2021 International Conference on Information and Communication Technology Convergence (ICTC) | 978-1-6654-2383-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICTC52510.2021.9621022

3D Volume Reconstruction from MRI Slices based on VTK

Nodirov Jakhongir[1], Akmalbek Abdusalomov[2] and Taeg Keun Whangbo[*]
Department of IT Convergence Engineering, Gachon University, Sujeong-Gu, Seongnam-Si, Gyeonggi-Do 461-701, Korea;
jakhon@gachon.ac.kr (J.N.), akmaljon@gachon.ac.kr (A.A.)
Correspondence: tkwhangbo@gachon.ac.kr

ABSTRACT – In today's fast-advancing world, Deep learning brought the huge potential to the healthcare system and it still undergoes different amazing new techniques. New automatic brain tumor segmentation models have been realized. As a result, it is being much more affordable and faster to save lives. However, most of the tumor detection works are still being conducted with 2D single slices of brain image, although, there are new 3D CNN [1] models with more benefits. Those 3D models enable to scan of brain images in 3d volume. 2D models accept only single slices as input and they innately fail to use context from neighboring slices. Missed voxel data from contiguous slices might affect the detection of tumors and decrease the accuracy of the model. 3D models address this issue by utilizing 3D convolutional kernels to make predictions from volumetric inputs. The capacity to use interslice features can increase the further performance of the model. Therefore, in practice, 3D volumes enable to obtain much more efficient and clear diagnoses. İn this paper we purpose our new 3D MRI reconstruction algorithm based on VTK toolkit [3].

Keywords: *3D Volume; Brain Image, Deep Learning; Medical Image Reconstruction*

1. INTRODUCTION

Typically, in medicine, medical images were analyzed by observing a number of slices of them to detect any abnormalities for many years. Recent years Deep learning came to main scene. Mainly CNN changed the way doctors do analysis by creating automatic end to end tumor detection algorithms. U Net [2], one of the most popular CNN architecture became popular in
medical field for its ability to extract low level features from medical images and do classification tasks successfully. Decreasing cost of computation power make affordable to use 3D CNN models and analysis three dimensional medical images such as X-ray, Computed Tomography (CT), Diffusion Tensor Imaging (DTI), Positron Emission Tomography (PET), Magnetic Resonance Imaging (MRI), and Functional MRI (fMRI). Major benefits of three-dimensional images are the extra contents between slices and voxel data which can be obtained by processing only 3D images. In this paper, we purpose three-dimensional medical image reconstruction model using VTK tool kit. This model simply takes one of the slices of the medical image and render its 3D version. Moreover, extra tools for visualization and interaction with 3D image are also available such as rotation, zoom.

This paper is organized as follows. Section 2 describe VTK toolkit. Section 3 describe the dataset we used and demonstrates our experimental results and provides discussion. Section 4 provides conclusions and future directions for study.

2.1 VTK Tool Kit

VTK is one of most common an open source, object-oriented system for image visualization and processing tool among computer vision researchers' community. Main use case is in medical image processing especially three-dimensional medical image reconstruction. VTK implements several internal functions like OpenGL Core for rendering 3D volume of medical images. This functionality performs different tasks such as composite, additive

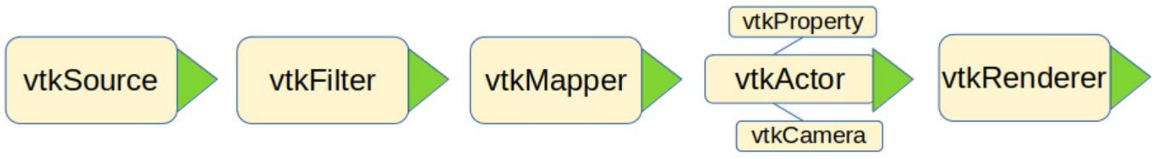


figure 1.1: Basic Visualization of VTK pipeline

blending, maximum and minimum intensity projections, and cropping. The image as input for VTK can be classified into different categories such as polygon data, line grid, unstructured grids, structured point, and unstructured point. During the processing an image with those image classifications, each image type is processed through different filters. The flow of VTK pipeline is as described below in figure 1.1.
vtkObject is one of the classes in VTK library which provides visualized method flow. The derived class of vtkObject that is vtkSource take an input. One of child class of vtk source is vtkFilter. We need it for processing data through filters. Another derived class of vtkObject is vtkMapper. This class process images with different filters and create geometric data that keeps related details between input and processed data. İt is the main function to generate output graphics from input data. And then vtkActor class proccess output of vtkMapper class to visualize. There are also vtkProperty classes which assists vtkActor to make the output more realistic and vtkCamera class which position and orient the view point and focal point. The final process is rendering the output and display the constructed image on window. For this purpose, we use vtkRendered which renders final output and process it to display.

2.2 Marching Cubes Algorithm

Marching cubes [4] are used for creating a grid of triangles to represent the surfaces of an input 3D object represented as a 3D array. The algorithm works by marching the entire image of a 3D object divided into cubes. These cubes are called a voxel. This algorithm checks whether the 3D image intersects the cube and appoints logical values to the angles of the cube correspondingly. If the values at all angles of the cube are equal to 1, the cube lies completely inside the surface. Likewise, if the value of all the angles of the cube is 0, then the cube lies totally outside the surface. The main purpose there is to define the triangles to their value such as some angles of the cube are equal to 1 and the rest are 0. Since the cube has 8 corners (voxels), there are 256 cubes. The final result is acquired by repeated linear interpolation. In our work, Marching cube algorithm was used to show part of the 3D reconstruction from 2-dimensional MRI slices.

2.3 Bilinear Interpolation

Bilinear interpolation [5] is one of the extensions of linear interpolation. We use it to interpolate a function of two variables on a straight 2D framework. The main goal is to perform linear interpolation. It the tool which detects unknown value of the pixel intensity g at the point (x, y) in the 2D framework. The values are assumed to be at four adjacent points (x, y), those are $p_{11}=(x_1, y_1)$, $p_{12}=(x_1, y_2)$, $p2_1=(x_2, y_1)$, and $p_{22}=(x_2, y_2)$. The gray level assigned to g using bilinear interpolation is given by $g(x, y) = a_0 + a_1x + a_2y + a_3xy$, where the four coefficients: a_0, a_1, a_2, a_3, are obtained by solving below formula.

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_1 & y_2 & x_1y_2 \\ 1 & x_2 & y_1 & x_2y_1 \\ 1 & x_2 & y_2 & x_2y_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} g(p_{11}) \\ g(p_{12}) \\ g(p_{21}) \\ g(p_{22}) \end{bmatrix}$$

Bilinear interpolation helps us to regain the lost information between slices which was lost during capturing 2d slices.

2.4 Bicubic Interpolation

Smoother curves can be produced with bicubic interpolation [6] than bilinear interpolation and presents fewer "artifacts" or pixels that significantly degrade the visible quality of the image. With MR images, large amount of noise can be observed along the edges. Bicubic interpolation is a better choice than bilinear interpolation to get straight edges. The bicubic interpolation method with a third-order polynomial function attempts to align the surface between the four corner points. For calculating bicubic interpolation, it is necessary to show the intensity values and the horizontal, vertical, and diagonal products at the four corner points. This is the interpolated surface *S (x, y)* represented by a third-order polynomial:

$$S_{x,y} = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$$

In the above equation, 16 coefficients indicated by a_{ij} and must be determined to compute the function for the interpolated surface. From the intensity values at the four corners all four coefficients are determined directly. Eight of the figures are determined from the spatial derivatives through horizontal and vertical ways, and the remaining four coefficients are determined from the diagonal derivatives. In this work, we used bicubic interpolation along the image edges to precisely reconstruct the missing parts of the edges between the slices. In our case, the four dots are divided into 2D pieces along two adjacent pixels in one piece and two corresponding axes in the next slice.

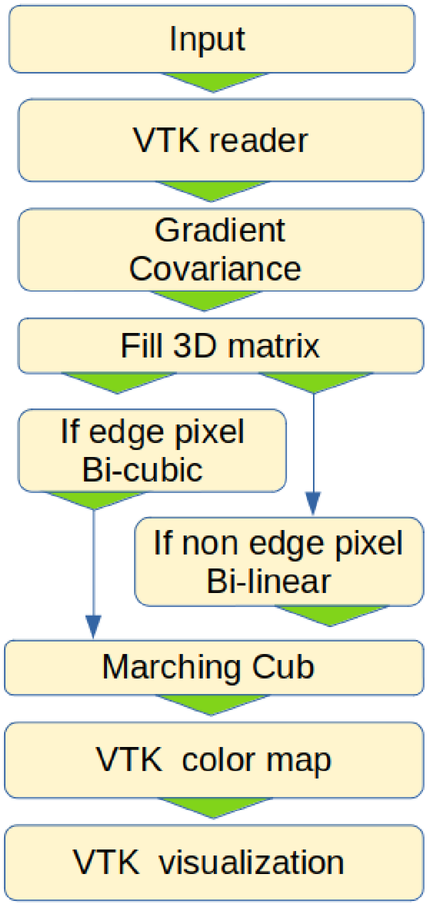
3.1 DATASET

We used Magnetic Resonance Images of Brain which contains 142 slices. İt is in DICOM format that is widely accepted among medicals around the world. Dicom format not only contains image itself but also keeps patient's information. İt is designed by the American College of Radiology and the National Electrical Manufactures Association. Therefore, it is one of the industry standard formats for storing medical data. İn our dataset, size of the slices is 512x512 with scanning interval of 3.2 mm.

3.2 VOLUME RENDERING EXPERIMENT

For the experiment we have used Windows OS. The system configurations were Intel® Core™ i7-9700KF CPU @ 3.60GHz × 8 with RAM of 31.3 GiB, Windows 10 64-bit operating system, VTK (9.0.3). İnput image size was 512x512 with 142 slices.

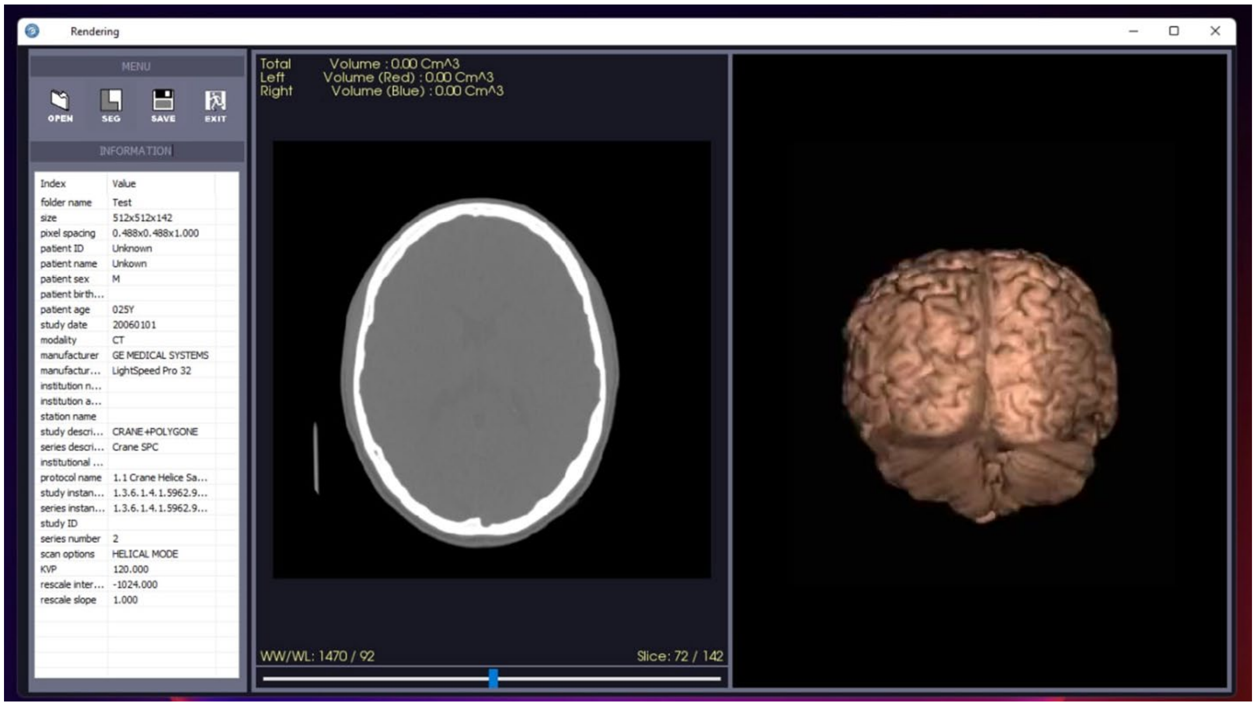
In this study, the proposed image to be reconstructed has size of 512x512x512 that is 3d matrix D(x,y,z) we take all slices into that matrix with 3,2 mm gap between slices. The gradient and covariance of each pixel in each segment was calculated. Our data is in DICOM format so we use vtkDICOMImageReader function to upload it to our model for further processing. Volume Rendering process follows the pipeline in the figure 1.2 below. We have imported our MRI data to our system in format of DICOM using the function of vtkDICOMImageReader. This function process file and vtkImageReader take the output and process. After processing the input and extracting the image data and patient's details, the image data is passed to vtkImageAlgorithm for further process with bilinear and bicubic interpolation. Patient's information is passed to vtkStringArray for displaying on our system window for last stage. The marginal power of each pixel is calculated based on the approximate gradient size $|G|=|G_x|+|G_z|$ is stored in the crown plane and in a separate matrix corresponding to the original matrix. Next, we compare the edge strength of the current pixel with the forces in the positive and negative gradient direction of each pixel. If the edge strength of the current pixel is greatest compared to other pixels in the mask of 8x8 neighbors of the pixel in the same direction and covariance $(x_i, z_i) \sim 0$ for most neighbors, then we apply bicubic interpolation along the y axis in



the original matrix to create the missing pixels,

figure 1.2: Volume Rendering pipeline

otherwise, we use interpolation to create the missing pixels. The process is briefly described in figure 2.1. We used both bicubic with bilinear interpolation, which we proposed for the preservation of minimally noisy edges using bicubic interpolation while we used bilinear interpolation to reconstruct other components using faster way. We take two consecutive pieces of slices and reach the top layer of the 3D matrix in the x -z plane. Next, we break down the 3D matrix in voxels and apply the marching cube algorithm for more accurate 3D reconstruction. For visualization, we flatten the image using VTK tools then apply a color map and activate the conversion process. Finally, the output can be visualized in our system. There are three windows as shown in figure 2.2 above. On the first left window detailed information of the patients will be printed, middle one is for visualization of single 2D image and the right one is for visualization of constructed 3D image.



Index	Value
folder name	Test
size	512x512x142
pixel spacing	0.488x0.488x1.000
patient ID	Unknown
patient name	Unkown
patient sex	M
patient birth…	
patient age	025Y
study date	20060101
modality	CT
manufacturer	GE MEDICAL SYSTEMS
manufactur…	LightSpeed Pro 32
institution n…	
institution a…	
station name	
study descri…	CRANE+POLYGONE
series descri…	Crane SPC
institutional …	
protocol name	1.1 Crane Helice Sa…
study instan…	1.3.6.1.4.1.5962.9…
series instan…	1.3.6.1.4.1.5962.9…
study ID	
series number	2
scan options	HELICAL MODE
KVP	120.000
rescale inter…	-1024.000
rescale slope	1.000

```
Total      Volume : 0.00 Cm^3
Left      Volume (Red) : 0.00 Cm^3
Right      Volume (Blue) : 0.00 Cm^3
```

```
WW/WL: 1470 / 92                    Slice: 72 / 142
```

figure 2.1: VTK based system

On top of left window, you can see the tools to help user to interact with the system. From left side of the window the first tool is used to import an input, second tool reconstruct the 3D image and then you can save the result with next tool. İn figure 2.1, on the right window you can see our reconstructed 3D image from MRI DICOM slices. You can save the 3D volume as a file and process for any research. The output size of 3D volume is 512x512x152.

4. CONCLUSION

In this paper we created the system which takes slices of MRI as an input in DICOM format. The system processes the series of slices with the VTK tools and finally output 3-dimensional Magnetic Resonance Image. The output can be used for different deep learning tasks such as tumor segmentation and classification or directly used to visualize and observe it if there is anything to consider.

The advent of technology especially computing powers enabled more accessible to work with 3D volume data. The accuracy of 3D data certainly higher than 2 dimensional one. Therefore, the there is an increasing demand for 3-dimensional data. Our method can help to meet those demand as much as possible. For further development, we will do deeper research on 3D image reconstruction and try to apply different other methods such as deep learning and machine learning techniques.

ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2017-0-01630) supervised by the IITP (Institute for Information & communications Technology Promotion)

REFERENCES

[1] S.P.S., L.W. 3D Deep Learning on Medical Images: A Review. 3 September 2020;

[2] O.R., P.F. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015

[3] https://vtk.org/documentation/

[4] L.W.E., C. H.E.: ‘ Marching cubes: a high resolution 3d surface construction, 1987

[5] Gao S., Gruev V.: ‘ Bilinear and bicubic interpolation methods for division of focal plane polarimeters’, Opt. Express, 2011

[6] Keys R.: ‘Cubic convolution interpolation for digital image processing’, 1981