Medical Imaging Report

Blanchon Marc - MsCV

Summary

Medical Imaging

Subject : Management and post-processing of prostate MRI

Medical Imaging

Abstract

In this report, we will talk about the development of a GUI in MATLAB to deal with DICOM images and process them in order to do segmentation and 3D modelling of prostate MRI.

In fact this subject can be extended as well till we are dealing with MRI and DICOM images, we can use this GUI to manually segment any images.

In order to be concise we will talk about general knowledges and main difficulties to develop the program and we will avoid going too deep in trivial subject such as basic knowledge of Matlab gui or functions.

UNIVERSITÉ DE BOURGOGNE

1st Stage

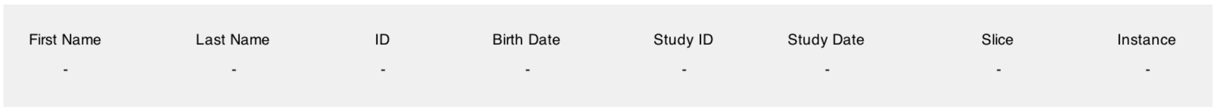
In this part we will explain the way that was used to deal with DICOM images, to anonymise the images and the choices made to develop such a process.

Display Images and informations

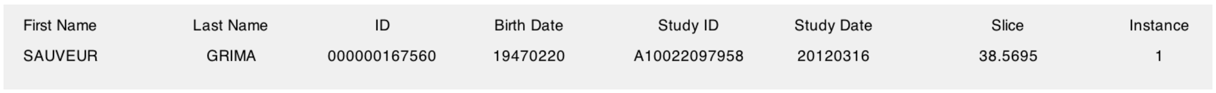
In our case we have to remind that DICOM is images that contain useful informations about patient and also the condition of capturing the MRI.
The first task was to display the image and the informations that were present inside the DICOM informations of each images.

After choosing and designing the gui with an image display, a list and the texts for the informations display we had to develop the browser, previous and next button and make a choice about displaying.

First we can see that the labelling is not modifiable, and this is a choice. We could think of a process that allow us to change the data of the DICOM format, and in future parts to be able to add information to a JPEG image in order to become a DICOM image. This behaviour haven't been developed, it can be a problem that we can't modify images information, but as the choice were made, the less the user can act on the fundamental data (images and labels) , the better it is, like this we also avoid not allowed characters, bad manipulations ….



First Name	Last Name	ID	Birth Date	Study ID	Study Date	Slice	Instance
-	-	-	-	-	-	-	-



First Name	Last Name	ID	Birth Date	Study ID	Study Date	Slice	Instance
SAUVEUR	GRIMA	000000167560	19470220	A10022097958	20120316	38.5695	1

For the image, the display is only made by using a viewBox that will be used just to display and draw on our gui.
Basic implementation of arrow to switch from one image to another have been developed but we absolutely take care of avoiding errors, so the very basic and necessary behaviour have been developed (avoidance of index less than 0 and more than the maximum image given by the user).

The last part but not least is the browser. In this case we took care of lot of exceptions. The main problem was to deal with DICOM and JPEG images, multiple or not. The problem is that we are using the same process to display JPEG or DICOM (imshow) but the extraction of information is not the same. So in order to be able to deal with any number of DICOM or only one jpeg, there is the usage of lot of if's statements, to know the type and the number.

The main problem is this code is that we can't input more than one JPEG. This is a problem that occur because of our first if statement to know which format we are using, we are also looking at the length, and to be concise, instead of overloading the code with multiple statements, the idea was to make the possibility to enter one or more DICOM but not more than one jpeg. And like this the code is lighter and understandable.

Obviously, the main idea of a browser is to search files, and Matlab does the work for us and provide lot of 'teminal-like' functions that avoid the reinvention of the wheel, we just have to add our parameters and the specifications we want and then Matlab is acting like a library doing the work in the backstage.

To end with this part, we can notify that we are using a formatting display for the JPEG. In fact, we have no information about the patient, so we are just filling the form by a character.
As we can guess for the following tasks, we had to convert a JPEG to DICOM and till we didn't implement the possibility of adding informations, we can already guess that this function is not present in the code.

Anonymise the images

The idea is anonymising the images is pretty simple, we remove all the labelling in the DICOM and we remove the display of informations.
We create a new folder to contain anonymous DICOM images and we change the work folder to this new folder.
This is exactly what is done. One more time with the 'terminal-like' functions of Matlab this task is made easier, but something is missing in Matlab : the regular expressions. So to avoid duplicates and erasing of folders, the first idea was to change the name of the folder each time we are creating a new anonymous folder.
The program turned in another way, instead of changing the folder, we are creating a new folder inside the previous folder. So the biggest child of the first Anonymous

folder is the last folder. Like this with the usage of condition, we are able to avoid erasing of data.

Conversion

For this part we will talk only about the half of the subject. As mentioned before, we didn’t succeed to convert a JPEG to a DICOM and we will first talk about this problem.
What makes the DICOM format? The informations and the image. It is simple to imagine extracting the informations that the user input and add it inside our new map of elements in order to recreate a DICOM. Matlab has functions for conversion of JPEG to DICOM and vis versa.
To be honest, in the primary approach we had labels that was modifiable and we were able to extract the informations and the image.
The problem is that for the function that Matlab provide we have also to provide the path and till we are always updating the path, moving forward and backward to avoid copy / erase / loss of information with the process explained in the anonymisation of images, we were not able to recover the exact path and only the image name and without the path, we can’t use this process. So to move forward and continue the gui coding, the choice were made to make this function unavailable.

For the DICOM to JPEG, in this case the process is easy, we can easily recover the path of our DICOM folder till this is the basic one (the one where we start), after this we use the function and this separate features from image and just take the image. Optimization problem is a concern for this project, a commentary have been written in order to discuss the choice of the parameters of the conversion function.

2nd Stage

Manual Segmentation

The manual segmentation looked the most easy function of the code but this was the most difficult one, and there is one reason, dimensions.

Matlab allow us to draw on any image using one simple process imfreehand(). Ot give the possibility to draw a line and save it.
The first approach was the brute force, extract data from the drawing, extract X and Y coordinates of the line (closed by the function also) and then we were able to do it for the 4 applications and process any calculations.

It turns out differently. After this part we will see the 3D modelling part and the biggest problems comes from this part. We developed 3 strategies and we will explain them. But before we need to explain the root of the dimensions errors. In the software the choice were made to being able to draw and redraw without problems, and also move from one image to another to draw another region. We were able to store in the handler as we did every time but what ensure us that the line we draw is of the same length, contain the same amount on points?

Strategy one : Create after each button push a matrix of zeros

This strategy was working using the size of our line, but when we were switching to another image, or redrawing on the same image we had a new problem and it was not dimension. At the end when we were modelling the 3D points, we saw that all the previous points were erased and became zeros. The storage was corrupted by the zero matrix at each push because the size was different to each time we recreate everything.

Strategy two : Reshape the matrix at each push of the button

This strategy were theoretically very good but in Matlab that was not the case. In fact make a matrix copy itself and resize was not an accepted solution by the compiler and gave errors so this way was forbidden very fast to end up with a new solution that combine the two firsts strategies.

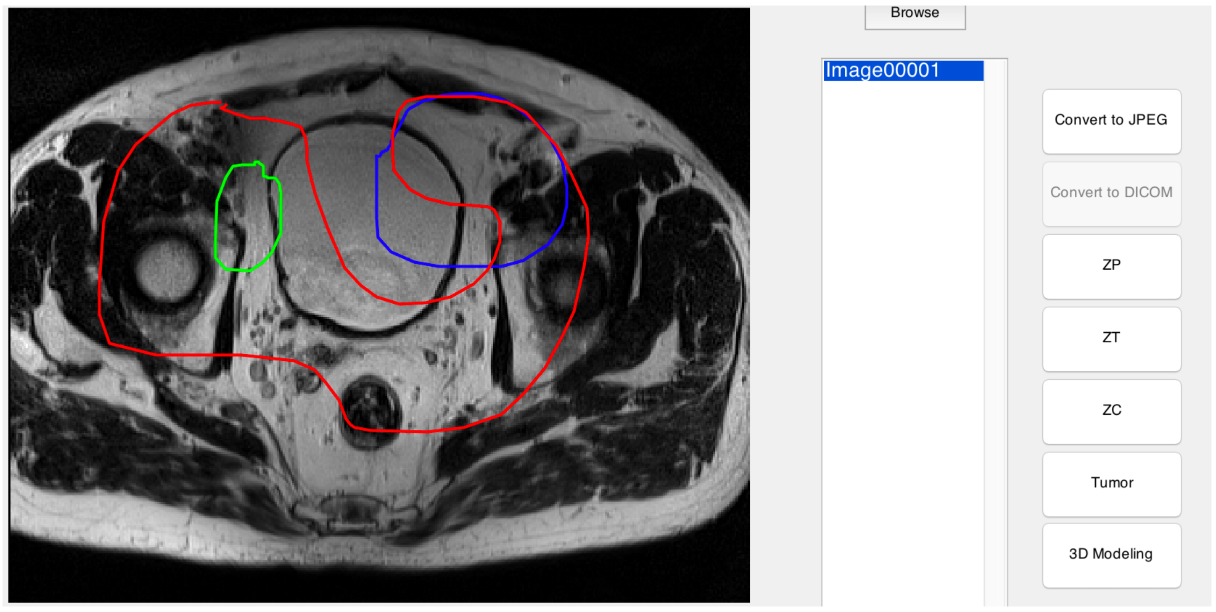
Strategy three (choose) : At each browse button push, create a very big matrix that can handle all the possible datas.

This is a very bad manner of developing but in fact this was the only solution that gaves good or sufficient results for our application. The problem is that we create a

size changing matrix of zeros (depending on the number of input images). We choose matrix of n column by 1000 values.
Like this we are able to change the desired values between 1 and the number of points for each column.

At the end we have a matrix of X and Y but of 1000 values, probably 15 or 40 are real values and the reste zeros.
Why 1000 values, because if someone have a mouse bug, or stay clicked and do a complex shape on the screen, we want to be able to take care of this case. The main idea is to never trust the user and ensure that the software is stable even if it is for scientific.

Till our computers are powerful we could extend it more and more without any problem of memory.



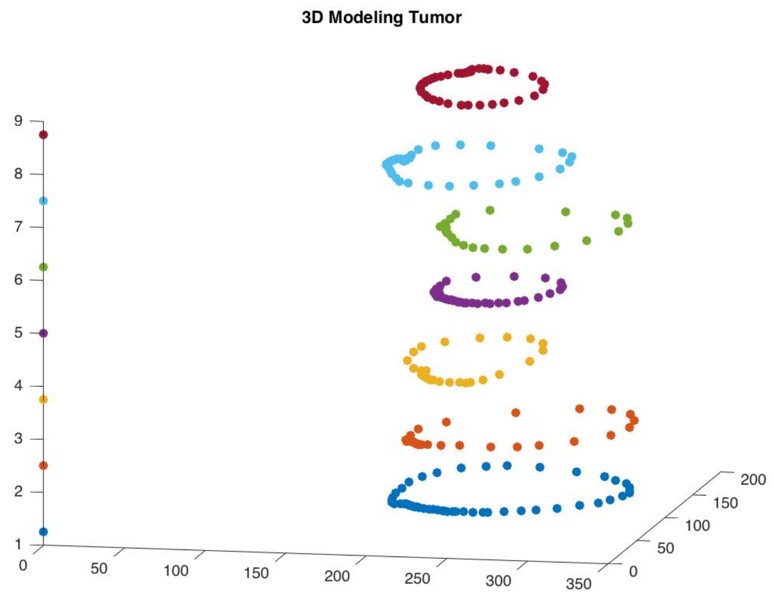
3D Representation

In this section, the most difficult part were discussed previously. For the 3D Modelling, the choice were made to not render / mesh and there is a reason, the control. In fact we always can use Matlab functions that we don’t know the process, the behaviour but for meshing and rendering in 3D with textures, this was not possible. In order to complete the work, the choice were made to use scatter3 and display the points that were drawn before by the user. It gives certainly a

rough idea of the 3D model, this is far very far from perfect but it gives a 3D idea of the tumor.
To be able to use volume, we used a multiple of the slice thickness to give the perspective.

We can discuss about improvements on this part, in fact lot of process can be used to have something smoother, better, more realistic, one easy process is to draw lines to do edges and draw line to the nearest point on the previous and next slice. This is just the computation of euclidian distance and comparison.
Another way is to build a 3D model using the inside of the draw of the user.
Another and last explained is maybe the best one. We trust the user accuracy and we say, for each consecutive slices, take a increment value (smallest is best) and link 4 points (2 from the working slice and 2 for the next), and draw smooth parallel lines between this 2 pairs of two points.
Using the increment value we will decrease, increase of stabilise our depth and we will fit line to the next couple of points.

For the result of the developed program, we can see a problem, lot of points in 0 coordinates, this come from the matrix initialised and discussed before. Till this is the only solution found of this dimensionality problem, we have to ignore these points and no doubt that it can be disturbing.



3rd Stage

Surface and volume computation

For this part, all the hints were given. In fact we had just to read the lab subject, extract the hinted value labels and then use our process in hand drawing by the user, extract the XY coordinated, compute the relative area and multiply by the pixel spacing and then multiply this result by the slice thickness.

This behaviour happen every time someone draw on the image, everything is displayed, from the features values, to the surface, passing by the relative area value.

Conclusion

In this project, we successfully developed a gui with Matlab that allow to read write images, take care of event, segment an image using manual segmentation, compute area and volume of a selected zone according to the space and to finish, draw a 3D model from the drawing.

As explained in the abstract, we can extend this behaviour in lot of fields, or even subject to stay in medical imaging.

This project was interestingly learning the GUI management and developing in Matlab through a topic that we are studying. I think that we had lot of class about segmentation so this was not the main goal but learning matlab.

References / Sources:

Stack Overflow
MathWorks