



Modification Rules for Improving Marching Cubes Algorithm to Represent 3D Point Cloud Curve Images

Dewi Rahmawati^{1,2} Riyanarto Sarno^{1*} Chastine Fatichah¹

¹*Informatics Department, Institut Teknologi Sepuluh Nopember Surabaya, Indonesia*

²*Software Engineering Department, Institut Teknologi Telkom Surabaya, Indonesia*

* Corresponding author's Email: riyanarto@if.its.ac.id

Abstract: Marching cubes is the most widely used isosurface algorithm for 3D reconstruction. For the case study, this paper used medical data from an MRI of brain images, especially in the corpus callosum (CC) part, and volume data from the stagbeetle dataset. This case study was selected to highlight the clinical importance of 3D image visualization. This study can help by showing solid anatomy shapes and locations, which can direct the location of a brain injury with a small error of less than 1 mm; therefore, it can support and minimize the risk of brain surgery. The case study is part of the brain called the corpus callosum, usually used as a reference for brain surgery. For the input data, this paper used 2D segmentation using deep learning methods to obtain the CC segments. This paper used 120 patients, 80% for training and 20% for testing from national hospitals. This paper found 11 sagittal slices containing the corpus callosum out of 166 slices for each patient. This work presents an improved MC algorithm that adds twenty new rules to the existing one, strengthening the rules for voxel representation and increasing the original marching cubes algorithm's 15 rules to 35. Thus, large holes are covered in the 3D reconstruction model, making it largely solid. The proposed 3D visualization achieved zero open edges for the datasets from the national hospital. The results showed that applying the improved marching cubes algorithm produced a 3D representation with better and more robust results, as evidenced by the presence of more vertices and triangles and the absence of open edges. Advanced marching cubes are a great way to remove open edges.

Keywords: 3D visualization, Marching cubes, 2D segmentation, Deep learning, Corpus callosum, Brain MRI.

1. Introduction

The popularity and general implementation of the marching cubes led to various algorithmic enhancements to handle ambiguities and appropriately track the interpolant's behaviour. One of the first isosurface extraction algorithms [1] was designed in 1987 to preserve the trilinear interpolant's topology. In his work, Chernyaev increases the number of cases in the triangulation lookup table to 33. [2] was the first to point out 1988 that the triangulation table presented by [1] Lorensen and Cline was inadequate and that certain marching cubes scenarios allow for multiple triangulations. In 1991 [3], the author noted uncertainties in the interpolant behaviour of the cube's face. The author suggested a test known as the asymptotic decider to monitor the

interpolant on the cube's faces accurately. The author noted in 1994 [4] that this ambiguity problem in marching cubes also exists within the cube. The author proposed a disambiguation test based on interpolant crucial points and added four new examples to the Marching Cubes triangulation table (subcases of cases 3, 4, 6, and 7). Despite the suggested changes to the algorithm and its triangulation table, the method and its triangulation table are insufficient.

Topological incoherencies persisted in the meshes produced by the marching cubes. Subsequently, marching cubes, a book written by author Montani in the same year [5], only addresses 23 of the 33 potential behaviours of the trilinear interpolant inside the cube. The marching cubes 33 method, developed in 1995 by [6] author Chernyaev, allows the creation of triangle models whose

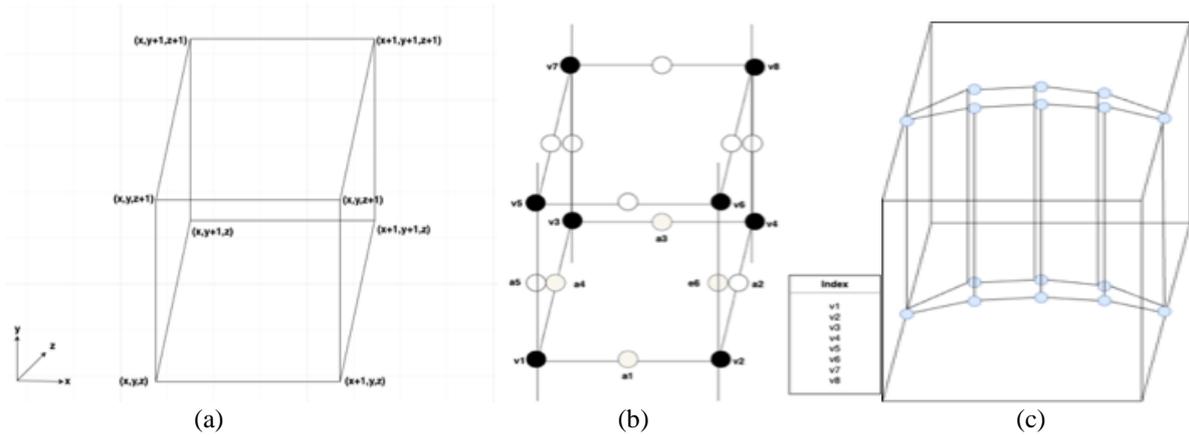


Figure. 1 (a) Illustration of a cube in the grid, (b) Cube numbering, and (c) Examples of curve cubes

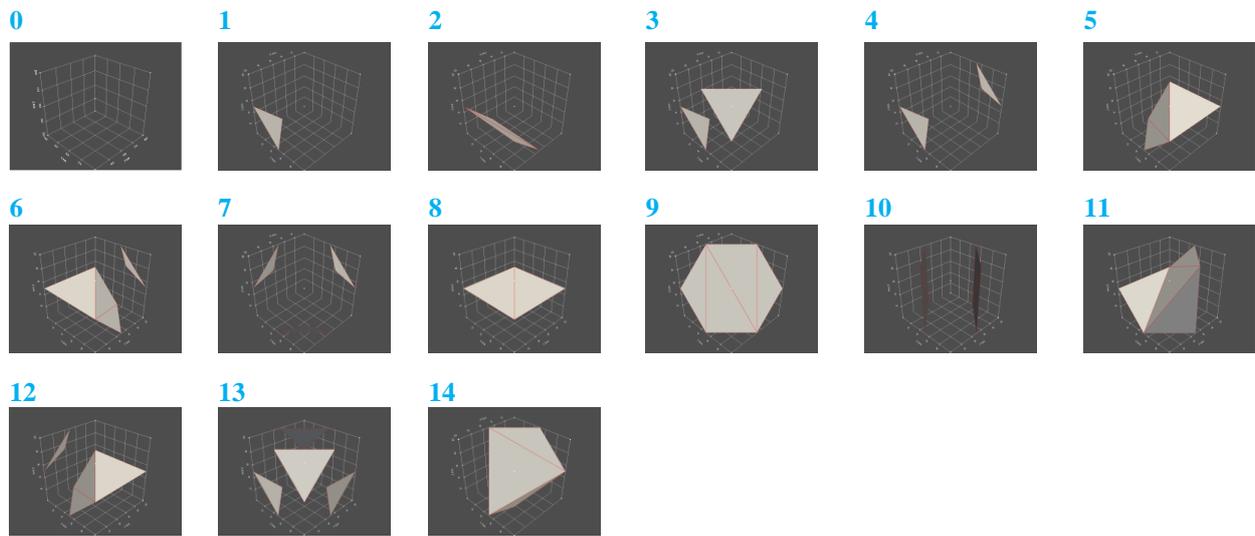


Figure. 2 Original lookup table marching cubes configuration (15 Rules)

topologies exactly match those of the trilinear function's isosurface for each cell, taken into consideration separately. As opposed to 15, as with the MC technique, it is demonstrated that there are 33 topologically distinct configurations. In 2000, the author [7] introduced a technique that results in topological inconsistencies in the resulting mesh because the ambiguities of the trilinear interpolant in the cube's interior should be considered while generating the triangulation. In 2003 [8], the author demonstrated that Chernyaev's lookup table is comprehensive and can capture all potential trilinear interpolant behaviours. The same title theme appeared in 2010 [9], 2013 [10], and 2020 [11]; marching cubes have been used in various research for medical pictures. The method [9] replaces cell edge interpolation by midpoint selection with the dual marching cubes approach, and it is based on the classic marching cubes algorithm coupled with a seed-occupying algorithm to increase the isosurface and prevent the detection of empty cubes. To ensure that the meshes produced by the marching cubes 33

were topologically accurate, [12] worked in 2013. The author described using the marching cubes 33 algorithm to create an expanded triangulation in 2019 [13]. The approach eliminates the degenerate triangles commonly found in marching cubes meshes by including grid vertices in the triangulation. The author coded the marching cubes 33 (MC33) algorithm in C and released it in the same year [14]. This implementation's procedures were designed with quick execution and little memory usage in mind.

The author published a method for accurately and efficiently evaluating volume for grid-based level sets in 2022 [15]. This method computes the volume of the shape(s) that are implicitly represented and is consistent with surface reconstruction using marching cubes. The author presented a technique for evaluating soft tissue data recreated using a marching cube algorithm in 2023 [16] and contrasted it with commercially sold software.

This paper combined volume data from the Stagbeetle dataset and medical data from an MRI of brain imaging in the corpus callosum (CC) region for

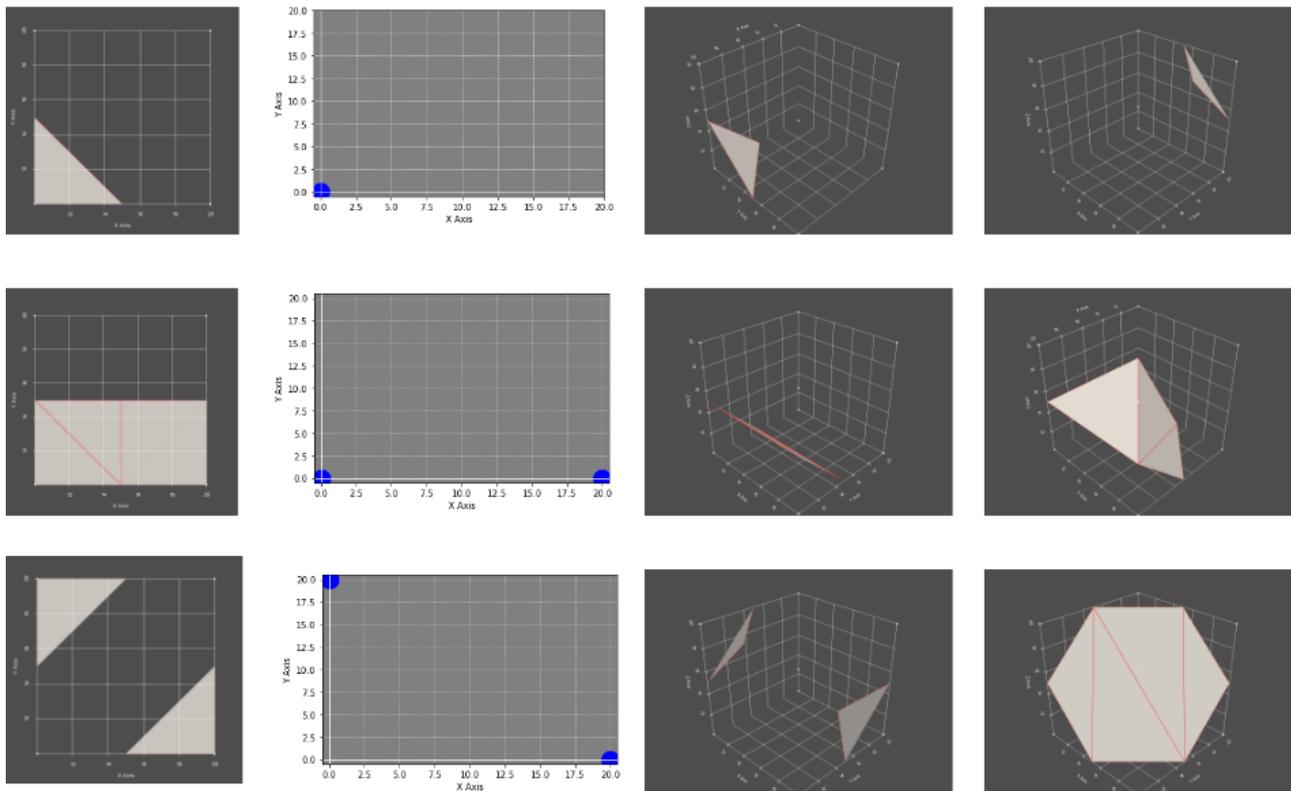


Figure. 3 The main problem rule configuration is on the shared faces of two cells in case of ambiguity

the case study. This case study was chosen to illustrate the practical use of 3D picture representation. By illustrating the forms and positions of solid anatomy, this study can help direct the placement of a brain lesion with an inaccuracy of less than 1 mm, supporting and reducing the risk associated with brain surgery. The corpus callosum, a brain region commonly used as a guide for brain surgery, is the subject of the case study.

The challenge in 3D reconstruction is the 3D visualization of multiple image layers, firstly because of the limitations of 2D MRI. Secondly, the quality of 2D medical images needs to be updated; hence, the use of 2D medical imagery continues to decrease. Thirdly, nowadays, most surgeons and medical experts prefer 3D visualizations due to the natural effect they create. Fourthly, more realistic visualization is required for further procedures. Fifthly, 3D imagery can present a visualization of actual body parts. Sixthly, the rapid development of technology supports the development of more advanced 3D visualizations. While surgeons require a 3D visualization of the corpus callosum to prepare for an operation, a visualization method with a high level of accuracy is needed to display the correct size and shape of the corpus callosum. In this study, 3D visualization of the corpus callosum was done using deep learning algorithms and the 3D marching cubes (MC) technique. We also attempted to reduce GPU

memory usage by using a patch-based method, splitting the augmented images into small patches, and training the dataset on these smaller patches. A few existing studies refer to the 3D visualization of parts of the brain. Many previous studies have implemented deep learning algorithms and 3D marching cubes to visualize parts of the brain without providing separate partitions of the brain. To solve this problem, the proposed method performs 3D visualization of the corpus callosum using an enhanced 3D marching cubes algorithm, our main proposal. The marching cube 33 algorithm was used for accuracy comparison.

The following is the motivation for this study:

1. Some studies have employed 3D visualization and spatial methods to discover the corpus callosum. However, no research used deep segments from images as the sample, even though this could provide an accurate 3D model of the corpus callosum. Moreover, the genu, as part of the corpus callosum, is not found in deep segments.
2. Several segmentation methods have been used in 3D visualisation, but no existing segmentation methods provide separate partitions of the corpus callosum (genu, rostrum, trunk, and splenium). Providing different sections is important because the application of this method is highly motivated by the possibility of observing the inside of the

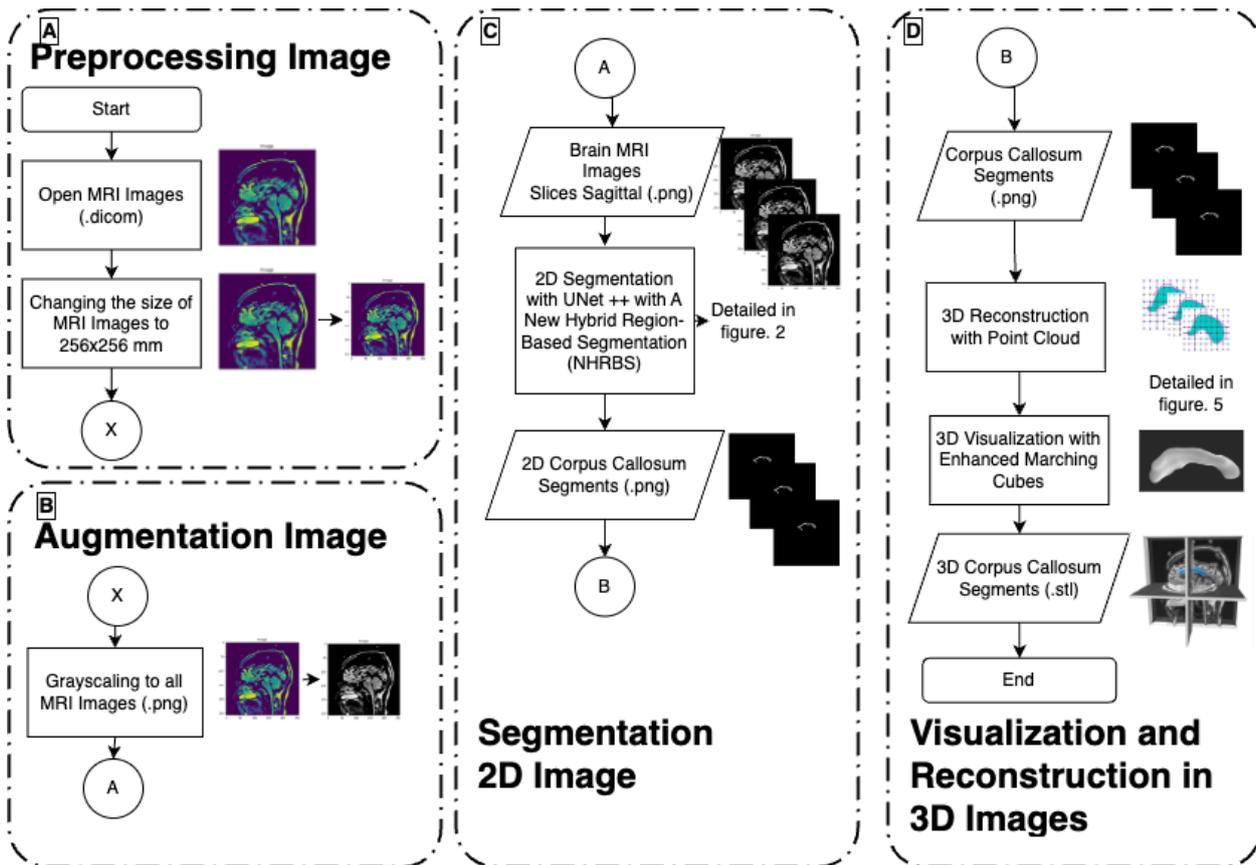


Figure. 4 Block diagram of the 3D visualization from (a), (b), and (c) 2D segmentation corpus callosum in brain MRI images using deep learning and (d) Enhanced marching cubes algorithm

brain.

3. This requires a reliable 3D visualisation of the corpus callosum so that its partitions can be accurately identified. Based on previous studies, marching cubes and deep learning algorithms can improve visualisation accuracy.

The main contributions of this study are related to these motivations:

1. Development of a new 2D segmentation method for 3D visualisation of corpus callosum data.
2. Detection of corpus callosum partitions from segments using deep learning by taking samples from MRI images.
3. We are enhancing the marching cubes algorithm for the 3D visualisation of corpus callosum data.

The rest of this paper is organised as follows. Section 1 presents the background of this research and some related works that also employed a corpus callosum segment detection framework and proposed enhancements of the marching cubes method.

2. Related works

Many related papers discuss 3D visualization

with marching cubes, but a few focus on 3D brain visualization with marching cubes. We examined and explored the qualities and shortcomings of many papers on 3D visualization with marching cubes. In 2019, [17] The author aims to solve the time problem to diagnose lung cancer as a whole with the need for many images from several sides problems by developing a system of 3D visualisation and reconstruction of lung cancer images. However, the algorithm used is a standard MC with 15 combinations of the cube. The two different shapes of cancer have different results. There is noise other than cancer that has not been separated from the cancer part on the parts. In [18], with the same author, the author aims to solve the time problem for the diagnosis of the heart condition of a patient. However, the dataset is limited, and the algorithm used is a standard MC with 15 cube combinations. Also, the limitation of this study is the error by lighting affection, a common problem in 3D image reconstruction.

In 2020, [19] the author improved the algorithm's performance. The author proposed to solve the surface void problem with Laplacian smooth iso-surface and use the polygon merging method to improve the display speed. 3D reconstruction was

performed using two sets of CBCT samples (jaw and dentition of two patients). However, the dataset is limited in CBCT image data based on .dicom. In 2020, By enhancing the voxel representation, the Author [20] proposed a refined MC algorithm. A significant reduction in the number of voxels that must be traversed during 3D reconstruction is achieved by increasing their volume. The simulation results demonstrate that the proposed algorithm is superior to the MC algorithm in terms of efficiency, smoothness, and distortion controllability in real time. However, this study just focused on voxel representation.

Later, In 2022, the Author [21] visualised 3D with the tool package VTK. The marching cube algorithm was used for surface rendering, and the ray casting algorithm was used for volume rendering and reconstruction. However, the dataset is limited, and the algorithm used is a standard MC with 15 cube combinations.

In 2023, the author [22] proposed a new cutting algorithm based on the marching cubes algorithm by modifying the voxel model. The simulation results show that the re-drawing process of the cutting surface based on the improved algorithm is more concise and can have higher cutting efficiency while ensuring cutting accuracy. However, as the algorithm requires different memory spaces depending on the discrete accuracy of the blank, when the discrete accuracy of the blank exceeds a certain level, it will increase the computational volume of the cutting calculation, which in turn reduces the efficiency of the cutting calculation.

In 2018, the author [23] conducted 3D visualisation using a 3D marching cubes strategy based on corpus callosum segmentation generated for a specified number of cross-sections from each survey. Improvements in the robustness and accuracy of the 3D marching cubes algorithm for isosurfacing were proposed in [23].

This study proposes a new 3D visualisation algorithm called enhanced marching cubes (EMC). Thus, large holes are covered in the 3D visualisation model, making it essentially smooth and solid. The proposed algorithm produced more faces and points, which resulted in a more complete and smoother surface. The dataset comprises general medical images (format MRI, png) and available mesh, namely the stagbeetle dataset (format dat, stl). The algorithm used enhanced marching cubes with 23 cube combinations.

Section 1 presents the introduction part that highlights the strengths of this proposed method. Section 2 presents related works and the drawbacks of each conventional technique and emphasises the

difference with other methods to clarify the position of this work further. Section 2 presents several articles discussed in the research survey. Section 3 presents materials and methods used in this research that show simulations and experiments. Section 4 presents the methodology and method by showing the corpus callosum segment detection framework results, the results of 2D segmentation using deep learning, and the proposed enhanced marching cube algorithm. Section 4 presents the effectiveness of the proposed method by comparing it with the state-of-the-art methods and showing comparison data. Section 5 presents the experimental results and their discussion. Finally, section 6 presents the conclusion showing this work's scientific contribution with concrete data.

3. Materials and method

The purpose of this study was to find corpus callosum partitions in brain MRI images using a brain segmentation framework. The proposed corpus callosum segment detection scheme is illustrated in Fig. 4, where the panels show how the framework detects the four corpus callosum partitions. In contrast, Fig. 5 illustrates a detailed overview of the proposed scheme of 2D Segmentations. The ground truth for validation of segmentation results was manually drawn by 3D SLICER 5.0.3, as in Fig. 6. The segmentation matches the manuals provided ground truth according to Eq. 1.

$$P(A, B) = \frac{A \cap B}{A \cup B} \times 100\% \quad (1)$$

Where $P(A, B)$ represents the segmentation accuracy in percentage, A and B represent the binary images of ground truth and target corpus callosum segmentation result, respectively. This study used datasets from two hospitals, as depicted in Table 1 and Fig. 2, conducted at Dr. Soetomo Hospital Surabaya and National Hospital Surabaya.

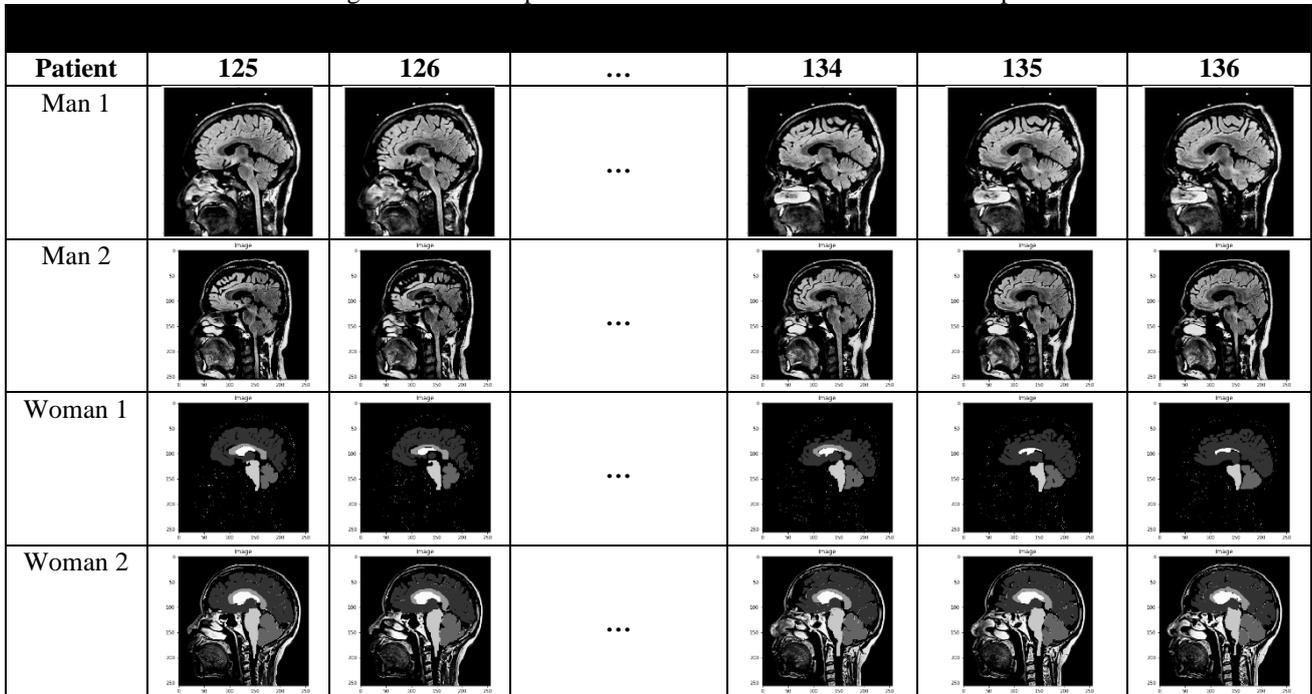
Details of the dataset, such as several data, image size and image type, can be seen in Tables 1 and 2. The first stage in Fig. 5 is the preprocessing and augmentation phase. The preprocessing step consists of resizing, orienting, and colour correction. After the preprocessing step, we apply an algorithm called new hybrid region-based segmentation. This algorithm uses local adaptive thresholding techniques to focus on the binarisation of grayscale images.

The algorithm consists of a novel thresholding technique using a combination of adaptive thresholds (a), large, connected components (b), k-means clustering (c), and mathematical morphology (d). For 2D segmentation using deep learning, this study used

Table 1. Demographic of sample data

Dataset	Total data	Training data	Testing data	Size	Type
National Hospital (NH) Surabaya 2020	80	64	16	256x256	.dicom
DR Soetomo (DRS) Surabaya 2019	40	32	8	256x256	.bin

Table 2. Brain MRI sagittal slices of 4 patients with 125 to 136 slices show that corpus callosum exists



Attention_Unet (1), FCN8s (2), myChannelUnet (3), UNet++ (4), R2Unet (5) and Unet (6) (7). The output of this algorithm is 2D segments of the corpus callosum (.png) as ground truth, as depicted in Fig. 4. After segmenting the 2D image phase, the output is used for 3D reconstruction with a point cloud. The next step is enhancing the marching cubes algorithm for 3D visualisation of corpus callosum data. The result of this algorithm is 3D segments of the corpus callosum (.stl).

The process was the same as in the data pre-processing step in Fig. 4. The resulting data were divided into training and test data, at 80% and 20%, respectively. The next step is the training phase using the training data. Deep learning is used to provide the input for the developed model. The model is carried out using training based on the system's epoch and batch size inputs, which are well generalised. The next step is to build a prediction image consisting of an input image, a prediction image, and a mask image.

The input image is the MRI image before the segmentation process. The predicted image is the result of the corpus callosum segmentation process. The mask image is the ground truth image used to create the corpus callosum segments. Each method produces axial, coronal and sagittal segments. Once finished, the generated model is the outcome, where

the format is .pth.

This stage is called the training phase. The developed model obtained from the training phase is then tested using new data, namely the test data, to compare the performance. This phase is called the testing phase.

4. Methodology and method

4.1 Corpus callosum segment detection framework and preprocessing scheme

The corpus callosum segment detection framework details are shown in Fig. 5. Figs. 5 (fields A, B, C, D, E, pre-processing phase) are based on previous work. Herrera [29] presented 2D corpus callosum segmentation in midsagittal slices. This investigation method is sensitive to the choice of the threshold and therefore, either fails or produces inaccurate results based on the source of the inflection points. Although the ROQS method in this study worked well, it had no post-training initiation or parameter selection requirements. Manic [30] proposed a multi-threshold method using the chaotic cuckoo search algorithm (CCS) and a threshold method using the Preferred Threshold method. Satapathy [31] proposed histogram-based two-level and multi-level thresholding methods. The proposed

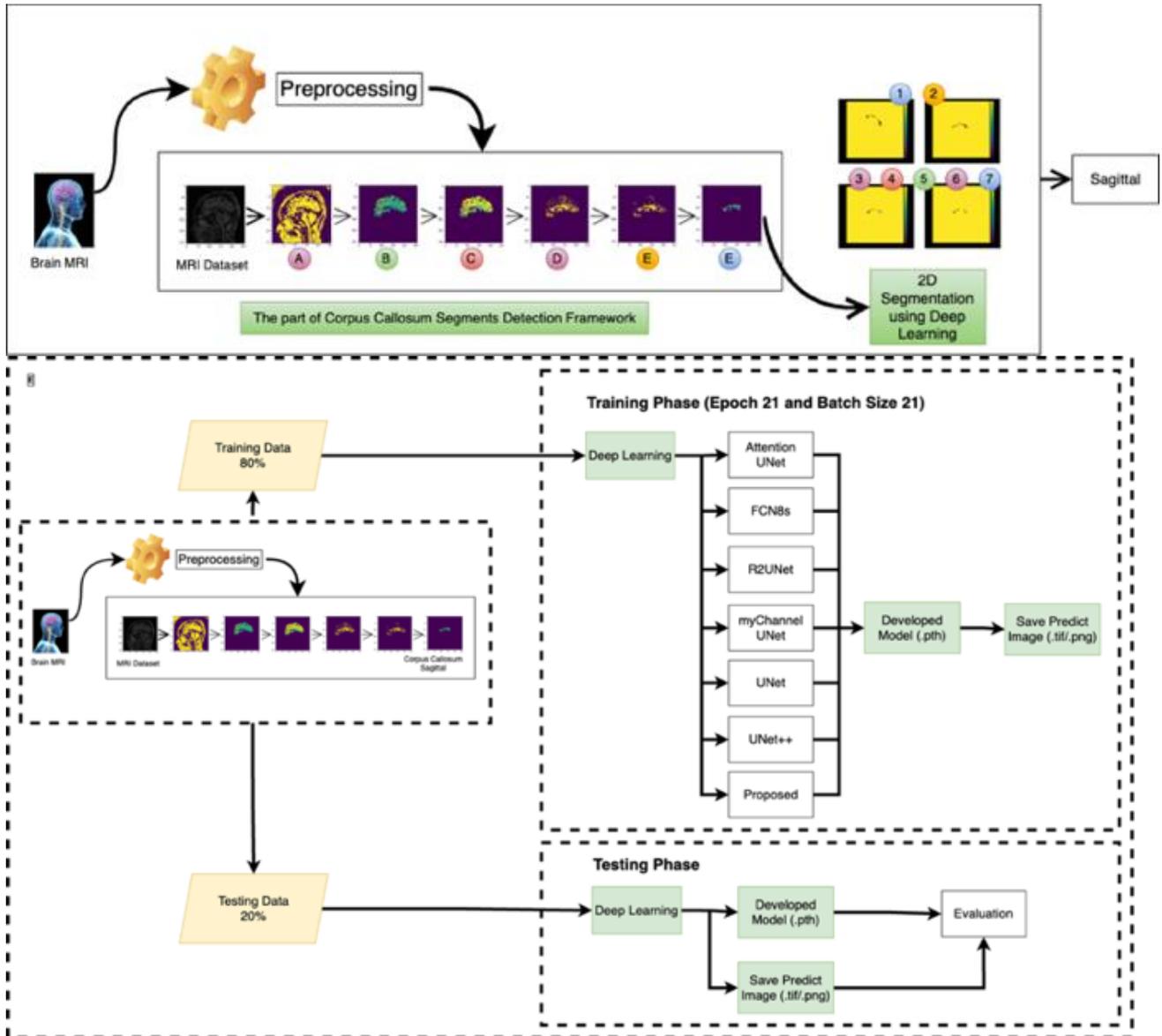


Figure. 5 Proposed partial corpus callosum segments detection scheme

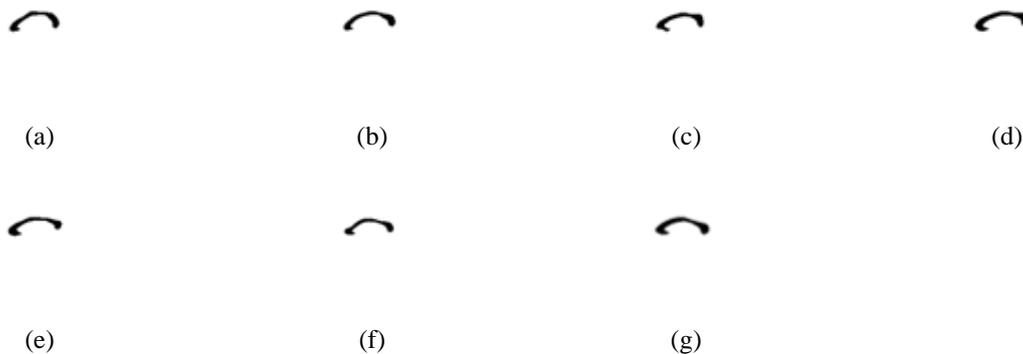


Figure. 6: (a) The first column; the ground truth of the corpus callosum, (b), (c), (d), (e), (f) and (g) Second column until seventh column: the result of deep learning approach (a) Ground truth corpus callosum patient 1, (b) Attention UNet segmentation result, (c) FCN8s segmentation result, (d) R2UNet segmentation result, (e) myChannelUNet segmentation result, (f) UNet segmentation result, and (g) UNet++ segmentation result

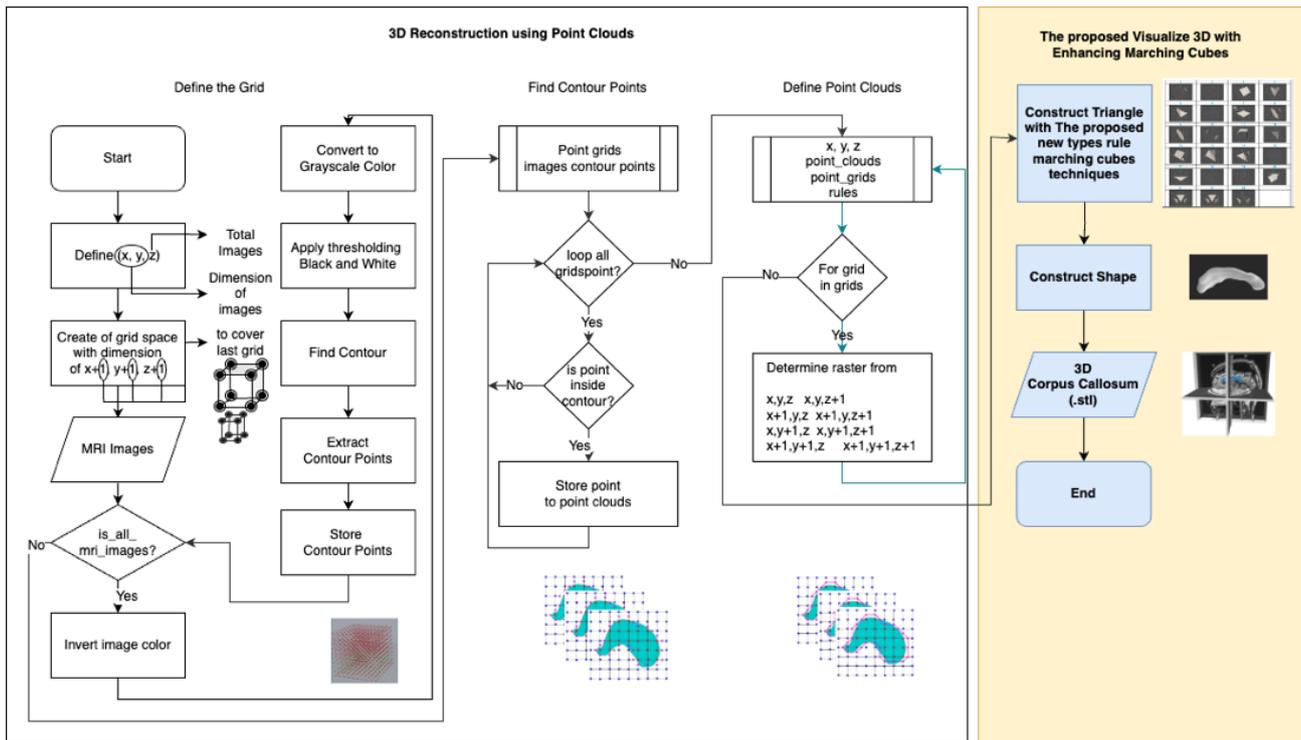


Figure. 7 System block diagram of the enhanced marching cubes algorithm

methods achieved good performance. They recommended extending the implementation in Ikeda maps to other chaos maps in future work. For the eyes, they used the DRIVE dataset. For the cell nucleus, they used DSB 2018 from the data competition of that year. The esophagus cancer dataset came from the First Affiliated Hospital of Sun Yat-sen University. This study used .dicom and .tiff for the corpus callosum dataset and .png for the eye, esophagus, and cell nucleus datasets as input data. For the corpus callosum, this study used ABIDE 2015 (<https://s.id/1galu>) and OASIS 2018 (<https://s.id/1galf>) from the data competitions of each respective year.

This study used datasets from two hospitals. These data were corroborated by physician diagnosis. Before using the data obtained from the hospitals, the data were processed so that no personal data such as names, addresses or phone numbers were stored in the dataset. As a result, third parties cannot identify personal data in the dataset. The hospitals from which the data were gathered and the study participants gave their consent. Every hospital staff member requested that the subjects fill out a consent form and submit a copy of their identification to get their consent. The paper [25] presents a novel hybrid region-based segmentation method that finds the local average of neighbouring pixels in the window regardless of window size. It substitutes the previous method of local thresholding with the integral-sum image methodology. Computing local thresholds with this

method is quicker than without it. Sauvola and Pietikainen's local binarization methodology can be used, and the binarization speed performs better than the global binarization method. Eq. (2) is used to formulate the threshold $T(x,y)$.

$$b(x,y) = \begin{cases} 0 & \text{if } I(x,y) \leq T(x,y) \\ 1 & \text{vice versa} \end{cases} \quad (2)$$

where $I(x,y) \in [0,1]$ is the pixel intensity at position (x,y) and $b(x,y)$ is the binarized picture. Fig. 5 illustrates the function of the new hybrid region-based segmentation algorithm.

4.2 Proposed enhanced marching cubes method

At the marching cubes stage, a voxel has eight vertices, and each vertex has two states, in-plane and out-plane, so there are $28 = 256$ cases of the relationship between voxels and iso-surfaces. Using the marching cubes' properties, the 256 cases can be reduced to 15 triangular patch configurations. This gives us a partial picture of the corpus callosum. This study proposes a new rule for the marching cube technique called custom marching cubes. The relationship between the voxels and the iso-surfaces produces these customary rules, as in Figs. 7 and 8. The first step in Algorithm 1, as illustrated in Fig. 7, is to define the grid using points X, Y, and Z. First, a 3D grid (xy, xz, yz) is created, and a group of points from the plane index are selected, where X and Y are

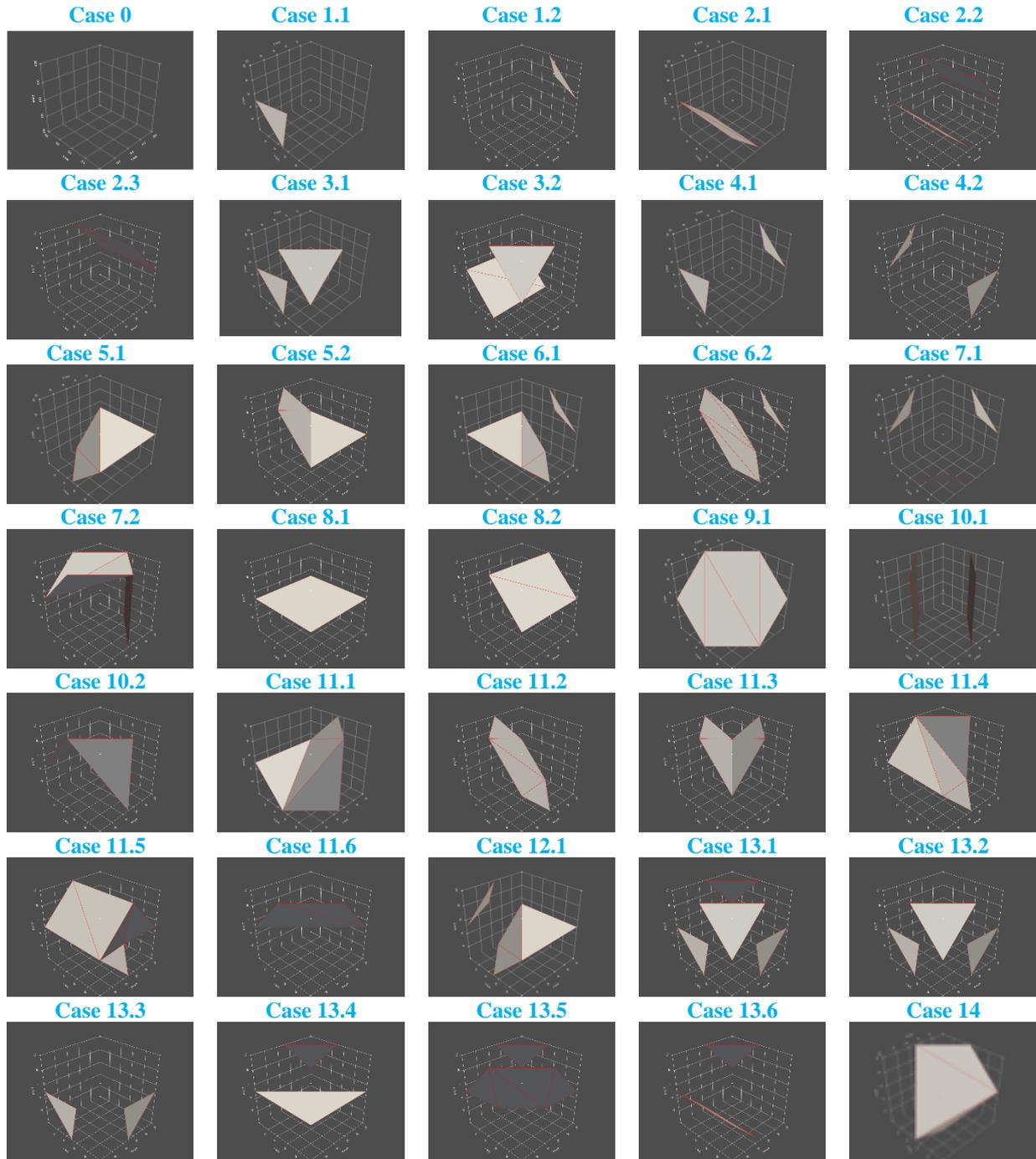


Figure. 8 The lookup table proposed rules for enhanced marching cubes

matrices with the grid coordinates, and Z is a matrix with z values at the grid points.

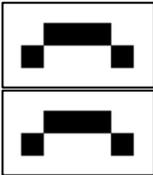
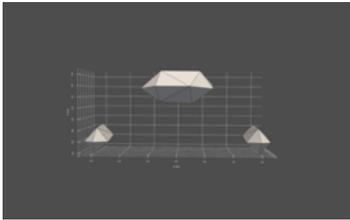
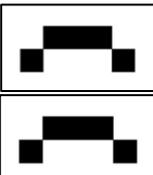
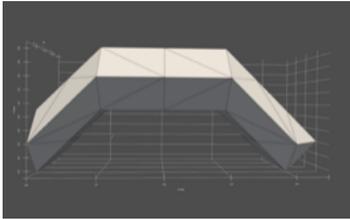
A mesh chart consists of lines that connect the points. The surface plot colours the areas within the mesh lines. Radar and surface charts are 3D plots functions of $z = f(x, y)$, where x and y are the independent variables and z is the dependent variable. This means that within a given domain, we can compute the z value for any combination of x and y . The radar and surface charts are created in three steps. The first step is to create a grid in the xy plane covering the functional area. The second step is to

compute the z -value at each point in the grid. The third step is to create the chart. The z value for each address is calculated from the corresponding x and y values. For example, z is given using Eq. (3).

$$z = \frac{xy^2}{x^2+y^2} \tag{3}$$

The advantages of our approach are that it enhances the rule for the representation of the voxels of the marching cubes so that holes are covered, and

Table 3. Configuration of construct triangle with the proposed new types rule marching cubes techniques

Input Curve Mesh	Rules	Output Curve Mesh
	<p>With Marching Cubes Original (15 rules): Faces: 60 Points: 36 Open Edges: 0</p>	<p>With Marching Cubes Original (15 rules): There is a hole faces</p> 
	<p>With Proposed Marching Cubes (15 + 20 new rules): Faces: 68 Points: 36 Open Edges: 0</p>	<p>With Proposed Marching Cubes (15 + 20 new rules): There are no hole faces</p> 

the model becomes solid mainly. A disadvantage is that we only do pre-processing on the corpus callosum dataset. The next step is to find a contour plot, as illustrated in Fig. 7. A contour plot is simply a graph in the xy plane that shows the contours of a function of two variables, $z = f(x,y)$. The contour lines are obtained from the surface by intersecting the horizontal plane. A contour plot is a collection of level curves labelled with function values. Given the formula for the function $z = f(x,y)$, we can easily find the contour equations. Each contour is obtained by slicing the surface of the horizontal plane $z = c$, so Eq. (4) for the contour at height c is straightforward.

$$f(x, y) = c \tag{4}$$

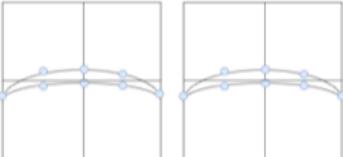
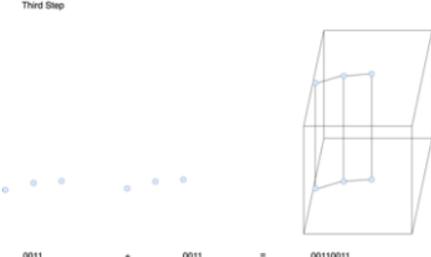
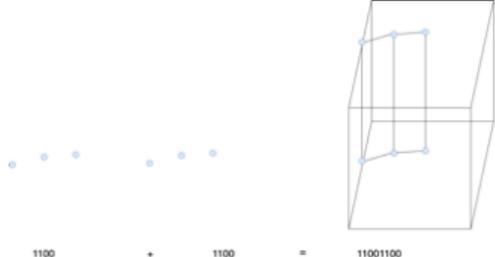
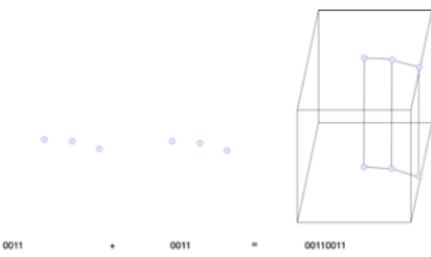
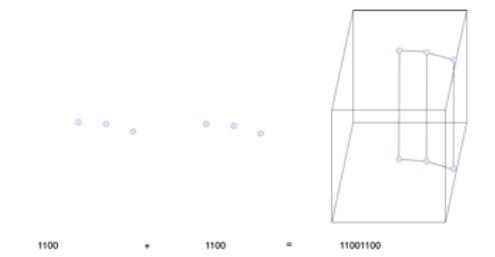
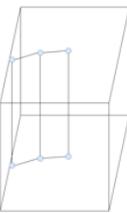
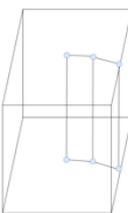
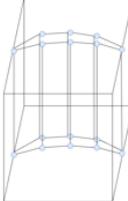
In the marching cubes algorithm, which is used for constructing polygonal representations of isosurfaces in 3D scalar fields, there are 256 possible unique configurations of binary values at the vertices of a cube. Each configuration can be associated with triangles that approximate the isosurface passing through that cube. However, not all these 256 configurations result in distinct shapes; some configurations are symmetric or equivalent to others. To reduce redundancy and improve efficiency, the number of unique cases is reduced to 15, as in Fig. 2, by considering symmetry and inversion, as in Fig. 1. When implementing the marching cubes algorithm, we typically use a lookup table or a similar data structure to determine the cube's configuration and retrieve the appropriate set of triangles to construct

the isosurface, as in Table 5. The selected triangles are then positioned based on the intersection points within the cube, creating a smooth approximation of the isosurface. The extended marching cubes algorithm is often used in contexts with higher accuracy, such as medical imaging or scientific visualization, which is required.

It provides a more precise depiction of the isosurface, crucial for precisely viewing complex structures. The "ambiguous case" in the marching cubes algorithm, as depicted in Fig. 3, refers to instances in which the decision to triangulate a cell is not uniquely determined by the placements of its vertices and the level of its isosurface. This ambiguity can lead to several possible triangulations, and the algorithm needs a way to manage these scenarios. The "ambiguous case" refers to a circumstance in which the algorithm finds uncertainty in determining how to triangulate a cell due to ambiguity.

Arrangement of vertices and the isosurface level. This can lead to multiple valid ways of triangulating the cell to represent the isosurface. It occurs when the data values at the vertices of a cube do not provide enough information to uniquely determine how the vertices should be connected to form triangles that accurately represent the isosurface. In Table 3, there are 2 cases of ambiguous faces. Case 1: The diagonal nodes are completely separated from each other. Nodes that are adjacent to each other are not divided. It is also obvious that if a face has three or four nodes of the same sign, they are non-separated. Case 2: The diagonal nodes are joined or connected inside the cell.

Table 4. Configuration of construct triangle with the proposed new types rule marching cubes techniques

Step	Configurations
Step 1	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>First Step</p>  </div> <div style="text-align: center;"> <p>Second Step</p>  </div> </div>
Step 2	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Third Step</p>  <p>0011 + 0011 = 00110011</p> <p>0011+0011=00110011</p> </div> <div style="text-align: center;">  <p>1100 + 1100 = 11001100</p> <p>1100+1100=11001100</p> </div> </div>
Step 3	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>0011 + 0011 = 00110011</p> <p>0011+0011=00110011</p> </div> <div style="text-align: center;">  <p>1100 + 1100 = 11001100</p> <p>1100+1100=11001100</p> </div> </div>
Step 4	<p>Fourth Step</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>00110011</p> </div> <div style="text-align: center;">  <p>00110011</p> </div> <div style="text-align: center;">  <p>11001100</p> </div> <div style="text-align: center;">  <p>11001100</p> </div> </div>

Resolving ambiguous faces ensures that the generated surface mesh accurately represents the object's surface geometry based on the underlying scalar field. The lookup tables and rules used in the enhanced marching cubes algorithm are designed to handle these cases and produce a coherent and

visually accurate representation of the object's surface, as in Fig. 8, Tables 3-5.

We make use of the fact that function F varies bilinearly over the surface and each plane parallel to the surface to resolve the ambiguity on the surface and within the cube. Our proposed new type of rules

Table 5. Truth table configuration (comparison between construction in MC and EMC; construction without holes)

Rule MC	Rule EMC	Configuration	Results 3D in MC	Results 3D in EMC
167	61	0b00111101		
168	62	0b00111110		
171	91	0b01011011		
173	94	0b01011110		
174	103	0b01100111		
175	107	0b01101011		
176	109	0b01101101		
177	110	0b01101110		
180	118	0b01110110		
181	121	0b01111001		
182	122	0b01111010		
183	124	0b01111100	FALSE	TRUE
185	151	0b10010111		
186	155	0b10011011		
187	157	0b10011101		
188	158	0b10011110		
189	167	0b10100111		
191	173	0b10101101		
194	181	0b10110101		
195	182	0b10110110		
196	185	0b10111001		
198	188	0b10111100		
199	199	0b11000111		

for the marching cubes technique, called custom marching cubes, is shown in Fig. 8 and Table 3. There are 23 rules in total. In the generated result, illustrated in Table 3, large holes are covered, and the 3D visualisation model is mainly solid—any fixed variable F in Eq. (6).

$$F(s, t) = A(1 - s)(1 - t) + B_s(1 - t) + C(1 - s)t \quad (6)$$

Where:

$$A = F_{000}(1 - q_0) + F_{100}(q_0)$$

$$B = F_{010}(1 - q_0) + F_{110}(q_0)$$

$$C = F_{001}(1 - q_0) + F_{101}(q_0)$$

$$D = F_{011}(1 - q_0) + F_{111}(q_0)$$

A, B, C, and D represent the values at the vertices of the ambiguous surface in a face ambiguity resolution based on the bilinear transformation of F on the surface, where A and C are positive and B and D are negative. The contour F (s, t) is a hyperbolic curve, as is readily apparent. The same values at the intersections are compared asymptotically with the bilinear interpolation values F to determine which nodes are isolated and which are F (s, t).

$$F(s, t) = \frac{AC - BD}{A + C - B - D} \quad (7)$$

Since the denominator of Eq. (7) is always positive, comparing the two products AC and BD is sufficient to identify which nodes are connected. The positive node is therefore connected while the negative node is detached if $AC > BD$;

Otherwise, positive nodes are disconnected, and negative nodes are connected. Enhanced marching cubes aim to improve the handling of these ambiguous configurations and provide more robust and accurate surface reconstructions. To deal with ambiguous cases, use additional rules or secondary lookup tables to make better decisions about the surface triangulation within the cube. By adding twenty new rules algorithms based on the shape of the surface. If the image is planar, 15 rules are enough; if the image is curved, it needs to be more complex, namely adding twenty rules to make thirty-five rules.

Configuration 0, 14: Simple configuration.

There are no ambiguous faces and no internal ambiguity in configurations 0 and 14.

Configuration 1-13: Complex configuration.

Configurations 1-13 have uncertain exteriors and interiors. For greater conformity of the triangle model to the isosurface, it is important to employ an additional face, vertex and nodes inside the cube. As seen in Fig. 8, it will be a common node for all triangles in the model.

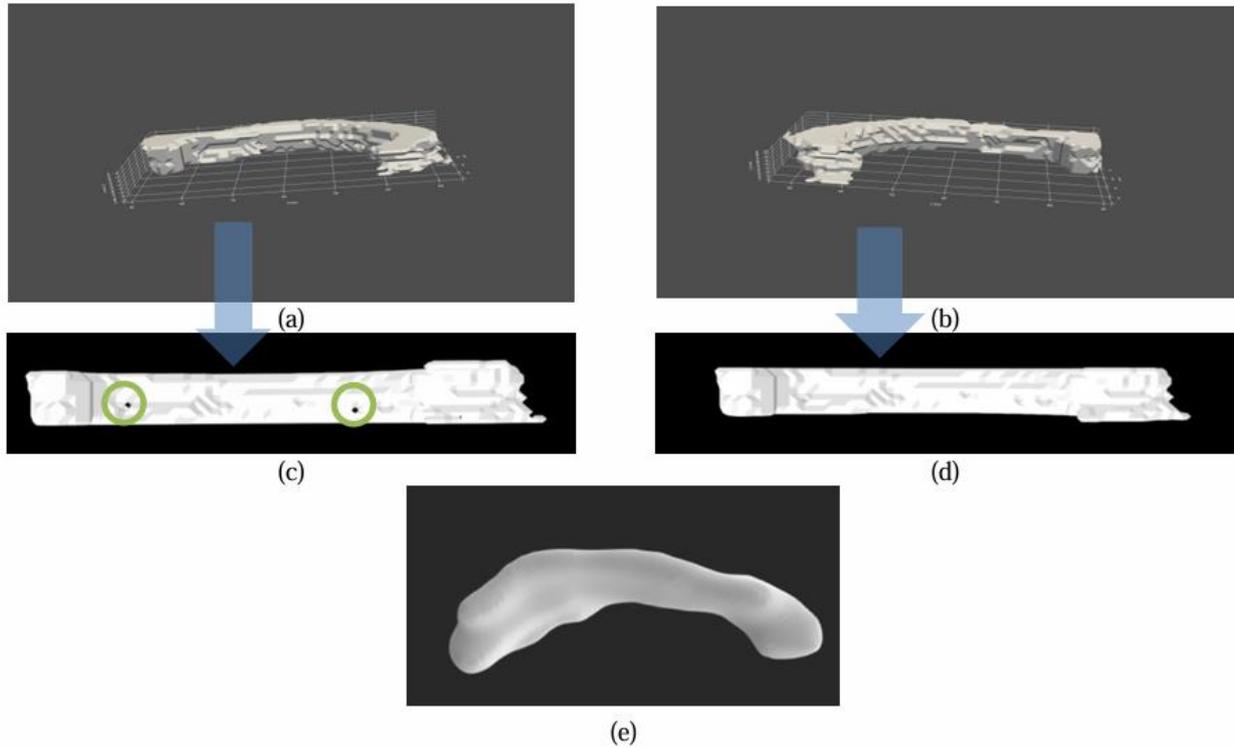


Figure. 9 Proposed 3D reconstruction of the corpus callosum made with the original and with the enhanced marching cubes algorithm: (a) Reconstruction made with the original marching cubes algorithm (grid-view), (b) Reconstruction made with the enhanced marching cubes algorithm (grid-view), (c) Reconstruction made with the original marching cubes algorithm (image-view), (d) Reconstruction made with the enhanced marching cubes algorithm (image-view), and (e) Reconstruction made with the enhanced marching cubes algorithm (grid-view)

Table 1. Volume and dimension measures of 3D based on 3D Alan Rossets (Online)

	3D Alan Rossets (Results EMC)	3D Alan Rossets (Results MC)
Material Volume	2.92 cm ³	2.94 cm ³
Box Volume	14.78 cm ³	14.78 cm ³
Surface Area	23.22 cm ²	23.14 cm ²
Model Weight	3.68 g	4.14 g
Model Dimensions	2.80 cm x 6.60 cm x 0.80 cm	2.80 cm x 6.60 cm x 0.80 cm

Table 2. Volume and dimension measures of 3D based on numpy_stl get_mass_properties (Python Offline)

	Enhanced Marching Cubes (EMC)	Marching Cubes (MC)
Volume	2922	3123
Visualization time/s	0.8969 s	0.1567 s

Table 3. Comparison of experimental 3D visualization results from the original and the (proposed) enhanced marching cubes algorithm with corpus callosum dataset

Code	Size Image	Vertices	Triangles
Traditional Marching Cubes	257, 257, 9	2992	5960
	257, 257, 10	3924	7826
Marching Cubes 33	257, 257, 9	2216	4430
	257, 257, 10	2706	5406
Our MC	257, 257, 9	2992	5980
	257, 257, 10	3924	7844

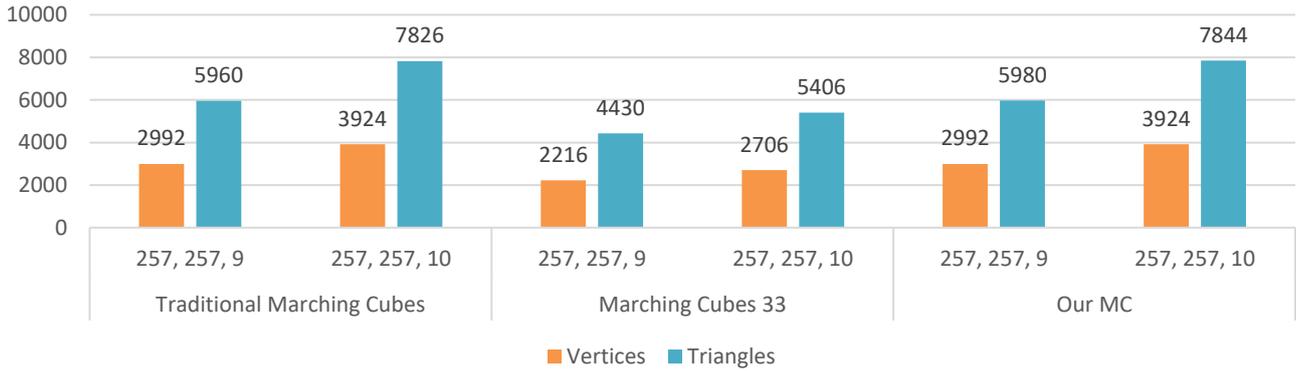


Figure. 10 Comparison of the TMC and MC with the 3D reconstruction efficiency of our improved algorithm with corpus callosum dataset

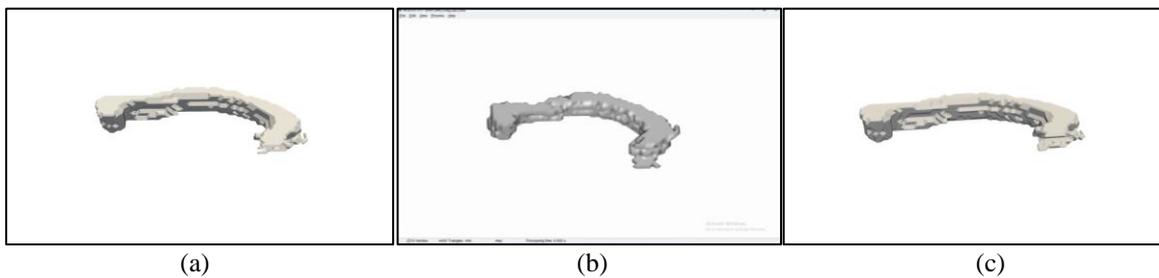


Figure. 12: (a) Surface extracted from the CC 257x257x9 dataset with traditional marching cubes, (b) with marching cubes 33, and (c) with proposed marching cubes

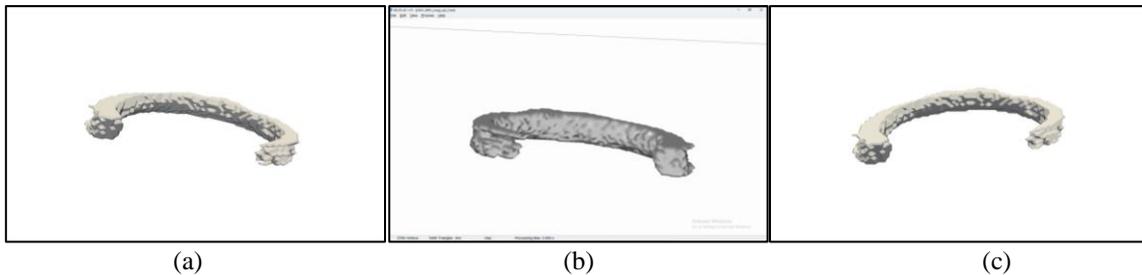


Figure. 13: (a) Surface extracted from the corpus callosum 257x257x10 dataset with traditional marching cubes, (b) with marching cubes 33, and (c) with proposed marching cubes

Table 4. Comparison of experimental 3D visualization results from the original and the (proposed) enhanced marching cubes algorithm with stagbeetle dataset

Code	Size Image	Vertices	Triangles
Traditional Marching Cubes	208x208x123	68664	136312
	277x277x164	123362	245628
	416x416x247	283968	566412
	832x832x494	1181018	2359710
Marching Cubes 33	208x208x123	68665	137242
	277x277x164	123362	246654
	416x416x247	283968	567924
	832x832x494	1181029	2362226
Our MC	208x208x123	68664	137228
	277x277x164	123362	246646
	416x416x247	283968	567920
	832x832x494	1181018	2362208

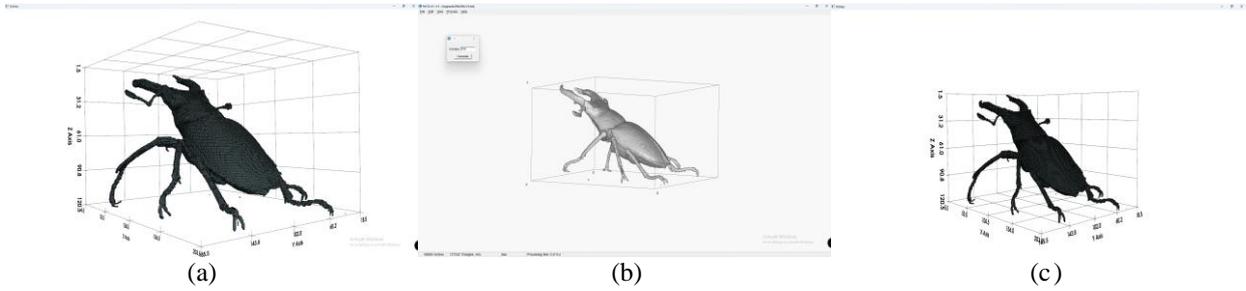


Figure. 14: (a) Surface extracted from the stag beetle 208x208x123 dataset with traditional marching cubes, (b) with marching cubes 33, and (c) with proposed marching cubes

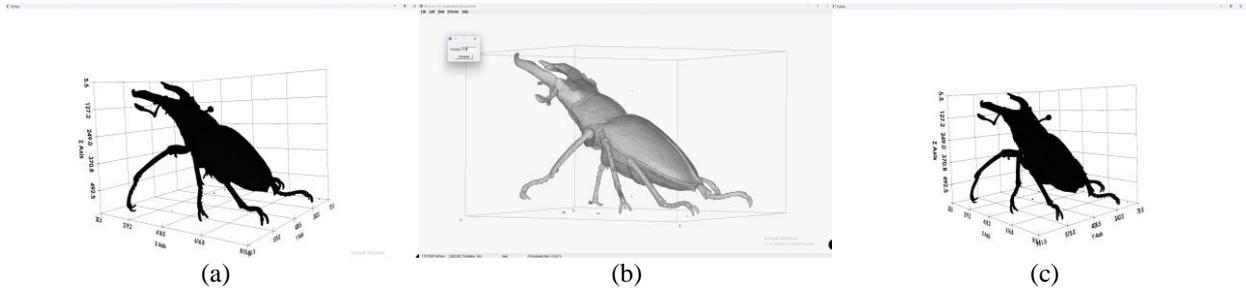


Figure. 15: (a) Surface extracted from the stagbeetle 832x832x494 dataset with traditional marching cubes, (b) with marching cubes 33, and (c) with proposed marching cubes

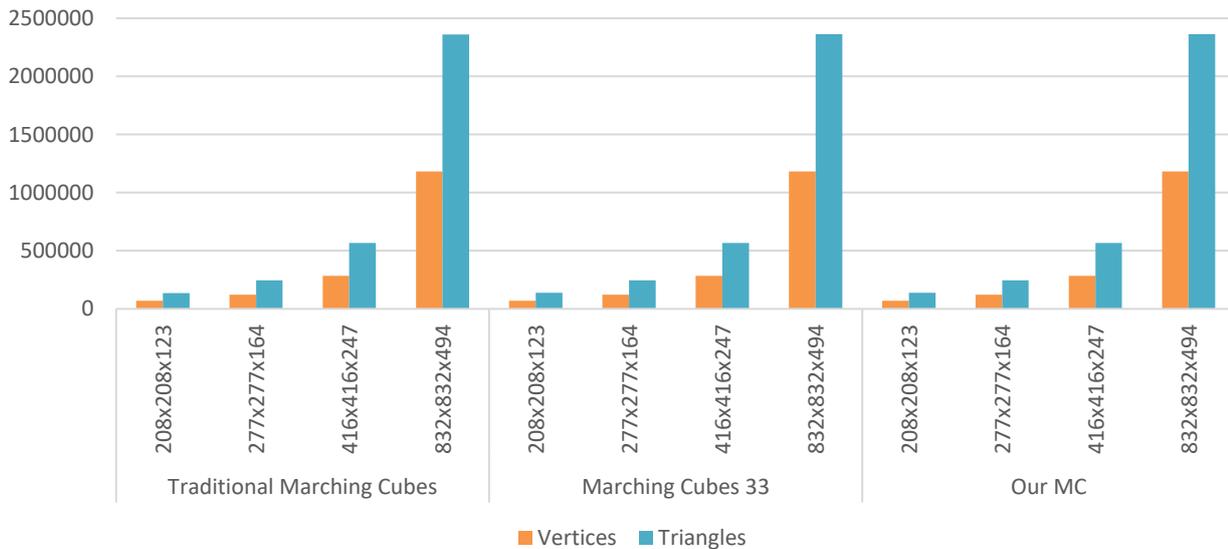


Figure. 16 Comparison of the TMC and MC with the 3D Reconstruction efficiency of our improved algorithm with stagbeetle dataset

5. Results and discussion

5.1 Results of corpus callosum segment detection and preprocessing scheme

The preprocessing methods are used for further processing in the 2D segmentation. The network architecture consisted of six deep learning algorithms, namely Attention_Unet FCN8s myChannelUnet

UNet++ R2UNet and UNet. The best Mean IOU is in architecture FCN8s with 99.8794%, and the best loss is in R2UNet with 0.96%. The second is in UNet with 99.8705%.

5.2 Results of proposed enhanced marching cubes method

A 3D visualization algorithm was implemented in this paper to demonstrate the above algorithms'

effectiveness based on the proposed enhanced marching cubes algorithm. This paper measures the number of faces, points and open edges (hollow) as in Table 5. First, the voxel representations for 3D visualisation were selected; three were used to perform 3D visualisation of the corpus callosum's surface. A comparison of meshes in the first voxel representation was the voxel representation of the traditional MC algorithm, as shown in Figs. 9(a) and 9(c). The second voxel representation was based on the enhanced rules, as shown in Figs. 9(b), 9(d), and 9(e).

The result of the experimental 3D visualisation using the original marching cubes algorithm was compared to that of the enhanced marching cubes algorithm. The number of triangles in corpus callosum mesh $257 \times 257 \times 9$ using the original marching cubes was 5960, while the number of triangles using the marching cubes 33 algorithm was 4430 and using the enhanced marching cubes algorithm was 5980. The result of the experimental 3D visualisation with enhancing marching cubes in other sizes also shows more numbers and vertices. The results are more number vertices and triangles with no ambiguous faces, and our technique can better retain the topology of the trilinear surface while achieving good triangles.

The original marching cubes algorithm produced fewer faces and points, resulting in incomplete 3D visualisation images, as evidenced by open edges (40).

Using the enhanced marching cubes algorithm, the number of points generated was 3924, while the number of faces generated was 7844. Thus, the enhanced marching cubes algorithm produced a 3D visualisation with improved and robust results, as evidenced by the absence of open edges. Advanced marching cubes are a great way to get rid of open edges. Second, this paper measures the surface area, dimensions and volume based on the number of triangles constructed in 3D, as in Tables 6 and 7. The surface area of any given object is the area or region occupied by the object's surface, while the volume is the amount of space available in an object. Area is a two-dimensional object, whereas volume is a three-dimensional object. Area is a planar figure, while volume is a solid figure. Area covers the outer space, and volume covers the inner capacity. Area is measured in square units, and volume is measured in cubic units. Surface area refers to the sum of the areas of all the solid figure's faces.

In contrast, the volume refers to the number of cubic units that form a solid figure. The dimensions are the same, but the volume is different. The empty volume calculates a box's volume; the material's volume is stiff or solid. The resulting numbers were

very far apart.

The corpus callosum and the stag beetle dataset were used to compare with other implementations. The corpus callosum dataset has $257 \times 257 \times 10$ and $257 \times 257 \times 9$ points, and it was obtained from the Local Hospital Soetomo and National Hospital Surabaya archives [25-27]. The stag beetle dataset has $208 \times 208 \times 123$, $277 \times 277 \times 164$, $416 \times 416 \times 247$ and $832 \times 832 \times 494$ points, and it was obtained from the Institut für Computergraphik und Algorithmen [28]. Figs. 11(a), 11(b), (c), 12(a), 12(b), and 12(c) show the isosurface of the corpus callosum dataset. Figs. 13(a), 13(b), 13(c), 14(a), 14(b), and 14(c) shows the isosurface of the stag beetle dataset. Information about enhanced marching cubes and the download link are available at https://github.com/dewirahmawati/marching_algo_test_001.

Tables 8, 9, Figs. 10 and 15 Comparison of the TMC (W. E. Lorensen, 1987) and MC-33 (David Vega, 2019) with the 3D reconstruction efficiency of our improved algorithm with corpus callosum dataset, executed on the desktop computers described above. The code was compiled with Python. It can be seen that there are more vertices and triangles in our implementation.

The result of the experimental 3D visualisation using the original marching cubes algorithm was compared to that of the enhanced marching cubes algorithm. The number of vertices and triangles in stag beetle mesh $832 \times 832 \times 494$ using the original marching cubes were 1181018 and 2359710, while the number of vertices and triangles using the marching cubes 33 algorithm were 1181029 and 2362226 and using the enhanced marching cubes algorithm were 1181018 and 2362208. The result of the experimental 3D visualisation with enhancing marching cubes in other sizes also shows more numbers and vertices. The results are more number vertices and triangles with no ambiguous faces, and our technique can better retain the topology of the trilinear surface while producing high-quality triangles.

6. Conclusion

The result of the experimental 3D visualisation using the original marching cubes algorithm was compared to that of the enhanced marching cubes algorithm. The number of triangles in corpus callosum mesh $257 \times 257 \times 9$ using the original marching cubes was 5960, while the number of triangles using the marching cubes 33 algorithm was 4430 and using the enhanced marching cubes algorithm was 5980. The result of the experimental 3D visualisation using the original marching cubes

algorithm was compared to that of the enhanced marching cubes algorithm. The number of vertices and triangles in stag beetle mesh $832 \times 832 \times 494$ using the original marching cubes were 1181018 and 2359710, while the number of vertices and triangles using the marching cubes 33 algorithm were 1181029 and 2362226 and using the enhanced marching cubes algorithm were 1181018 and 2362208. The result of the experimental 3D visualisation with enhancing marching cubes in other sizes also shows more numbers and vertices. The results are more number vertices and triangles with no ambiguous faces, and our technique can better retain the topology of the trilinear surface while producing high-quality triangles.

In conclusion, this paper presented an enhanced marching cubes algorithm. This study offers an implementation of the enhanced marching cubes algorithm with the following characteristics. The new rules were correctly performed, and the generated surfaces do not have repeated vertices or zero-area triangles.

The application used to test our implementation (enhancing marching cubes). The library source code can be obtained from the website: https://github.com/dewirahmawati/marching_algo_test_001. Better support for translation, scaling, and rotation in real-time has been added to the 3D reconstructed model. The enhanced marching cubes algorithm produces 3D visualisations with improved and robust results, as evidenced by the absence of open edges.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

This study can be completed because of the following research contribution: Conceptualization, Dewi Rahmawati, Riyanarto Sarno and Chastine Fatichah; methodology, Dewi Rahmawati and Riyanarto Sarno; software, Dewi Rahmawati; validation, Dewi Rahmawati; writing—original draft preparation, Dewi Rahmawati; writing—review and editing, Dewi Rahmawati; supervision, Riyanarto Sarno, Chastine Fatichah; Proposes problem ideas, Riyanarto Sarno and Chastine Fatichah.

References

[1] W. E. Lorensen and H. E. Cline, “Marching cubes: A high-resolution 3D surface construction algorithm”, In: *Proc of the 14th Annual Conference on Computer Graphics and*

Interactive Techniques, Aug. 1987, doi: 10.1145/37401.37422.

- [2] M. J. Dürst, “Re”, *ACM SIGGRAPH Computer Graphics*, Vol. 22, No. 5, p. 243, Oct. 1988, doi: 10.1145/378267.378271.
- [3] G. M. Nielson and B. Hamann, “The asymptotic decider: resolving the ambiguity in marching cubes”, In: *Proc. of Visualization '91*, doi: 10.1109/visual.1991.175782.
- [4] B. K. Natarajan, “On generating topologically consistent isosurfaces from uniform samples”, *The Visual Computer*, Vol. 11, No. 1, pp. 52–62, Jan. 1994, doi: 10.1007/bf01900699.
- [5] C. Montani, R. Scateni, and R. Scopigno, “A modified look-up table for implicit disambiguation of Marching Cubes”, *The Visual Computer*, Vol. 10, No. 6, pp. 353–355, Jun. 1994, doi: 10.1007/bf01900830.
- [6] E. Chernyaev, “Marching cubes 33: Construction of topologically correct isosurfaces”, In: *Proc. of Graphicon '95, St. Petersburg, Russian Federation*, 1995.
- [7] P. Bhaniramka, R. Wenger, and R. Crawfis, “Isosurfacing in higher dimensions”, In: *Proc. of Visualization 2000*, 2000, doi: 10.1109/visual.2000.885704.
- [8] G. M. Nielson, “Dual marching cubes”, In: *Proc. of IEEE Visualization 2004*, 2004, doi: 10.1109/visual.2004.28.
- [9] F. Gong and X. Zhao, “Three-Dimensional Reconstruction of Medical Image Based on Improved Marching Cubes Algorithm”, In: *Proc. of 2010 International Conference on Machine Vision and Human-machine Interface*, 2010, doi: 10.1109/mvhi.2010.45.
- [10] L. Guo, M. Hu, Y. Li, W. Yan, and L. Zhao, “Three dimension reconstruction of medical images based on an improved Marching Cubes algorithm”, In: *Proc. of 2013 6th International Conference on Biomedical Engineering and Informatics*, 2013, doi: 10.1109/bmei.2013.6746908.
- [11] M. Wang, H. Luo, and Q. Cui, “Three-Dimensional Reconstruction Based On Improved Marching Cubes Algorithm”, *Journal of Mechanics in Medicine and Biology*, Vol. 20, No. 9, p. 2040002, 2020, doi: 10.1142/s0219519420400023.
- [12] L. Custodio, T. Etienne, S. Pesco, and C. Silva, “Practical considerations on Marching Cubes 33 topological correctness”, *Computers & Graphics*, Vol. 37, No. 7, pp. 840–850, 2013, doi: 10.1016/j.cag.2013.04.004.
- [13] L. Custodio, S. Pesco, and C. Silva, “An extended triangulation to the Marching Cubes 33

- algorithm”, *Journal of the Brazilian Computer Society*, Vol. 25, No. 1, 2019.
- [14] D. Vega, J. Abache, and D. Coll, “A Fast and Memory Saving Marching Cubes 33 Implementation with the Correct Interior Test”, *Journal of Computer Graphics Techniques (JCGT)*, Vol. 8, No. 3, pp. 1-18, 2019 Available online: <http://jcgt.org/published/0008/03/01/>.
- [15] T. Takahashi and C. Batty, “Fast Marching-Cubes-Style Volume Evaluation for Level Set Surfaces”, *Journal of Computer Graphics Techniques*, Vol. 11, No. 2, 2022, Available online: <https://jcgt.org/published/0011/02/02/paper.pdf>
- [16] Y. Lee., J. M. Lee, S. H. Park, Y. J. Choi, S. H. Choi, and J. J. Hwang, “Author Correction: Three-dimensional soft tissue landmark detection with marching cube algorithm”, *Scientific Reports*, Vol. 13, No. 1, 2023, doi: 10.1038/s41598-023-29751-1.
- [17] K. A. Mandaliana, T. Harsono, and R. Sigit, “3D Visualization and Reconstruction of Lung Cancer Images using Marching Cubes Algorithm”, In: *Proc. of 2019 International Electronics Symposium (IES)*, 2019, doi: 10.1109/electsym.2019.8901667.
- [18] P. A. Nugroho, D. K. Basuki, and R. Sigit, “3D heart image reconstruction and visualisation with marching cubes algorithm”, In: *Proc. of 2016 International Conference on Knowledge Creation and Intelligent Computing (KCIC)*, 2016.
- [19] J. Wang, Z. Huang, X. Yang, W. Jia, and T. Zhou, “Three-dimensional Reconstruction of Jaw and Dentition CBCT Images Based on Improved Marching Cubes Algorithm”, *Procedia CIRP*, Vol. 89, pp. 239–244, 2020, doi: 10.1016/j.procir.2020.05.148.
- [20] D. Su and L. Ma, “A Novel 3D Reconstruction Algorithm of CT Images Based on Improved Marching Cubes Algorithm”, In: *Proc. of the 13th EAI International Conference on Mobile Multimedia Communications, Mobimedia 2020*, pp. 27-28, 2020, doi: 10.4108/eai.27-8-2020.2295136.
- [21] Y. Peng, Z. Zhao, Y. Zhao, Z. Wang, J. Li, and H. Zhang, “Three-dimensional reconstruction of magnetic resonance images of carp brain for brain control technology”, *Journal of Neuroscience Methods*, Vol. 366, p. 109428, 2022, doi: 10.1016/j.jneumeth.2021.109428.
- [22] S. Zhang, Y. Hu, and Z. Li, “Machining Simulation Application Based on Improved Marching Cubes Algorithm”, In: *Proc. of 2023 12th International Conference of Information and Communication Technology (ICTech)*, 2023, doi: 10.1109/ictech58362.2023.00048.
- [23] M. Ciecholewski and J. H. Spodnik, “Semi-automatic corpus callosum segmentation and 3D visualisation using active contour methods”, *Symmetry (Basel)*, Vol. 10, No. 11, pp. 1–25, 2018, doi: 10.3390/sym10110589.
- [24] A. Lopes and K. Brodlie, “Improving the robustness and accuracy of the marching cubes algorithm for isosurfacing”, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 9, No. 1, pp. 16–29, Jan. 2003.
- [25] D. Rahmawati, R. Sarno, and C. Fatichah, “A New Hybrid Region-Based Segmentation for 2D Corpus Callosum Segmentation”, In: *Proc. of 2021 5th International Conference on Informatics and Computational Sciences (ICICoS)*, 2021, doi: 10.1109/icicos53627.2021.9651815.
- [26] P. Damayanti, D. Yuniasri, R. Sarno, A. Fajar, and D. Rahmawati, “Corpus callosum segmentation from brain MRI images based on level set method”, In: *Proc. of 2020 Int. Semin. Appl. Technol. Inf. Commun. IT Challenges Sustain. Scalability, Security. Age Digit. Disruption, iSemantic 2020*, pp. 155–160, 2020, doi: 10.1109/iSemantic50169.2020.9234268.
- [27] L. Atikah, N. A. Hasanah, R. Sarno, A. Fajar, and D. Rahmawati, “Brain segmentation using adaptive thresholding, k-means clustering and mathematical morphology in MRI Data”, In: *Proc. of 2020 Int. Semin. Appl. Technol. Inf. Commun. IT Challenges Sustain. Scalability, Security. Age Digit. Disruption, iSemantic 2020*, pp. 161–167, 2020, doi:10.1109/iSemantic50169.2020.9234303.
- [28] M. E. Groller, G. Glaeser, and J. Kastner, 2005. Stag beetle dataset. URL: <http://www.cg.tuwien.ac.at/research/publication/s/2005/dataset-stagbeetle/>. Vol. 7, No. 12, p. 14.
- [29] W. G. Herrera, M. Pereira, M. Bento, A. T. Lapa, S. Appenzeller, and L. Rittner, “A framework for quality control of corpus callosum segmentation in large-scale studies”, *Journal of Neuroscience Methods*, Vol. 334, p. 108593, 2020, doi: 10.1016/j.jneumeth.2020.108593.
- [30] K. S. Manic, R. Biju, W. Patel, M. A. Khan, N. S. M. Raja, and S. Uma, “Extraction and Evaluation of Corpus Callosum from 2D Brain MRI Slice: A Study with Cuckoo Search Algorithm”, *Computational and Mathematical Methods in Medicine*, Vol. 2021, pp. 1–15, 2021, doi: 10.1155/2021/5524637.
- [31] S. C. Satapathy, N. S. M. Raja, V. Rajinikanth, A. S. Ashour, and N. Dey, “Multi-level image

thresholding using Otsu and chaotic bat algorithm”, *Neural Computing and Applications*, Vol. 29, No. 12, pp. 1285–1307, 2016, doi: 10.1007/s00521-016-2645-5.