

Texture Feature Extraction for Multispectral Satellite Image using Verilog HDL

P STEFFI SHAKILA¹, Dr.V VINOTH THYAGARAJAN², Dr.S.RAJARAM³

¹ PG scholar, Department of ECE, Thiagarajar college of Engineering, Tamilnadu, India

² Assistant professor, Department of ECE, Thiagarajar college of Engineering, Tamilnadu, India

³ Associate professor, Department of ECE, Thiagarajar college of Engineering, Tamilnadu, India

Abstract - Onboard processing of remotely sensed multispectral data is a highly desirable goal in many applications. For this aspiration, compact reconfigurable hardware modules such as field programmable gate arrays (FPGAs) are popularly used. Texture is one of the important characteristics used in identifying objects or regions of interest in an image. Texture features are used in different applications such as image processing, remote sensing and content based image retrieval. Feature extraction involves reducing the amount of resources required to describe a large set of data. These features can be extracted in several ways. The most common way is using a Gray Level Co-occurrence Matrix (GLCM). GLCM contains the second-order statistical information of neighboring pixels of an image. Textural properties can be calculated from GLCM to understand the details about the image content. In this paper propose a VLSI implementation for the extraction of four texture features using gray level co-occurrence matrix. Verilog language was used for hardware representation and Matlab was used for software simulation. The simulation part was carried out using Verilog HDL language in Quartus II tool and implementation was done in Altera cyclone II FPGA DE1 kit having EP2K20F484C7 device.

Key Words: Multispectral satellite image, gray level co-occurrence matrix (GLCM), features, field programmable gate arrays (FPGAs).

1. INTRODUCTION

Recently, Field Programmable Gate Array (FPGA) technology has become a viable target for the implementation of algorithms handling huge data set. In applications like remote sensing and satellite image processing the image size and bit, depth increases, as a result, the computational complexity and time increases. Hardware implementation offers the better solution, but one must consider the increase in development time inherent in creating a hardware design [1]. FPGA suites this types of application because of its prefabricated and reconfigurable structures. In electromagnetic spectrum the multispectral satellite image data within specific wavelength ranges. The multispectral image has 4-10 bands. Reconfigurable field programmable gate arrays (FPGAs) are promising platforms that allow hardware/software co-design and the potential provide effective onboard computing capabilities and adaptability at the same time [2]. The main challenge in

satellite image processing is reducing the data volume. FPGA provides an efficient compression of multispectral images in order to save bandwidth and storage space [3]. It reduces mathematical complexity, latency without losing reliability [6]. multispectral data onboard satellite imaging platforms in order to reduce downlink connection requirements and perform a more efficient exploitation of multispectral data sets in various applications [7].

Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately. When performing analysis of complex data one of the major problems stems from the number of variables involved [4]. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which over fits the training sample and generalizes poorly to new samples [5]. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Texture tactile or visual characteristic of a surface. Texture analysis aims in finding a unique way of representing the underlying characteristics of textures and represent them in some simpler but unique form, so that they can be used for robust, accurate classification and segmentation of objects. Though texture plays a significant role in image analysis and pattern recognition, only a few architectures implement onboard textural feature extraction. In this paper, Gray level cooccurrence matrix is formulated to obtain statistical texture features. A number of texture features may be extracted from the GLCM. Four second order features namely angular second moment, correlation, contrast, and entropy are computed. These four measures provide high discrimination accuracy required for motion picture estimation.

2. PROPOSED METHOD:

The proposed system design describe the analysis of texture feature parameter extracted using simulator and HDL. In this method the input is multispectral satellite image which is captured by IRS satellite. The input image is separated as a vegetation image and urban image by preprocessing method.

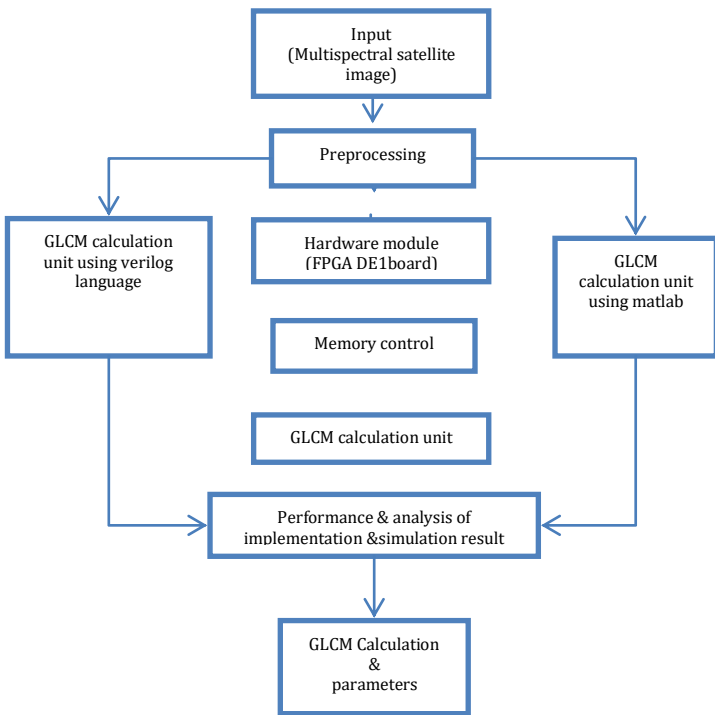


Fig-1: Design of the System

In this paper texture feature extraction for multispectral satellite image using GLCM has been used to identify the change detection. In preprocessing the image was converted into hex file because the hardware module does not understand the image. GLCM calculation unit used to extract the gray level co-occurrence matrix from the image. Four second order features namely angular second moment, correlation, energy, contrast are calculated from the GLCM matrix. Finally compare the feature parameters of hardware and software results. This process is explained in detailed manner as follows.

2.1 PREPROCESSING

2.1.1 Image specification

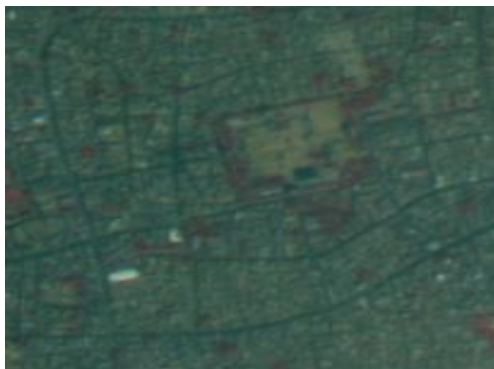


Fig-2: multispectral satellite image

The above fig-2 shows the input multispectral satellite image. The image was captured by IRS satellite using LISS IV sensor. This image have the three spectral bands 0.52µm-0.59µm (green), 0.62µm-0.68µm (red), 0.77µm-0.86µm (near IR). The latitude and longitude of that image is 9 ° 55'10.11"N to 9 ° 54' 37.07"N, 78 ° 7' 53.46"E to 78 °8' 38.60"E. The image shows the Madurai area acquired on march 5,2015. To know the vegetation and urban area the above image is cropped and it is shown in fig-3.

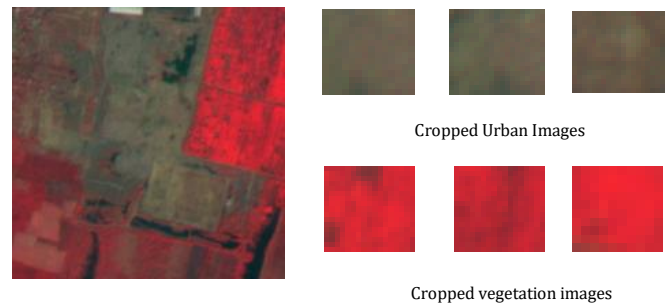


Fig-3: Preprocessing

In the preprocessing the input multispectral satellite image is cropped and separated by urban images and vegetation images in fig-3 . The cropped image size is(14*14). In QUARTUS II software it does not understand the image as a direct input .so that the image should be converted into hex file format. In that hex file the pixel value of the image are arranged in a array format is shown in fig-4. If the image size is(14*14) then the hex file have the 196 values arranged in a array format.

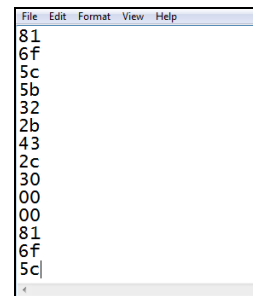


Fig-4: Hex file

2.2 GLCM CALCULATION UNIT

In statistical texture analysis, texture features are computed from the statistical distribution of observed combinations of intensities at specified positions relative to each other in the image. According to the number of intensity points (pixels) in each combination, statistics are classified into first-order, second order and higher-order statistics. The Gray Level Cooccurrence Matrix (GLCM) method is a way of extracting second order statistical texture features. The GLCM is created from a gray-scale image. The GLCM is calculates how often a pixel with gray-level value i occurs either horizontally, vertically, or diagonally to adjacent pixels with

the value j . The Gray Level Co-occurrence Matrix (GLCM) and associated texture feature calculations are image analysis techniques. Texture feature calculations use the contents of the GLCM to give a measure of the variation in intensity at the pixel of interest. The algorithm of co-occurrence matrices construction of a grayscale image can be summarized in fig-5 below

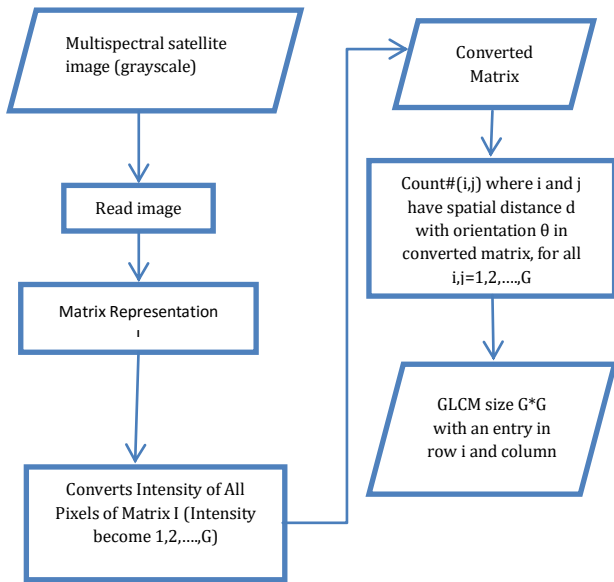


Fig-5 : construction of GLCM

Let $I(x, y)$ be used to represent a quantized image with M rows and N columns. The input parameters required for the computation of texture indicators are the distance between the neighborhood resolution cells d , the orientation α and the size of the rectangular neighborhoods' n .

The general steps followed in the implementation are:

1. Read the input image $I(x, y)$.
2. For every pixel $P(i, j)$ in $I(x, y)$ do:
 - a) Get the rectangular neighborhood for $P(i, j)$ of size n
 - b) Determine the maximum gray level (max_value) and minimum gray level (min_value) in the neighborhood.
 - c) Create a GLCM of size (max_value)
 - d) Count all the occurrences of gray levels i, j separated by distance d and an orientation α .
 - e) Normalize GLCM
 - f) Compute statistical indicator using GLCM: $output(i, j)$
 - g) Replace $P(i, j)$ with $output(i, j)$.
3. End

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Fig 6:sample GLCM calculation

The above fig-6 shows the sample GLCM calculation. The input matrix higher tone value is 8 so that the output matrix size is 8*8.

2.2.1 Realization of GLCM using Verilog HDL

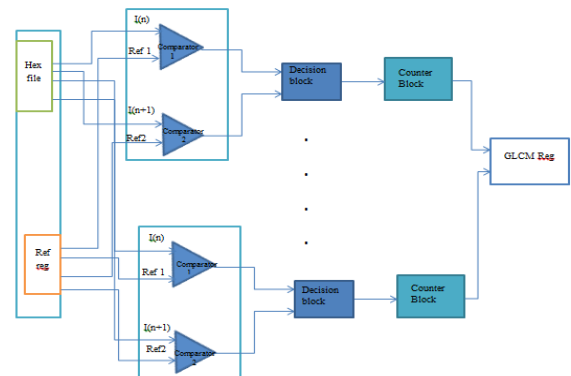


Fig-7 :Block diagram for GLCM using Verilog HDL

The above fig-7 shows the the GLCM calculation using Verilog HDL language. $I(n), I(n+1)$ shows the pixel values of the input image. The reference value is taken from the register. When the reference value is equal to the pixel value the output of the comparator given to the decision block. The decision block output is high when the two comparator value is high. Then the high value is stored in counter block. This process is continue until the whole pixel value of the image. The final output is stored in GLCM register. The parallel processing perform because to consume the time.

2.3 Extraction Of Texture Features Of Image

Gray Level Co-Occurrence Matrix (GLCM) has proved to be a popular statistical method of extracting textural feature from images. According to co-occurrence matrix, Haralick defines fourteen textural features measured from the probability matrix to extract the characteristics of texture statistics of remote sensing images. In this paper four important

features, Angular Second Moment, energy, Correlation, Entropy, and the Inverse Difference Moment are selected for implementation using Altera Quartus II 13.0 sp1.

2.3.1 Angular Second Moment

Angular Second Moment is also known as Uniformity or Energy. It is the sum of squares of entries in the GLCM. Angular Second Moment measures the image homogeneity. Angular Second Moment is high when image has very good homogeneity or when pixels are very similar

$$\text{Energy} = \sum_{i,j=0}^{N-1} P_{i,j}^2$$

Where $P_{i,j}$ = Element of the normalized GLCM

In Quartus II tool the input image is stored in register. Each image pixel occupies the 8-bit memory in a register. The image size is 14*14 then the input hex file contains 196 elements as shown in fig-4. The GLCM size is based on the maximum pixel value of that image. The maximum pixel value is 118 so that the output GLCM has the (118*118=13924) values. For the energy calculation in Quartus II, all the pixel value squares and the accumulation result are stored in separate registers. The MATLAB gives the normalized value of the energy. For the normalization purpose the accumulated value is divided by the sum of squares of the GLCM matrix; it gives the floating point value. In IEEE 754 floating point representation the input value is represented as a 32-bit value and the output value is also represented as a 32-bit value. IEEE 754 Standard Floating Point Numbers:

Sign bit(S)	Biased exponent bits(E)	Mantissa bits(M)
1 bit	8 bits	23 bits

Floating Point Division:

For the normalization purpose the floating point division is needed in Quartus II tool. The following steps show the 32-bit single precision floating point division.

Step 1: $S_3 = S_1 \text{Xor} S_2$

S_1, S_2 = input sign bit.

S_3 = output sign bit.

Step 2: $E3 = (E1 - E2) + \text{bias}$

$E3$ = output exponent bit; $E1$ & $E2$ = input exponent bits.

Step 3: Divide the mantissa bit (M1/M2)

$M1, M2$ = input mantissa bits

After that the floating point division then get the normalized energy value of the input multispectral satellite image using the GLCM algorithm.

2.3.2 Contrast

This statistic measures the spatial frequency of an image and is the difference moment of the GLCM. It is the difference between the highest and the lowest values of a contiguous set of pixels. It measures the amount of local variations present in the image. A low contrast image presents a GLCM concentration term around the principal diagonal and features low spatial frequencies.

$$\text{Contrast} = \sum_{i,j=0}^{N-1} P_{i,j} * (i - j)^2$$

In Quartus II the image pixel value is not stored in matrix format so that the i, j represent the place value of the image pixel, $P_{i,j}$ represents the pixel value of the image. To calculate the contrast, the sum of the pixel value is multiplied by the place value subtraction square of that image pixel. For the normalized contrast value the contrast value is divided by the sum of the GLCM matrix; it is represented as a floating point so that the floating point division is done to determine the normalized contrast value.

2.3.3 Homogeneity

This statistic is also called as Inverse Difference Moment. It measures image homogeneity as it assumes larger values for smaller gray tone differences in pair elements. It is more sensitive to the presence of near diagonal elements in the GLCM. It has a maximum value when all elements in the image are the same. GLCM contrast and homogeneity are strongly, but inversely, correlated in terms of equivalent distribution in the pixel pairs population. It means homogeneity decreases if contrast increases while energy is kept constant.

$$\text{Homogeneity} = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1 + (i - j)}$$

In the calculation of the homogeneity the pixel value of the image is divided by the place value subtraction of the image pixel so that the division operation gives the floating point value. For every pixel value it needs the floating point subtraction, division, and addition so that it creates the complexity to use the Quartus II tool software for this operation.

2.3.4 Correlation

$$\text{Correlation} = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - j) * P(i - j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

where μ_x, μ_y, σ_x and σ_y are the means and standard deviations of p_x and p_y respectively. The correlation feature is a measure of gray tone linear dependencies in the image. The rest of the textural features are secondary and derived from those listed above. The correlation operation also needs the floating point subtraction, division, and multiplication for every pixel so that the use of Quartus II tool is complex for this process.

3 RESULTS AND DISCUSSIONS

The simulation result of a FPGA implementation of texture feature extraction for the multispectral satellite image using GLCM algorithm is done using MATLAB, QUARTUS II software.

3.1 SIMULATION RESULTS USING QUARTUS II

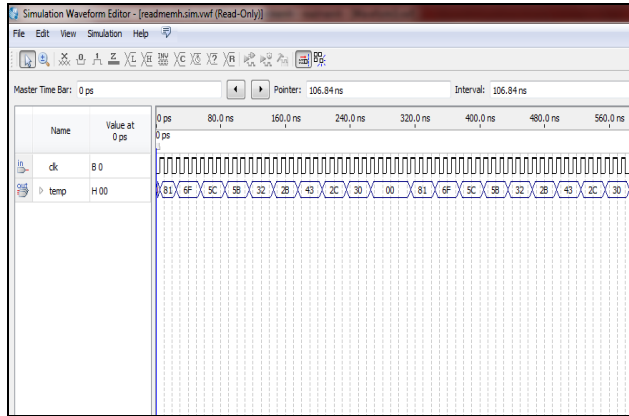


Fig-8 :Pixel values of multispectral satellite image

Every pixel value occupy 8bit memory in the register. The image pixel values are stored in hex file shown in fig-4. Using file reading the hex file is read and the pixel values are stored in temporary memory. The above fig-8 shows simulation result of pixel value of the multispectral satellite image

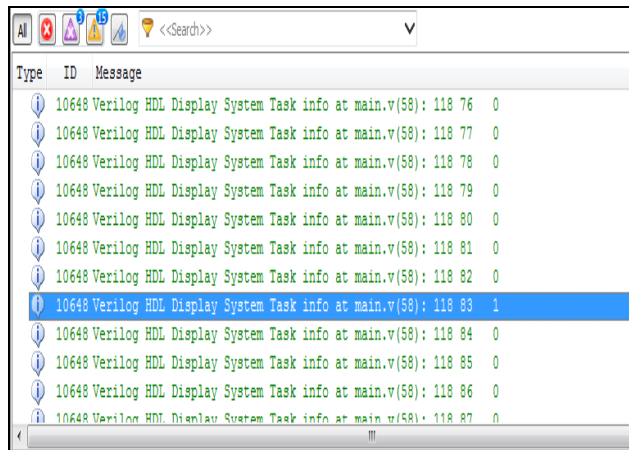


Fig-9: GLCM Using Verilog HDL Language

The above fig-9 shows the GLCM output of the multispectral satellite image using Verilog HDL language in Quartus II software. The input image size is (14*14) the maximum pixel value of that image is 118. So that the output GLCM have(118*118=13924) values.

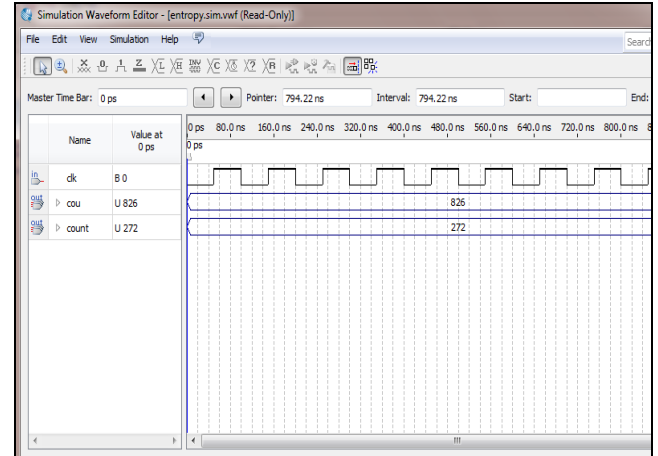


Fig-10: Angular second moment

INPUT : Clk(100 MHz)
OUTPUT: Count

The above fig-10 shows the energy value of the vegetation image. The count value (272) shows the the sum of the GLCM matrix. The cou value shows the energy of the vegetation image that is 826. For the normalization purpose the energy value divide by the sum of the GLCM matrix shown in fig-11. In quartus tool the floating point represented as a bit.

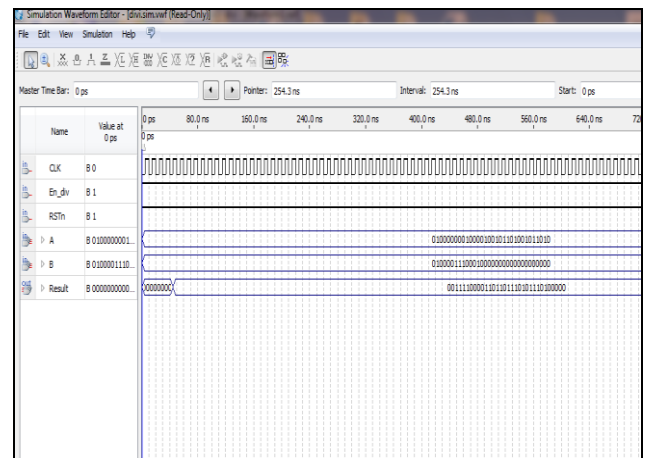


Fig-11: Normalized angular second moment=0.01116

INPUT : Clk (100 MHz),Rst,En-div,A,B
OUTPUT: Result

	Sign (1bit)	Exponent (8bit)	Mantissa(23bit)
Result (32 bit)	0	01111000	01101101110101110100000
	+	(120-127=-7) =2 ⁻⁷	1.429065704

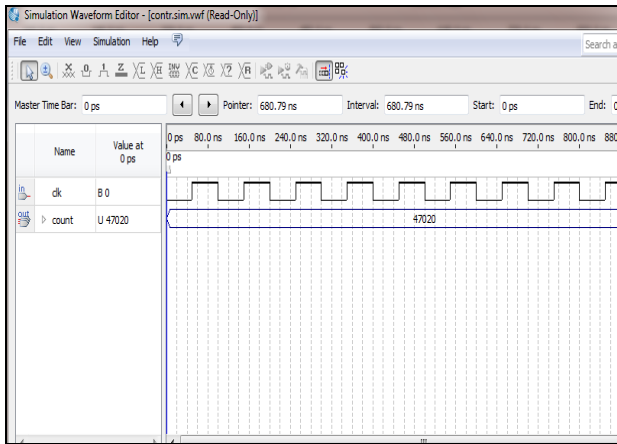


Fig-12: Simulated Contrast value

INPUT : Clk (100 MHz)
 OUTPUT: Contrast

The above fig 12 shows the output of the contrast value for the multispectral satellite image. For every clock cycle it calculate the contrast value. The fig 13 shows the normalized value of the multispectral satellite image. It is represented as a floating point IEEE754 format. In Quartus tool the floating point represented as a bit.

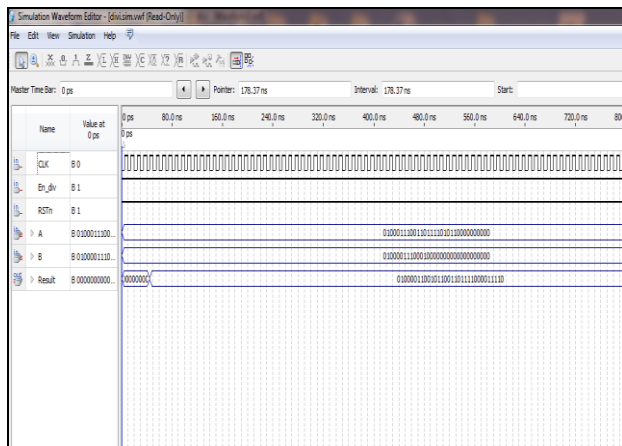


Fig-13 : Normalized contrast=172.86

INPUT : Clk (100 MHz),Rst,En-div,A,B
 OUTPUT:Result

	Sign (1bit)	Exponent(8bit)	Mantissa(23bit)
Result (32 bit)	0	10000110	01011001101111000011100
	+	(134-127=7) =2 ⁷	1.350528478

3.2. SIMULATION RESULTS USING MATLAB



Fig-14: Input Image

In above fig(14) shows the multispectral satellite image of Madurai city. This Image is captured by IRS satellite sensor name is LISS-IV. It have 3 spectral bands namely green(0.52-0.59µm), red(0.62-0.68 µm),near infrared(0.77-0.6 µm).

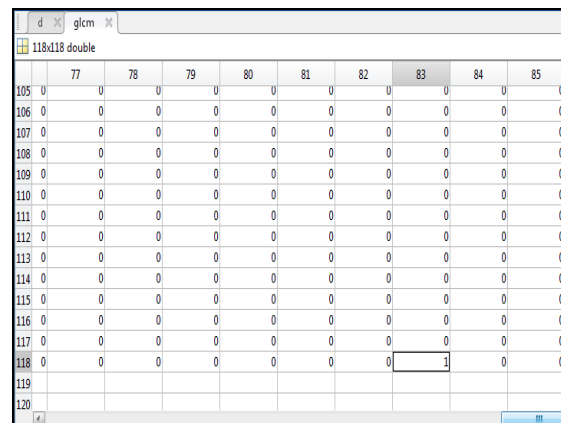


Fig-15: GLCM output

The above fig(15) shows the GLCM output of the input multispectral satellite image using MATLAB. The input image size is (14*14) the maximum pixel value of that image is 118. So that the output GLCM have(118*118=13924) values.

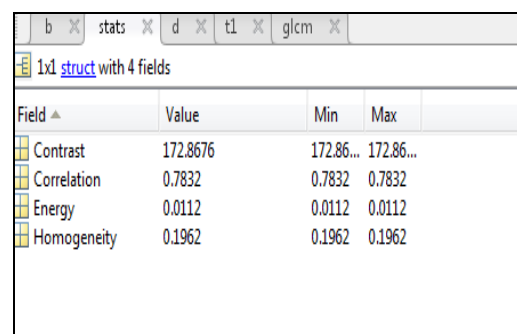


Fig-16 : Texture features

The above fig(16) shows the texture features for the multispectral satellite image namely contrast, correlation, energy and homogeneity.

Table 1: Data set for urban area image

S.No	Contrast	Correlation	Energy	Homogeneity
1	82.1758	0.8251	0.0154	0.2772
2	100.9804	0.8667	0.0102	0.2504
3	98.3137	0.8525	0.0095	0.2641
4	124.3342	0.8918	0.0085	0.2582
5	112.3342	0.8845	0.0090	0.2471

Table 2: Data set for vegetation area image

S.No	Contrast	Correlation	Energy	Homogeneity
1	172.8676	0.7832	0.0112	0.1962
2	150.2426	0.8013	0.0096	0.2120
3	234.1556	0.1736	0.0289	0.1818
4	154.6605	0.8090	0.0073	0.2418
5	257.7667	0.6895	0.0217	0.2419

The Table(1&2) shows the sample data set of texture features for the multispectral satellite image.

Table 3:Texture features Comparison

Image	Quartus output		Matlab output	
	Energy	Contrast	Energy	Contrast
Urban images	0.01539	82.1758	0.0154	82.1758
	0.01015	100.9804	0.0102	100.9804
Vegetation images	0.01116	172.8676	0.0112	172.8676
	0.00955	150.2426	0.0096	150.2426

The above Table(3) shows the comparison results of texture features extraction using GLCM method in quartus and matlab. The small variation in the result due to the truncation in quartus and round off in matlab software.

4. CONCLUSION

Recently Field Programmable Gate Array(FPGA) technology has become a viable target for the implementation of algorithms handling huge data set. The objective of the project was hardware implementation of texture feature extraction using GLCM algorithm on multispectral image and comparing the performance in term of feature extraction parameters for the simulation and implementation results. The objective has been achieved by using Matlab for simulation and Quartus II for hardware implementation. The novel propose of the extraction of texture features using Gray level co-occurrence matrix(GLCM) in verilog HDL language for multispectral satellite image was successfully implemented. The minute changes in the parameters due to the truncation effect in Quartus II software and round off in Matlab software

indicated in table(3). Thus the results of the simulation is compared and verified with the implementation. The entire process was implemented using Altera cyclone II EP2C20F484C7 device FPGA.

The real time implementation of dumping of the hex file of the chosen multispectral image in the onboard SRAM memory available in DE1 FPGA board was completed. As a future work, the sequential fetching of data from the memory and real time implementation of the texture feature extraction will be done.

REFERENCES

[1] Ernestina Martel, Raul Guerra, Sebastian Lopez and Roberto Sarmiento,—A GPU-Based Processing Chain for Linearly Unmixing Hyperspectral Images”, IEEE journal of selected topics in applied earth observations and remote sensing, vol. 10, no. 3, march 2017

[2]Carlos Gonzalez, Sergio Bernabe, Daniel Mozos, and Antonio Plaza,—FPGA Implementation of an Algorithm for Automatically Detecting Targets in Remotely Sensed Hyperspectral Images ||IEEE Journal of Selected Topics in Applied Earth Observations and Remote SensingVolume:9, Issue: 9, Sept. 2016 .

[3] Lucana Santos, Luis Berrojo, Javier Moreno, José Fco. López, and Roberto Sarmiento, —Multispectral and Hyperspectral Lossless Compressor for Space Applications (HyLoC): A Low-Complexity FPGA Implementation of the CCSDS 123 Standard”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing Volume:9, Issue: 2, Feb. 2016.

[4] V.Pavithra, R.Pounroja, Dr.B.SathyaBama, “Machine Vision Based Automatic Sorting of Cherry Tomatoes”, IEEE Sponsored 2nd International Conference On Eletronics And Communication System(icecs 2015).

[5] M.Harsha vardhan, S.Visweswara Rao, GLCM Architecture for Image Extraction, International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 3, Issue 1 January 2014

[6]Qian Xu, Srenivas Varadarajan, Chaitali Chakrabarti ,Lina J.Karam,—A Distributed Canny Edge Detector: Algorithm and FPGA Implementation”,IEEE Transactions on image processing volume: 23, Issue: 7, July 2014.

[7] Sebastian Lopez, Tanya Vladimirova, Carlos Gonza ´lez, Javier Resano, Daniel Mozos and Antonio Plaza, —The Promise of ReconFigureurable Computing for Hyperspectral Imaging Onboard Systems:A Review and Trends”IEEE | Vol. 101, No. 3, March 2013.

[8] Sebastian Lopez, Tanya Vladimirova, Carlos Gonza ´lez, Javier Resano, Daniel Mozos and Antonio Plaza, —The Promise of ReconFigureurable Computing for Hyperspectral

Imaging Onboard Systems:A Review and Trends"IEEE | Vol.101, No. 3, March 2013.

[9] C. Gonzalez, S. Sanchez, A. Paz, J. Resano, D. Mozos, and A. Plaza, –Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing,|| Integr .VLSI J., vol. 46, no. 2, pp. 89–103, 2013.

[10] Carlos González, Javier Resano, Antonio Plaza and Daniel Mozos,—FPGA Implementation of Abundance Estimation for Spectral Unmixing of Hyperspectral Data Using the Image Space Reconstruction Algorithm"IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing,Vol. 5, No. 1, February 2012

[11] C. Gonzalez, D. Mozos, J. Resano, and A. Plaza, –FPGA implementation of the N-FINDR algorithm for remotely sensed hyperspectral image analysis,|| IEEE Trans. Geosci Remote Sens., vol. 50, no. 2, pp. 374–388, Feb. 2012.

[12]C. A. Lee, S. D. Gasster, A. Plaza, C.-I. Chang, and B.Huang, –Recent Developments in High Performance Computing for Remote Sensing: A review, || IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 4,no. 3, pp. 508–527, Sep. 2011.

[13] C. Gonzalez, J. Resano, D. Mozos, A. Plaza, and D.Valencia,—FPGA implementation of the pixel purity index algorithm for remotely sensed hyperspectral image analysis,|| EURASIP J. Adv. Signal Process.,vol. 2010, no. 969806, pp. 1–13, 2010