Performance Analysis of Non Local Means algorithm using Hardware Accelerators

Daniel Sanju Antony and Rathna G.N

Department of Electrical Engineering, Indian Institute of Science Bangalore

Objectives

To analyze the performance of Non Local Means (NL-Means) Denoising Algorithm on following hardware platforms and compare their results.

- Implementation on CPU.
- GPU implementation using OpenCL.
- Implementation on FPGA.
- Fast NL-Means algorithm on CPU and GPU and its comparison with Gauss polynomial bilateral filter implementation.

Introduction

Image restoration forms an important part of image processing techniques. It is defined as the process of reducing or removing degradation that has incurred during capture of the image. Degradation can be due to noise or blurring. Image Denoising deals with images that are corrupted with noise. It tries to obtain clean image from corrupted image.

Commonly used Denoising algorithms are

- Anisotropic Filter
- Wavelet Filter
- Bilateral Filter
- Non Local Means Filter
- BM3D Filter

Figure 1 shows an example of image denoising.

Figure 1: Image 1 shows the original, image 2 is the noisy image and image 3 is the denoised image

Disadvantages of denoising filter:

- Computational complexity
- Increased processing time.
- Difficult to implement in real time.

Non Local Means Filter Using OpenCL

Figure 2 shows the differences in CPU and GPU hardware.

Figure 2: CPU vs GPU

NL-Means Algorithm was first introduced by Antoni Buades et al [1] in 2005.

Implementation of Non Local Means Algorithm on GPU using OpenCL is shown in the figure below 3.

- Box and Gaussian kernel algorithms are implemented and their PSNR values are compared and shown in table 1.
- The algorithm is implemented on GPU using OpenCL.

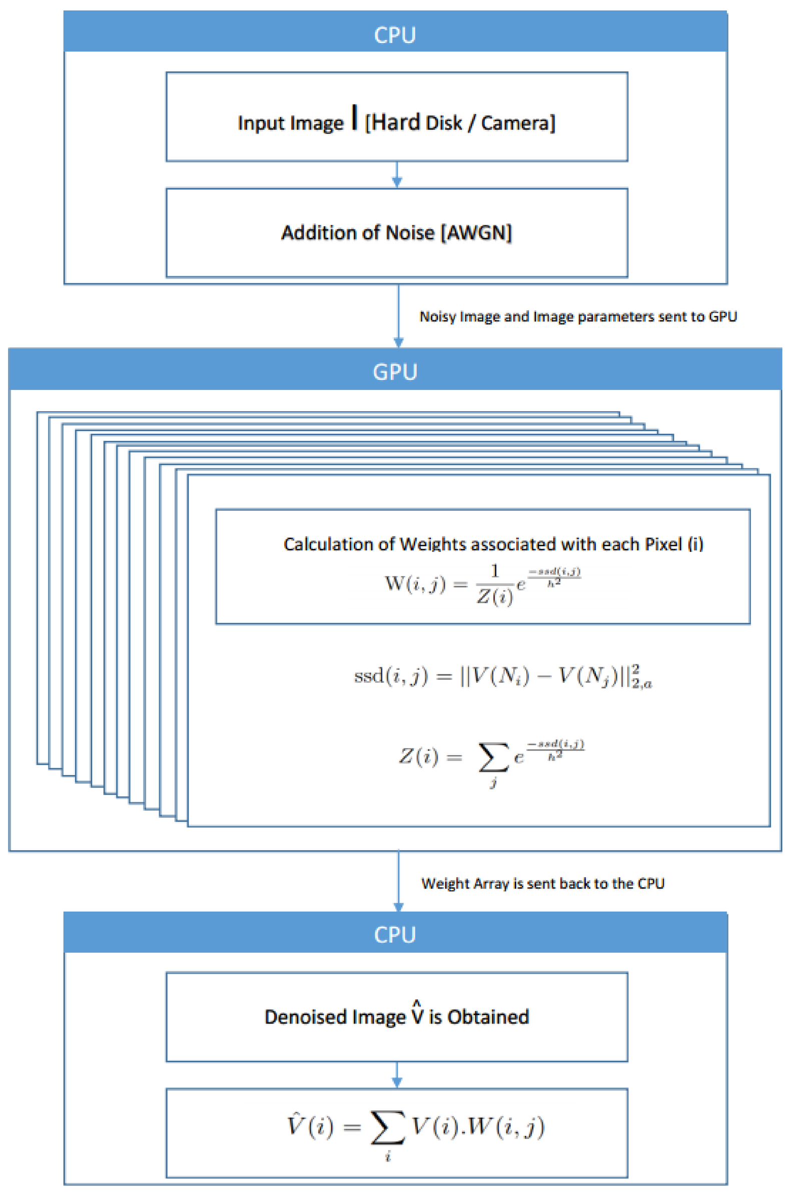

Figure 3: Basic Flow Diagram of Non Local Means Denoising

- Denoising time for CPU and GPU implementation is compared and given in table 2.

Table 1: PSNR Comparison between Gaussian Kernel and Box Kernel Filter

	PSNR Values (dB)		
Image Size	Noisy Image	Gaussian	Box
128×128	3.32	19.67	21.06
256×256	3.324	19.31	21.02
512×512	3.323	18.94	20.98

Table 2: Performance Comparison for NL-Means Algorithm

	Time Taken for Denoising (ms)		
Image Size	CPU	GPU	Speedup
64×64	172	36	4.77
128×128	967	78	12.4
256×256	3866	189	20.45
512×512	15373	615	24.99
1024×1024	61801	2314	26.70

GPU Based Fast Non Local Means Algorithm

Based on works by Darbon et.al [2] and YL Liu et.al.

- Most time consuming part of NL-Means is weight calculation.
- In the modified algorithm, efficient computation of weights is done with the help of sum squared image and its translations. The basis behind Sum Squared Image is *Integral Image* used in face detection algorithms.
- Modified NL-Means algorithm is used to improve the speed of weight calculation.

Consider a 1D image with n pixels i.e $\Omega = [0, n-1]$. Consider a new image Vdx and translation vector dx,

$$\text{Vdx}(p) = \sum_{j=0}^{p} (V(j) - V(j+dx))^2, p\epsilon\Omega \qquad (1)$$

where Vdx corresponds to sum of the squared difference of image V and its translation by dx.

Proper reparameterization can be used to make the computation of weights associated with each pixel dependent only on the Sum Squared Image Vdx and its translation images.

The fast NL-Means algorithm implementation in GPU is compared with the faster adaptation of Bilateral filter called Gauss-Polynomial Bilateral filter(GPF) by K.Chaudhury [3].

- GPF and a modified version of the algorithm with Box spatial kernel is implemented.
- Algorithms are parallelized and implemented on GPU using OpenCL API (application program interface).

The performance comparison is shown on table 3 and 4.

Table 3: Performance Comparison for Fast Non Local means

	Time Taken for Denoising (ms)		
Image Size	CPU	GPU	Speedup
256×256	40	12	3.33
512×512	198	16	12.38
1024×1024	787	18	43.72
2048×2048	2974	36	82.61
4096×4096	11448	132	86.72

Table 4: RUN TIME and PSNR COMPARISON of GPF and BsGPBF

	Denoising Time(ms)		PSNR Values	
Image Size	GPF	BsGPBF	GPF	BsGPBF
64×64	3.5	3.2	13.478	16.321
128×128	6.7	5.2	14.026	16.322
256×256	25.4	14	15.833	16.310
512×512	126.1	55.7	17.193	16.298
1024×1024	815.7	260.9	18.380	16.297

FPGA Implementation of NL-Means Algorithm

Why FPGA?

- Even though GPU based Fast NL-Means improve denoising time drastically, the power requirement associated with GPU is very high (150-250W) as compared to FPGA (3-5W).
- As a result, FPGA based implementation is suitable for low power devices.
- FPGA is also small in size and is a stand alone device thus enabling its use in embedded systems whereas GPU requires a separate PC for its operation.
- FPGA can be easily re programmable and therefore can replace ASIC implementation.

It has been found that the FPGA implementation of NL-Means algorithm for a 64x64 image takes 0.3ms to execute while implementation of the algorithm takes 172ms and 36ms in CPU and GPU respectively.

Conclusion

We investigated the performance of Non Local Means Algorithm on different hardware accelerators and the results are tabulated. It can be seen that the implementation of the algorithm in GPU and FPGA reduces the computation time required drastically. Algorithm is easier to implement in GPU using OpenCL than in FPGA. However the power consumption of GPU(150W) and CPU(80W) is very high compared to that of FPGA (3W).Therefore FPGA implementation is ideal in places where power consumption is an important factor like Satellite Imaging, Portable camera and Embedded systems whereas GPU implementation is useful in high performance systems.

References

[1] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[2] Jérôme Darbon, Alexandre Cunha, Tony F Chan, Stanley Osher, and Grant J Jensen. Fast nonlocal filtering applied to electron cryomicroscopy. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 1331–1334. IEEE, 2008.

[3] Kunal N Chaudhury. Fast and accurate bilateral filtering using gauss-polynomial decomposition. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2005–2009. IEEE, 2015.