



UNIVERSITÉ DE BOURGOGNE

Masters IN COMPUTER VISION AND ROBOTICS

---

## **Medical Image analysis**

Management and post processing of prostrate MRI

Lab report

---

Muhammad Zain BASHIR

Sepideh HADADI

Centre Universitaire Condorcet - uB, Le Creusot



May 2017

## 1. Task at hand

The task given was to design a graphical user interface using MATLAB which can handle DICOM MR images, display metadata, anonymize metadata, and convert the DICOM images into JPG and from JPG back to DICOM. The GUI should then be able to manually segment the prostate gland and its different regions (Central zone, Peripheral zone, Transition zone and Tumor) and show a 3D representation of the regions and then calculate the volume of each region.

## 2. Graphical user interface

The figure given below (*figure 1*) is a snapshot of the designed GUI with all its features labelled.

- *Load buttons:* These buttons are used to load the images to the List box. The buttons to load JPG format or DICOM format are given separately. Upon clicking either of the buttons a dialog box appears and the user can browse his hard drive to look for image files.
- *List box:* Displays the names of all the image files that the user has selected.
- *Image viewer:* Displays the current DICOM or JPG image the user has selected from the List box.
- *Metadata:* This fields shown here display the relevant metadata for the patient. The fields are labelled in the GUI.
- *Anonymize patient:* This button allows the user to anonymize patient data for the selected patient. It specifically anonymizes the fields 'Patient name', 'Patient ID' and 'Patient Birth date'.
- *Anonymize all:* This button anonymizes all patient being displayed in the List box.
- *Convert to JPG:* This button only functions when DICOM images are loaded on the list and converts all the loaded images to '.jpg' format at once. On clicking, a dialog box appears and asks the user the folder path to store JPG images. Along with converting to JPG it also stores all the metadata for that image to a MATLAB variable named the same as the patients' name and in the same folder as the JPG image.
- *Convert to DICOM:* Only functions when jpg images are loaded. This button converts all the loaded images to the DICOM format at once. The DICOM info for each image is read from the corresponding MATLAB variable that stores the information.



- *Segment checkbox:* If the checkbox is selected it turns on the segmentation mode. The user can segment the regions of interest using free hand drawing. The user however needs to select the region for segmentation from the drop-down list.
- *Choose region drop down list:* This drop-down list asks the user to choose the region to segment. Once the region is chosen, it equips the user with a different colored freehand pointer for each region. The regions to choose from are: Central zone, Transition zone, Peripheral zone and Tumor zone.
- *Calculate volume:* Calculate volume button calculates and displays the volume (in  $\text{mm}^3$ ) for different segmented regions. The volume is displayed in the bottom right corner.
- *Show 3D:* This button displays a 3D model of the segmented structures.
- *Change image buttons:* These buttons load images in increasing or decreasing order as being displayed in the List box.

### 3. Approach used

#### 3.1. For the first stage

To display DICOM information, the info was first read using the ‘dicominfo’ command and the relevant fields were accessed using the dot operator and displayed in the text box. For example if we stored the DICOM info in a variable called ‘info’, we access the patient data by just calling ‘info.PatientData’. Similar approach was used to anonymize the images. Each field to be anonymized was accessed individually and edited. After anonymization, the information was stored using the ‘dicomwrite’ command.

For conversion to jpg, the image and the DICOM info were read first and then using the ‘imwrite’ command the image is written in JPG format in the specified directory while the corresponding metadata is stored in a MATLAB variable using the ‘save’ command. For conversion back to DICOM, the image is read using the ‘imread’ command and the corresponding metadata is loaded. The image and the information are then combined to store it as a DICOM using the ‘dicomwrite’ command.

#### 3.2. For the second stage

For segmentation of each region, an *imfreehand* object is initialized for each region as specified by the user in the drop-down list. The user then segments the image region according to his understanding while the ‘createMask’ function binarizes the segmented region. Using the ‘bwboundaries’ command on the binary image we calculate the boundaries of the segmented area. When all the slices have been segmented, we our regions of interest from all the binary images. These binary slices are then used as an input to the ‘isosurface’ command which

makes each slice a surface of a given thickness while the 'patch' command joins all these surfaces to make a 3D representation. This can be done for all regions.

### 3.3. For the third stage

The third stage involved calculating surface area and volume for each slice and then the whole 3D figure. For each slice, the number of pixels are calculated using the area property of the 'regionprops' command. Then from the dicom info for each slice, the pixel spacing is retrieved to calculate the actual area for each pixel. The surface area is then calculated as a product of actual pixel area and the number of pixels. This measurement is converted to slice volume by simply multiplying the surface area with the thickness of the slice, again retrieved from the DICOM information of the image. For each region, the volume of each slice is calculated and added up to give the total volume of the region. This can be repeated for all regions.

## 4. Results

After manual segmentation of different regions, the 'Show3D' button gives the user an approximate 3D model of the segmented region. The figures following are the 3D representation of different regions.

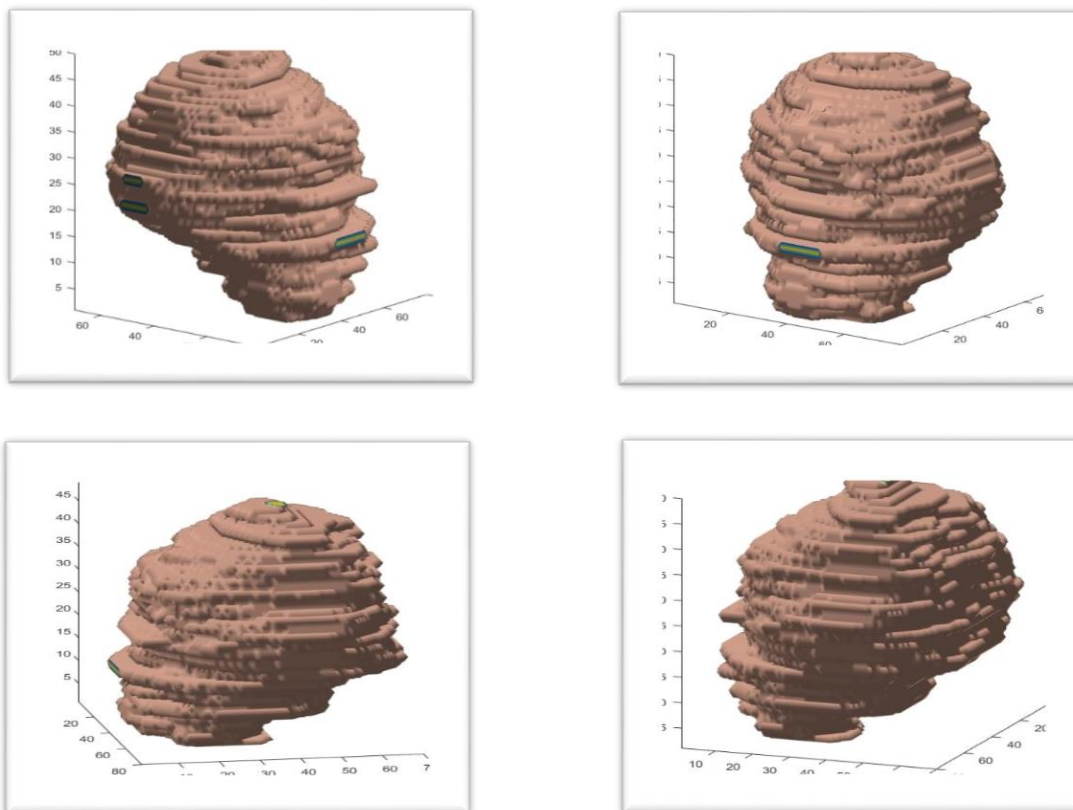


Figure 2: 3D representation of the whole prostate gland as viewed from different angles



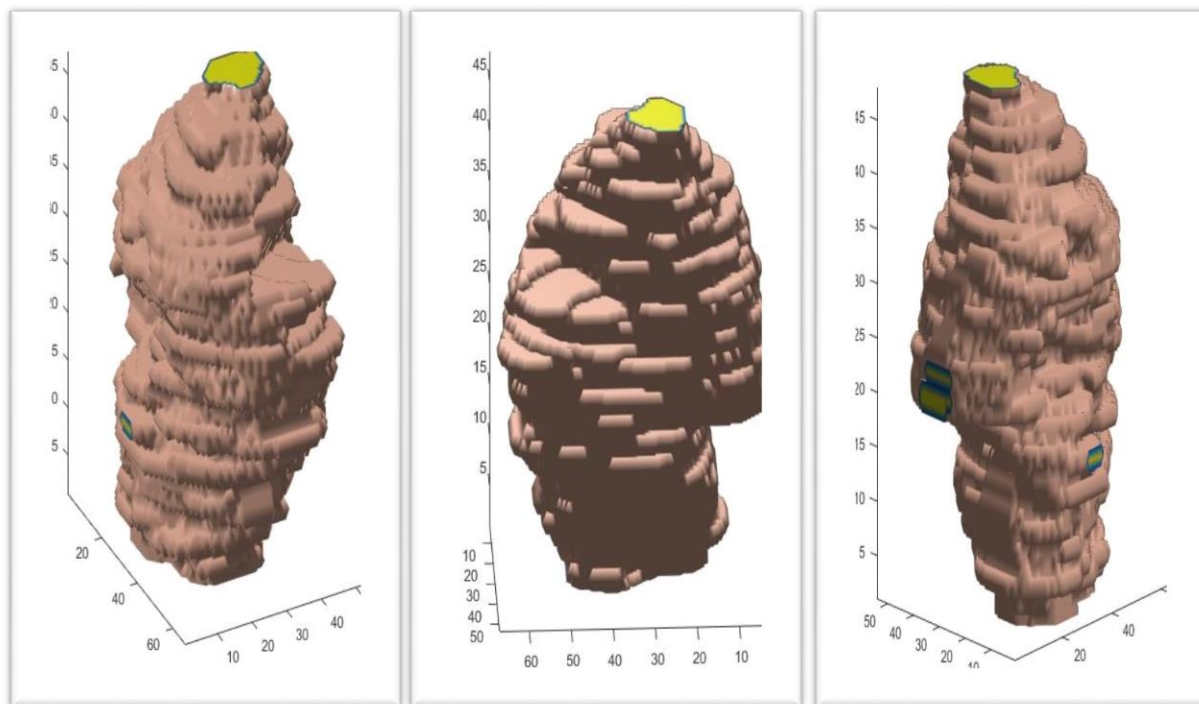


Figure 3: 3D representation of the whole central zone as viewed from different angles

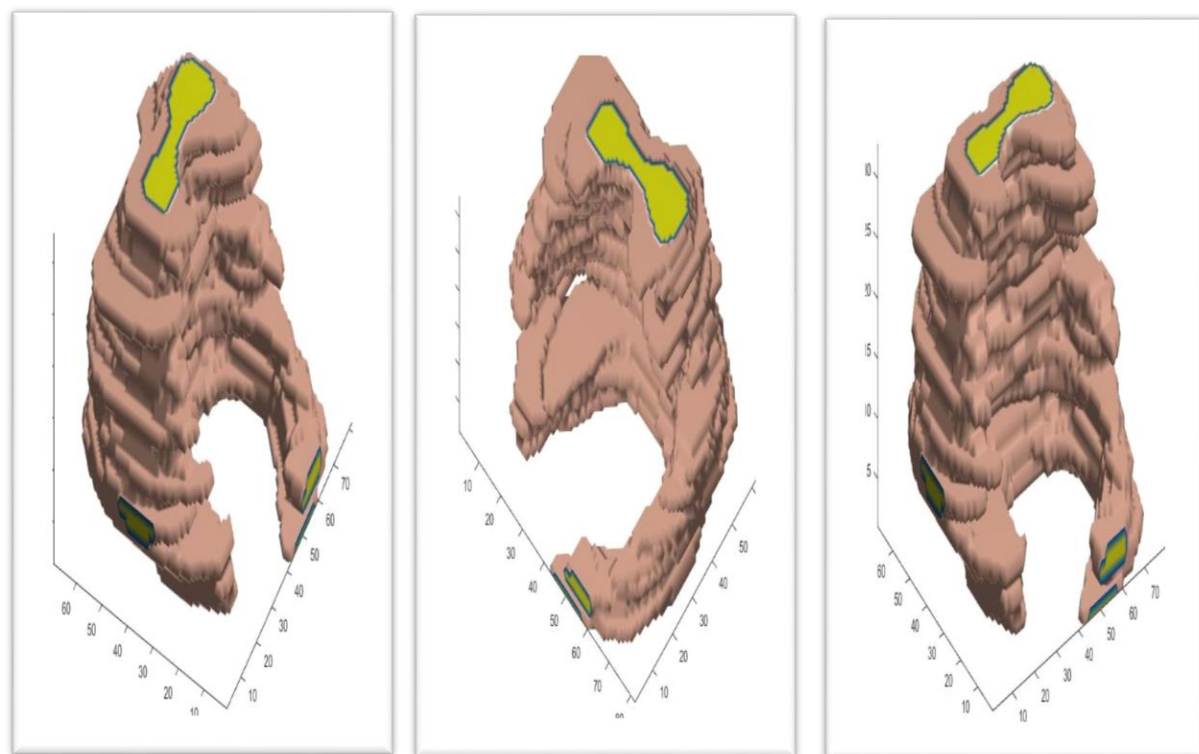


Figure 4: 3D representation of the whole peripheral zone as viewed from different angle

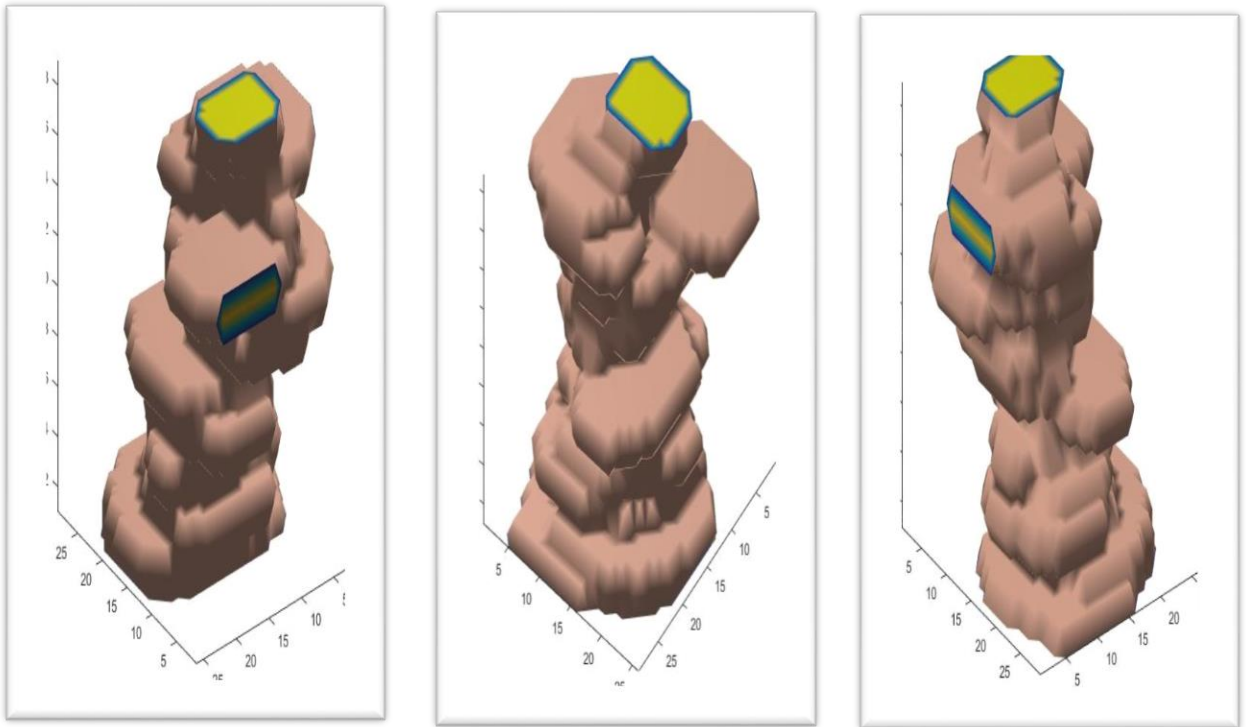


Figure 5: 3D representation of the whole Tumor as viewed from different angles

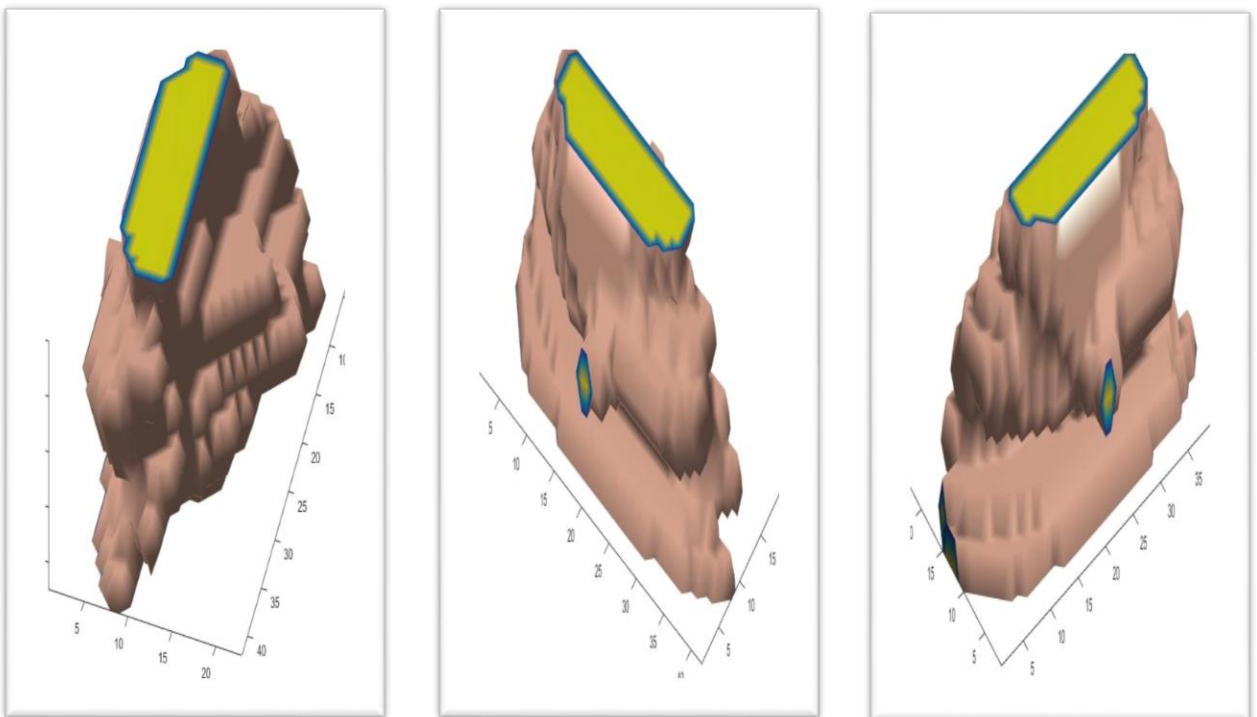


Figure 7: 3D representation of the Transition zone as viewed from different angles

#### 4.1. Calculating Volumes for each region

To calculate volume of a segmented region, the surface area for each slice was calculated first. Matlab's command 'regionprops' helps calculating the area of a binary image in number of pixels. Dicom info was accessed to calculate the area of each pixel. The information was used was 'pixleSpacing' The surface area was then calculated as:

$$area \text{ (in mm)} = \text{number of pixel} * \text{area of one pixel}$$

Once the area of one slice is calculated, it can then be multiplied by slice thickness to obtain the volume for that slice:

$$Volume \text{ of one slice (in mm}^3\text{)} = area * \text{slice thickness}$$

Where slice thickness in our case was 1.25 mm.

Finally, the total volume of a region was calculated by adding up the individual volumes for each slice. The table below shows the volumes of each region.

| Region             | Volume (in mm <sup>3</sup> ) |
|--------------------|------------------------------|
| 1) Prostrate       | 82963                        |
| 2) Tumor           | 4475                         |
| 3) Peripheral zone | 31441                        |
| 4) Transition Zone | 1835                         |