that of the sobel filter.

The datapath for the median filter have a few possible implementations. One trivial solution would be to use bubble sort implemented in hardware. This would be designed from scratch. There are also several existing designs for the median filter from research papers. These would have less latency and use up less space.

3) AWS: Xilinx FPGAs are available in the Amazon Elastic Compute Cloud (Amazon EC2) F1 instances. These Instances can be used to implement the filter on the cloud using the Xilinx Vivado design suite.
   AWS has Xilinx SDAccel - Development Environment allowing to run the RTL designs in the F1 instance. It also automates the acceleration of code written in C, C++ by building application-specific accelerators on the F1 instances.

Team members expertise and work distribution :

1) Liming & Anushka : CV experience - Software implementation of filter
2) Medini - Research on AWS and implementation of filter on AWS
3) All - VHDL hardware implementation.

Preliminary design: Software/hardware partitioning of the design, datapath design.

For the design of the datapath of the median filter, one simple design would be to simply use a bubble sort to sort the array of pixels and take the median. This would take 8+7+6+...+1=36 comparisons. Since the throughput has the most influence on the speedup, not the latency, this would be a viable solution. Considering only the median is needed, not the whole array need to be sorted, just the largest(smallest) 5 numbers. Thus the comparisons can be reduced to 8+7+6+5+4=30. This approach does not utilize any form of parallelism, thus is causing much latency and also uses a lot of hardware space.

A theoretical solution would be to compare each pair of pixels and use the whole 72 results to sort the array. This would only create a latency of 2, but requires a LUT with 72 bit input, which is not realistic.

The best approach would be a trade-off between latency and complexity. Three algorithms can be found in existing research papers. Using parallelism, in each stage multiple sets of data are processed. Each one of them takes up reasonable hardware space and causes much less latency. These would be the preferred solutions.

The controller design would be similar to that of the sobel filter, and the buffer would be the same as well, except the buffers would need to input 9 pixels at a time instead of 8.

Demonstration plan (priliminary proposal):

1) Implemented design of Median Filter on FPGA (Zed Board) and on AWS F1 Instance.
2) Functional and performance comparison against software implementation.
3) Comparison of the performance parameters with respect to FPGA and AWS platform.
   Test data : Images available online (using both small and larger images)

Proposed schedule for the remainder of the semester :

First two weeks: (22nd March- 8th April)

1) Research

2) AWS tutorial
3) Software Implemetation of Median filter.
4) Hardware - Datapath.

Remainder : (9th April - 29th April)

5) Hardware: controller ,memory.
6) Debugging and running on Zed Board.
7) Implementation using AWS F1.
8) Analysis of results.
9) Report writing & Presentation

References:

1) Leiou Wang, "A new fast median filtering algorithm based on FPGA," *2013 IEEE 10th International Conference on ASIC*, Shenzhen, 2013, pp. 1-4.
2) Gonzalez, Rafael C., and Richard E. Woods. Digital Image Processing. Pearson, 2018.
3) https://github.com/aws/aws-fpga
4) https://www.xilinx.com/products/design-tools/acceleration-zone/aws.html#gettingstarted
5) https://www.xilinx.com/products/design-tools/software-zone/sdaccel.html#overview
6) https://www.programming-techniques.com/2013/02/median-filter-using-c-and-opencv-image-processing.html