Method for Processing High-Resolution Satellite Images, Based on Multi-GPU Programming

Christian Ovalle[1][a], Sandra Meza[2][b], Wilver Auccahuasi[3][c], Oscar Linares[4][d], Kitty Urbano[5][e], Gabriel Aiquipa[6][f], Yoni Nicolas-Rojas[7][g], Aly Auccahuasi[8][h], Tamara Pando-Ezcurra[9][i] and Karin Rojas[10][j]

[1] *Universidad Tecnológica del Perú, Lima, Peru*
[2] *Universidad ESAN, Lima, Peru*
[3] *Universidad Privada del Norte, Lima, Peru*
[4] *Universidad Continental, Huancayo, Peru*
[5] *Universidad Científica del Sur, Lima, Peru*
[6] *Universidad Tecnológica de los Andes, Apurímac, Peru*
[7] *Escuela superior la Pontificia, Ayacucho, Peru*
[8] *Universidad de Ingeniería y Tecnología, Lima, Peru*
[9] *Universidad Privada Peruano Alemana, Lima, Perú*
[10] *Universidad César Vallejo, Lima, Peru*

Keywords: Satellite Image, Processing Time, Schedule, GPU, Multi GPU.

Abstract: Satellite images are being widely used today, due to the amount of information it represents, with greater emphasis on availability, we find many services where we can download images in their multiple formats and combinations, that is why we can mention that the Satellite images can be classified, by their resolution, use, amount of spectral bands, among others, but they have something in common, which is the size they occupy on the disk, we can find images that can weigh between 500 MB when it is made up of a single band and reduced coverage on the ground, up to 10GB images can weigh, containing more than 4 bands and with a greater coverage of the ground. To be able to work with these images, it is necessary to have workstations with great computational capacity, to be able to support this large volume of information, which is why in many cases clusters of workstations are used. In this work, a very practical mechanism will be presented for processing and working with these images, using GPU, in various configurations, such as a normal case that has a single GPU and in cases where it is counted With two GPUs, to test the methodology we work with different satellite images of different weights and they applied basic processes, processing times were measured in the CPU, in one GPU and in two GPUs, which were presented as results, we can indicate that the methodology can be scalable towards the processing of different images.

[a] https://orcid.org/0000-0002-5559-5684
[b] https://orcid.org/0000-0002-4650-1340
[c] https://orcid.org/0000-0001-8820-4013
[d] https://orcid.org/0000-0002-7952-9518
[e] https://orcid.org/0000-0003-2009-000X
[f] https://orcid.org/0000-0002-3755-7393
[g] https://orcid.org/0000-0001-6493-6084
[h] https://orcid.org/0000-0001-5069-0415
[i] https://orcid.org/0000-0003-0301-3440
[j] https://orcid.org/0000-0002-6867-0778

Ovalle, C., Meza, S., Auccahuasi, W., Linares, O., Urbano, K., Aiquipa, G., Nicolas-Rojas, Y., Auccahuasi, A., Pando-Ezcurra, T. and Rojas, K.
Method for Processing High-Resolution Satellite Images, Based on Multi-GPU Programming.
DOI: 10.5220/0011962900003612
In *Proceedings of the 3rd International Symposium on Automation, Information and Computing (ISAIC 2022)*, pages 618-626
ISBN: 978-989-758-622-4; ISSN: 2975-9463

1 INTRODUCTION

Today there are many applications to be able to work with satellite images, between commercial and free use, as well as many alternatives using different computational tools that carry out tests to optimize the use of the hardware that is intended to be able to work with the images having in consideration of the processing time and the response that the hardware has to the large size of the satellite image. Making a review of the literature, we found investigations where reference is made to the use of satellite images in its different types, such as optical and radar, using different hardware and software configurations.

We found works where reference is made to the analysis of land cover by analyzing chromatic characteristics, using synthetic aperture images, which are better known as Radar images (Auccahuasi, 2020). In the analysis of optical images we also find works on the analysis of land cover based on chromatic characteristics (Auccahuasi, 2018). In the work of optical images, we find works where reference is made to the use of GPGPU programming where the intention is to apply these techniques to improve performance in the process of satellite images (Auccahuasi, 2019).

In the study of large areas of land obtaining a level of detail, through satellite images which provide observation of the Earth which are composed of a series of spectral bands which will depend on the type of satellite mission which has been conceived of the optical instrument considered as a payload which have been represented as multidimensional and large arrays, which requires a good team to carry out image processing and requires specialized software allowing a visual interpretation of satellite images, with which it is possible to work with These images with different configurations are normally used with separate band configurations where each band is worked separately, however these images have a particularity such as high resolution with panoramic bands with the maximum resolution which can vary from metric to sub-metric where the red, green, blue bands and short wave and infrared significant wave found in a range with low spatial resolution (Auccahuasi, 2020).

We find satellite images with a panoramic approach where high resolution panoramic images and low resolution multispectral images are merged, which is considered an important image processing chain, the more the resolution increases and the number of images about the panoramic approach increases. processing time, so multithreaded programming and general-mode GPU programming (GPGPU) implementation which improves performance about applications over image processing to perform operations by analyzing individual pixels (Kutbay, 2020).

We find works where the use of convolutional neural network (CNN) models is used, for which it must be expanded with respect to graphic processing units (GPU) due to computational characteristics, to perform the classification and detection of objects, in the critical environments the reliability of the models has software errors, so our objective is to analyze the reliability of the VGG model, through a CNN architecture, which has 2 metrics such as the so-called Kernel Vulnerability Factor (KVF) used to identify vulnerable kernels and the Layer Vulnerability Factor (LVF) used to determine the propagation of track failures across layers, so the new model is run by an NVIDIA GPU with an assessment tool like the fault injector called SASSIFI, from which it was obtained as a result that Im2col has vulnerability between the cores and on the hardening of the core which realizes a reduction of the rate of corruption of silent data by 85.67% with a time penalty of 35.63%, with an average error which causes a misclassification of 19.7% in the high and unacceptable injection mode this failure mode will affect KVF and LVF because in the RF mode it is performed at the architecture level unlike the IOV and IOA modes at different levels then they have been compared between VGG, AlexNet and ResNet which indicates that LVF and KVF are considered within the CNN network architecture, due to the network layers, especially the convolutional layers, responsible for implementing the most vulnerable and invoked Im2col kernel in these CNN networks (Wei, 2020) .

In the use of satellite images for the analysis of disasters to ecology, such as the analysis of the oil spill, which have adverse effects on the environment and the economy, for which the best resources must be available for the response to cleaning and control incidents for which we use a multi-temporal remote detection Technologies (RS) for the detection of monitoring of oil spills in the ocean in the updated RS data have been used for the detection of hydrocarbons which require a lot time and over with high price so the proposal of the work is detection of oil spills starting from the voluminous multi-temporal images LANDSAT-7 with the use of high-performance technologies through a graphics processing unit (GPU) and the interface Message Pass-Through (MPI) to speed up the detection process and provide fast response, the GPU was based on Kepler's compute architecture (Tesla K40) using the Unified Computing Device (CUDA) containing parallel programming with detection algorithms, with this detection technique have been adapted to GPU-based processing include the band ratio and morphological attribute profile (MAP) based on 6 structural and shape description characteristics, namely, gray mean, standard deviation, elongation, shape complexity, robustness and orientation, the experiments showed significant gains about computational speed implemented with in a GPU and MP, making the comparison of a GPU versus CPU, an approach with an acceleration of around 10 × for MAP and 14 × s for band ratio approaches, which includes the cost of data transfer, can be seen with the MPI implementation that uses 64 cores outperforms the GPU and executes the task which takes a long time to compute in 18 min, unlike the GPU, it consumes about an hour (Bhangale, 2020).

We find works where the growth of space debris is analyzed for which a distributed monitoring is required in order to avoid possible space collisions, mediated a space surveillance that is based on a terrestrial telescope which allows to track the traffic about Space Objects Residents (RSO) in Earth orbit, for which various applications are used such as prediction of orbits and the evaluation of conjunctions, for this reason the research proposes an optimized and performance-oriented pipeline for the extraction of sources for the automatic detection of spatial debris in optical data which is a method for detection that is based on morphological operations and the Hough transform for lines, with near real time analysis which is obtained using general purpose computing in graphics processing units (GPGPU), with a high degree of processing parallelism provided by GPGPU pe Allowing to divide the data analysis into thousands of threads to process large data sets with limited computational time, this proposal was approved on a large and heterogeneous image data set with satellite images with various orbital ranges with multiple observation modes , which have been taken in the observation campaign carried out from the EQUO observatory that was installed in the Broglio Space Center (BSC) in Kenya, which is part of the ASI-Sapienza Agreement (Diprima, 2020).

One of the new techniques is related to being able to analyze about the performance demands of on-board computing taking into account both the performance from the control parts being a payload requiring advanced electronic systems capable of high computational power under the limitations of the harsh space environment, which is why a study about the application of integrated GPUs in space has started, which showed better performance due to the proliferation in consumer markets based on competitive European technology for which it is an analysis of the application domains in order to identify which software domains can benefit from their use, the domain of the integrated GPU was also analyzed to evaluate whether or not it meets the computing power necessary for adoption in space (Kosmidis, 2020).

In the use of radar technologies, they are used in remote detection by satellite in order to carry out monitoring of geographical hazards and risk management at a synoptic scale, there is an advanced algorithm of multi-temporal SAR interferometry with which it can be detected ground deformations and structural instability with millimeter precision for which capacity is needed in hardware resources, for which a system is implemented that contains the computation and GPU programming considered as an efficient implementation with time of image algorithms, with the which improves the emergency management service that is based on earth observation technology, the preliminary evaluation of the GPU processing potentials is made, then it was compared with the CPU and GPU implementations of the algorithm cores time-consuming InSARs (Guerriero, 2015).

After a detailed analysis of the different works related to the use of the GPU in image processing, we can indicate that our proposal helps the scientific and academic community in the analysis of satellite images, the proposal is to present a method to work with various hardware configurations, such as CPU, GPU and 2 GPUs, taking as reference a large image, which can be processed in each of the cases, the results will determine under what conditions it is useful to work with each of them.

2 MATERIALS AND METHODS

The materials and methods are based on Figure 1, which indicates the steps to follow to explain in detail the proposed methodology, where each of the procedures will be solved so that it can be applied in other problematic situations.

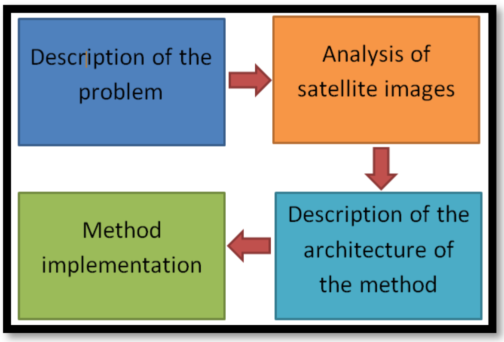

Figure 1: Proposal block diagram.

2.1 Description of the Problem

The description of the problem consists of being able to clearly explain the problem that one has when working with satellite images, one of the main factors is the format, it is the first obstacle that one has, depending on the level of processing these can change, the common format to work and that is read by most of the programs is the GEOTIFF. Another problem is the area of interest, in many cases the area of interest to be studied and analyzed is relatively smaller than the original image, so in order to extract the area of interest, it is necessary to load the original image and this is already a task that the station where you work must fulfill, another common problem is the amount of bands present, so most of the images have 4 bands. Making a comparison with an image in the color settings, this image has 3 color bands (R, G, B), so any visualization program can display it, but if the image has more than 4 bands, these will not can be read and much less visualized, it is where specialized applications are used to work with satellite images, which performs operations to separate the images into separate bands and thus be able to put them together in groups of 3 images, where the images can be viewed In the different combinations, in our case we will take this problem where we have the 3 bands in the image, in the following procedure we explain the analysis of these satellite images.

2.2 Analysis of Satellite Images

The satellite images have different configurations, size, resolution, weight in MB, number of bands among others, each of them depends on the mission that produces it, from this group they can also be presented in the different products that can be presented, to For the purposes of demonstrating the methodology, the use of the characteristic of "number of bands" and weight in MB of the images is used. For this reason, in Figure 2, a list of 6 images is presented grouped by size in MB, ordered by the smallest the higher, from where we can indicate that the image of a single band and of less weight is the first image in the list, with a weight in GB of 0.3, this image takes a time to be loaded in memory of 1.47 seconds and the image The largest image is the last image on the list with a size of 9.4 GB where the loading time of the image in memory is 29.54 seconds. Each one of the images presented in table 1, has its corresponding loading time in memory. These 6 images will be the ones we will use to check the methodology.

Let's consider that the loading time is due to the computational capacity with which we are working, an I7 CPU with 32 GB of memory and a solid state hard disk.

Table 1: Group of images that will be subjected to the methodology.

Satellite image type	Image size in GB	Time to load into memory in seconds
merged image	0.310	1.47
merged image	0.850	2.75
image with separate band	1.17	3.45
image with separate band	2.1	10.16
panchromatic image	2.5	12.48
panchromatic image	9.4	29.54

2.3 Description of the Architecture of the Method

The methodology consists of the use of hardware and software components, to be able to execute the different algorithms developed, to evaluate the methodology and demonstrate its applicability, as a base programming system, the use of the MATLAB computational tool is used, because it presents us with an environment of adequate programming as well as it has libraries for the access and direct work with the GPUs of the NVIDIA brand. As a hardware resource, a workstation composed of an i7 CPU is used, with 32 Gb of internal memory and a solid state hard disk, and as main hardware two graphics processors (GPU) model GTX 1050ti from NVIDIA that has 768 cuda cores. and 4GB of dedicated memory, in figure 2, the description of the architecture that we have to test the methodology is presented, one of the considerations that must be taken into account is that both GPUs have the same characteristics, this characteristic is important for the load balancing.

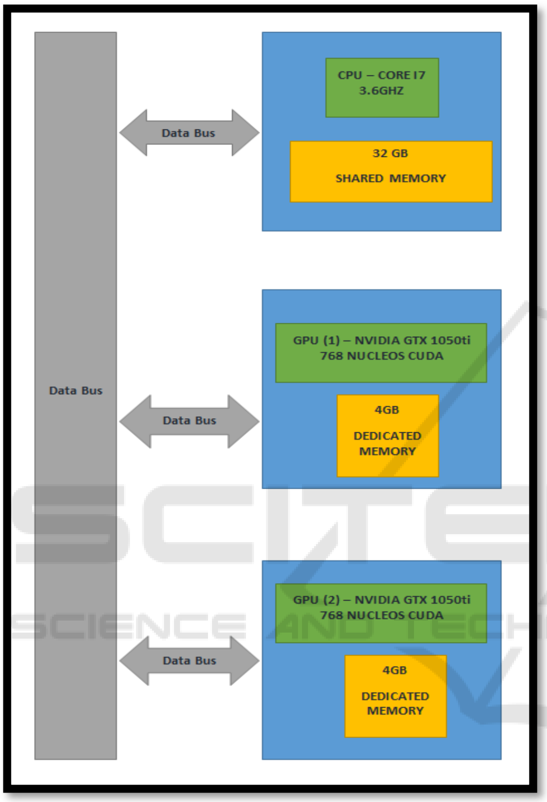

Figure 2: Configuration of the architecture available to demonstrate the methodology.

2.4 Method Implementation

The implementation method consists of analyzing the satellite images in such a way that its dimensions can be obtained in order to test the methodology, the mechanism consists of analyzing three situations, the first is to carry out the processing in the CPU, the second is to carry out the process in a GPU and the third is to perform the process on two GPUs, the processing is performed by performing the following operations on the images, represented in the following Pseudo code:

- We load the image
- We create a super image, joining all the images one after another
- We make the adjustment in the contrast of the image
- We perform the equalization of the histogram
- We apply a Gaussian filter
- We calculate the complement of the image

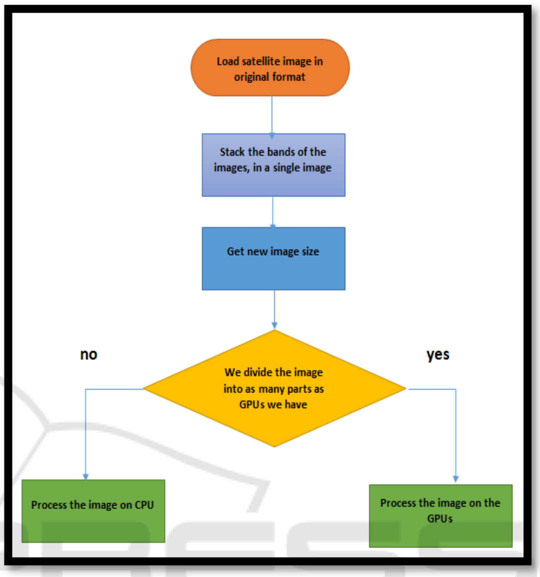

Figure 3: Flow chart for the selection process of the architecture to choose.

In figure 3, the flow diagram for the selection process of the architecture to choose is presented, the choice of choosing whether to use the CPU or the GPU consists of analyzing the size of the images, at this time to be able to analyze and demonstrate the methodology, procedures are performed manually.

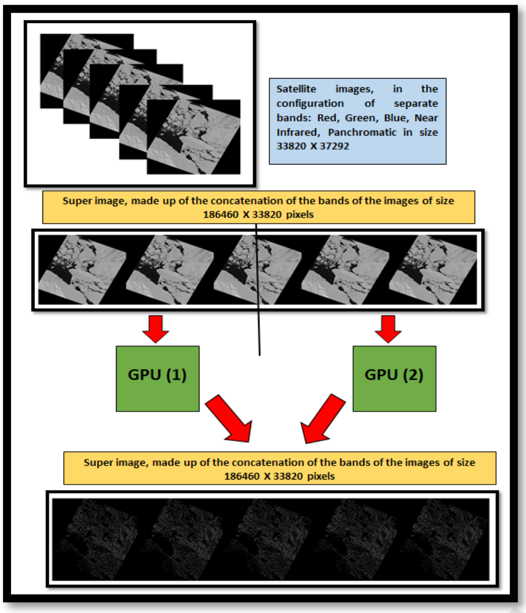

Figure 4: Working model diagram using two GPUs.

In figure 4, the working model is presented when using two GPUs, reading the original image, obtaining its size and dimensions; For the super image creation process, all the bands present in the images are concatenated, in a classic way as satellite images are presented, an image can contain 4 spectral bands that present a specific sensor, for example an image can contain the following bands (Red, green, blue, near infrared), depending on the level of image processing and the mission that performed the acquisition, these images may have more than one band. When you have the super image, the next process is to divide the super image into equal parts, in our case as we have two GPUs, the super image is divided into two images of equal sizes.

The processing is carried out in parallel on both images, after the result is obtained, the images are joined again, with which we have the image in the original format, for the processing that is considered necessary.

```
function final_image = stack_image(image)
[x y z ] = size (image);
first_image = image(:,:,1);
second_image = image(:,:,2);
third_image = image(:,:,3);
fourt_image = image(:,:,4);
final_image= [first_image second_image third_image fourt_image];
end
```

Figure 5: Super image creation function.

Figure 5 presents the super image creation function, the function receives the original image as a parameter, where first we obtain the image dimensions, then we decompose into separate images for each band and finally we concatenate all the images and the output of the function returns the super image.

```
function [image_1 image_2]= split_image(image)
[x y z ] = size (image);
limit = y/2;
image_1=image(:, 1:limit);
image_2=image(:, limit + 1:y);

end
```

Figure 6: Function for the division of the super image.

Figure 6 represents the division of the super image, the function receives the super image as an input parameter, then divides into two images of similar size, the function returns two images.

```
tic;
[image,map]=imread('uno.TIF');
final_image = stack_image (image);
imshow(final_image,map);
final_image_cpu_8 = im2uint8 (final_image);
final_image_cpu_8 =imadjust(final_image_cpu_8 );
final_image_cpu_8 = histeq(final_image_cpu_8 );
final_image_cpu_8 = imgaussfilt(final_image_cpu_8 );
final_image_cpu_8 = imcomplement(final_image_cpu_8 );
figure;imshow(final_image_cpu_8);
toc;
```

Figure 7: Code for CPU execution.

Figure 7 presents the code when operations are performed on the image, its main characteristic is that the entire process is carried out in the CPU, in order to measure performance, a measurement of the execution time is carried out from the loading of the image. in memory until the result is displayed on the screen.

```
tic;
[image,map]=imread('uno.TIF');
final_image = stack_image (image);
gpuDevice(1);
final_image_gpu = gpuArray(final_image);
imshow(final_image_gpu, map);
final_image_gpu_8 = im2uint8 (final_image_gpu);
final_image_gpu_8 =imadjust(final_image_gpu_8 );
final_image_gpu_8 = histeq(final_image_gpu_8 );
final_image_gpu_8 = imgaussfilt(final_image_gpu_8 );
final_image_gpu_8 = imcomplement(final_image_gpu_8 );
figure;imshow(final_image_gpu_8);
toc;
```

Figure 8: Code for execution on a GPU.

In figure 8, the code for the execution of the processing of the images in the GPU is presented, with the similar processes as the previous case, with the difference that it is executed in the GPU that has as index (1), similar to the process above, performance is measured by the time it takes between reading the image and displaying the result.

```
tic;
[image,map]=imread('uno.TIF');
final_image = stack_image (image);
[final_image_1 final_image_2] = split_image (final_image);
 gpuDevice(1);
 gpuDevice(2);
spmd (2)
if labindex == 1

 final_image_gpu1 = gpuArray(final_image_1);
figure;imshow(final_image_gpu1, map);
final_image_gpu1 = im2uint8 (final_image_gpu1);
final_image_gpu1 =imadjust(final_image_gpu1 );
final_image_gpu1 = histeq(final_image_gpu1 );
final_image_gpu1 = imgaussfilt(final_image_gpu1 );
final_image_gpu1 = imcomplement(final_image_gpu1);
% figure;imshow(final_image_gpu1);

elseif labindex == 2
 gpuDevice(2);
 final_image_gpu2 = gpuArray(final_image_2);
 figure;imshow(final_image_gpu2, map);
final_image_gpu2 = im2uint8 (final_image_gpu2);
final_image_gpu2 =imadjust(final_image_gpu2 );
final_image_gpu2 = histeq(final_image_gpu2 );
final_image_gpu2 = imgaussfilt(final_image_gpu2 );
final_image_gpu2 = imcomplement(final_image_gpu2);
% figure;imshow(final_image_gpu2);

end

end

image1 = gather(final_image_gpu1);
image2 = gather(final_image_gpu2);
final_image_gpu= [image1 image2];

figure;imshow(image1,map);
toc;
```

Figure 9: Code for execution on two GPUs.

In figure 9, the result of the execution in two GPUs is presented, where the processes are similar, in this code the parallelization function is called where 2 tasks are executed simultaneously, therefore in each of the tasks a part of the image, performance is measured by the time it takes between loading the image and displaying the results.

3 RESULTS

The results that are presented are grouped in two ways, the first shows the 6 images that have been worked in the methodologist's demonstration, and the second is represented in the times obtained when applying the three forms of processing. Here are the images.

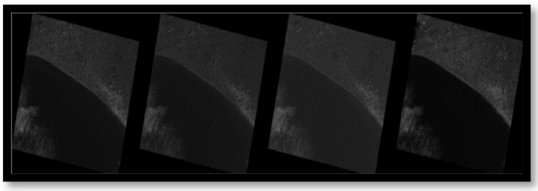

Figure 10: First image, made up of 4 colored bands.

In Figure 10, we present the images in their visualization of the 4 color bands, arranged one after the other, so that we can work simultaneously with the 4 color bands.

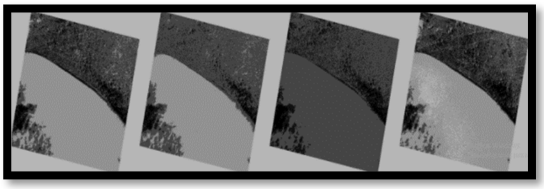

Figure 11: First image, after being processed.

In Figure 11, we present the images of the 4 color bands, processed one after the other, showing the simultaneous processing. Below we present the results, represented by the processing times obtained after executing the three forms of processing, the first processing was performed on the CPU, the second processing was performed on a GPU and the third was performed using 2 GPUs, we must Note that in the three cases, the result of the image processing is the same in all 3 cases, the only difference is the time it takes in each one.

Table 2: Processing times, performed on the CPU.

Satellite image type	image size in GB	Processing time (seg)
merged image	0.310	10.42
merged image	0.850	13.33
image with separate band	1.17	18.50
image with separate band	2.1	33.22
panchromatic image	2.5	39.81
panchromatic image	9.4	305.76

In table 2, the results are presented after processing the images using the first way, which consists of processing the images entirely in the CPU, in the calculated time it is considered from the loading of the image in memory, to the visualization of the result. In the end, the images are sorted in ascending order so that the larger the image, the longer the processing time will result.

Table 3: Processing times, performed on the GPU.

Satellite image type	image size in GB	Processing time (seg)
merged image	0.310	6.00
merged image	0.850	6.84
image with separate band	1.17	10.92
image with separate band	2.1	15.29
panchromatic image	2.5	20.70
panchromatic image	9.4	200.65

In table 3, the processing performed on a GPU is presented, with the group of images, in the results that are observed that the first images have a close value in the processing time, this characteristic is due to the fact that it is loaded first the image in system memory, then the process of changing the location from system memory to GPU memory is performed, in larger images the difference in processing time with processing times is more noticeable, due to As the image is larger, the image transfer time is compensated by the processing on the GPU.

Table 4: Processing times, performed on two GPUs.

Satellite image type	image size in GB	Processing time (seg)
merged image	0.310	6.52
merged image	0.850	4.75
image with separate band	1.17	7.45
image with separate band	2.1	12.16
panchromatic image	2.5	10.45
panchromatic image	9.4	105.87

In table 4, the processing times of the images are presented, considering that they have been processed in 2 GPUS, a characteristic that in the images of smaller size, the difference is short with respect to the processing time when it is done in a single GPU, because the transfer time between passing the image from the system memory to the memories of the GPUs, but this time is compensated with the processing time, this processing time is considered negligible when working with images of larger size, as is the case with the last images.

4 CONCLUSIONS

The conclusions reached at the end of the demonstration of the proposed methodology, is organized from three points of reference, the first from the use of software necessary for the processing of these images, the second is related taking as a reference the use of hardware and the optimization of its use, and the third criterion is related to the experience acquired in handling the images regarding the size and weight of the images, and we finalize with the final recommendations on the use and processing of satellite images in general, independent if they are optical, radar or other images.

Regarding the management of the software, there are different types of software that work with satellite images on the market, your choice will depend mainly on the type of final product that is required to obtain, in this sense they have a practical orientation, one of the characteristics to To be able to work with these, is that they require a workstation with good computational capacity, there are other ways of working with images, through the use of classic programs and compilers, that you can work with when you have limited computational capacity, that is When working under these conditions, the proposed methodology can be of great help, where we can control the flow of data at the time of processing.

With regard to hardware management, the methodology is very helpful, because it presents the different processing options depending on what is available, always with the same objective, which is the optimization of resources, an important consideration that must be taken, is the programming language that is developed, for this it is necessary to know if the library exists to be able to work with the GPUs, in our case MATLAB was chosen, because it has the libraries that are compatible with the NVIDIA brand GPUs. With regard to the size of the image, it is important to indicate that the results presented show that it is not always the best option to process in the GPUs, if it is what is required to have a lower processing time, this is due to a characteristic of the mode of work, any image that you want to work first is loaded into the system memory, that is why if the processing is carried out in the CPU, the time may be less, because there is no transfer time of the image between the memory of the system and the memory of the GPU, when the image is small, it may be the case that the processing time may be longer in the CPU, because of the transfer time. This characteristic changes considerably when the image to be processed is very large, because the processing time compensates for the transfer time, in some cases being considered negligible, as can be seen in the last image that has a weight of 9.4 GB.

Finally, we conclude that in the use of satellite images, it is important to know the characteristics of the images to be processed, such as image size, this data is important to be able to choose where it can be processed, as is the case of the CPU, a GPU or two GPUs, if the image is less than 1GB in weight can be processed on the CPU or a GPU, if the size is larger it is recommended to choose to process it in the configuration of two GPUs.

We can also indicate that we must know the type of processing, format and resolution, while the image is in its original format which is the GEOTIFF at 16 bits, it can only be worked by specialized software, to be able to work them it is necessary to convert them to JPG format of 8 bits, with which they can be observed with any standard visualization. When working with satellite images we must take into account the following: type of image, type of resolution, type of format, what hardware I have and finally the choice of software.

REFERENCES

Auccahuasi, W., Sernaque, F., Flores, E., Garzon, A., Barrutia, A., & Oré, E. (2020). Analysis of the chromatic characteristics, on land cover types using synthetic aperture images. Procedia Computer Science, 167, 2524-2533.

Auccahuasi, W., Bernardo, M., Núñez, E. O., Sernaque, F., Castro, P., & Raymundo, L. (2018, December). Analysis of chromatic characteristics, in satellite images for the classification of vegetation covers and deforested areas. In Proceedings of the 2018 the 2nd International Conference on Video and Image Processing (pp. 134-139).

Aiquipa, W. A., del Carpio, J., Garcia, J., Benites, R., Grados, J., & Flores, E. (2019, October). Analysis of High Resolution Panchromatic Satellite Images, Based on GPGPU Programming. In Proceedings of the 2019 2nd International Conference on Sensors, Signal and Image Processing (pp. 45-48).

Auccahuasi, W., Castro, P., Flores, E., Sernaque, F., Garzon, A., & Oré, E. (2020). Processing of fused optical satellite images through parallel processing techniques in multi GPU. Procedia Computer Science, 167, 2545-2553.

İ. S. Açıkgöz, M. Teke, U. Kutbay and F. Hardalaç, "Performance evaluation of pansharpening methods on GPU for RASAT images," 2015 7th International Conference on Recent Advances in Space Technologies (RAST), 2015, pp. 283-288, doi: 10.1109/RAST.2015.7208356.

Wei, J., Ibrahim, Y., Qian, S., Wang, H., Liu, G., Yu, Q., … Shi, J. (2020). Analyzing the impact of soft errors in VGG networks implemented on GPUs. Microelectronics Reliability, 110. https://doi.org/10.1016/j.microrel.2020.113648

Bhangale, U., Durbha, S. S., King, R. L., Younan, N. H., & Vatsavai, R. (2017). High performance GPU computing based approaches for oil spill detection from multi-temporal remote sensing data. Remote Sensing of Environment, 202, 28–44. https://doi.org/10.1016/j.rse.2017.03.024

Diprima, F., Santoni, F., Piergentili, F., Fortunato, V., Abbattista, C., & Amoruso, L. (2018). Efficient and automatic image reduction framework for space debris detection based on GPU technology. Acta Astronautica, 145, 332–341. https://doi.org/10.1016/j.actaastro.2018.02.009

Kosmidis, L., Rodriguez, I., Jover, Á., Alcaide, S., Lachaize, J., Abella, J., … Steenari, D. (2020). GPU4S: Embedded GPUs in space - Latest project updates. Microprocessors and Microsystems, 77. https://doi.org/10.1016/j.micpro.2020.103143

Guerriero, V. W. Anelli, A. Pagliara, R. Nutricato and D. O. Nitti, "Efficient implementation of InSAR time-consuming algorithm kernels on GPU environment," 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2015, pp. 4264-4267, doi: 10.1109/IGARSS.2015.7326768.