Real-time automated image segmentation technique for cerebral aneurysm on reconfigurable system-on-chip

Xiaojun Zhai[a,*], Mohammad Eslami[b], Ealaf Sayed Hussein[b], Maroua Salem Filali[b], Salma Tarek Shalaby[b], Abbes Amira[b], Faycal Bensaali[c], Sarada Dakua[d], Julien Abinahed[d], Abdulla Al-Ansari[d], Ayman Z. Ahmed[e]

[a]*Department of Electronics, Computing and Mathematics, University of Derby, Derby, UK*
[b]*Department of Computer Sicence and Engineering, Qatar University, Qatar*
[c]*Department of Electrical Engineering, Qatar University, Qatar*
[d]*Department of Surgery, Hamad Medical Corporation, Qatar*
[e]*Department of Neuro-Radiology, Hamad General Hospital, Qatar*

Abstract

Cerebral aneurysm is a weakness in a blood vessel that may enlarge and bleed into the surrounding area, which is a life-threatening condition. Therefore, early and accurate diagnosis of aneurysm is highly required to help doctors to decide the right treatment. This work aims to implement a real-time automated segmentation technique for cerebral aneurysm on the Zynq system-on-chip (SoC), and virtualise the results on a 3D plane, utilizing virtual reality (VR) facilities, such as Oculus Rift, to create an interactive environment for training purposes. The segmentation algorithm is designed based on hard thresholding and Haar wavelet transformation. The system is tested on six subjects, for each consists 512×512 DICOM slices, of 16 bits 3D rotational angiography. The quantitative and subjective evaluation show that the segmented masks and 3D generated volumes have admitted results. In addition, the hardware implement results show that the proposed implementation is capable to process an image using Zynq SoC in an average time of 5.2 ms.

Keywords: Cerebral aneurysm, Image segmentation, Zynq SoC, FPGA.

[*] Corresponding author
Email address: x.zhai@derby.ac.uk (Xiaojun Zhai)

1. Introduction

An aneurysm is a swelling on the side of a blood vessel wall and it can burst and lead to bleeding. It looks like a thin balloon or weak spot on an inner tube. The aneurysm in the brain is called cerebral aneurysm [1]. Statistics show that between 1.5% and 5% of the population have or will develop cerebral aneurysm [2]. Annually, almost from 0.5% to 3% of patients with a brain aneurysm may suffer from bleeding [1].

When an aneurysm ruptures, blood leaks into the subarachnoid space called subarachnoid hemorrhage (SAH). SAH represents one of the most prevalent and devastating diseases among adults worldwide. Endovascular approaches to treatment of intracranial aneurysms (ICAs) are more effective than other methods in terms of reducing operative risk, hospital stay, pains and indirectly cost [3]. These approaches, which are centered around the use of intra-aneurysmal coils, may sometimes fail because of incomplete occlusion of the defect, which could be due to the miscalculation of the aneurysm anatomy. Therefore, appropriate segmentation of cerebral aneurysm is always desired for an effective treatment planning (i.e. deciding the right size and type of the first coil) [4].

Since the result after applying the automated aneurysm segmentation algorithm is a 3D volume that contains features from the aneurysm, this helps doctors in diagnosing and deciding the right treatment, because knowing some parameter about the aneurysm is crucial for such purposes. Manual segmentation is typically utilized to get these parameters, nevertheless, it is not precise because it heavily depends on inter-observer variability. Hence, employing an automatic segmentation technique will be more accurate and reliable.

While image segmentation is a general field and has many applications, angiography also includes a wide range of anatomical applications (e.g. cerebral, retinal, hepatic, peripheral, pulmonary, cardiac, etc.) and modalities (e.g. X-ray, computed tomography angiography (CTA), Magnetic resonance angiography (MRA), ultrasound, 2D or 3D, etc.) [5, 6, 7]. Therefore, there are many methods in literature for vascular segmentation. The segmentation algorithms of Cerebrovascular MRA medical images can be categorized into the following four methods. The deformation model, the statistical model, the Hessian matrix and the region growing [8, 9]. Well known deformation models are parametric and geometric in which usually exploits several internal and external forces to find connected curves in the images [10, 11]. Statistical algorithms aim to fit a statistical model to the distribution of the intensities of the images and compute the parameters of model. Usually, they assume that brain MRA images consists of three different regions with different intensity margins and therefore try to determine these regions on the intensity histogram. The first region has the lowest intensities and consists of cerebrospinal fluid, cerebral bones, air and other organizations. The second region with medium intensity has cerebral gray and white matter. The third region with higher intensity includes cerebral vessels and subcutaneous fat [9, 11]. Cerebral vessels are worm and tube like and therefore have typical characteristics of tubes. Some of these characteristics are the eigenvalues and eigenvectors of the Hessian matrix. These method mostly used for vessel enhancement as a preprocessing step for segmentation [12, 13]. Alternatively, optimally oriented flux (OOF) was introduced by Law and Chung to overcome the shortcomings of Hessian-based filters. OOF and its anisotropic variations have gained attention for the segmentation of different anatomical structures including vessels [14, 15]. The region growing methods are a conventional algorithms for angiography and other applications of segmentation. Typically, there is a good connectivity and a complete topological structure for resulting vessel voxels [16, 17].

There are few other methods have been designed for the vascular aneurysm segmentation problem. Wilson et. al proposed a fully automatic, statistically based algorithm for segmenting the three-dimensional vessel information from time of flight (TOF) MRA data [18]. They introduced a mixture distribution for the data, motivated by a physical model of blood flow, that is used in a two stage segmentation algorithm. In the first stage they use a variant of the traditional EM algorithm to classify vessel voxels, on the assumption that all voxels are independent. Based on this initial segmentation, they then estimate the two thresholds to perform hysteresis thresholding. An algorithm based on a geometric deformable model with energy functional along with a non-parametric statistical framework which exploits high-order multiscale features is presented in [19]. The method is based on a geometric deformable model that uses also information from the image gradient and statistics of the different tissue regions. Cross-validation and feature selection techniques are used to determine the non-parametric statistical model and fit the model to the specific application and achieve the best tissue classification. In [20] an evaluation study is reported to evaluate the suitability of their automatic segmentation method based on geodesic active regions (GAR) for segmenting cerebral vasculature with aneurysms. Three aspects of the GAR method have been improved: execution time, robustness to variability in imaging protocols and robustness to variability in image spatial resolutions. They evaluate their method on 3D X-ray rotational angiography (3DRA) and time of flight magnetic resonance angiography (TOF-MRA) images. Similarly, the work presented in [21] introduce a new cerebral aneurysm segmentation approach, which is based on geodesic active contours (GAC). In this method, the wall of the aneurysm in 3D has been considered as the zero level set, and the convergence of the embedding function is used to define the surface of the extracted aneurysm. The results show that the prior de-noising shows slight improvement in the segmentation results, but the algorithm needs to initiate a seed point manually where the segmentation starts.

In [22], a new method has been proposed for CTA, where the segmentation is based on region growing and level set approaches. In the first stage, CTA scans are smoothed with the use of a median filter, and then, the region growing-based approach is used to segment the area of interest. Finally, the selection criteria of the connectivity of the points is applied to recognize the artery range. The proposed method demonstrates good results, however, it still needs the user to initiate a seed point to guide where the segmentation should be started. Authors in [23] proposes a threshold-based level segmentation (TLS) method for segmenting the cerebral aneurysm. The approach uses the Geodesic active contours and Chan-vese segmentation model. The proposed method combines the region and boundary information to decide the global threshold and gradient magnitude to be used in the segmentation. The threshold keeps updating through the segmentation process until the boundary of the aneurysm is reached. The TLS allowed promising results and more accurate results than other methods mentioned previously, and it does not require an initial seed point or intensity threshold. the segmentation of brain vascular from low contrast MRA is presented in [4], where the segmentation algorithm is based on principle component analysis (PCA). The PCA is used to filter the unwanted elements from the image and keep the details of the variation of the width of the vessels. However, since there is no prior noise filter used in the algorithm, the achieved results are not significant and de-noising has been suggested to improve the segmentation results further.

Field programmable gate array (FPGA) has been widely used to accelerate the image processing algorithms in biomedical imaging area. Although there are some implementations of image segmentation targeted on FPGAs, there is no FPGA implementation for automated cerebral aneurysm segmentation [24, 25, 26, 27]. In this paper, we design and implement an automated aneurysm segmentation algorithm on Zynq SoC. The segmented results are visualised on a 3D plane using virtual reality (VR) facilities to create an interactive environment for training purposes. The aneurysm segmentation approach is first implemented and simulated in MATLAB as a proof of concept, and then the appropriate C/C++ codes of the algorithm are written and implemented on hardware using Vivado HLS. The system is tested on six subjects, each subject consists of 512 × 512 16 Bit DICOM slices of Computed tomography angiography while the total number of images is 451. The main contributions of this paper can be summarized as follows:

- A novel SoC solution for real-time automated segmentation technique is introduced. In addition, a hardware friendly aneurysm segmentation algorithm has been proposed for hardware implementation. Finally, the segmented results are visualised on a 3D plane using virtual reality (VR) facilities to create an interactive environment for effective treatment planning and training purpose.

- The proposed approach introduces a way to integrate the aneurysm segmentation and processing unit into heterogeneous reconfigurable hardware. This allows the implementation of a high-performance state-of-the-art data processing system which is also highly adaptive. The communication, visualization, segmentation and flow simulation can be realized on one piece of hardware without making the compromise of resource sharing and time-consuming sequential execution of tasks.

The rest of the paper is organized as follows. Section 2 introduces the aneurysm segmentation algorithm. The corresponding software and hardware implementations are presented in Section 3. The experiment results are discussed in Section 4. Finally, Section 5 concludes the paper and highlights some perspectives of future work.

2. Proposed Method

The proposed aneurysm segmentation algorithm is based on wavelet transform [28] and hard thresholding [29] algorithms. The overall diagram of the proposed method is illustrated in Figure 1 and consists of the following six steps, described later with more details.

1. Intensity normalization: The intensity propagation of DICOM images is normalized to $[0 - 255]$ for each subject.
2. Haar wavelet decomposition: The Haar wavelet transform scheme is applied on each normalized slice image.
3. Hard thresholding: After applying the wavelet transform, hard thresholding is applied on the approximated coefficients.
4. Haar wavelet reconstruction: In this step, each slice is reconstructed by wavelet detail and thresholded approximation coefficients.

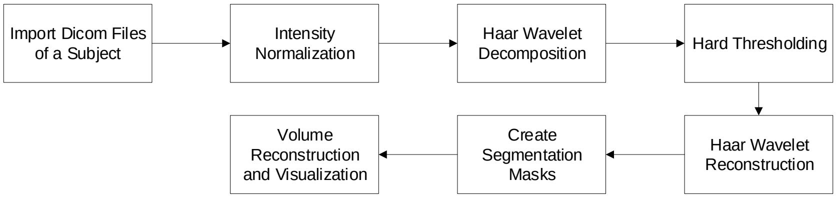

Import Dicom Files of a Subject
Intensity Normalization
Haar Wavelet Decomposition
Hard Thresholding
Haar Wavelet Reconstruction
Create Segmentation Masks
Volume Reconstruction and Visualization

Figure 1: The chart of proposed method for segmentation and aneurysm treatment.

5. Creating segmentation masks: In this step a threshold is used to convert the reconstructed slice to binary segmented mask. After that, a dilation operation may be implemented based on a proposed criterion.
6. Volume reconstruction and visualization: In this step, the segmented masks of slices are concatenated and processed to build the volume of vessels.

2.1. Intensity normalization

At the first step, since the range of intensities in DICOM images is variant for different subjects and different imaging devices, the intensity normalization is required. In this step, all of the DICOM images of a subject are imported and the maximum/minimum intensity of the subject will be normalized to 0/255 respectively. Suppose that the size of each DICOM slice ($S_i, i = 1, 2, \cdots, I$) is $N \times N$ and the subject has I images. Therefore we have all the images in a matrix $\boldsymbol{S} \in R^{N \times N \times I}$ and the minimum and maximum intensities of $\boldsymbol{S}$ are min_S and max_S. The Normalized version of $\boldsymbol{S}$ which is denoted as $\boldsymbol{X}$ can be computed by equation (1) where $\boldsymbol{X} \in R^{N \times N \times I}$ contains normalized slices, X_i for $i = 1, 2, \cdots, I$.

$$\boldsymbol{X} = 255 \times \frac{\boldsymbol{S} - min_S}{max_S - min_S} \tag{1}$$

2.2. Haar wavelet decomposition

The Haar wavelet transform is used to decompose the image into an approximated image as well as three detailed sub-band images, where the approximation image shows an approximated overview of the original image. The approximated image (LL) is produced as a result of applying a low pass filter on the rows followed by a low pass filter on the columns. The vertical detail (LH) results from applying low pass filter on the rows followed by a high pass filter on the columns. A high pass filter on the rows is followed by a low pass filter on the columns to produce the horizontal detail. Finally, a high pass filter is applied on the rows followed by a high pass filter on the columns producing the diagonal detail (HH). The results of applying wavelet transform on an original testing image is shown in Figure 2 in which a blood vessel and aneurysm is present.

For this paper, two different approaches of implementing the Haar wavelet were tested, using filter banks [28] and using running average/differencing [30]. Based on the testing results, the averaging and differencing techniques were chosen for their hardware implementation suitability to avoid memory issues.

Haar wavelet using averaging and differencing:

There are two steps in this approach [30], 1) apply running average and differencing on all the rows. 2) apply running average and differencing on all the columns. Lets consider a one dimensional example. Suppose that A_m and D_m denote the running average and running difference for $f = (f_1, f_2, f_3, f_4, \cdots, f_N)$ where $m = 1, 2, 3, \cdots, N/2$. The running average and difference can be computed by equations (2) and (3) respectively.

$$A_m = (\frac{f_{2m-1} + f_{2m}}{2}) \times \sqrt{2} = \frac{f_{2m-1} + f_{2m}}{\sqrt{2}} \tag{2}$$

$$D_m = (\frac{f_{2m-1} - f_{2m}}{2}) \times \sqrt{2} = \frac{f_{2m-1} - f_{2m}}{\sqrt{2}} \tag{3}$$

For example, Figures 3 and 4 show the process of applying the averaging and differencing on the rows and columns of a 8 × 8 image. Used notations in these Figures are as follow. $f(i, j)$ is the intensity of the element in i^{th} row and j^{th} column. Suppose that each T_{ij} is the computed element of type T in the i^{th} row and j^{th} column of result matrix. e.g. XA_{ij} is the approximated coefficient in i^{th} row and j^{th} column.

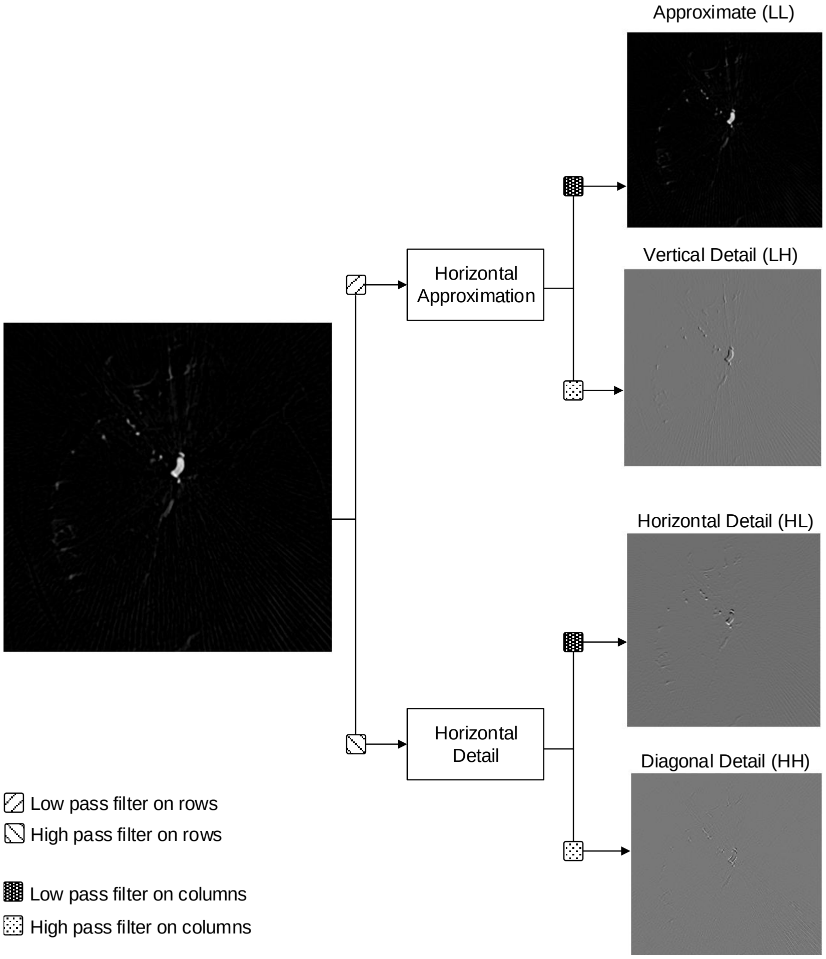
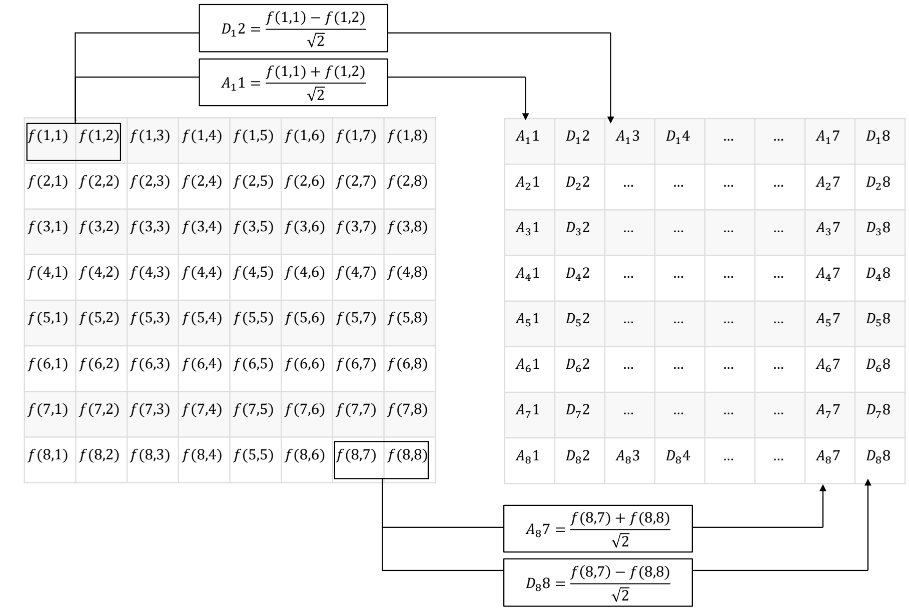


Figure 2: The results of applying wavelet transform on an original image.

2.3. Hard thresholding

In this step, hard thresholding is applied on the approximated coefficients. The hard thresholding was chosen as it does not affect the remained values, unlike the soft thresholding that either kills the values or shrink it based on the threshold.

Generally, the hard thresholding either keeps the value XA_{ij} which has an absolute value greater than or equal the threshold (T), or it sets values lower



$$D_1 2 = \frac{f(1,1) - f(1,2)}{\sqrt{2}}$$

$$A_1 1 = \frac{f(1,1) + f(1,2)}{\sqrt{2}}$$

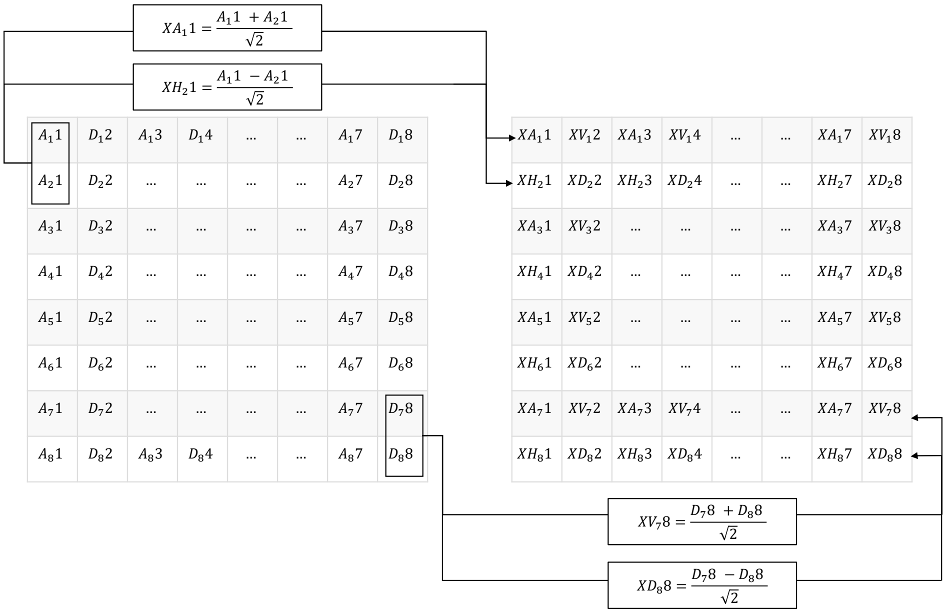
$f(1,1)$	$f(1,2)$	$f(1,3)$	$f(1,4)$	$f(1,5)$	$f(1,6)$	$f(1,7)$	$f(1,8)$
$f(2,1)$	$f(2,2)$	$f(2,3)$	$f(2,4)$	$f(2,5)$	$f(2,6)$	$f(2,7)$	$f(2,8)$
$f(3,1)$	$f(3,2)$	$f(3,3)$	$f(3,4)$	$f(3,5)$	$f(3,6)$	$f(3,7)$	$f(3,8)$
$f(4,1)$	$f(4,2)$	$f(4,3)$	$f(4,4)$	$f(4,5)$	$f(4,6)$	$f(4,7)$	$f(4,8)$
$f(5,1)$	$f(5,2)$	$f(5,3)$	$f(5,4)$	$f(5,5)$	$f(5,6)$	$f(5,7)$	$f(5,8)$
$f(6,1)$	$f(6,2)$	$f(6,3)$	$f(6,4)$	$f(6,5)$	$f(6,6)$	$f(6,7)$	$f(6,8)$
$f(7,1)$	$f(7,2)$	$f(7,3)$	$f(7,4)$	$f(7,5)$	$f(7,6)$	$f(7,7)$	$f(7,8)$
$f(8,1)$	$f(8,2)$	$f(8,3)$	$f(8,4)$	$f(5,5)$	$f(8,6)$	$f(8,7)$	$f(8,8)$

$A_1 1$	$D_1 2$	$A_1 3$	$D_1 4$	...	...	$A_1 7$	$D_1 8$
$A_2 1$	$D_2 2$	...	...	...	...	$A_2 7$	$D_2 8$
$A_3 1$	$D_3 2$	...	...	...	...	$A_3 7$	$D_3 8$
$A_4 1$	$D_4 2$	...	...	...	...	$A_4 7$	$D_4 8$
$A_5 1$	$D_5 2$	...	...	...	...	$A_5 7$	$D_5 8$
$A_6 1$	$D_6 2$	...	...	...	...	$A_6 7$	$D_6 8$
$A_7 1$	$D_7 2$	...	...	...	...	$A_7 7$	$D_7 8$
$A_8 1$	$D_8 2$	$A_8 3$	$D_8 4$	...	...	$A_8 7$	$D_8 8$

$$A_8 7 = \frac{f(8,7) + f(8,8)}{\sqrt{2}}$$

$$D_8 8 = \frac{f(8,7) - f(8,8)}{\sqrt{2}}$$

Figure 3: Applying averaging and differencing on the rows.



$$XA_1 1 = \frac{A_1 1 \ + A_2 1}{\sqrt{2}}$$

$$XH_2 1 = \frac{A_1 1 \ - A_2 1}{\sqrt{2}}$$

$A_1 1$	$D_1 2$	$A_1 3$	$D_1 4$	...	...	$A_1 7$	$D_1 8$
$A_2 1$	$D_2 2$	...	...	...	...	$A_2 7$	$D_2 8$
$A_3 1$	$D_3 2$	...	...	...	...	$A_3 7$	$D_3 8$
$A_4 1$	$D_4 2$	...	...	...	...	$A_4 7$	$D_4 8$
$A_5 1$	$D_5 2$	...	...	...	...	$A_5 7$	$D_5 8$
$A_6 1$	$D_6 2$	...	...	...	...	$A_6 7$	$D_6 8$
$A_7 1$	$D_7 2$	...	...	...	...	$A_7 7$	$D_7 8$
$A_8 1$	$D_8 2$	$A_8 3$	$D_8 4$	...	...	$A_8 7$	$D_8 8$

$XA_1 1$	$XV_1 2$	$XA_1 3$	$XV_1 4$	...	...	$XA_1 7$	$XV_1 8$
$XH_2 1$	$XD_2 2$	$XH_2 3$	$XD_2 4$	...	...	$XH_2 7$	$XD_2 8$
$XA_3 1$	$XV_3 2$	...	...	...	...	$XA_3 7$	$XV_3 8$
$XH_4 1$	$XD_4 2$	...	...	...	...	$XH_4 7$	$XD_4 8$
$XA_5 1$	$XV_5 2$	...	...	...	...	$XA_5 7$	$XV_5 8$
$XH_6 1$	$XD_6 2$	...	...	...	...	$XH_6 7$	$XD_6 8$
$XA_7 1$	$XV_7 2$	$XA_7 3$	$XV_7 4$	...	...	$XA_7 7$	$XV_7 8$
$XH_8 1$	$XD_8 2$	$XH_8 3$	$XD_8 4$	...	...	$XH_8 7$	$XD_8 8$

$$XV_7 8 = \frac{D_7 8 \ + D_8 8}{\sqrt{2}}$$

$$XD_8 8 = \frac{D_7 8 \ - D_8 8}{\sqrt{2}}$$

Figure 4: Applying averaging and differencing on the columns.

than the threshold to zero. Equation (4) illustrates this calculation:

$$(XA_{ij})_t = \begin{cases} XA_{ij} & |XA_{ij}| \geq T \\ 0 & |XA_{ij}| < T \end{cases} \tag{4}$$

The threshold value is chosen based on following facts. The fact is that, since the intensities are normalized in range $[0-255]$ and the blood segments have the highest intensity values in angiography, the value of interest in the normalized images will be $[128 - 255]$. On the other side, by considering running average equation, it can be inferred that each approximation coefficient (XA) is related to the value of interest. For example, equation (5) shows the relations of the first approximation coefficient (XA_{11}). Therefore, it has been decided to use the threshold value T as 240 in which $240 \approx 2 \times 128 - \epsilon$ and this value brings the appropriate results.

$$\begin{aligned} XA_{11} &= \frac{A_{11} + A_{21}}{\sqrt{2}} = \frac{\frac{f(1,1)+f(1,2)}{\sqrt{2}} + \frac{f(2,1)+f(2,2)}{\sqrt{2}}}{\sqrt{2}} \\ &= \frac{f(1,1) + f(1,2) + f(2,1) + f(2,2)}{2} \propto 2 \times f(.,.) \end{aligned} \tag{5}$$

2.4. Haar wavelet reconstruction

In this step, wavelet reconstruction is performed where the result of the thresholding step is used with the other detailed sub-bands. For wavelet reconstruction, generally the four decomposed images are used to reconstruct the original image. First, a low pass filter is applied on columns of the approximated image (LL) then its rows are passed through another low pass filter. The columns of the vertical detailed image (LH) pass through a high pass filter then its rows pass through a low pass filter. A low pass filter is applied on columns of the horizontal detailed image (HL) then its rows pass through a high pass filter. Finally, both the rows and the columns of the detailed diagonal image (HH) pass through a high pass filter. Haar wavelet reconstruction can be implemented simply by averaging and differencing too.

Inverse Haar wavelet using averaging and differencing:

suppose that the outputs of this step are X_i' for $i = 1, 2, \cdots, I$. This process is similar to the processes in the Haar transform, it starts by the columns then the rows, where the running average and differencing processes are firstly applied to each column and then each row respectively [30]. Pseudocode 2 describes the method with more details and step by step.

2.5. Creating segmentation masks

In this step, the binary mask of segmented object in each slice is created. For this reason another threshold, $T_2 = T/3$ is selected and applied on the reconstructed slices (X_i's) to generate binary segmentation masks (B_is) using (6).

$$B_i(k,j) = \begin{cases} 255 & |X_i'(k,j)| \geq T/3 \\ 0 & |X_i'(k,j)| < T/3 \end{cases} \tag{6}$$

In addition, since the proposed scheme is based on the thresholding on the approximate coefficients which are intrinsically related to the sum of intensities in a neighborhood, it is expected that the border of the objects can be affected in the reconstructed image.

In theory, dilation can be implemented either on grayscale reconstructed image (X_i') or on binary mask version (B_i). Both of them have been evaluated and it was found that there is no significant difference. Therefore, it was decided to dilate the object in the binary mask slices, B_is. The structure element is a 2×2 square and the mask will be dilated only if the maximum intensity value of the original normalized slice ($X_i \in \boldsymbol{X}$) is under the threshold T as shown in equation (7).

$$B_i' = \begin{cases} B_i & max(X_i) \geq T \\ Dilate\ B_i & max(X_i) < T \end{cases} \tag{7}$$

2.6. Volume reconstruction and visualization

When all the slices of a subject are segmented, the segmented slices should be concatenated and construct the volume of vessel. Suppose that $\boldsymbol{B'} \in R^{N \times N \times I}$ includes all of the B_i's for $i = 1, 2, \cdots, I$ where B_i's are the segmented slices

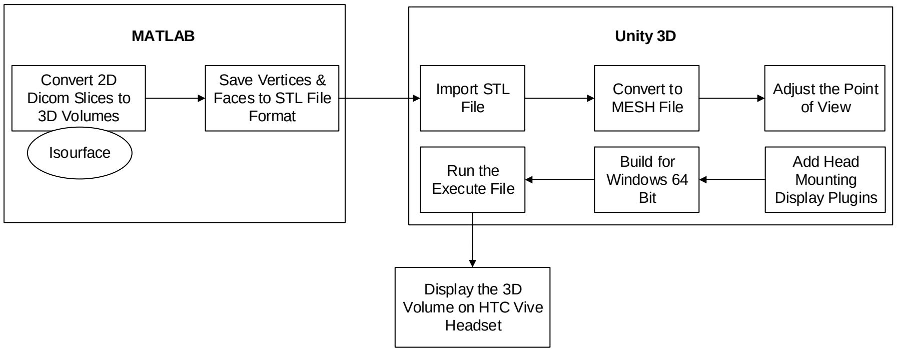

MATLAB
Convert 2D Dicom Slices to 3D Volumes
Isourface
Save Vertices & Faces to STL File Format
Unity 3D
Import STL File
Convert to MESH File
Adjust the Point of View
Run the Execute File
Build for Windows 64 Bit
Add Head Mounting Display Plugins
Display the 3D Volume on HTC Vive Headset

Figure 5: Design flow of the virtual reality scheme.

(binary masks, the output of step 5). First, the smoothed matrix is produced by using a 3D smoothing operation on $\boldsymbol{B}'$. The type and size of the convolution kernel is Box and $3 \times 3 \times 3$, respectively. Next the isosurfaces of the smoothed matrix are generated by using isovalue $IV = 200$. The isosurfaces are finally displayed.

Alternatively, in order to display 3D vessels on a virtual reality headset, the generated 3D volume of vessel are converted to vertices and faces and saved in STL file format. The STL files used by unity 3D to display interactively on a VR headset. The HTC Vive is used to visualize the results of the aneurysm segmentation, and the user can interact with the visualization via zooming and rotation. The diagram of the VR design is shown in Figure 5.

3. Hardware Implementation

The Zedboard is used for the hardware implementation of the proposed segmentation algorithm. The board is equipped with a Zynq SoC, which contains two subsystems: programmable logic (PL) and processing systems (PS) [31]. An SD card was used to store DICOM images and the aneurysm segmentation is performed in Zynq SoC. The segmented results are displayed on the monitor via HDMI interface. Alternatively, the segmentation results are sent to the PC and the generated 3D volumes of extracted vessels are displayed on a VR

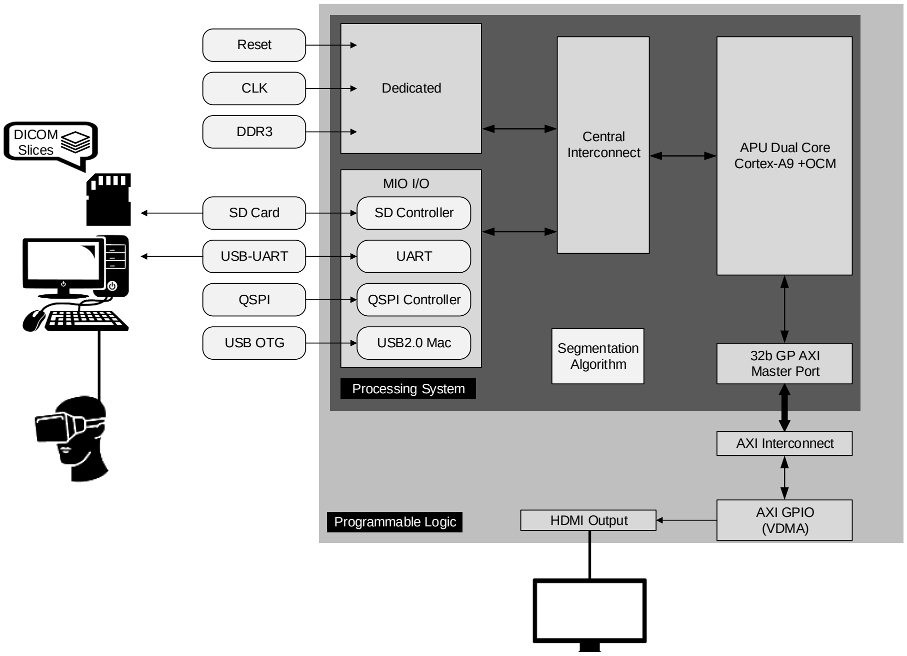


Figure 6: Architecture of the overall system.

headset. Figure 6 illustrates the architecture of the overall hardware system.

3.1. Hardware implementation of segmentation algorithm

The proposed segmentation algorithm is implemented in C++ using Vivado HLS, and then synthesised and translated to a hardware description language (HDL). A set of pragma directives are used to optimise the codes for hardware implementation, where the overall goal of the optimisation is to achieve the high throughput architecture with minima usage of hardware recourses. As mentioned before, in order to minimise memory usage of implementing the segmentation algorithm, the averaging and differencing approach is chosen for the hardware implementation. The following pseudocode 1 and 2 show the entire process of the Haar decomposition, hard thresholding and reconstruction.

In order to reduce the dimensions of the input array and to be compatible with the data bus, the input array is reshaped to one dimension array with size of $r \times p = q$, where r and l are number of rows and columns of the original input

Pseudocode 1: Haar transform and hard thresholding

```
Input: $X_i$ is the input array of an image.
Output: $X_{out}$ is the output array of the processed image.
for (row = 0; row <maximum number of rows; row++ ){
    for (col = 0; col <maximum number of columns; col = col + 2){
        $X_{out}$ [row][col] = ($X_i$[row][col] + $X_i$[row][col+1]) / $2^{1/2}$;
        $X_{out}$ [row][col + 1] = ($X_i$[row][col] - $X_i$[row][col+1]) / $2^{1/2}$;
    }
}
for (col = 0; col <maximum number of columns; col++ ){
    for ( row = 0; row <maximum number of rows; row = row + 2 ){
        m = $X_{out}$[row][col];
        n = $X_{out}$[row + 1][col];
        $X_{out}$ [row][col] = (m + n) / $2^{1/2}$ ;
        $X_{out}$ [row + 1][col] = (m - n) / $2^{1/2}$;
        if (row % 2 == 0 && col % 2 == 0 ){
            if(|$X_{out}$[row][col]| <= T)
            $X_{out}$[row][col] = 0;
        }
    }
}
```

Pseudocode 2: Haar inverse transform

```
Input: $X_{out}$ is the input array of a Haar Transformed image
Output: $X_i'$ is the output array of a Haar Inverse Transformed image
for (col = 0; col <maximum number of columns; col++ ){
    for (row = 0; row <maximum number of rows; row = row + 2 ){
        m = $X_{out}$ [row][col];
        n = $X_{out}$ [row + 1][col];
        $X_i'$ [row][col] = (m + n) / $2^{1/2}$;
        $X_i'$ [row + 1][col] = (m - n) /$2^{1/2}$;
    }
}
for (row = 0; row <maximum number of rows; row++ ){
    for (col = 0; col <maximum number of cols; col = col + 2 ){
        m = $X_i'$[row][col];
        n = $X_i'$[row][col + 1];
        $X_i'$ [row][col] = (m + n) / $2^{1/2}$ ;
        $X_i'$ [row][col + 1] = (m - n) / $2^{1/2}$;
    }
}
```

array.

A set of computation optimization pragmas has been used to guide high-level synthesis (HLS) compiler to fully utilize all the computational resources provided by the on-chip hardware in order to achieve higher overall performance.

Loop Pipelining: loop pipelining is the key optimization techniques in HLS to improve the throughput of the loop, where the execution of operations from different loop iterations are overlapped in an organized way. The inner loop will be unrolled where it is possible. The maximum throughput achieved is limited by resource constraints and data dependency in the loop.

Array Partition: array partition is one of the key optimization techniques in HLS to improve the bandwidth of memory. Since arrays are implemented as memory, and it only has a maximum of two data ports, which limit the throughput of a read/write (or load/store) within the pipeline. One of the way to improve the throughput is to split the array into multiple smaller arrays that utilize multiple memory elements. Therefore, the array $x_i[q]$ can be partitioned into f small arrays, where each array has size of q/f using *#pragma HLS ARRAY_PARTITION cyclic* $factor = f$. The element 0 is assigned to the first new array, element 1 to the second new array, element 2 is assigned to the third new array and then element 3 is assigned to the first new array again, until the $f-1$ element is assigned to the f^{th} new array. In other word, these array can be running in parallel within the pipeline, which could significantly improve the pipeline throughput.

The proposed hardware implementation of the aneurysm segmentation accelerator uses 32-bit floating point arithmetic, and the accelerator is implemented with Vivado HLS (v2016.3) [32]. C/RTL simulation is performed before exporting the RTL as a Vivados IP core. The RTL is exported as IP core to be synthesized and implemented in Vivado (v2016.3) using a Xilinx Zynq-7000 XC7Z020 all programmable SoC [33]. The aneurysm segmentation accelerator is connected via an AXI4 interface to the Accelerator Coherency Port (ACP) of the ARM CPU in the Zynq-7000 SoC device. The solution is then exported as an IP core connected with AXI4-Stream interface to the ACP on AP SoC PS.

The connection is made through a Direct Memory Access (DMA) core in the PL subsystem. SDSoC (v2016.3) is used to interface the AP SoC PL hardware, the peripheral, the DMA engine, an AXI timer as well as other data mover logics [34]. The SDSoC is also used to design the AP SoC PS software to manage the peripherals and loading the testing data from external SD card.

3.2. SDSoC platform

An SDSoC platform consists of a Vivado Design Suite hardware project and software libraries. Figure 7 illustrates the blocks of hardware design. The main block in this design is the ZYNQ Processing System (processing_system7_0) where there are three main interfaces through which the PS_7 core can access the PL side peripherals and vice versa. For this experiment, the interfaces used for interconnections are AXI_GP and AXI_HP and the processor also contains external memory and serial port (UART or USBUART). Moreover, there are two video related clocks for AXI4Stream based interconnect and IP cores and the HDMI output interface as well as another separate HDMI output interface clock to change the different video resolution without having to change the AXI4-Stream clock. Once the embedded processor design is created, an I2C controller is implemented with the following Xilinx IP core. The latter core allows the processor to configure the HDMI output hardware peripheral. After that the ZED HDMI display sub-modules (zed_hdmi_display) is added that contains minimal logic for the 16 bit video data sent to the HDMI output device.

4. Results and Analysis

In order to evaluate the proposed method, 512 × 512, 16-bit slices from six subjects, totally 451 images have been used.

The Vivado tool is used to complete the placement and routing of the proposed implementation on aneurysm segmentation. The resources utilization of the proposed implementations are reported in Table 1, where the non-optimized implementation contains architectures without using pipeline and array parti-

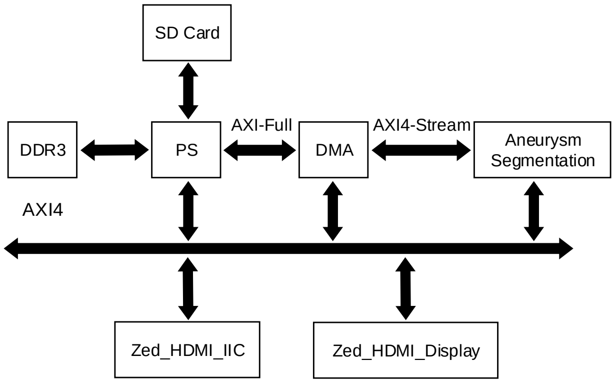


Figure 7: The hardware design and included blocks.

tion pragmas. In contrast, the optimized implementation uses both pipeline and array partition pragmas to reduce the latency of the design.

As it can be seen from Table 1, the hardware resource utilization of both non-optimized and optimized implementations is similar. The reason for that is that in order to reduce the memory usage, only the inner loops of Pseudocode 1 and 2 are unrolled, and pipelined. As result of this, there are no duplicated memory arrays are generated. Since the input and output arrays both contain large elements, they are implemented in the external memory and mapped to AXI bus addresses, therefore the on-chip architecture can access them via AXI data stream. In addition, the hardware utilization of both optimized implementations are also similar, only a slightly increase on the last implementation results, this is due to the extra the registers and instance are used for array partition.

Table 2 shows the different implementation results in terms of latency. As it can be seen from Table 2, the non-optimized implementation needs significant more clock cycles to complete the computations. Using pipeline pragmas in the implementation, the latency of the implementation has be dramatically reduced 917609 clock cycles, which is 5.6% of the latency in the non-optimized implementation. Finally, using both pipeline and array partition pragmas, the achieved the latency has further reduced to 524391 clock cycles, which further

Table 1: PL resource utilization of the proposed implementations.

Resources		DSP	BRAM	FF	LUT
Non-Optimized	Used	4	0	3439	5362
	Available	220	280	106400	53200
	Utilization (%)	1	0	3	10
Optimized (Pipeline)	Used	4	0	3568	5554
	Available	220	280	106400	53200
	Utilization (%)	1	0	3	10
Optimized (Pipeline & Array Partition)	Used	4	0	3615	5694
	Available	220	280	106400	53200
	Utilization (%)	1	0	3	10

Table 2: Performance estimates in terms of latency.

Performance Metrics	Latency (Clock Cycles)	Timing (ns)	FF	LUT
Non-Optimized	16257028	8.42	3439	5362
Optimized (Pipeline)	917609	8.42	3568	5554
Optimized (Pipeline + Array Position)	524391	8.31	3615	5694

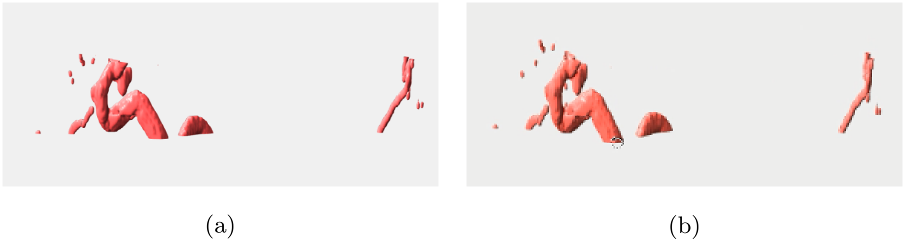

(a) (b)

Figure 8: A comparison between the segmented volumes resulting from MATLAB and Zedboard. (a) MATLAB 3D volume. (b) Zedboard 3D volume.

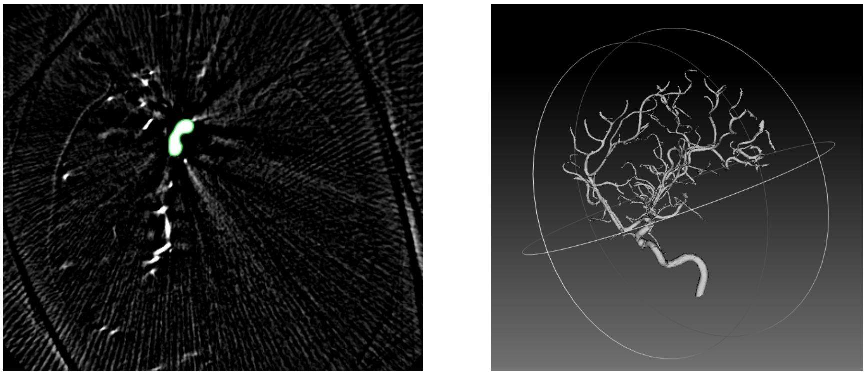
improves the latency by 42.9%. In addition, the timings of clock periods are also slightly improved 1.3%, which means that it can run up to 120.3 MHz. However, the required hardware resources are not significantly increased as reported in both Table 1 and Table 2. The processing speed for the hardware implementation is about 5.2 *ms*.

4.1. Testing

In Figure 8, the segmentation results from MATLAB and Zedboard are illustrated and compared. It can be seen that the results of implementing the algorithm on the Zedboard is same as the MATLAB on the PC. The runtime of the algorithm by MATLAB on PC is almost 9 seconds which is significant compared to mili-seconds of the Zedboad.

4.1.1. Quantitative segmentation performance

Since the brain aneurysm is a very patient specific study and the brain aneurysm data are not publicly available, the doctors have carefully chosen 451 images of six subjects of 3D rotational angiography (3DRA) from the Hamad medical corporation (HMC). The ground-truth for two subjects were determined by green contours where the region of interest is around aneurysm. The ground-truth for the other three subjects were STereoLithography (STL) files which contain the 3D shapes of brain vessels in standard triangle language. Figure 9 shows an example of the two types available ground-truths, green contour and STL file respectively.



(a) (b)

Figure 9: Ground truth examples. a) A Green contour segmented by surgeon. b) A 3D STL file showing the vessel in brain.

In order to access the performance of the proposed segmentation we use four different similarity metrics as follow. Dice similarity coefficient (DSC), Jaccard index, false positive rate (FPR) and false negative rate (FNR) [35]. Suppose that AS is the vessel area extracted by automatic segmentation and GT is the vessel area dedicated by manual segmentation in ground truth. DSC and Jaccard measure the similarity between AS and GT and range from 0 to 1, where 0 indicates no overlap between the results derived from the two areas and 1 corresponds to the best agreement between the two segmented areas. The Jaccard coefficient measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets. Jaccard and dice are similar in concept but Jaccard only considers the true positives. FPR and FNR measure the ratios of false positive ($AS \cap !GT$) and false negative ($!AS \cap GT$) with respect to the ground truth (GT). The corresponding definitions are stated in equation (8).

Table 3: Segmentation performance metrics for 5 different subjects.

Ground truth	STL 3D			Green Contour		
	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6
#	100	130	110	51	36	24
Jaccard	0.61	0.61	0.70	0.76	0.72	0.77
Dice	0.74	0.75	0.82	0.86	0.84	0.83
FPR	0.05	0.12	0.16	0.31	0.37	0.39
FNR	0.35	0.31	0.18	0.006	0.009	0.004

$$\begin{aligned} Jaccard &= \frac{AS \cap GT}{AS \cup GT} \\ DSC &= \frac{2 \times (AS \cap GT)}{AS + GT} \\ FPR &= \frac{AS \cap !GT}{GT} \\ FNR &= \frac{!AS \ \cap \ GT}{GT} \end{aligned} \tag{8}$$

The segmentation performance for 5 subjects are reported in Table 3. The # in the table is the number of DICOM images for each subject. It can be seen that the segmentation performance for our proposed simple algorithm which is exploited on the SoC is moderate and acceptable. Notice that, the region of interest for aneurysm detection and finding dangerous ones is the elementary branches of cerebral vessels (i.e. SAH) which are segmented correctly by our method. The extracted volumes for two different subjects via proposed segmentation algorithm are shown in Figure 10. One subject has aneurysm and another subject is without aneurysm.

4.1.2. Subjective segmentation performance

The segmentation algorithm was tested on the three subjects which have lowest dice and highest false negative rates. To evaluate the algorithm, subjective evaluation was carried where the resulted 3D volumes were compared

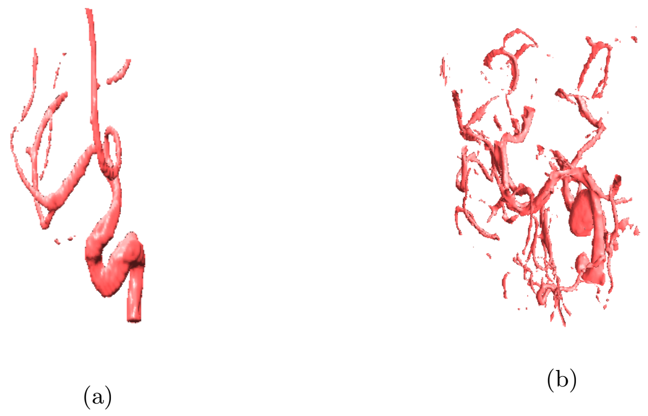


Figure 10: Extracted 3D volumes of vessels for two subjects via proposed segmentation method. a) Subject 1 b) Subject 4

with the Ground truth. Six evaluators participated in this process in which two of them are medicine graduates. They were given the results of the proposed segmentation and asked to assess the results on a scale from 5 (good) to 1 (bad), Table 4 shows the results of the subjective evaluation. The overall subjective evaluation was 3.81 out of 5 for this system. This shows that with some improvements, better results will definitely be accomplished.

The hard and soft thresholding techniques are tested with the same threshold value on the same subject. Figure 11 shows a comparison between the two techniques. As it can be seen in Figure 11, the soft thresholding technique caused a significant change in the results due to set pixels to zero or changes it according to the threshold. Therefore, the soft thresholding techniques would produce inaccurate results compared to the hard thresholding.

5. Conclusion

In this paper, a system for automatic aneurysm segmentation is developed and implemented on the Zynq SoC for different subjects with DICOM slices of Magngmetic Resonance Angiography. The segmentation algorithm is based on Haar wavelet and hard thresholding along with some additional steps and

Table 4: Subjective Evaluation for 3 different subjects.

	Evaluation Score. (5 is best)		
	Subject 1	Subject 2	Subject 3
Evaluator 1	3.5	3.5	4.5
Evaluator 2	3.75	4	4.5
Evaluator 3	3.5	3	4.5
Evaluator 4	3	3	4.5
Evaluator 5	4	3	5
Evaluator 6	3.5	4	3.75
Average	3.54	3.42	4.46

(a) Hard thresholding.

(b) Soft thresholding.

Figure 11: Comparison between the result by Hard and Soft Thresholding Techniques for a region.

criteria such as normalizing and correcting. The results show the proposed segmentation method and hardware implementation has an acceptable accuracy on the region of interest for aneurysm detection. The test data are from six subjects, each consisting 512×512, 16 bit DICOM slices of 3D rotational angiography. Comparison between the evaluator's ground-truth and automatic segmentation shows the dice score is above 70%. The mean of subjective scores which are assessed by the evaluators is 3.8 out of 5. Also, it was shown that, the proposed implementation is area-efficient and meet the real-time data processing requirements, where the Zynq SoC implementation is capable to process an image in an average time of 5.2 *ms* that is significantly faster than the runtime on a normal PC. In our next implementation phase, the automated aneurysm segmentation system on Zynq would be integrated with virtual reality facilities to create an interactive environment for effective treatment planning and training purpose.

Acknowledgement

This paper was made possible by National Priorities Research Program (NPRP) grant No. 5-792-2-328 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. We would also like to thank Mr. R.A.Y. Ayeshk and Mr. M.A. Hammami from the virtual reality Lab at Qatar University, for their assistance during the testing phase.

References

[1] R. T. Higashida, What you should know about cerebral aneurysms, Pamphlet. American Heart Association Cardiovascular Council.

[2] T. Mashiko, K. Otani, R. Kawano, T. Konno, N. Kaneko, Y. Ito, E. Watanabe, Development of three-dimensional hollow elastic model for cerebral aneurysm clipping simulation enabling rapid and low cost prototyping, World neurosurgery 83 (3) (2015) 351–361.

[3] A. Molyneux, I. S. A. T. I. C. Group, et al., International subarachnoid aneurysm trial (isat) of neurosurgical clipping versus endovascular coiling in 2143 patients with ruptured intracranial aneurysms: a randomised trial, The Lancet 360 (9342) (2002) 1267–1274.

[4] S. P. Dakua, J. Abinahed, A. Al-Ansari, A pca-based approach for brain aneurysm segmentation, Multidimensional Systems and Signal Processing (2016) 1–21.

[5] D. Lesage, E. D. Angelini, I. Bloch, G. Funka-Lea, A review of 3d vessel lumen segmentation techniques: Models, features and extraction schemes, Medical image analysis 13 (6) (2009) 819–845.

[6] C. Kirbas, F. Quek, A review of vessel extraction techniques and algorithms, ACM Computing Surveys (CSUR) 36 (2) (2004) 81–121.

[7] R. D. Rudyanto, S. Kerkstra, E. M. Van Rikxoort, C. Fetita, P.-Y. Brillet, C. Lefevre, W. Xue, X. Zhu, J. Liang, İ. Öksüz, et al., Comparing algorithms for automated vessel segmentation in computed tomography scans of the lung: the vessel12 study, Medical image analysis 18 (7) (2014) 1217–1232.

[8] F. Zhao, X. Xie, An overview of interactive medical image segmentation, Annals of the BMVA 2013 (7) (2013) 1–22.

[9] L. Wen, X. Wang, Z. Wu, M. Zhou, J. S. Jin, A novel statistical cerebrovascular segmentation algorithm with particle swarm optimization, Neurocomputing 148 (2015) 569–577.

[10] T. McInerney, D. Terzopoulos, Deformable models in medical image analysis: a survey, Medical image analysis 1 (2) (1996) 91–108.

[11] S. Zhao, M. Zhou, Y. Tian, P. Xu, Z. Wu, X. Shang, An effective brain vasculature segmentation algorithm for time-of-flight mra data, in: Virtual Reality and Visualization (ICVRV), 2015 International Conference on, IEEE, 2015, pp. 238–245.

[12] A. F. Frangi, W. J. Niessen, K. L. Vincken, M. A. Viergever, Multiscale vessel enhancement filtering, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 1998, pp. 130–137.

[13] C. Xiao, M. Staring, Y. Wang, D. P. Shamonin, B. C. Stoel, Multiscale bi-gaussian filter for adjacent curvilinear structures detection with application to vasculature images, IEEE Transactions on Image Processing 22 (1) (2013) 174–188.

[14] M. W. Law, A. C. Chung, Three dimensional curvilinear structure detection using optimally oriented flux, in: European conference on computer vision, Springer, 2008, pp. 368–382.

[15] F. Benmansour, E. Türetken, P. Fua, Tubular geodesics using oriented flux: an itk implementation, Tech. rep. (2013).

[16] R. Adams, L. Bischof, Seeded region growing, IEEE Transactions on pattern analysis and machine intelligence 16 (6) (1994) 641–647.

[17] Q. Li, Z. Wei, C. Zhao, Optimized automatic seeded region growing algorithm with application to roi extraction, International Journal of Image and Graphics 17 (04) (2017) 1750024.

[18] D. L. Wilson, J. A. Noble, Segmentation of cerebral vessels and aneurysms from mr angiography data, in: Biennial International Conference on Information Processing in Medical Imaging, Springer, 1997, pp. 423–428.

[19] M. Hernandez, A. F. Frangi, Non-parametric geodesic active regions: Method and evaluation for cerebral aneurysms segmentation in 3dra and cta, Medical image analysis 11 (3) (2007) 224–241.

[20] H. Bogunović, J. M. Pozo, M. C. Villa-Uriol, C. B. Majoie, R. van den Berg, H. A. Gratama van Andel, J. M. Macho, J. Blasco, L. San Román, A. F. Frangi, Automated segmentation of cerebral vasculature with aneurysms

in 3dra and tof-mra using geodesic active regions: An evaluation study, Medical physics 38 (1) (2011) 210–222.

[21] A. Firouzian, R. Manniesing, Z. H. Flach, R. Risselada, F. van Kooten, M. C. Sturkenboom, A. van der Lugt, W. J. Niessen, Intracranial aneurysm segmentation in 3d ct angiography: Method and quantitative validation with and without prior noise filtering, European journal of radiology 79 (2) (2011) 299–304.

[22] A. Nikravanshalmani, M. Karamimohammdi, J. Dehmeshki, Segmentation and separation of cerebral aneurysms: A multi-phase approach, in: Image and Signal Processing and Analysis (ISPA), 2013 8th International Symposium on, IEEE, 2013, pp. 505–510.

[23] Y. Sen, Y. Qian, A. Avolio, M. Morgan, Development of image segmentation methods for intracranial aneurysms, Computational and mathematical methods in medicine 2013.

[24] R. Hemalatha, N. Santhiyakumari, S. Suresh, Implementation of medical image segmentation using virtex fpga kit, in: Signal Processing And Communication Engineering Systems (SPACES), 2015 International Conference on, IEEE, 2015, pp. 358–362.

[25] N. Sudha, N. Santhiyakumari, B. Lay, Segmentation of bowel images and its implementation using virtex fpga kit, in: Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference on, IEEE, 2015, pp. 1–5.

[26] P. Dillinger, J. Vogelbruch, J. Leinen, S. Suslov, R. Patzak, H. Winkler, K. Schwan, Fpga based real-time image segmentation for medical systems and data processing, in: Real Time Conference, 2005. 14th IEEE-NPSS, IEEE, 2005, pp. 5–pp.

[27] X. Zhai, A. A. S. Ali, A. Amira, F. Bensaali, Mlp neural network based gas classification system on zynq soc, IEEE Access 4 (2016) 8138–8146.

[28] M. Vetterli, C. Herley, Wavelets and filter banks: Theory and design, IEEE transactions on signal processing 40 (9) (1992) 2207–2232.

[29] D. L. Donoho, J. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, biometrika 81 (3) (1994) 425–455.

[30] P. Morton, A. Petersen, Image compression using the haar wavelet transform, College of the Redwoods.

[31] X. Zhai, A. A. S. Ali, A. Amira, F. Bensaali, Ecg encryption and identification based security solution on the zynq soc for connected health systems, Journal of Parallel and Distributed Computing 106 (2017) 143–152.

[32] Vivado hls user guide, `www.xilinx.com`, accessed: April, 2017.

[33] Zynq-7000 all programmable soc, `www.xilinx.com`, accessed: April, 2017.

[34] Sdsoc design guide, `www.xilinx.com`, accessed: April, 2017.

[35] A. A. Taha, A. Hanbury, Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool, BMC medical imaging 15 (1) (2015) 29.