

Project Report on Post-Processing of Prostate MRI

Mohit Kumar Ahuja and Gopikrishna Erabati

Supervisor : Dr. Christian Mata Miquel



Master in Computer Vision
UNIVERSITE DE BOURGOGNE
Centre Universitaire Condorcet - UB, Le Creusot
13 May 2017

Contents*

1. Introduction	3
1.1 Diagnostic	4
1.2 Dicom Images	4
2. The Design of GUI and its usage	5
3. Tasks	8
3.1 1 st Stage	8
3.2 2 nd Stage	9
3.3 3 rd stage	16
4. Conclusion	16
References	17

Note: In the report we have colour coded the **problems we faced in **red** and **solutions** followed in **green**, for quick access.*

1. Introduction:

Adenocarcinoma of the prostate appears in older men. About 85% of cases are diagnosed in men over 60 years. Prostate cancer is a common cancer whose incidence and mortality are now steadily increasing (85,000 new cases per year in Europe) [1]. It is the second most common cancer after lung cancer and the third leading cause of cancer death in men (9% of all cancer deaths in men in Europe). The location of Prostate gland is shown in Fig. 1.

Currently, there are four anatomically glandular areas within the prostate (as shown in Fig. 2):

- Peripheral zone (ZP)
- Central zone (ZC)
- Transition zone (ZT)
- Anterior Fibromuscular Tissue (AFT)

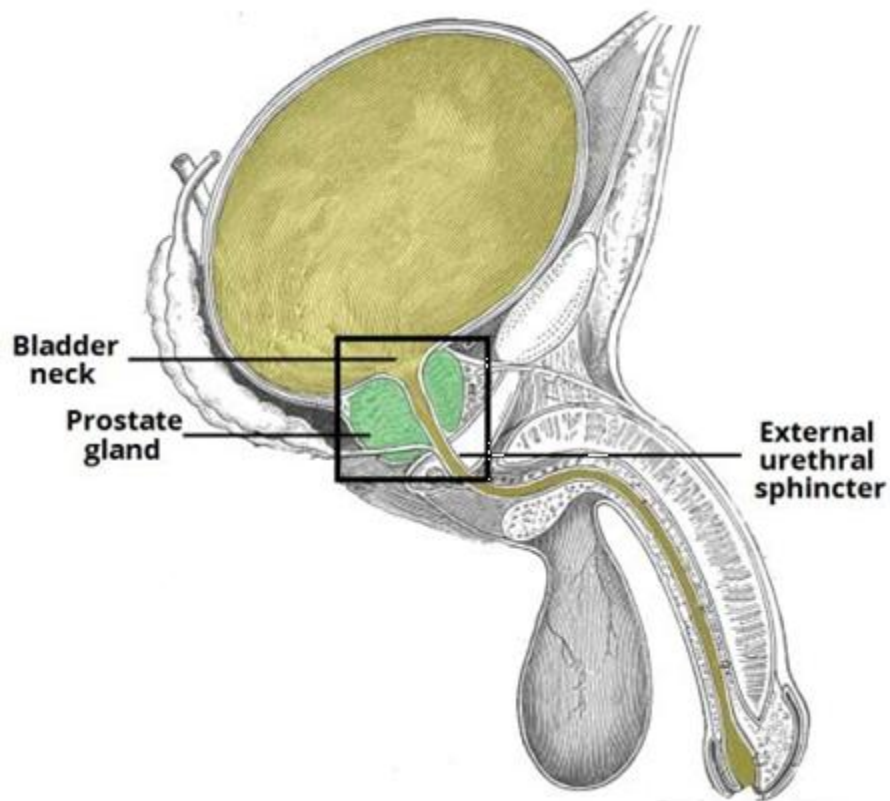


Figure 1: Location of Prostate Gland

ZONES OF THE PROSTATE

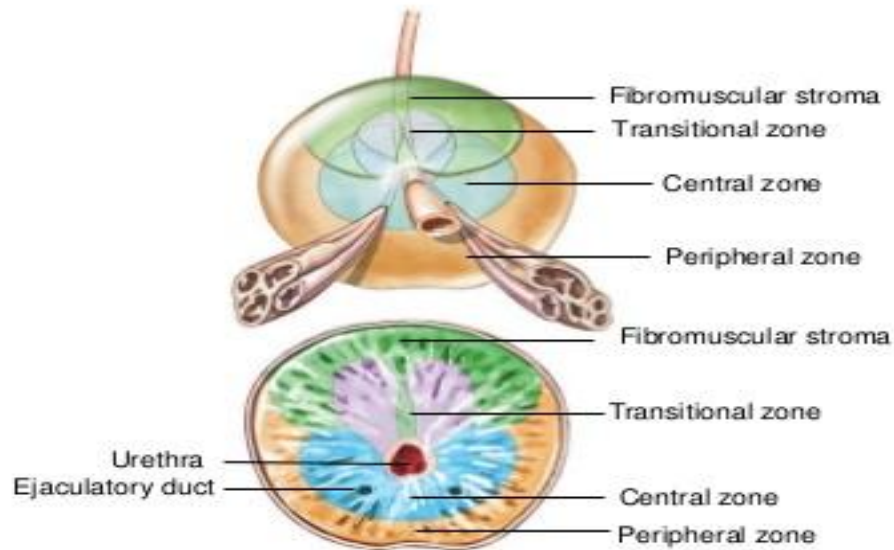


Figure 2: Anatomy of the prostate in transversal and sagittal plane

Adenocarcinoma arises in 75% of cases in the ZP, in 20% of cases in the ZT and in 5% of cases in the ZC.

1.1 Diagnostic:

The Magnetic Resonance Imaging (MRI) provides high-resolution images in all planes of space, allowing precise anatomical visualization of the prostate. MRI of prostate cancer benefits from technological advances that expand more indications. MRI is a valuable tool to guide therapeutic management of prostate cancer through acquisition sequences[2] as follows:

- Anatomical imaging– 3D - T2 weighted imaging
- Diffusion imaging – ADC (Apparent Diffusion Coefficient)
- Perfusion imaging –DCE (Dynamic Contrast Enhancement)
(Observation from signal-intensity time curve)
- Spectroscopy

1.2 Dicom Images [3]:

In this project we are dealing with DICOM format images so it's compulsory to first know what a DICOM image is. Digital Imaging and Communications in Medicine (DICOM) is a standard for handling, storing, printing, and transmitting information in medical imaging. It includes a file format definition and a network communications protocol. The communication protocol is an application protocol that uses TCP/IP to communicate between systems. DICOM files can be

exchanged between two entities that are capable of receiving image and patient data in DICOM format.

A DICOM format encapsulates an image and data object, data object consists of a number of attributes, including items such as name, ID, etc., and also one special attribute containing the image pixel data (i.e. logically, the main object has no "header" as such, being merely a list of attributes, including the pixel data). A single DICOM object can have only one attribute containing pixel data. For many modalities, this corresponds to a single image. However, the attribute may contain multiple "frames", allowing storage of cine loops or other multi-frame data. Another example is NM data, where an NM image, by definition, is a multi-dimensional multi-frame image. In these cases, three- or four-dimensional data can be encapsulated in a single DICOM object.

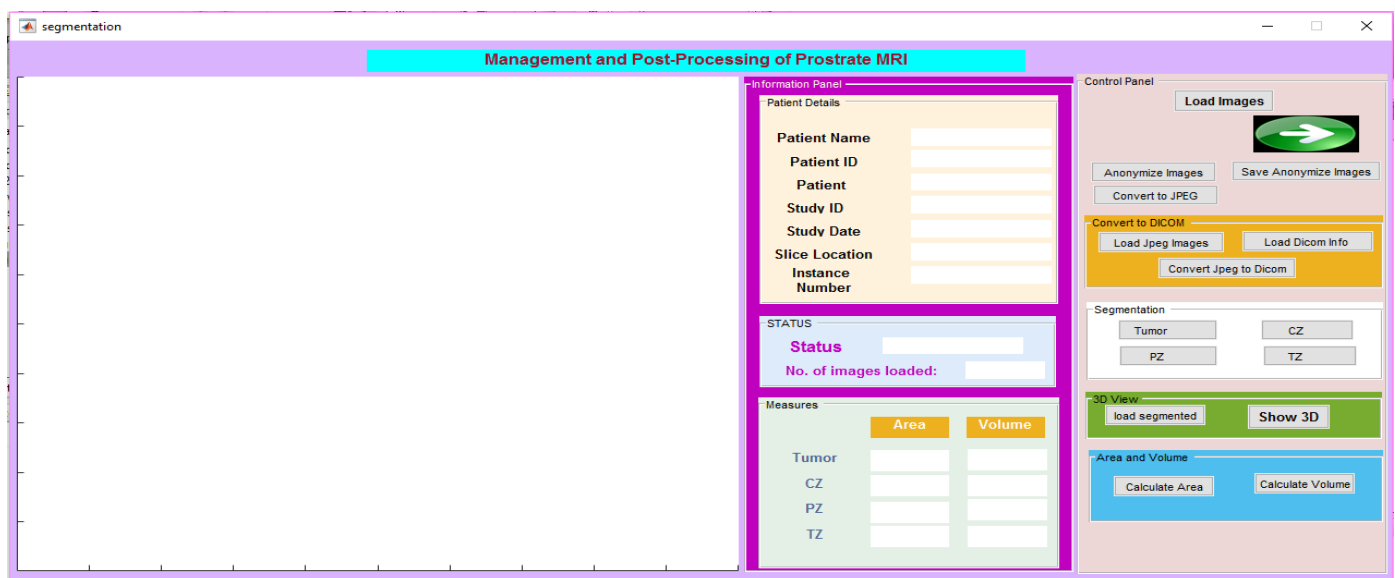


Figure 3. The Design of GUI

2. The Design of GUI and its usage

The GUI is designed using MATLAB as shown in Fig. 3. This section explains the user on how to use it for further process.

1. We first load all the images in the database (DICOM format) into our GUI using "Load images" button in the control panel section (as shown in Fig. 4), by selecting the folder which contains images in a popup dialog box that appears after clicking "Load Images" button. Here, while getting files from the folder selected, we faced a **problem** of hidden files ('.' and '..') in windows folder so we **solved** this issue by writing a 'dir2.m' function which neglects the hidden files while reading files from folder. Initially it was taking time to load the images in the database, so we faced the **problem** of not knowing whether the process is running in the background or not, so we thought of a **solution** and implemented "Status" to know the current status of our program. As explained by selecting the folder in popup dialog, we load all the

images and status shows "Loading Images..." and when the status turns to "Loading Done!" , which means that every image is loaded (as shown in Fig. 5) and we can perform the further action. By doing this, we need not to load every single image again and again.

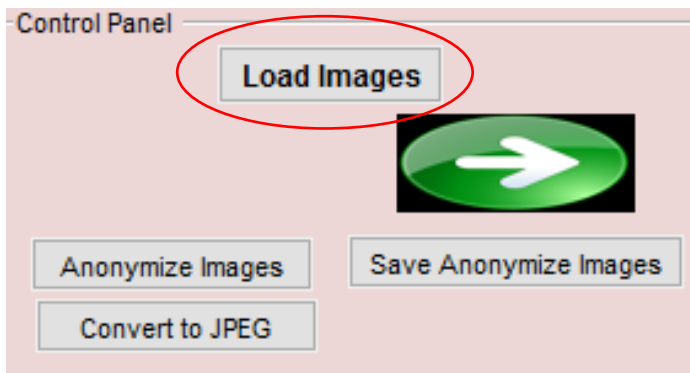


Figure 4: Control Panel

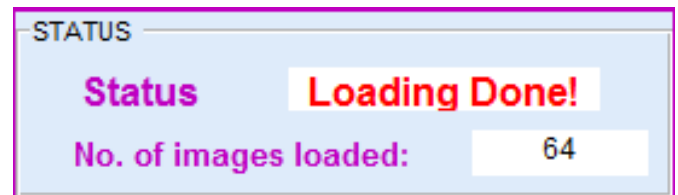


Figure 5: Status panel



Figure 6: Complete GUI at first image

As all the images are loaded into GUI, it will show the first image in the image section and corresponding patient related data in "Patient Details" panel and the number of images loaded in the status panel. As we can see the number of images loaded is "64" as there are 64 images in our database (as shown in Fig. 5).

2. As we loaded all the images and data into our GUI, we faced **problem** with showing all the images and data in our GUI, and we **solved** this issue by implementing the "Next Image" and "Previous Image" buttons for traversing to next and previous images as required. This we

implemented by getting 'currentindex' of image displayed and incrementing and decrementing the index as required. Here we had one more **problem** with the "Next Image" and "Previous Image" buttons when we are at first image and last image of our database, so we **solved** this problem by enabling and disabling the buttons according to some simple logic which can be easily understood in the code.

3. The "Anonymize Images" button anonymize the images as required and "Save Anonymize Images" button saves the images and data in DICOM format to 'Anonymized Images' folder in current working directory(cwd). The "Convert to JPEG" button converts the DICOM images to JPEG and saves the metadata in 'Meta Data' folder in cwd. Here, we simply converted dicom images to gray format and scaled it to 255 and converted to unsigned integer 8 (uint8) and saved in 'Jpeg Images' folder in cwd.

4. In "convert to Dicom" panel, the "Load Jpeg Images" button opens a popup dialog box to select folder for Jpeg images. Here, we faced a **problem** of filenames as windows file dialog reads filenames, as for example, it reads 'file1', 'file2', 'file10', 'file12' as 'file1', 'file10', 'file12', 'file2'. So we faced the problem of order of filenames and order of images loaded and we **solved** this using a code from 'Mathworks' [4] which sort the files according to order we require. 'natsort.m', 'natsortfiles.m' codes does this job for us.

"Load Dicom Info" button loads the metadata required to convert to dicom format. "Convert Jpeg to Dicom" button converts the Jpeg images to Dicom images and saves them in 'Dicom Images' folder in cwd.

5. In the ""Segmentation" panel , the four buttons with prostrate regions names are used to manually segment the regions accordingly and the segmented regions images will be saved into the "Segmented Images" folder in cwd with their standard names of regions (for example tumor.jpeg).

6. The "3D View" panel is used to 3D visualize the selected slices of gland.

NOTE This 3D view takes images of all slices of a region of gland and constructs its 3D view. Please select files for these 3D view from 'All Slices' folder in cwd.

Load the images from the above said folder for a 3D reconstruction and click "Show 3D" button which displays the 3D view.

7. The "Area and Volume" panel contains button to calculate area and volume of selected four regions of the gland.

NOTE We need to perform step 5 to before proceeding to the area and volume.

3. Tasks

NOTE : There is a README.txt file in the code folder, which explains about the m-files in the project and steps required to run the GUI and the code is well commented for user understanding.

3.1 1st Stage

- **Display the information of DICOM (Patient Name, Patient ID, Patient Birth Date, Study ID, Study Date, Slice Location and Instance Number)**

Solution: As explained earlier, there is a "Load Images" button would load the images and metadata using `'dicominfo'` and `'dicomread'` commands, The required data is read from the structure by accessing the `'currentindex'` of image and data, and information is displayed using text boxes. The 'Next image' button changes the image and data as explained in section 2.

- **Anonymize DICOM images: Modify the following fields (Patient Name, Patient ID and Birth Date) for all the set of images.**

Solution: For Anonymization of images, there is a button in the GUI named "Anonymize Images" which will erase the previous information of Patient Name, Patient ID and Birth Date. By running this in loop, all the images will be anonymized with a new information of Patient Name, Patient ID and Birth Date.

- **Save the anonymized images (Create a new folder containing the new anonymized images)**

Solution: There is a button named "Save Anonymized Images" just next to "Anonymize Images" which is used to save all the images with updated metadata to a new folder named 'Anonymized images' in cwd. The `'dicomwrite'` command helps us to save the images and updated metadata into new file.

- **Convert the DICOM image in JPG format (Be careful with the DICOM information)**

Solution: Before converting the data into JPG format we have to save the metadata of each image in a `'mat'` file so that our metadata will be safe ,because while converting the images to JPG format, we dont consider the metadata.

For converting the images to JPG format, there is a button in the control panel named "Convert to JPEG" which will first save the metadata of each image in a `'mat'` file. So, by doing this we secured the metadata of DICOM images. It will also create two new folders: one for metadata

and the other for jpg images. And by just using `'imwrite'` command, we save all the converted images to jpg format.

- **Convert the JPG image into DICOM format**

Solution: As explained earlier, we have already saved the metadata of each image in a 'mat' file. And now we not only need to convert the JPG images into DICOM format but also concatenate the metadata of each DICOM image with it.

So, to convert to DICOM format there is a special panel in GUI named “Convert to DICOM” (as shown in Fig. 7). In which three buttons are used, one to load the JPG images and the second to load the metadata of those images and the third button to convert the jpg images to DICOM format by using `'dicomwrite'` command.

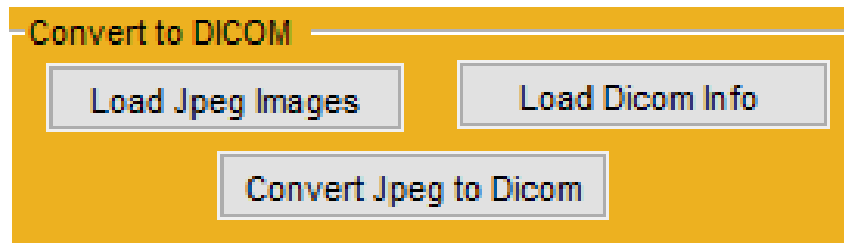


Figure 7: Convert to DICOM Panel

3.2 2nd stage

- **Manual segmentation for each region (only ZP, ZT, ZC, and the tumor region).**

Solution: For manual segmentation of images, we used a predefined function `'imfreehand(gca,'closed','false')` where closed is a scalar logical that controls whether the freehand region is closed. True creates a closed freehand region. False creates an open freehand region.

So, we used `'imfreehand'` function for manually segmenting each region ZP, ZT, ZC and the tumor region. You can find a dedicated panel for just segmentation of images (as shown in Fig. 8). This function creates an object for manual segmentation which later can be used for different tasks.

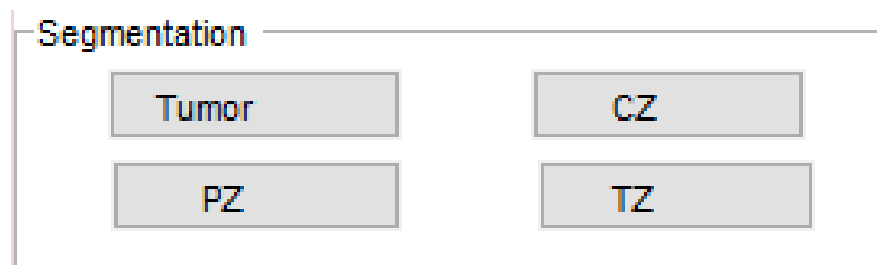


Figure 8: Segmentation Panel

So, when we segment the region of tumor for instance, we would create a binary image (mask) of the segmented area using the object created by `'imfreehand'` command.. And we save that binary image to "Segmented images" folder in cwd.

- **Show a 3D representation of the regions of prostate gland**

Approach 1: Using One Slice

We manually segmented the regions of tumor and full prostate gland from one slice which we found good detection of tumor and other regions (slice 29 , in our case). So, after getting the binary images of all the regions and full prostate we find boundary of the binary image using `'bwboundaries'` command and find the pixel coordinates of boundary and plot the same pixel coordinates in 3D by stacking the same pixel coordinates in third dimension.

The results of above approach can be seen in Fig. 9 shows the 2D representation of the Central Zone (ZC), Fig. 10 shows the 3D representation of the Central Zone (ZC), Fig. 11 shows the 2D representation of the Peripheral Zone (ZP), Fig. 12 shows the 3D representation of the Peripheral Zone (ZP). Fig. 13 shows the 2D representation of the Tumor Zone (ZT), Fig. 14 shows the 3D representation of the Tumor Zone (ZT), Fig. 15 shows the 2D representation of the full prostate and Fig. 16 shows the 3D representation of the full prostate.

But, we were **not satisfied** by the results so we proceeded to second approach.

Note: This code is not added to the GUI but you can access the m-file explicitly to check this approach. 'draw3D.m' file will execute this approach.

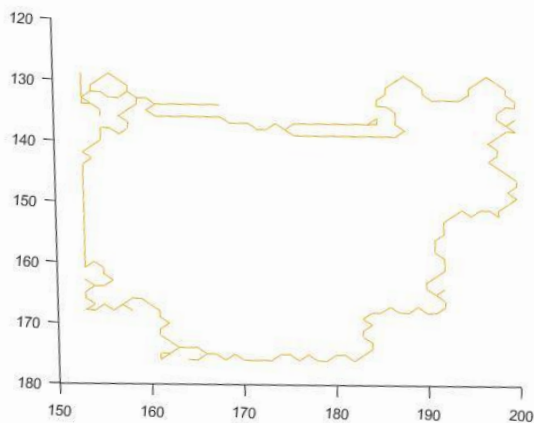


Figure 9: 2D view of ZC

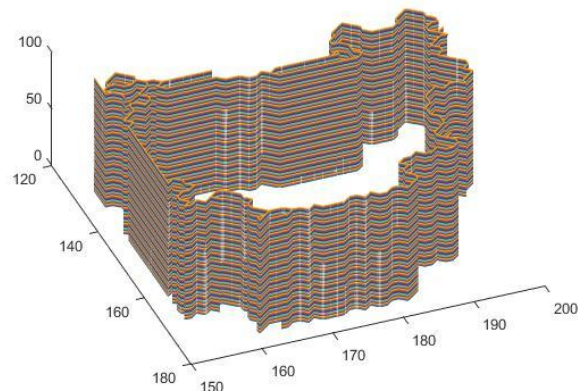


Figure 10: 3D view of ZC

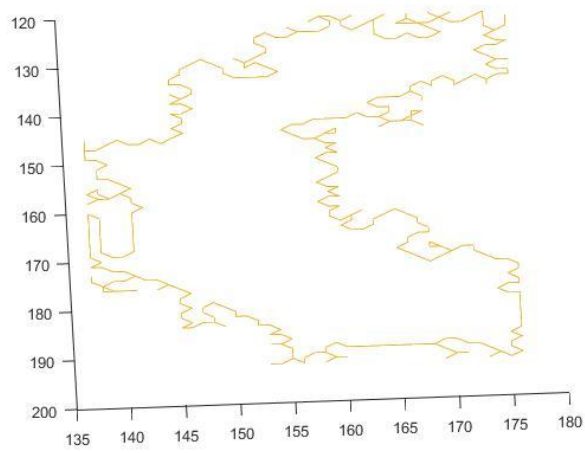


Figure 11: 2D view of ZP

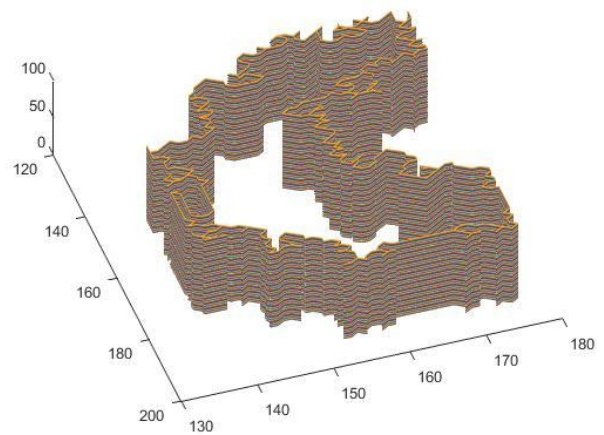


Figure 12: 3D view of ZP

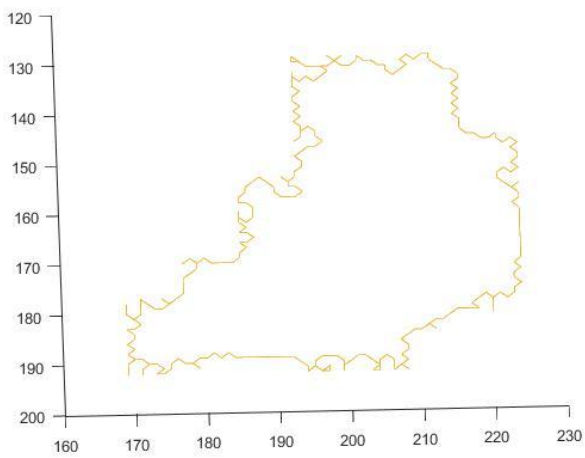


Figure 13: 2D view of ZT

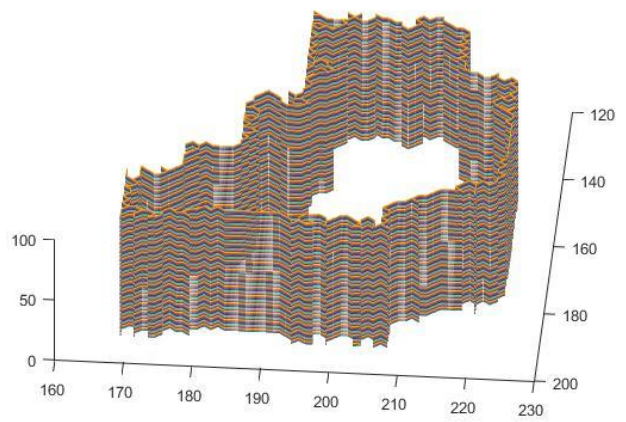


Figure 14: 3D view of ZT

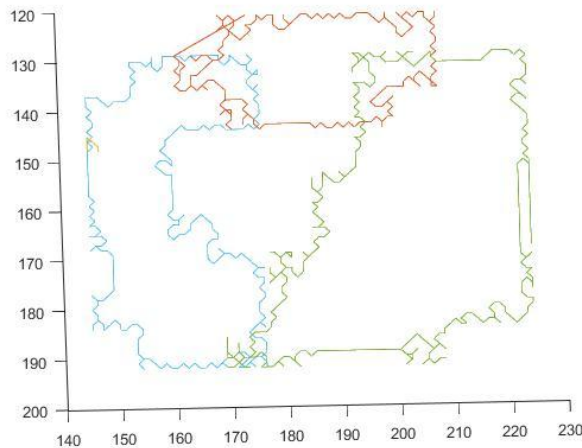


Figure 15 2D view of full prostate

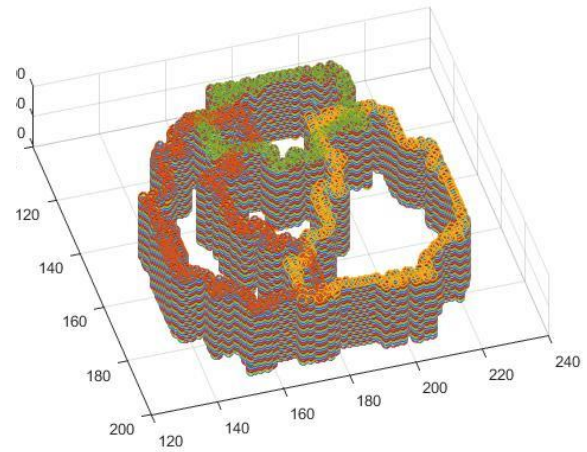


Figure 16 3D view of full prostate

Approach 2: Using many Slices

We manually segmented some of the slices for regions and full prostate gland, and saved in the folder 'All Slices' in cwd. We use "3D View" panel in The GUI as shown in Fig. 17 for this approach. The "load segmented" button is used to load all the slices which we segmented earlier as said before. And "Show 3D" button is used to show the 3D model generated using the selected slices. here, our approach is to use the 'surface' functions of MATLAB. As a pre-processing step we smooth the frames using '*smooth3*' command by using Gaussian kernel. After preprocessing we use '*isosurface*' to create an iso-surface for the selected frames and then extract patch for these surface and form '*isonormals*' for this patch. We set some properties for the patch to get a proper display and lighting. This approach is coded in 'drawSurface.m' file.



Figure 17: 3D View Panel

The results obtained by this approach can be seen in Fig.18 - 20.

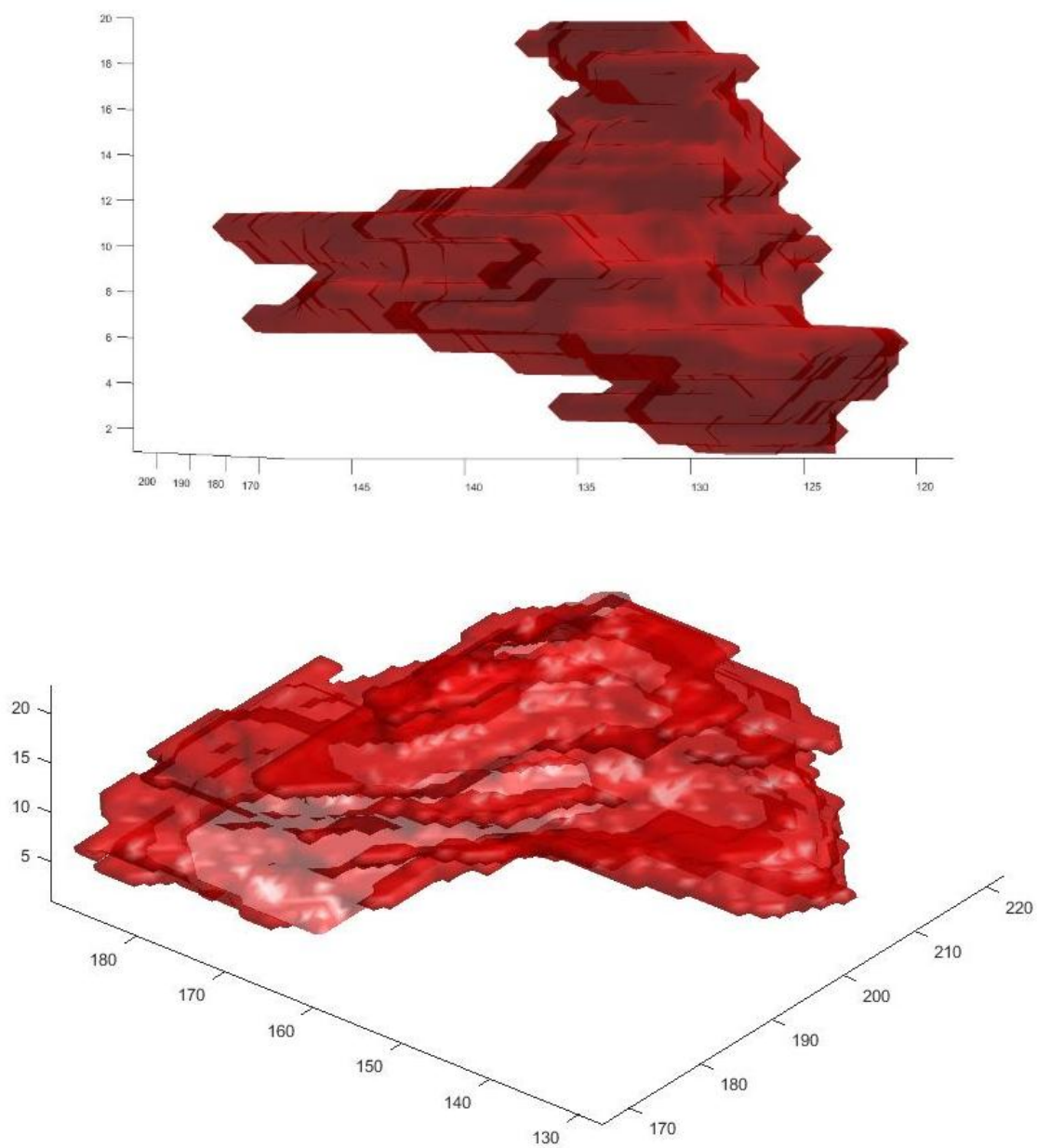


Figure 18 3D view of Tumor regions using approach 2

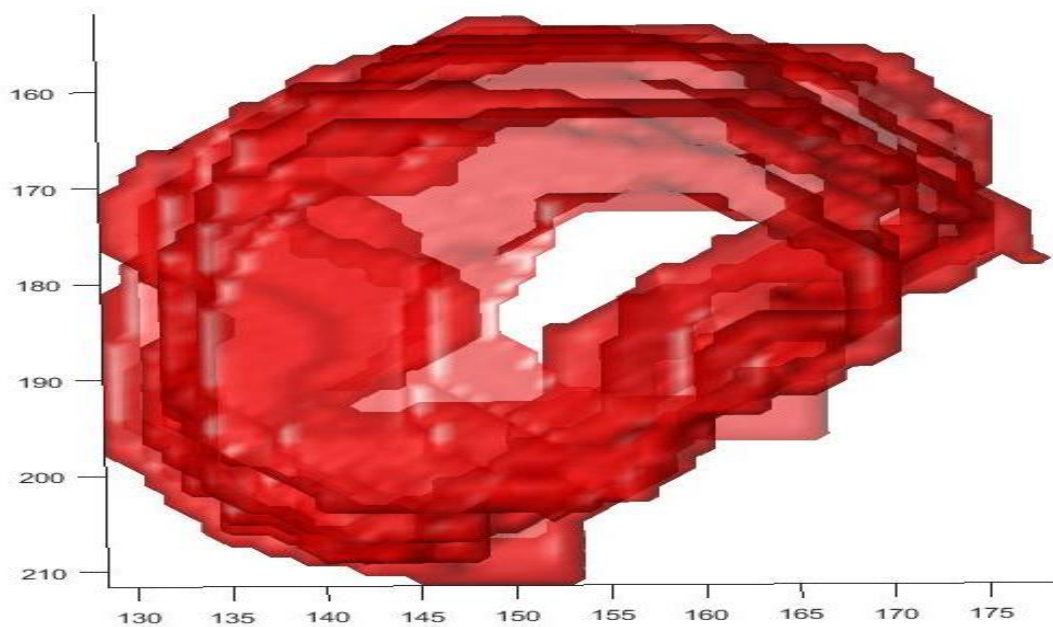
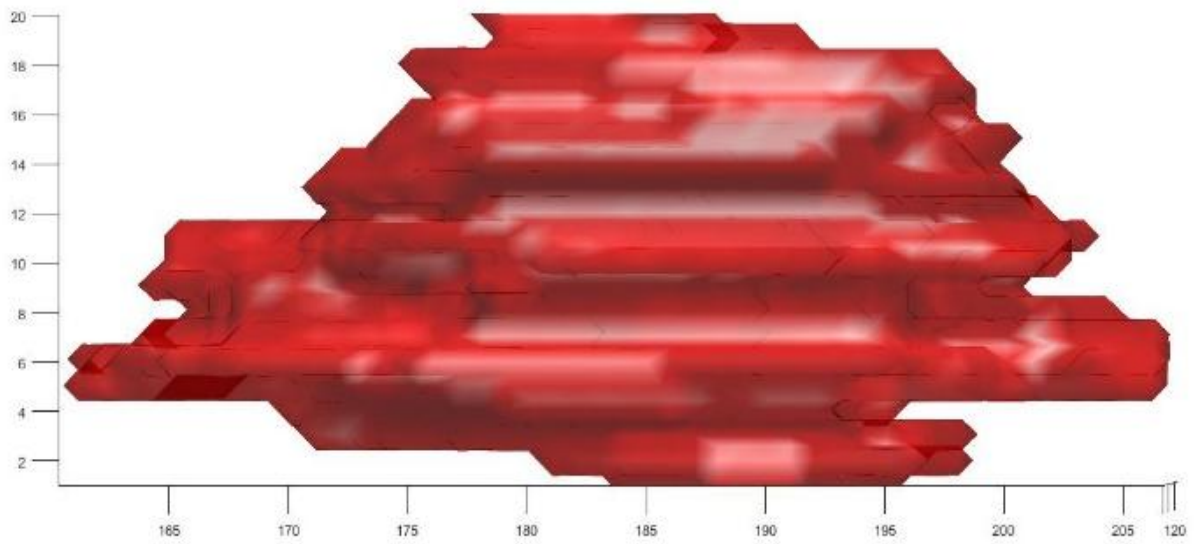


Figure 19 3D view of full prostrate using approach 2

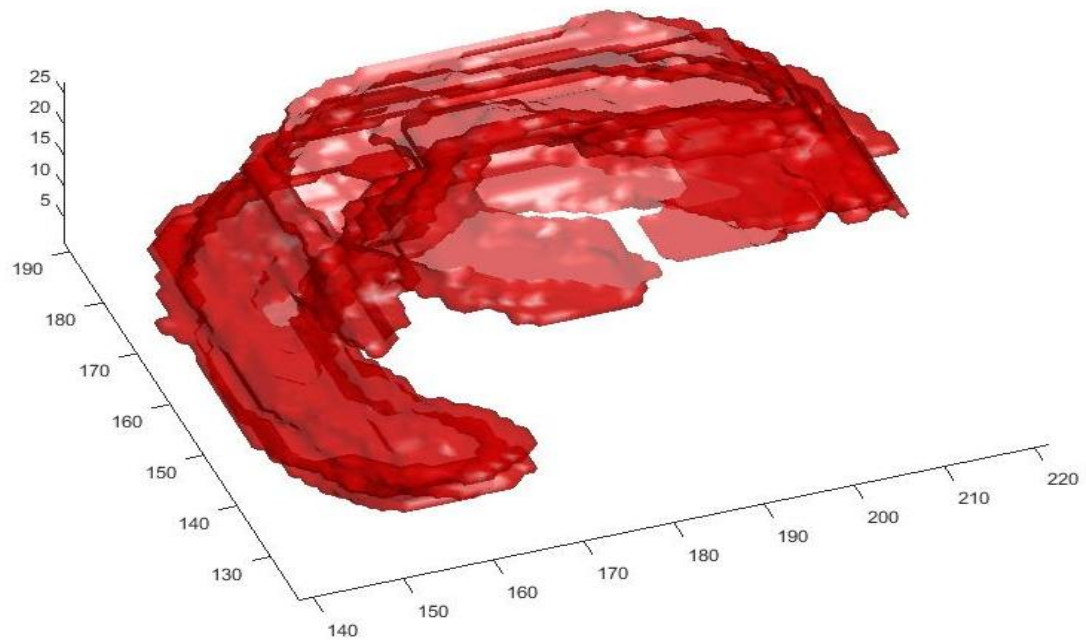
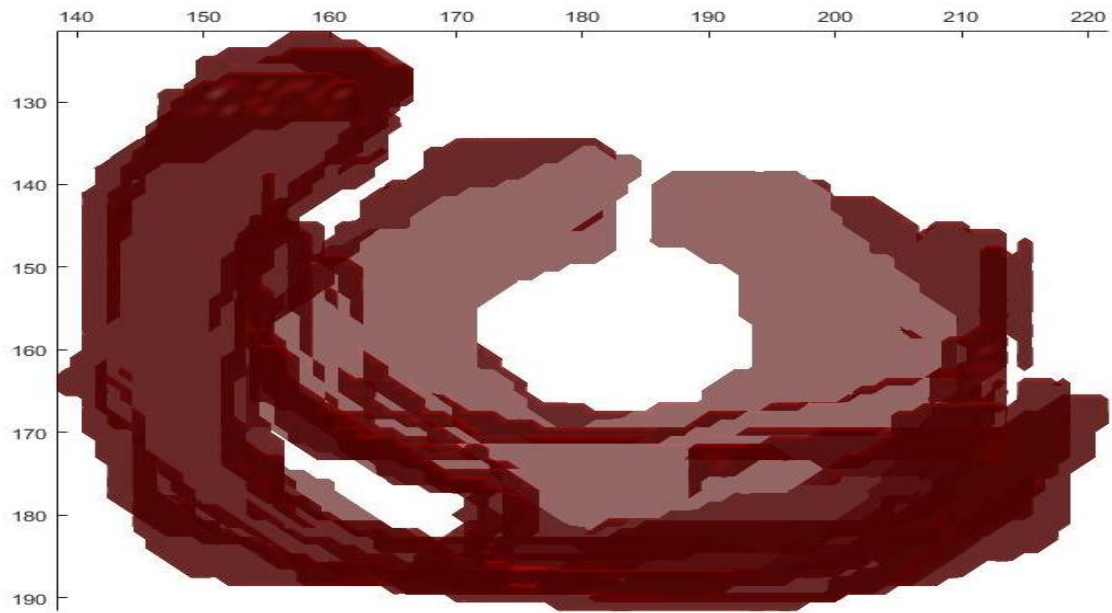


Figure 20 3D view of peripheral region using approach 2

Note: The results with approach 2 cannot be clearly seen in above images, but we can load the images(in "3D View" panel) from 'All Slices' folder in cwd and click "show 3D" button to visualize the 3D view clearly. This "3D view" panel can be accessed independent of other panels, so you can directly open gui and load images and view 3D reconstruction.

3.3 3rd stage

In order to calculate the surface and volume we should use the DICOM information.

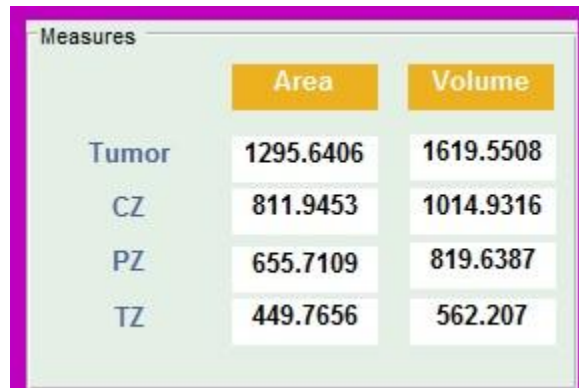
- **For each region calculate the surface (Surface = number of pixels x spatial resolution)**

Solution: Here, we use the segmented region information of result of task 3.2 We use 'regionprops' command to define region properties object. By using this object we call 'area' function to get total number of pixels in the region. After we get number of pixels we multiply this with spatial resolution information obtained from metadata to calculate area of each region. The 'calculateArea.m' file does this task.

- **Calculate the volume for each region.**

Solution: Here, we multiply the area we got from previous step with slice thickness information obtained from metadata to calculate volume of the region. The 'calculateVolume.m' files does this task.

The area and volume information is shown in text boxes in GUI as shown in Fig. 21.



	Area	Volume
Tumor	1295.6406	1619.5508
CZ	811.9453	1014.9316
PZ	655.7109	819.6387
TZ	449.7656	562.207

Figure 21 Area and volume in GUI

4. Conclusion

We have performed the post processing of MRI images of prostate gland by accessing the metadata and converting between image formats and anonymizing the data which are basic steps and followed by manual segmentation of regions of prostate gland and representing the regions in 3D view by reconstruction and also calculated the area and volume of the segmented regions.

We have learned some useful image processing tools like manual segmentation and created binary image of the same. We have also learned 3D reconstruction using iso-surfaces nad by using iso-normals. We were also able to access the region properties for the segmented images.

References

- [1] website : <http://www.cancerresearchuk.org>
- [2] Lecture slides of Dr. Mata
- [3] Dicom wikipedia
- [4] Mathworks Central