

# Computer Exercise Medical Image Visualization :

## Medical Image Visualization and Processing

November 15 2012

### Objectives

- Discover the Scientific Visualization Tool Vtk
- Test several 3D medical data visualization methods
- Apply scientific visualization technique to medical data
- Create your own medical image viewer application

## 1 Environment Installation

This project use the following free, open source and multi-plateforms software tools :

- The Graphical User Interface library [Qt](#)
- The `Makefile`/Windows/Unix/Mac projects generator [CMake](#)
- The scientific data visualization C++ library [Vtk](#)

### 1.1 Qt Installation

Normally, Qt libraries are already installed on the Computers.

### 1.2 CMake Installation

Check that `CMake` is not already installed on your computer.

If it is not installed, make a local installation :

- Create a `local` directory in your home directory
- Follow the instructions at [this address](#)

### 1.3 Vtk Installation

Get the last version of VTK at [this address](#) and compile/install it carefully checking the option `USE_QVTK` in your local directory.

### 1.4 Test

Get the example code archive [ExampleMedical.zip](#).

Use CMake to create a projet (e.g KDEvelopp project) with the given sources in the `build` subdirectory.

Compile and execute the subprojects `Medical1`, `Medical2` and `Medical3`.



In the following, you can look at the code to help you with your application project.

## 1.5 Computer Exercise Application Skeleton

Download the following [code archive](#) and uncompress it in a proper directory.

Use CMake to generate a project. Compile and execute the project using the data `image01.mha` in the `data` subdirectory.

## 2 Getting Started

### 2.1 Application Skeleton

Execute the program `GICAOImageViewer` and give the first image of the `data` directory as parameter. You should obtain an image which looks like the following Figure :

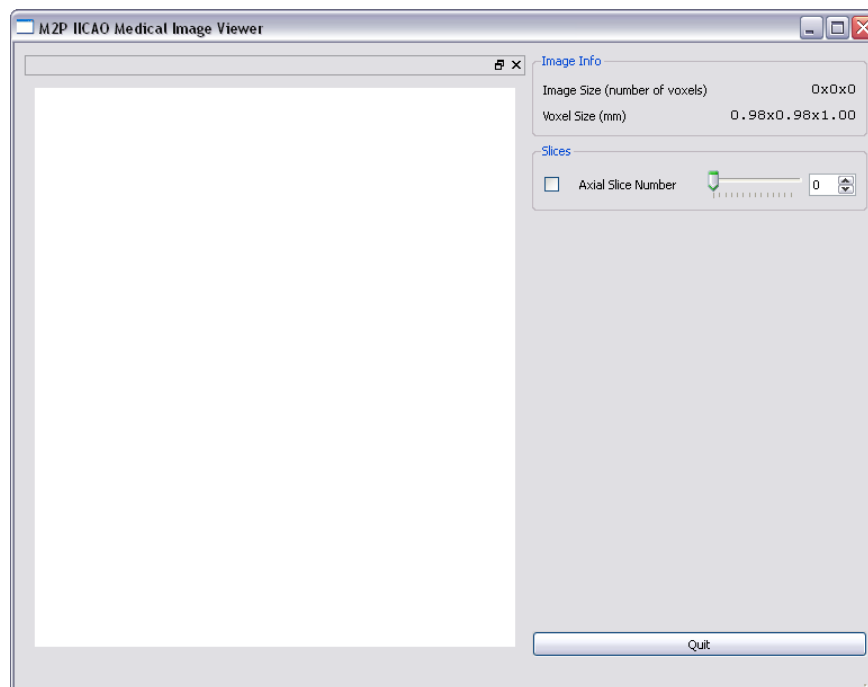


FIGURE 1 – Starting page you should obtain.

You will enhance the the applicaton you will modify the code of the class `MedicalViewer` in the files `MedicalViewer.h/cxx` as well as the Graphic User Interface (GUI) in the file `MedicalViewer.ui` with `QtDesigner`.



The code you will write will mainly be performed with `vtk`. Do not hesitate to use and abuse of the library documentation [here](#) . You will also find example code source [here](#).

The image contained in the file `image01.mha` is a volume image. Open the file `image01.mha` in a text editor.

#### Question 2.1

What information are given in this file?  
Where are the data?  
How are they implemented?

Start the `GICAOImageViewer` application and check the `Axial Slice Number` check-box. Change the slice number (either by the *slider* or with the *spin box*).

#### Question 2.2

What do you observe? How have these images been obtained? How can you perform this observation them (sketch an explicative figure)?

In the file `MedicalViewer.cxx`, you can see that :

- the GUI initialization is made into 2 methods :
  - `ui.setupUi(this)` (Qt method) and
  - `initWindow()` (method of the class `MedicalViewer`) which creates a Vtk window in a Qt frame (if you want to change the background color, that's here!)
- Image reading is made by Vtk in the `readVolumeImage` method and the image volume is referenced in an attribute of the class `MedicalViewer`
- Once the image is read, information about the image format is updated in the `setImageInfo()` method.

#### Question 2.3

What information do the following attributes of the class `MedicalImageViewer` contain : `extent`, `spacing`, `origin` and `center`.

#### Question 2.4

What is the total size of the image in `cm3`?

## 3 Medical Image Slices

The present application only displays axial slices.

#### Question 3.1

Modify the application to allow to also visualize coronal and sagittal slices of the patient. You will thus carefully modify the GUI to display or not these new slices and to choose their numbers.



You may want to add the following methods to the class `MedicalViewer` :

```
- virtual void slotIsCoronalDisplaied( bool )
- virtual void slotIsSagittalDisplaied( bool )
- virtual void slotCoronalSliceNumberChanged( int )
- virtual void slotSagittalSliceNumberChanged( int
)
- void displayCoronalSlice(int sliceNumber)
- void displaySagittalSlice(int sliceNumber)
```

#### Question 3.2

Sketch the corresponding vtk pipeline.

## 4 Medical Image Histogram

Several vtk classe allow to visualize an image histogram as illustrated [here](#).



The class `BarCharActor` cannot represent more than 255 gray levels, which is not enough for the images we will use.

#### Question 4.1

Add to the application a window displaying the histogram of the open image.

## 5 Medical Image Look Up Tables

Open the image `image02.mha` with the application

#### Question 5.1

Why does this image appears black ?



Look at the image histogram.

#### Question 5.2

Modify the application so that the user can choose a minimum and a maximum gray level for the `Look Up Table`.

#### Question 5.3

What does the `image02.mha` represent ? How was it obtained ?

#### Question 5.4

Modify the application to be able tu use different *colormaps*.

## 6 Medical Image Isosurfaces (Project Only)

Observe the `Medical3` code example (in the files given in the `EexampleMedical.zip` archive to test your installation). How are the isosurfaces handled with `Vtk`?

### Question 6.1

Draw the `Vtk` pipeline allowing to represent an isosurface for a given gray level.

### Question 6.2

Add a *checkbox* to your application allowing to visualize or not an isosurface; as well as a line allowing to enter the isosurface gray level.

### Question 6.3

Show separately the skin and the skeleton of the patient on image `image01.mha`.

### Question 6.4

On which eye does the patient of image `image01.mah` wears a patch?

### Question 6.5

Is it possible to represent only the brain of the patient of `image02.mha` in the same way?  
Why?

## 7 Medical Image Volume Rendering (Project Only)

### Question 7.1

Explain what is volume rendering by ray-tracing and how it works in a few lines .

A `Vtk` volume rendering example is given [here](#) (the language used in this example is `tcl` and not `C++`, but it is clear enough to understand how it works even if you do not know `tcl`. Beside, the `Vtk` pipeline is easily recognisable).

### Question 7.2

Draw the `Vtk` pipeline allowing to represent the ray casting of a volume image.

### Question 7.3

Add to your GUI buttons allowing to enter 4 points of the color and opacity points of the ray tracing functions.



Your new Qt widget may look like this :

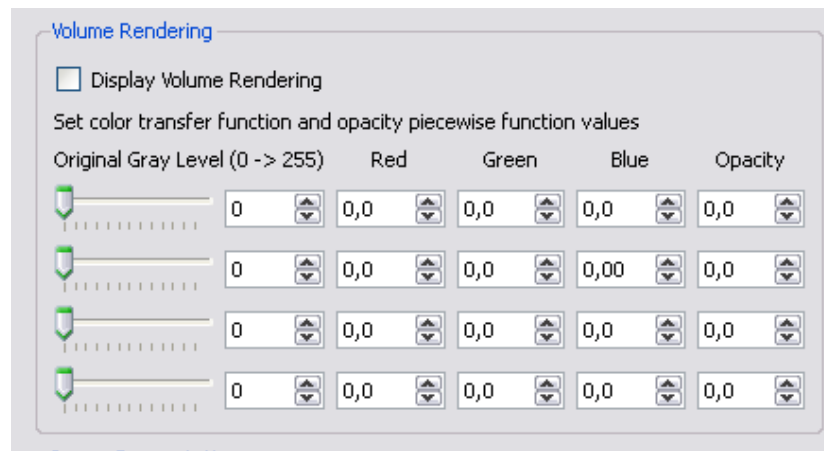


FIGURE 2 –

**Question 7.4**

Show the images `image01.mha` and `image02.mha` thanks to volume rendering.

Draw the color/opacity curve of the points you used.