

# A Software tool for 3D visualization and slicing of MR images

Somoballi Ghoshal  
University of Calcutta  
India  
somoballi@gmail.com

Pubali Chatterjee  
University of Calcutta  
India  
pubali88@gamil.com

Sourav Banu  
University of Calcutta  
India  
soumou.banu@gmail.com

Prof. Amlan Chakrabarti  
University of Calcutta  
India  
acakcs@caluniv.ac.in

Assoc. Prof. Eleni Mangina  
University College Dublin  
Ireland  
eleni.mangina@ucd.ie

## ABSTRACT

In case of Magnetic resonance imaging (MRI), the image of an object is taken in all 3 directions. But, all the analysis and study based on MR images is carried out on 2D data, more precisely taking a single view at a time rather than considering all 3 views. Hence, the analysis is not always accurate. A solution is to create a 3D figure that will include all three views in it, so that the doctor can see any view as per his/her wish with just one click using virtual scissors. We have developed a tool that will generate the 3D from a set of MR images in a single plane and from that we can slice out along any axis to get the view from different angle in different planes as per our wish using virtual scissors. The results show that the slices generated after reconstruction are very close to the ground truth images.

## CCS CONCEPTS

**Software and its engineering** → **Virtual World Software**; Virtual World Training Simulations; **Human Centered Computing** → Interactive systems and tools; General Terms: Design, Algorithms, Visualization

## KEYWORDS

Virtual Reality, Augmented Reality, 3D modeling, 3D reconstruction

## ACM Reference format:

S. Ghoshal, P. Chatterjee, S. Banu, A. Chakrabarti and E. Mangina 2017. A Software tool for 3D visualization and slicing of MR images. 2017. In *Proceedings of ACM Simutools conference, Hong Kong, China September 2017 (SIMUTOOLS 2017)*, 5 pages. <https://doi.org/10.1145/3173519.3173527>

\* Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Requestpermissions from Permissions@acm.org. SIMUTOOLS '17, September 11–13, 2017, Hong Kong, China © 2017 Association for Computing Machinery. ACM ISBN 978-1-4503-6388-4/17/09...\$15.00 <https://doi.org/10.1145/3173519.3173527>

## 1 INTRODUCTION

While MR image is captured in three directions, but the analysis on it is carried out on a two directional data at a time. The data that we get at hand is 2D, as the image is captured in 3 different planes - transverse, coronal and sagittal, in 2D. The 2D analysis of MR images is not always accurate. The slices in each plane are not continuous as there always exist a gap between the slices while capturing. A solution to this can be to create figures that include proper interpretation and can emanate 3D information. Such figures can be created and produced using standard functional magnetic resonance imaging (fMRI) analysis packages, such as SPM [1], AFNI [2] with SUMA, and FreeSurfer [3], as well as some more specialty programs, such as MRICroGL [4], 3DSlicer [5] and the Mango [6]. While these several options helps in 3D rendering of MRI data, but the internal structures and functionalities are usually lost, and moreover, all these above mentioned procedures uses all three set of views generated during MRI scan for 3D reconstruction. During the MRI process the patient is subjected to tremendous noise, which affects the nervous system. Hence if the time can be reduced for MRI then the exposure to this noise will also be reduced. In an MRI centre if the time can be reduced, hence more patients can be dealt in a day, which will decrease the cost of imaging. The brain MR images have specific characteristics having complicated changes of the gray-scales and highly irregular boundaries.

Marching cubes [13] is the most used algorithm in medical image 3D surface construction, being an easy and widely used algorithm. The algorithm creates “triangle models of constant density surfaces from 3D-medical data” by dividing the region of interest into cubes and then creating triangles for which the normal to the surface at each vertex of each triangle are calculated. It iterates over a grid of cubes that are superimposed over a region on a function and if all 8 vertices of the cube are below or above then they are not discarded and are considered part of the surface otherwise some vertices and triangles are generated. With 8 vertices there are 256 ways a surface can intersect the cube. A lookup table is generated for surface-edge intersections. Triangulation for each case has proven to be either a rotation or

mirror of 15 unique patterns, and therefore reduces the number of required triangulation computation. An index is given to each case, which will state which edge the surface intersects, then a linear interpolation is used along the edge of the surface intersection. It uses information from the original data to determine inter-slice (real-world depth) connectivity between each slice, surface location, and surface gradient. The resulting triangle model can be displayed using standard rendering algorithms.

Traditional 3-D reconstruction algorithms [7], [8], [9], [11] are not capable of solving this problem. Previously, the realization of 3D visualization was mainly by two reconstruction methods, surface rendering and volume rendering [10]. The problem of surface rendering is that if the image is not properly captured then it gives erroneous results. Volume rendering [10] is the most commonly used approach for 3D reconstruction as both volume and surface are preserved.

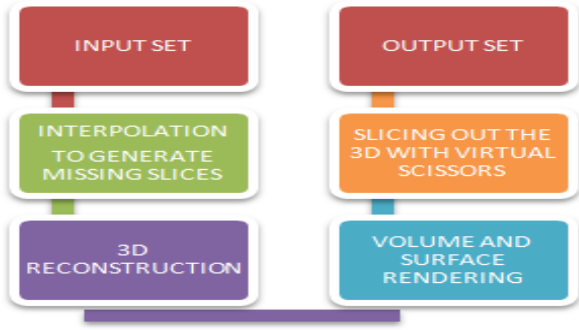


Figure 1: Block diagram of the proposed method

## 2 METHOD

There are software that generate the 3D of MRI brain, taking into account the slices of all three planes. It allows the doctor to view slices as per his/her wish. But this is time consuming, our goal is to reduce the time and cost of MR imaging. With the development of 3D volume image from images of a single plane and generating all possible slices from it, we can achieve a reduction in reconstruction without compromising the quality of visualization. This visualization will provide assistance for medical diagnostics and surgery at a low cost [12].

Normal MRI in India now-a-days takes about 25-30 mins time whereas with this software being installed only 5-6 mins will be required for a MRI scan i.e up to 80%.

Fig. 1 shows the block diagram of our proposed method. First we, concatenate all the 2D arrays corresponding to the 2D aligned slices as 3D array. Then, we apply bilinear interpolation to generate intermediate 3 slices between every two corresponding slices.

In mathematics, bilinear interpolation [14] is an extension of linear interpolation for interpolating functions of two variables (e.g.,  $x$  and  $y$ ) on a rectilinear 2D grid. The key idea is to perform linear interpolation first in one direction, and then again in the other direction. Suppose that we want to find the value of the unknown function  $f$  at the point  $(x, y)$ . It is assumed that we know the value of  $f$  at the four points  $Q_{11} = (x_1, y_1)$ ;  $Q_{12} = (x_1, y_2)$ ;  $Q_{21} = (x_2, y_1)$ ; and  $Q_{22} = (x_2, y_2)$ . For bilinear interpolation assigned gray label is given by

$$f(x, y) = a_0 + a_1x + a_2y + a_3xy \quad (1)$$

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_1 & y_2 & x_1y_2 \\ 1 & x_2 & y_1 & x_2y_1 \\ 1 & x_2 & y_2 & x_2y_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(Q_{11}) \\ f(Q_{12}) \\ f(Q_{21}) \\ f(Q_{22}) \end{bmatrix} \quad (2)$$

Now, we squeeze the generated 3D array. Smoothing algorithm using box filter is applied on the squeezed array. Iso-surface and iso-caps are calculated for patching the image. Then, we set the aspect ratio and add color to the image using Color map. Now, we set the parameters for rotation and display the 3D MRI image. This 3D MRI image is fed as input and a doctor using virtual scissors can cut through it to get the 2D image along any axis as per his/her wish. The user can also generate all possible slices along all 3 axes by specifying the distance between slices.

---

### ALGORITHM 1: 3D visualisation

---

*Input: The slices (x plane) of MRI Ma*

*Output: 3D volumetric representation of MRI data.*

*/\*Reconstruction\*/*

*begin*

*Step 1: T=Concatenate(Ma)*

*Step 2: T=Bilinear\_interpolation(Ma)*

*Step 3: T=Squeeze(T).*

*Step 4: T=Smooth(T)*

*Step 5: MI = Iso\_surface(T) and MC=iso\_caps(T)*

*Step 6: A=aspect\_ratio(T).*

*Step 7: T= Colormap(T).*

*Step 8: Rotate\_para(T).*

*Step 9: Display(T)*

*/\*Slicing\*/*

*Step 10 [m,n,p]=Getdimension(T)*

*Step 11: while j <= m                    j is the gap between two slices*  
*begin*

*Step 12: M1 = c(j, :, :).*

*Step 13: M1=squeeze(M1)*

*Step 14: M1=imresize(M1,[400 256])*

*Step 15: M1 = imrotate(M1,90)*

*Step 16: Imshow(M1)*

*Step 17: j=j+3;*

*end*

*Step 18: Repeat steps 11-17 for n and p to get the slices in the other two planes to get all required slices*

**end**

---

### 3 RESULTS

We have used MATLAB dataset and BRAINWEB dataset to test our method. Fig.2 shows the set of input axial slices with slice gap= 5mm and Fig. 3 shows the 3D output corresponding to the input. Fig 4 shows the GUI in which we can select the plane along which we want to generate the slices, the path where we want to store the images after slicing and the gap between two slices. After specifying all these, we have to click on Check button and the slices will be stored in the specified path. Fig. 5 shows the slices along Z-axis with slice gap = 3 mm for the given input along X-axis as in Fig.2. After clicking on Check button in Fig. 4 the screen in Fig.5 appears, which asks the user whether he/she wants to view all slices or any particular slice. If the user views all slices, then the output appears as shown in Fig.7 and if he/she wants to view a particular slice then the output is as shown in Fig.8. Fig. 9 shows the output when a selected number of continuous slices are to be viewed. Fig 10 shows the ground truth images and the sliced images after reconstruction corresponding to them and the statistical measure for similarity is given in Table 1 when MATLAB's data was used as ground truth.

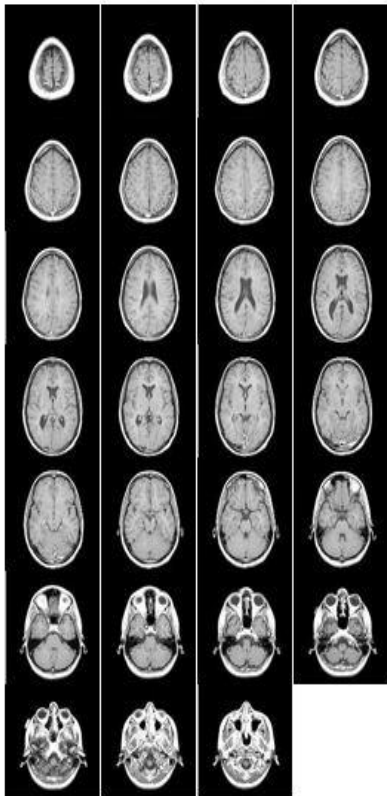


Figure 2: Input Axial Slices

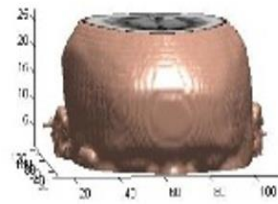


Figure 3: 3D reconstructed output from input as shown in Fig.1



Figure 4: GUI of tool for slicing along a specified axis

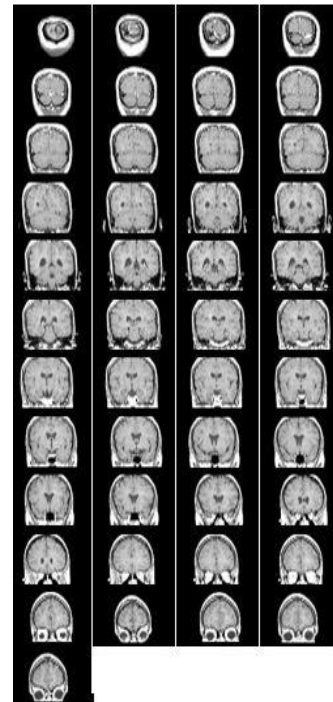


Figure 5: Output after slicing Fig. 2 along Z-axis with slice gap=3mm

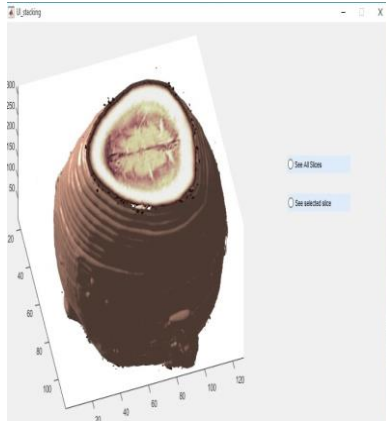


Figure 6: GUI screen to select whether to view all slices or particular slices



Figure 7: Output when all slices are to be viewed along X-axis with slice gap= 1 mm

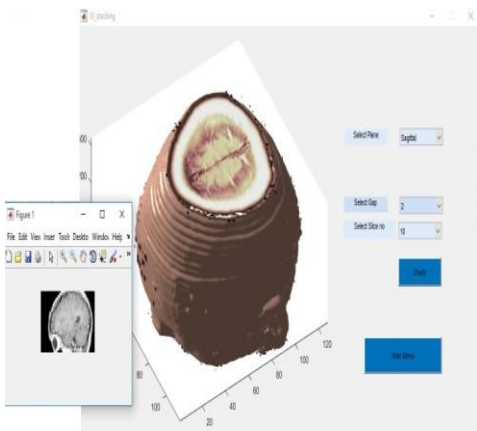


Figure 8: Output when a particular slice is viewed along Y-axis with slice gap= 2 mm GUI screen to select whether to view all slices or particular slices

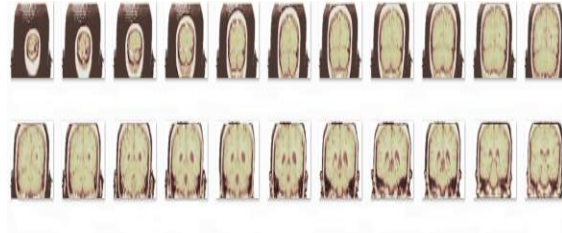


Fig.9 Output when a particular set of slices is viewed along Z-axis with slice gap= 1 mm and number of slices= 22

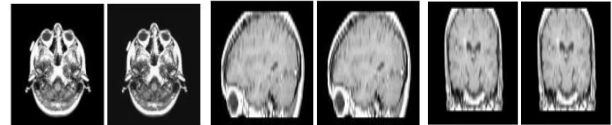


Fig.10 From left to right: Ground truth axial slice, corresponding sliced axial slice. Ground truth sagittal slice, corresponding sliced sagittal slice. Ground truth coronal slice, corresponding sliced coronal slice.

Table 1: Similarity measure between sliced image after reconstruction and ground truth image

Slice	Entropy	Mutual Information
Axial	0.0919	6.9075
Sagittal	0.1679	6.9006
Coronal	0.0659	7.0003

## 4 CONCLUSIONS

We have designed a tool for generating all 3 set of slices of MR image from a single set of slices in a plane, thus reducing the time and cost for diagnosis. We have validated our results using entropy and mutual information for the slices obtained from MATLAB's data. This tool can be effectively used for training data for 3D visualization at a faster rate too. We further plan to segment out tumor and estimating its volume, shape, size and exact location through 3D reconstruction for proper surgical planning.

## ACKNOWLEDGMENTS

This work is supported by UGC UPE-II Project of University of Calcutta.

## REFERENCES

- [1] SPM, <http://www.fil.ion.ucl.ac.uk/spm/>
- [2] AFNI with SUMA, <http://afni.nimh.nih.gov>
- [3] FreeSurfer, <https://surfer.nmr.mgh.harvard.edu>
- [4] MRICroGI, <http://www.mccauslandcenter.sc.edu/mricrogi>
- [5] 3DSlicer, <http://www.slicer.org>
- [6] Madan CR. Creating 3D visualizations of MRI data: A brief guide. F1000Research. 2015;4:466. doi:10.12688/f1000research.6838.1.
- [7] Cheng-Sheng Hung, Chien-Feng Huang, Ming Ouhyoung, Fast Volume

Rendering for Medical Image Data, Communication and Multimedia Laboratory Department of Computer Science and Information Engineering National Taiwan University, Taiwan.

- [8] Lei Wang, Guizhi Xu, Lei Guo, Xuena Liu and Shuo Yang, "3D Reconstruction of Head MRI Based on One Class Support Vector Machine with Immune algorithm", Proceedings of the 29th Annual International Conference of the IEEE EMBS Cité International, Lyon, France August 23- 26, 2007.
- [9] Paul Hergehelegiu, Marius Gavrilescu, Vasile Manta, "Visualization of Segmented Structures in 3D Multimodal Medical Data Sets", Advances in Electrical and Computer Engineering, Volume 11, No. 3, 2011
- [10] Pavarino, E., Neves, L. A., Machado, J. M., Godoy, M. F., Shiyu, Momente, Y. C M. Borse , S.B. Patil, B.S.Patil, "Literature survey for 3d reconstruction of brain mri images", IJRET Volume: 02 Issue: 11,Nov-2013, Available @ <http://www.ijret.org>
- [11] Manjunath BS, Chellappa R. A feature based approach to face recognition. In Proceedings of IEEE conference on computer vision and pattern recognition, Champaign, IL, 1992.p. 373-8.
- [12] Kirchberg, K. J., Wimmer, A., & Lorenz, C. H. (2006). Modeling the Human Aorta for MR-Driven RealTime Virtual Endoscopy. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006, 470-477 Springer Berlin Heidelberg.
- [13] Lorensen, W. E., & Cline, H. E. (1987). MARCHING CUBES: A HIGH RESOLUTION. In ACM siggraph computer graphics, (Vol. 21, No. 4, pp. 163-169).
- [14] Definicition of Bilinear Interpolation, The Computer Language Inc., <http://www.pcmag.com/encyclopedia/term/38607/bilinear-interpolation>, Last accessed August 2017.