

## A Hardware Approach for Real-Time Fuzzy Wavelet Based Textures Segmentation

<sup>1</sup>Labonnah F. Rahman, <sup>2</sup>Md. Mamun, <sup>1</sup>H. Husain, <sup>1</sup>Mohd. Marufuzzaman and <sup>1</sup>Yushaizad Yusuf  
<sup>1</sup>Department of Electrical, Electronic and Systems Engineering, <sup>2</sup>Systems Design Lab,  
Universiti Kebangsaan Malaysia, 43600, UKM, Bangi, Selangor, Malaysia

---

**Abstract:** Texture refers to the surface properties that can be easily described by its primitives (ton) and their spatial relationship. Texture analysis is a process to find the shape, segment or identify the region of interest on the object. In this research, researchers approach to implement unsupervised texture segmentation in hardware. First, the Discrete Wavelet Transform (DWT) is used to extract the input image and sample it into different frequency bands. After this process, the input image becomes smaller and compressed. This input image is then fed into fuzzy K-mean clustering algorithm. Fuzzy K-mean is a well-known and precise supervised clustering algorithm that divides the image into different segmentations and many computations are needed for the correct segmentation assignment. Hence, the total execution time for segmentation is improved using compressed image as input. From the simulation result, images with 2-5 types of textures were successfully detected around 0.025-0.033 sec. The proposed hardware approach for wavelet based texture segmentation is able to reduce the execution time to enhance the performance for 128×128 pixels which can be considered as fast segmentation solution.

**Key words:** Texture segmentation, fuzzy, wavelet, FPGA, texture, Malaysia

---

### INTRODUCTION

Texture segmentation or image classification is the process of identifying the region of interest and grouping it into one partition. This texture segmentation is widely used in medical image, satellite picture processing or remote sensing of the earth resources. In general, there are two different type of segmentation algorithm, namely supervised and unsupervised. Supervised segmentation is a clustering algorithm that needs the prior knowledge on the number of cluster in order to do segmentation whereas the unsupervised segmentation is able to group the image based on its texture characteristic without prior knowledge of the types of texture or the frequency and orientation characteristics of the image. As the result, the unsupervised segmentation is more desirable especially for the real time segmentation application. Various texture segmentation algorithms have been developed to provide the most effective solution to detect the region of interest. Segmentation using histogram, edge detection, morphology and region based segmentation and clustering method are the most commonly approaches to perform the segmentation result (Lezoray and Cardot, 2002).

Lezoray and Cardot (2002) proposed to use the thresholding to perform the segmentation. The result of this segmentation is always over segmented and the

number of the segmentation is always higher than the expected. The second approach of the segmentation is edge detection based segmentation. Edge detection is identifying the discontinuity of regions using the texture characteristic (Kaganami and Beiji, 2009). If the noise exists in somewhere, the edge detection may treat the noise as another edge of region. Moreover, the edge pixels produced by the edge detection are quit discontinuous and it provides insufficient information to segmentation (Liu and Jezek, 2004). Morphological image processing is a window (normally 3×3) process that pass through the input image pixels by pixels using some logical operation such as AND, OR, XOR and NOT (Serra, 2007). This segmentation algorithm gives a promising result but it only can group the pixels into 2 distinct values (binary image segmentation) (Russ *et al.*, 2009). Region based segmentation is the direct construction of region of interest. It provides better solution than using the edge detection method (Qin and Clausi, 2010). The segmentation result using this approach is very depends on the merging criterion and it is impossible to reuse the same stopping criterion for all type of input images. In this research, the fuzzy discrete wavelet based approach is followed to perform the segmentation. Discrete Wavelet Transform (DWT) is the subband coding used in decomposing image into different bank of frequencies (Hussain *et al.*, 2009). The low

frequencies was decreased to make sure that the lowest frequency bank stores the small changes of image pixels while the highest frequency bank will stores the high frequency changes. From the compressed image and higher frequency banks, fuzzy classification has been performed to get the segmented image. With this approach, the image redundancy and computational complexity has been reduced by performing the DWT before segmentation. Fuzzy classification has also been used to assure that the texture with slow changes of pixels or periodically will be assigned into the same region.

The Field-Programmable Gate Arrays (FPGA) offers a potential alternative to speed up the hardware realization. From the perspective of computer-aided design, FPGA comes with the merits of lower cost, higher density and shorter design cycle. It comprises a wide variety of building blocks. Each block consists of programmable look-up table and storage registers where interconnections among these blocks are programmed through the hardware description language (Reaz *et al.*, 2007).

This programmability and simplicity of FPGA made it favorable for prototyping digital system. FPGA allows the users to realize easily and inexpensively their own logic networks in hardware. FPGA also allows modifying the algorithm easily and the design period for the hardware becomes shorter by using FPGA (Coussy *et al.*, 2009). This research project approaches to develop a fast segmentation algorithm using extraction method while Discrete Wavelet Transform (DWT) is used for extraction.

Lastly, the unsupervised texture segmentation algorithm has been modeled using the hardware description language VHDL for hardware realization. The use of VHDL for modeling is especially appealing since it provides a formal description of the system and allows the use of specific description styles to cover the different abstraction levels (architectural, register transfer and logic level) employed in the design (Mohd-Yasin *et al.*, 2004). In the computation of method, the problem is first divided into small pieces each can be seen as a submodule in VHDL.

Following the software verification of each submodule, the synthesis is then activated. It performs the translations of hardware description language code into an equivalent netlist of digital cells. The synthesis helps integrate the design work and provides a higher feasibility to explore a far wider range of architectural alternative (Aker *et al.*, 2008a). The method provides a systematic approach for hardware realization, facilitating the rapid prototyping of the fuzzy based subway train braking system.

## MATERIALS AND METHODS

After comparing and discussing all the segmentation algorithms and its possible improvement ways, this project is proposed to use first level DWT to extract the texture characteristics and at the same time reduce the execution time for the segmentation. For the segmentation part, fuzzy K-mean provides the best segmentation result among the others and it definitely becomes the choice for the project.

The objective of this research is presenting a fast, unsupervised segmentation algorithm as compare to previous projects. Since, fuzzy K-mean clustering itself is a supervised clustering algorithm and also the seed initialization is important to get the correct result, some adjustment must be make to fulfill the objective of project.

**Discrete Wavelet Transform (DWT):** Wavelet transform is the multi resolution analysis that computes for different segment of time domain signal at different frequencies (Hussain *et al.*, 2009). At the high frequencies, DWT has good time resolution and poor frequency resolution whereas good frequency resolution and poor time resolution for low frequencies. This property allows the sampling of signal with high frequency changes at short time (image) and also low frequency component for long duration (video frames).

DWT is the process to decompose the input signal into high pass and low pass frequency bank. Then the lower frequency banks may further sub sampled to high pass and low pass until it reach the desired decomposition level as shown in Fig. 1. After this process, the lower frequency banks store the small changes of frequency whereas the higher pass filter stored the fast changing component of the signal. If DWT is used for image processing then the low pass filter is stored all small changes of the pixel values (almost same as the original image) where the high frequency bank is stored the fast changing pixels value (the edge of the image) as shown in Fig. 2. DWT is widely used as the extraction method in texture analysis (Salari and Ling, 1995) and the image compression such as JPEG2000 (Cruz *et al.*, 2000). Instead of using convolution method, DWT also can be implemented using lifting scheme. The lifting scheme has higher speed of execution because it allowed 1st and 2nd generation wavelet to construct first (Kuzmanov *et al.*, 2002). The basic operation of lifting scheme can be described using Fig. 3.

**Split phase:** Let's assume the decomposition started at zero and defined the current data as  $\lambda_{0,k}$  where zero is the root level and k represent the current data. Then the data

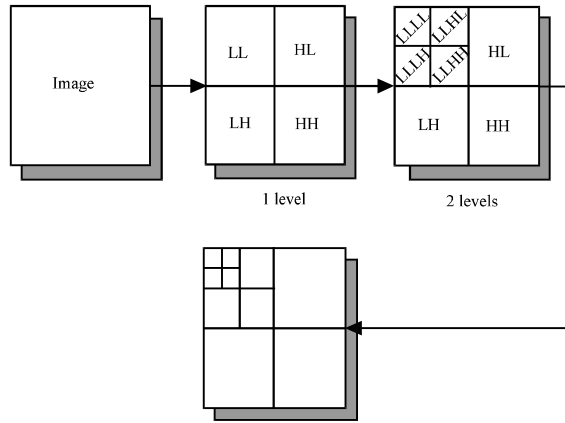


Fig. 1: The 2 levels DWT for image

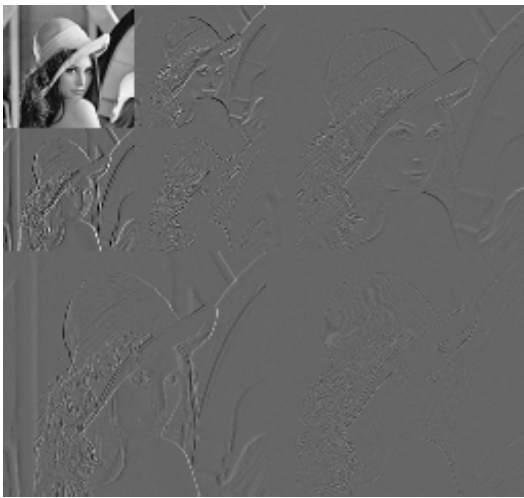


Fig. 2: The 2 levels DWT for Lena

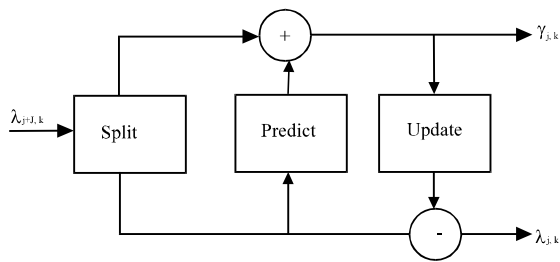


Fig. 3: Lifting scheme block diagram (Kuzmanov *et al.*, 2002)

is split into two samples namely even sample ( $\lambda_{-1,k}$ ) and odd sample ( $\gamma_{-1,k}$ ). The lazy wavelet transform is applied here because the filter only sub sampling the data into even and odd sample and not de-correlate the data. The negative indices are used to represent smaller size of data set. After the split phase, the samples become:

$$\lambda_{-1,k} = \lambda_{0,2k} \quad (1)$$

$$\gamma_{-1,k} = \gamma_{0,2k+1} \quad (2)$$

**Predict phase:** Based on the even sub-set  $\lambda_{-1,k}$ , the odd sub-set  $\lambda_{-1,k}$  is predicted using prediction function  $P(\lambda_{-1,k})$ . The predicted value  $\lambda_{-1,k}$  will be more accurate if more correlation present on the data. Later on, the value of  $\lambda_{-1,k}$  is updated by the difference between itself and predicted value:

$$\gamma_{-1,k} = \lambda_{-1,k} - P(\lambda_{-1,k}) \quad (3)$$

**Updates phase:** With the help of the neighboring wavelet coefficients ( $\gamma$ ) coefficients  $\lambda_{-1,k}$  are lifted using some scalar quantity such as mean:

$$Q(\lambda_{-1,k}) = Q(\lambda_{0,k}) \quad (4)$$

After that the updates function  $U$  operator is introduced and operator  $U$  must preserve the quantity by using the wavelet coefficient of the current level  $\lambda_{j,k}$  to update  $N$  even sample of the same level ( $\lambda_{j,k}$ ) where the  $N$  is the number of vanishing moment and may not equal to  $N$ .

**Le Gall 5/3 filters:** There are various types of DWT with different filter coefficients and NS had compared all these filter coefficients and concluded that 5/3 (five low pass coefficients and 3 high pass coefficients) is having the best performance among other coefficients (Jung and Prost, 1998). From the discussion, it is being concluded that the lifting scheme is proven much easier to implement compare to convolution approach and lifting scheme usage less resources and allowed pipelines stage in VHDL. Based on these, DWT using Le Gall 5/3 lifting filter is used in this research.

Assuming  $x(n)$  is the partial coefficient sequence and  $y(n)$  is the wavelet coefficient sequence, the DWT can be expressed as (Descampe *et al.*, 2006):

$$y(2n+1) = x(2n+1) - \left\lfloor \frac{x(2n) + x(2n+2)}{2} \right\rfloor \quad (5)$$

$$y(2n) = x(2n) + \left\lfloor \frac{y(2n-1) + y(2n+1) + 2}{4} \right\rfloor \quad (6)$$

Where:

$$\left\lfloor \frac{x(2n) + x(2n+2)}{2} \right\rfloor$$

is rounding operation to prediction function  $P_1(Z)$  and:

$$\left\lfloor \frac{y(2n-1) + y(2n+1) + 2}{4} \right\rfloor$$

is rounding operation for update function  $U_1(Z)$ . After deciding to implement DWT using Le Gall 5/3 lifting filter, the next consideration is how many levels of DWT needed to obtain the best segmentation result. Fauzi and Lewis (2003) compared the segmentation on different DWT levels and found out the root level (lowest frequency bank) provides most accurate result.

**Fuzzy clustering algorithm:** Clustering algorithm is grouping the data into  $k$  identical regions by selecting the shortest distance between itself and all cluster centroids. From the various type of fuzzy clustering, fuzzy K-mean is the simplest clustering algorithm to implement and at the same time provides satisfy segmentation result (Isa *et al.*, 2009). There are the steps that are followed to implement fuzzy K-mean:

**Step 1:** Sample all the data in to appropriate feature space.

**Step 2:** Assume that number of the  $K$ -clusters is known and assign  $k$  number of clusters into the feature space.

**Step 3:** For each of the pixels, calculate its distance to  $k$ -cluster centroids and assign the pixel into the cluster membership if the shortest distance was found.

**Step 4:** After finish assigning all the pixels, recalculate the cluster centroid by taking the average values of all the same cluster memberships.

**Step 5:** Go back to step 3 to evaluate the distance between pixel and cluster centroids until no changes of cluster membership.

Based on the understanding of the algorithm, fuzzy K-mean is actually a supervised segmentation algorithm where  $k$  is the number of clusters need to be known in advance.

Moreover, the decision of assigning minimum distance of cluster centroid is actually involving heavy computation, especially for larger values of  $k$ . The fuzzy clustering is also sensitive to the initial assignment and the algorithm might be trapped at local optimum if the values of  $k$  is small (Isa *et al.*, 2009).

**Hardware modeling:** As the project is about processing the segmentation, the input file is needed to feed inside the process and it must produce the output file once the

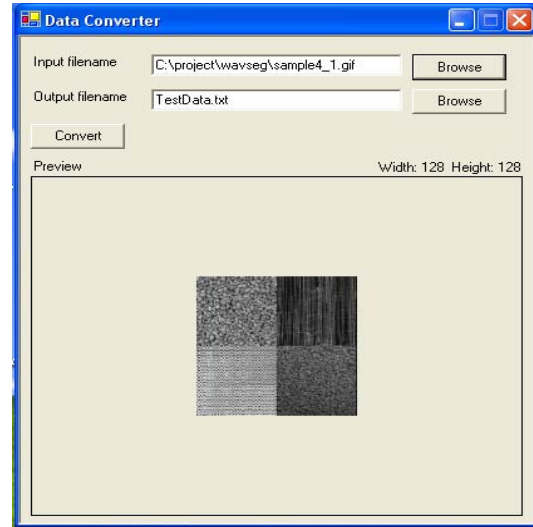


Fig. 4: Input image to TestData.txt using Data Converter.exe

execution was finished. In VHDL, the library STD\_TEXTIO and STD\_LOGIC\_TEXTIO support the input/output file conversion. STD\_TEXTIO is the standard input output library that supports the reading and writing process of file in VHDL environment. STD\_LOGIC\_TEXTIO is an advance reading and writing library that support these processes in binary, octal and hexadecimal forms. Depending on the format of input image, STD\_TEXTIO is picked when the data are in decimal or both the libraries are needed if input is in the form others than decimal.

**Image converter:** The next step was to find a converter to convert the image into array of data. These data are representing the pixel values in each position it defined. Actually, some software are available to perform this conversion but it mainly comes from the license software that not so convenient to use. Therefore in this research project, the DataConverter.exe used by Akter *et al.* (2008b) is replicated as the tool to convert the image into array of data.

This DataConverter.exe is written in Microsoft Visual Basic code and the execution can be done in any Microsoft Windows operating system that supported. NET framework. Perhaps, it is the best convert solution for this project because the code can be executed in most computers and conversion process involving click and drop only. Figure 4 shows the input image was converted into TestData.txt where in Fig. 5 the converter was used to view the image using TestData.txt. After deciding to use the DataConverter.exe, the format of the data representation was investigated for implementation in

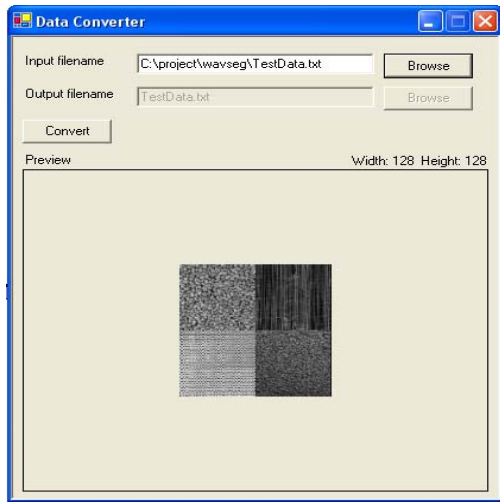


Fig. 5: Input file to image using DataConverter.exe



Fig. 6: Input output file conversion in whole segmentation process

VHDL. This converter is using 8 bit signed hexadecimal with 2 dimensional represented as width and height. Therefore, both STD\_TEXTIO and STD\_LOGIC\_TEXTIO library are need to perform these hexadecimal reading and writing. The process of converting picture can be summarized as shown in Fig. 6. At the end of entire segmentation process, the segmentation result can be viewed in SegResult.txt using DataConverter.exe.

**Top entity:** The overall texture segmentation module is comprised of three submodule. They are memory module, DWT module and segmentation module. The top entity of this project includes all these three modules in it. Figure 7 shows WavSeg as the top entity for this project which is responsible for granting access to DWT and segmentation. The clock signal for this top entity is inout instead of the internal clock to provide the flexibility to control by higher entity in future. As the top entity, WavSeg is responsible to allocate the task to all the lower entities and observe the switching data in memory. These task allocations are based on the acknowledgement signals from lower entities to identify each process is completed. Figure 7 shows the top entity (WavSeg) and its relationship to other entities. As shown DWT2D and seg2d are two distinct operations where the memory entity is shared by these two processes as external memory. The acknowledgement signals that are used in WavSeg entity are described as:

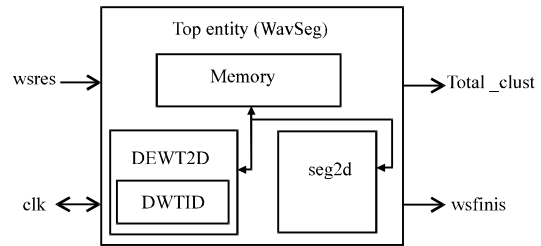


Fig. 7: Top entity with some lower entities

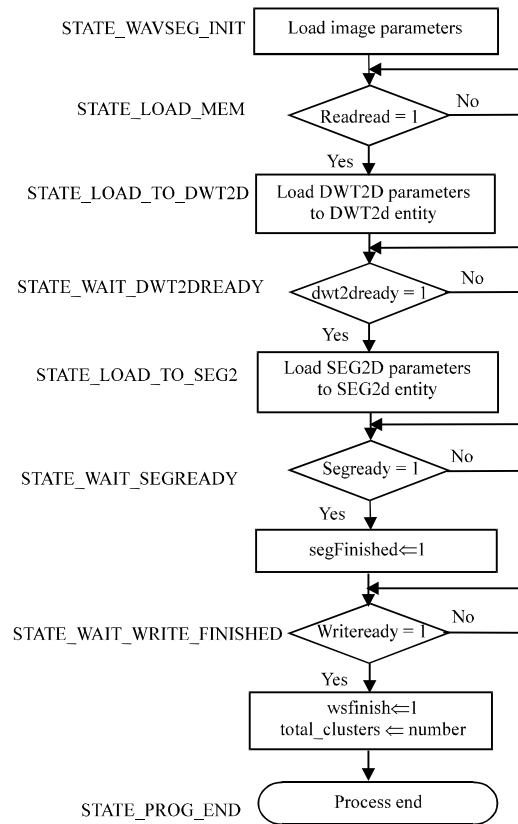


Fig. 8: Flowchart for Wav Seg module

- Readready signal is sent from memory entity to indicate the input image was stored in the memory matrix
- Dwt2dready used to indicate DWT process was done
- Segready seg2d sent this signal to acknowledge WavSeg that segmentation was finished
- Writeready signal is sent from memory entity to indicate the output result was fully written to output file

By using these few acknowledgement signals, the processes in segmentation can be traced and allocated easily by top entity. The flowchart for WavSeg was drawn as shown in Fig. 8.

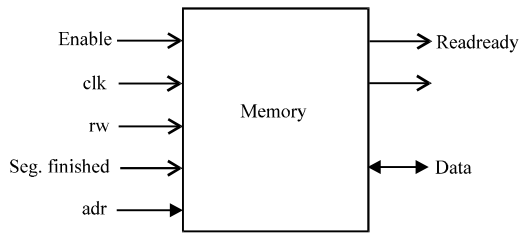


Fig. 9: Memory entity module

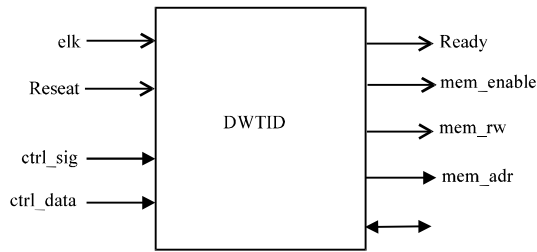


Fig. 10: DWT1D entity module

**Memory module:** The memory module in Fig. 9 is treated as external memory to hold array of data during the image processing. This module is needed when >2 different entities sharing the same data. Basically, Akter *et al.* (2008b) had presented a very strong and reliable memory module and some modification had been made to suit this project. In order to ensure the data is safely read in and write to memory, some memory control signals are needed to handle the read and write process. In this module, the rw signal is used to implement the read or write process. When rw is high then the memory performs the read process while the write process is performed when the signal goes low.

**DWT module:** As discussed earlier, first level Le Gall 5/3 filter is used as pre-process before the segmentation. Each of the 2D-DWT is the combination of horizontal 1D-DWT and vertical 1D-DWT (Descampe *et al.*, 2006). For an easy implementation, Akter *et al.* (2008b) implemented this DWT method using two different entities, namely DWT and DWT2D. DWT entity is responsible to perform 1D-DWT where DWT2D is the higher entity to control DWT entity to perform the decomposition. Thus, these two entities developed by Akter *et al.* (2008b) become the references design for the project.

DWT1D as shown in Fig. 10 is the entity to perform 1D-DWT after received the instruction from DWT2D entity. For enabling the communication between these two entities, ctrl\_sig and ctrl\_data were used to transfer in the control signal and data from the higher entity. Basically, DWT2D entity had almost the same design as

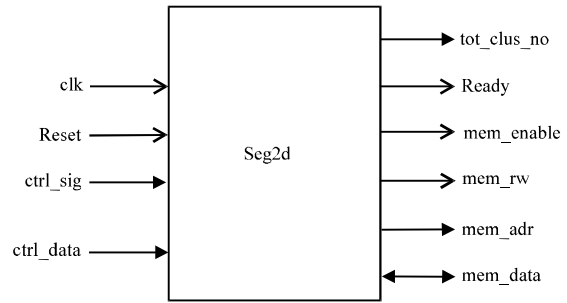


Fig. 11: Seg2d entity

DWT1D in Fig. 10. The only difference that the DWT2D module has is separate control signal stage. This control signal was controlled by the top level entity to perform 2D-DWT. In this case, 4 stages or 2 bits were used as control signal to control DWT2D.

The function of DWT1D is to calculate the high and low pass filter values and store it into respective addresses without knowing where these addresses come from. While, the function of the DWT2D is to assign all these useful addresses to DWT1D using DWT1D control signal function (Dwt1DCtrlSigFunc). In other words, DWT2D executed 1D-DWT line per line and responsible to determine the input and output addresses as well as the range for DWT1D.

**Segmentation module:** The seg2d as shown in Fig. 11 is the entity that is responsible to perform fuzzy K-mean clustering algorithm after DWT process. Fuzzy K-mean is an algorithm that involves heavy computation and refers the data frequently, so external memory is not desirable as it will degrade the execution speed and performance. Therefore, internal memory is used to temporally store all the results and only final result is send back to the external memory as shown in Fig. 9 to reduce the reading and writing time significantly. In general, the pins on seg2d are mostly equivalent to DWT2D except that seg2d has its own ctrl\_sig values and tot\_clus\_no as the output to store the total clusters number.

## RESULTS AND DISCUSSION

After successfully compiled the code, the correctness of the code is checked through simulation. The simulation was executed on Modelsim PE 5.3a and WavSeg as the top entity is loaded to perform the simulation. According to Fig. 7, WavSeg can be divided into different phases. Here, the final simulation state of the WavSeg entity has been depicted. For the simulation purpose, the clock is set to 50 ns with 50% duty cycle. Firstly, the input image was converted to TestData.txt using DataConverter.exe.

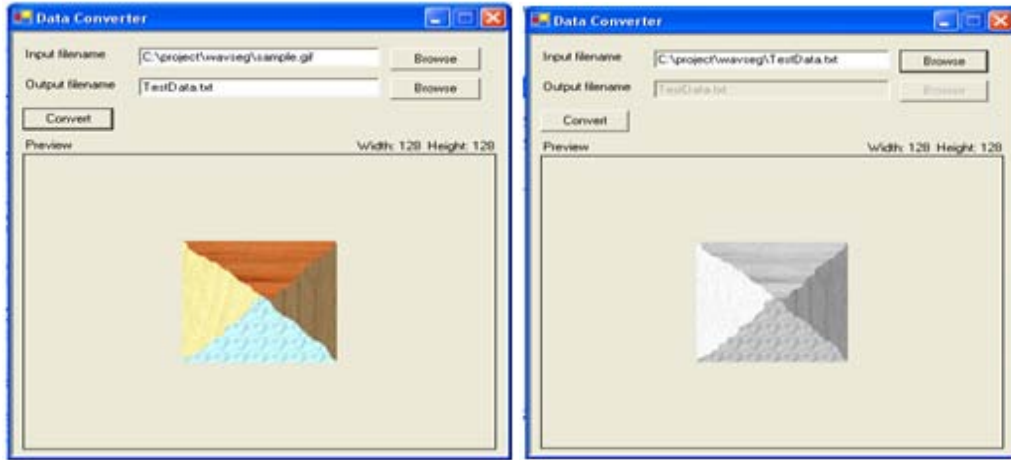


Fig. 12: Input image convert to TestData.txt

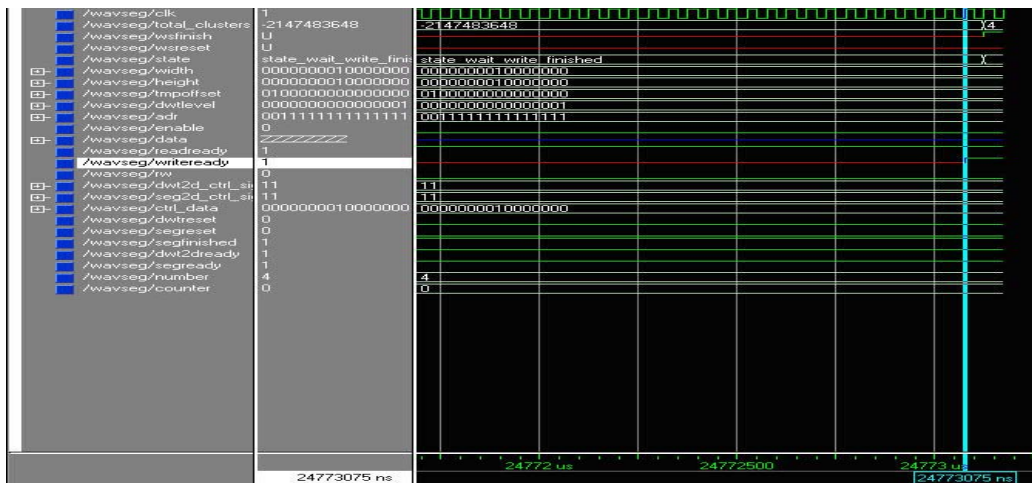


Fig. 13: STATE\_WAIT\_WRITE\_FINISHED phase

In Fig. 12, the left hand side of original input image was fed into DataConverter.exe and produced 8-level gray color data in TestData.txt. The converted image is then fed into the WavSeg module for the segmentation process. In Fig. 13, writeready signal was set high at 24.773075 ms to indicate the result was successfully written to output text file (SegResult.txt). The result for number of segmentation was then sent to total\_clusters pin at WavSeg. Once the total\_clusters get the four as result, the simulation was stopped by command. Finally, the segmentation result was observed using Data Converter.txt. As shown in Fig 14, the left hand side was the input file of the image where the right hand side was the segmentation result. From the segmentation result, four different types of textures were represented by four different types of colors.

**Comparison of result:** The 10 sample images were selected to run the segmentation process and the data were recorded. From the result, the time executions were quit consistent and independent from the number of textures which is shown in Fig. 15. In other words, the total execution time is not fully depends on the number of textures and it might vary due to different textures orientation or contrast between two textures. The overall time elapse for this project was between 0.025-0.033 sec which can be considered as fast. Guan *et al.* (2011) proposed K-mean+ as the solution to unsupervised fuzzy k-mean clustering algorithm and their findings were included in Fig. 16 and 17. By observing Fig. 16, the initial numbers of clusters yield a constant execution time which is around 345 sec. In Fig. 17, the execution time increased exponentially when numbers of data were increased. For

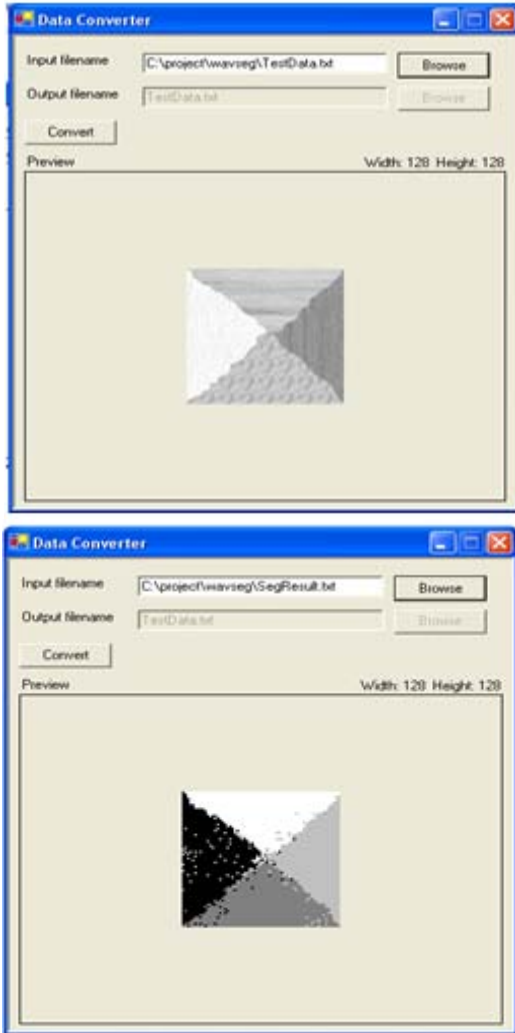


Fig. 14: Post process phase

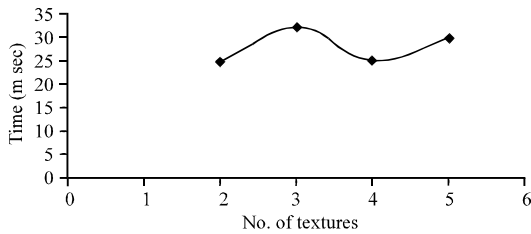


Fig. 15: Time elapse for different number of textures

these two cases, Dell Dimension 2300 with a Celeron CUP 1.80 GHz and a RAM of 256 MB was used as testing platform.

In this research, the initial number of clusters is 16 and the target input image size is  $128 \times 128$  pixels. From the Fig. 15, the time elapse for K-mean+ with 16 as the initial number of clusters is about 345 sec which is much higher as compare to this project. In addition, input image

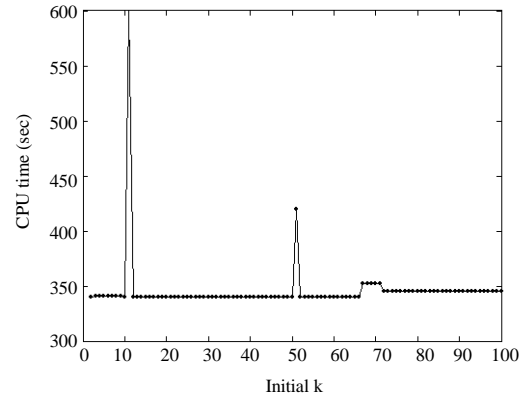


Fig. 16: CPU time vs. Initial number of cluster (Guan *et al.*, 2011)

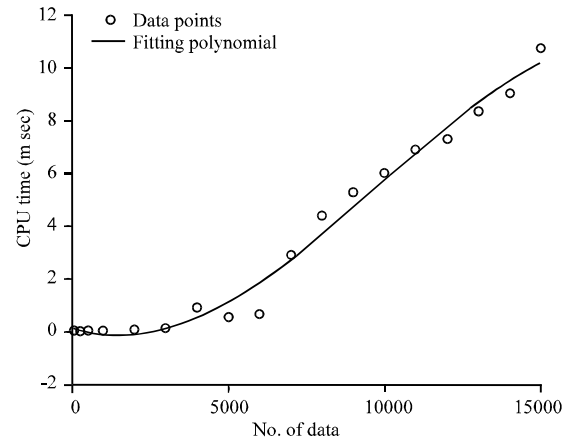


Fig. 17: CPU time vs. the number of data (Guan *et al.*, 2011)

size with  $128 \times 128$  or 16384 data can be executed between 0.025 and 0.033 sec for this project where k-mean+ algorithm implemented in Java used  $>1000$  sec to run the segmentation as shown in Fig. 16.

## CONCLUSION

In this research, fuzzy K-mean clustering algorithm was chosen as segmentation algorithm based on the justification of the study. To reduce the execution time, the segmentation preprocess was needed to extract the input image to enhance the performance. Le Gall 5/3 filter as DWT implementation was used for this project to decompose the input image into smaller size. In order to perform unsupervised segmentation on supervised fuzzy K-mean algorithm, some modifications were done and implemented successfully in VHDL. From the simulation result, this research was able to detect the texture image from two types of texture up to five types. All these segmentations done within 0.025-0.033 sec for  $128 \times 128$  pixels which can be considered as fast segmentation solution.



## REFERENCES

- Akter, M., M.B.I. Reaz, F. Mohd-Yasin and F. Choong, 2008a. A modified-set partitioning in hierarchical trees algorithm for real-time image compression. *J. Commun. Technol. Elect.*, 53: 642-650.
- Akter, M., M.B.I. Reaz, F. Mohd-Yasin and F. Choong, 2008b. Hardware implementations of image compressor for mobile communications. *J. Commun. Technol. Electron.*, 53: 899-910.
- Coussy, P., D.D. Gajski, M. Meredith and A. Takach, 2009. An Introduction to high-level synthesis. *IEEE Design Test Comput.*, 26: 8-17.
- Cruz, D.S., T. Ebrahimi, J. Askelof, M. Larsson and C. Christopoulos, 2000. An analytical study of JPEG2000 functionalities. *Proc. Int. Conf. Image Process.*, 2: 49-52.
- Descampe, A., F.O. Devaux, G. Rouvroy, J.D. Legat, J.J. Quisquater and B. Macq, 2006. A flexible hardware jpeg 2000 decoder for digital cinema. *IEEE Trans. Circuits Syst. Video Technol.*, 16: 1397-1410.
- Fauzi, M.F.A. and P.H. Lewis, 2003. A fully unsupervised texture segmentation algorithm. *Proceedings of the British Machine Vision Conference*, September 9-11, 2003, British Machine Vision Association, pp: 519-528.
- Guan, Y., A.A. Ghorbani and N. Belacel, 2011. K-mean+: An autonomous clustering algorithm. National Research Council of Canada.
- Hussain, M.S., M.B.I. Reaz, F. Mohd-Yasin and M.I. Ibrahimy, 2009. Electromyography signal analysis using wavelet transform and higher order statistics to determine muscle contraction. *Expert Syst.*, 26: 35-48.
- Isa, N.A.M., S.A. Salamah and U.K. Ngah, 2009. Adaptive fuzzy moving K-means clustering algorithm for image segmentation. *IEEE Trans. Consumer Elect.*, 55: 2145-2153.
- Jung, H.Y. and R. Prost, 1998. Lossless subband coding system based on rounding transform. *IEEE Trans. Signal Process.*, 46: 2335-2540.
- Kaganami, H.G. and Z. Beiji, 2009. Region-based segmentation versus edge detection. *Proceedings of the 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, September 12-14, 2009, Institute of Electrical and Electronics Engineers, USA., pp: 1217-1221.
- Kuzmanov, G., B. Zafarifar, P. Shrestha and S. Vassiliadis, 2002. Reconfigurable DWT unit based on lifting. *Proceedings of the 12th Annual Workshop on Circuits, Systems and Signal Processing*, November 2002, The Netherlands, pp: 325-333.
- Lezoray, O and H. Cardot, 2002. Hybrid color image segmentation using 2D Histogram clustering and region merging. *Int. Conf. Image Signal Process.*, 1: 22-29.
- Lezoray, O. and H. Cardot, 2002. Histogram and watershed based segmentation of color images. *Proceedings of the 1st European Conference on Color in Graphics, Imaging and Vision*, Poitiers, France, April 2002, The Society for Imaging Science and Technology, pp: 358-362.
- Liu, H. and K.C. Jezek, 2004. Automated extraction of coastline from satellite imagery by integrating Canny Edge Detection and locally adaptive thresholding methods. *Int. J. Remote Sensing*, 25: 937-958.
- Mohd-Yasin, F., A.L. Tan and M.I. Reaz, 2004. The FPGA prototyping of iris recognition for biometric identification employing neural network. *Proceedings of the 16th International Conference on Microelectronics*, December 6-8, 2004, Tunis, Tunisia, pp: 458-461.
- Qin, A.K. and D.A. Clausi, 2010. Multivariate image segmentation using semantic region growing with adaptive edge penalty. *IEEE Trans. Image Process.*, 19: 2157-2170.
- Reaz, M.B.I., F. Choong, M.S. Sulaiman and F. Mohd-Yasin, 2007. Prototyping of wavelet transform, artificial neural network and fuzzy logic for power quality disturbance classifier. *Elect. Power Compon Syst.*, 35: 1-17.
- Russ, J.C., P.H. Matthew and W.D. Michael, 2009. Erosion, dilation, opening and closing. *Optical Microscopy Primer*.
- Salari, E. and Z. Ling, 1995. Texture segmentation using hierarchical wavelet decomposition. *Pattern Recog.*, 28: 1819-1824.
- Serra, J., 2007. Mathematical morphology and image segmentation. *Proceedings of the 6th Annual Conference of the Network GEONET*, June 10-17, 2007, Benin, pp: 1-30.