1.7

IET Image Processing

Research Article

3D reconstruction of spine image from 2D MRI slices along one axis

ISSN 1751-9659
Received on 26th June 2019
Revised 31st March 2020
Accepted on 28th April 2020
E-First on 8th September 2020
doi: 10.1049/iet-ipr.2019.0800
www.ietdl.org

Somoballi Ghoshal[1] ✉, *Sourav Banu*[1], *Amlan Chakrabarti*[1], *Susmita Sur-Kolay*[2], *Alok Pandit*[3]
[1]*A. K. Choudhury School of Information Technology, University of Calcutta, Kolkata, India*
[2]*A.C.M.U., Indian Statistical Institute, Kolkata, India*
[3]*Bangur Institute of Neurosciences, Kolkata, India*
✉ *E-mail: somoballi@gmail.com*

Abstract: Magnetic resonance imaging (MRI) is a very effective method for identifying any abnormality in the structure and physiology of the spine. However, MRI is time consuming as well as costly. In this work, the authors propose an algorithm which can reduce the time of MRI and thus the cost, with minimal compromise on accuracy. They reconstruct a three-dimensional (3D) image of the spine from a sequence of 2D MRI slices along any one axis with reasonable slice gap. In order to preserve the image at the edges properly, they regenerate the 3D image by using a combination of bicubic and bilinear interpolation along the orthogonal axis. From the reconstructed 3D, they use a simple geometric method to slice out any possible location along any axis and get the information in that region. They have tested their algorithm on real data, and found that their algorithm reduces the time by 80%, with high internal data preservation accuracy of about 96%.

1 Introduction

Medical imaging is the method of capturing the visual orientation of the interior of a specimen, and also the visualisation of the function of some of its organs or tissues [1]. This technique can be used for various purposes such as to diagnose patients, to analyse fossils and so on.

The commonly used methods incorporate radiology, which uses the imaging technologies of X-ray radiography, computer aided tomography (CT), magnetic resonance imaging (MRI), medical ultrasonography or ultrasound, endoscopy, elastography, tactile imaging, thermography, medical photography. In addition, there are also nuclear medicine based functional imaging techniques such as positron emission tomography and single-photon emission computed tomography.

MRI is a medical imaging technique primarily used in radiology to form pictures of the anatomy and the physiological processes of the body in both health and disease. It is widely used for diagnosis of any brain and spine abnormality/disease [2] because it captures the tissue structure of the body most effectively. The diversity and complexity of lesion cells, particularly in functionally critical organs, make it very challenging to visualise a lesion in MRI [3].

MRI images are typically captured in 2D. For accurate medical diagnosis of three-dimensional (3D) lesions, 3D visualisation [4–7] can facilitate surgeons to decide on the portion of the tumour/lesion to be removed.

There are two types of 3D reconstruction techniques [6, 8], namely, surface rendering and volume rendering. In case of surface rendering [8], the surface of the 3D object is reconstructed and preserved. In case of volume rendering [6, 8–10], the internal structure of the 3D object is also reconstructed to visualise a volumetric view. Thus, when this 3D data is sliced into 2D we also get to view the internal structures of the 3D object also. With the development of image processing technology, 3D visualisation has become an important method for the medical diagnosis [11], as it offers abundant and accurate information for medical experts.

2 Motivation

At present, there are several techniques for 3D MRI scan, but these are mostly used for brain imaging. Since the area of concern for spine is large, 3D image capturing of spine would be far more time consuming and noise prone due to any slightest movement of the subject during the long period of imaging. During the MRI, a patient is also subjected to deafening noise and it takes about 30–45 min to perform a whole spine MRI [3]. Thus, for MRI of the spine, three sets of 2D axial, sagittal and coronal slices, are available to the medical experts as shown in Fig. 1.

There is a uniform gap ranging from 1 to 5 mm between two consecutive slices of an MRI. Hence, our goal is to create efficient accurate 3D reconstruction and visualisation of MRI of the spine from a single sequence of 2D slices, and also providing an user interface for the surgeons to cut that reconstructed 3D image as needed with virtual scissors and to view any slice in any of the other planes. Thus, the scan time will be reduced from 45 min [12] to around 6 min (ideally the time taken to capture a single set of slices along a specified axis). Since the time is proportional to cost, the huge cost of MRI, especially in the developing countries, can also be reduced and better health-care can be attained. This would also facilitate exploratory diagnosis easily.

The human spine MR images have unique characteristics [13–18]: it has broadly three fine layers: dura mater, arachnoid mater and spinal cord which are very similar and hard to distinguish. Spine is a long structure with different shape and features for each vertebra, therefore posing a challenge to the traditional 3D reconstruction algorithms. In [19], we can find 3D reconstruction and slicing of brain. In this work, we focus on that problem for spine MRI. Since the spine covers a larger area, and both hard and soft tissues are of equal importance, the method for brain is not directly applicable to the features of MRI spine.

In our work, we have

- reconstructed the human spine from the 2D MRI slices taking into account only single sequence of the MRI slices with an inter slice gap of 3–5 mm, either sagittal, axial or coronal, by applying a combination of bilinear and bicubic interpolation to reconstruct the missing slices (the gap) in between consecutive slices.
- sliced this reconstructed 3D as per user request to view the internal structure accurately

To the best of our knowledge, this is a first of its kind technique for 3D reconstruction of MRI of spine from a sequence of slices in one plane and slicing this regenerated 3D as per user instructions for viewing the internal structure.

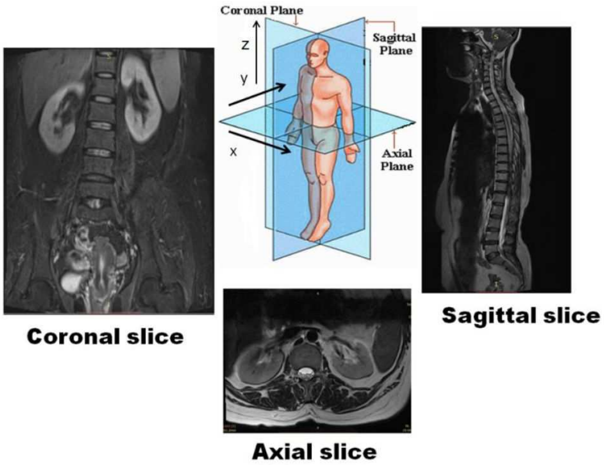

Fig. 1 *Anatomical planes of a human body as captured during MRI*

3 Preliminaries

In this section, we present the fundamentals of interpolation techniques, which have been used in our proposed methodology. Then, the metrics for image similarity metrics are briefly discussed which have been used to validate and establish our results.

3.1 Interpolation

Interpolation is the process of generating missing data points within a given data range. We have used a combination of three interpolation techniques for generating the missing points between consecutive slices. The techniques are discussed as follows.

3.1.1 Bilinear interpolation: Bilinear interpolation [20] is an extension of linear interpolation. It is used for interpolating a function of two variables (say, x and y) on a rectilinear 2D grid. The principle objective is to perform linear interpolation first in one variable, and then in the other one. Let us determine the unknown intensity value of pixel g at the point (x, y) in the 2D grid. It is assumed that the values at the the four neighbouring points of (x,y), namely $p_{11} = (x_1, y_1), p_{12} = (x_1, y_2), p_{21} = (x_2, y_1)$ and $p_{22} = (x_2, y_2)$ are known. The grey level assigned to g by bilinear interpolation is given by (1), where the four coefficients a_0, a_1, a_2, a_3 are obtained by solving (2)

$$g(x, y) = a_0 + a_1 x + a_2 y + a_3 xy \tag{1}$$

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1 y_1 \\ 1 & x_1 & y_2 & x_1 y_2 \\ 1 & x_2 & y_1 & x_2 y_1 \\ 1 & x_2 & y_2 & x_2 y_2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} g(p_{11}) \\ g(p_{12}) \\ g(p_{21}) \\ g(p_{22}) \end{bmatrix} \tag{2}$$

We have used bilinear interpolation in our proposed algorithm to reconstruct the slices in between the consecutive slices in the given sequence, which are missing due to the gap between slices while capturing the images.

3.1.2 Bicubic interpolation: Bicubic interpolation [20, 21] creates smoother curves than bilinear interpolation, and introduces fewer 'artefacts', or pixels that stand out and conspicuously deteriorate the apparent quality of the image. In case of MR images, there is a significant amount of noise along the edges. In order to get smoother edges, bicubic interpolation is a better choice over bilinear interpolation. The bicubic interpolation method attempts to fit a surface between four corner points using a third-order polynomial function. In order to compute a bicubic interpolation, the intensity values and the horizontal, vertical and diagonal derivatives at the four corner points need to be specified. The interpolated surface, $S(x, y)$, described by third-order polynomial is as follows:

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

There are 16 coefficients specified by a_{ij} that need to be determined in order to compute the function for the interpolated surface in the above equation. Four of the coefficients are determined directly from the intensity values in the four corners; eight of the coefficients are obtained from spatial derivatives in the horizontal and vertical directions, and the remaining four coefficients are determined from diagonal derivatives. In our proposed algorithm, we have used bicubic interpolation along the edges of an image for accurate reconstruction of missing portions of edges in between slices. The four points in our case happen to be two neighbouring pixels in a slice and the corresponding two in the following slice in the sequence of 2D slices along an axis.

3.1.3 Marching cubes algorithm: The Marching cubes [22] is a simple iterative algorithm for creating a mesh of triangles to represent the surfaces for a given 3D object specified as a 3D array. The algorithm works by 'marching' over the entire image of the 3D object which has been equally sub-divided into cubes. Each cube is called a voxel. The algorithm then determines whether the 3D image intersects a cube, and assigns boolean values to the corners of the cube accordingly. Intuitively, suppose the values at all the corners of the cube (i.e. the voxel) are 1. Then the cube is said to lie entirely inside the surface. Similarly, if all the corners of the cube have value 0, then the cube is said to lie entirely outside the surface. In both the cases, there would be no triangular surface passing through the cube. The main aim of the algorithm is to determine triangles (its intersection points, normals) in the cases where some corners of a cube are 1 and the others are 0. As there are 8 corners in a cube (voxel), there are 256 cube configurations which are stored in a look-up table. Then the final mesh is obtained through iterative linear interpolation. We have used Marching cube algorithm [22] for surface rendering part of the 3D reconstruction from the 2D slices of MRI.

3.2 Similarity metrics for images

The definitions of four most popular similarity metrics for images that we have used for validating our results are presented next.

3.2.1 Root mean square error (RMSE): The RMSE [23, 24] is a frequently used measure of the differences between values predicted by an estimator and the values observed. It is the square root of the average of the square of the errors. RMSE of an images $f_1(x, y)$ with respect to an image $f_2(x, y)$ is defined as the square root of the mean square error (MSE) [24]

$$\text{MSE} = \frac{1}{MN} \sum_{n=1}^{M} \sum_{m=1}^{N} [f_2(n, m) - f_1(n, m)]^2 \tag{4}$$

where $M \times N$ is the size of the image matrix. Thus, the RMSE= $\sqrt{\text{MSE}}$.

If RMSE is close to 0, the probability of the two images being identical is higher. We have used RMSE to compare our sliced results with the ground truth images.

3.2.2 Mutual information: Mutual information [24, 25] is a quantitative measurement of information about one random variable (Y) with respect to another random variable (X). However, information is a reduction in the uncertainty of a variable. So, the higher is the mutual information between X and Y, the lower is the uncertainty of X given Y or vice versa. Let G and R be the ground truth and reconstructed images, respectively. The mutual information MI_{GR} between them is defined as

$$MI_{GR} = \sum_{g,r} p_{G,R}(g, r) \log \frac{p_{GR}(g, r)}{p_G(g) p_R(r)} \tag{5}$$

where $p_{G,R}$ is the jointly normalised histogram of G and R, p_G and p_R are the normalised histograms of G and R, and g, r represent the pixel value of image G and image R, respectively.

A large value of *MI* indicates accurate reconstruction. Mutual information has been used for checking accuracy of the generated missing data, compared to the available ground truth data. The greater is the mutual information between a generated slice and the original MRI slice, the better is the accuracy of our algorithm.

3.2.3 Structural similarity index method (SSIM): SSIM [26] is used for measuring the similarity between two images. The measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference. The SSIM index between two images X and Y is obtained as

$$\mathrm{SSIM}(X,Y)=\frac{(2\,\mu_X\mu_Y+c_1)(2\sigma_{XY}+c_2)}{(\mu_X^2+\mu_Y^2+c_1)(\sigma_X^2+\sigma_Y^2+c_2)} \qquad (6)$$

where μ_X and μ_Y are the mean of X and Y, respectively, σ_X^2 and σ_Y^2 are the standard deviation of X and Y, respectively, σ_{XY} is the covariance of X and Y, and c_1 and c_2 are two constants which stabilise the ratio with a weak denominator. If *SSIM* is nearly equal to 1, then the two images can be considered to be identical. We have used *SSIM* to compare our sliced results with the ground truth images.

4 Related works

There are several works for 3D reconstruction from 2D images using CT images [27, 28], but they fail to give good accurate results for MR images. In CT images, most of the information being for hard tissue, geometric features and edges are more prominent than in the case of MRI images. Thus, these methods do not accurately generate the information of the missing planes if the gap between two consecutive slices is more than 1 mm.

The Marching cubes [22] algorithm is widely accepted for reconstructing a 3D surface from a given 3D image. For approximating contours, it uses patterned cubes or isosurfaces. However, it requires certain techniques to reduce memory and time for reconstructing a surface from large volumetric data. The usual way to solve this problem [29] is by diminishing the size of a volumetric image, but the quality of the surface of 3D reconstructed image becomes substandard if only sub-sampling is applied. Due to poor reconstruction by only volumetric sub-sampling, another method is proposed which improves the quality of a surface reconstructed from the sampled volumetric data. It is based on a pipeline of Visualization Toolkit (VTK) [30, 31]. They used an approach that consists of three major steps: preprocessing, reconstructing and displaying. In [31], the preprocessing steps focused on thresholding, sampling and Gaussian filtering. In particular, the standard deviation parameters for Gaussian filtering and the effect of the sub-sampling factors were studied. Further, memory and time utilisation were considered in this research as well. In [32], the authors have used tri-linear interpolation for reconstruction of MRI of brain and it gave better results than marching cube, but again fails in preserving the minute internal details. In [33], the authors have used edge based interpolation for correction of blurred and noisy edges in a 2D plane, but this work is inadequate for generating a large number of missing points in a 3D image. For 3D reconstruction by this technique applied on our real data, the accuracy after slicing is obtained to be 79.62%.

5 Proposed methodology

5.1 3D reconstruction

In this work, at first we have carried out de-noising of the slices using Shearlet transform [34], a recent widely accepted technique for de-noising of MRI images. We worked with the slices along the coronal plane (*x,z*), assuming that the slices are already registered with each other. The 3D image to be reconstructed is represented by a 3D matrix $\boldsymbol{D}(i,j,k)$, which has a typical size of $512\times512\times512$ since 2D slices are of 512×512 pixels. The length of an adult spine is around 450 mm. Initially, it is partially filled with the available data for 2D slices taken at a gap of 5 mm or 3 mm. For example, if the source data is for 5 mm gap, then the matrix $\boldsymbol{D}$ is filled for values $j=1,6,11,\ldots446$. We calculate the gradient and covariance of each pixel in each slice. The edge strength of each pixel is calculated based on the approximate absolute gradient magnitude $|G|$=$|G_x|$ + $|G_z|$ in the coronal plane and is stored in a separate matrix corresponding to the original matrix. Next, we compare the edge strength of the current pixel with that of each of the pixels in the positive and negative gradient directions. If the edge strength of the current pixel is the largest compared to the other pixels in the mask of 8×8 neighbours of the pixel with the same direction, and *covariance*$(x_i,z_i)\sim0$ for most of the neighbours, then we apply bicubic interpolation along y-axis in the original matrix to generate the missing pixels, else we apply bilinear interpolation to generate the missing pixels.

```
Input: The sequence of 2D MRI slices (x-z plane/ x-y plane/ y-z plane) of spine ($SC$), $gap$.
Output: 3D volumetric representation of spine
begin
        Step 1: Denoise all the slices in $SC$ using shearlet transform
        Step 2: D = $Concatenate$(SC); /*partially fill up 3D matrix D with the available data, maintaining the slice gap*/
        Step 3: G = $Gradient$(SC); /*a sequence of 2D matrices computed for each pixel in each 2D slice*/
        Step 4: Compute $covariance(x_i, z_i)$; /*a sequence of 2D matrices computed for each pixel in each 2D slice*/
        Step 5: for each 2D slice in $SC$
                begin
                    for each pixel
                    begin
                        Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions.
                        if the edge strength of the current pixel is the largest compared to the other pixels in the mask with the same direction and $cov(x_i, y_i) \sim 0$ for most of the neighbors (upto 8 neighbors in same slice to be considered)
                        then $D$ =Bicubic_interpolation($D$,SC(i))
                        else $D$ =Bilinear_interpolation($D$,SC(i))
                    end
                end
        Step 6: D = $Marching\_cube$(D)
        Step 7: D = $Smooth$(D). /*MATLAB standard function*/
        Step 8: D = $Colormap$(D). /*MATLAB standard function*/
        Step 9: $Rotate\_para$(D). /*MATLAB standard function*/
        Step 10: $Display$(D).
end
```

Fig. 2 *Algorithm 1: proposed algorithm: 3D reconstruction*

We have combined bicubic with bilinear interpolation in our proposed Algorithm 1 (see Fig. 2), in order to preserve the edges having minimum noise with the help of bicubic interpolation and to reconstruct the other components using the faster bilinear interpolation. This also reduces the time complexity compared to that of using bicubic interpolation for the full reconstruction, yet preserves the internal structures as required.

The above procedure is repeated until we reach the topmost layer of the 3D matrix in the *x–z* plane, taking two consecutive slices at a time. Then, we break the 3D matrix in voxels and further apply the Marching cubes algorithm for more accurate 3D reconstruction. For visualisation, we smooth the image by Savitzky-Golay filters [35] and local regression, then apply *colormap*, and activate the rotation operation (*Rotate_para*) so that the user can rotate the 3D image and view all sides of it as needed. Finally, we display the 3D reconstructed image on screen. A brief description of the method is illustrated in Fig. 3.

3D reconstruction can also be performed on the sequence of 2D slices along the sagittal (*y–z*) plane, or axial (*x–y*) plane with the corresponding value of *gap*, using the same algorithm.

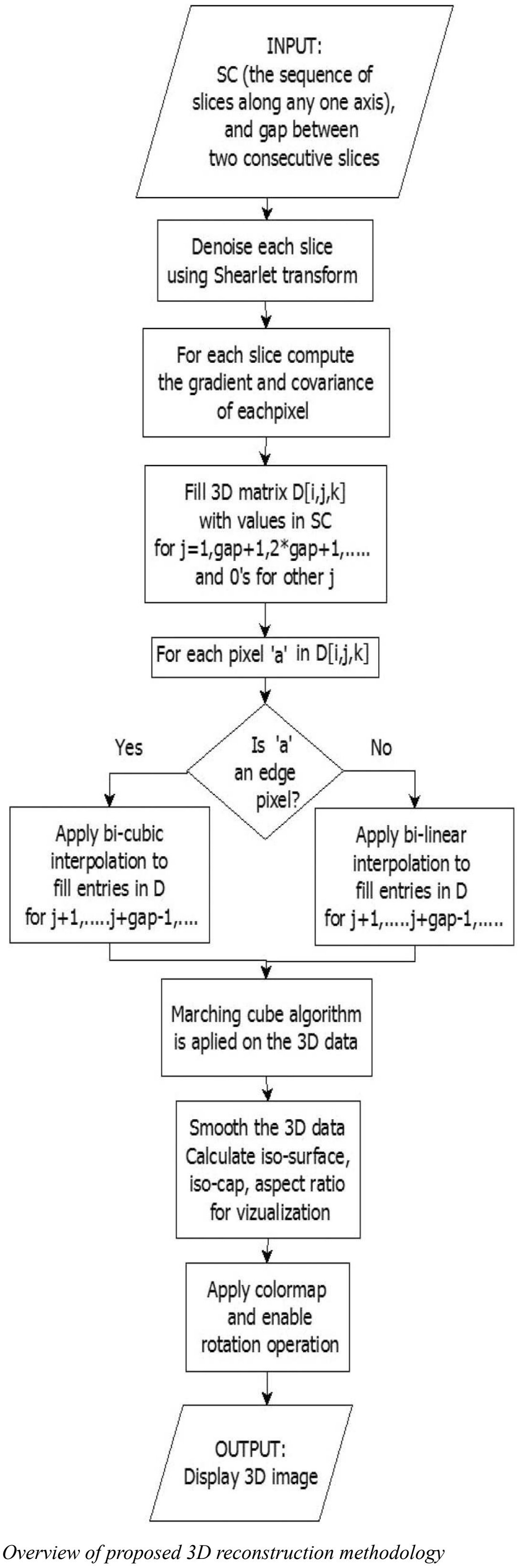
INPUT:
SC (the sequence of slices along any one axis), and gap between two consecutive slices
Denoise each slice using Shearlet transform
For each slice compute the gradient and covariance of eachpixel
Fill 3D matrix D[i,j,k] with values in SC for j=1,gap+1,2*gap+1,..... and 0's for other j
For each pixel 'a' in D[i,j,k]
Is 'a' an edge pixel?
Yes
No
Apply bi-cubic interpolation to fill entries in D for j+1,.....j+gap-1,....
Apply bi-linear interpolation to fill entries in D for j+1,.....j+gap-1,.....
Marching cube algorithm is aplied on the 3D data
Smooth the 3D data Calculate iso-surface, iso-cap, aspect ratio for vizualization
Apply colormap and enable rotation operation
OUTPUT:
Display 3D image

Fig. 3 *Overview of proposed 3D reconstruction methodology*

5.1.1 Computational complexity of proposed algorithm: The time complexity of bicubic interpolation and bilinear interpolation are both $O(n^2)$, where n is the total number of pixels in the $2D$ matrix along a single plane where the missing pixels are being calculated. An additional computation time for bicubic interpolation is to compute the gradient and cross-derivative at each re-sampled pixel position as it considers 16 nearest pixels as compared to 4 pixels in case of bilinear. The time complexity for interpolation is $O(n^2)$ where n is the number of known pixel intensity values and we interpolate along one axis, the computational complexity of local gradient and covariance is $O(m)$, where m is the total number of pixels in the 2D slice. Hence, time complexity of the proposed algorithm is $O(n^3)$ if $(n \times n \times n)$ is the size of the 3D matrix.

```
Input: $D$, the 3D matrix for the reconstructed volume data, $gap$
Output: Sequence of Sagittal, Coronal and Axial slices
begin
Step 1: $[m, n, p]$=Getdimension($D$); $j$= 1;
Step 2: while $j \leq m$ do /*Slice along sagittal axis*/
begin
Step 3: $M1 = D(j, :, :)$ /* Matrix Estimation*/.
Step 4:$M1$=squeeze($M1$)/* convert 3D matrix to a 2D one*/
Step 5: $M1 = Smooth(M1)$
Step 6: Display($M1$)
Step 7: $j$=$j$+$gap$;
end
Step 8: Steps 2-8 must be repeated for each of the other two indices of $D$ to get the required slices in the other two planes
end
```

Fig. 4 *Algorithm 2: proposed algorithm: slicing a 3D image*

5.2 2D slicing from 3D

The process of breaking the 3D image into several components along any axis is called slicing of the 3D image. In case of MR imaging, the images along the three axes are taken separately and hence it is expensive and time consuming. In order to reduce the total time and cost of imaging, we have designed an algorithm, Algorithm 2 (see Fig. 4) to generate the slices along all possible planes after re-constructing the 3D image from a sequence of 2D slices along one axis.

After the 3D image is created using Algorithm 1 (Fig. 2), we can slice out the image, along x, y or z axis given the specified gap by the user using Algorithm 2 (Fig. 4). We can also specify the location, with specific values for x, y and z and extract out the information based on that and display it as an image with a specified aspect ratio. The brief methodology is illustrated in Fig. 5.

First, the size of the 3D matrix $\boldsymbol{D}$ in terms of m, n and p is extracted and the gap between slices is stored in *gap* variable. Initialising $j = 1$, while j is less than or equal to m, the value of $\boldsymbol{D}(j, :, :)$ (matrix estimation) is extracted and j is increased by *gap*. The *squeeze* function is used to change the dimension of the extracted slice from 3D to 2D. Then *smooth* function is used on the extracted image for eliminating excess data, and the aspect ratio to view the image on screen is set. The image is rotated so that the display is always in x–y plane and then we either display it or store it in a specified location as instructed by the user. The same steps are repeated for the slices along y and z axis by replacing m with n and p, respectively.

The time complexity of the slicing algorithm is $O(n^2)$ where $n \times n$ is the size of each 2D slice.

6 Results and analysis

We used a workstation with Intel(R) Core(TM) i3-3340S processor @ 2.80 GHz, 4 G.B RAM and 64 bit operating system to implement our code using MATLAB 2018. We ran it on 25 real life T2-MRI datasets (512*512 pixels) of spine, collected from Bangur Institute of Neurosciences, S.S.K.M, Kolkata. We have focused on T2 weighted MRI as any deformity or disease is most highlighted in this format [3]. The execution time for 3D reconstruction part of our proposed algorithm is approximately 3 min and that for slicing is 0.036 s (if all slices are to be generated with 1 mm inter-slice gap and size of each image is on an average 280 kB).

We have effectively carried out 3D reconstruction of the slices in the coronal plane and sliced out all other planes from it. Time taken to capture all possible slices along all plane using a commonly used MRI machine is about 20 min approximately in one format (T1 weighted or T2 weighted or diffusion weighted etc.). The time taken to capture a single sequence of slice in one format is approximately 6 min. The most widely used techniques

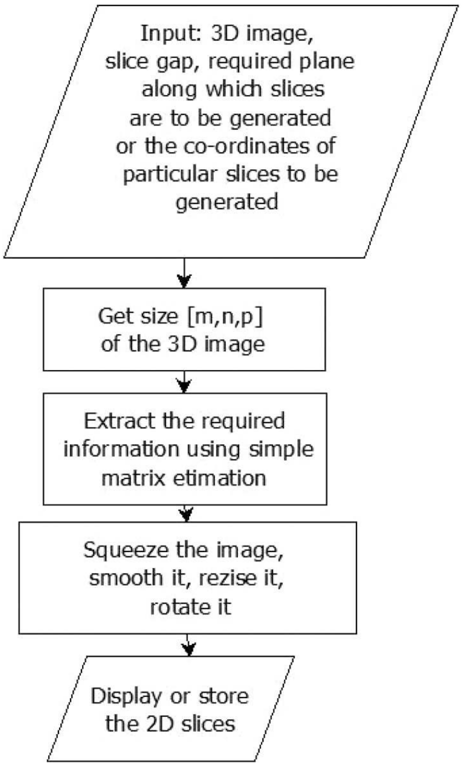
Input: 3D image, slice gap, required plane along which slices are to be generated or the co-ordinates of particular slices to be generated
Get size [m,n,p] of the 3D image
Extract the required information using simple matrix etimation
Squeeze the image, smooth it, rezise it, rotate it
Display or store the 2D slices

Fig. 5 *Overview of proposed 3D slicing methodology*

used in the subcontinent takes about 2 min time for reconstruction using all three planes [6] but in this case the patient is kept under MRI scanner for over 45 min approximately. Our algorithm takes approximately 3 min to reconstruct the 3D from slice sequence of a single plane and generate all possible slices from the reconstructed 3D. Hence, time reduced per format of data to be captured is approximately 9 min, thus, reducing the time of capturing the data by 80%.

We have used 12 sets of cervical spine data, among which 7 data sets were captured with an interslice gap of 3 mm and 4 data sets were captured with an interslice gap of 5 mm, 10 sets of lumbar spine data, among which 3 data sets were at an interslice gap of 3 mm, 2 data sets at an interslice gap of 1 mm and 5 data sets at an intersilce gap of 5 mm, and 3 sets of full spine data, all of which were taken with an interslice gap of 5 mm.

In the proposed work, we have used bicubic interpolation along the edges and bilinear otherwise in order to preserve the edges properly without increasing the time of computation excessively. Using only bi-cubic interpolation increases the time as well as smoothens the magnetic resonance image of fine tissues along the edges resulting in errors. The output of our 3D reconstruction on *Cervical spine data set 1* is shown in Fig. 6. Next, this reconstructed 3D is sliced as shown in Fig. 7 and the resulting slices are compared with the original data using MI, RMSE and SSIM as shown in Table 1.

Let $k \in \{Cervicalspine, LumbarSpine, Fullspine\}$, the set of three different types of data, $g \in \{1\,\text{mm}, 2\,\text{mm}, 3\,\text{mm}, 4\,\text{mm}, 5\text{mm}\}$, the set of five different possible gaps between slices. Let p_k be the total number of dataset of each type k, and q_g be the total number of slices of a single dataset k with a specific gap g.

The values reported in Table 1 for a dataset type k and inter-slice gap of g are calculated as follows:

(see (7))

(see (8))

(see (9))

In Fig. 8, we have shown the output of 3D reconstruction and slicing on data set 14 of full spine. The average time taken for 3D reconstruction is 3 min approximately based on the size of the input data set and the time taken for slicing is approximately 2 s.

In Table 2, the average accuracy percentage with respect to each slice axis is calculated as

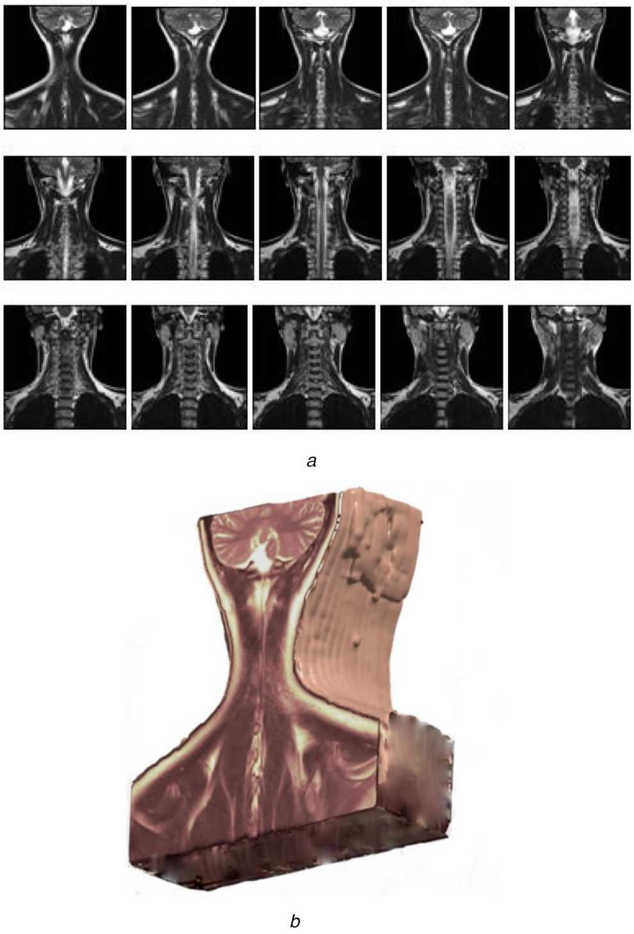

Fig. 6 *3D reconstruction of cervical spine (Data set 1) from coronal MRI slices*

(a) Input - Coronal Slices at 5 mm gap, ***(b)*** Reconstructed 3D image

$$A_{\text{axis}}\% = \text{Avg}(A_{\text{axis}}^{MI}\%, A_{\text{axis}}^{RMSE}\%, A_{\text{axis}}^{SSIM}\%) \qquad (10)$$

where

(see (11))

(see (12))

(see (13))

where G is the set of ground truth images and $Recon$ is the set of sliced images after reconstruction. The mutual information $MI(G, Recon)$ is between the original ground truth image and the sliced reconstructed image, and $MI(G, G)$ is the mutual information if the original image and the sliced image are equal. For the subsets of MRI data sets of each of the three regions of spine along each of the three axes perpendicular to the plane of slicing after reconstruction with five different values of inter-slice gap, Table 1 reports the values of mutual information, entropy difference, RMSE and structural similarity index measure, averaged over all data sets for a particular region over all three axes and all five inter-slice gaps obtained by our proposed method in comparison to the original data. Table 2 shows the average accuracy of the slices after applying our method along all 3 sequence of slices. Since the original data was at a slice gap of either 5 mm, 3 mm or 1 mm slices, hence these slices could be exactly matched with the original dataset, whereas in case of 2 and 4 mm gap we can compare only the matching slices in the original dataset considering 1mm=4pixels. The average accuracy of the slices generated after reconstruction compared to the original slices is 96%.

Our proposed technique has been separately compared with bilinear interpolation as well as bicubic interpolation. The comparison results show that our algorithm gives good accuracy with respect to *MI*, *RMSE* and *SSIM* as shown in Fig. 9.

The technique in [6] for 3D reconstruction of MRI images considering slices in 3-planes needed around 15 min for one format

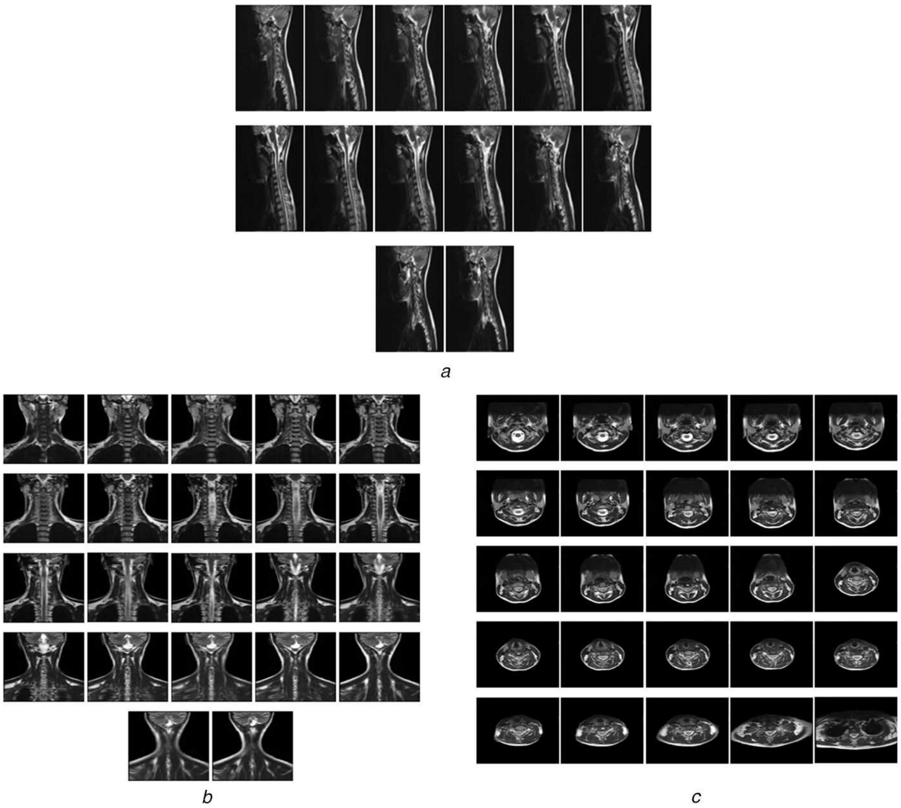

Fig. 7 *Slices from 3D reconstructed image of cervical spine (Data set 1) as shown in Fig. 6b*
(a) Sliced Sagittal with 3 mm gap, ***(b)*** Sliced Coronal with 2 mm gap, ***(c)*** Sliced Axial with 3 mm gap

Table 1 Average mutual information, root mean square and structural similarity index of slices along all three axes for cervical, lumbar and full spine

Region	Inter-slice gap	Avg MI	Avg RMSE	Avg SSIM
cervical spine	1 mm	5.72	0.04	0.97
	2 mm	5.7	0.045	0.962
	3 mm	5.7	0.049	0.96
	4 mm	5.7	0.045	0.96
	5 mm	5.72	0.04	0.97
lumbar spine	1 mm	6.12	0.035	0.98
	2 mm	6.09	0.04	0.978
	3 mm	6.1	0.04	0.977
	4 mm	6.09	0.04	0.978
	5 mm	6.12	0.032	0.98
full spine	1 mm	5.44	0.045	0.959
	2 mm	5.43	0.05	0.952
	3 mm	5.4	0.05	0.952
	4 mm	5.4	0.049	0.955
	5 mm	5.442	0.045	0.96

$$Avg\_MI_k(g) = \left(\frac{1}{p_k}\left(\frac{\sum_{i=1}^{p_k}\sum_{j=1}^{q_g}(MI(G_j, \text{Recon}_j))}{(\text{total\_number\_of\_slices\_with\_}k\text{\_type\_datasets\_and\_gap\_}g)}\right)\right) \tag{7}$$

$$Avg\_RMSE_k(g) = \left(\frac{1}{p_k}\left(\frac{\sum_{i=1}^{p_k}\sum_{j=1}^{q_g}(RMSE(G_j, \text{Recon}_j))}{(\text{total\_number\_of\_slices\_with\_}k\text{\_type\_datasets\_and\_gap\_}g)}\right)\right) \tag{8}$$

$$Avg\_SSIM_k(g) = \left(\frac{1}{p_k}\left(\frac{\sum_{i=1}^{p_k}\sum_{j=1}^{q_g}(SSIM(G_j, \text{Recon}_j))}{(\text{total\_number\_of\_slices\_with\_}k\text{\_type\_datasets\_and\_gap\_}g)}\right)\right) \tag{9}$$

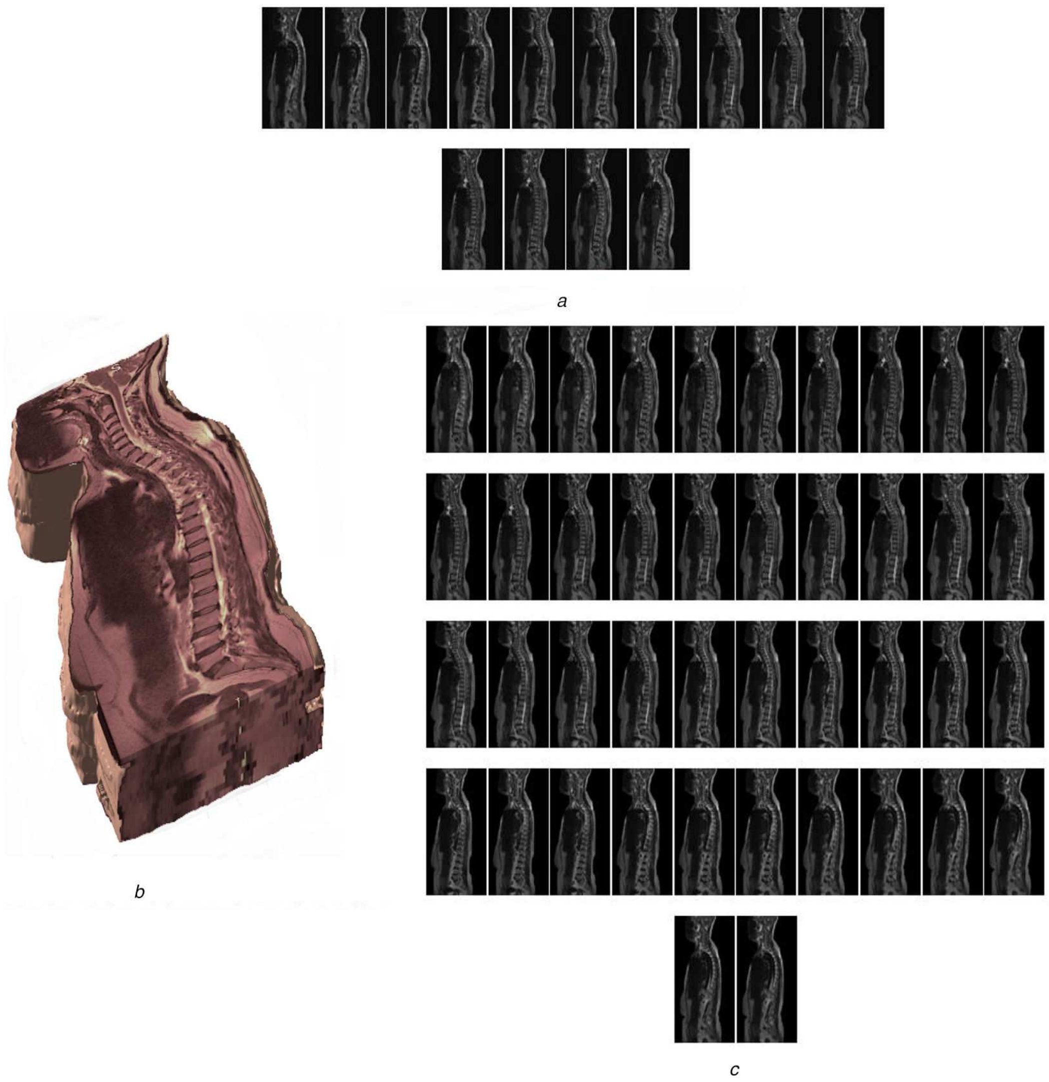

Fig. 8 *3D reconstruction and slicing of MRI spine (full spine: data set 14)*
(a) Input: sagittal slices with 5 mm gap, ***(b)*** Reconstructed 3D, ***(c)*** Sliced sagittal at 1 mm gap from the reconstructed 3D

$$A_{\text{axis}}^{MI}\% = \left[\left(\frac{1}{5}\frac{\sum_{\text{gap}\,=\,1}^{5}\sum_{j\,=\,1}^{m}(MI(G_j), G_j) - (MI(G_j, \text{Recon}_j))}{(\text{total\_number\_of\_slices(axis)\_over\_all\_gaps})}\right)\right] * 100 \tag{11}$$

$$A_{\text{axis}}^{RMSE}\% = \left[\left(\frac{1}{5}\frac{\sum_{\text{gap}\,=\,1}^{5}\sum_{j\,=\,1}^{m}RMSE(G_j, \text{Recon}_j)}{(\text{total\_number\_of\_slices(axis)\_over\_all\_gaps})}\right)\right] * 100 \tag{12}$$

$$A_{\text{axis}}^{SSIM}\% = \left[1 - \left(\frac{1}{5}\frac{\sum_{\text{gap}\,=\,1}^{5}\sum_{j\,=\,1}^{m}(1 - SSIM(G_j, \text{Recon}_j))}{(\text{total\_number\_of\_slices(axis)\_over\_all\_gaps})}\right)\right] * 100 \tag{13}$$

(say, T1 weighted [3, 4, 26]) as the total execution time from acquiring the 2D slices, reconstructing the 3D using them and again slicing from the reconstructed 3D as per user's input along any plane through any given inter slice gap. The results after slicing on our datasets using 3-plane method [6] shows approximately an average RMSE 0.04, average mutual information 5.7 and average structural similarity index measure 0.975 which are very close to our experimental results as shown in Table 1 but the execution time for our proposed method is approximately 3 min only (for one format say, T2 weighted). Our proposed algorithm works on any of the commonly used formats of MRI data.

We have also compared our work with the average performance of 3-plane methods [6] and a robust edge directed interpolation technique for reconstruction [27] based on accuracy and time as shown in Fig. 10. Our proposed algorithm gives better accurate result in much less time.

6.1 User interface

We have designed an user interface as shown in Fig. 11, with which an user can generate a 3D view from a set of 2D MRI slices along a single plane and slice out this 3D as per specifications to see the slices along any plane in different orientations.

7 Conclusion

We have carried out our experiments with real life data from the subcontinent. We have compared our results with the ground truth data and it shows that our algorithm provides an average accuracy

Table 2 Average accuracy of each sequence of cut along axial, sagittal and coronal planes considering inter-slice gaps (1, 2, 3, 4 and 5mm)

Data set	Region of spine	% Accuracy (axial)	% Accuracy (sagittal)	% Accuracy (coronal)
1	cervical	98.675	92.562	97.006
2	lumbar	98.800	92.957	98.052
3	lumbar	98.876	93.774	98.004
4	full	97.826	92.000	97.098
5	cervical	98.990	93.788	97.935
6	cervical	98.987	92.989	97.996
7	cervical	98.980	92.895	97.986
8	lumbar	98.959	92.723	97.845
9	lumbar	98.977	93.555	98.684
10	lumbar	98.841	93.912	97.887
11	cervical	98.018	94.851	98.106
12	cervical	98.114	93.194	98.238
13	lumbar	98.944	93.186	97.866
14	full	98.003	92.365	97.474
15	cervical	97.833	92.699	97.830
16	lumbar	98.847	92.423	97.653
17	cervical	98.970	92.660	97.800
18	cervical	98.391	93.198	97.855
19	cervical	98.682	92.654	97.807
20	lumbar	98.772	93.766	97.880
21	full	98.104	92.326	97.398
22	lumbar	98.974	93.194	97.863
23	lumbar	98.976	93.256	97.861
24	cervical	98.769	92.823	97.894
25	cervical	98.622	92.543	97.777

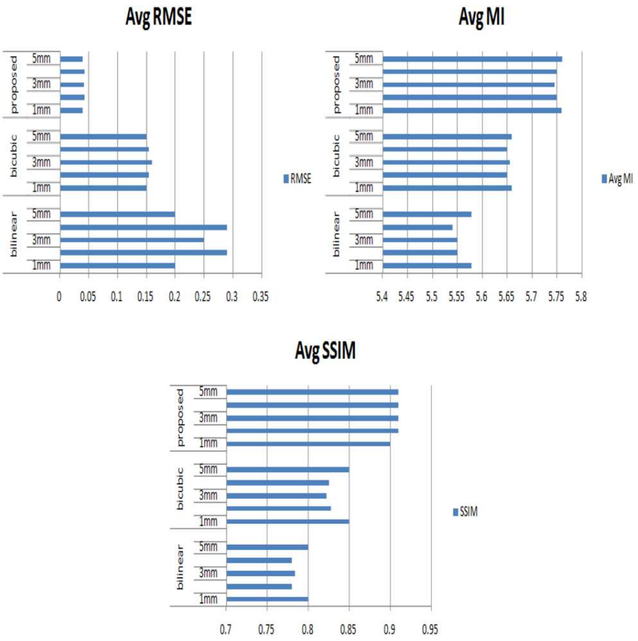

Fig. 9 *Comparison of our proposed algorithm with those by only bilinear and only bicubic interpolation*

of 96%. To the best of our knowledge, we could not find a similar work of reconstructing 3D image from a single sequence of MRI slices with more than 2 mm gap between slices and then cutting out slices from the 3D as per the user's wish, in existing literature and hence were unable to show a direct comparison of our results with any of the earlier works.

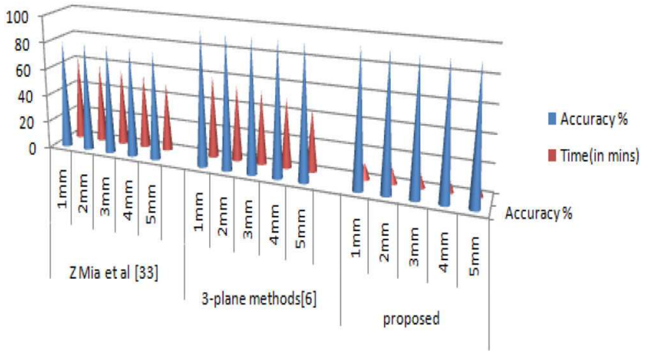

Fig. 10 *Comparison of our proposed algorithm with 3-plane methods [6] and Mia et al. [33] with respect to time and accuracy*

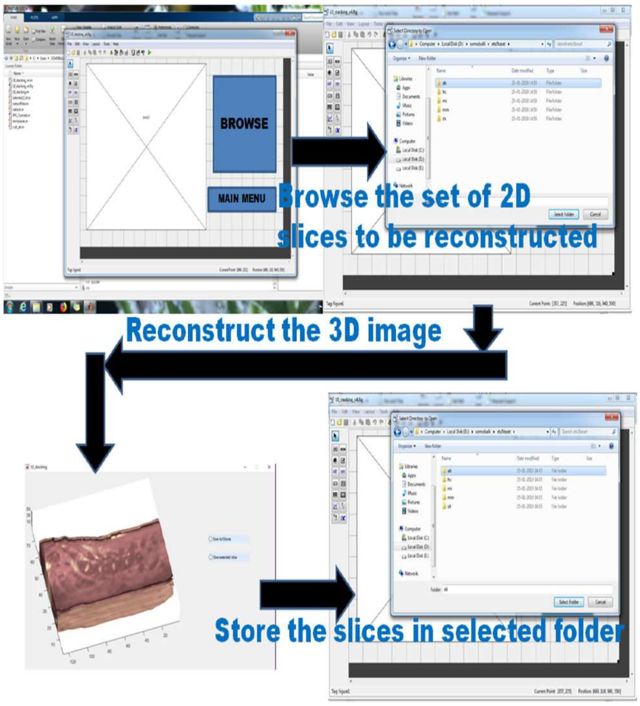
BROWSE
MAIN MENU
Browse the set of 2D slices to be reconstructed
Reconstruct the 3D image
Store the slices in selected folder

Fig. 11 *User interface for 3D reconstruction and slicing*

8 References

[1] https://www.medicalradiation.com/types-of-medical-imaging/

[2] Smith-Bindman, R., Miglioretti, D.L., Johnson, E.: 'Use of diagnostic imaging studies and associated radiation exposure for patients enrolled in large integrated health care systems 1996–2010', *JAMA*, 2012, **307**, (22), pp. 2400–2409, doi: 10.1001/jama.2012.5960

[3] Hammer, M.: 'MRI Physics: Diffusion-Weighted Imaging 2013–2014'. XRayPhysics. Retrieved 2017-10-15

[4] Kleut, D., Jovanovic, M., Reljin, B.R.: '3D visualisation of MRI images using MATLAB', *J. Autom. Control University Of Belgrade*, 2006, **16**, pp. 1–3

[5] Herman, G.T., Liu, H.K.: 'Three-dimensional display of human organs from computed tomograms', *Computer Graphic Imaging Proceedings*, 1979, **9**, pp. 1–21

[6] Borse, M., Patil, S.B., Patil, B.S.: 'Literature survey for 3d reconstruction of brain mri images', *Int. J. Res. Eng. Technol.*, 2013, **02**, (11), pp. 743–748. Available at http://www.ijret.org

[7] Herghelegiu, P., Gavrilescu, M., Manta, V.: 'Visualization of segmented structures in 3D multimodal medical data sets', *Adv. Electr. Comput. Eng.*, 2011, **11**, (3), pp. 99–104

[8] Udupa, J., Hung, H.M., Chuang, K.S.: 'Surface and volume rendering in three- dimensional imaging', *J. Digit. Imag.*, 1991, **4**, p. 159. Available at https://doi.org/10.1007/BF03168161

[9] Hung, C.S., Huang, C.F., Ouhyoung, M.: 'Fast Volume Rendering for Medical Image Data', Communication and Multimedia Laboratory Department of Computer Science and Information Engineering National Taiwan University, Taiwan

[10] Rubin, G.D., Beaulieu, C.F., Argiro, V., *et al.*: 'Perceptive volume rendering of CT and MR images: applications for endoscopic imaging', *Radiology*, 1996, **199**, pp. 321–330

[11] Hu, X., Tan, K.K., Levin, D.N., *et al.*: 'Three-dimensional magnetic resonance images of the brain: application to neurosurgical planning', *J. Neurosurgery*, 1990, **72**, pp. 433–440

[12] Rinck, P.: 'Magnetic Resonance in Medicine'. The Basic Textbook of the European Magnetic Resonance Forum. 11th edition; 2017. Electronic version 11, published 1 June 2017

[13] Zhu, X., He, X., Wang, P., *et al.*: 'A method of localization and segmentation of intervertebral discs in spine MRI based on gabor filter bank', *BioMed. Eng. OnLine*, 2016, **15**, pp. 32, doi 10.1186/s12938-016-0146-5

[14] Michopoulou, S.K., Costaridou, L., Panagiotopoulos, E., *et al.*: 'Atlas-based segmentation of degenerated lumbar intervertebral discs from MR images of the spine', *IEEE Trans. Biomed. Eng.*, 2009, **56**, pp. 2225–2231

[15] Castro-Mateos, I., Pozo, J.M., Lazary, A., *et al.*: '2D segmentation of intervertebral discs and its degree of degeneration from T2-weighted magnetic resonance images', *Med. Imag. Comput. Aided Diagn.*, 2014, **17**, pp. 9035

[16] Haq, R., Aras, R., Besachio, D.A., *et al.*: '3D lumbar spine intervertebral disc segmentation and compression simulation from MRI using shape-aware models', *Int. J. Comput. Assist. Radiol. Surg.*, 2015, **10**, pp. 45–54

[17] Law, M.W.K., Tay, K., Leung, A., *et al.*: 'Intervertebral disc segmentation in MR images using anisotropic oriented flux', *Med. Image Anal.*, 2013, **17**, pp. 43–61

[18] Chevrefils, C., Cheriet, F., Grimard, G., *et al.*: 'Watershed segmentation of intervertebral disk and spinal canal from MRI images', *Image Anal. Recognit.*, 2007, **4633**, pp. 1017–1027

[19] Ghoshal, S., Chatterjee, P., Banu, S., *et al.*: 'A software tool for 3D visualization and slicing of MR images'. Proc. of the 10th EAI Int. Conf. on Simulation Tools and Techniques (SIMUTOOLS)', New York, NY, USA, 2017, pp. 103–107

[20] Gao, S., Gruev, V.: 'Bilinear and bicubic interpolation methods for division of focal plane polarimeters', *Opt. Express*, 2011, **19**, (27), p. 26161

[21] Keys, R.: 'Cubic convolution interpolation for digital image processing', *IEEE Trans. Acoustics, Speech, Signal Process.*, 1981, **29**, (6), pp. 1153–1160

[22] Lorensen, W.E., Cline, H.E.: 'Marching cubes: a high resolution 3d surface construction algorithm', *SIGGRAPH Comput. Graph.*, 1987, **21**, (4), pp. 163–169

[23] Gu, K., Zhai, G., Yang, X., *et al.*: 'Subjective and objective quality assessment for images with contrast change'. IEEE Int. Conf. on Image

Processing, Melbourne, VIC, 2013, pp. 383–387, doi: 10.1109/ICIP.2013.6738079

[24] Zhang, S., Liu, Z., Liu, B.*, et al.*: 'Medical image registration by using salient phase congruency and regional mutual information'. 4th Int. Congress on Image and Signal Processing, Shanghai, People's Republic of China, 2011, pp. 760–764, doi: 10.1109/CISP.2011.6100379

[25] Gonzalez, R.C., Woods, R.E.: '*Digital image processing*' (Pearson Education, UK., 2009, 3rd edn.)

[26] Wang, Z., Bovik, A.C., Sheikh, H.R.*, et al.*: 'Image quality assessment: from error visibility to structural similarity', *IEEE Trans. Image Process.*, 2004, **13**, (4), pp. 600–612, doi: 10.1109/TIP.2003.819861

[27] Caliskan, A., Cevik, U.: 'Three-dimensional modeling in medical image processing by using fractal geometry', *J. Comput.*, 2017, **12**, pp. 479–485. 10.17706/jcp.12.5.479-485

[28] Huang, Y., Qiu, Z., Song, Z.: '3D reconstruction and visualization from 2D CT images'. 2011 IEEE Int. Symp. on IT in Medicine and Education, Cuangzhou, 2011, pp. 153–157. doi: 10.1109/ITiME.2011.6132078

[29] Nielson, G.M.: 'On marching cubes', *IEEE Trans. Vis. Comput. Graphics*, 2003, **9**, (3), pp. 283–297, doi: 10.1109/TVCG.2003.1207437

[30] Purchase, H.C., Andrienko, N., Jankun-Kelly, T.J.*, et al.*: 'Theoretical foundations of information visualization'. Information Visualization LNCS 4950, Berlin, Germany, 2008, pp. 46–64

[31] Narkbuakaew, W., Sotthivirat, S., Gansawat, D.*, et al.*: '3d Surface Reconstruction Of Large Medical Data Using Marching Cubes In Vtk', National Electronics and Computer Technology Center, Phahon Yothin Rd, Klong Luang, Pathumthani, Thailand

[32] Thanh, C.Q.T., Hai, N.T.: 'Trilinear interpolation algorithm for reconstruction of 3D MRI brain image', *Am. J. Signal Process.*, 2017, **7**, (1), pp. 1–11, p-ISSN: 2165-9354 e-ISSN: 2165-9362

[33] Mai, Z., Rajan, J., Verhoye, M.*, et al.*: 'Robust edge-directed interpolation of magnetic resonance images'. 2011 4th Int. Conf. on Biomedical Engineering and Informatics (BMEI), Shanghai, 2011, pp. 472–476, doi: 10.1109/BMEI.2011.6098244

[34] Shahdoosti, H.R., Khayat, O.: 'Image denoising using sparse representation classification and non-subsampled shearlet transform', *Signal, Image and Video Processing*, 2016, **10**, pp. 1081. Available at https://doi.org/10.1007/s11760-016-0862-0

[35] Shekhar, C.: 'On simplified application of multidimensional savitzky-golay filters and differentiators', *Prog. Appl. Math. Sci. Eng.*, 2015, **1705**, (1), p. 020014