ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES



ADVANCED PAN SHARP TECHNIQUES WITH CUDA

M.Sc. Thesis by

Aykut KAŞKAYA

Department of Computer Engineering

June, 2019

ANKARA

ADVANCED PAN SHARP TECHNIQUES WITH CUDA

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering, Department of Computer Engineering

by

Aykut KAŞKAYA

June, 2019

ANKARA

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"ADVANCED PAN SHARP TECHNIQUES WITH CUDA"** completed by **AYKUT KAŞKAYA** under the supervision of **ASSOC. PROF. DR. BAHA ŞEN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Baha ŞEN

Supervisor

Prof. Dr. Fatih V. ÇELEBİ

Jury Member

Assist. Prof. Dr. Fatih NAR

Jury Member

Prof. Dr. Ergün ERASLAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: 14.06.2019 **Signature:**

Name & Surname: Aykut KAŞKAYA

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Dr. Baha ŞEN for his tremendous support and motivation during my study. His immense knowledge and precious recommendations constituted the milestones of this study. His guidance assisted me all the time of my research and while writing this thesis.

I also would like to express my gratitude to my beloved wife for her patience during this study and to my lovely son.

2019, 14 June **Aykut KAŞKAYA**

ADVANCED PAN SHARP TECHNIQUES WITH CUDA

ABSTRACT

Satellite imagery is the taking photographs of a location by using artificial satellites in orbits. However, they are not the images, as we usually understand. The satellites that obtain the images contain some special sensors to capture and record multi-spectral data. These sensors obtain data from a tiny part of the Earth's (or wherever) surface. RGB color space and panchromatic band are some of the data. RGB color space is consist of three components that are red, green and blue also known as primary colors. A color can be obtained by mixing these three base colors. This color space is used by computers, graphic cards and monitors. Panchromatic band is a single-band and a type of grayscale image that has a wide bandwidth. Obtaining a high-resolution multi-spectral image from a satellite is an expensive process. However, getting a high-resolution panchromatic image is relatively cheaper. It is possible to achieve getting a high-resolution multi-spectral image using a low-resolution multi-spectral image and a high-resolution panchromatic image. This process is called as pan sharpening and the output is called as pan sharped image.

In this study, the aim is how to make pan sharpening techniques with CUDA -trademark of NVIDIA- which is a parallel computing platform and programming model. Throughout this study three pan sharpening techniques are selected to deal with. These are Simple Mean Algorithm, Brovey Transformation and Hue-Saturation-Value (HSV) Transformation.

As a result of studies, implementation of the algorithms on the GPU instead of on the CPU is efficient depending on the dimension of the image and the nature of the algorithm. Twenty images from different sources is used in this study and in every case, a speed-up (CPU-GPU ratio) is acquire.

Keywords: Satellite imagery, multi-spectral image, panchromatic band, pan sharpening, pan sharped image, CUDA, Simple Mean, Brovey, HSV

CUDA İLE GELİŞMİŞ PAN KESKİNLEŞTİRME TEKNİKLERİ

ÖZ

Uydu görüntüleme bir konumun yörüngeden görüntülenmesidir. Ancak bu görüntüleme tekniği genel olarak anlaşılan görüntülerden farklıdır. Görüntülerin elde edildiği uydular multispketral verilerin yakalanmasını ve kaydedilmesini sağlayan özel alıcılara sahiptir. Bu alıcılar Dünya yüzeyine (ya da her neresi isteniyorsa) ait ufak bir bölümden veri toplar. RGB renk uzayı ve pankromatik bant bu verilerden bazılarıdır. RGB renk uzayı ana renkler olarak da bilinen 3 temel renkten –kırmızı, mavi ve yeşil- oluşur. Herhangi bir renk bu üç rengin birleşiminden elde edilebilir. Bu renk uzayı bilgisayarlar, grafik kartları ve ekranlarda kullanılmaktadır. Pankromatik bant ise tek bir banttan oluşan ve geniş bant genişliğine sahip bir çeşit gri tonlamalı görüntüdür. Uydulardan yüksek çözünürlüklü multispektral görüntü elde etmek pahalı bir işlemdir. Ancak yüksek çözünürlüklü pankromatik görüntü elde etmek nispeten daha ucuzdur. Düşük çözünürlüklü multispektral görüntü ve yüksek çözünürlüklü pankromatik görüntü kullanılarak yüksek çözünürlüklü multispektral bir görüntü elde etmek mümkündür. Bu işlem pan keskinleştirme olarak adlandırılırken çıkan ürün ise pan keskinleştirilmiş görüntü olarak adlandırılmaktadır.

Bu çalışmada, hedeflenen NVIDIA'nın ticari bir markası olan paralel hesaplama platformu ve programlama modeli CUDA üzerinde pan bileme tekniklerinin nasıl yapılacağının keşfedilmesidir. Çalışma boyunca üç adet pan keskinleştirme tekniği üzerinde uğraşılmıştır. Bu teknikler Basit Ortalama Algoritması, Brovey Dönüşümü ve Ton-Doygunluk—Değer (HSV) Dönüşümüdür.

Çalışmaların sonucunda, işlemci (CPU) yerine grafik birimi (GPU) üzerinde bu algoritmaların uygulanmasının, görüntünün boyutu ve uygulanan algoritmanın doğasına göre değişiklik göstermekle beraber, verimli olduğu görülmüştür. Bu çalışmada farklı kaynaklardan elde edilen yirmi adet görüntü kullanılmış ve her bir görüntü için hızlanma (CPU-GPU oranı) elde edilmiştir.

Anahtar Kelimeler: Uydu görüntülenme, multipsktral görüntü, pankromatik bant, pan keskinleştirme, pan keskinleştirilmiş görüntü, CUDA, Basit Ortalama, Brovey, HSV

CONTENTS

THESIS EXAMINATION RESULT FORM……ii

ETHICAL DECLARATION ……iii

ACKNOWLEDGMENTS……iv

ABSTRACT……v

ÖZ……vi

CONTENTS……vii

NOMENCLATURE……ix

LIST OF TABLES……x

LIST OF FIGURES……xii

CHAPTER 1 – INTRODUCTION……1

1.1 BASIC INFORMATION……1

1.2 DEFINITIONS……2

1.2.1 Multi-Spectral Image……2

1.2.2 Pan Chromatic Image……4

1.2.3 Pan Sharpened Image……5

1.2.4 Simple Mean Algorithm……7

1.2.5 Brovey Transformation……8

1.2.6 Hue-Saturation-Value (HSV) Transformation……8

CHAPTER 2 - DATA COLLECTING AND MATLAB ENVIRONMENT……10

2.1 DATASET……10

2.2 MATLAB TESTS……10

CHAPTER 3 – CUDA IMPLEMENTATION……12

3.1 CUDA ENVIRONMENT……12

3.2 IMPLEMANTATION OF THE ALGORITHMS……13

3.2.1 Programming The Basis Of The Application……...……13
3.2.2 Memory Allocation……...……...……13
3.2.3 Implementation Process Of Algorithms……...……..14

CHAPTER 4 - RESULTS……..…….16

CHAPTER 5 - CONCLUSION…….…….37

REFERENCES……..39

APPENDICES……..40
Appendix A – Simple Mean Algorithm Result Comparison Table.…….……...…40
Appendix B – Brovey Transformation Result Comparison Table...…….……...…41
Appendix C – HSV Transformation Result Comparison Table…...…….……...…42
CURRICULUM VITAE……...43

NOMENCLATURE

Roman Letter Symbols

PS_R	Red color band of the pan-sharpened image
PS_G	Green color band of the pan-sharpened image
PS_B	Blue color band of the pan-sharpened image
MS_R	Red color band of the multi-spectral image
MS_G	Green color band of the multi-spectral image
MS_B	Blue color band of the multi-spectral image
PAN	Panchromatic Image
H	H channel of HSV Transformation
S	S channel of HSV Transformation
H	V channel of HSV Transformation

Greek Letter Symbols

δ	Brovey Transformation Intermediate Value
n	Number of channels of the multi-spectral image

LIST OF TABLES

Table 3.1 Device query for Nvidia GeForce 1050 Ti……...10

Table 4.1 Result comparison table for the image #1……15

Table 4.2 Result comparison table for the image #2……16

Table 4.3 Result comparison table for the image #3……17

Table 4.4 Result comparison table for the image #4……....18

Table 4.5 Result comparison table for the image #5……19

Table 4.6 Result comparison table for the image #6……20

Table 4.7 Result comparison table for the image #7……21

Table 4.8 Result comparison table for the image #8……22

Table 4.9 Result comparison table for the image #9……23

Table 4.10 Result comparison table for the image #10……24

Table 4.11 Result comparison table for the image #11……25

Table 4.12 Result comparison table for the image #12……26

Table 4.13 Result comparison table for the image #13……27

Table 4.14 Result comparison table for the image #14……28

Table 4.15 Result comparison table for the image #15……29

Table 4.16 Result comparison table for the image #16……30

Table 4.17 Result comparison table for the image #17……31

Table 4.18 Result comparison table for the image #18……32

Table 4.19 Result comparison table for the image #19……33

Table 4.20 Result comparison table for the image #20……34

LIST OF FIGURES

Figure 1.1 A figure of the electromagnetic spectrum that shows various properties across the range of frequencies and wavelengths……2

Figure 1.2 An illustration of how an RGB image consists……. ……..4

Figure 1.3 (a) Low-resolution multi-spectral image of the Bosphorus. This image is up-scaled to the same size of pan chromatic image. (b) High-resolution pan chromatic image of the same location of (a). ……...…5

Figure 1.4 Pan sharpened image by using multi-spectral and pan chromatic image in Figure 1.3. Brovey transformation is selected as the algorithm ……. ……..6

Figure 1.5 Pan sharpening process image by using multi-spectral and pan chromatic image in Figure 1.3. Brovey transformation is selected as the algorithm ….……. ……..6

Figure 1.6 Figure that shows how an interpolation upscale a dataset to a given size..……..7

Figure 4.1 Result comparison and the zoomed area for the image #1…….15

Figure 4.2 Result comparison and the zoomed area for the image #2…….16

Figure 4.3 Result comparison and the zoomed area for the image #3…….17

Figure 4.4 Result comparison and the zoomed area for the image #4…….18

Figure 4.5 Result comparison and the zoomed area for the image #5…….19

Figure 4.6 Result comparison and the zoomed area for the image #6…….20

Figure 4.7 Result comparison and the zoomed area for the image #7…….21

Figure 4.8 Result comparison and the zoomed area for the image #8…….22

Figure 4.9 Result comparison and the zoomed area for the image #9…….23

Figure 4.10 Result comparison and the zoomed area for the image #10…….24

Figure 4.11 Result comparison and the zoomed area for the image #11…….25

Figure 4.12 Result comparison and the zoomed area for the image #12…….26

Figure 4.13 Result comparison and the zoomed area for the image #13…….27

Figure 4.14 Result comparison and the zoomed area for the image #14…….28

Figure 4.15 Result comparison and the zoomed area for the image #15…….29

Figure 4.16 Result comparison and the zoomed area for the image #16…….30

Figure 4.17 Result comparison and the zoomed area for the image #17…….31

Figure 4.18 Result comparison and the zoomed area for the image #18…….32

Figure 4.19 Result comparison and the zoomed area for the image #19…….33

Figure 4.20 Result comparison and the zoomed area for the image #20…….34

CHAPTER 1

INTRODUCTION

Engineering as a definition is solving a problem with best, cheapest and efficient way. Thus, motivation of an engineer should origin from this mentality. Whatever an engineer does should help the humanity in such ways. During this study, I learned so many things that I did not know before. Then, I have just try to find a way to improve them.

Visualization of something makes understanding the problem on it easier. We can see the environment that we lived in with a restricted sight and we think this environment is what we saw. Hence, when we try to model any problem, we will stick to the obstacles if we could not cross the borders. Same environment is much different for a bird's eye. However, it is much more different for a view from a satellite. It is possible to see anywhere throughout a satellite with a couple of clicks. Nevertheless, what you see is –most probably- combination of images that processed multiple times. Actually, in theoretical, it may be possible to obtain a satellite image that can be directly used. However, in reality, because getting a high-resolution multi-spectral image –RGB in that case- is an expensive process, an artificial high-resolution multi-spectral image that is an output of high-resolution pan chromatic image and low-resolution multi-spectral is being used. These images are called as pan-sharped images. In other words, this process can be defined as extracting color information from multi-spectral image and combine it with a pan chromatic image that has detail information.

1.1 BASIC INFORMATION

Processing of satellite images is not a new topic. So many commercial product process the satellite images or serve the processed images. This study focuses on the pan-sharpening phase of the whole process. Twenty images from different sources is collected and three pan-sharpening algorithms is used. Yet, the main motivation of this study is whether using parallel programing is useful or not. Satellite images are mostly have high dimensions. Parallel programming methods is more beneficial if the data is bigger. As parallel programing environment CUDA that is trademark of NVIDIA is

selected. CUDA helps the developers to direct access to GPU. It works with C, C++ and Fortran. It was introduced as an acronym for Compute Unified Device Architecture.

1.2 DEFINITIONS

1.2.1 Multi-Spectral Image

A multi-spectral image is an image that has data within specific wavelength ranges throughout the electromagnetic spectrum given in Figure 1.1 [1].

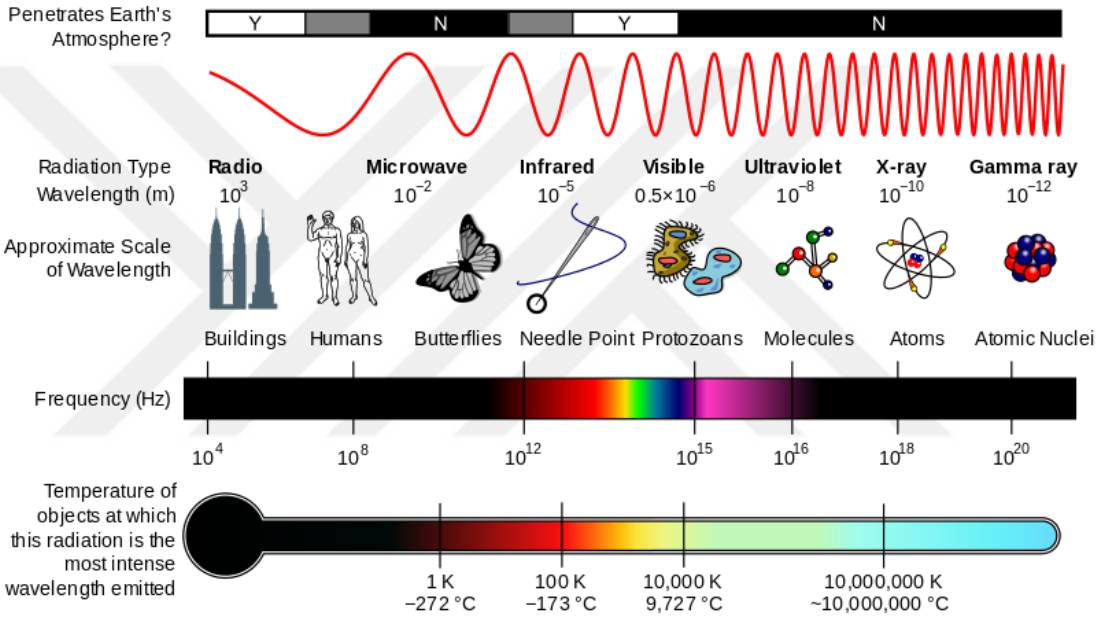


Figure 1.1 A figure of the electromagnetic spectrum that shows various properties across the range of frequencies and wavelengths

Electromagnetic waves are ordered in electromagnetic spectrum and this order is continuous. There is no gap between the waves. Naturally, because there is an inverse ratio between wavelength and the frequency, the waves are also ordered in frequency. In the spectrum, if wavelength increases the frequency decreases. Electromagnetic spectrum is divided into seven parts [2].

Radio wave has at least 1 millimeter wavelengths. It has the largest wavelength in the spectrum so it has the smallest frequency. It can be generated electrical vibration and can be found anywhere in the universe.

Microwave has wavelength between 1 millimeter and 1 meter. It can be used in radars, mobile phones, wireless internet communication and microwave ovens.

Infrared is a radiation that has 1 millimeter and 710 nanometers wavelengths and it can be produced by all materials that has some temperature. A healthy human with a 37 °C body temperature can emit infrared radiation with 10-micrometer wavelengths per second.

Visible light is the first discovered and the only part of the electromagnetic spectrum that can be detected by human eye. This part consists of wavelength between 400 millimeters and 700 millimeters and it is the narrowest part of the spectrum. All wavelengths in this gap are detected as different colors by human eye. Red color has the largest wavelength and the purple has the smallest wavelength. The sun is a natural source for the visible lights.

Ultraviolet radiation has a value of wavelengths between 10 nanometers and 400 nanometers. Although it cannot be seen by human eye, some animals can detect it.

X-Ray radiation has wavelengths between 0.01 nanometer and 10 nanometers. It can be produced by slowing of high-speed atoms or the transitions between the electron orbits within the atom. It can be harmful for creatures because it is a high-energy radiation.

Gamma ray is a value of wavelengths less than 0.01 nanometer. It is the most energetic part of the spectrum. It is based directly on events in the atomic nucleus. It has destructive effect on creatures because it has high-energy.

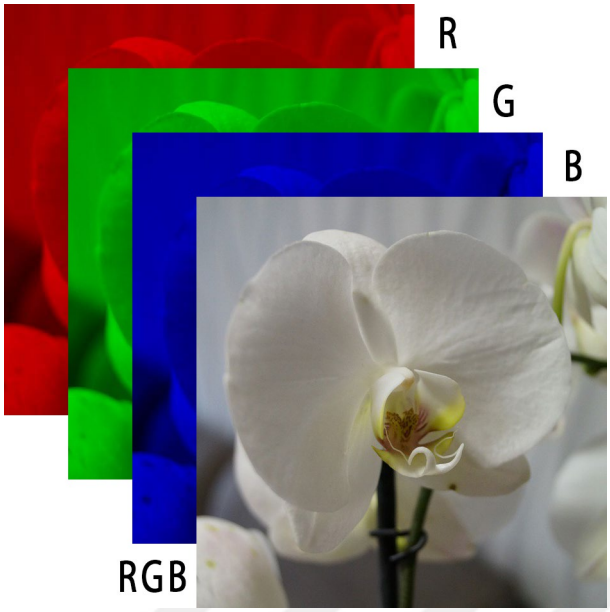


Figure 1.2 An illustration of how an RGB image consists. R is the red channel, G is the green channel, B is the blue channel of the image and RGB is the image itself.

RGB color space is a kind of multi-spectral image that consist of red, green and blue color band as showed in Figure 1.2. This color space is being used by computers, graphic cards and monitors. An RGB image is actually is a three dimensional matrices. Each dimension refers to corresponding color space. Hence, every pixel in an image has three values for each channel. The combination of these color information makes pixel colorful for us.

1.2.2 Pan Chromatic Image

Pan chromatic image, unlike multi-spectral image, has just one wide band. It looks like a grayscale image. Pan chromatic image of an area has much more details than same size multi-spectral image. It is cheaper than get a high-resolution multi-spectral image. Hence, by using high-resolution pan chromatic image and low-resolution multi-spectral image it is possible to achieve an artificial high-resolution multi-spectral image. This process is called as pan sharpening and the output image is called as pan-sharpened image.

1.2.3 Pan Sharpened Image

A satellite has a limited physical area and it costs too much. Hence, there are some limitations when designing it. A multi-spectral sensor can detect multiple bands but mostly not in higher spatial resolution. Actually, a satellite can be collect high-resolution multi-spectral data but such a sensor costs too much and may need more physical area. However, getting such a data can be sacrificed relying on fusion methods. Pan chromatic sensor collects just single but wide band. Therefore, it is cheaper and take less space.



(a) (b)

Figure 1.3 (a) Low-resolution multi-spectral image of the Bosphorus. This image is up-scaled to the same size of pan chromatic image. (b) High-resolution pan chromatic image of the same location of (a).

Pan sharpened image is the result of all sharpening process. It is produced by a fusion of low-resolution multi-spectral image showed in Figure 1.3.a and high-resolution pan chromatic image showed in Figure 1.3.b. Depending on the method that is used, it is possible to get a high-resolution and multi-spectral result showed Figure 1.4 just as the satellite may give. It is obvious that this process in not a real time result. However, the algorithm that is selected and system configuration where process is done can reduce to process time almost to real time process. The result of this process showed in Figure 1.5 is often quite acceptable.



Figure 1.4 Pan sharpened image by using multi-spectral and pan chromatic image in Figure 1.3. Brovey transformation is selected as the algorithm.

In this study, Simple Mean, Brovey Transformation and HSV Transformation algorithms are used as pan sharpening methods.

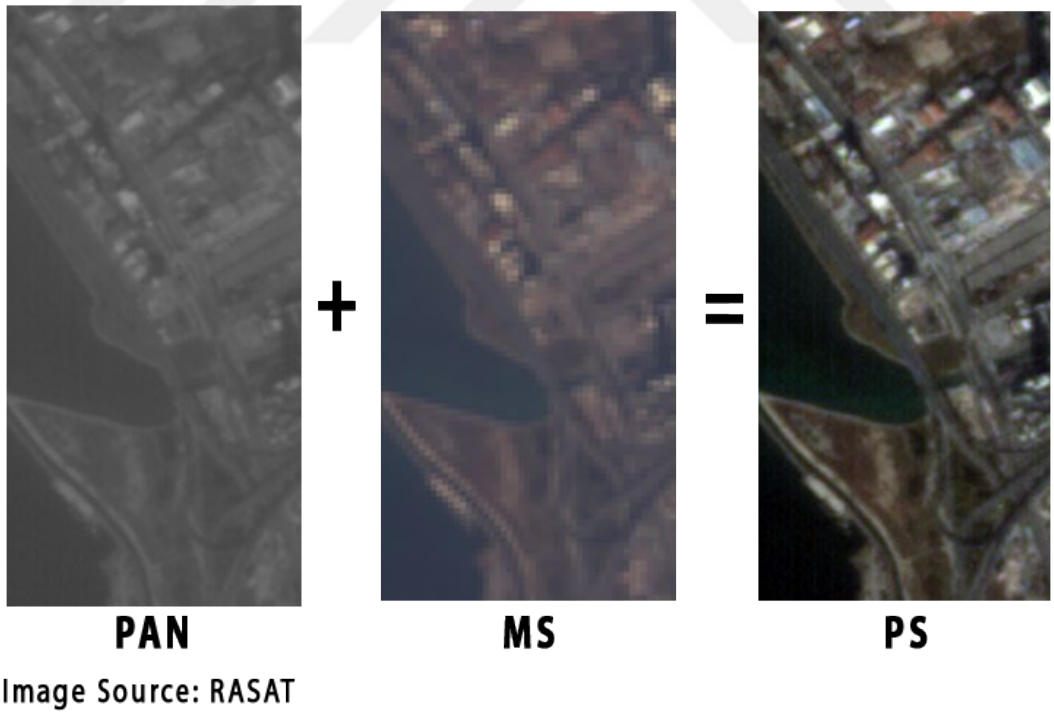


Figure 1.5 Pan sharpening process image by using multi-spectral and pan chromatic image in Figure 1.3. Brovey transformation is selected as the algorithm.

1.2.4 Simple Mean Algorithm

Simple mean algorithm is just a simple mathematical mean calculation of every pixel of multi-spectral image with related pixel of pan chromatic image. Each color channel of multi-spectral image is calculated with pan chromatic image separately and then they should be combined again to get output image.

$$PS_R = \frac{1}{2} * (MS_R + PAN) \qquad (\mathbf{1.1})$$

$$PS_G = \frac{1}{2} * (MS_G + PAN) \qquad (\mathbf{1.2})$$

$$PS_B = \frac{1}{2} * (MS_B + PAN) \qquad (\mathbf{1.3})$$

where PS_i is the color channel of pan sharpened image, MS_i is the color channel of multi spectral image and PAN is the pan chromatic image.

Not only simple mean algorithm but also the other algorithms are generally matrix operations. Thus, all matrices, image data in this case, should have same dimensions. Multi-spectral image has a lower dimension than pan chromatic image. Therefore, an interpolation process should have done just before the operations to make all matrices have same size figured in Figre 1.6. The way of interpolation may change the quality of the result. Throughout this paper, nearest neighborhood interpolation have been used.

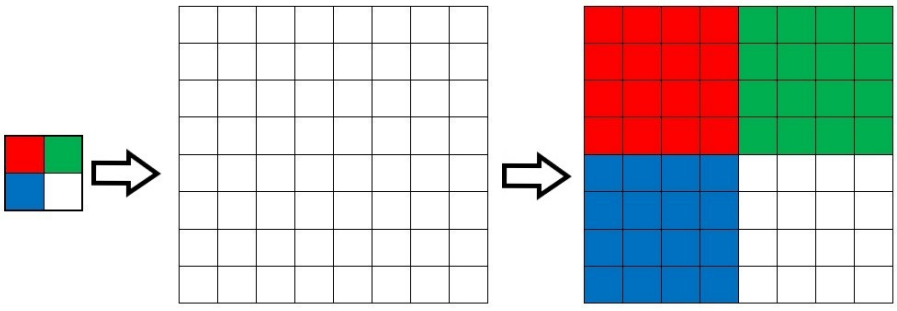


Figure 1.6 Figure that shows how an interpolation upscale a dataset to a given size.

1.2.5 Brovey Transformation

Brovey Transformation highlights sharpness. First, as an intermediate value, the multiplication of pan chromatic image and the number of channels of the multi-spectral image is calculated. After that, every channel of multi-spectral image is multiplied by this intermediate value and divided into summation of multi-spectral image channel values. Obtained result gives use the related channel of the pan-sharpened image.

$$\delta = \frac{n * PAN}{MS_R + MS_G + MS_B} \quad (\mathbf{1.4})$$

$$PS_R = \delta * MS_R \quad (\mathbf{1.5})$$

$$PS_G = \delta * MS_G \quad (\mathbf{1.6})$$

$$PS_B = \delta * MS_B \quad (\mathbf{1.7})$$

where n is the number of channels of the multi-spectral image (3 in this case), PS_i is the color channel of pan sharpened image and MS_i is the color channel of multi spectral image.

1.2.6 Hue-Saturation-Value (HSV) Transformation

HSV Transformation needs the multi-spectral image's hue, saturation and value (intensity) information to work. Therefore, a RGB to HSV transformation is applied to multi-spectral image. In HSV domain, value information (V) is replaced with pan-chromatic image and then a HSV to RGB transformation is applied. The output is pan sharpened image.

$$m_{max} = \max(R, G, B) \quad (\mathbf{1.8})$$

$$m_{min} = \min(R, G, B) \quad (\mathbf{1.9})$$

$$\Delta = max - min \quad (\mathbf{1.10})$$

$$H = \begin{cases} 0°, if\ \Delta = 0 \\ 60° * \left(\frac{MS_G - MS_B}{\Delta} mod 6\right), if\ m_{max} = MS_R \\ 60° * \left(\frac{MS_B - MS_R}{\Delta} + 2\right), if\ m_{max} = MS_G \\ 60° * \left(\frac{MS_R - MS_G}{\Delta} + 4\right), if\ m_{max} = MS_B \end{cases} \quad (\mathbf{1.11})$$

$$S = \begin{cases} 0, if\ m_{max} = 0 \\ \frac{\Delta}{m_{max}}, if\ m_{max} \neq 0 \end{cases} \quad (\mathbf{1.12})$$

$$V = m_{max} \quad (\mathbf{1.13})$$

After RGB-HSV transformation is done, V is interchange with PAN value. Then, HSV-RGB transformation is need.

$$C = V * S \quad (\mathbf{1.14})$$

$$X = C * \left(1 - \left|\left(\frac{H}{60°}\right) mod 2 - 1\right|\right) \quad (\mathbf{1.15})$$

$$m = V - C \quad (\mathbf{1.16})$$

$$(Red', Green', Blue') = \begin{cases} (C, X, 0), & 0° \leq H < 60° \\ (X, C, 0), & 60° \leq H < 120° \\ (0, C, X), & 120° \leq H < 180° \\ (0, X, C), & 180° \leq H < 240° \\ (X, 0, C), & 240° \leq H < 300° \\ (C, 0, X), & 300° \leq H < 360° \end{cases} \quad (\mathbf{1.17})$$

$$PS_R = (Red' + m) * 255 \quad (\mathbf{1.18})$$

$$PS_G = (Green' + m) * 255 \quad (\mathbf{1.19})$$

$$PS_B = (Blue' + m) * 255 \quad (\mathbf{1.20})$$

where PS_i is the color channel of pan sharpened image.

CHAPTER 2

DATA COLLECTING AND MATLAB ENVIRONMENT

2.1 DATASET

In this study, I used twenty images from different sources. The dataset contains different image size from 256x256 to 15201x15221. Most of the images are from GOKTURK-2 satellite, which is a discovery satellite that is developed by TUBITAK and TAI. It launched at 18 December 2012 in China. The 409 kg satellite has 2,5 meter black/white and 5 meter colorful resolution.

Other images are collected from work of Jamel Saeedi and Kerim Faez titled as “A new pan-sharpening method using multi-objective particle swarm optimization and the shiftable contourlet transform” and work of Hamid Reza Shahdoosti and Hassan Ghassemian titled as “Matlab Code and MS and PAN Datasets”.

2.2 MATLAB TESTS

Before the implementation of the algorithms in CUDA, I need to learn and test the algorithms in a safe environment. Hence, I chose MATLAB environment to make tests. During these tests, my focus was determining how the algorithms works, what they need and how they can implement in CUDA. As I chose C++ for implement the algorithms in CUDA, and MATLAB uses same syntax it would be easy to transfer my code to CUDA environment.

The first problem that I faced was images that had different dimensions. Images were multi-spectral RGB images but some of them had a fourth band. This band contains some extra information about the image but I do not need that. Therefore, I had to optimize all multi-spectral images to 3-channel RGB image.

The next problem was about spatial dimensions of pan chromatic images and multi-spectral images. Images will be represent as matrices. Because there are some matrix operations in the algorithms, all image have to be same matrix dimensions. To do that,

an interpolation should be applied. All multi-spectral images that have not same size with related pan chromatic image was resized using nearest neighborhood interpolation. This process was done while the dataset was prepared. All images for every set was set to same dimension. In MATLAB tests, the other methods for interpolation have been tested but in CPU and GPU environment, because the images are ready and have same dimension size, any additional or different interpolation method has not been applied.

The other problem was bit depth. Most images have 24-bit-depth. However, some of them have bit-depth more than 24. Thus, some calculations need to be done on these images to decrease bit-depth. Because of that, I linearly decreased bit-depth of the images that had more than 24-bit-depth.

When the tests on the MATLAB environment was done, I were ready to start CUDA implementation.

CHAPTER 3

CUDA IMPLEMENTATION

3.1 CUDA ENVIRONMENT

Throughout the study, I used 10.1 CUDA Driver Version and Runtime Version on a Nvidia GeForce GTX 1050 Ti. CUDA capability Major/Minor version number is 6.1. Total amount of global memory is 4096 Mbytes. It has 6 multiprocessors (MP) and 128 CUDA cores/MP, so there is 768 CUDA Cores. Some other important information is given as Table 3.1.

Table 3.1 Device query for Nvidia GeForce 1050 Ti

Device 0: "GeForce GTX 1050 Ti"	
GPU Max Clock rate:	1392 MHz (1.39 GHz)
Memory Clock rate:	3504 Mhz
Maximum Texture Dimension Size (x,y,z)	1D=(131072), 2D=(131072, 65536), 3D=(16384, 16384, 16384)
Maximum Layered 1D Texture Size, (num) layers	1D=(32768), 2048 layers
Maximum Layered 2D Texture Size, (num) layers	2D=(32768, 32768), 2048 layers
Total amount of constant memory:	65536 bytes
Total amount of shared memory per block:	49152 bytes
Total number of registers available per block	65536
Warp size:	32
Maximum number of threads per multiprocessor:	2048
Maximum number of threads per block:	1024
Max dimension size of a thread block (x,y,z):	(1024, 1024, 64)
Max dimension size of a grid size (x,y,z):	(2147483647, 65535, 65535)

3.2 IMPLENTATION OF THE ALGORITHMS

3.2.1 Programming the Basis of the Application

I used Microsoft Visual Studio 2017 Community Edition as development environment and selected C++ programming language to code. All computations in CPU and GPU environment have done with C++ programming language and no additional library/framework etc. was used. The results depend on C++ code and MATLAB results are not considered or compared. In CPU environment only one thread was used for process. The application should work with both CPU and GPU to measure speed-up performance. Because I focused on only CUDA implementation, I did not include any other library to the project. This caused to a requirement for images to be saved as mathematical models. Hence, I prepared all dataset as txt format for easy to read by the application. All multi-spectral images were split into R-G-B channels and with all corresponding pan chromatic images, they were saved in different folders.

First, in the application I prepared a welcoming screen that actually checks whether the data files are exist and read them if exist. The application needs msr.txt, msg.txt, msb.txt and pan.txt files to run the code in both CPU and GPU.

The next phase is selecting the algorithm. There are six option to select;

1) Simple Mean (GPU)
2) Brovey (CUDA)
3) HSV (CUDA)
4) Simple Mean (CPU)
5) Brovey (CPU)
6) HSV (CPU)

According to the selection of user, the application calls related function.

3.2.2 Memory Allocation

CPU part of all algorithms are smoothly. However, in GPU you need to be careful about some general concepts. Although a variable in CPU can be reached from GPU,

a variable in GPU cannot be reachable by CPU. Hence, a connection between variables should be constructed. Just before calling the kernel in CUDA, a function is called as kernel in CUDA environment, all variable should be allocated in GPU.

In all kernels there are 7 parameters and all of these are defined variable in CUDA platform. All data for an image should be transferred and defined in GPU. To do that, memory allocation with data type, data size and size of the data type is need. All values in data are integer, so what I need is size of each dimension. Each color channel is represented as a 2D array, and the size of the array is the product of the image width and the height of the image.

Connection between variables in CPU and GPU can be made some defined function. Actually it is a copy process. Before and after the calculation same function, cudaMemcpy, is used to copy values between CPU and GPU by only changing the last parameter. This parameter is either cudaMemcpyHostToDevice or cudaMemcpyDeviceToHost. When all variables are ready, it is possible to calculate output by using memory address of variables as parameters in kernel.

However, a critical problem is determining number of blocks and threads per block in CUDA. This value changes with respect to device that is used. Therefore, the limit of device is key point for this determination. I did not want to change number of blocks and threads per block for all images. Hence, I prepared a code block that calculates number of blocks and threads per block using the warp size of the device, which is 32.

3.2.3 Implementation Process of Algorithms

Simple Mean algorithm is easy to code. That is why I did not faced so much problem. All you need is two nested FOR loop. All pixel values in each color band is added to corresponding pan chromatic pixel value and then is divided into two to get output for this pixel.

Brovey Transformation needs more attention from simple mean algorithm. In CPU calculation, they are very similar. There is only one intermediate value as a multiplier. This multiplier, using pan chromatic value, gives weighted approach of RGB value.

HSV Transformation is relatively hard algorithm. First, a RGB to HSV transformation is need. In HSV domain, Value (some sources indicate that as Intensity) will be replaced with PAN value. After that, an inverse transformation from HSV to RGB gives us the pan sharpened image. RGB-HSV-RGB transformation requires some mathematical process but they are not extremely hard calculations.

In all process, there are two clocks. One is for elapsed time in CPU and the other one is for elapsed time in GPU. In CPU, I have used high_resolution_clock::now() function from chrono library just before and after the function. Difference between them gives the elapsed time in CPU. In GPU, there is a built-in variable as cudaEvent_t and also there are some built-in functions like cudaEventCreate, cudaEventRecord, cudaEventSynchronize and cudaEventElapsedTime to calculate elapsed time in GPU.

In any case, after the algorithm is called and the process is done, a function is called to write results to an output file. Color channels for pan chromatic image is written to files according to their channel information. In any error case or at the final, all variables in GPU and CPU is be set to free from memory.

CHAPTER 4

RESULTS

In the experiments, I have observed there is a remarkable speed up ratio that depends on the image and algorithm. However, the speed up is not only the criteria to choose. Because, although for the same image, some algorithms have a good speed-up ratio, they last much. Though the process takes seconds in simple mean and Brovey transformation, HSV transformation takes hundreds of seconds. This difference decreases when the image size increases. Especially the images that have size greater than 4000x4000, HSV transformation makes better result. Simple mean algorithm takes few seconds but the output is not remarkable. Brovey transformation has much better results respected to the simple mean. If the data has a size below 4000x4000 pixels, Brovey transformation is good to choose. However, when data size is greater than 4000x4000, it is logical to choose HSV transformation.

The elapsed time information that is collected in tests includes memory allocation in graphic card, memory copy from host to device, computations, memory copy from device to host. Writing the results to the txt files are not added to result time.

The following tables and images show the results that I obtained for every image and how the output image looks like.

Table 4.1 Result comparison table for the image #1

Size = 256x256	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	1.27000	0.59389	2.14
Brovey Transformation	1.36910	**0.53175**	**2.57**
HSV Transformation	279.83570	124.67000	2.24

This image retrieved from Saeedi and Faez works [3]. For this image, best algorithm is Brovey transformation according to Table 4.1. Figure 4.1 shows how the output looks like using Brovey transformation.

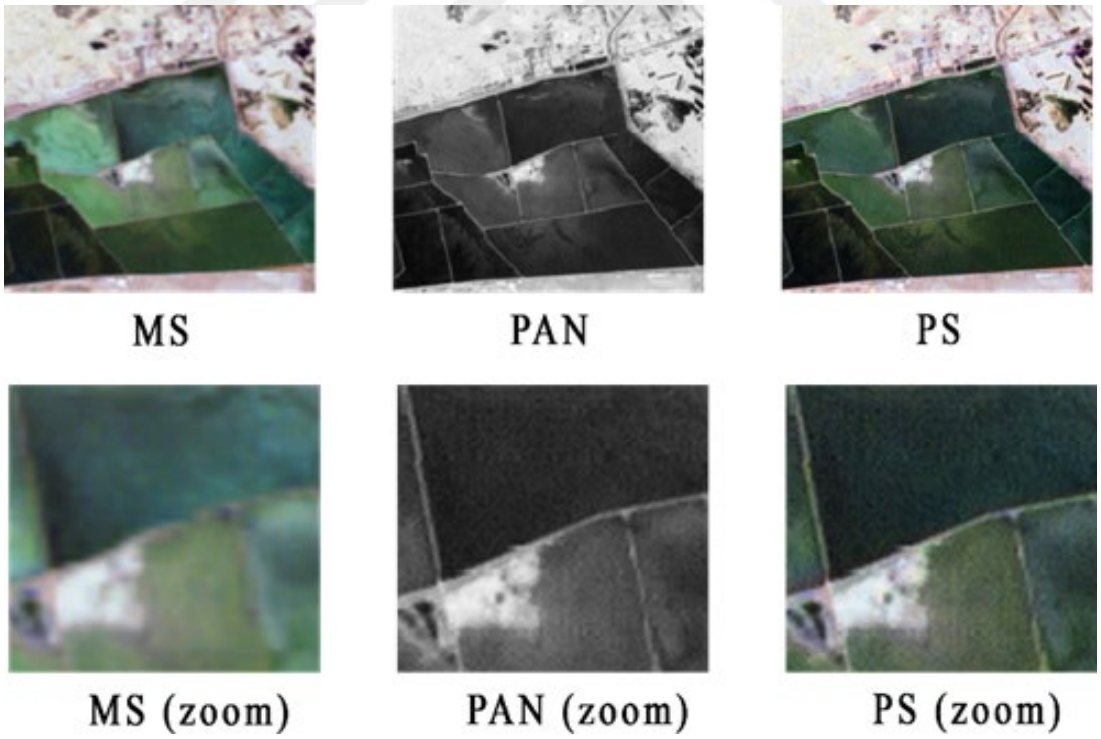


Figure 4.1 Result comparison and the zoomed area for the image #1

Table 4.2 Result comparison table for the image #2

Size = 512x512	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	4.73830	**2.09921**	2.26
Brovey Transformation	5.48350	2.21766	**2.47**
HSV Transformation	366.19170	150.63000	2.43

This image retrieved from Saeedi and Faez works [3]. For this image, the best time is in Simple Mean algorithm but Brovey Transformation hast the best speed up ratio according to Table 4.2. Figure 4.2. shows how the output looks like using Brovey transformation.

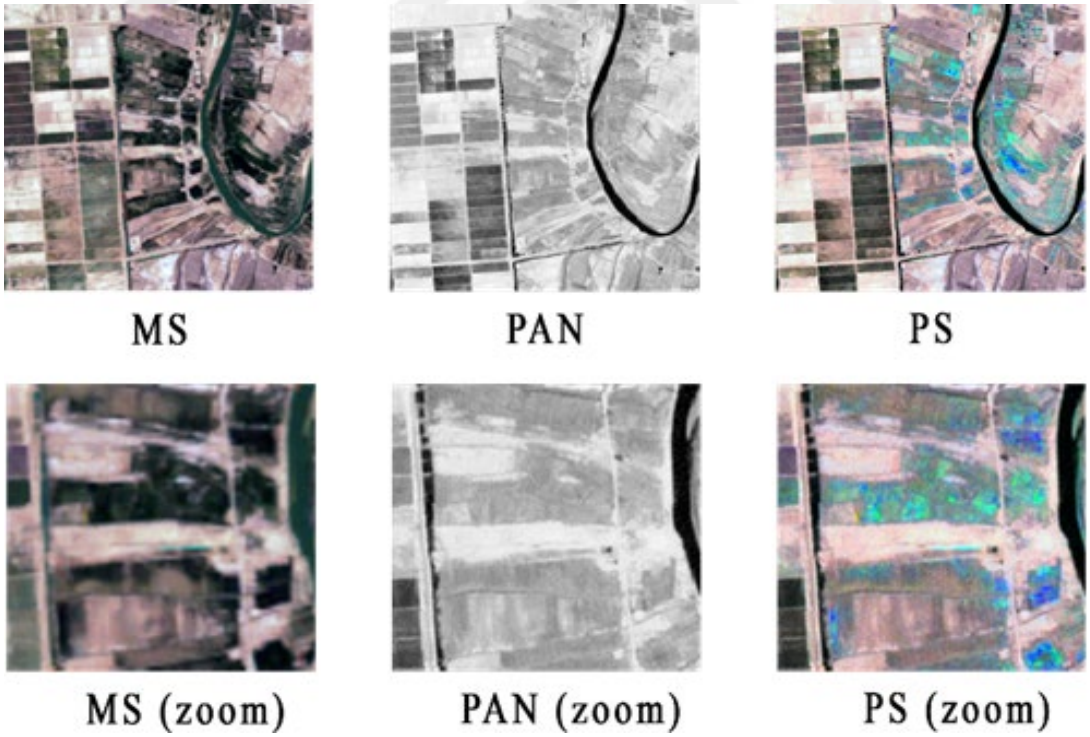


Figure 4.2 Result comparison and the zoomed area for the image #2

Table 4.3 Result comparison table for the image #3

Size = 512x512	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	4.75820	2.21099	2.15
Brovey Transformation	5.82830	**2.11248**	2.76
HSV Transformation	579.71690	150.24641	**3.86**

This image retrieved from Saeedi and Faez works [3]. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.3. Figure 4.3 shows how the output looks like using Brovey transformation.

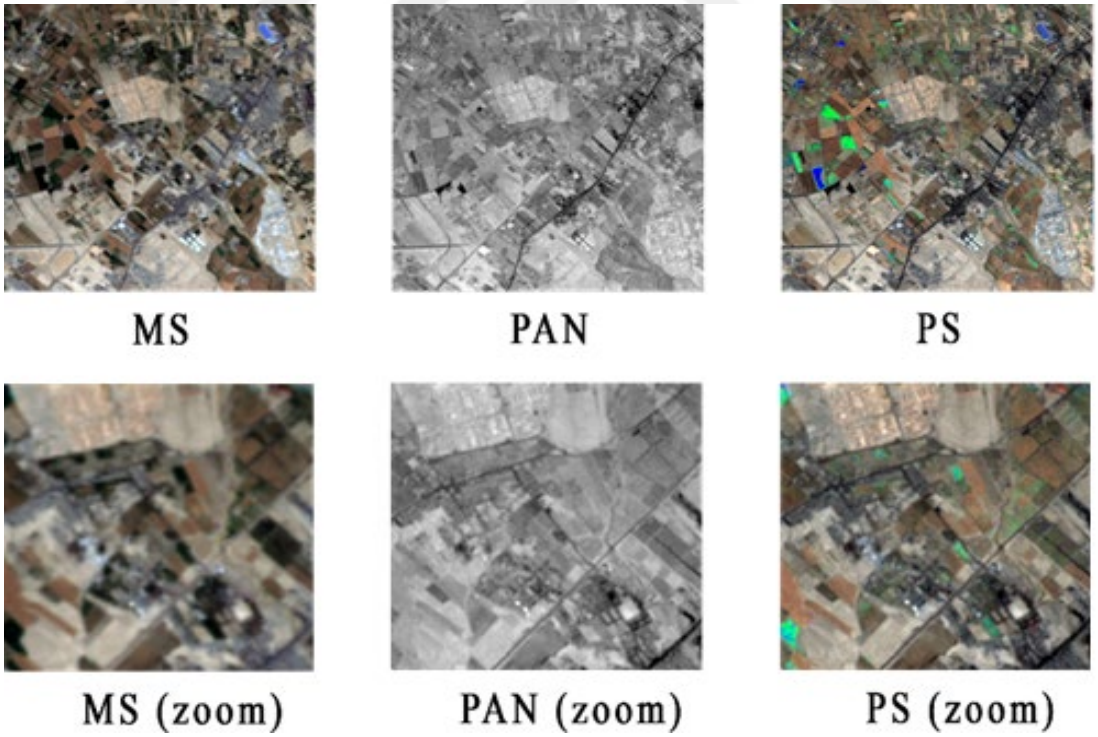


Figure 4.3 Result comparison and the zoomed area for the image #3

Table 4.4 Result comparison table for the image #4

Size = 512x512	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	4.01150	2.36701	1.69
Brovey Transformation	6.64290	**2.14449**	3.10
HSV Transformation	545.80690	140.15000	**3.89**

This image retrieved from Saeedi and Faez works [3]. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.4. Figure 4.4 shows how the output looks like using Brovey transformation.

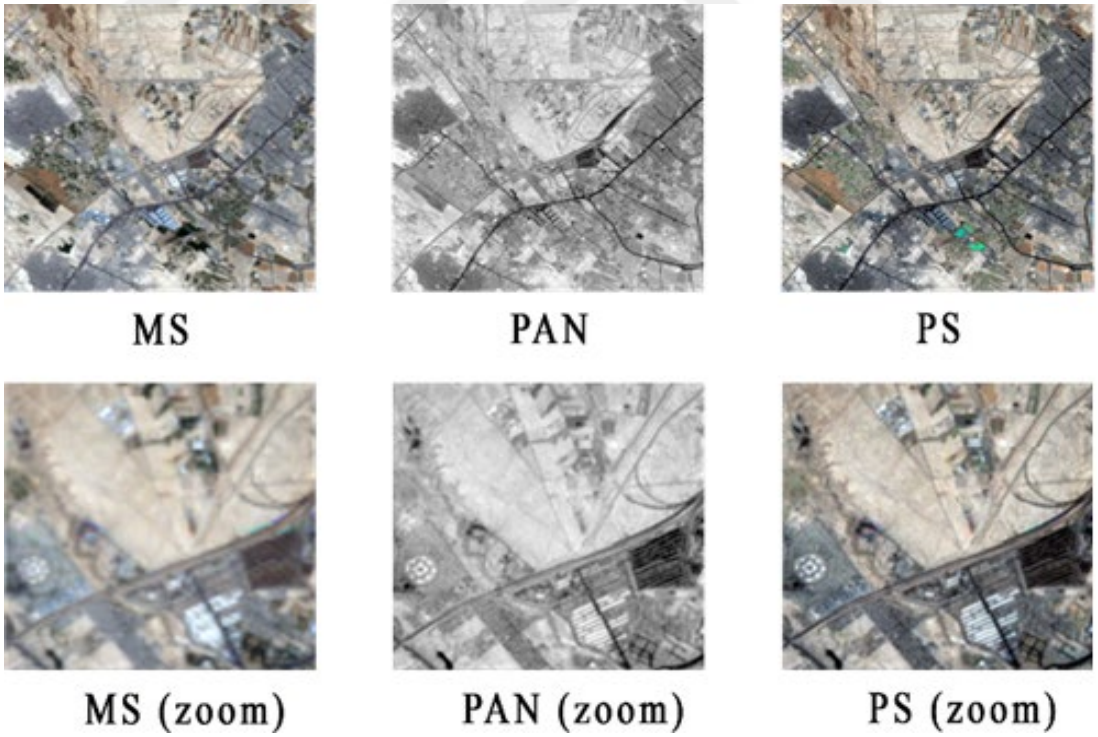


Figure 4.4 Result comparison and the zoomed area for the image #4

Table 4.5 Result comparison table for the image #5

Size = 512x512	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	4.58500	2.21893	2.07
Brovey Transformation	5.27780	**2.21542**	2.38
HSV Transformation	548.29650	140.04541	**3.92**

This image retrieved from Saeedi and Faez works [3]. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.5. Figure 4.5 shows how the output looks like using Brovey transformation.

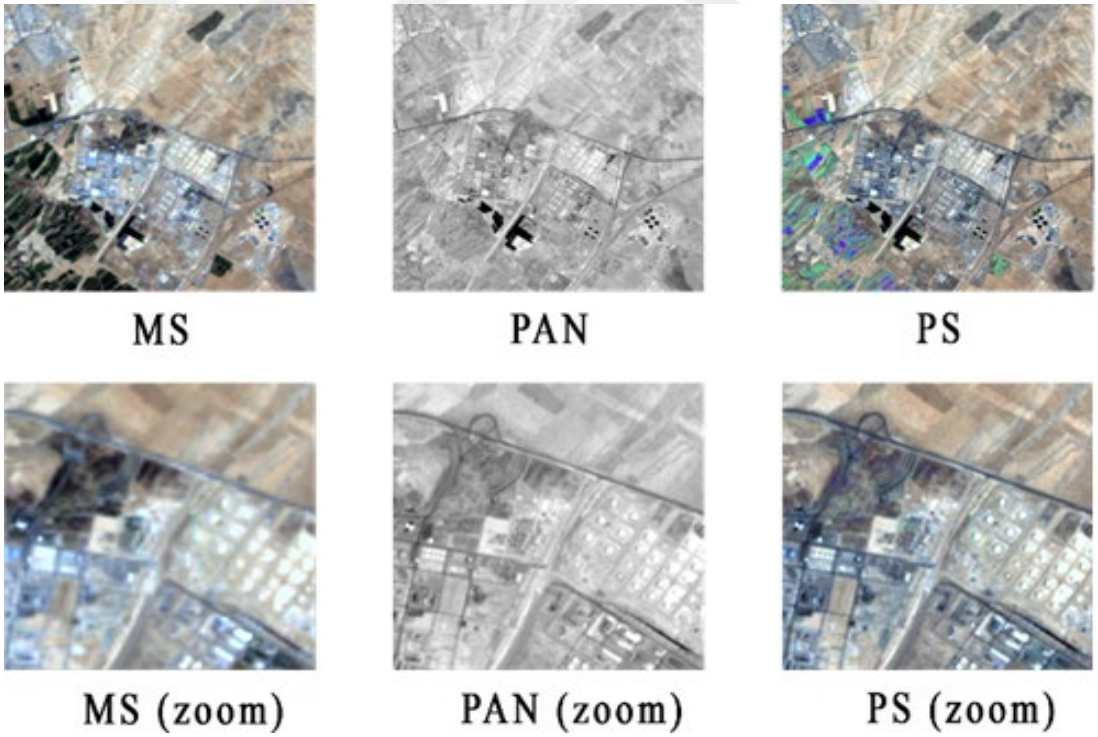


Figure 4.5 Result comparison and the zoomed area for the image #5

Table 4.6 Result comparison table for the image #6

Size = 512x512	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	4.99000	**2.29815**	2.17
Brovey Transformation	5.21560	2.37642	2.19
HSV Transformation	614.97230	139.97521	**4.39**

This image retrieved from Saeedi and Faez works [3]. For this image, the best time is in Simple Mean algorithm but HSV Transformation hast the best speed up ratio according to Table 4.6. Figure 4.6 shows how the output looks like using Brovey transformation.

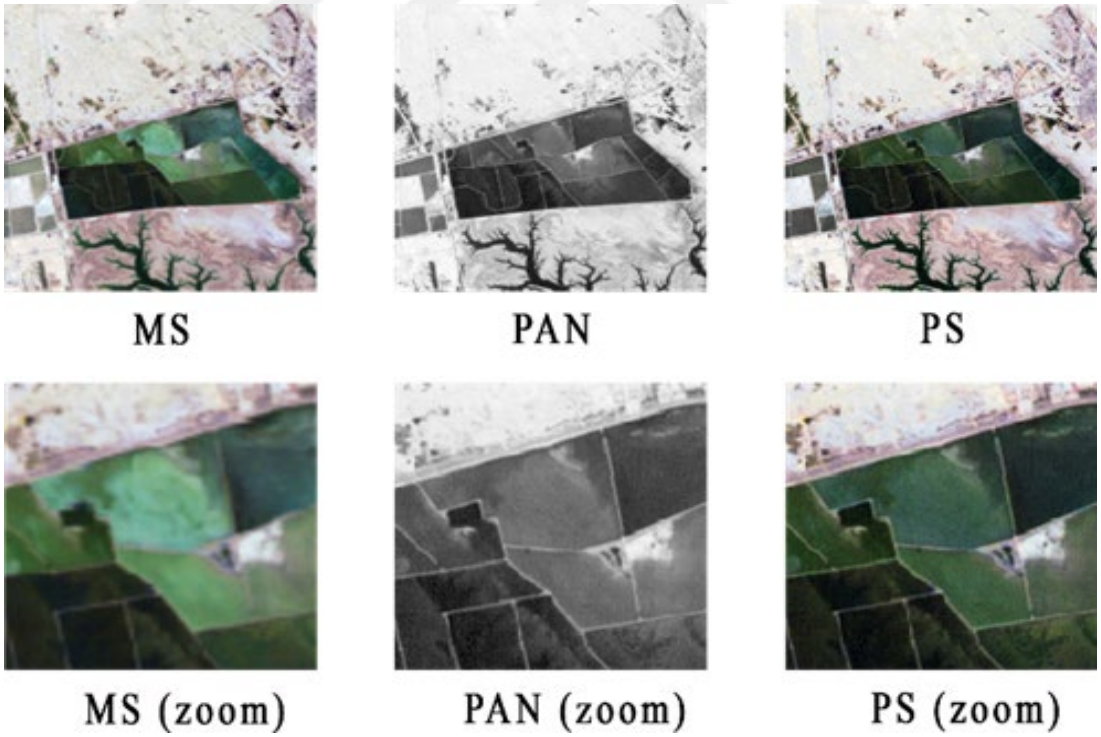


Figure 4.6 Result comparison and the zoomed area for the image #6

Table 4.7 Result comparison table for the image #7

Size = 512x512	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	4.68470	**2.21654**	2.11
Brovey Transformation	5.62640	2.22351	2.53
HSV Transformation	624.09440	151.55215	**4.12**

This image retrieved from Saeedi and Faez works [3]. For this image, the best time is in Simple Mean algorithm but HSV Transformation hast the best speed up ratio according to Table 4.7. Figure 4.7 shows how the output looks like using Brovey transformation.

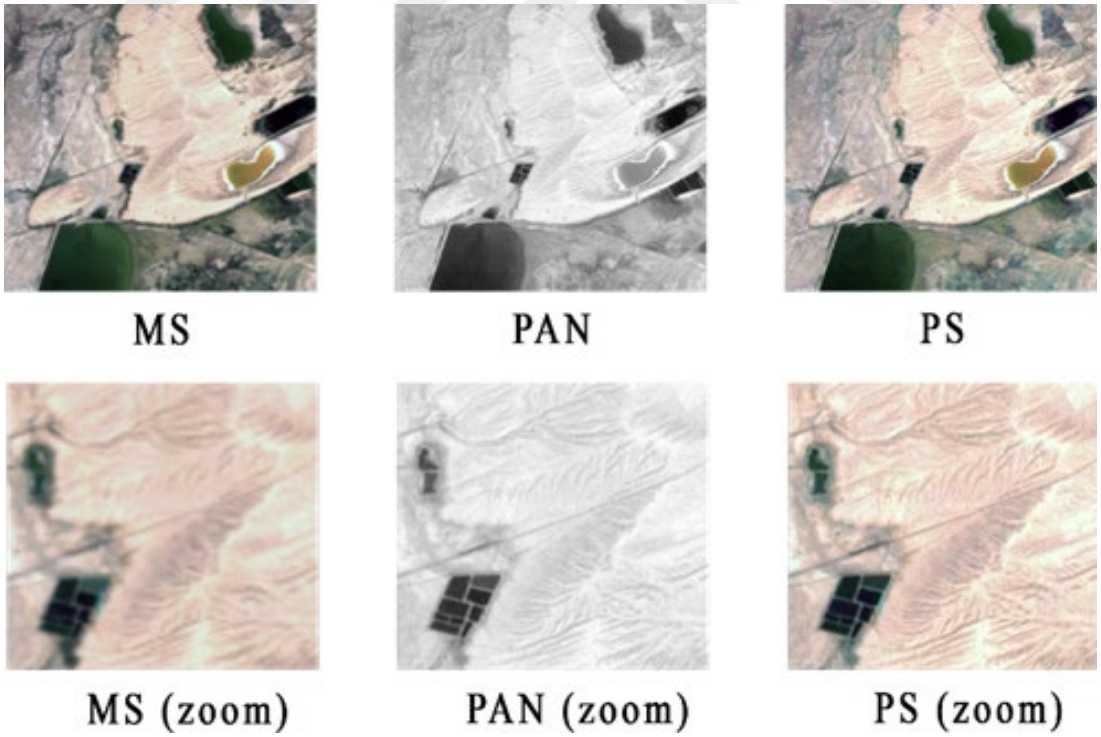


Figure 4.7 Result comparison and the zoomed area for the image #7

Table 4.8 Result comparison table for the image #8

Size = 512x512	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	4.70330	2.32190	2.03
Brovey Transformation	5.39140	**2.15412**	2.50
HSV Transformation	703.93630	145.20663	**4.85**

This image retrieved from Saeedi and Faez works [3]. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.8. Figure 4.8 shows how the output looks like using Brovey transformation.

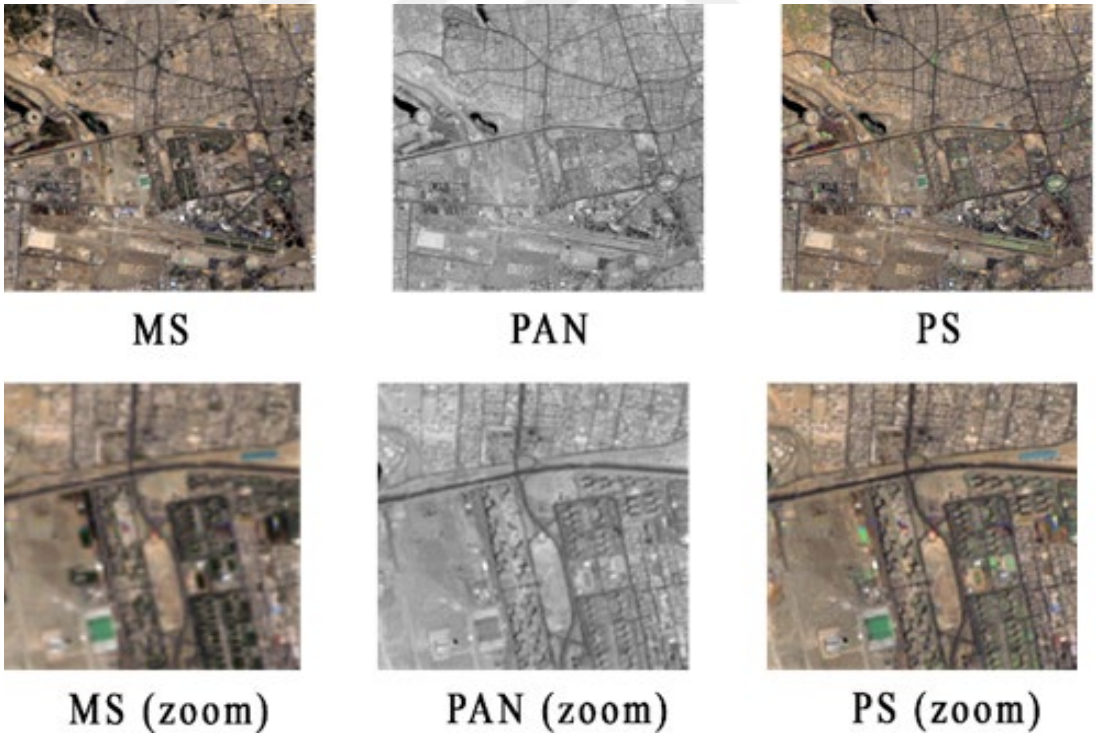


Figure 4.8 Result comparison and the zoomed area for the image #8

Table 4.9 Result comparison table for the image #9

Size = 512x512	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	4.21220	2.24527	1.88
Brovey Transformation	5.25600	**2.14957**	2.45
HSV Transformation	585.08130	148.26783	**3.95**

This image retrieved from Shahdoosti and Ghassemian works [4]. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.9. Figure 4.9 shows how the output looks like using Brovey transformation.

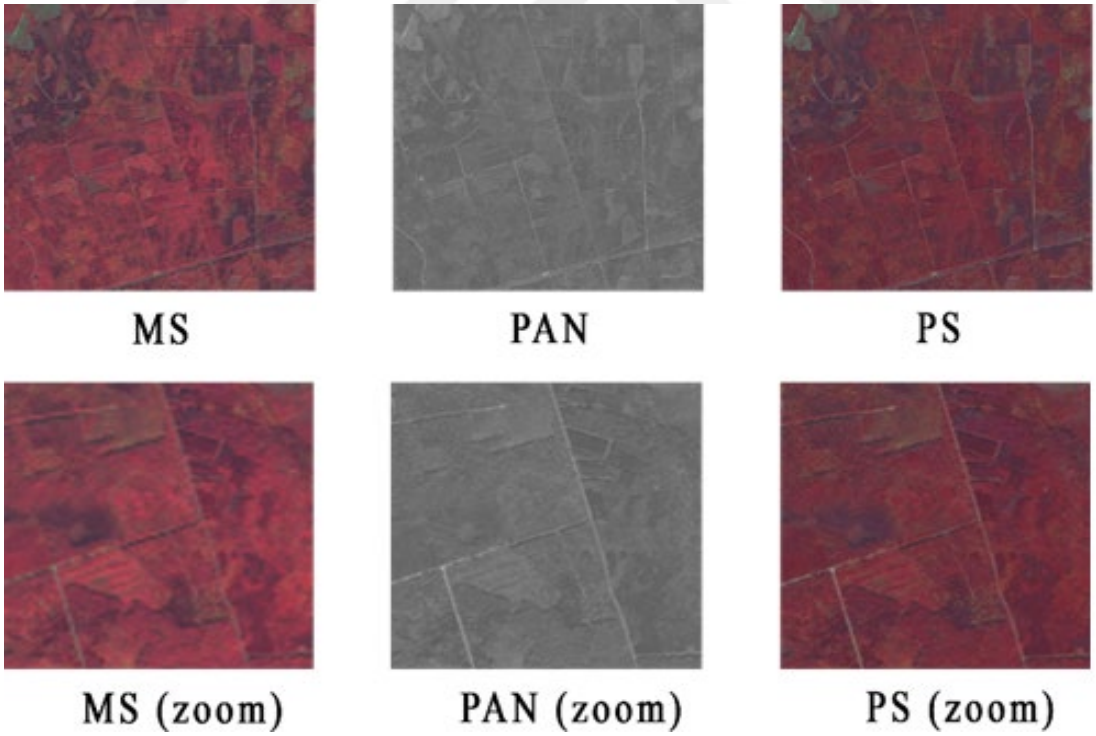


Figure 4.9 Result comparison and the zoomed area for the image #9

Table 4.10 Result comparison table for the image #10

Size = 700x700	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	7.03390	4.21299	1.67
Brovey Transformation	8.89950	**4.01842**	2.21
HSV Transformation	645.62400	139.38522	**4.63**

This image retrieved from Shahdoosti and Ghassemian works [4]. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.10. Figure 4.10 shows how the output looks like using Brovey transformation.

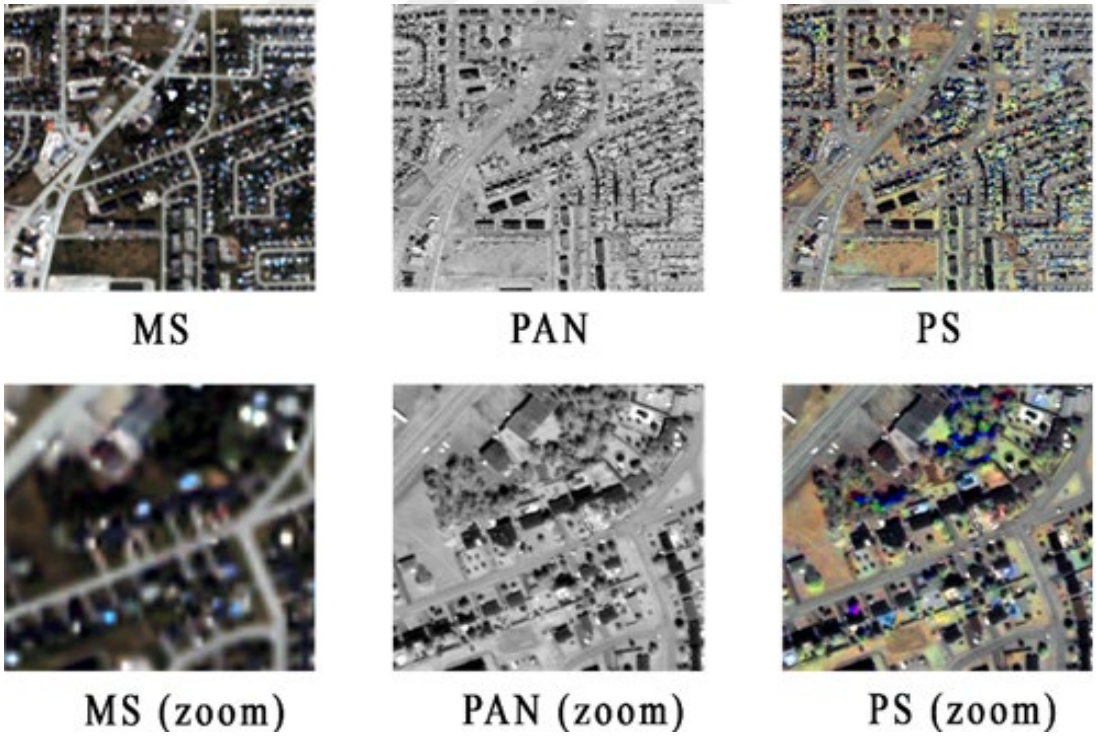


Figure 4.10 Result comparison and the zoomed area for the image #10

Table 4.11 Result comparison table for the image #11

Size = 400x399	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	2.21200	1.39560	1.58
Brovey Transformation	2.66970	**1.36771**	1.95
HSV Transformation	402.42000	138.22430	**2.91**

This image retrieved from Shahdoosti and Ghassemian works [4]. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.11. Figure 4.11 shows how the output looks like using Brovey transformation.

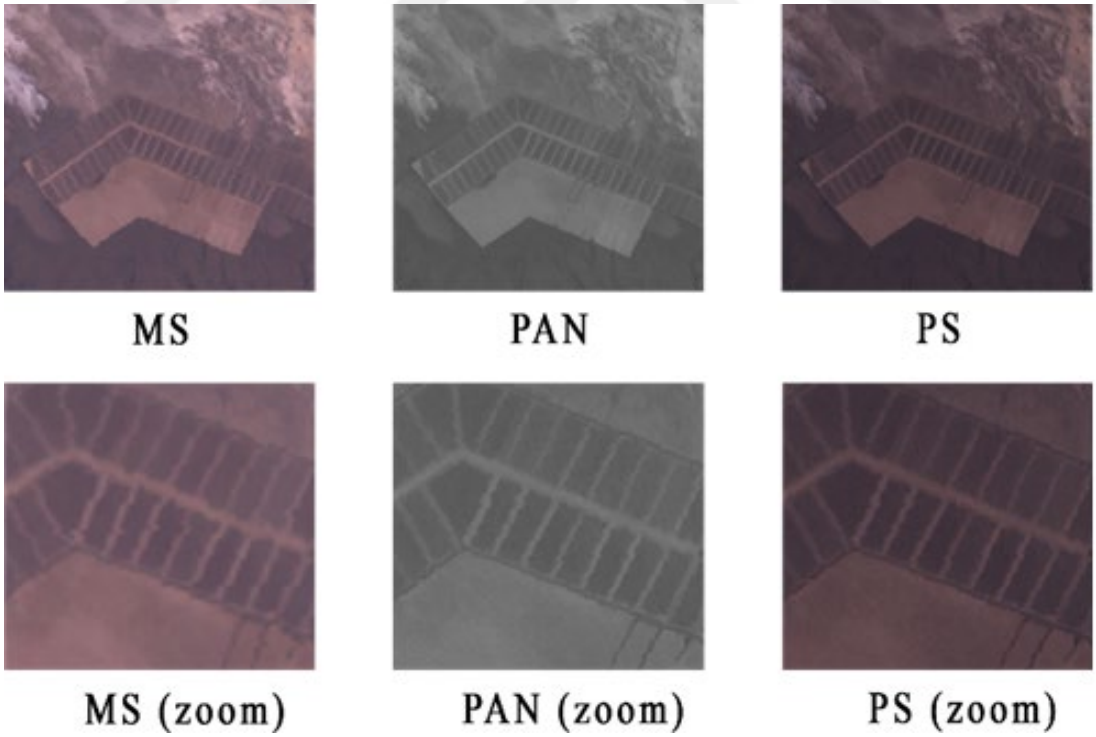


Figure 4.11 Result comparison and the zoomed area for the image #11

Table 4.12 Result comparison table for the image #12

Size = 1200x1200	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	21.52000	**10.48000**	2.05
Brovey Transformation	22.04030	12.07347	1.83
HSV Transformation	459.54200	205.21600	**2.24**

This image retrieved from Shahdoosti and Ghassemian works [4]. For this image, the best time is in Simple Mean algorithm but HSV Transformation hast the best speed up ratio according to Table 4.12. Figure 4.12 shows how the output looks like using Brovey transformation.

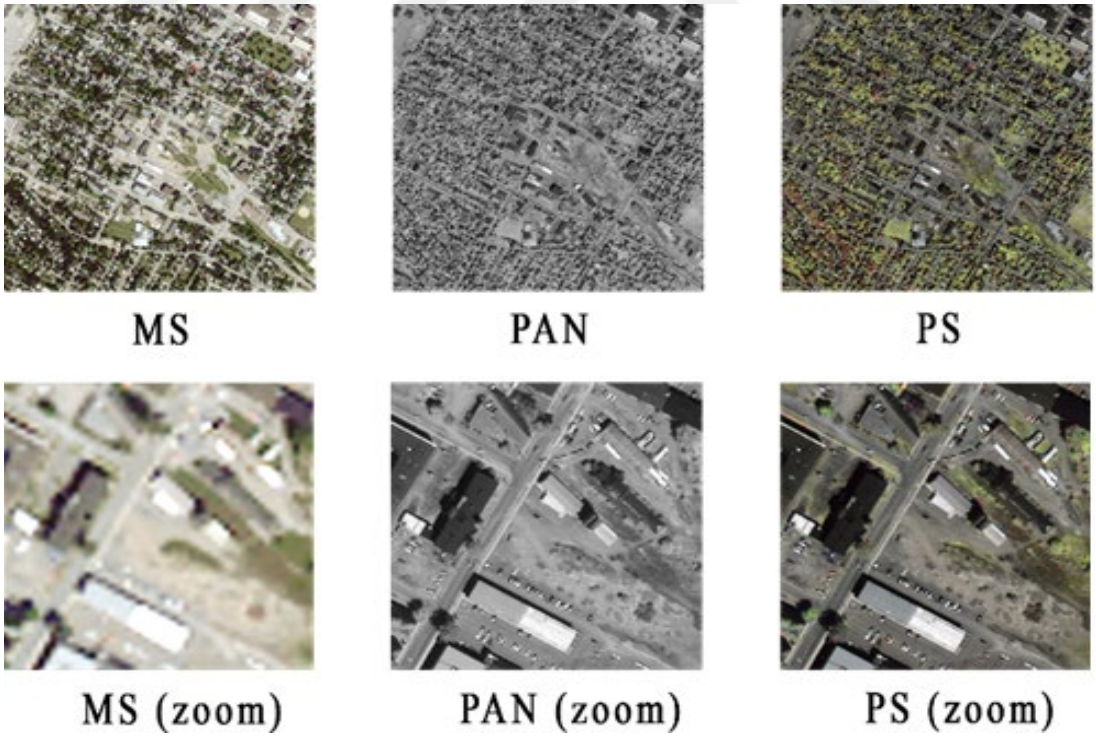


Figure 4.12 Result comparison and the zoomed area for the image #12

Table 4.13 Result comparison table for the image #13

Size = 4040x4032	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	244.31000	148.15000	1.65
Brovey Transformation	294.31990	**136.57513**	2.16
HSV Transformation	464.13870	145.08763	**3.20**

This image retrieved from RASAT [5]. It is an aerial photograph of the Bosphorys. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.13. Figure 4.13 shows how the output looks like using Brovey transformation.

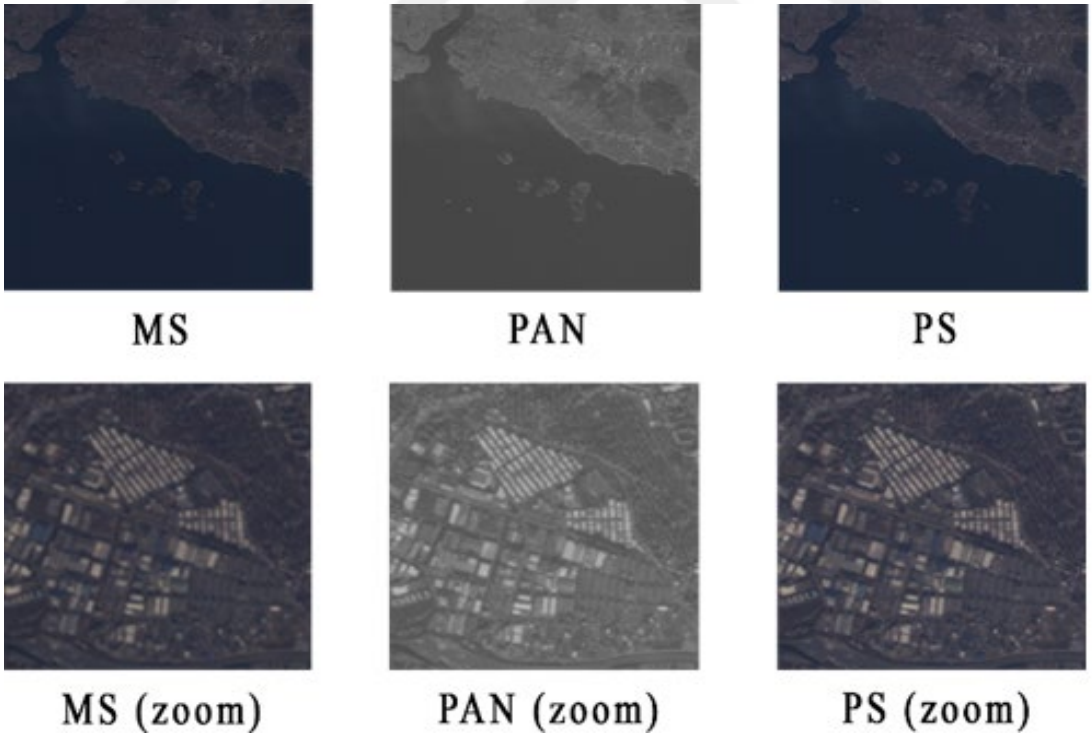


Figure 4.13 Result comparison and the zoomed area for the image #13

Table 4.14 Result comparison table for the image #14

Size = 5344x4484	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	320.09400	178.70800	1.79
Brovey Transformation	467.38435	223.78221	2.09
HSV Transformation	464.13870	**145.08763**	**3.20**

This image retrieved from WorldView-2 [6]. For this image, the best algorithm is HSV transformation according to Table 4.14. Figure 4.14 shows how the output looks like using Brovey transformation.

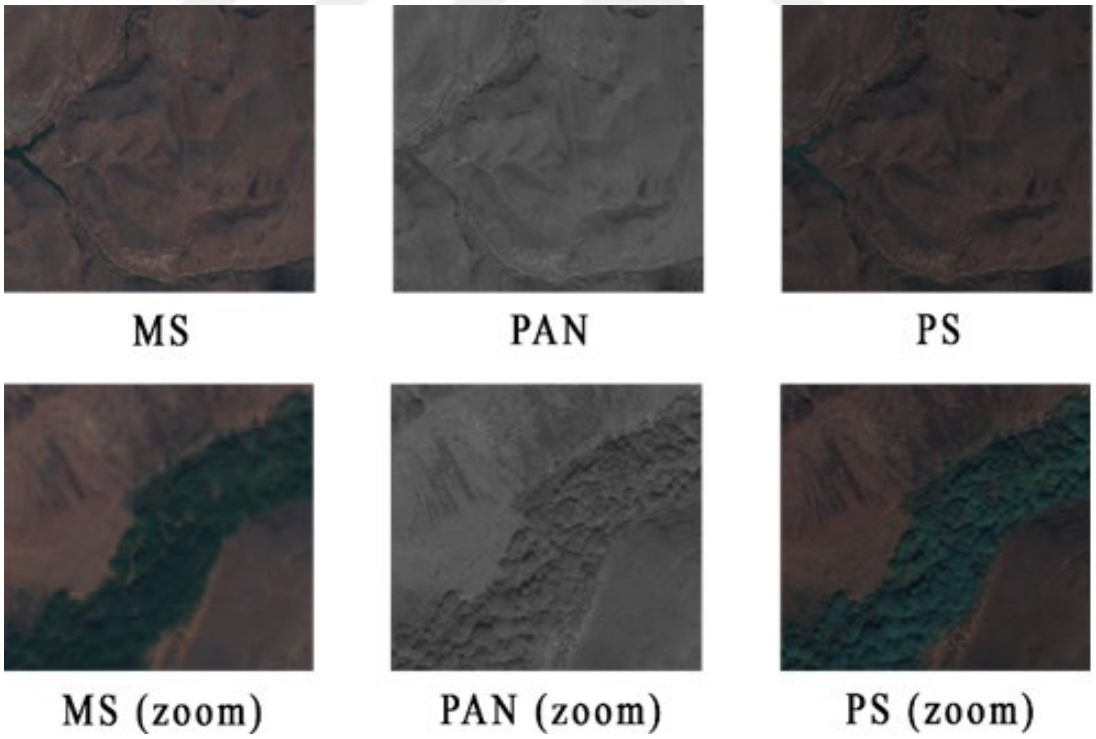


Figure 4.14 Result comparison and the zoomed area for the image #14

Table 4.15 Result comparison table for the image #15

Size = 15201x15221	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	2512.36640	1993.90200	1.26
Brovey Transformation	4562.58850	**1932.24916**	2.36
HSV Transformation	7213.66413	2345.90960	**3.07**

This image retrieved from Landsat 7 [7]. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.15. Figure 4.15 shows how the output looks like using Brovey transformation.

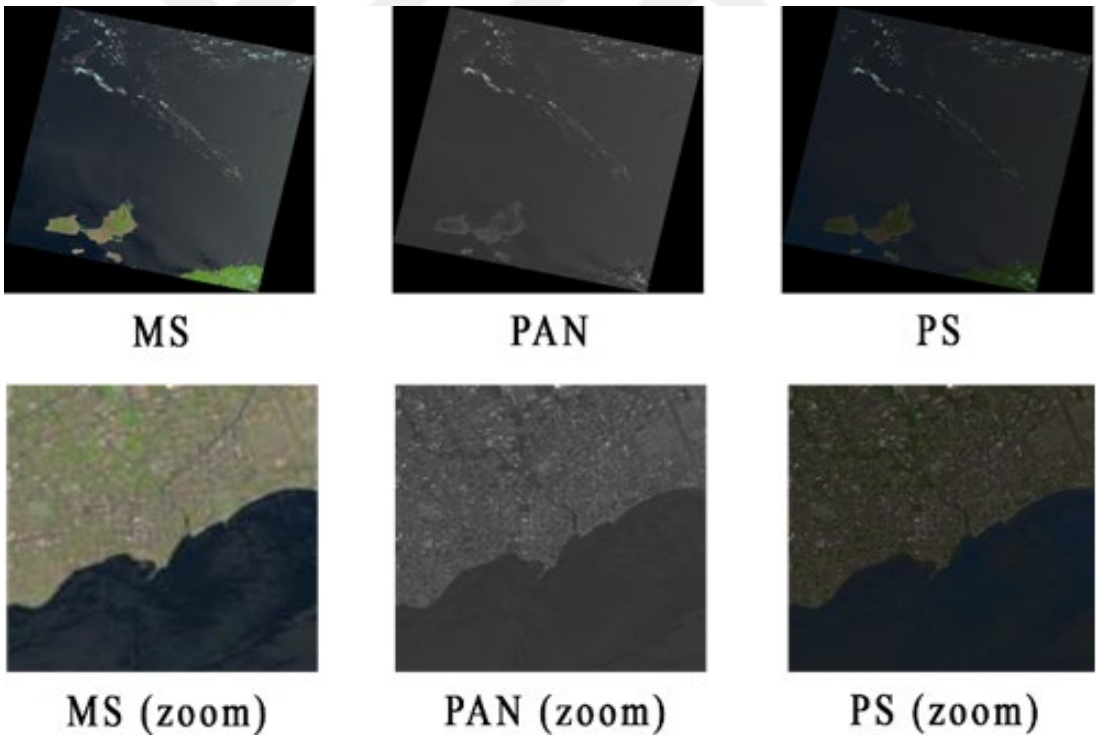


Figure 4.15 Result comparison and the zoomed area for the image #15

Table 4.16 Result comparison table for the image #16

Size = 4040x4032	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	288.63050	169.73000	1.70
Brovey Transformation	317.71960	152.12318	2.09
HSV Transformation	459.20830	**142.24821**	**3.23**

This image retrieved from RASAT [5]. It is an aerial photograph of Ankara. For this image, HSV Transformation is the best algorithm according to Table 4.16. Figure 4.16 shows how the output looks like using Brovey transformation.

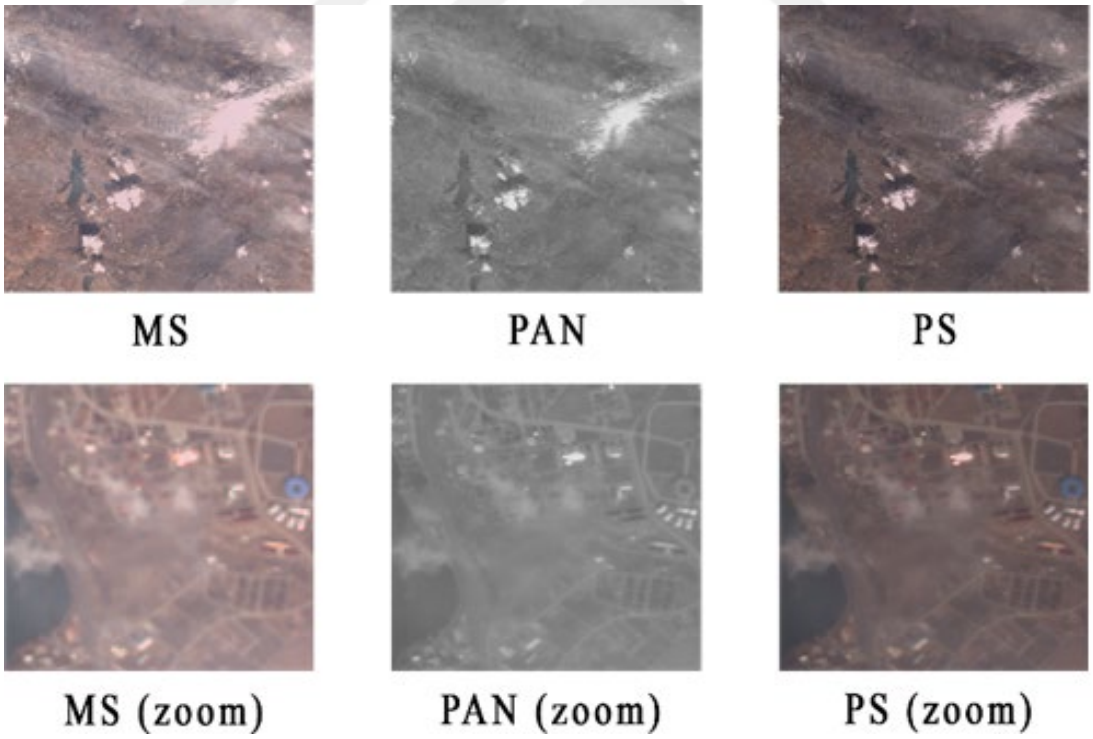


Figure 4.16 Result comparison and the zoomed area for the image #16

Table 4.17 Result comparison table for the image #17

Size = 4040x4032	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	274.43760	165.44000	1.66
Brovey Transformation	321.21650	**136.03469**	2.36
HSV Transformation	451.23240	143.36943	**3.15**

This image retrieved from RASAT [5]. It is an aerial photograph of Ankara. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.17. Figure 4.17 shows how the output looks like using Brovey transformation.

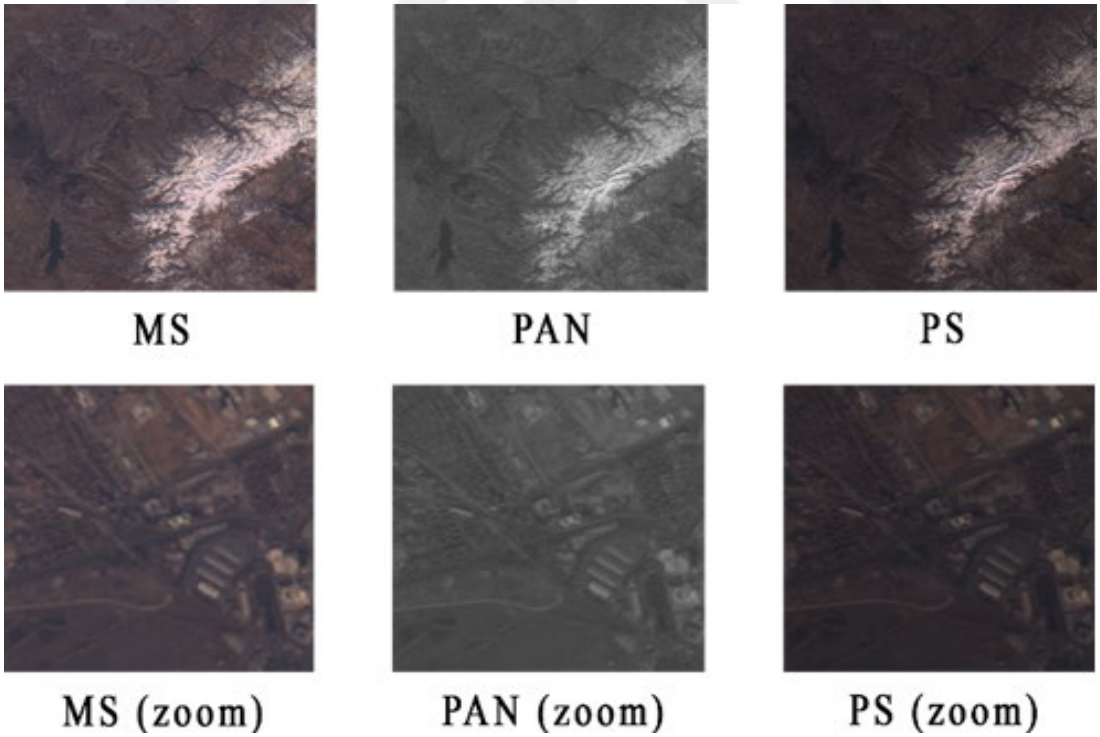


Figure 4.17 Result comparison and the zoomed area for the image #17

Table 4.18 Result comparison table for the image #18

Size = 4040x4032	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	253.22630	175.38000	1.44
Brovey Transformation	319.31040	**152.56995**	2.09
HSV Transformation	495.06710	153.71800	**3.22**

This image retrieved from RASAT [5]. It is an aerial photograph of the Bosphorus. For this image, the best time is in Brovey Transformation but HSV Transformation hast the best speed up ratio according to Table 4.18. Figure 4.18 shows how the output looks like using Brovey transformation.

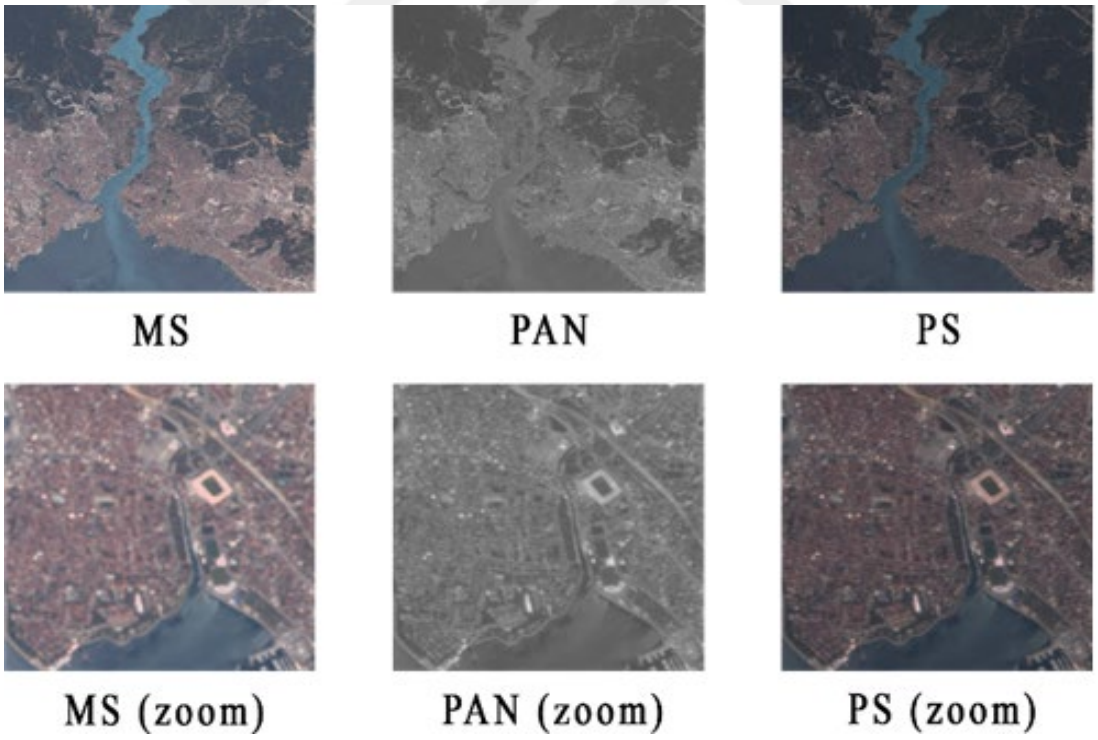


Figure 4.18 Result comparison and the zoomed area for the image #18

Table 4.19 Result comparison table for the image #19

Size = 4040x4032	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	249.90020	158.58000	1.58
Brovey Transformation	351.01740	151.37813	2.32
HSV Transformation	507.85800	**144.03664**	**3.53**

This image retrieved from RASAT [5]. It is an aerial photograph of İzmir. For this image, HSV Transformation is the best algorithm according to Table 4.19. Figure 4.19 shows how the output looks like using Brovey transformation.

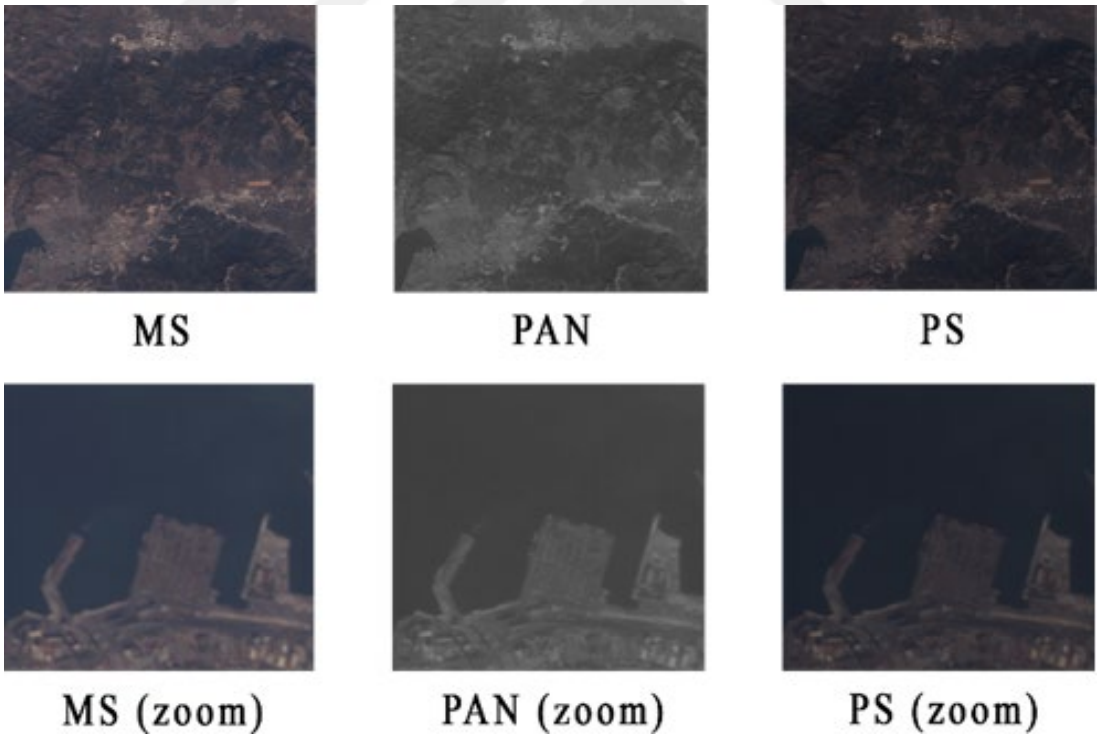


Figure 4.19 Result comparison and the zoomed area for the image #19

Table 4.20 Result comparison table for the image #20

Size = 4040x4032	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
Simple Mean	298.89330	145.63000	2.05
Brovey Transformation	349.98220	177.21465	1.97
HSV Transformation	521.75460	**141.93761**	**3.68**

This image retrieved from RASAT [5]. It is an aerial photograph of Çanakkale. For this image, HSV Transformation is the best algorithm according to Table 4.19. Figure 4.19 shows how the output looks like using Brovey transformation.

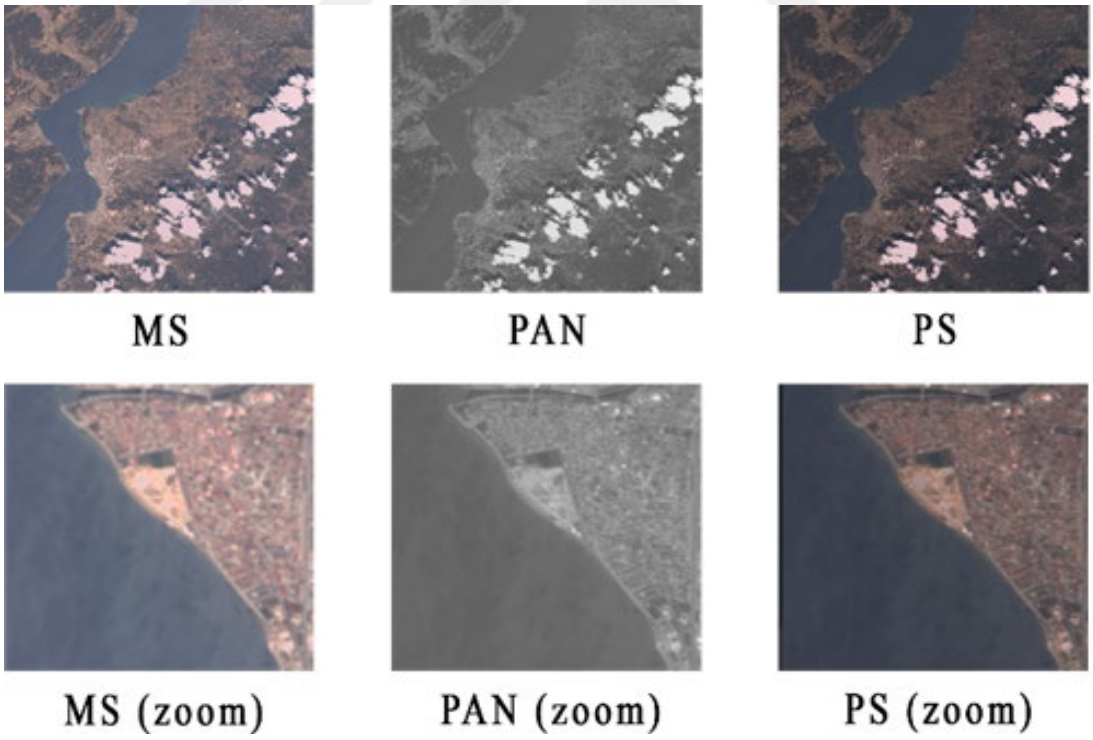


Figure 4.20 Result comparison and the zoomed area for the image #20

CHAPTER 5

CONCLUSION

Main motivation of the parallelization is getting a remarkable speed up value according to the CPU time. It is not possible to make every algorithm works in parallel. If an algorithm cannot give an acceptable speed up ratio, it is not reasonable to force this algorithm to work in parallel.

This study focus on observing whether parallelization of pan sharpening algorithm is beneficial or not with CUDA environment. Throughout the study, any additional library, framework or script is not used. It is just a pure C++ code and CUDA environment. Therefore, all images was need to be in mathematical model not in images as they were. For this reason, the images were pre-processed and converted into mathematical model as matrices in .txt files as a comma-separated values (csv) format. The outputs are also in mathematical format. Thus, to view the output image, all need is combining these matrices as a single file with a format of image.

According to the results that I have obtained, it is obvious that Simple Mean algorithm gives the worst result by quality of output image. However, in some cases, this algorithm is the fastest. It may be seen as a correct choice but when the quality is considered, this algorithm is unworthy. The output image of this algorithm can be told having a blur or a curtain over it. Hence, it cannot be used in a professional or a commercial product. It can be used for educational purpose only.

Brovey Transformation, in contrast, is the most beneficial algorithm until a size of 4000x4000. In some results, although HSV or Simple Mean has the best speed up value, by means of quality and time, Brovey Transformation gives better results. Given the mathematical model of the algorithm which is quite easy, it gives more than expected. It can be used in any purpose.

HSV Transformation gives much better results when the image has size more than 4000x4000. Below this size, the algorithm takes longer from other algorithms. The elapsed time for this algorithm is in a specified range for all images independent from the size. Nevertheless, above the size of 4000x4000, the other algorithms approach to

the same values with HSV Transformation. If all these three algorithm have close values of time and speed up, the best choice will be HSV transformation by means of quality. With help of some other library or frameworks, the elapsed time can be reduced. Thus, HSV Transformation can give much better results.

REFERENCES

[1] *Electromagnetic Spectrum Wikipedia*. Retrieved May 7, 2019 from https://en.wikipedia.org/wiki/Electromagnetic_spectrum

[2] Özkarayel L. *Elektromanyetik Spektrum Bize Ne Anlatıyor?*. Retrieved May 7, 2019 from http://www.acikbilim.com/2013/11/dosyalar/elektromanyetik-spektrum-bize-ne-anlatiyor.html November 4th, 2013

[3] Saeedi, Jamal & Faez, Karim. A new pan-sharpening method using multiobjective particle swarm optimization and the shiftable contourlet transform. ISPRS Journal of Photogrammetry and Remote Sensing, 2011

[4] Shahdoosti, Hamid & Ghassemian, Hassan. Fusion of MS and PAN Images Preserving Spectral Quality. Geoscience and Remote Sensing Letters, IEEE, 2015

[5] *Gezgin Geoportal* Retrieved May 7, 2019 from https://gezgin.gov.tr

[6] *High Resolution Satellite Imagery. Harris Geospatial Solutions* Retrieved May 7, 2019 from https://www.harrisgeospatial.com/Data-Imagery/Satellite-Imagery/High-Resolution/WorldView-2

[7] *Earth Explorer – USGS – Landsat 7* – Retrieved May 7, 2019 from https://earthexplorer.usgs.gov/

Appendix A – Simple Mean Algorithm Result Comparison Table

Function	Data	Size	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
SIMPLE MEAN	Image 1	256x256	1,27000	0,59389	2,14
	Image 2	512x512	4,73830	2,09921	2,26
	Image 3	512x512	4,75820	2,21099	2,15
	Image 4	512x512	4,01150	2,36701	1,69
	Image 5	512x512	4,58500	2,21893	2,07
	Image 6	512x512	4,99000	2,29815	2,17
	Image 7	512x512	4,68470	2,21654	2,11
	Image 8	512x512	4,70330	2,32190	2,03
	Image 9	512x512	4,21220	2,24527	1,88
	Image 10	700x700	7,03390	4,21299	1,67
	Image 11	400x399	2,21200	1,39560	1,58
	Image 12	1200x1200	21,52000	10,48000	2,05
	Image 13	4040x4032	244,31000	148,15000	1,65
	Image 14	5344x4484	320,09400	178,70800	1,79
	Image 15	15201x15221	2512,36640	1993,90200	1,26
	Image 16	4040x4032	288,63050	169,73000	1,70
	Image 17	4040x4032	274,43760	165,44000	1,66
	Image 18	4040x4032	253,22630	175,38000	1,44
	Image 19	4040x4032	249,90020	158,58000	1,58
	Image 20	4040x4032	298,89330	145,63000	2,05

Appendix B – Brovey Transformation Result Comparison Table

Function	Data	Size	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
BROVEY TRANSFORM	Image 1	256x256	1,36910	0,53175	2,57
	Image 2	512x512	5,48350	2,21766	2,47
	Image 3	512x512	5,82830	2,11248	2,76
	Image 4	512x512	6,64290	2,14449	3,10
	Image 5	512x512	5,27780	2,21542	2,38
	Image 6	512x512	5,21560	2,37642	2,19
	Image 7	512x512	5,62640	2,22351	2,53
	Image 8	512x512	5,39140	2,15412	2,50
	Image 9	512x512	5,25600	2,14957	2,45
	Image 10	700x700	8,89950	4,01842	2,21
	Image 11	400x399	2,66970	1,36771	1,95
	Image 12	1200x1200	22,04030	12,07347	1,83
	Image 13	4040x4032	294,31990	136,57513	2,16
	Image 14	5344x4484	467,38435	223,78221	2,09
	Image 15	15201x15221	4562,58850	1932,24916	2,36
	Image 16	4040x4032	317,71960	152,12318	2,09
	Image 17	4040x4032	321,21650	136,03469	2,36
	Image 18	4040x4032	319,31040	152,56995	2,09
	Image 19	4040x4032	351,01740	151,37813	2,32
	Image 20	4040x4032	349,98220	177,21465	1,97

Appendix C – HSV Transformation Result Comparison Table

Function	Data	Size	CPU (sec)	CUDA (sec)	Speed-Up (CPU-GPU)
HSV TRANSFORM	Image 1	256x256	279,83570	124,67000	2,24
	Image 2	512x512	366,19170	150,63000	2,43
	Image 3	512x512	579,71690	150,24641	3,86
	Image 4	512x512	545,80690	140,15000	3,89
	Image 5	512x512	548,29650	140,04541	3,92
	Image 6	512x512	614,97230	139,97521	4,39
	Image 7	512x512	624,09440	151,55215	4,12
	Image 8	512x512	703,93630	145,20663	4,85
	Image 9	512x512	585,08130	148,26783	3,95
	Image 10	700x700	645,62400	139,38522	4,63
	Image 11	400x399	402,42000	138,22430	2,91
	Image 12	1200x1200	459,54200	205,21600	2,24
	Image 13	4040x4032	464,13870	145,08763	3,20
	Image 14	5344x4484	678,25631	229,25564	2,96
	Image 15	15201x15221	7213,66413	2345,90960	3,07
	Image 16	4040x4032	459,20830	142,24821	3,23
	Image 17	4040x4032	451,23240	143,36943	3,15
	Image 18	4040x4032	495,06710	153,71800	3,22
	Image 19	4040x4032	507,85800	144,03664	3,53
	Image 20	4040x4032	521,75460	141,93761	3,68

CURRICULUM VITAE

PERSONAL INFORMATION



Name Surname : **Aykut KAŞKAYA**
Date of Birth : **01.01.1989**
Phone : **+90 530 689 1262**
E-mail : **aykutkaskaya@gmail.com**

EDUCATION

High School : **Afyonkarahisar Dinar Anatolian High School**
Bachelor : **Gaziantep University**
Engineering Faculty
Department of Electrical and Electronics Engineering

WORK EXPERIENCE

R&G Engineer: Duru Engineering (2014 - 2015)
System Analyst: Prime Ministry of Republic of Turkey (2015 - 2018)
System Analyst: Presidency of Republic of Turkey (2018 - …)

TOPICS OF INTEREST

- Software Development
- Machine Learning
- System and Network Administration
- Virtualization