Improve Method Using Spatial Fast Filters and GPU to Detection of Parcels of Land in Satellite Images for Cadaster Purposes

Luis Cadena, Franklin Cadena, Alexander Legalov, Alexander Zotin.

Abstract— **Nowadays, remotely sensed images are used for various purposes in different applications. One of them is the cadastral application using high resolution satellite imagery. The edge detection has an important role in image processing, especially in the detection and the extraction physical features, those which are useful to their enforcement in the analysis of cadasters. An improve methodology is shown for detection of parcels of land in satellite images using GPU and fast filter for improve time process to reduce the noise, dynamic histogram equalization instead classic histogram equalization and Canny edge detection which is the best of others as Sobel, Prewitt, etc, for accelerated time process we used parallel Matlab GPU.**

Index Terms— **satellite image, digital image, Canny edge detector, fast filters, dynamic histogram equalization, SSIM measure, parallel programming, GPU.**

I. INTRODUCTION

Nowadays, developments in remote sensing and image processing technologies have specifically provided the opportunity of determination of large areas in detail and in this respect, production of reliable, extended and recent data quickly. Thus, the fast developments in urban areas can be followed and strategies of directing those developments can be formed. In this respect, automatic object extraction have recently become necessary for large-scale topographic mapping from the images, determining the changes of topography and revising the existing map data. For mapping from high resolution imagery or GIS database construction and its updates, automatic object-based image analysis have been generally used for remote sensing applications in recent years.

Agricultural production, territorial organization, global warming, etc., validate the importance of the use of satellite images as a global means of capturing information for the inventory of natural resources. In this sense, one of the most relevant applications is the obtaining of cartography for cadastral purposes very important in the planning and administration of a territory; as well as the use and coverage of the existing soil in a geographic space. For this, different classification methods have been used, and restitution techniques have been used. Methods that mainly serve for low and medium resolution images; while today, new high-resolution satellites have been launched and even unmanned aerial vehicles are being used, which involve discrimination of smaller objects and the search for new classification methods to organize objects, as well as their Automation so that the user can generate studies of systems of agricultural production, afforestation, organization of the territory, environmental management, extraction of resources, etc. The characteristics of this area, shown in Fig. 1, besides, in the urban area the determination of parcels, roads and buildings is much easier than in city areas [1].

Manuscript received July 02, 2018. This work was supported by Universidad de las Fuerzas Armadas ESPE, Av. Gral Ruminahui s/n, Sangolqui Ecuador.

L. Cadena, is with Electric and Electronic Department at Universidad de las Fuerzas Armadas ESPE, Av. Gral Ruminahui s/n, Sangolqui Ecuador. (phone: +593997221212; e-mails: ecuadorx@gmail.com).

F. Cadena is with College Juan Suarez Chacon, Quito, Ecuador (e-mail: fcfc041@gmail.com)

A. Legalov is Siberian Federal University, 79 Svobodny pr., 660041 Krasnoyarsk, Russia (e-mail: alexander.legalov@gmail.com)

A. Zotin is with Department of Informatics and Computer Techniques, Reshetnev Siberian State University of Science and Technology, 31 krasnoyarsky rabochy pr., Krasnoyarsk 660037, Russian Federation (e-mail: zotinkrs@gmail.com)

II. DIGITAL PROCESSING OF IMAGES

The digital processing of images consists of algorithmic processes that transform an image into another in which certain information of interest is highlighted, and/or the information that is irrelevant to the application is attenuated or eliminated [2-10].

1- Fast filters for noise reduction.

Classic and fast average filter

Arithmetic average classic filter takes pixels from image to kernel matrix-window, evaluate average from kernel matrix and take average value and put in study pixel. Kernel size can be different and depends on task.

The fast averaging filter solves the same problem, but it is much faster. The fast filter blurs the image as the simple filter does. Using the following basic idea, it is possible to reduce the number of operations per pixel from W*W to ≈4: The filter first calculates and saves the sum of the gray values in each column of W pixels, while the middle pixel of each column lies in the current row of the image. The filter then directly calculates the sum over the window having its central pixel at the beginning of a row, i.e. by adding up the sums saved in the columns. Then the window moves one pixel along the row, and the filter calculates the sum for the next location by adding the value of the column at the right border of the window and by subtracting the value of the column at the left border. It is necessary to check, whether the column to be added or subtracted is in the image. If it is not, the corresponding addition or subtraction must be skipped.

Applying a similar procedure to sum up the columns, the average number of additions/subtractions per pixel is reduced to ≈2+2=4.

When proceeding to the next row of the image, the filter updates the values of the columns by adding the gray value below the lower end and subtracting the gray value at the upper end of each column. Also in this case it is necessary to check, whether the gray value to be added or subtracted is in the image.

As soon as the sum of the gray values in a window is calculated, the filter divides (with rounding) the sum by the number of pixels in the intersection of the window with the image and saves the result in the corresponding pixel of the output image.

Figure 1 shows difference in processing time of classical and fast average filter on image with size 512×512 for different kernel [10].

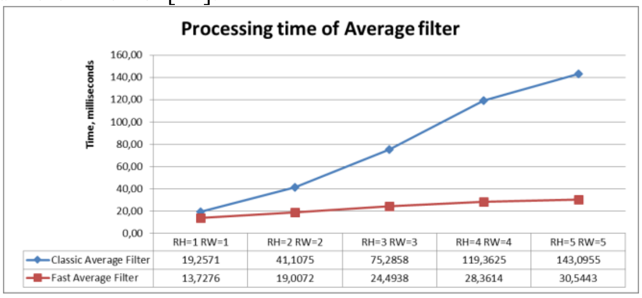

Fig. 1. Comparison of classical and fast average filter. Timing of the proposed algorithm [10] owned by author.

Classic and fast median filter

Classic median filter takes pixels from image to kernel matrix-window, kernel to array 1D and sort it, and take median term from array 1D and put in study pixel. The majority of the computational effort and time is spent on calculating the median of kernel.

Fast median filter based in histogram

The majority of the computational effort and time is spent on calculating the median of kernel. Because the filter must process every pixel in the image, for large images, the efficiency of this median calculation is a critical factor in determining how fast the algorithm can run. The classic implementation involves sorting of every entry in the kernel to find the median. However since only the middle value in a list of numbers is required, for median filter can be used much more efficient selection algorithms. Furthermore in image processing the histogram of spectrum for median calculation can be far more efficient because it is simple to update the histogram from window to window, and finding the median of a histogram is not difficulty.

Comparison of processing time of classical and histogram based fast median filter is shown on Figure 2.

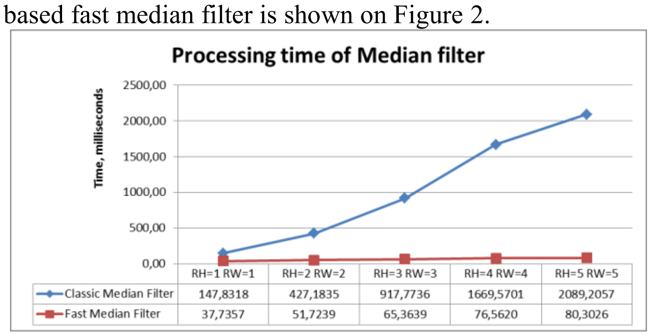

Fig. 2. Comparison of classical and histogram based fast median filter [10] owned by author.

Classic and fast 2D Gauss filter

Gauss 2D classic filter calculate kernel Gauss bell G(x,y), take pixels from gray value image A in kernel area and add to sum considering Gaussian coefficient, and put obtained value in study pixel in image B

The Gaussian filter uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image.

The equation of a Gaussian function in one dimension is

$$G(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{x^2}{2\sigma^2}}$$

In two dimensions, it is the product of two such Gaussians, one in each dimension:

$$G(x, y) = \frac{1}{2\pi \cdot \sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

Since the image is represented as a collection of discrete pixels it is necessary to produce a discrete approximation to the Gaussian function before perform the convolution. Depends on kernel size and σ some of coefficients can be out range of kernel. Theoretically the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel. In practice it is effectively zero more than about three standard deviations from the mean. Thus it is possible to truncate the kernel size at this point. Sometimes kernel size truncated even more. Thus after computation of Gaussian Kernel, the coefficients must be corrected that way that the sum of all coefficients equals 1. Once a suitable kernel has been calculated, then the Gaussian smoothing can be performed using standard convolution methods. The convolution can in fact be performed fairly quickly since the equation for the 2-D isotropic Gaussian is separable into y and x components. In some cases the approximation of Gaussian filter can be used instead of classic version [2-10].

Difference in processing time of classical 2D and double 1D implementations of Gaussian filter shown on Figure 3.

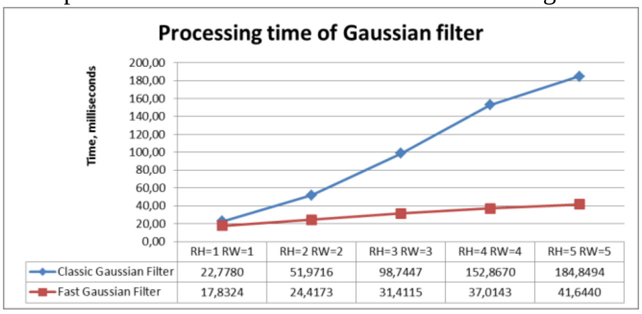

Fig. 3. Comparison of classical 2D and double 1D implementation of Gaussian filter [10] owned by author.

2- Dynamic Histogram Equalization of the image.

Histogram equalization is a spatial domain method that produces output image with uniform distribution of pixel intensity means that the histogram of the output image is flattened and extended systematically. This approach customarily works for image enhancement paradigm because of its simplicity and relatively better than other traditional methods. We acquire the probability density function (PDF) and cumulative density function (CDF) via the input image histogram. Apply these two functions PDF and CDF for replacing the input image gray levels to the new gray levels, and then we generate the processed image and histogram for the resultant image. And when we discriminate input image histogram with the processed image histogram we found that the gray level intensities are stretched and depressed systematically. Consequently, we obtain that the histogram of the output image is systematically distributed. Yet, this accords the over enhancement in images above the actual gray scale span. During histogram equalization approach the

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

mean brightness of the processed image is always the middle gray level without concerning of the input mean. This procedure is not very convenient to be enforced in consumer electronics, by the reason of that the method tends to introduce irrelevant visual deterioration like the concentration effect. The particular explanation for this issue is to conquer this weakness is by perpetuating the mean brightness of the input image indoor the output image.

Dynamic Histogram Equalization (DHE) was essentially popularized in 2007 by Wadud et al., to eliminate the influence of higher histogram components on lower histogram components in the image histogram and to regulate the amount of spreading of gray levels for objective enhancement of the image appearance by using local minima separation of histogram. DHE displays continuous and better enhancement of the image than the traditional paradigm. Withal, the DHE oversight the mean brightness perpetuation and influences to intensity saturation artifacts. DHE technique has overcome the drawbacks of histogram equalization and has shown a better brightness preserving and contrast enhancement than HE. DHE reinforces the image beyond making any destruction in image particulars. However, if user is not satisfied, may control the extent of enhancement by adjusting only one parameter. Besides, DHE is transparent and computationally adequate which makes it easy to implement and can be operated in real time systems.

DHE separates the histogram depends on local minima. Formally, it implements a one-dimensional smoothing filter on the histogram to dispose meaningless minima. Then it makes sub-histograms taking the portion of histogram that falls between two local minima [11].

The algorithm for the DHE is follow:

a) Input the image
b) Get the histogram of the image
c) Find the local minima in the histogram
d) Based on the local minima image histogram is divided
e) Assign specific gray levels to each partition of the histogram
f) On each partition of histogram HE is applied

3- Filters for edge detection.

Used following filters to edge detection: Prewitt, Sobel, Roberts, LoG and Canny, and compared its by measure SSIM. Following describe Canny edge detector.

The Canny edge detector is the first derivative of a Gaussian and closely approximates the operator that optimizes the product of signal-to-noise ratio and localization. The Canny edge detection algorithm is summarized by the following notation. Let $J[i,j]$ denote the image. The result from convolving the image with a Gaussian smoothing filter using separable filtering is an array of smoothed data,

$$S[i,j] = G[i,j;\sigma] * I[i,j]$$

where σ is the spread of the Gaussian and controls the degree of smoothing.

The gradient of the smoothed array $S[i,j]$ can be computed using the 2×2 first-difference approximations to produce two arrays $P[i,j]$ and $Q[i,j]$ for the x and y partial derivatives:

$$P[i,j] \approx (S[i,j+1] - S[i,j] + S[i+1,j+1] - S[i+1,j])/2$$
$$Q[i,j] \approx (S[i,j] - S[i+1,j] + S[i,j+1] - S[i+1,j+1])/2$$

The finite differences are averaged over the 2×2 square so that the x and y partial derivatives are computed at the same point in the image. The magnitude and orientation of the gradient can be computed from the standard formulas for rectangular-to-polar conversion:

$$M[i,j] = \sqrt{P[i,j]^2 + Q[i,j]^2}$$
$$\theta[i,j] = \arctan(Q[i,j], P[i,j])$$

where the arctan function takes two arguments and generates an angle over the entire circle of possible directions. These functions must be computed efficiently, preferably without using floating-point arithmetic. It is possible to compute the gradient magnitude and orientation from the partial derivatives by table lookup. The arctangent can be computed using mostly fixed-point arithmetic with a few essential floating-point calculations performed in software using integer and fixed-point arithmetic [12].

4- Morphological Treatment. Previous morphological operations are carried out with the intention of reducing noise. Then, the thinning of the elements that make up the image is performed. Small elements resulting from thinning are sought and eliminated [13].

5- Investment of the Binary Image. In order to work morphologically on the plots it is necessary to invert the image. First the erosion operation is carried out to finish separating the plots and then fill small gaps that the plots have. To eliminate unwanted objects the image is eroded and then dilated so that the plots take their original size.

6- Elimination of small elements. A minimum area value is defined and parcels that do not comply with this premise are searched and then eliminated.

7- Measure SSIM

The structural similarity (SSIM) index is designed to improve on traditional methods such as peak signal to noise ratio (PSNR) and mean squared error (MSE), which have proven to be inconsistent with human visual perception.

Structural information is the idea that the pixels have strong interdependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. Luminance masking is a phenomenon whereby image distortions (in this context) tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is significant activity or "texture" in the image.

The mean structural similarity index is computed as follows:

Firstly, the original and distorted images are divided into blocks of size 8 x 8 and then the blocks are converted into vectors. Secondly, two means and two standard derivations and one covariance value are computed from the images as:

$$\mu_x = \frac{1}{T}\sum_{i=1}^{T} x_i \qquad \mu_y = \frac{1}{T}\sum_{i=1}^{T} y_i$$

$$\sigma_x^2 = \frac{1}{T-1}\sum_{i=1}^{T}(x_i - \bar{x})^2$$

$$\sigma_y^2 = \frac{1}{T-1}\sum_{i=1}^{T}(y_i - \bar{y})^2$$

$$\sigma_{xy}^2 = \frac{1}{T-1}\sum_{i=1}^{T}(x_i - \bar{x})\,(y_i - \bar{y})$$

Thirdly, luminance, contrast, and structure comparisons based on statistical values are computed, the structural similarity index measure between images x and y is given by:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where c_1 and c_2 are constants [14-15].

III. SATELLITE IMAGE PROCESSING IN GPU

Recent tendency of GPU have gained great popularity in the field of high-performance processing for scientific and engineering applications such as image processing. GPU-based desktop computer's advantage of low cost, compact size hardware, high memory bandwidth and high parallelism are what make a this system appealing alternative to massively parallel system made up of commodity CPUs. A good example to GPU parallelism programming language CUDA (compute Unified Device Architecture) is given in NVIDIA. CUDA is remarkably increased programmability for escape from classical General-Purpose computing on Graphics Processing Units (GPGPU) way of transformation to Graphics API, for example OpenGL and DirectX.

The CUDA provides a programming model that is C++, extended with several keywords and constructs. The researchers encode a single program that contains both the CPU (host) and the GPU (device) code. Researchers using CUDA allows write device code in C++ functions called kernels. Kernel is disparate from a regular function in that it is executed by many GPU threads in a Single-Instruction Multiple-Data (SIMD) style. This style is called Single-Instruction Multiple-Threads (SIMT) that each thread executes the entire kernel once. Each GPU thread is given a special thread ID that is accessible within the kernel, through the built-in variables blockIdx and threadIdx. Threads have access to varied GPU memories during execution to kernel. Each thread can read and write or read or write its private registers and local memory. In addition, single-cycle access time, registers in the GPU memory hierarchy are the fastest. In contrast local memory in the GPU memory hierarchy is the slowest. Each thread block has its private shared memory. All threads have read and write access to the global memory, and read-only access to the constant memory and the texture memory. Since GPU threads can't access the host memory, the data by a kernel must be copy to above mentioned GPU memories before it is executed [16].

When used the GPU to compute, operation with high parallelism will produce a very high computational efficiency. The larger the scale of operation will get a higher efficiency. When used the GPU to compute, try to avoid loops and switches, and reduce times of data transfer between RAM and VRAM. According to the characteristics of the GPU, in scientific computing, CPU/GPU collaborative computing in large-scale scientific computing will get a better efficiency. In scientific computing [20]. Matlab is a visualization software which contains numerical analysis, matrix operations, signal processing and graphical display. It contains rich toolbox function, and can get a good solution of problems in the field of system simulation and calculation which encountered in the study. But Matlab computing efficiency is low, compared with other high-level languages, Matlab program execution is low. Scientific researchers always use two measures to improve the speed of Matlab. One is to buy some expensive equipment such as servers or workstations to increase hardware performance, although this method is able to solve the speed problem, but it gives researchers a financial burden. The other way is transplanting MATLAB algorithms and reprogramming by high level languages such as C++, which can improve efficiency of computing, but it demand programming skills for researchers, and some algorithms are complex high-level languages such as C++ don't provide the appropriate library functions, this requires researchers to start at the bottom. It would be very time-consuming and laborious, and increase research effort.

In order to solve this problem, MathWorks announced that by using Parallel Computing Toolbox, MATLAB can provide support for NVIDIA GPU. This support will allow engineers and scientists make MATLAB computing to be faster, and without having to do reprogramming, or increase equipment cost [17-21].

This paper is to examine the effectiveness of Matlab as a tool to express parallel computation with performance characteristics on GPU, and we propose this study for improving the speed of high-resolution satellite image data processing.

III. EXPERIMENTAL RESULTS

Image processing tasks include noise suppression, contrast enhancements, removal of undesirable effects on capture such as blurring or distortion by optical or motion effects, geometric mapping, color transformations, and so on. The big problem with edge detection is its great sensitivity to noise. Previous morphological operations are carried out with the intention of reducing noise. Then the thinning of the elements that make up the image is performed. Small elements resulting from thinning are sought and eliminated [13]. Finally, various morphological operations are carried out. In general these are dilation to reverse the thinning.

Methodology

This work propose follow algorithm to detection of parcels of land in satellite image.

1- Noise Reduction using Fast Gaussian filtering.

2- Dynamic Histogram Equalization of the image

3-Edge detection using operator Canny.

4- Morphological Treatment.

For the experiment used Notebook ASUS K75VJ core i7, 16Gb RAM, graphic card Nvida GeForce GT-635M, software Matlab 2015 (The MathWorks) [22].

The figure 4-8 show the process to obtained the parcel detection for the satellite image.

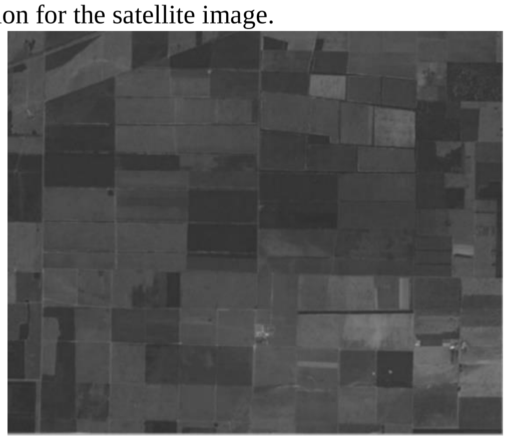

Fig. 4. Model image for experiment.

Fig. 5. Histogram equalization with DHE.

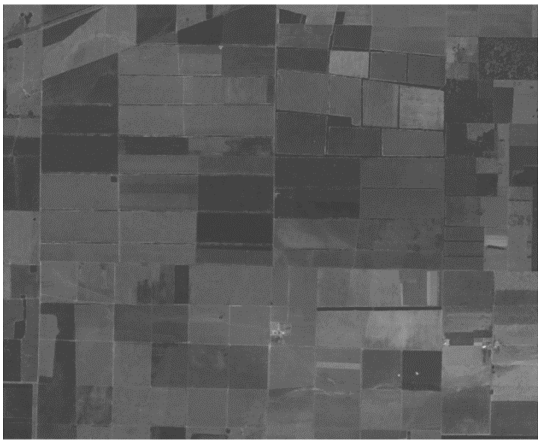

Fig. 6. Reduce the noise with media filter.

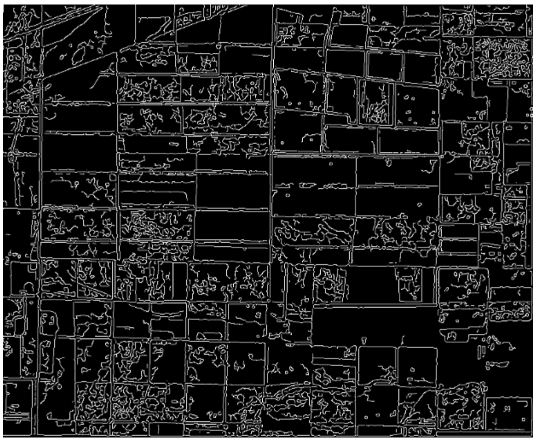

Fig. 7. Results of the edge detection by Canny.

Evaluation time process in seconds for noise reduction with classic and fast filters for kernel 5×5, the results are shown in Table I.

TABLE I.
TIME (s) PROCESSES FOR NOISE REDUCTION WITH CLASSIC AND FAST FILTERS FOR KERNEL 5×5

Filters	Time (s)
Classic average	0.0198
Fast average	0.0150
Classic median	0.0339
Fast median	0.0073
Classic 2D Gauss	0.0434
Fast 2D Gauss	0.0428

Evaluation fast spatial filters for noise reduction with SNR metric for kernel 5×5, the results are shown in Table II.

TABLE II.
SNR(dB) MEASURE IMAGE QUALITY FOR NOISE REDUCTION FOR KERNEL 5×5

Filters	SNR (dB)
Fast average	20.5706
Fast median	22.1724
Fast 2D Gauss	19.1489

Evaluation edge filters Sobel, Prewitt, Roberts, Canny and LoG with the SSIM metric and the results are shown in Table III and Figure 8.

TABLE III.
SSIM MEASURE IMAGE QUALITY FOR CONTOUR IMAGES

	Sobel	Prewitt	Roberts	Canny	LoG
Sobel	1.0000	1.0000	0.9991	0.9955	0.9977
Prewitt	1.0000	1.0000	0.9991	0.9955	0.9976
Roberts	0.9991	0.9991	1.0000	0.9953	0.9973
Canny	0.9955	0.9955	0.9953	1.0000	0.9963
LoG	0.9977	0.9976	0.9973	0.9963	1.0000

Fig. 8. Image for SSIM measure value 0.9977.

IV. CONCLUSIONS

The automated process decreases time, expense and provides users with another mechanism for obtaining parcels of land of satellite images, effectively and reliably, will be used in cartography for cadastral purposes which is important for the planning and administration of a territory.

Many of the parcels were not detected due to the high noise content their possess. Based on the experiment observed and taking into account that false parcels detected are few, we can say that the results are satisfactory.

From the parcels detected, some cases were found that are presented as one when they would appear to be two or more. Others are divided when they seem to be one.

To improve the time process and quality, were used media fast filter for reduce the noise, Dynamic Histogram Equalization, Canny edge detector, programming GPU in Matlab, and SSIM metric with satisfactory results which show in Tables I, II, III.

REFERENCES

[1] M. Alkan and M.A. Marangoz. Creating cadastral maps in rural and urban areas of using high resolution satellite imagery. Applied Geoinformatics for Society and Environment 2009 - Stuttgart University of Applied Sciences. Pags 89-95.
[2] Gonzalez RC, Woods RE. Digital Image Processing 3rd edition, Prentice-Hall, 2008. ISBN-13: 978-0131687288, (2008)
[3] Szeliski R. Computer vision. Algorithms and applications. Springer-Verlag London Limited. 2011.
[4] Chandel et al. Image Filtering Algorithms and Techniques: A Review // International Journal of Advanced Research in Computer Science and Software Engineering 3(10), pp. 198-202, (2013)

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[5] Lukin A Tips & Tricks: Fast Image Filtering Algorithms. 17-th International Conference on Computer Graphics GraphiCon'2007: 186–189, (2007).
[6] Kovalevsky Vladimir. Noise suppression. 2010. Available at http://www.kovalevsky.de/ImProc/Noise_e.htm
[7] Perreault S, Hebert P. Median filtering in constant time. IEEE Transactions on Image Processing 16(9): 2389–2394, (2007)
[8] Gupta B, Singh Negi S Image Denoising with Linear and Non-Linear Filters: A REVIEW // International Journal of Computer Science Issues, Vol. 10, Issue 6, No 2, pp. 149-154, (2013)
[9] Pascal G. A Survey of Gaussian Convolution Algorithms. Image Processing On Line 3: 286–310, (2013)
[10] Luis Cadena et al. "Noise Reduction Techniques for Processing of Medical Images," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2017, 5-7 July, 2017, London, U.K., pp496-500.
[11] Ravindra Pal Singh and Manish Dixit. Histogram Equalization: A Strong Technique for Image Enhancement. International Journal of Signal Processing, Image Processing and Pattern Recognition. Vol.8, No.8 (2015), pp.345-352. http://dx.doi.org/10.14257/ijsip.2015.8.8.35
[12] Canny J.A. Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov. 1986.
[13] Gerardo Galo. Procesamiento de Imágenes Satelitales - Detección de Parcelas, available at https://prezi.com/gjb0_pswtwzm/procesamiento-de-imagenes-satelitales-deteccion-de-parcelas/
[14] Guan-Hao Chen, et al. Edge-based structural similarity for image quality assessment. 142440469X/06/$20.00 ©2006 IEEE. ICASSP 2006.
[15] Wang Zhou, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image Quality Assessment: From Error Measurement to Structural Similarity", IEEE Transactions on Image Processing, Vol. 13, No.4, pp600-613, April 2004.
[16] In-KyuJeong,Eun-Jin Im,JoonsooChoi, Yong-SeungKim, ChoenKim. Performance comparison of gpu and cpu for highresolution satellite image processing. The 33rd Asian conference on remote sensing. November 26-30, 2012. Pattaya, Thailand.
[17] Mingzhe Wang et al. Analysis of GPU Parallel Computing based on Matlab.
[18] Liu Shaobo, Liu Minggui, Zhang Guohua. Model of accelerating MATLAB computation based on CUDA [J]. Application Research of Computers, 2010,27 (6): 2140-2143.
[19] Zhong Lianbo. Comparative Analysis of GPU and CPU [J]. Technology and Market, 2009, 16(9): 13-14.
[20] Wang Heng, Gao Jianling. Research of MATLAB Calculations and Simulation based on GPU [J]. Journal of Guizhou University, 2012, 29 (6): 95-112.
[21] Nikolaos Ploskas, Nikolaos Samaras. GPU Programming in MATLAB. © 2016 Elsevier Inc.
[22] The MathWorks. Inc. Image Processing Toolbox™ User's Guide. 2015.