A Novel Parallel Feature Extraction Method using HGAPSO and GLCO based SFTA

Inspiring Excellence

Supervisor: Dr. Jia Uddin

Md. Junayed Hasan 12101050

Nazmul Kabir Khan 12101078

Navila Alam Hridi 12101080

Fariha Tahsin Ontora 12101065

Department of Computer Science and Engineering,

BRAC University.

Submitted on: 20th April 2016

DECLARATION

We, hereby declare that this thesis is based on the results found by ourselves. Materials ofwork found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor	Signature of Author
Dr. Jia Uddin	Md. Junayed Hasan
	Navila Alam Hridi
	Md. Nazmul Kabir Khan
	Fariha Tahsin Ontora

ACKNOWLEDGEMENTS

All thanks to Almighty ALLAH, the creator and the owner of this universe, the most merciful, beneficent and the most gracious, who provided us guidance, strength and abilities to complete this research.

We are especially thankful to Dr. Jia Uddin, our thesis supervisor, for his help, guidance and support in completion of my project. We also thankful to the BRAC University Faculty Staffs of the Computer Science and Engineering, who have been a light of guidance for us in the whole study period at BRAC University, particularly in building our base in education and enhancing our knowledge.

Finally, we would like to express our sincere gratefulness to our beloved parents, brothers and sisters for their love and care. We are grateful to all of our friends who helped us directly or indirectly to complete our thesis.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Content based visual information retrieval system (CBVIR) is an important system to know the information of the images. Image is much more powerful than a document because it can say a lot more than a document itself. Feature extraction is one of the major steps of CBVIR system. For image classification, there are bunch of methods.Segmentation Based Fractal Texture Analysis (SFTA) is an efficient texture feature method among them for its higher precision and accuracy. For large number of Dataset, it is necessary for optimizing the feature extraction time and accuracy. As a result, we bring a new approach on SFTA algorithm on our research. We use an optimum multilevel thresholding hybrid method of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), called HGAPSO with our proposed GLCO (Grey Level Classification Based Optimization) method for increasing the effectiveness of the SFTA technique. To avoid the computational complexity we have implemented our proposed HGAPSO based SFTA algorithm on NVIDIA Graphics Processing Unit (GPU).GeForce GTX 610 is fully utilized to perform the scanning to see its efficiency. Our experimental results show average 95.5% classification accuracy for our tested dataset and also the GPU based implementation experiences 120+ X speedup over conventional CPU implementation.

CHAPTER 01

INTRODUCTION

1.1 Motivations

For image retrieval problem, Content Based Visual Information Retrieval (CBVIR) [1] performs pre-processing, extraction of features, creation of feature database and the work of matching the features of the query images to the dataset; these four basic steps. For these four steps, we are focusing on the Feature extraction techniques. As we know, texture extraction is important and it is needed to be faster for large scale dataset, we, in this thesis, proposed a parallel efficient SFTA technique. SFTA already achieved higher precision and accuracy for classification [2]. In addition, it is already 3.7 times faster than Gabor and 1.6 times faster than Haralick [2]. Though it is faster already, we again emphasized on making it more accurate and further faster for large number of dataset. In this research a new approach for SFTA is proposed. In the proposed method, we use hybrid multilevel threshold method called HGAPSO for finding out a significant number of threshold values for a better expected result than previous model. In [3] and [4], Ghamisi *et al*. proposed the HGAPSO method. This hybrid algorithm is mainly consisting of two portions: one is for combining the standard velocity and another is for updating the rules of PSOs (Particle Swarm Optimization) with the ideas of selection, crossover and mutation from GA (Genetic Algorithm) [5].There is a specific reason behind the replacement of multilevel thresholding approach. Existing Otsu method can easily be extended to solve the problem of determining the optimal threshold for multilevel, but it is not efficient as the computation time will increase exponentially [6]. Especially the problem occurs when the number of threshold increase on Otsu. So in our proposed SFTA model, we use HGAPSO method and our proposed GLCO (Grey level

Classification Based Optimization) method for optimizing the value of threshold value, which increases more accuracy.

Another concerning issue is total computation time for large scale dataset. The challenging phase is how much faster you can compute a task. With the availability of GPUs it can be believed that using its parallel techniques SFTA algorithm can have a faster processing time. An implementation of our proposed HGAPSO based SFTA algorithm is handled by GPU using Compute Unified Device Architecture (CUDA), will reduce the processing time a lot than sequential CPU processing time.

There are scopes to speed up the whole image feature extraction process and meet the demand of accelerating it. This paper illustrates how this task is handled by a GPU using Compute Unified Device Architecture (CUDA). CUDA by NVIDIA, a parallel computing architecture uses parallel compute engine in NVIDIA GPUs to solve many computationally intensive problems in a more efficient way than on CPU [7]. Using its parallel techniques we demonstrate how computation can speed up. We evaluated the proposed effective parallel SFTA feature extraction method for two different datasets: KTH-TIPS [11] and Textured Surfaces [12] which are publicly available [13-15].The symbols used throughout the paper are listed in Table 1.

Table 1 – Symbol Table

Symbol	**Definition**
I	Grayscale Image
I_b	Binary Image
n_l	Gray Level range
T_{op}	Set of optimized Threshold Values
n_t	Number of Threshold
n_{tg}	Number of GLC ranged Threshold
ε	Box Size in Box Counting Algorithm
Δ	Border Image
V_{SFTA}	SFTA Feature Vector
Kr	Kernel
P_l	Parameter List
D	Fractal Dimension

1.2 Contribution Summary

The summary of the main contributions is as follows:

- Otsu method can easily be extended to solve the problem of determining the optimal threshold for multilevel, but it is not efficient as the computation time will increase exponentially [6]. Especially the problem occurs when the number of threshold increase on Otsu. So in our proposed SFTA model, we use HGAPSO method for optimizing the value of threshold, which increases more accuracy.
- After applying HGAPSO, for further filtration for more accuracy, we proposed an algorithm named GLCO (Grey Level Classification based Optimization) to bring more accuracy to SFTA and make it reliable.
- By using CPU-GPU based implementation of our proposed work, the whole process

experiences 120+ X speedup over conventional CPU implementation.

1.3 Thesis Orientation

The rest of the thesis is organized as follows:

- Chapter 02includes the necessary background information regarding the proposed approaches of SFTA algorithm parallel.
- Chapter 03 presents the proposed model of our SFTA approach.
- Chapter 04 demonstrates the experimental results and comparison.
- Chapter 05 concludes the thesis and states the future research directions.

CHAPTER 02

BACKGROUND INFORMATION

2.1 GPU Architecture and CUDA

When we consider a processor that is more power efficient and have a better performance, GPU comes in top of that list. Comparing to a CPU, a GPU provides a better performance because they offer a higher peak GFLOPS (Giga floating-point operations per second) [8]. The GPU that we used for the experimentations is GeForce GTX 610. GeForce GTX 610 has several Multiprocessors (SM). The SM is paired into a GPC (Graphics processing clusters) block. Those GPC blocks are connected to the L2 cache and the cache is connected to the memory controller. There are mainly two types of memory in GPU. One is on-chip memory and the other is off-chip memory. The on-chip memory has low access latency but a relatively small size. On the other hand the off-chip memory has larger size and also higher access latency [9].

CUDA (Compute Unified Device Architecture) introduced by NVIDIA is a general purpose parallel computing platform and programming model [10]. The CUDA functions are called kernels. Unlike C functions that run only once this kernel runs N times in parallel by N different CUDA threads. Each thread that executes the kernel is given a unique thread ID. The threads are organized in a hierarchy consisting of blocks and grids as shown in the Figure 2.1 When calling a kernel function the size of the grid and thread block is specified. Below is an example of the function to call kernels.

$$Kr<<<dim\ Grid,\ dim\ Block>>>\ (P_l) \tag{1}$$

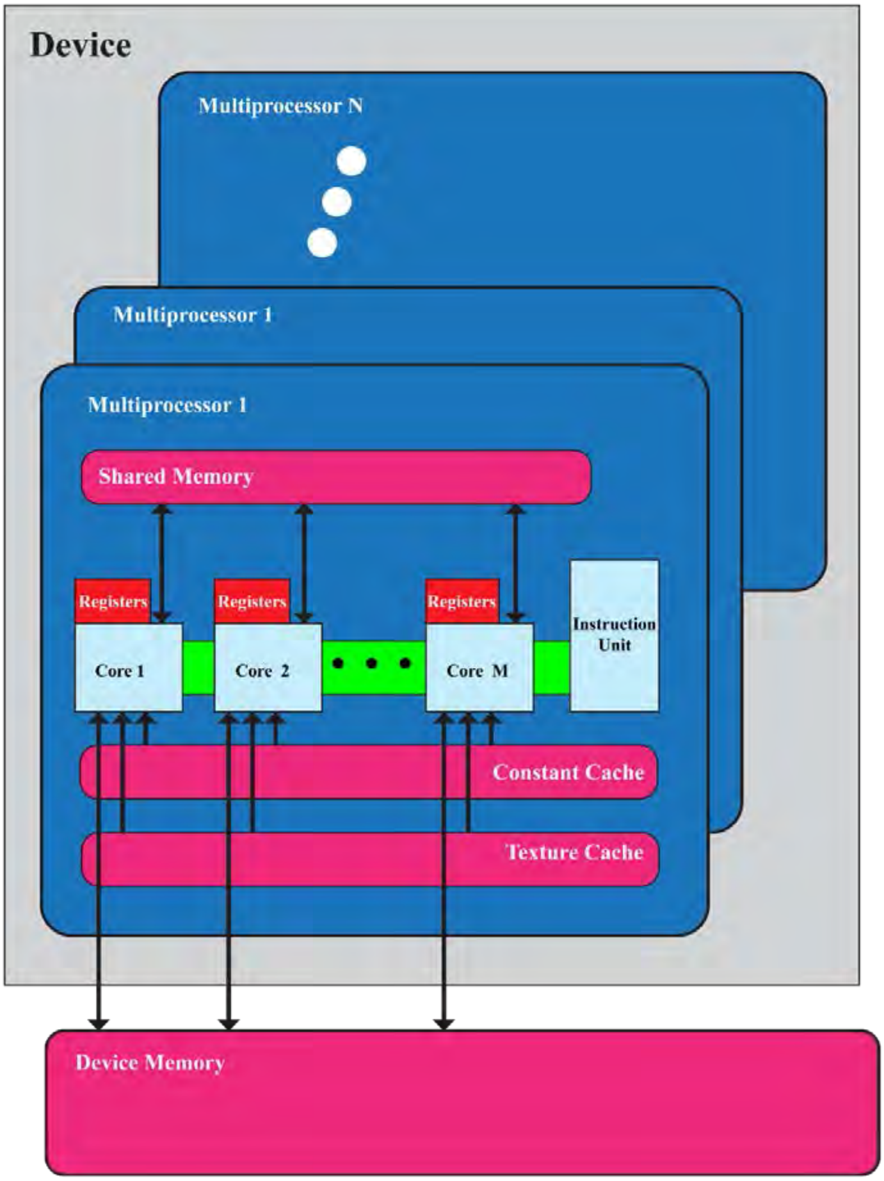

Fig 2.1: GPU Architecture.

2.2 SFTA Algorithm

The samples of the vibration signal in the time domain are utilized to present the signal in the 2-D dimension. Firstly, the matrix based on sample value of signal is generated by segmenting the signal into the same length sub-parts and arranging each sub-part as the column of the matrix. In this case, length of sub-part will decide the height of a matrix and the number of sub-part will define the width of a matrix. Finally, the value of samples will be normalized in the range 0-255 of gray image to convert the matrix into an image.

2.2.1 Two Threshold Binary Decomposition (TTBD)

First of all, the Two-Threshold Binary Decomposition (TTBD) takes as input a grayscale image I(x, y) and returns a set of binary images. The first step of TTBD consists in computing a set T of threshold values by Otsu multilevel threshold algorithm. Here Otsu algorithm is applied to each image region recursively till the desired number of thresholds is not obtained.

The next step of the TTBD algorithm consists in decomposing the input grayscale image I(x, y) into a set of binary images. This is achieved by selecting pairs of thresholds from the set of threshold values and applying a two-threshold segmentation [2] as follows:

$$I_b(x,y) = \begin{cases} 1 & if\ t_l < I(x,y) \le t_u) \\ 0 & 0, otherwise \end{cases} \qquad (2)$$

Where $t_l$and t_u are lower and upper threshold value. Where t_l and t_u are lower and upper threshold values. The set of binary images is obtained by applying the two threshold segmentation (eq. 2) to the input image using all pairs of contiguous thresholds from T ∪ { n_l} and all pairs of thresholds {t, n_l}, t ∈ T, where n_l corresponds to the maximum possible gray level in I(x, y) [2]. Thus, the number of resulting binary images is $2n_t$ [2].

Here is a problem regarding the number of the resulting binary images. On practical work, the number is not$2n_t$, it is $(2n_t - 1)$. The explanation is given on the chapter 3.3.

Another main issue in here is multilevel thresholding using Otsu. When Otsu method is extended to solve multilevel thresholding, it is not efficient to determine the optimal threshold for exponential computation time [6] along with increasing the number of thresholds.

So we propose our new approach based on HGAPSO in section 3 to get better accuracy and optimization.

2.2.2 SFTA Extraction Algorithm

Figure 2.2 shows a flow diagram of SFTA feature extraction algorithm. Where, After the TTBD being applied to the input gray level image, the SFTA feature vector is constructed as the resulting binary images' size, mean gray level and boundaries' fractal dimension. The regions 'boundaries of a binary image I_b(x, y) are the representation of a border image [2] denoted by Δ(x, y) and computed as follows:

$$(x,y) = \begin{cases} 1\ if\ \exists\ (x',y') \in N8[(x,y)]: \\ I_b(x',y') = 0\ \Lambda \\ I_b(x\ ,y) = 1 \\ 0, otherwise \end{cases} \tag{3}$$

Δ(x, y) takes the value 1 if the pixel at position (x, y) in the corresponding binary image I_b (x, y) has the value 1 and having at least one neighboring pixel with value 0. Otherwise, Δ(x, y) takes the value 0[2]. Here the resulting borders are one-pixel wide. The fractal dimension D is computed from each border image using the box counting algorithm. Fig 2 illustrates the main SFTA feature extraction Algorithm. First the input image is decomposed into a set of binary image by the TTBD algorithm [2]. Then, the SFTA feature vector is constructed as the resulting binary images 'size, mean gray level [2].

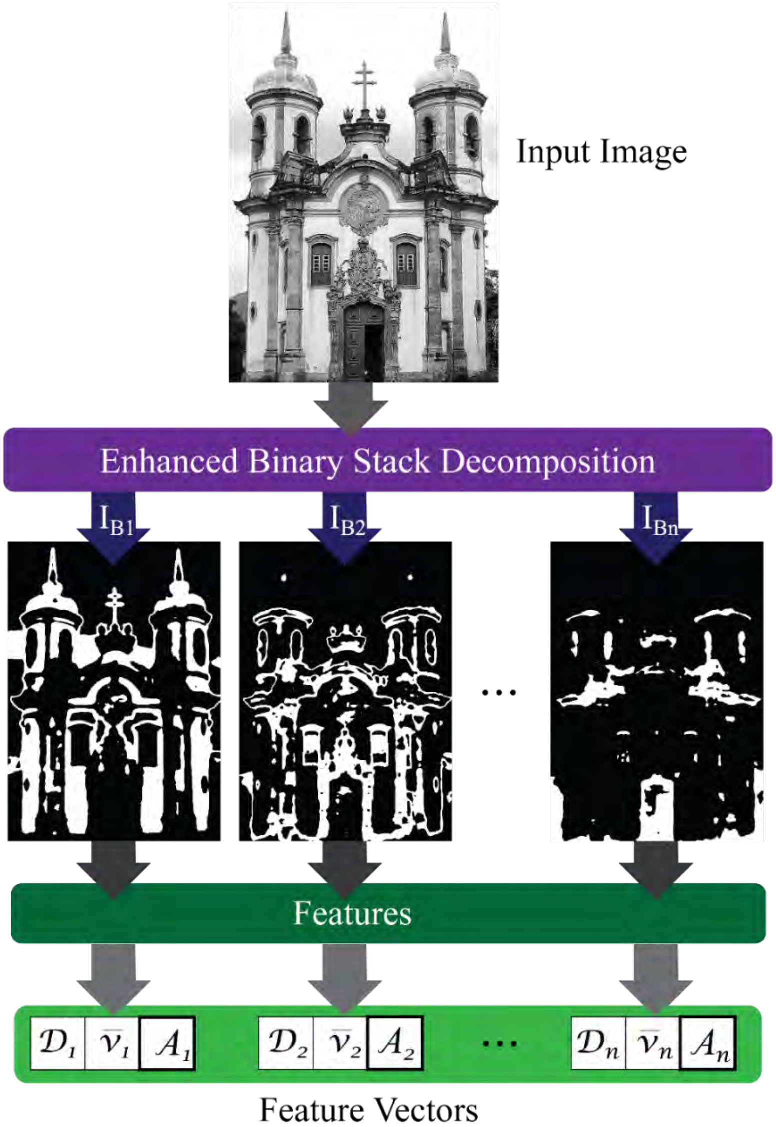

Figure 2.2: SFTA Extraction Diagram.

CHAPTER 03

PROPOSED HGAPSO and GLCO BASED SFTA APPROACH

3.1 Introduction

Figure 3.1 demonstrates a detailed implementation of our proposed model. It demonstrates how the algorithm is set up. It brings light to what part of the code is sent to GPU for execution. In our proposed model, first of all we apply multilevel hybrid threshold method HGAPSO to get desired number of threshold value. Then we apply our proposed GLCO (Grey Level Classification Based Optimization) method for optimization and bring more accuracy on threshold value finding. Then after the end of TTBD stage, we apply SFTA extraction portion to make our efficient SFTA algorithm. Finally we implement that efficient SFTA algorithm by using GPU.

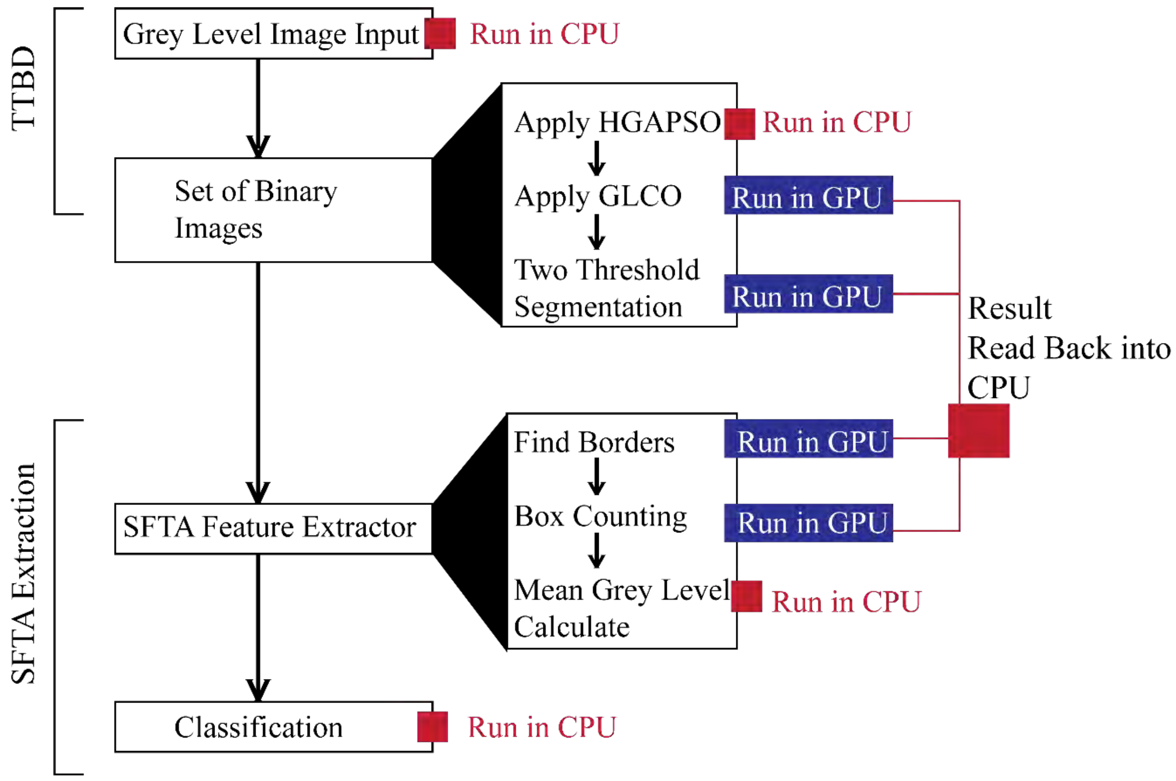

Fig 3.1:HGAPSO and GLCO based SFTA CPU-GPU Approach.

3.2 Apply HGAPSO

Based on the GA (Genetic Algorithm) and PSO (Particle Swarm Optimization) theorem [18, 20], it is found that each of them can be combined to obtain a better optimization results [3, 4]. In [3] and [4], Ghamisi et al. proposed the HGAPSO method. This hybrid algorithm has two main parts. One is to combine the standard velocity and another is to update the rules of PSOs with the ideas of selection, crossover and mutation from GA [5].

PSO (Particle Swarm Optimization) consists of a set of solutions, which we call swarm or population. Each solution consists of a set of parameters and represents a point in the multidimensional space, denoted as a particle [21]. The velocity of the particles are adjusted based on the historical behavior of each particle and its neighbors while flying through the search space. A particle's motion is highly affected by its current position and its memory of previous useful parameters along with the cooperation and knowledge of the swarm [5, 22]. Therefore, the particles have a tendency to fly towards a better search area over the search process course [20]. The velocity of the i-th particle in the k-th iteration is determined as follows:

$$V_{id}^{k+1} = WV_{id}^{k} + c_1 r_1 \left(pb_{id}^{k} - x_{id}^{k}\right) + c_2 r_2 \left(gb_{id}^{k} - x_{id}^{k}\right) \tag{4}$$

Where c_1 and $c_2$are acceleration constants and r_1 and r_2 are random values in the range of 0 and 1. The parameter W is regarded as the inertia weight. The parameter shows the position of each particle in the d-dimensional search space [5].

The parameter W (eq. 4) is regarded as the inertia weight. The parameter x_{id}^{k}shows the position of each particle in the d-dimensional search space. The best previous position of each particle is represented by pb_{id}^{k} and considered as particle best position.gb_{d}^{k} is the best position of all particles, being denoted as global best particle [5]. The i-th particle position is updated by:

$$x_{id}^{k+1} = x_{id}^{k} + V_{id}^{k} \tag{5}$$

On the other hand, Genetic algorithm (GA); an evolutionary optimization technique based on the genetic process [5, 23] It can emphasize much stronger on global, as opposed to local search and optimization [5, 22]. Furthermore, GA is able to find an optimal solution by not exploring the whole search space [5]. GA starts optimization with several solutions, called chromosome or individual. These chromosomes are consisting of several genes which can have different values [5]. The attributes of each individual are carried by these genes, which represents the fitness value of the chromosome. A set of the chromosomes establish a population. Among them, the most fitted ones are selected for generation. The two most fitted chromosomes are selected and their chromosomes are combined to make a new solution on the generation phase [5]. The act of combination is done by crossover, followed by a mutation mechanism applied on each child individually [23]. This cycle is repeated iteratively till a termination criterion is not met [24]. The hybrid algorithm HGAPSO combines the standard velocity and updates rules of PSOs with the ideas of selection, crossover and mutation from Gas [3, 4]. The Hybrid GAPSO is made for optimization of problems in continuous and multidimensional search spaces [25].

We are using this HGAPSO method for finding out a particular user defined number of threshold values with a better optimization.So finally our HGAPSO based SFTA algorithm is given here by using pseudo code.

3.3Apply GLCO (Grey Level Classification based Optimization)

From HGAPSO, we get the number of threshold values. Then again we pass the number of threshold values on this method to get the newly defined optimized with more accuracy threshold values. First of all, we equally classify the grey level density 0 to 255 into equal class. Then from all threshold values, we find their occurrence number on those classes. After that from each class we take the mean value from thresholds. Thus our proposed GLCO method helps to find out the defined number of optimized threshold values. The pseudo code is describing the whole GLCO process here:

1. Define GLC (Grey Level Class) by equally spaced band of numbers ranging from 0 – 255 [intensity of grey level image range]

*2. For each **t ∈ T**; find in which GLC; the t belongs to.*

3. For each GLC; calculate the mean of all threshold values of that class range to calculate the optimized threshold values.

4. If no threshold value in a GLC, threshold value = ***(Range start point + Range End point)/2***

5. Finally this function returns the optimized number of threshold called T_{op}

So finally our HGAPSO and GLCO based SFTA algorithm is given here by using pseudo code.

Algorithm 1 HGAPSO & GLCO Based SFTA extraction algorithm.

Require: Grayscale image I and number of thresholds n_t

Ensure: Feature vector VSFTA.

1: $T \leftarrow$ **HGAPSO** (I, n_t)

2: $T_{op} \leftarrow$ **GLCO** (T, n_{tg})

3: $T_A \leftarrow \{\{t_i, t_{i+1}\}: t_i, t_{i+1} \in T_{op},\ i \in [1..|T_{op}| - 1]\}$

4: $T_B \leftarrow \{\{t_i, n_l\}: t_i \in T_{op},\ i \in [1..|T_{op}|]\}$

5: $i \leftarrow 0$

6: for $\{\{t_l, t_u\}: \{t_l, t_u\} \in T_A \cup T_B\}$ **do**

7: $I_b \leftarrow$ **TwoThresholdSegmentation** (I, t_l, t_u)

8: $\Delta(x, y) \leftarrow$ **Find Borders** (I_b)

9: $V_{SFTA}[i] \leftarrow$ **BoxCounting** (Δ)

10: $V_{SFTA}[i + 1] \leftarrow$ **MeanGrayLevel** (I, I_b)

11: $V_{SFTA}[i + 2] \leftarrow$ **Pixel Count** (I_b)

12: $i \leftarrow i + 3$

13: end for

14: return V_{SFTA}

3.4The problem with the number of two threshold binary image and practical experiment based proposal

According to the proposed SFTA algorithm, Costa *et el.* [2] showed that, the number of resulting binary images is $2n_t$, but it should be $2n_t - 1$. Where t_l and t_u are lower and upper threshold value. The set of binary images is obtained by applying the two threshold segmentation (eq. 2) to the input image using all pairs of contiguous thresholds from T ∪ { n_l } and all pairs of thresholds {t,n_l}, t ∈ T, where n_l corresponds to the maximum possible grey level in I(x, y). Thus, the number of resulting binary images is$2n_t - 1$.Figure 3.2 shows the details of this result process [2].

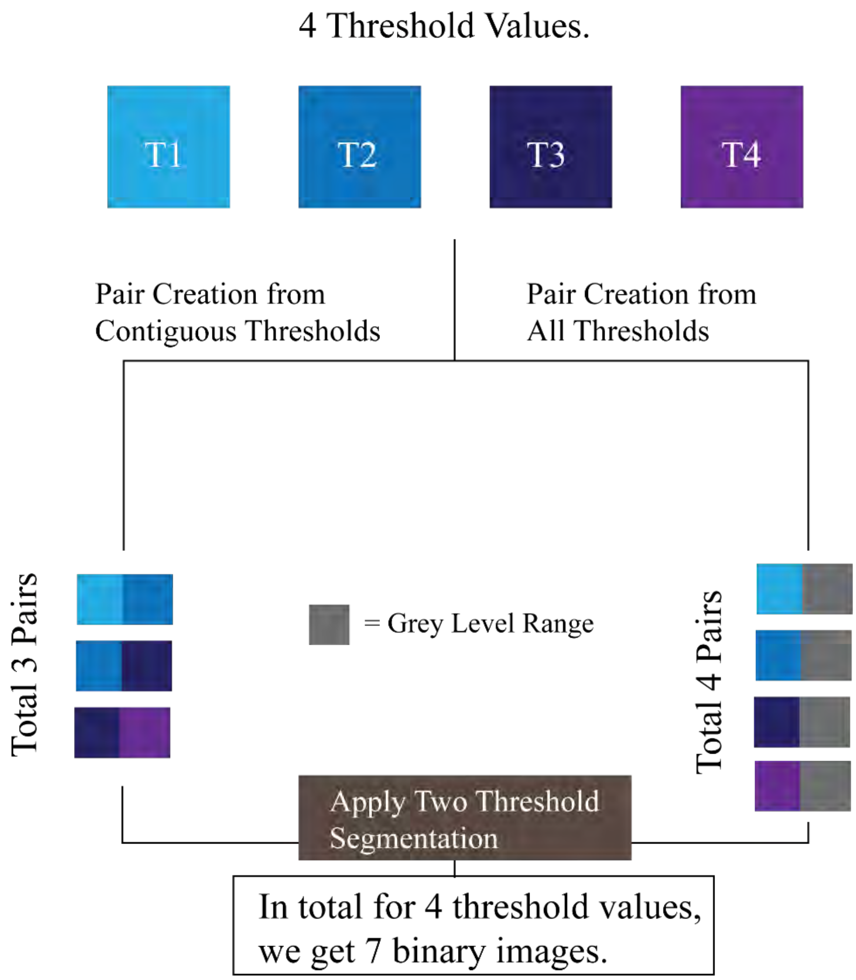

Figure 3.2: Binary Image Creation from Threshold Values.

So according to this number of $2n_t - 1$binary images, in Figure 3.3, we get the decomposition of region from a satellite image by applying TTBD, where number of threshold value was 8 and we got 15 binary images.

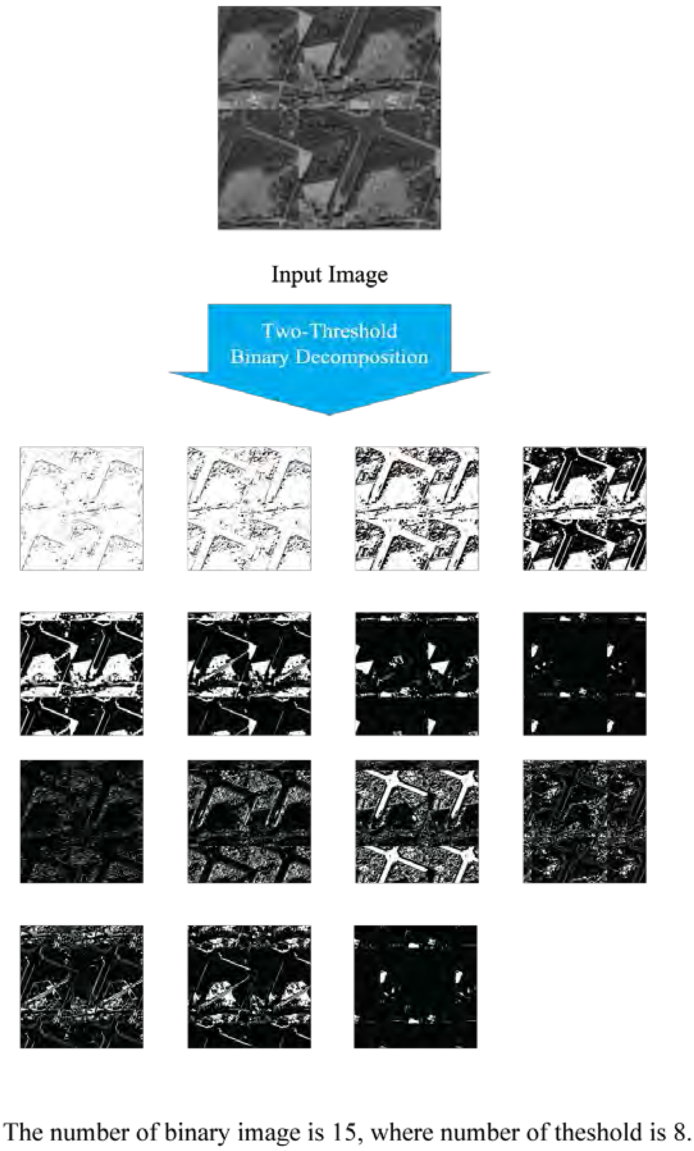

Figure 3.3: Binary Image Decomposition Form Satellite Image.

3.5Kernel Call

From the TTBD portion, the TTBD decomposition technique is performed on GPU. With the equal length class distribution of grey level range from 0 to 255, each segment is sent to corresponding threads. After the complementation of the work, the results are read back to the host (CPU) for processing. Again, from the SFTA extraction portion the find borders and box counting functions are performed in the second kernel. After the complementation of the work, the results are read back to the CPU for processing like kernel 1. Figure 3.4 illustrates the whole process.

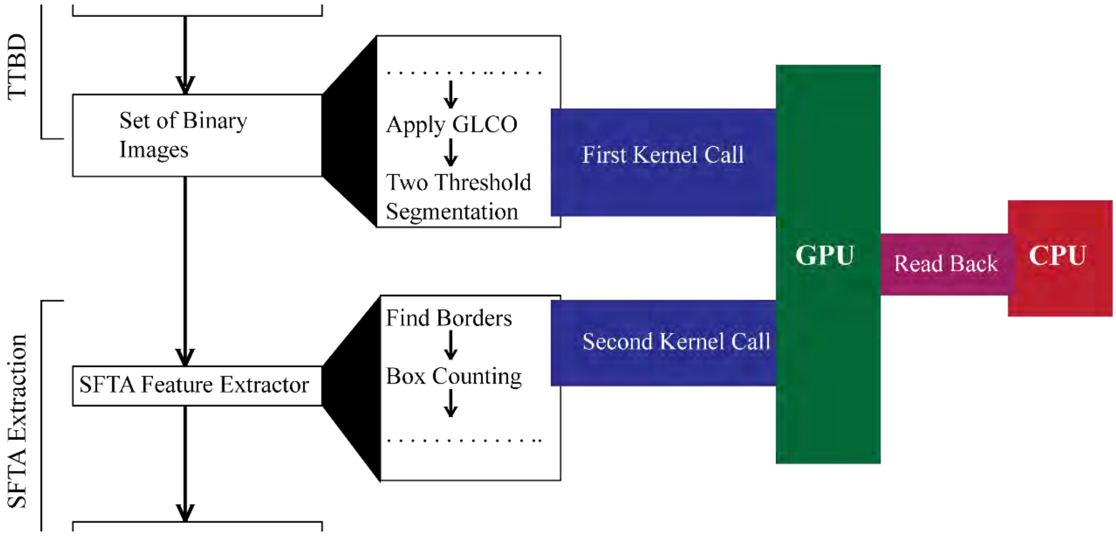

Figure 3.4: GPU Kernel Call.

CHAPTER 04

EXPERIMENTAL ANALYSIS

4.1 Introduction

Figure 3.1 demonstrates a detailed implementation of our proposed model. It demonstrates how the algorithm is set up. It brings light to what part of the code is sent to GPU for execution. In our proposed model, first of all we apply multilevel hybrid threshold method HGAPSO to get desired number of threshold value. Then we apply our proposed GLCO (Grey Level Class Based Optimization) method for optimization and bring more accuracy on threshold value finding. Then after the end of TTBD stage, we apply SFTA extraction portion to make our efficient SFTA algorithm. Finally we implement that efficient SFTA algorithm by using GPU.

For our proposed algorithm CUDA Toolkit 7.5 is used and the graphics hardware with specification: NVIDIA GT 610 All the experiments are done in a personal computer (PC) with the configuration Intel(R) Core i3-4160 CPU @ 3.6 GHz, 8GB RAM, running Windows 8.1.

We applied our proposed HGAPSO and GLCO based SFTA algorithm on two publicly available dataset: KTH-TIPS [11] and Texture Surface [12]. On these two different dataset, we apply the original SFTA algorithm and our proposed HGAPSO and GLCO based SFTA algorithm by taking same numbers of threshold values. Like main SFTA approach [2], we did not notice any major change on classification accuracy increasing phase after the threshold value reached 10 to 11. Until 10 to 11, the accuracy was in increasing order. The interesting fact is our proposed HGAPSO and GLCO based SFTA algorithm gave more accuracy on each threshold value till 9 than original SFTA algorithm. Mainly after 9, no additional texture patterns has been identified by our proposed TTBD phase. For particular image training on each dataset, we use SVM (Support Vector Machine).

After the accuracy had been tested, we make the GPU compatible version of our code. Then we calculate the runtime both in CPU and GPU of our proposed algorithm. It gave 15 times faster result than only CPU based code. We used several functions on GPU (figure 6) and for other functions, we used CPU. The main result of the total test were calculated on CPU.

4.2 KTH-TIPS Dataset

The KTH-TIPS (Textures under varying Illumination, Pose and Scale) is made of 810 grayscale images [11]. The dataset has 10 texture classes such as: aluminum foil, brown bread, corduroy, cotton and sponge. Figure 4.1 shows some sample images form the dataset.

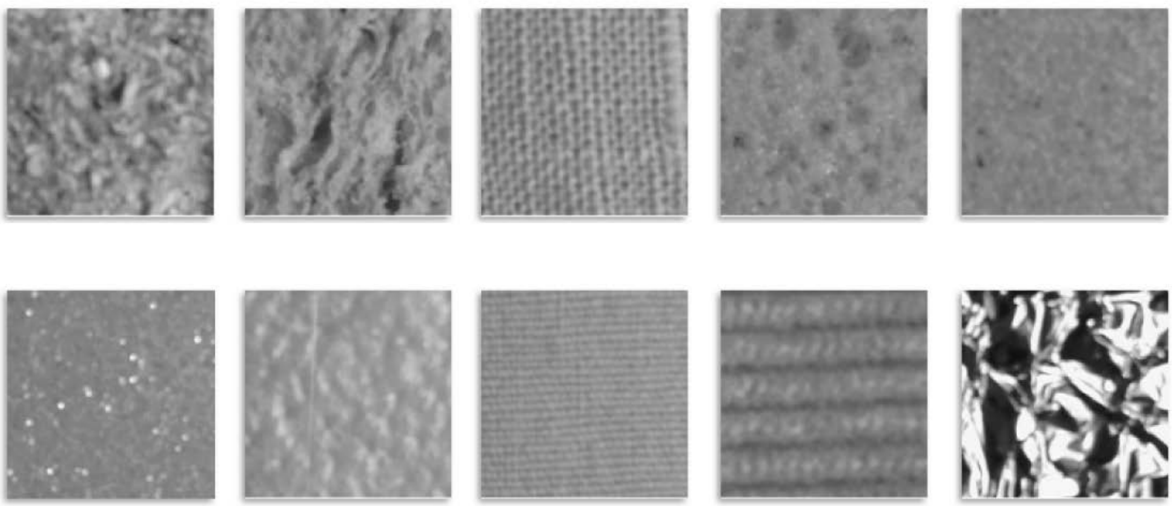

Figure 4.1: Sample Images from KTH-TIPS.

4.3The Textured Surfaces Dataset

The Textured Surfaces Dataset [12] has several surfaces composed of materials such as wood, marble and fur under varying viewpoints, scales and illumination conditions. 4.2 shows some sample images form the dataset.

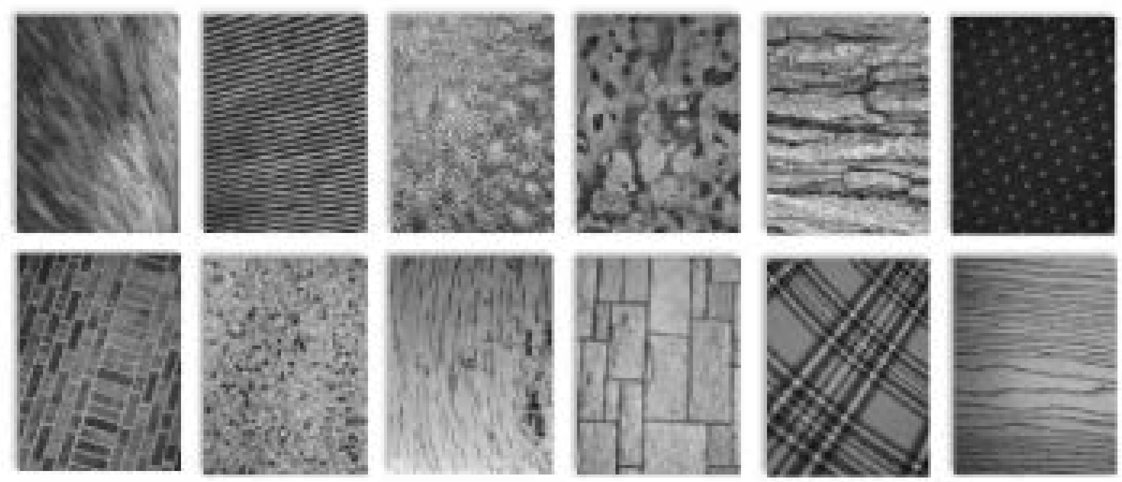

Figure 4.2: Sample Images from Texture Surfaces Dataset.

4.4Accuracy Measurement

Figure 4.3 shows the accuracy percentage for dataset KTH-TIPS [11] for number of threshold values. From here, it is clearly visible that our proposed algorithm can give more accuracy than SFTA [2] algorithm.

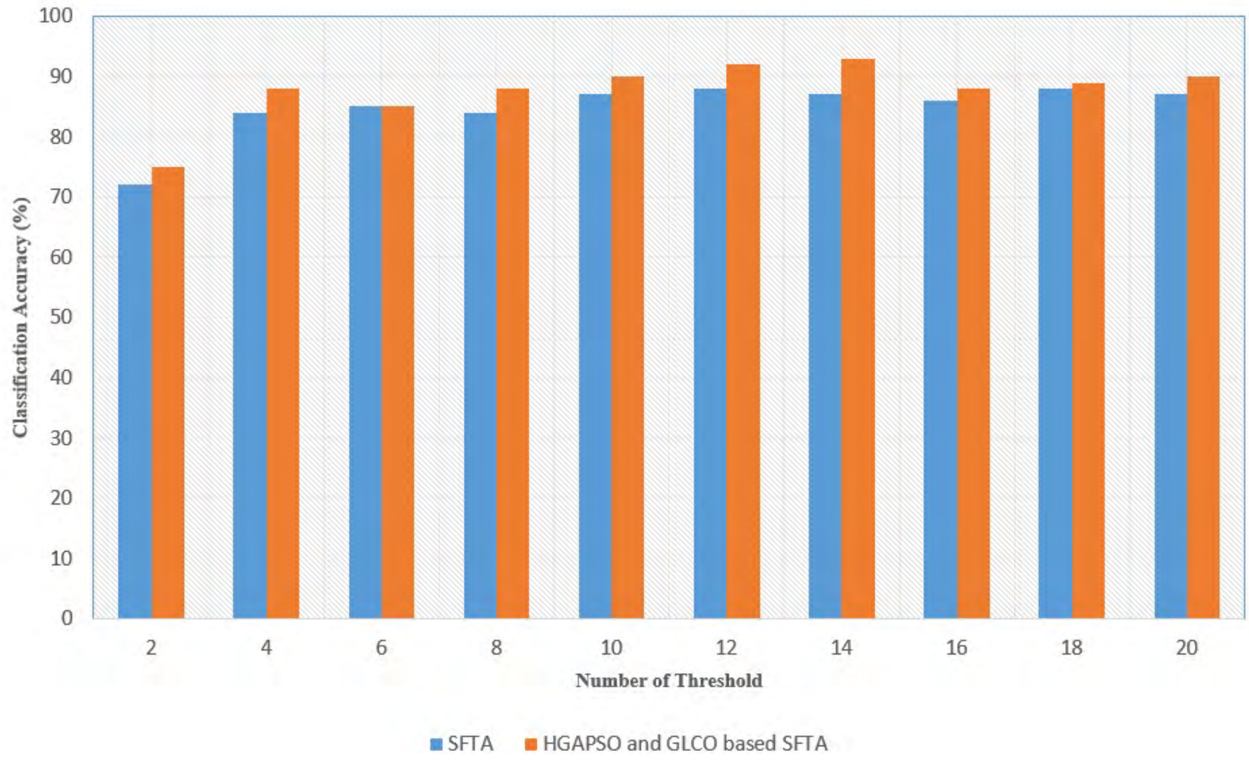

Figure 4.3: Classification Accuracy for KTH-TIPS Data.

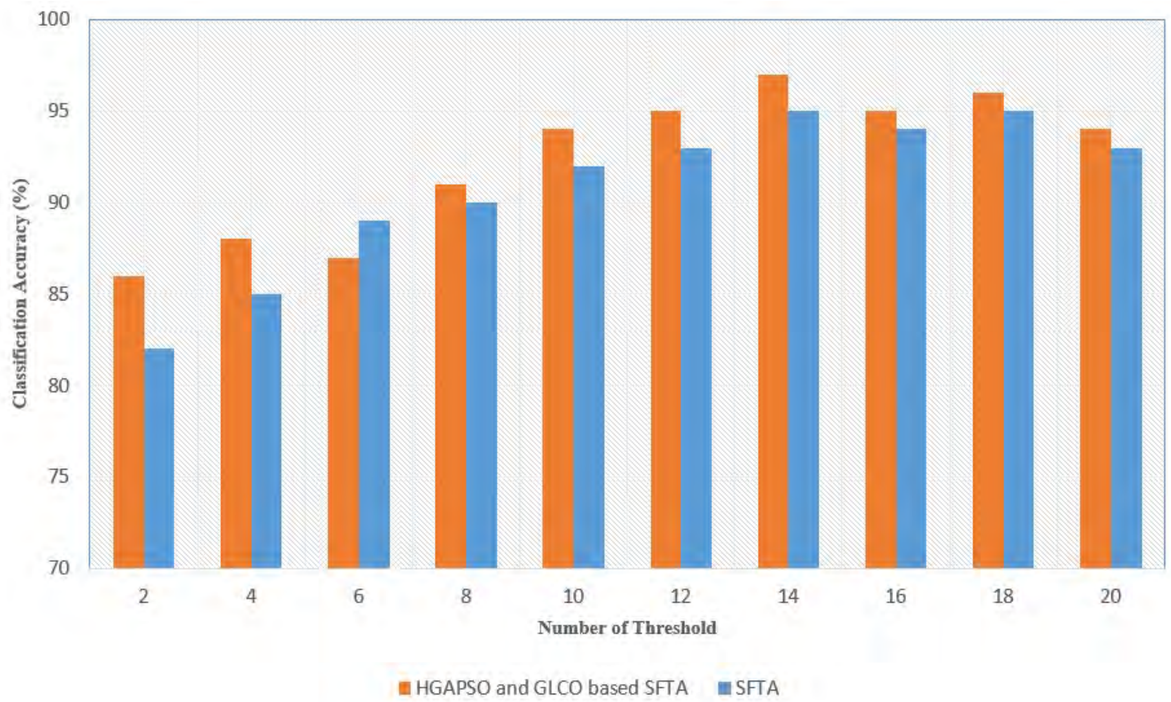

Figure 4.4: Classification Accuracy for Textured Surfaces Dataset.

Similarly Figure 4.4 shows the accuracy percentage for Texture Surfaces Dataset [12] for number of threshold values. Also from that, it is clearly visible that our proposed algorithm can give more accuracy than SFTA [2] algorithm.

4.5 Runtime Measurement

For CPU-GPU approach, we used 512 GPU blocks and 256 threads. We separated the datasets into several segments by dividing them into particular number of images to train them training purpose for classification. Then we run our proposed HGAPSO based approach on CPU first. After that we did the same thing, but this time we used our GPU-CPU based approach, which is the parallel process by using CUDA kernel. Then we compared the testing runtime of our proposed HGAPSO based SFTA algorithm with conventional SFTA method. We took different images from different varieties.

Table 2 shows the CPU-GPU based runtime and CPU based runtime for HGAPSO and GLCO based SFTA algorithm for texture surface dataset [12]. The surprising fact is on CPU-GPU based approach, the runtime is more than 120 x times faster in each case.

Table 2 – Runtime Comparison Chart for Textured Surface Dataset

Number of Images	GPU Block	GPU Thread	GPU CPU Based Runtime (seconds)	CPU Based Runtime (seconds)
20	512	256	2.89	355.67
40	512	256	5.43	717.36
80	512	256	11.31	1482.49

CHAPTER 05

CONCLUSIONS AND FUTURE WORKS

5.1 Concluding Remarks

In this thesis, based on the experimental result, showed that our new approach on SFTA algorithm based on HGAPSO and GLCO brings more accuracy on image classification with identification. Also this project proved that NVIDIA GPU based parallel approach on CUDA kernel can improve the runtime a lot, which is really amazing. Experimental results show average 95.5% classification accuracy for our tested dataset, where conventional Otsu based thresholding SFTA exhibits 93% classification accuracy. The proposed GPU based implementation experiences 120+ X speedup over conventional CPU implementation.

As everyday more GPU are coming and the architecture is improving, we do believe that the next generation GPU architecture will make the work much faster along with a result with better accuracy. Feature extraction will be much easier.

5.2 Future Works

The potential future directions for research based on the results presented in this thesis can be characterized into the following sections.

5.2.1 Image Conversion Techniques

Exploring the characteristics of the other time-frequency representations.

5.2.2 Feature Extraction and Classification

1. New texture features of an image are necessary to devise, which are robust to noise and can significantly decrease performance degradation in the distinction and also for more accuracy for finding the classifications.
2. Different statistical classifiers, including both supervised and unsupervised, should be investigated and experimented.

REFERENCES

[1] Ryszard S. Choras, “Image Feature Extraction Techniques and their applications for CBIR and Biometrices Systems”, International Journal of Biology and Biomedical Engineering, vol. 1,no. 1, pp. 6-16, 2007.

[2] A. F. Costa, G. E. Humpire-Mamani, A. J. M. Traina, “An Efficient Algorithm for Fractal Analysis of Textures”, Conference on Graphics, vol. 1, pp. 39-46, 2012.

[3] P. Ghamisi, J. A. Benediktsson, “Feature Selection Based on Hybridization of Genetic Algorithm and Particle Swarm Optimization”, IEEE Geoscience and Remote Sensing Letters, vol.12, pp.309-313, 2015.

[4] P. Ghamisi, “A Novel Method for Segmentation of Remote Sensing Images based on Hybrid GA-PSO”, International Journal of Computer Applications, vol. 29,pp. 7-14, 2011.

[5] S.K. Nahavandi, P.Ghamisi, L.Kumar, M.S.Couceiro, “A novel adaptive compression technique for dealing with corrupt bands and high levels of band correlations in hyperspectral images based on binary hybrid GA-PSO for big data compression”, International Journal of Computer Applications, vol. 109(8), pp.18–25,2015.

[6] Fauzi, Muhammad Ali, “OPTIMUM MULTILEVEL THRESHOLDING HYBRID GA-PSO BY ALGORITHM”, Jurnal Ilmu Komputer dan Informasi, vol. 6, no. 1, p. 1-5, 2013. [Online].Available: Jurnal Ilmu Komputer dan Informasi, http://jiki.cs.ui.ac.id/index.php/jiki/article/viewArticle/210. [Accessed March 18, 2015]

[7] Cheng, John, Max Grossman, and Ty KcKercher. Professional CUDAC Programming. Indianapolis: Wrox, 2014. Print.

[8] NVIDIA GTX 680 Whitepaper. Retrieved June 5, 2012 from http://www.nvidia.es/content/PDF/productspecifications/GeForce_GTX_60_Whitepaper_FIN

AL.pdf.

[9] Z. Jing, W. Hao, L. Heshan, and F. Wu-chun, “CUBLASTP: FineGrained Parallelization of Protein Sequence Search on a GPU”, in Parallel and Distributed Processing Symposium, 2014 IEEE 28th International, vol., no., pp.251-260, 2014.

[10] NVIDIA Corporation. NVIDIA CUDA C Programming Guide, 2015. Version 7.0.

[11] B. Caputo and E. Hayman, “Class-specific material categorization” ,Computer Vision, pp. 1597–1604 Vol. 2, 2005.

[12] S. Lazebnik, C. Schmid, and J. Ponce, “A sparse texture representation using local affine regions”, Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1265–1278, Aug. 2005.

[13] M. Varma and A. Zisserman, “A statistical approach to material classification using image patch exemplars,” IEEE transactions on pattern analysis and machine intelligence, vol. 31, no. 11, pp.2032–2047, Nov. 2009.

[14] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao, “WLD: a robust local image descriptor.” IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, pp. 1705–20, Sep. 2010.

[15] T. Ahonen, J. Matas, and C. He, “Rotation invariant image description with local binary pattern histogram fourier features,” Image Analysis, pp. 61–70, 2009.

[16] Otsu N, A threshold selection method from gray-level histograms, IEEE., 1979.

[17] Kapur, J. N., Sahoo, P. K., & Wong, A. K. C., “A new method for gray-level picture thresholding using the entropy of the histogram,” Computer Vision, Graphics and Image Processing, vol. 29, pp. 273-285, 1985.

[18] Yin, Peng-Yeng, A fast shceme for multilevel thresholding using genetic algorithms.

Signal Processing, 72, 85-95, 1999.

[19] Fogel, D. B., Evolutionary computation: Toward a new philosophy of machine intelligence. 2nd ed., Piscataway, NJ: IEEE Press, 2000.

[20] R.C. Eberhart, J. Kennedy, “A new optimizer using particle swarm theory”, In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39-43, 1995.

[21] P. Ghamisi, Micael S. Couceiro, Jon Atli Benediktsson, Nuno M. F. Ferreira “An Efficient Method for Segmentation of Images Based on Fractional Calculus and Natural Selection”, Expert Systems with Application Publisher, 39 (2012) 12407-12417.

[22] J. Kennedy and R.C. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, San Francisco, 2001.

[23] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Reading, Massachusetts, USA: Addison–Wesley Longman.

[24] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, A Novel Feature Selection Approach Based on FODPSO and SVM, IEEE Transaction on Geoscience and Remote Sensing, [In press].

[25] Peter J. Angeline. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In V. William Porto and et al., editors, Evolutionary Programming, volume 1447 of Lecture Notes in Computer Science, pages 601–610. Springer, 1998.