



# Revista Brasileira de Geografia Física

Homepage: <https://periodicos.ufpe.br/revistas/rbgf>



## The Surface Energy Balance Algorithm for Land (SEBAL) framework in Graphics Processing Units (GPU) using Cuda and OpenCL

Raphael de Souza Rosa Gomes<sup>1</sup>, Victor Hugo de Moraes Danelichen<sup>2</sup>, Marcelo Sacardi Biudes<sup>1</sup>, Maísa Caldas Souza<sup>1</sup>, Josiel Maimome de Figueiredo<sup>1</sup>, José de Souza Nogueira<sup>1</sup>

<sup>1</sup>Física Ambiental, Universidade Federal de Mato Grosso, Cuiabá, Brasil, CEP 78068-600. (65) 3615-8738. [thesivis@gmail.com](mailto:thesivis@gmail.com). <sup>2</sup>Professor doutor e Pesquisador, Mestrado em Ciências Ambientais, Universidade de Cuiabá, Campus Barão, 222, CEP 78005-300, Cuiabá, Mato Grosso. (65) 3363-1729. [danelichen@fisica.ufmt.br](mailto:danelichen@fisica.ufmt.br) (autor correspondente).

Artigo recebido em 07/05/2020 e aceito em 23/04/2021

### ABSTRACT

Evapotranspiration (ET) is the second largest component of the terrestrial hydrological cycle on a global scale. The estimation of ET at regional scale using satellite imagery and algorithms allow the conversion of instantaneous measures in daily totals which represents a major contribution to the study of energy exchange between the biosphere and atmosphere. One of the most important remote sensing algorithms used to estimate the spatial partitioning of energy is the SEBAL - Surface Energy Balance Algorithm for Land. The time processing of this algorithm can be very high depending on the size of the image. Thus, the objective of this study was to propose a framework of the use of graphic processing units (GPU) to run the SEBAL to increase the performance and reduce the time required to run it. For implementation of SEBAL's framework was used local data collected in a micrometeorological tower installed in an area located in the Northern Pantanal (Brazil) and Landsat 5 - TM image. We used the Model Maker function of the ERDAS 9.2 Software to run SEBAL algorithm, and its implementation in Java, CUDA and OpenCL languages. A daily ET estimated by SEBAL in JAVA, CUDA and OpenCL languages were similar to the value performed by ERDAS software. The daily ET computed by the ERDAS software was 10% lower and the SEBAL computed in JAVA language, CUDA and OpenCL was 9% lower than the ET obtained by the Bowen ratio method. The results showed that CUDA was better due to the Speed Up performance up to 35,000 times.

Keywords: Wetland, evapotranspiration estimation, hyper seasonal savanna.

### Introduction

The water demand is growing rapidly in the world, suppressing their supply, which challenges its availability and food production (Vörösmarty et al., 2010; Burek et al., 2016). Agricultural activities depending directly on water consumption for its development affect population growth and its demands on food, which is competing with the water supply in industries, homes and environmental (Vörösmarty et al., 2010; FAO, 2018). It is estimated an average rate of 4.7% growth of the agricultural area in Brazil per year over the next decade, one of the largest in the world, and much of that growth is in the Cerrado, Pantanal and Amazon (Menke et al., 2009; Soares et al., 2020).

The evapotranspiration (ET) has been the best indicator to assess the water consumption and water use for human activities (Leivas et al., 2011; Angelini et al., 2017). The ET of land surface is an important component of energy and water balance.

It is the second largest component of the terrestrial hydrological cycle on a global scale, accounting for about 60% of the precipitation from the surface to the atmosphere (Mu et al., 2011). In addition, ET is an important energy flow, which uses more than half the total solar energy absorbed by the surface (Mu et al., 2011). Therefore, the correct characterization of ET is crucial in the study of terrestrial ecosystem in climate dynamics and the hydrological cycle (Allen et al., 2011; Danelichen et al., 2020).

Recent studies have been demonstrated the possibility of estimating ET at regional scale through observations of remote sensing data combination with surface atmospheric data (Tian et al., 2013; Angelini et al., 2017). The perimeters of the ET on a regional scale, based on satellite images and algorithms that allow the conversion of instantaneous measurements in total daily evapotranspiration represents a major contribution to studies of energy exchange between the

biosphere and atmosphere (Biudes et al., 2009; Allen et al., 2011; Bégué et al., 2018).

One of the most useful remote sensing algorithms and widely used in studies of the partitioning of energy in soil-plant-atmosphere system is the Surface Energy Balance Algorithm for Land (SEBAL) (Bastiaanssen et al., 1998; Bastiaanssen, 2000; Glenn et al., 2007; Kalma et al., 2008; Allen et al., 2011; Bégué et al., 2018). This algorithm uses satellite images and a few surface data as input for estimating the net radiation and soil, sensible and latent heat fluxes. Once implemented and validated the use of these has the advantage of providing the energy balance at the effective and economically surface without the need of ground stations. Although remote sensing depends on the orbital sensor that feeds data channels, it provides a wide spatial coverage and good spatial resolution (Allen et al., 2011; Mu et al., 2011; Cardim et al., 2018). The SEBAL has been used to estimate the effective albedo, surface temperature, and surface emissivity as well as the energy balance components with Landsat - 5 (Allen et al., 2011; Mendonça et al., 2012; Ferreira Junior et al., 2013; Danelichen et al., 2020) and the MODIS sensor aboard the Terra satellite/Aqua (Mu et al., 2011; Ruhoff, 2011; Tian et al., 2013).

The processing time of SEBAL can be high, especially if we use a complete satellite image due to the number of pixels that an image contains can spend around 100 million pixels (image 12000 by 8600 pixels), generating a file at least 700 MB in size, so the time of the calculations can be unsatisfactory (Oliveira et al., 2018). The use of parallel processing (with hundreds of cores) performing tasks at the same time in computing the energy balance by remote sensing can significantly reduce the processing time than central processing units.

In this sense the objective of this research was to implement the use of graphics processing units (GPU) to compute the SEBAL algorithm to increase performance and reduce the computation time required.

## Methodology

### *Surface Energy Balance Algorithm for Land (SEBAL)*

The ET is obtained after conversion of the instantaneous value of the latent heat flux - LE ( $\text{W m}^{-2}$ ), in total ET (mm) in for 24 h in SEBAL (Bastiaanssen, 2000). The LE is obtained as the

residue from the classical equation of energy balance at the surface, given by Equation (1).

$$R_n = LE + H + G \quad (1)$$

where  $R_n$  is the net radiation, LE is the density of latent heat flux, H is the density of sensible heat flux, and G is the density of heat flow in the soil, all in  $\text{W m}^{-2}$ .

The net radiation at the surface ( $R_n$ ) is computed using the following Equation (2) of the radiation balance of surface radiation (Allen et al., 2002; Silva et al., 2005).

$$R_n = R_{s\downarrow}(1 - \alpha_{sup}) - R_{L\uparrow} + \varepsilon_0 R_{L\downarrow} \quad (2)$$

where  $R_{s\downarrow}$  is the incident shortwave radiation ( $\text{W m}^{-2}$ ),  $\alpha_{sup}$  is the albedo of each pixel (dimensionless),  $R_{L\downarrow}$  is the long wave radiation emitted by the atmosphere in the direction of each pixel ( $\text{W m}^{-2}$ ),  $R_{L\uparrow}$  is long wave radiation emitted by each pixel ( $\text{W m}^{-2}$ ) and  $\varepsilon_0$  is the emissivity of each pixel.

The heat flux on the ground (G) is the rate of heat storage in the soil and vegetation due to conduction. The G is calculated by Equation (3) using the following empirical equation developed by Bastiaanssen (2000) representing values nearby midday.

$$G = (T_s - 273.16) \times (0.0038 + 0.0074 \times \alpha) \times (1 - 0.98 \times \text{NDVI}^4) \times R_n \quad (3)$$

where  $T_s$  is the surface temperature ( $^{\circ}\text{C}$ ),  $\alpha$  is the surface albedo and NDVI index is the normalized difference vegetation. For the purpose of correction of the values of heat flow to the ground water bodies ( $\text{NDVI} < 0$ ) can be used the following expression:  $G = 0.3R_n$  used by Silva et al. (2006).

The sensible heat flux (H) is the rate of heat loss to the air by convection and conduction due to a difference in temperature. The H is estimated based on the wind speed and the surface temperature by using an internal calibration of the difference in temperature of the air (2 m) and surface temperature. The H is a function of the temperature gradient, surface roughness, and velocity of wind (Equation 4). To facilitate the computation, we used two pixels "anchors" (as the values of H can be predicted and estimated dT) and the wind speed at given height (Bastiaanssen et al., 1998; Mendonça et al., 2012).

$$H = \frac{\rho c_p dT}{r_a} \quad (4)$$

where  $\rho$  is air density ( $\text{kg m}^{-3}$ ),  $c_p$  is the specific heat of air ( $1004 \text{ J Kg}^{-1} \text{ K}^{-1}$ ), dT (K) is the temperature difference, and  $r_a$  is the aerodynamic resistance to heat transport ( $\text{s m}^{-1}$ ).

The ET is determined based on the instantaneous evaporative fraction ( $FE_i$ ), defined by Equation (5) as the ratio between LE and ( $R_n - G$ ). Studies have shown that  $FE_i$  is equals daily

evaporative fraction (FE) (Bastiaanssen et al., 2000; Allen et al., 2011).

$$FE_i = \left( \frac{LE}{Rn - G} \right) = FE_{24h} = \frac{LE_{24h}}{Rn_{24h}} \quad (5)$$

Assuming that daily soil heat flux ( $G_{24h} = 0$ ), the daily latent heat flux ( $LE_{24h}$ ) is estimated by Equation (6).

$$LE_{24h} = FE_i \times Rn_{24h} \quad (6)$$

in which  $Rn_{24h} = R_{s,24h} (1 - \text{albedo}) - 110 \times \tau_{24h}$  where  $R_{s,24h}$  is the daily short-wave radiation incident ( $W m^{-2}$ ) and the  $\tau_{24h}$  is the daily transmissivity (Allen et al. 2002). The conversion of  $LE_{24h}$  in daily evapotranspiration ( $ET_{24h}$ ) ( $mm day^{-1}$ ) is given by Equation (7).

$$ET_{24h} = (FE_i \times Rn_{24h} \times 86,4) / 2450 \quad (7)$$

#### Data requirements of the SEBAL

The SEBAL requires three input data sets:

- Estimated data using satellite images: Vegetation indices (NDVI, SAVI and LAI), surface albedo, emissivity, and surface temperature.
- Data obtained by the weather station: air temperature, atmospheric pressure, and wind speed.
- Remotely sensed data, such as long wave radiation and shortwave radiation.

#### Data for testing

In this study we used data from a tower of the Federal University of Mato Grosso (UFMT), installed in an area located in the private protected area of the Social Service of Commerce (RPPN SESC Pantanal) located in coordinates  $16^{\circ}39'50''S$  and  $56^{\circ}47'50''W$ , whose altitude is 120 m, in Baão de Melgaço-MT, distant 160 km from Cuiabá - MT, Brazil (Figure 7). We used a Landsat-5 TM image of study area on 4 September 2008 (presenting clear sky conditions), provided by the National Institute for Space Research - INPE (<http://www.dgi.inpe.br/CDSR/>). The Bands 1, 2,

3, 4, 5 and 7 of the Landsat-5 TM has spatial resolution of  $30 \times 30$  m, and the band 6 (thermal) has spatial resolution of  $120 \times 120$  m, the Landsat-5 TM has temporal resolution of 16 in 16 days, and their passage to 9:30 pm local time. The image had a size of  $1,688 \times 1,279$  pixels. All pre-processes (stacking, georeferencing, and clipping) were all performed using the software ERDAS 9.2. The tests were run on the GPU using a NVIDIA 520m with 48 cores at 1.5 Ghz and NVIDIA GeForce GTX 560 with 384 cores at 1.5 Ghz.

#### Graphic processing unit

The architecture of the GPU's circulating around a vector scalable multi-threaded, streaming multiprocessor (SM) (NVIDIA, 2011), and each Streaming Processor (SP) manages memory allocation, synchronization, and communication between the SP's (Papakonstantinou et al., 2009), which facilitates access the parallelism of the GPU's. Performing instructions on the GPU pipeline uses, and each stage is performed by a specific hardware parallel. However, there is a possibility of the stages run on a single hardware using programmable unit (Owens et al., 2008).

#### OpenCL

OpenCL is an open standard for parallel programming of heterogeneous platforms, providing access to high-and-low-level processes in parallel devices. OpenCL works on both the CPUs and the GPUs from various manufacturers (NVIDIA, 2009; Augusto and Barbosa, 2013). OpenCL is a framework that also includes language, libraries, and API (Khronos, 2011). Each device on the OpenCL (Figure 1) has  $p$  compute units (CU), which in turn is composed of  $q$  processing elements (PE) (Augusto and Barbosa, 2013).

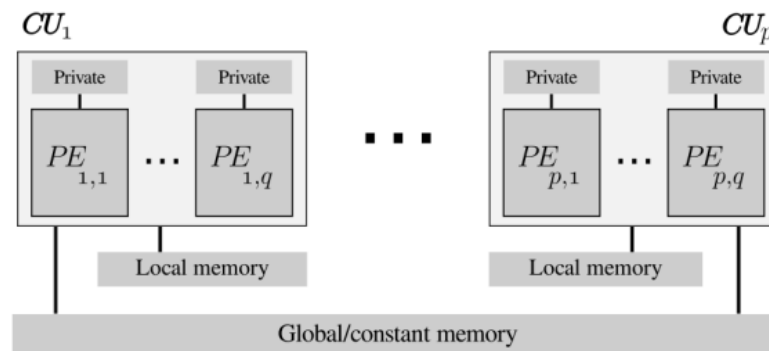


Figura 1. Concept of device architecture (Augusto and Barbosa, 2013).

The OpenCL performs parallelization on the GPU as shown in Figure 2, assuming that associated values for X and Y, then the multiplication of  $X*Y$  determines the number of work-items (processes or threads) that will exist on the card. By grouping a number p of work-items, determine if a workgroup, which is performed

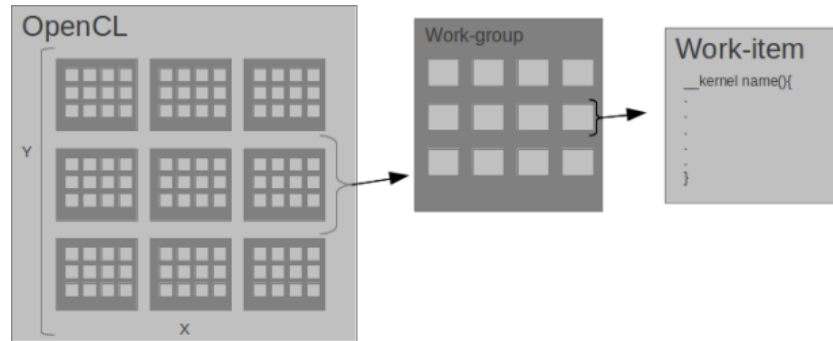


Figura 2. Work-items and workgroup of the OpenCL

$$WG_t = D_1 \times D_2 \times D_3 \quad (8)$$

$$WG_t \leq WG_T \quad (9)$$

where  $WG_T$  is the number of full workgroups supported by the video card and  $D_i$  ( $i = 1, 2, 3$ ) represents the number of workgroups that dimension. Another requirement is that the number of work-items in each dimension must be a multiple of the amount of workgroup of the size in question (Equation 10), or the division should result in an integer Z.

$$Z = \frac{WI_i}{WG_i} \quad (10)$$

where  $WI_i$  is the number of items in the work-dimension ( $i = 1, \dots, 3$ ). With this structure can divide tasks into smaller tasks that are executed simultaneously on different work-items, generating performance gain applied to solving the problem (NVIDIA, 2009).

### CUDA

CUDA is an architecture and programming model for parallel plates NVIDIA GPU to solve complex computational problems in a more efficient than on a CPU. It has also been developed to support multiple languages such as CUDA C, CUDA Fortran, and OpenCL (NVIDIA, 2011).

The CUDA-works gathering threads created by the program block (Figure 3) and each block is then executed independently of the other in each core (NVIDIA, 2011), as shown in figure 4. The blocks are also divided into grids.

using the SIMD architecture. This division can be made between one and three dimensions and required that the result of multiplying the amount of workgroups for each dimension (Equation 8), shall not exceed the number of work-groups maximum board (Equation 9).



Figura 3. Division into blocks of threads in CUDA.

There are some restrictions in CUDA and OpenCL. One is that the total of multiplying the number of threads per block in each dimension may not exceed the total number of threads per block supported by the plate (Equation 11).

$$T_T \geq T_1 \times T_2 \times T_3 \quad (11)$$

where  $T_T$  is the total number of threads per block supported by the plate and  $T_i$  ( $i = 1, \dots, 3$ ), is the number of threads per block of size  $i$ . Another constraint is that the number of grids by size cannot exceed the number of grids that dimension borne by the plate (Equation 12).

$$G_i \geq g_i \quad (12)$$

where  $G_i$  is the number of grid dimension ( $i = 1, \dots, 3$ ) supported by the plate, and  $g_i$  is the number of grids defined in dimension  $i$  in developed program (NVIDIA, 2011).

### Framework development for SEBAL

The framework was developed in order to provide three distinct resources for the developerto use for the calculation of SEBAL. One for the pure Java language, another using CUDA and another for OpenCL. All based on LandSat 5 satellite images. The framework was divided into 3 distinct parts:

1 - Pre-processing: it is the component where LandSat images are processed to generate the initial SEBAL information

2 - Calculation of pixels "hot" and "cold": this is the part where you can calculate the pixels that will serve as input for determining the energy balance.

3 - Processing: it is the component that receives all outputs from steps 1 and 2 and performs all SEBAL calculations with 3 Java, CUDA and OpenCL languages.

Figure 4 shows the class diagram that represents the framework. The LandSat class is where all the pre-processing of images takes place

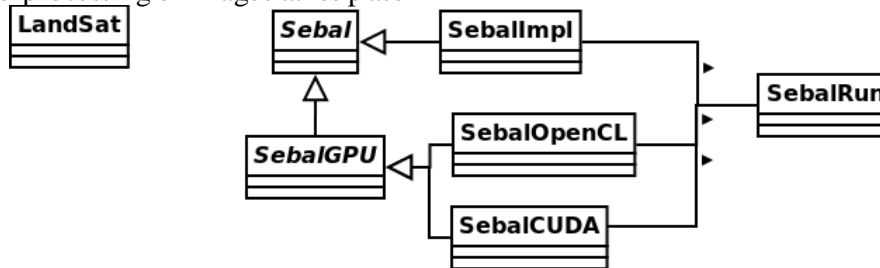


Figure 4. Class diagram of the framework.

The Algorithm 1 shows how to use the framework to calculate SEBAL using all available languages. After the implementation of the entire framework, we performed some tests to verify its performance in relation to ERDAS, and for that we use a performance measure described below.

Table 1 - Example algorithms for using the framework.

1	data's initialization
2	SebalRun sebal = new SebalRun();
3	List<DataStructure> datas;
4	System.out.println("CUDA");
5	datas = sebal.calculate(arqs, ExecutionEnum.CUDA);
6	System.out.println("OpenCL");
7	datas = sebal.calculate(arqs, ExecutionEnum.OPENCL);
8	System.out.println("Java");
9	datas = sebal.calculate(arqs, ExecutionEnum.NORMAL);
10	Utilities.exportTXT(System.getProperty("user.dir") + "/OutputSebal", datas);

### Performance measurement

We used the speedup in this study for measures performance (Equation 13) (Staroba, 2004). The speedup of a parallel system can be defined as the ratio between the time spent by a single processor to solve a given problem and the time spent by a

with the output of NDVI, SAVI, LAI, albedo, balance of radiation, emissivity, surface temperature and long wave radiation. The Sebal abstract class has a method for calculating the coefficients of the "hot" and "cold" pixels. And the SebalImpl, SebalCUDA and SebalOpenCL classes represent the SEBAL calculation using Java, CUDA and OpenCL respectively. Along with these classes, a class called SebalRun was developed that has the function of executing all the necessary procedures for the execution of SEBAL receiving only the input images and the type of execution that the developer wants, for example, Java, CUDA or OpenCL.

parallel System of n processors to solve the same problem (Hwang, 1993).

$$S(n) = \frac{T_1}{T_n} \quad (13)$$

where  $T_1$  is the execution time with a processor and  $T_n$  is the execution time with n processors.

### Statistical analysis

The data are normally distributed by the Kolmogorov-Smirnov tests. The evaluation of ET estimated by SEBAL via function ModelMaker software ERDAS 9.2 image processing algorithm implemented in the JAVA GPU was performed using some indicators: accuracy - Willmott index "d" (Equation 14); root mean square error "RMSE" (Equation 15) and the mean absolute error "EMA" (Equation 16). The accuracy is related to the remoteness of the estimated values from those observed (ERDAS software). Mathematically this approach is given by a designated index of agreement that can be widely applied to the comparison between models (Willmott et al., 1985). Their values range from zero, with no agreement, 1 with perfect agreement.

$$d = 1 - \left[ \frac{\sum (P_i - O_i)^2}{\sum (|P_i - O_i| + |O_i - P_i|)^2} \right] \quad (14)$$

where  $P_i$  is the estimated value, the observed value  $O_i$  and  $O$  is the average of the observed values.

The RMSE indicates how the model fails to estimate the variability of measurements around the mean and measures the change in the estimated values around the measured values (Willmott & Matsuura, 2005). The lower limit of RMSE is 0, which means that there is full compliance between the model estimates and measurements.

$$RMSE = \sqrt{\frac{\sum (P_i - O_i)^2}{n}} \quad (15)$$

The EMA indicates the distance (deviation) of the mean absolute values estimated from the values measured. Ideally, the values of the EMA and NDE were near zero (Willmott & Matsuura, 2005).

$$EMA = \sum \frac{|P_i - O_i|}{n} \quad (16)$$

## Results and discussion

We have implemented the SEBAL in three languages: Java, CUDA, OpenCL which allows comparison of the time response between them. However, we automated the calculation of the A and B coefficients of SEBAL which facilitates the calculating process of the SEBAL algorithm by Java, CUDA and OpenCL languages. This automation refers to the calculation of the temperature variation of the hot and cold pixel, thereby reducing the possibility of errors that previously its calculation was performed manually in a spreadsheet (Allen et al., 2002; Danelichen et

la., 2020). Another important automation was to obtain evapotranspiration in a single step, as it was obtained in several steps before automation (Santos et al, 2010; Lima et al., 2021).

The estimated time was only the execution time of each test, disregarding the time of reading and writing, not part of the scope of this work. Each test was run several times and the highest and lowest time were removed and made the average response time for analysis. The time response was reduced at least 400 times independently from the resource utilized (Figure 5). The best performance for processing SEBAL in ERDAS software was obtained with the CUDA with fastmath (fm) option and 10 stream multiprocessors (SM), featuring an increase of approximately 36,000 times. The fastmath option had more effect on board with 48 cores than stream multiprocessor configuration. Moreover, the stream multiprocessor option had more effect on the board with 384 cores than fastmath configuration. It is noteworthy that the fastmath option in compiling CUDA makes GPU uses optimized implementations for some functions, such as log and divide (NVIDIA, 2011). The time response was reduced from approximately 7 to 90 times using Java code (Figure 6). The best performance to processing SEBAL in relation to the implemented in Java was obtained by CUDA with fastmath (fm) option and 10 stream multiprocessors (SM). The fastmath option had more effect on the board with 48 cores than stream multiprocessor configuration. Moreover, the stream multiprocessor option had more effect on the board 384 cores than fastmath configuration.

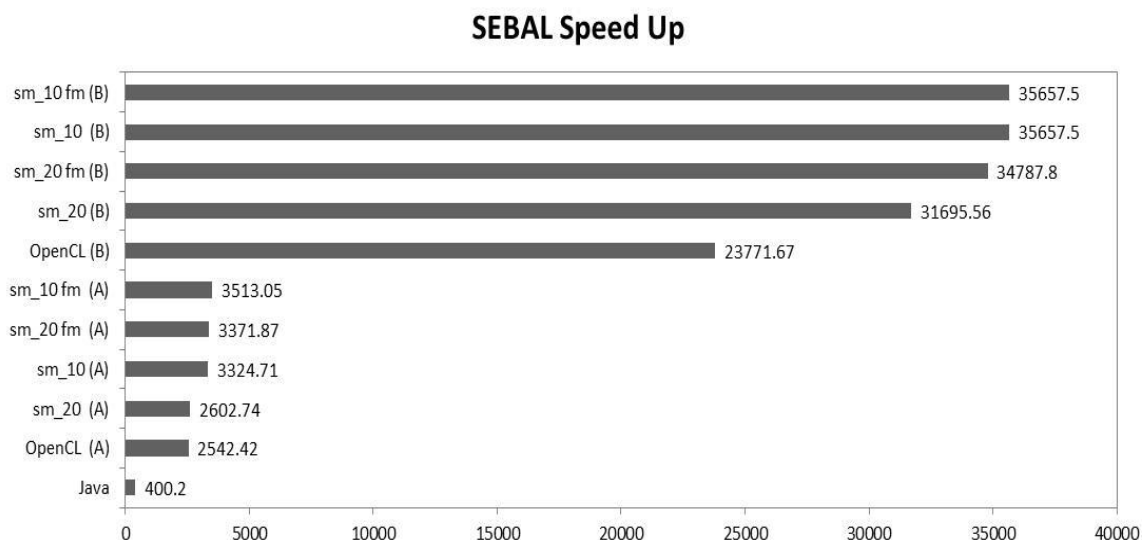


Figure 5. Speedup relative to ERDAS software.

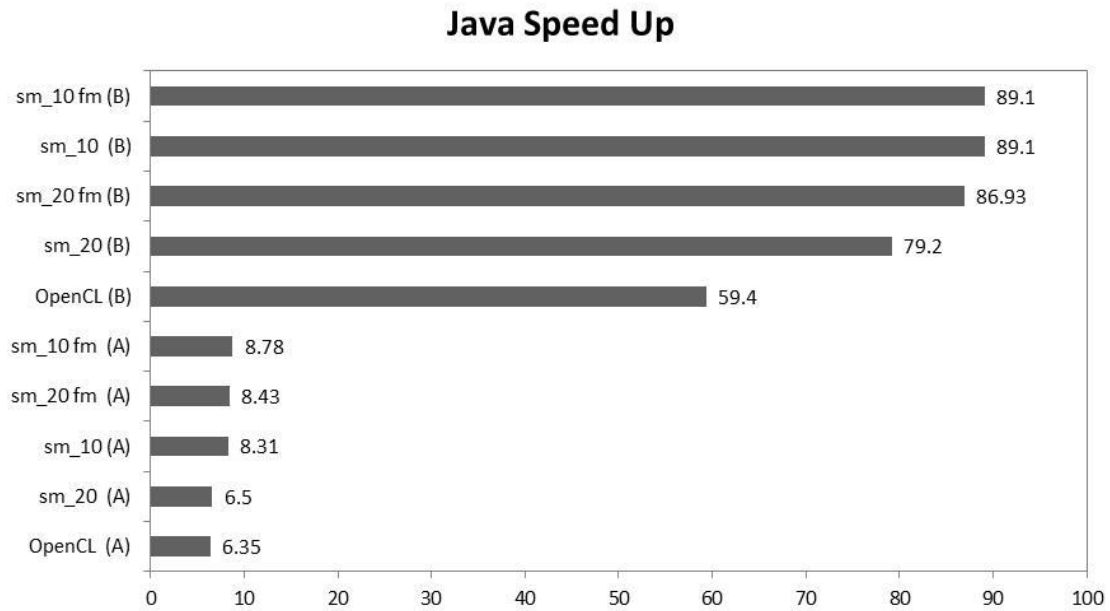


Figure 6. Speedup relative to Java code.

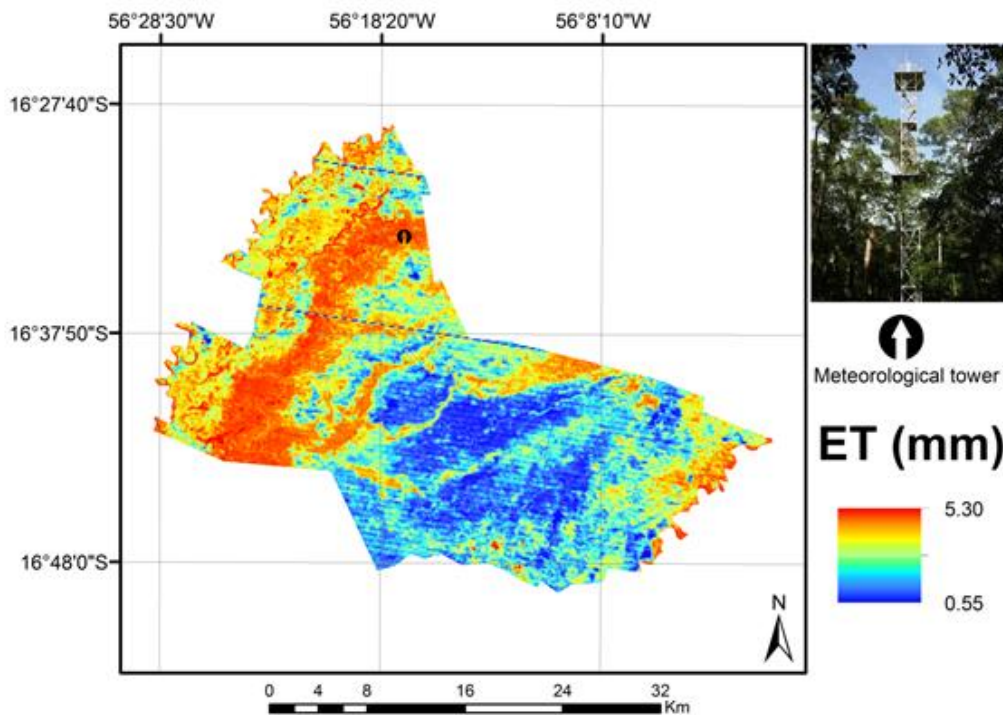


Figure 7. Evapotranspiration on September 04th 2008 in the study area located in the RPPN SESC Pantanal.

The daily estimated evapotranspiration (ET) by SEBAL using JAVA language, CUDA and OpenCL and the algorithm executed by ERDAS software had errors near to zero and correlation coefficient and Willmot concordance index were close to 1 (Figure 7). The values obtained in this study showed a correlation of 0.99, EMA of 0.136, RMSE of 0.065 and  $d$  equal to 0.99. These values were due to the number of dT interactions to stabilize aerodynamic resistance (Mendonça et al.,

2012; Danelichen et al., 2020). When computed for a spreadsheet the number of iterations required was 16, stabilizing an error of 2.2%. By the other hand, the running algorithm by JAVA, CUDA and OpenCL languages in which the interaction processes were automated, the number of interactions was 31 with an error less than 0.01%.

Biudes (2009) obtained daily ET of 4,863 mm for the same day chosen in this research using the Bowen ratio method to data from the

meteorological tower of RPPN SESC Pantanal. The daily ET computed in this study using the software ERDAS ( $ET = 4.3605 \text{ mm day}^{-1}$ ) was 10% lower than the ET obtained by Biudes (2009) and 9% lower than the value obtained by SEBAL using the JAVA language, CUDA and OpenCL ( $ET = 4.383 \text{ mm day}^{-1}$ ).

## Conclusions

Three new implementations of SEBAL algorithm using Java, CUDA and OpenCL were introduced. These implementations were tested using a consistent data set and the codes were timed. At best case, it was possible to achieve about 36,000 (90) times speed up (going from 8 hours of work to 2 minutes including the time needed to read the input data from hard drive and store it back). This speed up was achieved by using only one CUDA-enable card. To further speed up the program one can use multiple NVIDIA card and in other platforms using OpenCL. The daily ET values estimated by Java, CUDA and OpenCL languages were like ET estimated by SEBAL algorithm in ERDAS software.

## Software availability

- Software Name: JGPUSebal
- Developers: This paper authors.
- Hardware Requirements: PC equipped with CUDA-Enabled NVIDIA Graphic card.
- Software Requirements: GCC, OpenCL, JOCL, CUDA, Jcuda
- Cost: free
- Programming Language: Java, OpenCL, CUDA
- Download instruction: Requires subversion. To download anonymously (no username or password required) issue the following command: `svn checkout http://jgpusebal.googlecode.com/svn/trunk/jgpusebal-read-only`.

## References

Allen, R.; Bastiaanssen, W.; Waters, R.; Tasumi, M.; Trezza, R., 2002. Surface energy balance algorithms for land (SEBAL), Idaho implementation - Allen, R. G. Assessing integrity of weather data for use in reference evapotranspiration estimation. *Journal of Irrigation and Drainage Engineering*, 22. 97-

- 106, 1996. Advanced training and user's manual, version 1.0, 2002. 97p.
- Allen, R.; Irmak, A.; Trezza, R.; Hendrickx, J.M.H.; Bastiaanssen, W.; Kjaersgaard, J., 2011. Satellite-based ET estimation in agriculture using SEBAL and METRIC. *Hydrological processes* 25, 4011-4027.
- Angelini, L. P.; Silva, P. C. B. S.; Fausto, M. A.; Machado, N. G.; Biudes, M. S. 2017. Energy Balance in Terms of Land Use Changes in the Southern of Mato Grosso State. *Revista Brasileira de Meteorologia*, v. 32, p. 353-363.
- Augusto, D. A.; Barbosa, H. J. C., 2013. Accelerated parallel genetic programming tree evaluation with OpenCL. *Journal of Parallel and Distributed Computing*, 1, p. 86-100.
- Bastiaanssen, W. G. M.; Menenti, M.; Feddes, R. A.; Holtslag, A. A. M. A. 1998a. Remote Sensing Surface Energy Balance Algorithm for Land (SEBAL) 1. Formulation. *Journal of Hydrology*, v. 212-213, p. 198-212.
- Bastiaanssen, W. G. M., 2000. SEBAL - Based sensible and latent heat fluxes in the irrigated Gediz Basin, Turkey. *Journal of Hydrology*, v. 229, p. 87-100.
- Bégué, A.; Arvor, D.; Bellon, B.; Betbeder, J.; De Abelleira, D.; Pd Ferraz, R.; Lebourgeois, V.; Lelong, C.; Simões, M.; R Verón, S., 2018. Remote sensing and cropping practices: A review. *Remote Sensing*, 10(1):99.
- Biudes, M.S.; Campelo Júnior, J.H.; Nogueira, J.S.; Sanches, L., 2009. Estimativa do balanço de energia em cambarazal e pastagem no norte do Pantanal pelo método da razão de Bowen. *Revista Brasileira de Meteorologia*. 24, 135-143.
- Burek, P.; Satoh Y., Fischer G., Kahil, M.T., Scherzer, A., Tramberend, S., Nava, L. F., Wada Y., 2016. Water futures and Solution: Fast Track Initiative (Final Report). IIASA Working Paper (International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria).
- Cardim, G. P.; Silva, E. A. Da; Dias, M. A.; Bravo, I.; Gardel, A., 2018. Statistical Evaluation and Analysis of Road Extraction Methodologies Using a Unique Dataset from Remote Sensing. *Remote Sensing*, 10, 620.
- Danelichen, V. H. M.; Gomes, R. S. R.; Figueiredo, J. M.; Velasque, M. C. S.; Machado, N. G.; Nogueira, José De Souza; Biudes, M. S., 2020. Automation of choosing hot and cold pixels process in the estimate of the sensitive heat flow and evapotranspiration. *Revista Ibero-americana de Ciências Ambientais*, 11, 638-651.

- Glenn, E., Huete, A., Hirschboeck, P.N.K., Brown, P., 2007. Integrating remote sensing and ground methods to estimate evapotranspiration. *Critical Reviews in Plant Sciences* 26, 139-168.
- Food and Agriculture Organization of the United Nations (FAO), 2018. AQUASTAT [www.fao.org/nr/aquastat/](http://www.fao.org/nr/aquastat/).
- Ferreira Junior, P. F.; Souza, A. M.; Vitorino, M. I.; Souza, E. B.; Souza, P. J. O. P., 2013. Estimate of evapotranspiration in the eastern Amazon using SEBAL. *Rev. Cienc. Agrar.*, v. 56, n. 1, p. 33-39.
- Kalma, J., Mcvicar, T., McCabe, M., 2008. Estimating land surface evaporation: a review of methods using remotely sensed surface temperature data. *Surveys in Geophysics* 29, 421-469.
- Khronos OpenCL Working Group, 2011. The OpenCL specification, version 1.1. Disponível em <https://www.khronos.org/registry/OpenCL/specs/opencv-1.1.pdf>.
- Hwang, K. 1993. Advanced Computer Architecture: Parallelism, Scalability, Programmability. Computer Engineering Serie. McGraw-Hill.
- Leivas, J. F.; Andrade, R. G., Nogueira, F. S.; Bolfe, E. L., 2011. Estimativa da Evapotranspiração em áreas com diferentes características ambientais utilizando o algoritmo Sebal. XVII Congresso Brasileiro de Agrometeorologia – SESC Centro de Turismo de Guarapari, Guarapari - ES.
- Lima, C. E. S.; Costa, V. S. O.; Galvêncio, J. D.; Silva, R. M.; Santos, C. A. G., 2021. Assessment of automated evapotranspiration estimates obtained using the GP-SEBAL algorithm for dry forest vegetation (Caatinga) and agricultural areas in the Brazilian semiarid region. *Agricultural Water Management*, v. 250, 106863.
- Mendonça, J. C.; Souza, E. F.; Bouhid, A.; Silva, B. B.; Ferreira, F. J., 2012. Estimativa do fluxo de calor sensível utilizando o algoritmo SEBAL e imagens MODIS para a região Norte Fluminense, RJ. *Revista Brasileira de Meteorologia*, 27, 85 – 94.
- Menke, A. B.; Junior, O. A. C.; Gomes, R. A. T.; Martins, E. S.; Oliveira, S. N., 2009. Análise das mudanças do uso agrícola da terra a partir de dados de sensoriamento remoto multitemporal no Município de Luis Eduardo Magalhães (BA – Brasil). *Sociedade & Natureza*, 21 (3): 315-326.
- Mu. Q.; Zhao. M.; Running. S. W., 2011. Improvements to a MODIS global terrestrial evapotranspiration algorithm. *Remote Sensing of Environment. Numerical Terradynamic Simulation Group, Department of Ecosystem and Conservation Sciences, The University of Montana, Missoula, MT 59812, USA.*
- NVIDIA. Nvidia opencl jump start guide, 2009. Disponível em [http://developer.download.nvidia.com/OpenCL/NVIDIA\\_OpenCL\\_JumpStart\\_Guide.pdf](http://developer.download.nvidia.com/OpenCL/NVIDIA_OpenCL_JumpStart_Guide.pdf).
- NVIDIA. NVIDIA CUDA C Programming Guide. 2011. Disponível em [http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA\\_C\\_Programming\\_Guide.pdf](http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf).
- S. S. T. De Oliveira, V. J. Do S. Rodrigues, L. G. Ferreira, W. S. Martins., 2018. P-Twdtw: Parallel Processing of Time Series Remote Sensing Images Using Manycore Architectures. 2018 Symposium on High Performance Computing Systems (WSCAD), São Paulo, Brazil, p. 252-258.
- Owens, J. D.; Houston, M.; Luebke, D.; Green, S.; Stone, J. E.; Phillips, J. C., 2008. GPU Computing. *Proceedings of the IEEE, IEEE*, v. 96, n. 5, p. 879–899.
- Papakonstantinou, A.; Gururaj, K.; Stratton, J. A.; Chen, D.; Cong, J.; Hwu, W. M. W., 2009. Fcuda: Enabling efficient compilation of cuda kernels onto fpgas. In: *Application Specific Processors. IEEE 7th Symposium on Application Specific Processors*, p.35 –42.
- Ruhoff, A. L., 2011. Sensoriamento remoto aplicado à estimativa da evapotranspiração em biomas tropicais. Tese de doutorado. Universidade federal do Rio Grande do Sul, Instituto de Pesquisas Hidráulicas, Programa de Pós-Graduação em Recursos Hídricos e Saneamento Ambiental, Porto Alegre, BR – RS.
- Santos, C. A. C.; Bezerra, B. G.; Silva, B. B.; Rao, T. V. R., 2010. Assessment of daily actual evapotranspiration with SEBAL and S-SEBI algorithms in cotton crop. *Revista Brasileira de Meteorologia*, v. 25, p. 383-392.
- Silva, B. B. Da, Lopes, G. M.; Azevedo, P. V., 2005. Balanço De Radiação Em Áreas Irrigadas Utilizando Imagens Landsat 5-TM. *Revista Brasileira de Agrometeorologia*, 2, 243-252.
- Silva, B. B.; Bezerra, M. V. C. 2006. Determinação dos fluxos de calor sensível e latente na superfície utilizando imagens TM - Landsat 5. *Revista Brasileira de Agrometeorologia*, 14 (2), pp. 174-186.
- Soares, J.; Danelichen, V. H. M.; Pereira, O. A.; Martins, A. L. 2020. Estudo da dinâmica espaço-temporal do NDVI no Município de

- Sorriso-MT. Revista Brasileira De Geografia Física, v. 13, p. 834.
- Staroba, I. J., 2004. Parallel Performance Modeling, Prediction and Tuning. Bmo University of Technology, Faculty of Information Technology, 1-79.
- Tian, J.; Su, H.; Sun, X.; Chen, S.; He, H.; Zhao, L., 2013. Impact of the Spatial Domain Size on the Performance of the Ts-VI Triangle Method in Terrestrial Evapotranspiration Estimation. Remote Sens.22. 5(4), 1998-2013.
- Vörösmarty, C. J., McIntyre, P. B., Gessner, M. O., Dudgeon, D., Prusevich, A., Green, P, 2010. Global threats to human water security and river biodiversity. Nature, 467, 555-561.
- Willmott, C.J.; Ckleson, S.G.; Davis, R.E.; Feddema, J.J.; Klink, K.M.; Legates, D.R.; O'donnell, J.; Rowe, C.M., 1985. Statistics for the evaluation and comparison of models. Journal of Geophysical Research, 90, n.C5, p.8995-9005.
- Willmott, C.J.; Matssura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research, 30, 79-92.