DOI: http://dx.doi.org/10.26483/ijarcs.v8i8.4704

ISSN No. 0976-5697

Volume 8, No. 8, September-October 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

IMPLEMENTATION OF GRADIENT BASED ADAPTIVE INTERPOLATION ON FPGA

Usha Patel
Department of Computer Science and Engineering
Institute of Technology, Nirma University
Ahmedabad, India

Vibha Patel
Department of Computer Science and Engineering
Institute of Technology, Nirma University
Ahmedabad, India

Abstract: FPGA (Field Programmable Gate Array) structure can exploit the temporal and spatial parallelism of the system, subject to hardware constraints and processing mode of the system. Due to this property, real time image processing algorithms can implemented using this platform. Now a days High level Synthesis Tools (HLS) available. Which can used to configure FPGA. This paper presents a real time implementation of gradient-based adaptive interpolation using HLS. We received a speedup of 14X over general purpose processor.

Keywords: FPGA, Super resolution, Interpolation, Altium Designer 10, Altium nanoboard NB2DSK, real time

1. INTRODUCTION

Many image processing application required High resolution (HR) images for further processing. This HR images used for Computer Vision and Image classification algorithms. Due to hardware limitation of camera some time it is difficult to take HR images. Technique to convert low resolution (LR) to High resolution is called super resolution. Which is low cost solution of HR images?

Real time super resolution is ability to perform super resolution at higher rate. Real time image processing with 25 frame per second and resolution is 720 X 580 required almost 1,01,400 operations. Here we are not consider any overhead of data reading.

Super resolution requires several operations on a single pixel. So time constraint is the major Challenge in implementing such kind of applications.

SR image reconstruction method having three steps registration, interpolation, and restoration. [1] This paper presents a real time implementation of gradient based interpolation for reconstruction of SR image. This system is implemented on FPGA device because it provide re-configurability. It also provide the requirements of real time image processing algorithm. [2]

To reduce design times and raise the design abstraction level we use High Level Synthesis Tool (HLS). The HLS reduce design time as well as it also help in verification of design and power analysis. In order to create an FPGA design, a designer has several options for algorithm implementation. The implementation consist of Altium NanoBoard 2 and the design in Altium Designer 10.

In this paper the algorithm for gradient based interpolation is described in Section 2. Section 3 described the experimental setup. In section 4 results and numerical analysis are shown, and the paper conclusion is in Section 5.Future scope is given in section 6.

2. THEORY

Super resolution image reconstruction is a process to recover high resolution image data from aliased data in multiple low resolution image of same scene.

Multiple LR frame of same scene having spatial or temporal motion between sub-pixels. So each LR frame having different set of pixels of same scene. So in order to reconstruct SR frame if motion between multiple LR frames is estimated then LR frame can mapped to appropriate HR grid position. The first step of SR reconstruction is registration where motion between multiple LR frames is estimated. For this frequency domain registration algorithm is used. [3] Using this motion value LR frames are mapped to HR grid. The second step of SR reconstruction is interpolation where gradient based adaptive interpolation is used. [3] The third step of SR reconstruction is reconstruction where wiener filter is used to reduce the effects of noise and blurring caused by the system.

In these implementation we assume that motion vectors are known, and based on that motion vectors Gradient based adaptive interpolation method is implement on hardware. In this approach of image reconstruction it consider the distance of neighbouring pixel and local gradient of original image.

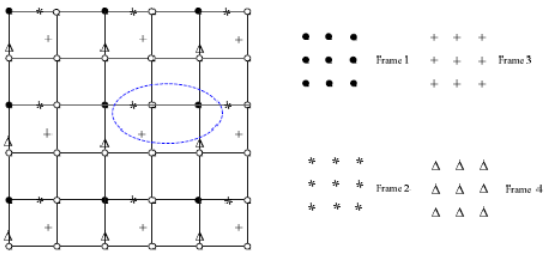

Fig. 1 High Resolution Grid

The algorithm steps are as follow: Take the reference LR image and place it on HR grid at [0, 0]. Take other LR frame of same scene and determine the position relative to the HR grid. For that apply motion estimation algorithm in frequency domain.

Find the three nearest pixels around the interpolated one in the high resolution grid points. In order to find nearest pixel Euclidean distances is used between the neighbouring pixels and its interpolated pixel.

To compute interpolated point value from three nearest points using gradient-based adaptive interpolation algorithm as follow.

$$S(x,y) = \frac{\sum_{i=1}^{i=3} T(x_i,y_i)D(x_i,y_i)S(x_i,y_i)}{\sum T(x_i,y_i)D(x_i,y_i)}$$

Where, S(x, y) is the grey value of the interpolated point and D is the distance function.

$$D = (1 - \Delta x)(1 - \Delta y)$$

Where delta x is horizontal distance and delta y is vertical distance between calculated and nearest pixel. And T is the gradient weight function.

$$T(x_i,y_i) = (-\mu Gd(x_i,y_i) + 1)^m$$

μ and m both are positive values, and Gd is the local gradient value.

$$Gd(x_i,y_i) = \frac{f'x(x_i,y_i) + f'y(x_i,y_i)}{\sqrt{f'x(x_i,y_i)^2 + f'x(x_i,y_i)^2}}$$

Where, f'x is 3 X 3 horizontal sobel mask and f'y is 3 X 3 vertical sobel mask in LR image which (xi ,yi) belongs to. Signal processing, image processing, real time application and many such applications require high processing and high computation and at a very fast rates. It is difficult for general purpose processors because of power consumption and the size of the computer system and computation overhead are of concern. So, for embedded system special hardware solutions are required which can be Field Programmable Gate Array (FPGA) or digital signal processor (DSP). As increasingly complex algorithms and applications are being developed, the performance demands of these algorithms increasing exponentially.

The FPGA internal architecture provide fine grain parallelism, which is most suited for parallel algorithms. [4]

The FPGA provide small size of on-chip memory and now memory interfaces, lack of wide-word processing units. And it is difficult to implement complex numerical operations like square root and division. FPGA's are semiconductor devices which can be reconfigured. [5]

The FPGA has configurable logic blocks (CLBs), input/output blocks (IOBs) and Interconnects major elements. All this three components we can reconfigure. The CLBs having the functional elements which is used for design user's logic. The IOBs provides connection between internal signal lines and package pins. The programmable interconnect use to connect input and outputs of the IOBs and CLBs. Using these three basic components FPGA can implement any functions. [6]

These basic components are building blocks for FPGA which provides flexibility of implementation.

3. IMPLEMENTATION

To reduce design times and raise the design abstraction level we use High Level Synthesis Tool (HLS). In order to create an FPGA design, a designer has several options for algorithm implementation. [7] For implementation consist of Altium NanoBoard 2 which is design in
Altium Designer 10. The design system specifications are as below:

- Input Data: 320 X 240 resolution video as per the CIF standard
- Output Data: 640 X 480 resolution video as per the SDTV standard
- Frame rate: 30 frame/second
- Gray scale images
- Assumptions: No noise is present in image and no blur effect in input images

*A. **Hardware Design:***

Altium NB2 having Video peripheral board PB01 which is connected to Spartan3. Following are the steps to design hardware for project.

- Using DVD Player give input to Altium NB2 via PB01 (Video peripheral board). Video decoder (TVP5150AM1) used to convert analog input to digital (YCbCr format).
- Using BT656 controller the formatted data stored in external video memory. Which can further used by Spartan 3.

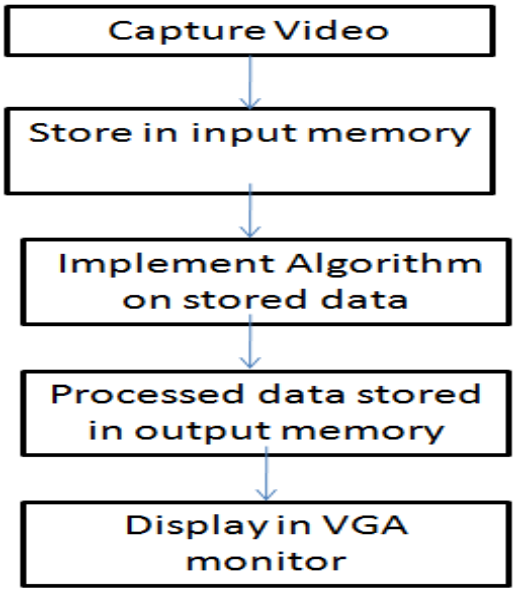

Fig. 2 Design Flow

Device Resources	Usage Summary
4 Input LUTs	12,761 / 26,624 (47%)
Block RAMs	23/32 (71%)
I/O Pins	222/487 (45%)
Global Clocks	3 / 8 (37%)
MULT 18X18s	12 / 32 (37%)
Slice Flip Flops	7,925 / 26,624 (29%)
Slices with only related	9,579 / 9,579 (100%)

logic	
Slices with unrelated logic	0 / 9,579 (0%)
Slices	9,579 / 13,312 (71%)
Total 4-Input LUTs - Logic	14,158 / 26,624 (53%)

- TSK3000A read and processed memory content and returned to external memory.
- Then the VGA controller reads the content of external memory and output is seen on the VGA monitor connected to VGA connecter.

Three major components were used as a part of design in open bus document are BT656 Controller, 8-bit VGA Controller, TSK3000A processor (soft core)
From within the OpenBus System, this mapping can be verified using the Configure Processor Memory and Configure Peripherals dialogs, respectively. After placing all the components in the design we now first we need to configure it. After the hardware design, we need to add software to project.

Fig. 3 Setup

4. RESULT

The result give comparative analysis of implementation of MATLAB with Nanoboard 2. Here we compare the computation power of Nanoboard 2 with MATLAB. The obtain result for the algorithm Gradient based adaptive interpolation. The MATLAB implemented system specification are as below:

- Processor : Intel Core I5 CPU @ 2.50GHz
- Memory(RAM) : 4:00GB
- System type(Operating System) : 64-bit

The algorithm implementation using Altium can handle 24 frames per second. The same algorithm implemented using MATLAB and result analysis is given as below:

Table 1: COMPARISON OF RESULTS IN MATLAB AND NANOBOARD

MATLAB	**Altium Nanoboard (ASP)**	**Speedup**
0.38 sec	0.042 sec	14X

Resource utilization is measured for project with the help of design summary utility provided by Altium Designer. Below is the resource utilization summary. [8]

5. CONCLUSION

In this paper, we have discussed the real time implementation of Gradient based adaptive interpolation method. We have analysed properties of super-resolution method applied to the interpolation of a single image. It was shown that gradient based method can gives the good visual quality of resulting images. The goal of hardware assisted super resolution is to achieve low power usage, real-time performance, reliability, and ease of integration with existing display devices. A parallel architecture of the system has been designed and implemented capable of outputting 25 frames per second with an image size of 640 X 480 pixels. FPGAs helps in achieving better performance by bypassing the fetch-decode- execute overhead of general purpose processors.

6. FUTURE WORK

In future the motion estimation algorithm can be implemented in hardware. Further improvements can be achieved by identifying parallel execution paths and implement the same in FPGA. The algorithm can be improve and tested in presence of different types of noise. The system can be implemented in hardware descriptive language and evaluate the performance of HLS.

REFERENCES

[1] Park, Sung Cheol, Min Kyu Park, and Moon Gi Kang. "Super-resolution image reconstruction: a technical overview." IEEE signal processing magazine 20.3 (2003): 21-36.
[2] Li, Xuelong, et al. "A multi-frame image super-resolution method." Signal Processing 90.2 (2010): 405-414.
[3] Chu, Jinyu, et al. "Gradient-based adaptive interpolation in super-resolution image restoration." Signal Processing, 2008. ICSP 2008. 9th International Conference on. IEEE, 2008.
[4] Nana, Lv, et al. "Super resolution video reconstruction in DSP+ FPGA based on lifting wavelet." Electronic Measurement & Instruments (ICEMI), 2011 10th International Conference on. Vol. 3. IEEE, 2011.
[5] Brown, Stephen, and Jonathan Rose. "Architecture of FPGAs and CPLDs: A tutorial." IEEE Design and Test of Computers 13.2 (1996): 42-57.
[6] Narasimha, P. Lakshmi, et al. "An improved real time superresolution FPGA system." Proc. SPIE. Vol. 7328. 2009.
[7] Callico, Gustavo M., et al. "Analysis of fast block matching motion estimation algorithms for video super-resolution systems." IEEE Transactions on consumer electronics 54.3 (2008).
[8] www.altium.com.