Proceedings of the 16th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2016
4–8 July, 2016.

Efficient Implementation of Morphological Index for Building/Shadow Extraction from Remotely Sensed Images

Luis Ignacio Jiménez[1], Javier Plaza[1] and Antonio Plaza[1]

[1] *Hyperspectral Computing Laboratory, Department of Computer Technology and Communications, University of Extremadura*

emails: luijimenez@unex.es, jplaza@unex.es, aplaza@unex.es

Abstract

Morphological building index (MBI) and morphological shadow index (MSI) are recently developed techniques that aim at automatically detect buildings/shadows using high resolution remotely sensed imagery. Traditional mathematical morphology operations are usually time-consuming as they are based in the consideration of a wide range of image-object properties such as brightness, contrast, shapes, sizes and in the application of series of repeated transformations (e.g. classical opening and closing operators). In the case of MBI and MSI, the computational complexity is also increased due to the use of multiscale and multidirectional morphological operators. In this paper, we provide a computationally efficient implementation of MBI and MSI algorithms which is specifically developed for commodity graphic processing units (GPUs) using Nvidia CUDA. We perform the evaluation of the parallel version of the algorithms using two different NVidia architectures and three widely used hyperspectral datasets. Experimental results show that the computational burden introduced when considering multidirectional morphological operators can be almost completely removed by the developed implementations.

Key words: mathematical morphology, high resolution, remotely sensed imagery, graphic processing units (GPUs).

1 Introduction

The efficient and precise location/identification of buildings is an increasingly important task for a great part of the most developed countries of the world, as it provides crucial information for population estimation and territorial planning [1]. This is possible due to the

availability of high-resolution Earth Observation (EO) instruments that now provide almost complete spatial information about the surface of the Earth that can be efficiently used to complement available spectral information [2]. This allows for an increase in the separability of spectrally similar classes. With this purpose, several sophisticated (supervised or semi-supervised) segmentation techniques have been developed for building extraction [3, 4, 5].

Recently, most efforts have been focused on the generation of a feature index that can be applied to building detection without the need for training data or complex segmentation processes. The morphological building/shadow index has been recently proposed with this aim [6]. The main idea of MBI is to relate the implicit characteristics of buildings (e.g. brightness, size and contrast) with morphological operators (e.g. top-hat reconstruction, granulometry and directionality).

As mentioned before, while integrated spatial/spectral developments hold great promise for Earth science image analysis, they clearly introduce new processing challenges that, combined with the complex and large size of EO datasets, limits the possibility of utilizing those algorithms in time-critical applications [7, 8]. Particularly, the use of multiscale and multidirectional morphological operators introduces a significant computational burden in MBI algorithm [9] which can be alleviated if parallel implementations are developed.

Even though EO data processing algorithms map nicely to clusters and heterogeneous networks [10, 11], these systems are generally expensive and difficult to adapt to on-board data processing scenarios, in which low-weight and low-power integrated components are essential to reduce mission payload where field programmable gate arrays (FPGAs) and graphic processing units (GPUs) can further provide a response in (near) real time, which is believed to be acceptable in many remote sensing applications [12]. In this paper we present the first GPU-based parallel implementation of the MBI/MSI algorithm for EO data exploitation using the NVidia CUDA framework.

The remainder of the paper is organized as follows. Section II describes the original implementation of MBI and MSI algorithms and our proposed C implementation (optimized for memory usage). Section III briefly introduces GPU architectures and the NVidia CUDA framework, and further describes the newly proposed GPU implementation for MBI and MSI algorithms. Section IV evaluates the proposed GPU implementations in terms of building/shadow detection accuracy and computational performance. Section V concludes this paper with some remarks and hints at plausible future research lines.

2 Morphological Building/Shadow Index

Morphological building/shadow index is based on the construction of a relationship between the implicit characteristics of buildings (e.g. brightness, size and contrast) with morphological operators (e.g. top-hat reconstruction, granulometry and directionality). Some of the basics of this relationship are introduced below.

- Brightness. We can define the brightness of a pixel as the maximum value of the pixel at all the contained spectral bands. Building areas are characterized by high brightness scores while shadow areas brightness should be smaller due to their low spectral reflectance. White/black top-hat transformation will be used to point out bright/dark structures with a determined size in order to identify buildings/shadows.
- Contrast. Building areas are generally characterized by their high contrast, due to the difference between the spectral reflectance values of roof and spatially adjacent shadows. The case of shadows is exactly the opposite (high contrast between shadows and the neighboring areas) The MBI algorithm is able to characterize the local contrast of buildings extracting the differential morphological profiles (DMP) of the white top-hat transformed data. MSI algorithm relies on the extraction of the DMP over the black top-hat transformed data.
- Shape. The most widely adopted shape descriptor for building areas is the rectangle. Therefore, the length-width ratio can be used to filter out structures with similar spectral response.
- Size and directionality. In order to assist with the removal of spectrally similar structures, a series of linear structuring elements (SEs), designed to measure the size and directionality of structures, is implemented in both MBI and MSI.

Based on the above concepts, we can describe the steps of the MBI algorithm as follows (the main differences betwee MSI and MBI are also highlighted):

1. Calculation of the brightness. Using all the spectral bands of the considered image, we select the brightest component of each pixel as follows:

$$b(x) = \max_{1<=k<=K} band_k(x), \tag{1}$$

where x is the pixel, K is the number of components of the pixel's spectral signature, and $band_k(x)$ is the pixel value in the k-th band.

2. The white top-hat by reconstruction for MBI and the black top-hat by reconstruction for MSI (2). White top-hat is then computed to highlight bright structures:

$$W - TH(d, s) = b - \gamma_b^{re}(d, s), \tag{2}$$

while the black top-hat aims at highlighting dark structures:

$$B - TH(d, s) = \varphi_b^{re}(d, s) - b, \tag{3}$$

where γ_b^{re} represents the opening by reconstruction, φ_b^{re} represents the closing by reconstruction, and d denotes the size and directionality of the linear SE.

 ISBN: 978-84-608-6082-2

3. The morphological profiles (MPs) of the white top-hat are now defined as:

$$\begin{cases} MP_{W-TH}(d,s) = W - TH(d,s) \\ MP_{W-TH}(d,0) = b \end{cases} \tag{4}$$

4. To complete the calculation of the MBI and MSI we need to define the differential morphological profiles (DMP) as:

$$DMP_{W-TH}(d,s) = |MP_{W-TH}(d,s+\Delta s) - MP_{W-TH}(d,s)|, \tag{5}$$

$$DMP_{B-TH}(d,s) = |MP_{B-TH}(d,s+\Delta s) - MP_{B-TH}(d,s)|, \tag{6}$$

where Δs is the interval of the profiles between and $s_{min} <= s <= s_{max}$.

5. The MBI and MSI are defined as the average of the DMP of the white top-hat and the black top-hat, respectively:

$$MBI = \frac{\sum DMP_{W-TH}(d,s)}{D * S}, \tag{7}$$

$$MSI = \frac{\sum DMP_{B-TH}(d,s)}{D * S}, \tag{8}$$

where D is the number of the directions applied to the linear SE and $S = ((s_{max} - s_{min})/\Delta s) + 1$. Buildinds and shadows are respectively represented by larger values in each index.

Algorithm 1 provides a pseudode description of the original MBI/MSI algorithm implemented in Matlab. When calculating the black top-hat and white top-hat, it should be noticed that opening and closing by reconstruction morphological operations are complementary.

For the C version we introduce some changes in order to optimize the code to the new language (specially focusing on the memory management). In the Matlab version, iteration t partially repeats operations from the previous iteration $(t-1)$ oriented to the white top-hat and black top-hat creation, while in the C implementation, we preserve the operations already executed in previous iterations due to the absence of memory restrictions. Besides, we eliminate the creation of the linear SE by applying the direction of it to the erosion morphological operator. The resulting pseudocode of the C implementation is provded in Algorithms 2 and 3. Using this latest version as a reference, we have developed a CUDA version that implements each step as a CUDA kernel. This version is described in the following section.

 ISBN: 978-84-608-6082-2

Algorithm 1 Pseudocode of **Morphologic Building Index** and **Morphologic Shadow Index** in Matlab code

```
image = ReadImage
img = CalculationBrightest/DarkestScene(image)
for s = smin:Δs:smax
   for dir = 1:1:D
         se = CreateStructureElement(dir)
         a = W-TH(se,s,img) / B-TH(se,s,imgC)
         b = W-TH(se,s+Δs,img) / B-TH(se,s+Δs,imgC)
         DMP(dir) = b - a
   end
end
DMP = DMP/(D*S)
MBI/MSI = ∑DMP(dir)
Scale(MBI,0,1)
```

3 Parallel Implementation

GPUs can be understood in terms of a stream model, under which all data sets are represented as streams (i.e., ordered data sets), and each of them is processed by a multiprocessor, which means that a GPU also can be seen as a set of multiprocessors (MPs). Each multiprocessor is characterized by a single instruction multiple data (SIMD) architecture. Each processor has access to a local shared memory and also to local cache memories in the multiprocessor, while the multiprocessors have access to the global GPU (device)memory. Algorithms are constructed by chaining so-called *kernels* which operate on entire streams and are executed by a multiprocessor, taking one or more streams as inputs and producing one or more streams as outputs. The kernels can perform a kind of batch processing arranged in the form of a *grid* of *blocks*, where each block is composed by several *threads* which share data efficiently through the shared local memory and synchronize their execution for coordinating accesses to memory. As a result, there are different levels of memory in the GPU for the thread, block and grid concepts. There is also a maximum number of threads that a block can contain but the number of threads that can be concurrently executed is much larger (several blocks executed by the same kernel can be managed concurrently, at the expense of reducing the cooperation between threads since the threads in different blocks of the same grid cannot synchronize with the other threads). Our GPU implementation of MBI is based in the following kernels.

BrightnessImage: this kernel implements the first step of MBI and MSI algorithm, where the brightness is calculated according to equation 1. In the case of MSI, a final step to complement the returned structure is performed. The number of threads is set to

 ISBN: 978-84-608-6082-2

Algorithm 2 Pseudocode of **Morphologic Building Index** in C code

```
image = ReadImage
img = CalculationBrightestScene(image)
for s = smin:Δs:smax
   for dir = 1:1:D
         erode=Erosion(dir, img)
         recon=Reconstruction(erode,img)
         wth1(dir) = img - recon
         if(s != smin)
            DMPWTH(dir) += (wth2(dir)-wth1(dir))
         else
            DMPWTH(dir) = 0
         wth2(dir) = wth1(dir)
   end
end
MBI = ∑DMPWTH(dir)
Scale(MBI,0,1)
```

the maximum allowed by the device and the number of blocks equals the number of pixels divided by the number of threads.

ErodeOperator: the morphological erosion operator is implemented using different kernels depending on the direction of the linear SE being applied. The original work explains that the number of directions is set to four (NE, N, NW and W) because changing to an eight-connected neighborhood resulted in a similar outcome with a significant computational time increase. As result, one kernel computes the erosion for the four directions storing the erode images consecutively in memory. This kernel uses a two-dimensional grid setting the x-dimension to the number of lines and the y-dimension to the number of samples both divided by the block size for each dimension, which is the same making a square block of size 32.

Reconstruction: this step performs the morphological reconstruction using two kernels that performs the raster scan (*x_forward*) and the antiraster scan (*x_backward*) of the four directions erode images; *x_forward* finds the maximum value within the NE, N, NW, W neighbors and the origin pixel from the top-left to the bottom-right of the image; and *x_backward* computes the maximum value within the E, SE, S, SW neighbors and the origin pixel from the bottom-right to the top-left of the image. The same way as the erosion kernel, the reconstruction is performed for the four directions considered in the same call. Both kernels set the number of block to the number of samples divided by a number of threads empirically set to 32, in order to maintain a balance between blocks and threads.

Algorithm 3 Pseudocode of **Morphologic Shadow Index** in C code

```
image = ReadImage
img = CalculationDarkestScene(image)
for s = smin:Δs:smax
   for dir = 1:1:D
         erode=Erosion(dir, img)
         recon=Reconstruction(erode,img)
         wth1(dir) = recon - img
         if(s != smin)
            DMPWTH(dir) += (wth2(dir)-wth1(dir))
         else
            DMPWTH(dir) = 0
         wth2(dir) = wth1(dir)
   end
end
MBI = ∑DMPWTH(dir)
Scale(MSI,0,1)
```

Substraction: this stage computes, in one kernel called *substract*, the difference between consecutive iterations reconstructed images, acumulating the results to perform the average in a subsequent step. Once the iterave process is finished, other kernel performs the averaging of the results based in the number of iterations between the minimum,*smin*, and maximum, *smax*, structure size values used.

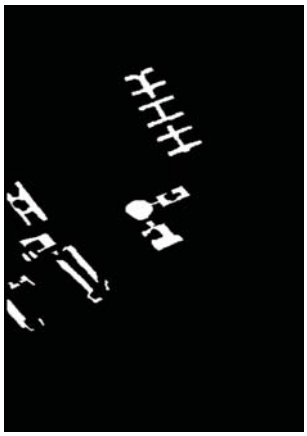
4 Experimental validation

4.1 Hyperspectral data and hardware architectures

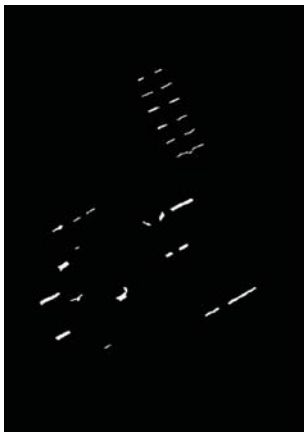
Our experiments have been carried out using three different hyperspectral images. The first considered hyperspectral image is the well-known Pavia University hyperspectral dataset (Figure 1a), acquired by the ROSIS optical sensor during a flight campaign over the urban area of the University of Pavia, Pavia, Italy. The original Pavia University dataset consists on $610 \times 340 pixels$, with high spatial resolution of 1.3 m per pixel. The number of data channels in the acquired image is 103 (with the spectral range from 0.43 to 0.86 μm). Nine thematic land-cover classes are available, from which we select metal sheets, self-blocking bricks and bitumen to generate the class building (see Figure 1b). In addition, a shadow class is also provided in the ground-truth information (see Figure 1c).

The second hyperspectral dataset used was acquired by the NSF-funded Center for Airborne Laser Mapping (NCALM) over the University of Houston campus and its neighboring

(a) Pavia University image



(b) Building ground-truth



(c) Shadow ground-truth

Figure 1: Pavia University hyperspectral dataset. (a) False color composition of the Pavia University image. (b) Reference spatial distribution of the buildings. (c) Reference spatial distribution of the shadow class.

area. This hyperspectral data has 144 spectral bands in the 380-1050 nm spectral region and spatial resolution of 2.5 m. The image size in pixels is 349 $\times$ 1905. Figure 2a shows a false color composite of the image. Ground-truth information is available as 15 different land-cover classes. The building groud-truth has been generated by the fusion of residential and commercial original land-cover classes.

The last hyperspectral image scene used for experiments in this work was collected by the AVIRIS sensor, which was flown by NASAs Jet Propulsion Laboratory over the World Trade Center area in New York City on 16 September 2001, just 5 days after the terrorist attacks that collapsed the two main towers and other buildings in the WTC complex. The selected subset consists of 500 $\times$ 1600$pixels$, 224 spectral bands, and a total size of (approximately). The spatial resolution is 1.7 m/pixel. Extensive reference information, collected by U.S. Geological Survey (USGS), is available for the WTC scene.

We have used two different computer architectures for the experimental validation of the proposed approaches: a compute cluster with 44 NVidia TESLA S2070 GPU nodes (2 M2075 per node), each with an Intel Xeon CPU E5645 at 2.40 GHz and a total of 24 GB of RAM, divided in 12 modules of 2 GB each (hereinafter Architecture 1) and a desktop computer (Intel i7 920 CPU at 2.67 GHz and 6 GB of RAM) with an NVidia GTX 580 GPU equipped with 512 processor cores operating at 1.54 GHz and 1536 MB of RAM memory (hereinafter Architecture 2).

[1] http://www.ceta-ciemat.es

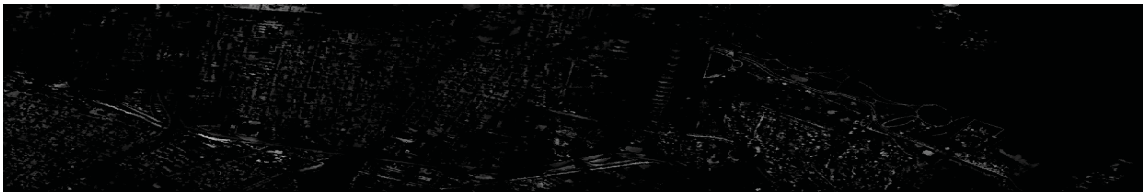
(a) Houston campus image

(b) Building ground-truth

Figure 2: Houston campus hyperspectral dataset. (a) False color composition of the Houston campus image. (b) Reference spatial distribution of the buildings.

(a) World Trade Center image

(b) Building Index



(c) Shadow Index

Figure 3: False color composition of the World Trade Center hyperspectral dataset acquired by the AVIRIS instrument.

 ISBN: 978-84-608-6082-2

4.2 Analysis of algorithm precision

In this section we will focus on analyzing the parallel MBI and MSI implementations using the two datasets with groud-truth information about building and shadows. Particularly, Table 1 shows results based on the generation of a binary image applying different threshold values ($th1 = 1/255$, $th2 = 50/255$, $th3 = 100/255$, $th4 = 150/255$ and $th5 = 200/255$) over the MBI and MSI estimation images over Pavia University and Houston datasets (we remove all the pixels with a value below the thereshold, considering the range of 0 to 255 based on the original RGB algorithms scale). Then, we calculate the MBI/MSI values (being 100 the best case and 0 the worst) by comparing the each thresholded image with the ground-truth. It can be seen that the results obtained by the CPU and GPU implementations are almost the same (the slight differences are due to the removal of the queue structure, which seems to benefit the parallel implementations). It should be noticed that no ground-truth information is available for the shadow class in the Houston dataset and, therefore, no precision results can be shown for this particular case.

		Pavia Univ. (610 × 340)					Houston Univ. (349 × 1905)				
Algorithm	Implementation	th1	th2	th3	th4	th5	th1	th2	th3	th4	th5
MBI	CPU	100	18,79	10,68	2.61	0	100	5.85	4.10	1.31	0
	GPU	100	18,24	10,45	2,84	0	100	6,68	4.57	1,79	0
MSI	CPU	100	97.36	47.94	19.64	3.69	–	–	–	–	–
	GPU	100	97,99	50,05	18,80	6.44	–	–	–	–	–

Table 1: Mean execution time (in seconds) for the CPU and GPU implementations along with the obtained speedup after 10 Monte-Carlo runs over each of the considered architectures over the two processed hyperspectral datasets with available groundtruth.

4.3 Analysis of parallel performance

Table 2 shows the obtained speedups in the two considered architectures for the three selected scenes. The results obtained by the Architecture 2 are better due to the fact that the NVidia TESLA S2070 includes error checking and correction (NVidia GTX 580 does not include this characteristic) that guarantees more stable results at the expense of a slightly reduced performance. The time taken by data transfers between the CPU and the GPU is included in the execution times. As it can be seen, speedups around $\times 5$ can be achieved when considering the two large datasets. It is important to emphasize that parallel implementation is able to overlap the processing of the four considered directions, thus providing a significant performance improvement with regards to the serial implementation.

		Pavia Univ. (610 × 340)			Houston Univ. (349 × 1905)			WTC (500 × 1600)		
	Hardware	CPU time	GPU time	Speedup	CPU time	GPU time	Speedup	CPU time	GPU time	Speedup
MBI	Architec. 1	0.794	0.508	x1.563	2.735	0.608	x4.498	3.850	0.886	x4.345
	Architec. 2	0.695	0.426	x1.632	2.575	0.496	x5.188	3.537	0.708	x4.995
MSI	Architec. 1	0.809	0.503	x1.608	2.792	0.605	x4.615	3.920	0.892	x4.395
	Architec. 2	0.699	0.427	x1.639	2.625	0.494	x5.314	3.556	0.704	x5.050

Table 2: Mean execution time (in seconds) for the CPU and GPU implementations of MBI and MSI along with the obtained speedup after 10 Monte-Carlo runs over each of the considered architectures over the three considered hyperspectral datasets.

5 Conclusions and future reserch lines

In this paper we have presented a GPU implementation of MBI/MSI algorithms for building/shadow detection in high resolution remote sensing images. The implementation are based in optimized implementations developed in C, which reduce the amount of memory required. In addition, an efficient raster image processing scheme is implemented on the GPU. As a result, we achieve independence between the execution time of the parallel implementation and the number of considered directions when applying the MBI/MSI multidirectional morphological operators. In our experiments, four different directions have being processed simultaneously in the GPU implementation, achieving speedups over $\times 5$ for some of the considered images. Future research lines will focus on improving both the accuracy and the computational performance of the proposed approaches. We will also explore the use of FPGAs as a specialized device with low power consumption and onboard processing capabilities in order to accelerate the MBI/MSI algorithms.

Acknowledgements

This work has been supported by Junta de Extremadura (decreto 297/2014, ayudas para la realización de actividades de investigación y desarrollo tecnológico, de divulgación y de transferencia de conocimiento por los Grupos de Investigación de Extremadura, Ref. GR15005). This work was supported by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETA-CIEMAT), funded by the European Regional Development Fund (ERDF). CETA-CIEMAT belongs to CIEMAT and the Government of Spain.

References

[1] A. Marinoni and P. Gamba, "Accurate detection of anthropogenic settlements in hyperspectral images by higher order nonlinear unmixing," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 9, no. 5, pp. 952–961, 2016.

[2] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, pp. 2025–2041, 2002.

[3] M. Pesaresi and A. Gerhardinger, "Improved textural built-up presence index for automatic recognition of human settlements in arid regions with scattered vegetation," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 4, pp. 16–26, 2011.

[4] J. Plaza, A. Plaza, P. Gamba, and G. Trianni, "Efficient multi-band texture analysis for remotely sensed data interpretation in urban areas," in *Urban Remote Sensing Joint Event (IEEE URS2007), Paris, France*, 2007.

[5] F. Dell'Acqua, P. Gamba, A. Ferrari, J. A. Palmason, J. A. Benediktsson, and K. Arnason, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 4, pp. 322–326, 2010.

[6] X. Huang and L. Zhang, "A multidirectional and multiscale morphological index for automatic building extraction from multispectral geoeye-1 imagery," *Photogramm. Eng. Remote Sens.*, vol. 77, pp. 721–732, 2011.

[7] J. Plaza, A. Plaza, and C. Barra, "Multi-channel morphological profiles for classification of hyperspectral images using support vector machines," *Sensors*, vol. 9, pp. 196–218, 2009.

[8] J. Delgado, G. Martin, J. Plaza, L. I. Jimenez, and A. Plaza, "Fast spatial preprocessing for spectral unmixing of hyperspectral data on graphics processing units," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 9, pp. 952–961, 2016.

[9] X. Huang and L. Zhang, "Morphological building/shadow index for building extraction from high-resolution imagery over urban areas," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 5, pp. 161–172, 2012.

[10] A. Plaza, J. Plaza, and H. Vegas, "Improving the performance of hyperspectral image and signal processing algorithms using parallel, distributed and specialized hardware-based systems," *signal*, vol. 50, pp. 293–315, 2010.

[11] A. Plaza, J. Plaza, and A. Paz, "Parallel heterogeneous cbir system for efficient hyperspectral image retrieval using spectral mixture analysis," *Concurrency Computat. Pract. Exper.*, vol. 22, no. 9, pp. 293–315, 2010.

[12] A. Plaza, J. Plaza, A. Paz, and S. Sanchez, "Parallel hyperspectral image and signal processing," *IEEE Signal Process. Mag.*, vol. 28, pp. 196–218, 2011.