Proceedings of the 18th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2018
July 9–14, 2018.

GPU computation of Attribute Profiles for Remote Sensing Image Classification

Pablo Quesada-Barriuso[1], Dora B. Heras[1], Francisco Argüello[2] and Begüm Demir[3]

[1] *Centro singular de Investigación en Tecnoloxías da Información (CiTIUS), Universidade de Santiago de Compostela*

[2] *Departamento de Electrónica y Computación, Universidade de Santiago de Compostela*

[3] *Faculty of Electrical Engineering and Computer Science, Technische Universitt Berlin*

emails: pabloqb@gmail.com, dora.blanco@usc.es, francisco.arguello@usc.es, demir@tu-berlin.de

Abstract

Classification of multi and hyperspectral remote sensing images is a common task. It usually requires a previous step consisting of a technique for extracting the spatial information from the image being profiles a common approach. In particular, attribute profiles are based on the application of a morphological filter to the connected components of the image producing relevant spatial information at different levels of detail. The information is built based on attributes such as area or standard deviation. Their high computational cost makes the attribute profiles good candidates for their execution on commodity GPUs. In this paper, the first parallel implementation of attribute profiles over multispectral images in CUDA for Nvidia GPUs is proposed. The GPU proposal is based on the construction of a max-tree that is traversed from the leaves to the root by merging the connected components of the tree obtaining a considerable reduction in execution time over the CPU execution.

Key words: Remote sensing, attribute profiles, supervised classification, real-time, GPU.

1 Introduction

Classification plays a fundamental role in the analysis of remote sensing images because it is a key step for applications such as crop monitoring, land-use analysis or risk management,

 ISBN: 978-84-697-7861-6

for example. The classification techniques that exploit not only the spectral information contained in multi and hyperspectral images but also the spatial information are especially efficient[1].

A large group of methods considering an adaptive neighborhood in order to extract the spatial information to be introduced in the classification stage are based on profiles. Profiles, such as morphological profiles (MPs) build granulometries of the image through opening and closing transformation with an arbitrary structuring element. Attribute profiles (APs) are a generalization of MPs that have been proposed to consider a metric based on an attribute in such a way that the different components of the profile do not depend on the size and shape of an structuring element, thus overcoming this limitation of the MP[2]. The most common attributes are area, standard deviation, diagonal of the bounding box or moment of inertia. The objective of all the profiles is to discard irrelevant spatial details and simultaneously preserve the relevant geometrical characteristics of the other regions.

It is important to note that the computational requirements of the classification methods for multispectral and hyperspectral datasets are high, so these are good candidates to be projected in high performance computing (HPC) infrastructures such as clusters or specialized hardware devices[3]. GPUs provide a cost-efficient solution to carry out onboard real-time processing of remote sensing hyperspectral data[4].

In this paper the first attribute filtering algorithm for building APs on GPU based on the max-tree representation of the remote sensing image is proposed. The algorithm requires building a max-tree that it is traversed from the leaves to the root by merging the connected components at each level of the tree. Execution times for CPU and for a commodity GPU programmed in CUDA are analyzed showing the benefits of a GPU implementation.

2 Attribute Profiles

APs and their extension to multidimensional images which is called extended attribute profiles (EAPs) are obtained by the sequential application of morphological attribute filters (AFs). The EAP is formed by stacking the MPs built for each band or reduced component of the image in a single dataset.

In this section, the concepts and notations regarding mathematical morphology that are required for understanding the morphological APs are introduced. Then, the attribute filtering and the attribute opening are reviewed. Finally, the EAP is described.

AFs, also known as attribute openings, are connected operators that process an image according to a given criterion such as the area or the standard deviation of the pixel intensity, removing connected components that do not satisfy that criterion. A *connected component* (CC) is a subgraph of an image f in which pixels are connected to each other by paths. AFs keep or merge the CCs based on a logical criterion T if a given attribute is greater/lower than an arbitrary reference, such as $T_\lambda^{\mathtt{att}}(CC) = \mathtt{att}(CC) > \lambda$, where $\mathtt{att}$ is an attribute

 ISBN: 978-84-697-7861-6

and λ is an arbitrary reference value [2].

The *binary connected opening* Γ_x of an image f at a pixel x preserves only the connected component X which contains x. The *binary trivial opening* Γ_T uses an increasing criterion T to filter X. If the criterion is satisfied, the connected component is preserved, otherwise it is removed according to:

$$\Gamma_T(X) = \begin{cases} X & \text{if} \quad T(X) = \text{true}, \\ \emptyset & \text{otherwise.} \end{cases} \tag{1}$$

The *binary attribute opening* Γ^T of a binary image f is defined as:

$$\Gamma^T(f) = \bigcup_{x \in f} \Gamma_T[\Gamma_x(f)]. \tag{2}$$

The attribute opening is the union of all the trivial openings $\Gamma_T(f)$ which meet the increasing criterion T. The *greyscale attribute opening* γ^T is given by the maximum grey level of the results of the filtering for each pixel and can be mathematically presented as [5]:

$$\gamma^T(f)(x) = \max\{k : x \in \Gamma^T(f_k)\}. \tag{3}$$

The attribute filtering can be applied to an image in a sequence of increasing criteria, that is, with progressively higher threshold values, building the AP. Considering a family of increasing criteria $T = \{T_\lambda : \lambda = 0, \ldots, l\}$, with λ a set of scalar values used as reference in the filtering procedure, and being T_0 true for all the CCs of an image f, the AP is composed of an attribute opening profile Π_{γ^T} and an attribute closing profile Π_{ϕ^T}, which are defined as follows:

$$\Pi_{\gamma^T}(f) = \Big\{\Pi_{\gamma^{T_\lambda}} : \Pi_{\gamma^{T_\lambda}} = \gamma^{T_\lambda}(f), \forall \lambda \in [0, \ldots, l]\Big\}, \tag{4}$$

$$\Pi_{\phi^T}(f) = \Big\{\Pi_{\phi^{T_\lambda}} : \Pi_{\phi^{T_\lambda}} = \phi^{T_\lambda}(f), \forall \lambda \in [0, \ldots, l]\Big\}, \tag{5}$$

with γ^{T_λ} and ϕ^{T_λ} denoting the morphological attribute opening and closing, respectively. The AP is defined as the concatenation of Eq. (4) and Eq. (5), including the image itself:

$$\text{AP}(f) = \Big\{\Pi_{\phi^T}(f), f, \Pi_{\gamma^T}(f)\Big\}. \tag{6}$$

The AP can be extended to multidimensional images by applying Eq. (6) to each band of the image. This approach is known as Extended Attribute Profile (EAP) and it is defined as follows:

$$\text{EAP}(f) = \Big\{\text{AP}(f)_1, \text{AP}(f)_2, \ldots, \text{AP}(f)_r\Big\}, \tag{7}$$

with r the number of components (bands) of the multidimensional image f.

 ISBN: 978-84-697-7861-6

3 Attribute Profile Implementation on GPU

This section describes the implementation of the attribute profiles on GPU based on the max-tree representation.

The attribute filtering algorithm presented in this work for building the attribute profiles on GPU is based on the max-tree representation of the image. The proposal does not build the max-tree and avoids a recursive flooding procedure and the use of priority queues. In addition, we have implemented the connected component labeling (CCL) algorithm on GPU proposed in [6] and adapted it for greyscale images. CCL is used for creating the initial set of connected components used for traversing the max-tree.

Algorithm 1 shows the pseudocode for computing the AP on GPU proposed in this work, which requires a greyscale image f and a set of scalar values T as input parameters. The algorithm computes the attribute opening profile Π_{γ^T} (line 2), the attribute closing

Algorithm 1 Attribute Profile on GPU.

Require: Input image f and a set of scalar values $T = \{T_\lambda : \lambda = 0, \ldots, l\}$
Ensure: Attribute Profile on GPU: AP(f)

1: **procedure** AttributeProfile(f, T)
2: $\quad \Pi_{\gamma^T} \leftarrow$ AttributeOpeningProfile(f, T)
3: $\quad f^c = 255 - f$ ▷ Complement the input image
4: $\quad \Pi_{\phi^T} \leftarrow$ AttributeOpeningProfile(f^c, T)
5: $\quad \Pi_{\phi^T} = 255 - \Pi_{\phi^T}$ ▷ Complement the result
6: $\quad \mathrm{AP} = \{\Pi_{\phi^T}, f, \Pi_{\gamma^T}\}$
7: **end procedure**

profile Π_{ϕ^T} (line 4) and concatenates them with the image itself (line 6), as defined by Eq. (6).

The implementation used for Π_{γ^T} is also used for computing Π_{ϕ^T}, simply complementing the input image f and then complementing the result, lines 3 and 5, respectively. Accordingly, the same kernels are used for computing the attribute opening and attribute closing profiles on GPU without further modifications.

The pseudocode of the function *AttributeOpeningProfile* in Algorithm 1 is shown in detail in Algorithm 2. The kernels are placed between <> symbols in the pseudocode. The max-tree traversal is carried out in kernels *MergeRegions* and *FindRoots*, and the attribute profile is built in kernels *InitializeAtt*, *UpdateAtt* and *AttProfile*. The pseudocode also includes the Tex, Con, GM and SM acronyms to indicate the use of the texture, constant, global and shared memory spaces on GPU, respectively. The output of Algorithm 2 represents the attribute opening profile Π_{γ^T}.

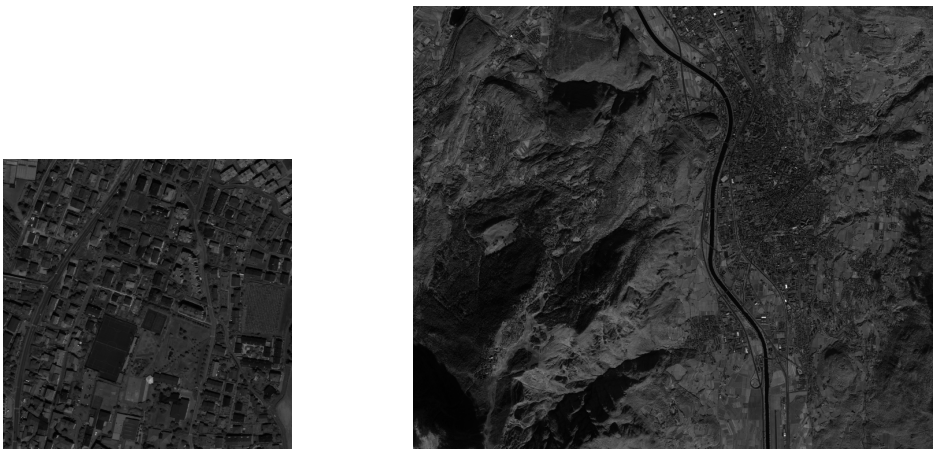
 ISBN: 978-84-697-7861-6

Algorithm 2 Attribute Opening Profile on GPU.

Require: Input image f and a set of scalar values $T = \{T_\lambda : \lambda = 0, \ldots, l\}$
Ensure: Attribute opening profile on GPU: $\Pi_{\gamma^T}(f) = \gamma^{T_\lambda}(f), \forall \lambda \in [0, \ldots, l]$

1: **procedure** AttributeOpeningProfile(f, T)
2: $\Pi_{\gamma^T} \leftarrow f$; $T_\lambda \leftarrow T$ ▷ Initial CPU-GPU transfer
3: H $\leftarrow$ <histogram>(Π_{γ^T}) ▷ GM
4: i = H.length() $- 1$; h = H[i]; $root$ = H[0] ▷ Levels of the max-tree
5: `CCL` $\leftarrow$ <componentLabeling>($\Pi_{\gamma^T}, root$) ▷ GM,SM
6: `att` $\leftarrow$ <InitializeAtt>(`CCL`) ▷ GM
7: **while** $h > root$ **do**
8: `att` $\leftarrow$ <UpdateAtt>(`CCL`) ▷ GM
9: ▷ Tex,Con,GM
10: $\Pi_{\gamma^T} \leftarrow$ <AttProfile>(Π_{γ^T}, `CCL`, `att`, h, T_λ)
11: **do**
12: ▷ Tex,GM
13: <MergeRegions>(Π_{γ^T}, `CCL`, h, $h-1$)
14: <FindRoots>(`CCL`) ▷ GM
15: **while** ($oldRoot \neq newRoot$)
16: $h \leftarrow$ H[i]; $i = i - 1$
17: **end while**
18: **end procedure**

 ISBN: 978-84-697-7861-6



(a) (b)

Figure 1: Remote sensing images used in the experiments: (a) dataset-1, a panchromatic image, and (b) dataset-2, a multispectral image (the fourth band is shown in the figure).

4 Experimental Results

This section presents the results for building and processing attribute profiles on GPU for two different remote sensing images in terms of execution time and speedup over a CPU implementation. The construction of the EAPs is evaluated as well as a final supervised classification stage by SVM executed over the generated EAPs.

In the following, we describe the experimental setup, the remote sensing images as well as the performance of the GPU implementation.

As the performance of the attribute profiles depends mainly on the size of the image, as well as on the number and size of its regions, two different remote sensing images, taken from the city of Trento, Italy, have been used in the experiments. The first image, labeled as dataset-1, is a subset of 900×900 pixels of a panchromatic image acquired by the Quickbird scanner. This is a one-band image. The second image is a Geoeye satellite multispectral image of 4700×4000 pixels with four spectral bands and named dataset-2. These images are shown in Figure 1.

The attribute profiles built from dataset-1 are based on the attributes of area (AP_a) and standard deviation (AP_s). For the AP_a the λ values considered for thresholding the area attribute are $49, 169, 361, 625, 961, 1369, 1849, 2401$, as suggested in [5]. For the standard deviation attribute the set of values $10, 20, 30$ are used for filtering the regions of the image. For dataset-2, we used the values $49, 169, 361$ for the area and the values $10, 20, 30$ for the standard deviation. In all the cases a connectivity of four was considered for computing

 ISBN: 978-84-697-7861-6

Image	dataset-1	dataset-2
Image size	$900 \times 900 \times 1$	$4700 \times 4000 \times 4$
EAP_a size	$900 \times 900 \times 17$	$4700 \times 4000 \times 28$
EAP_s size	$900 \times 900 \times 7$	$4700 \times 4000 \times 28$

Table 1: Spatial and spectral dimensions (rows × columns × bands) for each image, and the size of the AP and EAP based on area and standard deviation.

the connected components of each dataset. The spatial dimensions of the images and the resulting size of the EAPs, calculated as indicated in Eq. (6) and Eq. (7), are shown in Table 1.

The supervised classification is carried out by a SVM classifier using on CPU the library for SVM (LIBSVM) [7]. In the case of GPU a multi-class implementation for classification of n-dimensional images is executed [8]. A Gaussian radial basis function (RBF) is used as the activation function with the best parameters (C, γ) for training the SVM determined by 5-fold cross-validation, in the range $C = [1, 4, 16, 64, 128], \gamma = [0.5, 0.25, 0.125, 0.0625]$, for each image.

The experiments are executed on a personal computer with a 2.80 GHz Intel Core i7 microprocessor with 16 GB of RAM and a GTX TITAN GPU based on the Pascal architecture with 12 GB of global memory. The EAPs on GPU is implemented using CUDA 8.0. The execution times on GPU include memory allocation and input/output data transfers from CPU to GPU and viceversa. They are the average of 20 individual executions. The input data is normalized between 0 and 255 before the generation of the attribute profiles and between 0 and 1 before the classification by SVM.

Table 2 and Table 3 show the execution times required to compute the the EAPs for dataset-1 and dataset-2, respectively. These tables also include the time for the supervised classification by the SVM. In the case of dataset-1, as this is a one-band image, the EAPs would be the same as the APs for one band. For dataset-2, Algorithm 1 is executed over each one of the four spectral bands of the image.

It can be observed in Table 2 that the execution time is reduced by 18.6× building the EAP_a on GPU for dataset-1. In Table 3, a higher speedup is observed for the EAP_a for dataset-2, with a speedup of 33× over the CPU execution. In particular, for this last image the computation time is reduced from 5 minutes on CPU down to less than 10 seconds on GPU. The reason for a higher speedup is the bigger size of this dataset that better exploits the parallel computing capabilities of the GPU.

 ISBN: 978-84-697-7861-6

	Area		Standard deviation	
	EAP_a	SVM	EAP_s	SVM
CPU	4.174s	25.802s	4.273s	29.755s
GPU	0.224s	0.710s	0.795s	0.347s
Speedup	18.6×	36.3×	5.4×	85.7×

Table 2: Execution times (in seconds) on CPU and GPU for the EAP construction and classification by SVM for dataset-1.

	Area		Standard deviation	
	EAP_a	SVM	EAP_s	SVM
CPU	308.237s	1515.338s	319.496s	2095.109s
GPU	9.287s	61.909s	74.486s	109.449s
Speedup	33.0×	24.5×	4.3×	19.1×

Table 3: Execution times (in seconds) on CPU and GPU for the EAP construction and classification by SVM for dataset-2.

5 Conclusion

In this paper a CUDA GPU implementation of the construction of extended attribute profiles for remote sensing multispectral images is presented. The implementation is based on the max-tree representation of the image. The proposal does not build explicitly the max-tree and a connected component labelling algorithm is used for creating the intial set of connected components used for traversing it. The attributes considered in the experiments are area and standard deviation although the approach could be extended to other attributes. The experiments are performed over a panchromatic and a four-band multispectral images obtaining good speedups with a reduction from minutes to tens of seconds in execution times over the corresponding CPU implementations.

Acknowledgements

This work was supported in part by the Consellería de Cultura, Educación e Ordenación Universitaria [grant numbers GRC2014/008 and ED431G/08] and Ministry of Education, Culture and Sport, Government of Spain [grant number TIN2016-76373-P]. Both are co–funded by the European Regional Development Fund (ERDF).

 ISBN: 978-84-697-7861-6

References

[1] M. Fauvel, Y. Tarabalka, J.A. Benediktsson, J. Chanussot, and J.C Tilton, *Advances in spectral-spatial classification of hyperspectral images*, Proceedings of the IEEE, vol. 101, no. 3, pp. 652-675, 2013.

[2] P. Ghamisi, M.D. Mura, and J.A. Benediktsson, *A survey on spectralspatial classification techniques based on attribute profiles*, IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 5, pp. 2335-2353, 2015.

[3] E. Christophe, J. Michel, and J. and Inglada, *Remote sensing processing: From multicore to GPU*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 4, no. 3, pp. 643-652, 2011.

[4] J. López-Fandiño, P. Quesada-Barriuso, D.B. Heras, and F. Argüello, *Efficient ELM-Based Techniques for the Classification of Hyperspectral Remote Sensing Images on Commodity GPUs*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2884-2893, 2015.

[5] M.D. Mura Mura, J.A. Benediktsson, B. Waske,and L. Bruzzone, *Morphological Attribute Profiles for the Analysis of Very High Resolution Images*, IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 10, pp. 3747-3762, 2010.

[6] V. Oliveira, and R. de Alencar Lotufo, *A Study on Connected Components Labeling algorithms using GPUs*, Conference on Graphics, Patterns and Images, Proc. 23rd SIBGRAPI, 2010.

[7] C.-C. Chang, and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 27:1-27:27, 2011.

[8] P. Quesada-Barriuso, F. Argello, D.B. Heras, and J.A. Benediktsson, *Wavelet-Based Classification of Hyperspectral Images Using Extended Morphological Profiles on Graphics Processing Units*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2962-2970, 2015.

 ISBN: 978-84-697-7861-6