

PARALLEL IMPLEMENTATION OF MORPHOLOGICAL PROFILE BASED SPECTRAL-SPATIAL CLASSIFICATION SCHEME FOR HYPERSPECTRAL IMAGERY

B. Kumar^a, O. Dikshit^b

^a Department of Computer Science & Information Technology, MJP Rohilkhand University, Bareilly, India- bkumar@mjpru.ac.in

^b Department of Civil Engineering, Indian Institute of Technology Kanpur, India- onkar@iitk.ac.in

Commission VII, WG VII/4

KEY WORDS: Hyperspectral, Extended Morphological Profile (EMP), spectral-spatial classification, parallel processing, GPU

ABSTRACT:

Extended morphological profile (EMP) is a good technique for extracting spectral-spatial information from the images but large size of hyperspectral images is an important concern for creating EMPs. However, with the availability of modern multi-core processors and commodity parallel processing systems like graphics processing units (GPUs) at desktop level, parallel computing provides a viable option to significantly accelerate execution of such computations. In this paper, parallel implementation of an EMP based spectral-spatial classification method for hyperspectral imagery is presented. The parallel implementation is done both on multi-core CPU and GPU. The impact of parallelization on speed up and classification accuracy is analyzed. For GPU, the implementation is done in compute unified device architecture (CUDA) C. The experiments are carried out on two well-known hyperspectral images. It is observed from the experimental results that GPU implementation provides a speed up of about 7 times, while parallel implementation on multi-core CPU resulted in speed up of about 3 times. It is also observed that parallel implementation has no adverse impact on the classification accuracy.

1. INTRODUCTION

Hyperspectral imaging (Goetz, 2009) sensors capture an image scene in hundreds of fine contiguous spectral bands in ultraviolet to infrared region providing rich spectral and spatial information. Classification is an important tool for hyperspectral image analysis having applications in many areas including urban development, environmental studies, agricultural monitoring, and defense etc. However, the high dimensionality poses certain challenges to the supervised pixelwise classification due to curse of dimensionality and limited availability of training samples. Often dimensionality reduction is performed as a pre-classification step to mitigate such problems. PCA (Jolliffe, 2002), discrete boundary feature extraction (DBFE) (Lee and Landgrebe, 1993), discrete wavelet transform (DWT) (Bruce et al., 2002), and maximum noise fraction (MNF) (Chang and Du, 1999), etc. are among the most important feature extraction techniques used for dimensionality reduction. Unlike conventional spectral approach, better classification accuracies can be achieved by integrating spectral and spatial contents. Although, spatial information is not directly inherent with the pixels and usually determined using pixel neighborhood relationships. Some classes in an image may have similar spectral characteristics and therefore, complementary spatial information can help better discriminating the classes.

Mathematical morphology as been successfully used for spatial feature extraction in the form of structural information. Morphological operations require *a priori* definition of structuring element (SE) with a specified size and shape. However using a fixed size SE is not appropriate as one particular size may well detect the objects from a specific class but may not be suitable to detect objects from some another class. Benediktsson et al. (2003) used morphological operators with a multiscale approach to collect structural information. In

multiscale approach, successive opening and closing operations are performed using a series of SEs of increasing sizes to build morphological profiles (MPs). For hyperspectral images the MPs are built on each of the individual band images. These MPs all together are called extended MP (EMP). Considering high dimensionality of hyperspectral images, usually EMPs are built on the components obtained from some feature extraction technique. Benediktsson et al. (2005) built EMPs on principal components (PCs) and reported better results over spectral methods by experimenting on urban data. Fauvel, et al. (2008) computed EMPs from PCs and stacked with spectral features to perform classification with SVM. A different approach is used by Lv et al. (2014) to build MPs from PCs using multi-shaped SEs.

Considering volume of data, processing hyperspectral images results in higher computational cost. Although, feature extraction alleviates computational burden to some extent, multicore processors and graphics processing units (GPUs) can be used to further speed up such operations in parallel processing environment at desktop level. Tan et al. (2015) presented parallel version of SVMs on GPUs using compute unified device architecture (CUDA) (Sanders and Kandrot, 2011) to accelerate classification. Wu et al. (2015) proposed parallel implementation of composite kernels on SVMs for hyperspectral image classification. The small calculations are executed at CPU and intensive computations are ported on GPUs. The GPU based sparse representation classification of hyperspectral images is addressed in references (Wu et al., 2015a, Wu et al., 2015b). Extreme learning machine algorithm is developed on GPU for spectral-spatial classification for hyperspectral imagery by Lopez-Fandino et al. (2015) to significantly reduce execution time.

This paper presents an EMP based spectral-spatial classification method for hyperspectral images in parallel processing architecture. The rest of the paper is organized as follows. The

*Corresponding author.

process of creating EMPs is described in Section 2 and Section 3 presents parallel implementation of EMP based method. The experimental results with analysis are provided in Section 4. Finally Section 5 concludes the work.

2. EMP

Mathematical morphology is a useful technique for extracting structural information from an image. *Erosion* and *dilation* are the two fundamental operators in mathematical morphology (Soille, 2004), which are applied with a structuring element. Using erosion and dilation, other morphological operators can be defined among which, *opening* and *closing* are two widely used operators. Application of opening isolates the bright structures in the image while closing operation isolates the structures in darker areas. The concept behind the opening is to dilate an eroded image to recover eroded image structures bigger than structuring element. In contrast the closing operation is achieved by eroding the dilated image in order to recover initial shape of the dilated image structures. The determination of size of the structuring element is an important issue as most often the size of the structures in consideration is not exactly known. For this reason a range of different growing structuring element sizes are explored to record the proper response of the structures in the image. This concept is called granulometry (Soille, 2004). Pesaresi and Benediktsson (2003) defined the concept of morphological profile Φ using granulometry to analyze the image structures. The opening profile $\Phi^n(x)$ at pixel x of image I , which is a collection of n opening outcomes, is defined as

$$\Phi^n(x) = \{\phi^{(n)}(x), \phi^{(n-1)}(x), \dots, \phi^{(1)}(x)\} \quad (1)$$

where $\phi^i(x)$ is opening by reconstruction with structuring element of size i . The closing profile $\Psi^n(x)$ at pixel x is given by

$$\Psi^n(x) = \{\psi^{(1)}(x), \psi^{(2)}(x), \dots, \psi^{(n)}(x)\} \quad (2)$$

where $\psi^i(x)$ is closing by reconstruction with structuring element of size i . The morphological profile at pixel x is defined by stacking $\Phi^n(x)$ and $\Psi^n(x)$ as follows

$$\begin{aligned} MP(x) &= \{\Phi^n(x), \Psi^n(x)\} \\ &= \{\phi^{(n)}(x), \dots, \phi^{(1)}(x), \\ &\quad I(x), \psi^{(2)}(x), \dots, \psi^{(n)}(x)\} \end{aligned} \quad (3)$$

where $\Phi^{(1)}(x) = \Psi^{(1)}(x) = I(x)$. Eq. (3) defines morphological profile for one band image. For high dimensional images the profiles are created from m -band transformed data to form EMP. The EMP for a pixel x is formed as follows

$$EMP(x) = \{MP^1(x), MP^2(x), \dots, MP^m(x)\} \quad (4)$$

where MP^k is the morphological profile built on k th band image. The serial implementation approach for creating EMPs are described in Figure 1, where pixels are operated one by one with the help of a loop. A method for spectral-spatial hyperspectral image classification is presented in Figure 2. Although, EMPs consist of both spectral and spatial components but in this methodology explicit spectral features are combined with EMPs to fully exploit spectral information.

3. PARALLEL IMPLEMENTATION

In traditional approach, the operations are carried out pixel by pixel. However, since same operations are performed on all the pixels, this methodology is well suitable for single instruction

Input: Principal components **PC** of hyperspectral image, size vector **s**

Output: Extended morphological profile **EMP** of the image

```

for i=1 to npc //npc: number of PCs
    for j=1 to np //np: number of pixels
        k=(i-1)*2*np+(j-1)*2+1;
        EMP[k]=opening(PC,i,j,s);
        //opening is function to create opening profile
        EMP[k+1]=closing(PC,i,j,s);
        //closing is function to create closing profile
    end for
end for

```

Figure 1: Serial approach for creating EMP

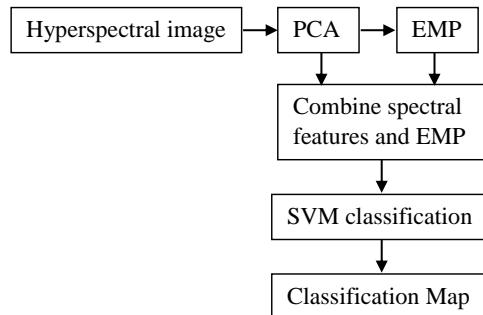


Figure 2: EMP based spectral-spatial classification

multiple data (SIMD) parallel architecture. Modern processors with multiple cores have capabilities to perform parallel operations. The methodology given in Figure 1 can be implemented on such processor by distributing computational load among the multiple cores to speed up the execution. The different cores can simultaneously and independently execute operations. Languages like MATLAB have in-built features like *parfor* to write parallel programs. In order to create morphological profiles in parallel, the inner loop in Figure 1 can be replaced by *parfor* loop as described in Figure 3. The *parfor* loop uses processor cores to perform identical operations on different elements. MATLAB creates multiple copies known as workers, which are invoked and released as and when required. If workers are less than the simultaneous identical tasks needed, the job scheduler switches and coordinates cores among the tasks.

GPUs are massively parallel processors based on SIMD programming model. GPU can execute same instruction on multiple data elements with the help of thousands of concurrent threads. Individual threads handle different data elements. In GPU programming model, the parallel threads are arranged in a 2-D grid of blocks. Each block has multiple threads that are organized in a 3-D array. The grid and blocks are logically managed by the programmer, however upper limit on the sizes vary from device to device. In CUDA based implementations a special function known as *kernel* is created to specify instructions for parallel threads. GPUs are optimized for intensive computations (Wu et al., 2015a). Therefore, a hybrid CPU-GPU model is proposed for executing small operations on CPU while transferring memory intensive calculations on GPU as shown in Figure 4. As shown in the figure, after feature extraction, image data elements are transferred to the GPU. GPU creates EMPs and classifies with the help of SVM. For K -class problem, an ensemble of $K(K - 1)/2$ binary SVMs is created (Tan et al., 2015). Different binary SVMs can be handled by different processing threads. The classification map is then written back to the main memory.

```

Input: Principal components PC of hyperspectral image, size vector s
Output: Extended morphological profile EMP of the image

mycluster=parcluster('local');           // create cluster profile
mycluster.NumWorkers=numWorkers          //numWorkers: no. of workers
saveprofile(mycluster);
parpool(mycluster); //Pool of workers is created
for i=1 to npc                         //npc: no. of PCs
    parfor j=1 to np                     //np: no. of pixels
        k=(i-1)*2*np+(j-1)*2+1;
        EMP[k]=opening(PC,i,j,s);
        //opening is function to create opening profile
        EMP[k+1]=closing(PC,i,j,s);
        //closing is function to create closing profile
    end parfor
end for
delete(gcp('nocreate'));                //close pool of workers

```

Figure 3: Creating EMP on multi-core CPU

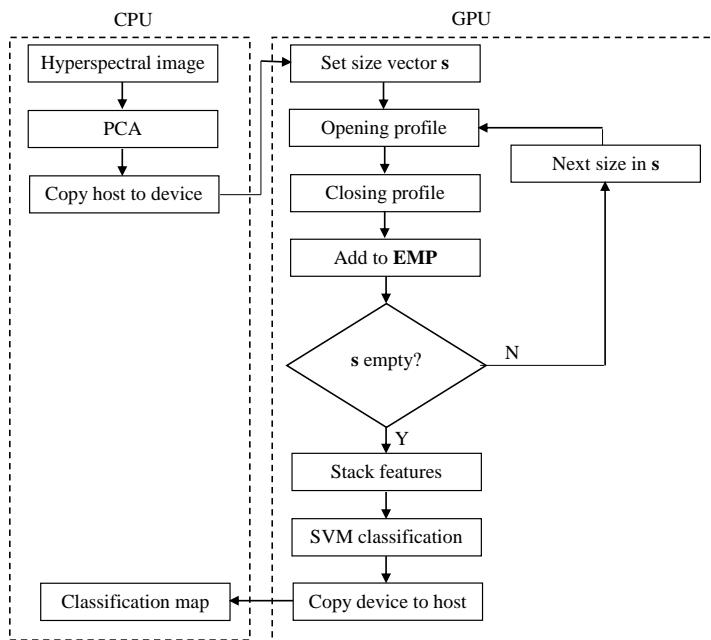


Figure 4: Framework for CPU-GPU implementation of EMP based spectral-spatial classification

4. EXPERIMENTS AND DISCUSSION

The experiments are carried out to evaluate the impact of parallel implementations on the execution time and classification accuracy. The details of experimental setup and results obtained are discussed in this section.

4.1 Hyperspectral images

The experiments are performed on two hyperspectral images Pavia University and La Mancha Alta. Pavia University image is taken over the Engineering School at University of Pavia, Italy by German aerospace agency with ROSIS-3 sensor. It is 610×340 pixels image having 103 spectral bands in $0.43 - 0.86\mu\text{m}$ range. The pixel size is 1.3m . The scene has nine information classes of interest: (1) Asphalt, (2) Meadows, (3) Gravel, (4) Tree, (5) Metal sheets, (6) Soil, (7) Bitumen, (8) Bricks, and (9) Shadow.

La Mancha Alta image having dimensions of 512×512 is captured by DIAS-7915 sensor in spectral range of

$0.40 - 12.5\mu\text{m}$ over an agriculture land known as La Mancha Alta in Spain. This image has 65 bands and pixel size of 5m . There are eight land cover classes of interest: (1) Bricks, (2) Pasture land, (3) Soil, (4) Vineyard, (5) Wheat, (6) Hydrophytic veg, (7) Salt, and (8) Water. The false color composites (FCCs) of both images are shown in Figure 5.

4.2 Experimental setup

The SVM is implemented with LIBSVM (Chang et al., 2011) tool using radial basis function (RBF) kernel. The SVM parameters cost C and spread of kernel γ are optimally determined using 5-fold cross validation. The functions of LIBSVM are also used for parallel SVM implementation. The training samples are randomly taken from the ground reference image and rest of the reference pixels are used to test the accuracy of the results. The EMPS are created using SE of *disk* shape and size is varied as 3×3 , 5×5 , 7×7 , and 9×9 . The implementation is done on Intel Xeon E5-1650 operating at 3.50 GHz that features 6 cores and 16 GB RAM. The GPU is

Approach	Pavia University		La Mancha Alta	
	Time (s)	Speed up	Time (s)	Speed up
Serial	29.85	-	36.25	-
Multi-core	10.47	2.85	11.33	3.2
CPU-GPU	4.88	6.12	5.08	7.14

Table 1: Execution time and speed up for different approaches

NVIDIA Tesla T10 GPU having 240 cores operating at 1.296 GHz. The dedicated GPU memory is 4 GB.

The classification accuracies are determined in terms of kappa coefficient (Congalton and Green, 1999) defined as follows:

$$\kappa = \frac{N \sum_{i=1}^K x_i - \sum_{i=1}^K y_i z_i}{N^2 - \sum_{i=1}^K y_i z_i} \quad (5)$$

where N is the total number of pixels, K is the number of classes in the image, x_i represents the number of correctly classified pixels in i th class, y_i is the number of pixels in class i , and z_i represents the number of pixels classified in class i .

4.3 Results and analysis

The results are obtained for two parameters: execution time and classification accuracy (κ). The reported results for each experiment are average of ten trials. First, experiments are carried out for execution time. The execution times for serial, multi-core, and CPU-GPU implementations for both images are reported in Table 1. It can be observed that by exploiting multiple cores of CPU, the execution can be accelerated by a factor of ≈ 3 . Since CPU used has only six cores, different elements are frequently scheduled to have run time leading to significant overhead. However, GPU being a massive parallel system is able to achieve speed up of ≈ 7 times. Larger the size of the image, more would be the speed up on GPU as also evident from the Table 1. La Mancha Alta image has more number of pixels than Pavia University image to be processed. Therefore, speed up is better for La Manch Alta image.

The classification accuracies are given in Table 2. Both classwise and global κ values are reported. It can be observed from the table that there is no adverse effect on the classification accuracy by parallel execution of the operations. Although, the accuracies in different implementations are different but differences are not significant. The observed differences in accuracies are due to random selection of training samples from the ground reference.

5. CONCLUSION

In this work parallel implementations of EMP based spectral-spatial classification of hyperspectral imagery are presented. The implementations are done on multi-core processor and CPU-GPU hybrid systems. The experiments on two different kind of hyperspectral images demonstrated that execution was accelerated significantly without compromising classification accuracies. The mult-core CPU implementation accelerated the execution by ≈ 3 times, while CPU-GPU hybrid implementation resulted in speed up by ≈ 7 times.

ACKNOWLEDGEMENTS

The authors are thankful to Prof. Paolo Gamba of University of Pavia, Italy and Prof. Mahesh Pal from NIT Kurukshetra, India for providing hyperspectral images.

REFERENCES

- Benediktsson, J. A., Palmason, J. A., and Sveinsson, J. R., 2005. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 480-491.
- Benediktsson, J. A., Pesaresi, M., and Arnason, K., 2003. Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1940-1949.
- Bruce, L. M., Koger, C. H., and Li, J., 2002. Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 10, pp. 2331-2338.
- Chang, C.-C., and Lin, C.-J., 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1-27:27.
- Chang, C.-I., and Du, Q., 1999. Interference and noise adjusted principal components analysis. *IEEE Transaction Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2387-2396.
- Congalton, R. G., and Green, K., 1999. *Assessing the accuracy of remotely sensed data- principles and practices*. Boca Raton, Lewis Publisher.
- Fauvel, M., Benediktsson, J. A., Chanussot, J., and Sveinsson, J. R., 2008. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, pp. 3804-3814.
- Goetz, A. F. H., 2009. Three decades of hyperspectral remote sensing of the Earth: a personal view. *Remote Sensing of Environment*, vol. 113, pp. S5-S16.
- Jolliffe, I., 2002. *Principal Component Analysis*. Springer-Verlag, New York.
- Lee, C., and Landgrebe, D. A., 1993. Feature extraction based on decision boundaries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 4, pp. 388-400.
- Lv, Z. Y., Zhang, P., Benediktsson, J. A., and Shi, W. Z., 2014. Morphological profiles based on differently shaped structuring elements for classification of images with very high spatial resolution. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 12, pp. 4644-4652.
- Lopez-Fandino, L., Quesada-Barriuso, P., Heras, D. B. and Arguello, F., 2015. Efficient ELM-based techniques for the classification of hyperspectral remote sensing images on commodity GPUs. *IEEE Journal of Selected Topics on Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2884-2893.
- Pesaresi, M., and Benediktsson, J. A., 2003. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Transaction Geoscience and Remote Sensing*, vol. 39, no. 2, pp. 309-320.
- Sanders, J., and Kandrot, E., 2011. *CUDA by example: an introduction to general-purpose GPU programming*. Addison Wesley.
- Soille, P., 2004. *Morphological Image Analysis*. Berlin, Germany: Springer-Verlag.

Classwise κ											
Pavia University						La Mancha Alta					
Class	Train	Test	Serial	Multi-core	CPU-GPU	Class	Train	Test	Serial	Multi-core	CPU-GPU
1	548	6304	0.9882	0.9854	0.9878	1	478	9250	0.9966	0.9975	0.9985
2	540	18146	0.9631	0.9597	0.9647	2	274	5358	0.9998	1.0000	1.0000
3	392	1815	0.9919	0.9941	0.9941	3	105	2159	0.9129	0.9109	0.9307
4	524	2912	0.9941	0.9946	0.9932	4	530	10916	0.9827	0.9745	0.9741
5	265	1113	1.0000	1.0000	1.0000	5	617	12100	0.9557	0.9588	0.9627
6	532	4572	0.9755	0.9866	0.9869	6	416	8060	0.9646	0.9560	0.9713
7	375	981	0.9915	0.9915	0.9867	7	75	1520	0.8601	0.8757	0.8707
8	514	3364	0.9967	0.9963	0.9938	8	125	2440	0.9346	0.9310	0.8901
9	231	795	0.9965	0.9947	1.0000	-	-	-	-	-	-
Global κ	-	-	0.9795	0.9794	0.9814	-	-	-	0.9681	0.9662	0.9682

Table 2: Global and classwise classification accuracies (κ)

Tan, K. et al., 2015. GPU parallel implementation of support vector machines for hyperspectral image classification. *IEEE Journal of Selected Topics on Applied Earth Observations and Remote Sensing*, vol 8, no. 10, pp. 4647-4656.

Wu, Z. et al., 2015. Parallel spatspectral hyperspectral image classification with sparse representation and Markov random fields on GPUs. *IEEE Journal of Selected Topics on Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2926-2938.

Wu, Z. et al., 2015. GPU implementation of composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 9, pp. 1973-1977.