SKIN SEGMENTATION AND SKULL SEGMENTATION FOR MEDICAL IMAGING

Eric Yogi Tjandra[1], Liliana[1*], Leow Wee Kheng[2]

[1]Informatics Engineering Study Programs, Faculty of Industrial Technology, Petra Christian University
Jl. Siwalankerto 121-131, Surabaya 60236
[2]Department of Computer Science, School of Computing, National University of Singapore, COM1, 13 Computing Drive,
Singapore 117417, Republic of Singapore.
Email: eric.13292@gmail.com, lilian@peter.petra.ac.id, leowwk@comp.nus.edu.sg
*Corresponding autor

Abstract: In this paper aims we present tools for medical imaging applications to do skin and skull segmentation in a short time. The desired output for skin segmentation is a 3D visualization of the facial skin without any cavities or holes inside the head, while skull segmentation aims to create a 3D visualization of the skull bones. The algorithm used for skin segmentation is thresholding the image, extracting the largest connected component, and holefilling to fill the unnecessary holes. As for the skull segmentation, the process is done by removing the spines which is connected to the skull, and then extracting the largest connected component. Afterwards, mesh generation is done to produce the 3D objects from the processed images. This mesh generation process is done using the marching cubes algorithm. The testing results show that the skin and skull segmentation process will work well when there are no other objects that are connected to the skin or the skull. Skin segmentation process takes a significant amount of time, primarily caused by the holefilling process.

Keywords: Medical image, segmentation, largest connected component.

INTRODUCTION

Medical Image processing has important role in medical subject. With the help of this technology, diagnosing and treatment has developed to a better standard.

Skin segmentation is a tool to segment the facial skin from a DICOM image. The desired result of this tool is a 3D visualization of the facial skin without any cavities or holes inside the head and no other foreign objects. The algorithm consists of thresholding, largest connected component extraction, and hole-filling. The 3D visualization is done by generating mesh using marching cubes algorithm.

Skull segmentation is a tool to segment the skull bones from a DICOM image. The desired output of this segmentation is a 3D visualization of a skull bones without any unnecessary parts. These un-necessary parts include the spine, blood vessels, etc. The algorithm uses largest connected component extraction and thresholding, with the help of user input to remove the spines.

With these tools, users can save the facial mesh data or skull data for further processing. While DICOM image contains so many parts of body, these segmentation tools will be able to obtain the face's skin and skull, providing users the information of skull and faces data of the patient.

LITERATURE STUDY

Thresholding

Grey-level segmentation, or also called thresholding is a conversion proses between gray-level and bi-level image. [1] Thresholding has become a main method for image segmentation due to its intuitive and practical implementation characteristic.[2] One way to extract an object from its background is by determining a Threshold T to divide the object from the background. Each point (x,y) that fulfill f(x,y) > T is called an object point, while other points are called background point.

A thresholded image can be stated as:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases} \quad (1)$$

Every pixel labeled as 1 is an object, while pixel labeled 0 is background.

Connected Component Extraction

Pixel connectivity is the fundamental concept in digital imaging concept.[3] To determine if two pixels are connected, both pixels should be checked if they are neighbors and has gray level intensity that fulfill a certain similarity range. In general, there are 2 types of

adjacency: 4-adjacency and 8-adjacency. 4-adjacency accepts pixels that are adjacent in 4 directions: up, down, left, and right; while 8-adjacency accepts pixels that are adjacent in 8 directions: upper left, up, upper right, left, right, lower left, down, and lower right. Assume Y is a connected component in image A and a point p is inside Y. This following iterative equation states all element in Y:

$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3, \ldots \qquad (2)$$

where $X_0 = p$, and B is a desired structuring element. The algorithm run until $X_k = X_{k-1}$, and set $Y = X_k$.

Marching Cubes

Marching cubes is an algorithm that will construct 3D surface from a 3D array data. This algorithm is invented by Lorensen dan Cline (1987) [4] and is a widely popular isosurface algorithm. [5]

Marching Cubes uses a divide and conquer approach to find the position of a surface in an 8-pixel logical cube, 4 pixels each from adjacent slice. The algorithm determines the intersection of the surface with a cube, then moves to the next cube. The position of intersection between the surface and cube will determine the shape of triangle generated. To locate the intersection in a cube, we assign a value 1 if a value of vertex in a cube is larger or equals to the value of the surface we want to construct. These vertices are inside the surface. If the value is lower than the surface's value, it will be assigned with a 0 and are outside the surface. The cube will intersect with an edge of the surface where one of the vertex of the edge is inside the surface and the other vertex is outside.

There will be $2^8 = 256$ ways a cube can intersect the surface. Considering the symmetry property, complementary and rotational symmetry property, those 256 ways can be reduced to 14 patterns. Figure 1 show the 14 patterns of triangulated cubes.

The intersection points will define the vertices of the triangles, and the collection of the triangular facets across all the cubes will form the triangular mesh, creating an isosurface.

SYSTEM DESIGN

Skin Segmentation

The problem in facial skin segmentation is how to select the skin in the image, eliminating parts that are not needed and close the cavity parts of the face (e.g., nostrils, throat cavity, and so on). The expected outcome of this process is to show skin without any other parts that are not necessary.

The skin part usually has a low intensity in the image. Therefore, obtaining the facial skin part of the image can be done simply by thresholding an image with a low lower bound value. However, simply thresholding it will not be able to remove the cavities.

In this method, the steps for skin segmentation are thresholding, largest connected component extraction, and holefilling. All low value pixels will be thresholded into high intensity. Assuming that after the threshold the face part is the largest part in the image, the largest connected component extraction will extract the face's skin, removing all noise and also unnecessary parts. To remove the cavities, we will do holefilling process. The holefilling algorithm used in this project is based on the floodfill algorithm. However, the principle of the floodfill algorithm is used for marking all parts of image that is outside.

For every Z, we do markOutside process to mark a pixel whether it is outside or inside. This marking is stored in an array. After that, the checking begins. If a pixel is marked as inside and has low value, then it is considered as cavity and will be changed into a high intensity pixel. The flowchart of skin segmentation are shown in Figure 2.
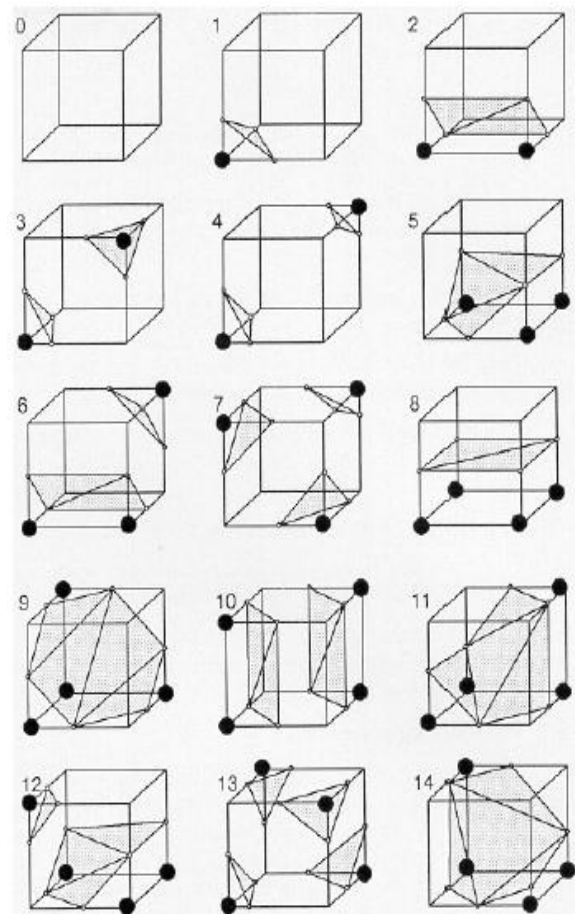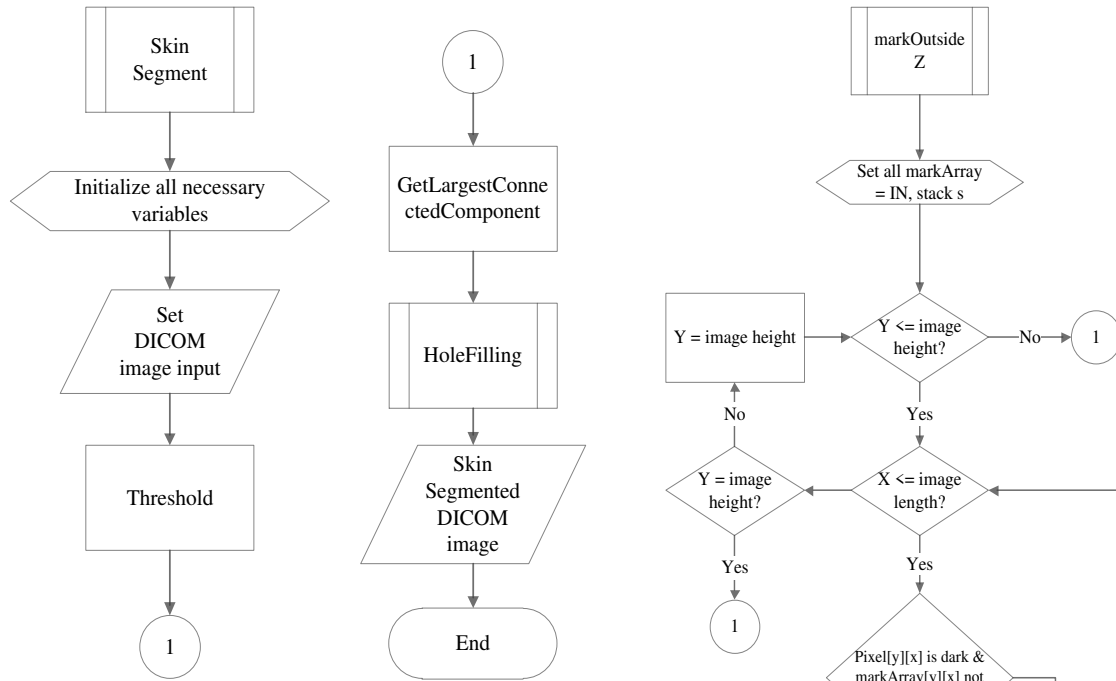

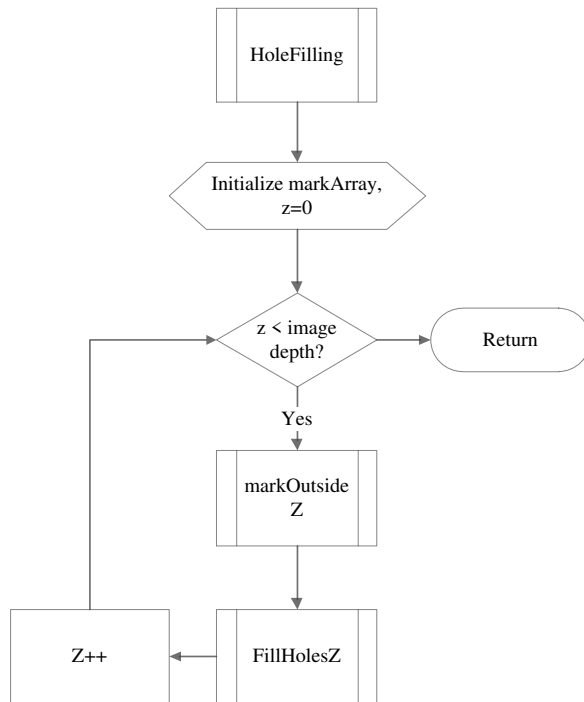**Figure 1.** Triangulated patterns

Figure 2. Skin Segmentation Flowchart
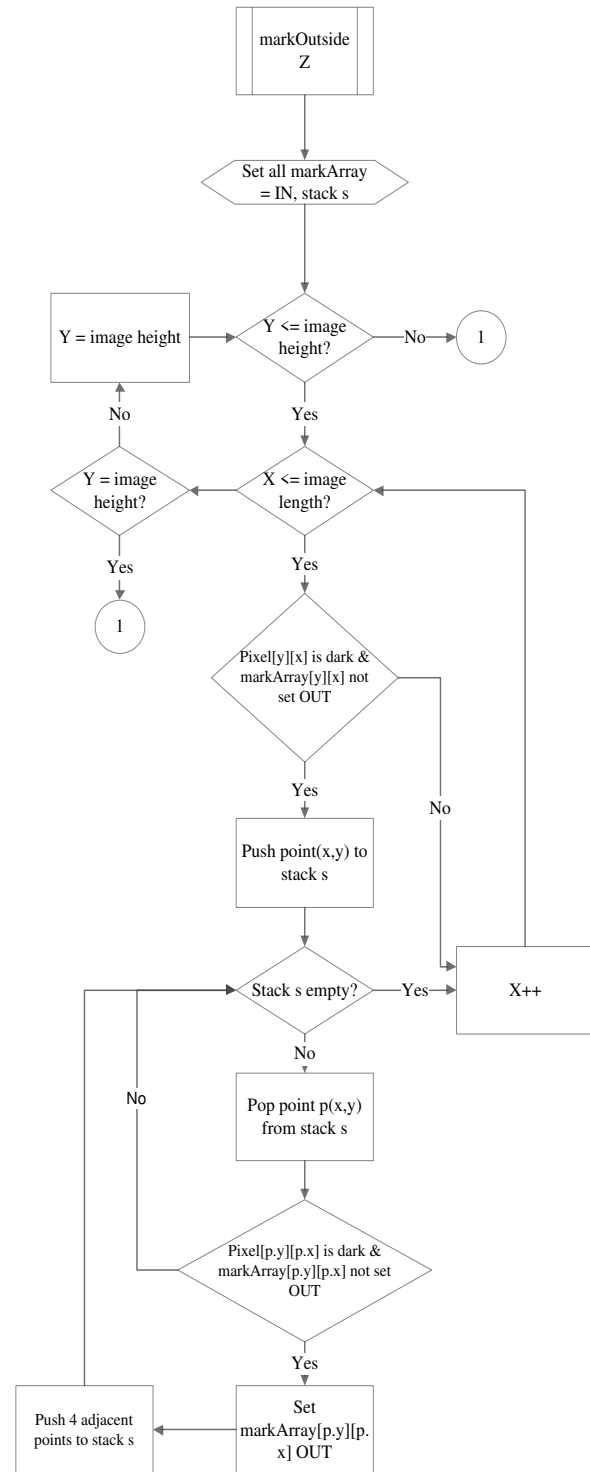
Figure 3. HoleFilling Flowchart
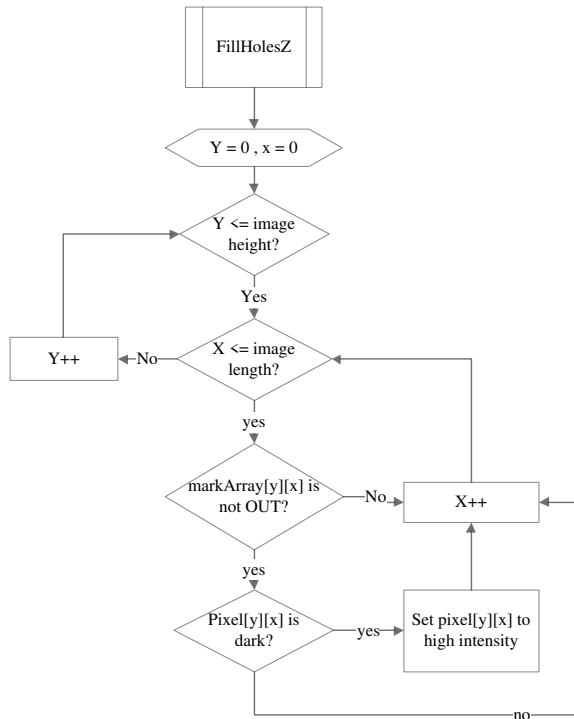
Figure 4. MarkOutsideZ Flowchart

Figure 5. FillHolesZ Flowchart

The holefilling process applied in this project, as shown in Figure 3, is based on floodfill algorithm using stack. In holefilling, there are 2 main methods: markOutside and fillHoles. MarkOutside is used to mark all pixels outside of object, while fillHoles will fill all holes inside the object. If a pixel is not marked as outside and have low intensity then it will be filled with high intensity. The process MarkOutside and FillHoles is done in 2 directions, Z-direction and X-direction. This method will remove all cavities and holes inside the head. The process markOutside and fillHoles in Z-direction is shown in Figure 4 and Figure 5.

Skull Segmentation

The problem in the skull segmentation is how to select the part of the skull and removing other parts that are not needed. The main characteristics of the skull can be seen in the image DICOM is the intensity level is high enough, so that thresholding method will be able to solve most of the problems in this process. Skull segmentation is done by thresholding and Largest connected component extraction. Thresholding will be selecting only the parts that have high intensity. This is done because the skull has a high intensity. But other parts that also have high intensity (e.g., blood vessels, etc.) will also be selected. Basically, the bones of the skull are always connected to the spine, while the desired results only contain the skull. Therefore running Largest connected component extraction, separation between the skulls and spine needs to be done by user. This can be done simply by selecting a particular area of the slice you want to delete. After that, Largest connected component extraction will automatically remove the spine. Skull segmentation flowchart is shown in Figure 6.
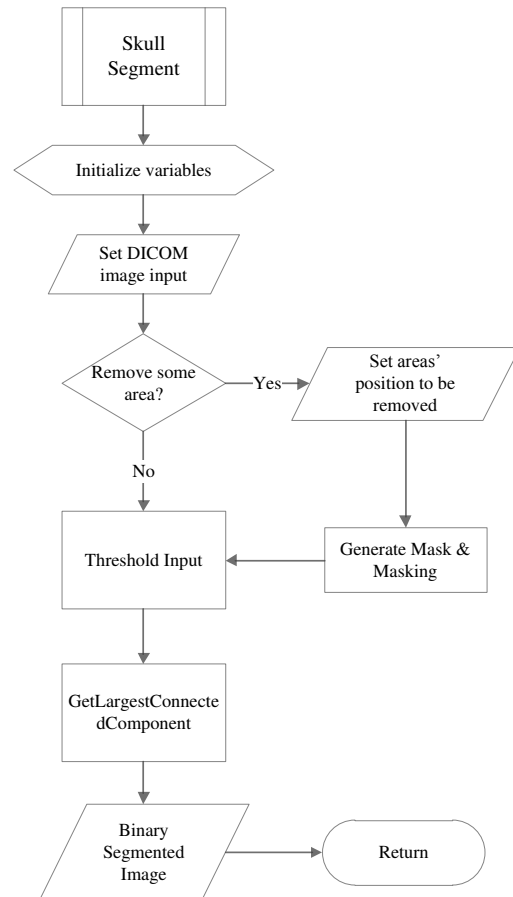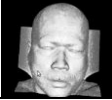

Figure 6. Skull Segmentation Flowchart

TESTING

Tests are conducted on the performance of the application of the algorithm. The tests are divided into quality testing and quantity testing. Quality testing is testing the quality of the results of the Skin Segmentation and Skull Segmentation algorithms. Quantity testing is testing the running time for Skin Segmentation and Segmentation Skull.

In Table 1 we can see that the results of skin segmentation successfully eliminated the holes on the head. All cavities and holes like nostrils, throat and the other closed. In addition, there is no noise. However, there are pads on the back of the head that come segmented. This happens because the original image, the pads are connected to the skin. Therefore, the algorithm used cannot delete these objects.

Table 1. Skin Segmentation Result on file Brain2

1	Filename:	Brain2 (512 x 512 x 381)
2	Skin Segmentation Result: (on middle slice)	
3	Mesh Generation Result:	
4	Keterangan	Tidak ada lubang, terdapat objek: bantalan

In Table 2 also shows the results of skin segmentation with better results. No foreign objects that is included in the segmented image.

Table 2. Skin Segmentation Result on file Brain3

1	Filename:	Brain3 (512 x 512 x 393)
2	Skin Segmentation Result: (on middle slice)	
3	Mesh Generation Result:	
4	Explanation	Tidak ada lubang

Table 3. Skull Segmentation Result on File Brain2

1	Filename:	Brain2 (512 x 512 x 381)
2	Skull Segmentation Result: (on middle slice)	
3	Mesh Generation Result:	

Table 4. Skull Segmentation Result on File Head

1	Filename:	Head (512 x 512 x 234)
2	Skull Segmentation Result: (on middle slice)	
3	Mesh Generation Result:	

Tables 3 and 4 show the results of the skull segmentation. The resulting image quality is quite good, the whole skull can be segmented and the spine can be removed. However, in Table 3, there are parts of the artifacts in the mouth which suffered little damage. Such damage is part of the original image

that cannot be solved by the algorithm used. While in Table 4 it can be seen that the shape of the skull can be obtained perfectly.

Table 5 shows that the Running time of the algorithm for the segmentation of skin is directly proportional to the size of the image. The larger the size of the image is processed, the greater the time required for the algorithm. The average time used by the algorithm to the size of the images used on these samples was 123.85 seconds or less than 2 minutes. The obtained time is still acceptable, considering there is no other algorithm that can perform this process with a faster time and the results are quite good. But in practice, it requires shorter time so that the system can actually be used.

One of the factors that led to the slow running time of this algorithm is in the process of holefilling. In the image, each pixel will be accessed at least two times. Once for checking and once again to change the pixel values.

Table 5. Testing of Skin Segmentation Running Time

No	File Name	Skin Segmentation				
		Threshold Value	LCC Time (secs)	Hole Filling Z (secs)	Hole Filling X (secs)	Total Time (secs)
1	Brain2	-184	36.67	34.25	44.85	115.77
2	Brain3	-199	34.10	32.48	43.02	109.60
3	Brain4	-355	29.91	36.04	47.21	113.16
4	Brain5	-128	33.82	29.68	38.93	102.43
5	FCL88E	-255	66.39	61.20	79.99	207.58
6	Head	-184	28.55	28.55	37.31	94.41

Table 6. Testing of Skull Segmentation Running Time

No	Nama File	Skull Segmentation		
		Area Removed	Threshold Value	Time (secs)
1	Brain2	X: 83-144 Y: 73-106 Z: 159	105	**26.02**
2	Brain3	X: 84-138 Y: 76-107 Z: 150	126	**29.95**
3	Brain4	X: 92-148 Y: 92-119 Z: 180	274	**23.4**
4	Brain5	X: 70-152 Y: 66-125 Z: 160	128	**24.1**
5	FCL88E	X: 144-231 Y: 182-236 Z: 156	232	**46.31**
6	Head	X: 213-285 Y: 225-289 Z: 151	128	**13.93**

Table 6 shows that the average running time of the skull segmentation is about 25 seconds. The obtained time is quite fast and can be accepted, with the quality of the results is quite satisfactory.

CONCLUSION

Based on the test results can be concluded as follows:

- This series of algorithms that are used for skin segmentation (Largest Connected Component and Hole Filling) work well when there are no other objects are attached directly to the skin of the face in the image.
- The running time the skin segmentation process takes is quite significant, mainly due to the Hole Filling process.
- This series of algorithms that are used for segmentation skull successful when there are no other objects are attached directly to the skull in the picture.

REFRENCES

[1] Parker, J.R., *Algorithms for Image Processing and Computer Vision*. Wiley Computer Publishing. 1997.

[2] Hussain, Z. *Digital Image Processing Practical Applications of Parallel Processing Techniques*. Singapore: Ellis Herwood Limited. 1991.

[3] Gonzales, R.C., Woods, R.E. *Digital Image Processing (2nd ed.)*. Upper Saddle River: Prentice-Hall. 2004.

[4] Lorensen, W.E., Cline, H.E. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *In Computer Graphics*, 21(4), 1987. pp. 163-169.

[5] Newman, T.S., Yi, H. *A Survey of the Marching Cubes Algorithm*. Elsevier ltd. 2006.