A Study for Choosing The Best Pixel Surveying Method by Using Pixel Decision Structures in Satellite Images

Seyyed Emad MUSAVI and Amir ABUHAMZEH

Key words: pixel processing, pixel surveying, image processing, satellite images

SUMMARY

Nowadays the processing based on pixel surveying does have many applications such as producing binary images from black and white, colored and spectral aerial images using different types of processing filters and masks extracting and detecting objects from images, interpreting photogrammetry and satellite images, decision making in GIS systems and new methods of decision making process such as fuzzy logic, neural networks and etc. which all of the mentioned applications depend on pixel decision making structures.
In these types of activities the main role is the responsibility of the programmer.
Due to large amount of visual information in a satellite image and the increasing development rate of these data and the developments in satellite and aerial image capturing and also the fact that the current computer hardwares face a lack of ability for fast and prompt process of these data, the need of an optimized way for time saving purposes with lower amount of processing is being felt more and more each day.
In this paper by using main structures of programming in pixel processing in satellite images, different methods have been investigated and tested and in the end the best and the most optimized method for pixel processing in satellite images for the mentioned applications is presented.

A study for choosing the best pixel surveying method by using pixel decision structures in satellite images

Seyyed Emad MUSAVI and Amir ABUHAMZEH

1. INTRODUCTION

Processing based on decision making on each element of an image have nowadays become very highly applicable in geomatics sciences. The unstoppable evolution of the world from analog towards digital has made computer systems and softwares the most important device vastly used everywhere and for everything and geomatics has been vastly influenced by this change.

The growing need for interpretation of satellite and aerial images, the subject of close ranging photogrammetry and the need for analyzing the highly accurate images of it, providing information system for geospatial information systems (GI S) which are closely in contact with images and pixels. Apparently images are one of the main tools for GIS.

Digital image processing has 10 steps [1] which the key to all of them is pixel processing, and in case the whole image is processed it is called pixel surveying. There are quite different algorithms for pixel processing but they are rather complicated and time consuming. The object of this study is to complete the task in an uncomplicated way.

For serving our purpose in programming works of this paper MATLAB software has been used. Since its easier to use and now a days is being used by specialists and engineers more than ever.

As it will be demonstrated all the algorithms are as simple as possible. Simplicity, time and accuracy are the main concerns for the goal of this paper and all the efforts have been made to fulfill this goal.

Pixel processing is the basic step of image processing and the purpose of this paper is to investigate the best algorithm for optimizing the procedure of pixel surveying and pixel processing.

2. WHAT IS PIXEL PROCESSING?

An image may be defined as a two-dimensional function f(x,y), where x and y are spatial (plane) coordinates, the amplitude of f at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. When x, y, and the intensity values of f are finite, discrete quantities, we call the image a digital image. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called picture elements, image elements, pels, and pixels.

From the storing point of view, each element represents a value which the color assigned to the pixel is recognized by that value or number and in case of the necessity it will be displayed on an output hardware such as a printer or monitor. There are different color systems for storing, processing and displaying images which RGB, HIS, CMYK are among them. In each of these systems, numbers assigned to pixels, methods of color formations using numbers, and storing methods are fairly different in comparison with the other ones but the concept of pixel exists in all of them.

In satellite sensors one or a set of arrays, lines, points, 2D sensors, sensitive to absorbed light, exist, which regarding the absorbed wavelength and intensity these sensors generate electric pulses. Which proportional to this pulse a digital number is assigned to that cell of the sensor. This number is the base and scale for the displayed color of that pixel.

In the pixel processing procedure on each pixel, in each band of an image a logical or mathematical operation takes place, in other words, every single pixel of that image will be processed.

Image correction, restoring, enhancement, and extraction of geometric and attribute information from an image create what is now called pixel processing. **

For instance producing a binary image with the color depth of more than 2-bit is a pixel processing procedure, which can be operated on an image in the spectral range of infrared and the result will be the division of the infrared area to two parts of black and white.

The term “pixel surveying” in this paper is referred to the action of choosing a specific pixel from anywhere in an image and consecutively repeating the same action till the last pixel.

One of the sections of each pixel processing algorithm is “pixel surveying”. In satellite images due to the high dimension and great number of bands, execution of pixel surveying alone, takes a lot processing time.

3. THE NECESSITY OF OPTIMIZATION OF PIXEL PROCESSING AND PIXEL SURVEYING ALGORITHMS USED IN SATELLITE IMAGES

Processing based on decision making on each element of an image have nowadays become very highly applicable in geomatics sciences. The unstoppable evolution of the world from analog towards digital has made computer systems and softwares the most important device vastly used everywhere and for everything and geomatics has been vastly influenced by this change.

The growing need for interpretation of satellite and aerial images, the subject of close ranging photogrammetry and the need for analyzing the highly accurate images of it, providing information system for geospatial information systems (GI S) which are closely in contact with images and pixels. Apparently images are one of the main tools for GIS.

Digital image processing has 10 steps [gonzales] which the key to all of them is pixel processing, and in case the whole image is processed it is called pixel surveying. There are quite different algorithms for pixel processing but they are rather complicated and time consuming. The object of this study is to complete the task in an uncomplicated way.

For serving our purpose in programming works of this paper MATLAB software has been used. Since its easier to use and now a days is being used by specialists and engineers more than ever.

As it will be demonstrated all the algorithms are as simple as possible. Simplicity, time and accuracy are the main concerns for the goal of this paper and all the efforts have been made to fulfill this goal.

Pixel processing is the basic step of image processing and the purpose of this paper is to investigate the best algorithm for optimizing the procedure of pixel surveying and pixel processing.

Due to the fast growing rate of development in data acquisition tools from the surface of the earth in Remote Sensing fields, the experts face a huge amount of data to be processed and day after day the number of sensors orbiting the earth increases. In addition to this by advances in electronic sciences access to more up to date and precise images is easier than ever. In other words sensors with better and higher spatial, spectral, radiometric and temporal resolution are being launched which in mean time grant access to higher accuracies and therefore to larger and more complete amount of data. This naturally raises the need for more capacitive storing and saving systems and especially in case of this paper, faster processors.

Applying some massive processing operations on pixels such as complex mathematical transformations for noise removal purposes, masks and filters for restoring and enhancement, feature extraction algorithms, automatic methods for satellite image interpretation and etc, put a roughly high pressure due to the processing on computer systems.

By the increase of the processing amount, processing machines should spend more time and it will consequently cost more money to complete the task and in some special cases such as collecting data for unexpected events, the lack of high speed processing systems might even cause more losses of life since the needed information has not been prepared in it’s right time.

In order to prevent the mentioned problems, it is the programmer who should, by using and creating the proper algorithms, help the pixel processing procedure to be done more quickly and lead to a faster and better and more accurate analysis, correction and extraction of necessary information.

Applying different algorithms on a single pixel processing procedure results in different time durations for finishing the operation, this means should the proper algorithm for pixel processing not be used, more time has to be spent for the procedure to be completed than the optimized duration for the same case, therefore choosing the optimized and proper algorithm will lead to faster processing and less processing load.

Following, by investigating and studying some basic programming algorithms and the calculated duration of procedures, the role of a right decision to choose an optimized algorithm for a specific pixel processing procedure will be discussed and elaborated.

4. DESCRIPTION OF STUDY

In order to choose the optimized algorithm for pixel processing, under different processing conditions, basic commands of programming have been taken into consideration.

"Pixel surveying" is one of the most common algorithms of pixel processing but due to the high resolution satellite images and their high dimension, applying the operation is extremely vital and time consuming.

In this paper the algorithms for pixel surveying have been based on a combination of basic loops in programming ("for" and "while").

According to this study approximately 100 different algorithms can be found for the purpose of pixel processing which have been classified into 3 categories in this study:

1) Algorithms in which the purpose is to find one specific grey level in an image (the number of the "target" grey level is 1) target grey levels are the grey levels which should be located in the image.
2) Algorithms in which the purpose is to find more than one grey level in an image (number of the target grey level can be "n")
3) Algorithms in which the purpose is to separate some pixels and to assign a similar value to those in some specific intervals.

Using this classification, 14 algorithms out of 100 were chosen, it is to be said that for algorithms in the 2nd category, the number of target grey levels are considered 5 and for the algorithms for the 3rd category, the number of intervals are 6.

Due to the very large dimension of satellite images, each algorithm was written with 10 masks, these masks are responsible for decision making for some neighboring pixels by one time execution of each of the 2 main surveying loops in the image.

The purpose of this operation is to study the effect of reducing the number of executions, of the 2 main loops, in the consumed time for surveying the whole image.

All the masks are created with direct programming and coding and not by using new loops and reversal functions and other reiterative programming methods, not only to achieve better accuracies in certain time durations and preventing the complexity of the program, but also to reach and achieve more feasible and clear results for analysis.

This type of programming has also increased the number of lines and density of the program which some advantages and disadvantages of this method will be discussed in the next section.

In this step all of the chosen algorithms were applied on satellite images by dimensions of 512×512, 1000×1000, 2000×2000, 3000×3000, 5000×5000 and the time durations for each one were calculated and saved.$ the used images are shown in figure 1.

For increasing the accuracy and reliability of time durations, the designed masks for each algorithm were applied 1000 times and consequently 1000 different time durations were achieved, the average time for each algorithm or masks used in algorithms has been considered as the “criterion” for processing time. the written algorithms are listed in table 1.

Table 1. The coded algorithms

	code	first Loop structure(horizontal)	Second loop structure(vertical)	Third loop	Decision structure
First Category	A1	for	for	none	If
	A2	while	while	none	if
Second Category	B1	for	for	for	If
	B2			none	If+or
	B3			None	If+elseif
	B4			none	switch
	B5	While	While	while	If
	B6			none	If+or
	B7			None	If+elseif
	B8			none	switch
Third Category	C1	For	For	for	If
	C2			None	If+elseif
	C3	While	While	while	If
	C4			None	If+elseif

To have a scale in hand for the accuracy of the operation, standard deviation of the achieved time durations from each algorithm or masks used in algorithms have also been analyzed in the results section.$ the standard deviation of each algorithm and each mask listed in table 2.

It is to be mentioned that due to simplicity of usage, being comprehensive and powerful in processing and high flexibility in programming, all programs and procedures have been written and completed using Matlab© R2008a also all the time durations mentioned in the results section have been derived using the own functions of Matlab software.

All of the images used in this paper are satellite images from different sensors in “JPEG” format and color depth of 24 bit and are the courtesy of sensor manufacturing companies. (the original images used in the paper are available). Also for achieving a higher accuracy, images have been used by their grey scale system of color instead of RGB, the transformation of an RGB image to a grey scale is made using Matlab. The estimation of any time duration of the procedure on an RGB image, is approximately 3 times more than the duration achieved in a grey scale image.

The CPU used for this experiment is an 8400 dual core processor manufactured be Intel company on a “SONY-VAIO-FW 160D (PCG-3B4L)” laptop with the listed configuration in table 3.

Table 2.standard deviation of the times for each algorithm (the image size is 512*512)

A1	0.000446478899093204	0.000432390256057451	0.000250394721273389	0.000179940312786864	0.000229154581286702	0.000192551794446048	0.000221942027023875	0.000256372956974315	0.000273326170291681	0.000822557022395987
A2	0.00268583215352311	0.000831964223684321	0.000405662752185416	0.00216390727055175	0.00640758452147653	0.000124219087396168	9.48435321810439e-05	0.000147033135331067	7.85002098907463e-05	0.00778301538272673
B1	0.00142734608179975	0.00154807264841982	0.00145787514124498	0.000179940312786864	0.00133838613783875	0.000192551794446048	0.00126027107690339	0.000256372956974315	0.00227998800718245	0.00238439943555529
B2	0.000585150859414525	0.000646063789846922	0.000499992440943201	0.000494466712167262	0.000730970566980171	0.00141345848270705	0.00209048504190104	0.00233566780057409	0.00276239182866202	0.00264919943469172
B3	0.000626637058423110	0.000599286462090032	0.000499175442044048	0.000492996241095897	0.00200146965875460	0.00242822085140122	0.00258845208287252	0.00287455012015895	0.00319464723845979	0.00391845694327779
B4	0.000501260343529944	0.000625823834419384	0.000329386960704645	0.000351487179964967	0.000328479156475878	0.00237607198795827	0.00250498434694252	0.00287448988827134	0.00392029720229025	0.00395651913193698
B5	0.00105994405676516	0.000772149679651751	5.56748028340711e-06	0.00286597935628805	0.00260464319245385	0.00315008880857168	0.00260718313586856	0.00271695857749651	0.00287082717000082	0.0355442651325966
B6	0.000623293873930178	0.000599844192597369	0.000513827265814841	0.000524060507055122	0.000674138113930668	0.00107623171013130	0.00214472332418568	0.00234001765996603	0.00254172180576961	0.00274683482629849
B7	0.000473606144279033	0.000817529635401891	0.000640611225974020	0.000538330531680634	0.00236610385408512	0.00250410398785832	0.00272590354980095	0.00307187222830386	0.00352726898746309	0.00429807725167577
B8	0.000457796696837600	0.000564188759748909	0.000450491193905735	0.000339589714080960	0.000356740718400502	0.00237545153330933	0.00259495022650686	0.00296994815626187	0.00357670893747277	0.00430188338420023
C1	0.00193219515560351	0.00167243553289621	0.00120774534092803	0.00112046089228456	0.00111215537784964	0.00108131803493778	0.00112573090338665	0.00393143061966398	0.00374070416218236	0.00337233799860545
C2	0.000763426444787110	0.00101085105431119	0.000793805204531222	0.00269863085630818	0.00231766956477190	0.00291889542972037	0.00382154277425821	0.00419701466196798	0.00487968124444724	0.00592285755234762
C3	0.00126568239331789	0.00119373360968318	0.00106539788821065	0.00107244601504903	0.000983604922589799	0.000965831263379391	0.00102647767317304	0.00372761022468741	0.00415272487461419	0.00367533424574120
C4	0.000659980203604405	0.00101441490602862	0.000798437585544130	0.00264030949265126	0.00291264786612905	0.00328131153451970	0.00413991743568861	0.00477522367724897	0.00505638027727031	0.00629300628244698

Table 3. The computer system configurations, used for studies$

Feature	Description
processor	Intel Core 2 Dou P8400 2.26GHz 32bit
Memory	4G DDR2
Graphic	ATI Mobile Radeon HD 3400 1530 MB
OS	Windows Vista Home Premium (6001) 32bit
Motherboard	Intel m45 express chipset Bus Speed 1066 MHz

Figure 1. used satellite images (up left:512×512, up right:1000×1000, down left:3000×3000, down right:5000×5000)

5. ANALYSIS:

5.1 Role of loops:

In all of the written algorithms, one of the loop structures “for” or “while” for pixel surveying has been used. In the “for” algorithm from the processing point of view an “addition” operation and a “compare”, simultaneously take place. This means that in every execution command the compiler, regarding the “step” of the loop, adds a number to the variable used in the loop and then compares it with the number that has been defined to refer to the number of the loop execution.

In the algorithm using “while” only one act of “compare” will take place. This means that through the continuation of the pixel surveying, the counter of the loop should increase in the end of the loop. It can be theoretically implied that the processing load in the pixel surveying procedure is nearly the same as the “for” algorithm.

The achieved result on the comparison of loops are illustrated in figure 3 which can be easily seen that despite the simplicity of “for” loops and the similarities between the “for” and “while” loops, in the procedure of pixel surveying, “while” loops did provide better time duration. This means that by using “while” loops instead of “for” loops in the pixel surveying procedure the processing time duration decreased approximately 50% which is a significant achievement in optimizing the pixel surveying procedure.

5.2 Role of masks:

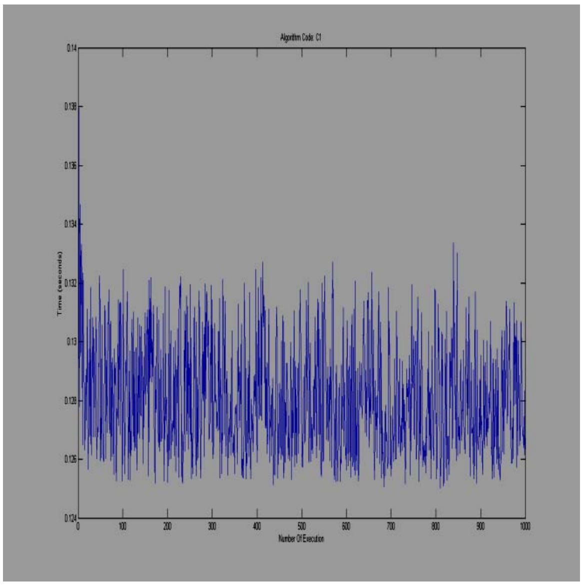
All of the algorithms which are available in the code table 1 have been written with 10 square shaped masks with dimensions of 1, 2,…, 10. As seen in figure number 2despite of the simplicity of the masks, they can reduce the processing load, therefore by increasing the dimension of the masks, the duration of processing decreases and this is because of less execution of main loops. When an $n \times n$ mask is moving on an image, with one time execution of both the main loops, which makes the mask move on one band of the image, for an n2 number of pixels, the execution process is decreased. The number of the loops and the total amount of processing load will decrease and as a result, the time of the processing will decrease.

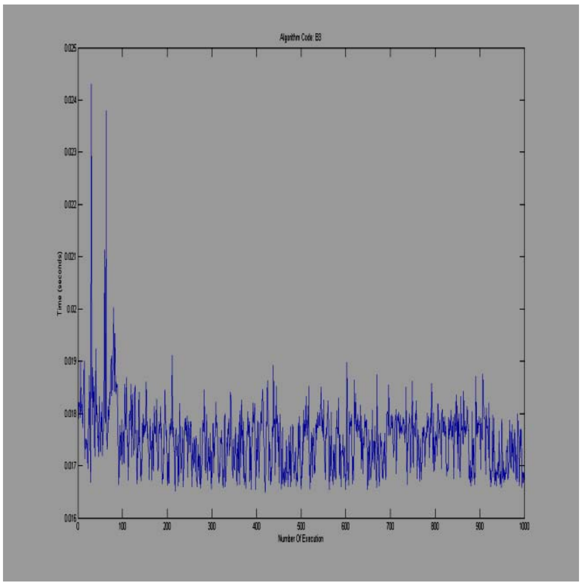
But it is to be emphasized that by increasing the dimension of the mask more than a certain limit, the speed of the process will decrease and the time duration will increase as well. It can be suggested that this is due to an escalation in the coding of these masks. As mentioned before all of the codes written for the masks are as simple and unadorned as possible by not using functions and other reiterative structures. You can see the number of lines that have been coded for each algorithm or mask in table 4

The more the coding the more the time processing, the reason for this is because the compiler should cope with a large amount of coding more than the optimized case, this can be seen in figure 4.

Table 4. number of code lines in each algorithm and mask.

81	430	360	510	360	981	1090	610	360	1030	1100	360	1580	540	2300
100	500	440	620	440	1200	1340	750	440	1270	1350	430	1940	650	2840
Algorithm Code Mask Side Dimension	A1	A2	B1	B2	B3	B4	B5	B6	B7	B8	C1	C2	C3	C4
1	20	30	30	20	30	30	30	30	30	40	30	40	30	40
4	40	40	40	40	70	60	50	40	80	80	50	100	60	140
9	90	60	80	60	130	150	90	70	140	150	70	200	90	280
16	130	90	120	90	220	230	140	100	220	240	100	330	130	480
25	170	130	180	140	330	350	210	130	340	360	140	500	190	730
36	230	180	240	170	460	500	280	180	470	510	180	720	260	1040
49	290	230	320	230	620	670	380	230	630	680	230	970	340	1400
64	360	290	410	290	786	860	490	290	810	880	300	1250	430	1830





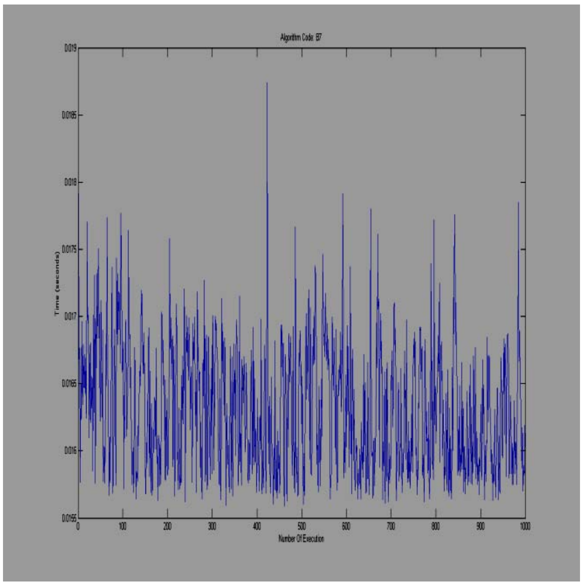
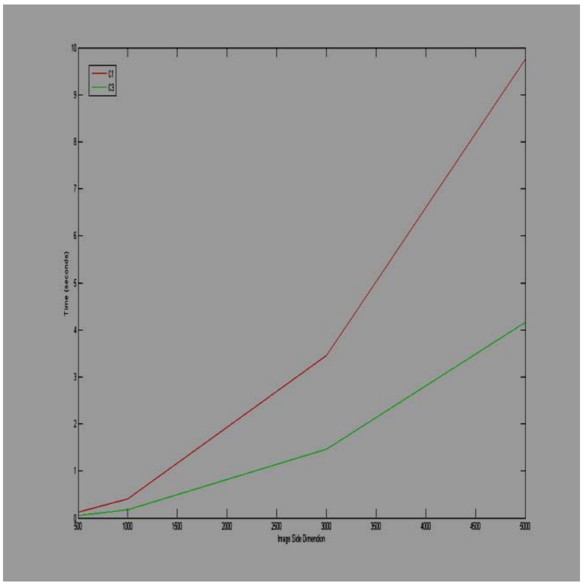
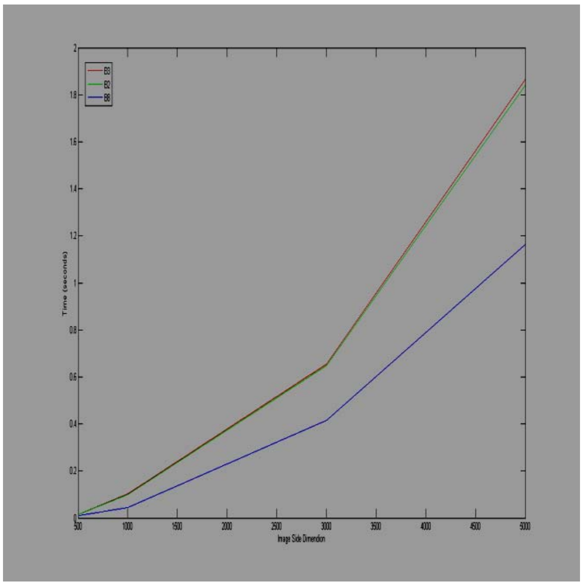
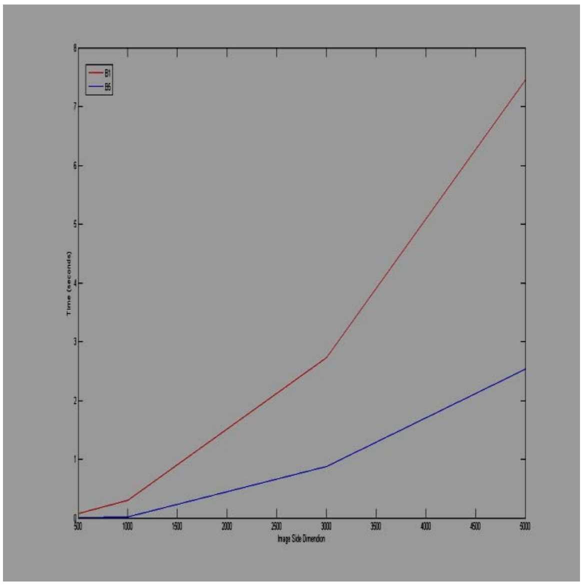
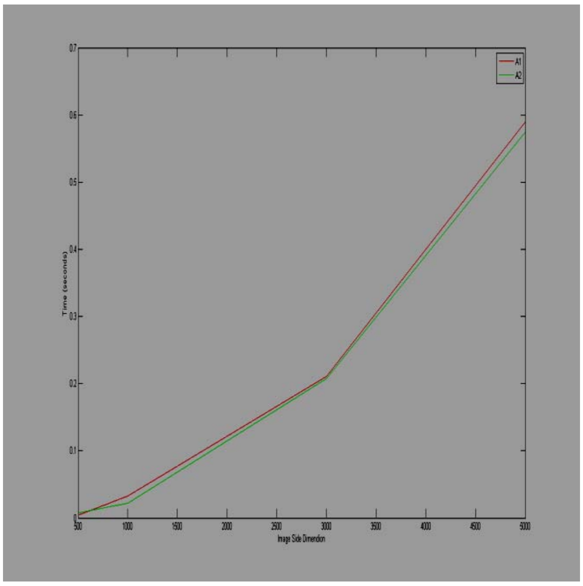


Figure 2.graph of achieved times for 1000 times execution of C1 (top left), B3 (top right) and B7 (bottom) on image with 512 side image









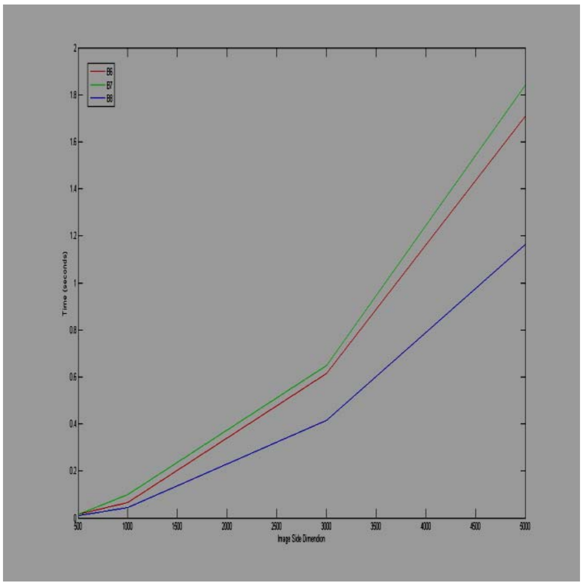
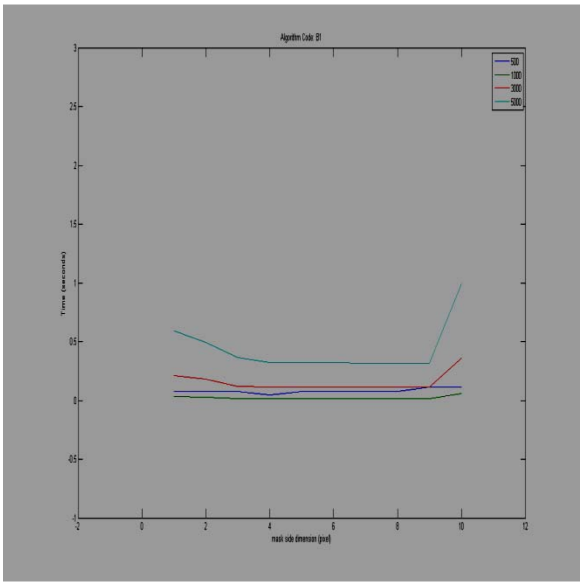
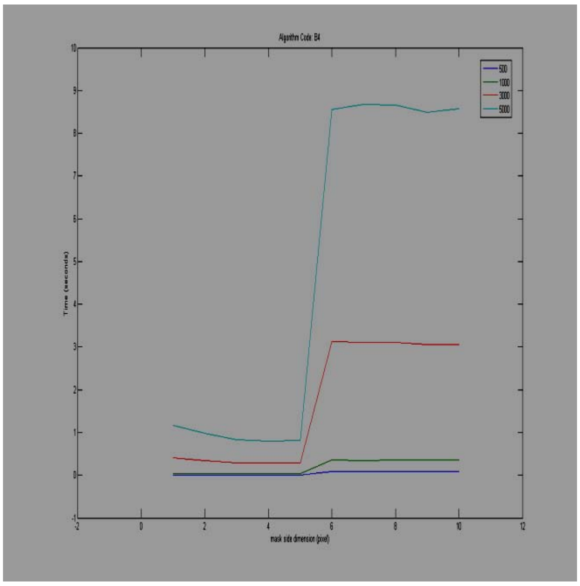


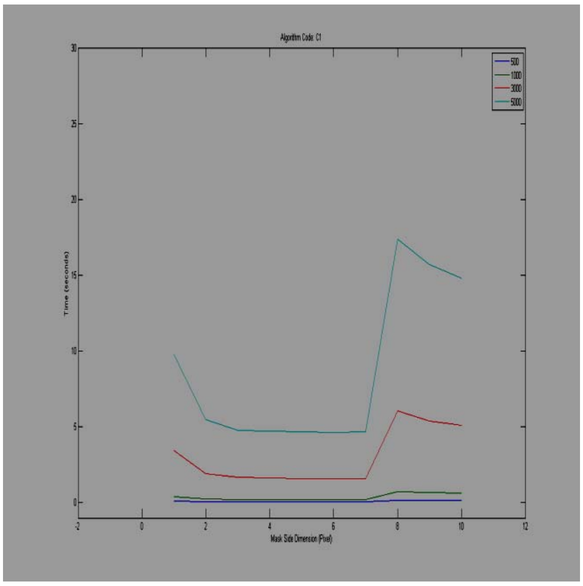
Figure 3.

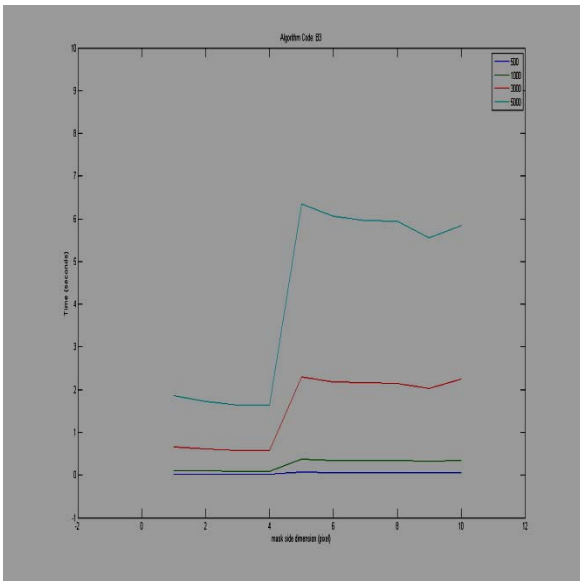
Comparing the execution time for all images between algorithms A1 (red) and A2 (green) (middle right), C1 (red) and C3 (green) (top left) and between B1 (red) and B5 (blue) (middle left).

Comparing the execution time for all images between algorithms B2 (green), B3 (red) and B8 (blu) (top right) and between B6 (red), B7 (green) and B8 (blue) (bottom)









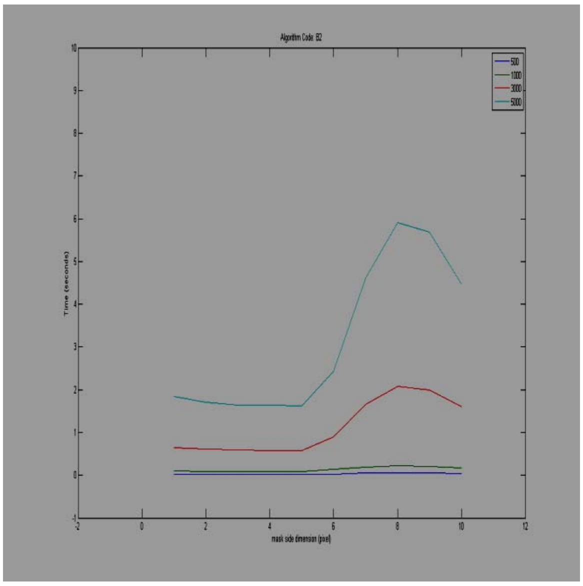


Figure 4.

Comparing all the mask size for image side dimension 512 for the algorithm B1 (top left), B4 (top right), C1 (middle left), B3 (middle right) and B2 (bottom)

5.3 Role of decision structures:

The decision making structures used in this study are simple structures, chosen from the basic commands of programming, the reason for this selection beside its simplicity, is that every pixel process is somehow a combination of the basic commands. Therefore comparing the results can make a good stand point for the programmers to evaluate and analyze the decision making commands and their optimized usage.

In the first studied structures seen in figure 3, as was expected the processing load of algorithms B2 and B4 are nearly the same, but the B8 algorithm in which the “switch” command has been used for the decision making process, for the same dimension images in comparison with other algorithms, the time duration is shorter.

Another point which can be seen in the graph is that by increasing the dimension of the image in case of using command, less time will be considered in the process. In the process of the second structure, the point made in the first one is also true. And in addition, in this figure, the difference between B8 and B6 has increased in comparison with the previous case, so it can be said that in case of not using the “switch” command, the combination of “if” and the logical gate “OR” is better than large number of “if” or “if+elseif”.

6. CONCLUSION AND RESULTS

In this paper 14 different algorithms for pixel surveying and pixel processing , and the experiment results were written, studied and investigated. For the masks it can be said that by increasing the dimension of the masks up to a certain number, the speed of processing can be increased but the amount of written codes by the programs increase as well. This advantage of speed is more effective in dealing with more than one or two images with a large amount of the calculations.

About the decision making structures, as can be seen in figure 3, the advantage of the switch command regarding 2 other structures which are both written by “if” commands, it can clearly be seen that the “OR” and “if” combination is more optimized than the “if-else-if”.

About loops as it is clear in figures 3 using the “while” loop is strongly effective in optimizing pixel surveying algorithm with their time duration of procedure.

At the end authors are willing to indicate some points. All of the represented analyses in this paper do not have any intention of entering the realm of specialized fields of “program evaluation”, but the only endeavor of the authors was to present an optimized applicable algorithm towards the processing of high resolution satellite images, resulting in the decrease of the processing load and consequently the time of the process and cost. Also it is to be mentioned that the results presented in this paper are all in a unique condition of software and hardware and any change in even small details of each of the mentioned parameters might cause a complete different result in comparison to this paper’s results, therefore the results as emphasized in the text are completely valid only for this paper’s conditions and can be compared in the same condition as this case, but in case of changing the condition, evaluation and comparing can only be done under a complete software and hardware specialist supervision.

AKNOWLEGEMENTS

Authors would like to thank Dr Saied Fazli, for his technical helps during the digital image processing course which was the inspiration for writing this paper.

REFERENCES:

[1]: R.C.Gonzalez and R.E.Woods and S.L.Eddins ,” *digital image processing*” ,2004,Prentice Hall. [2]: ISO Info. Processing Systems, Interfacing Techniques for Dialogue with Graphical Devices, Parts 1-6, IS9636, 1991.

[3]: A.K.Jane and N.J.Englewood cliffs,”fundamentals digital image processing”,1989 , Prentice Hall.

[4]: . M .Kasser and Y. Egels,“*Digital Photogrammetry*”,2002, Taylor and Francis.

CONTACT:

Amir Abuhamze

21 khorshid alley, toos street, felestin square, Tehran, iran-1416654167

aa_eng02@yahoo.com

Seyyed Emad Musavi

49 vahy street, khorramshahr street, zanjan iran-4515835533

Emad.musavi@gmail.com