

# Segmenting with ITK (II)

## Some notes

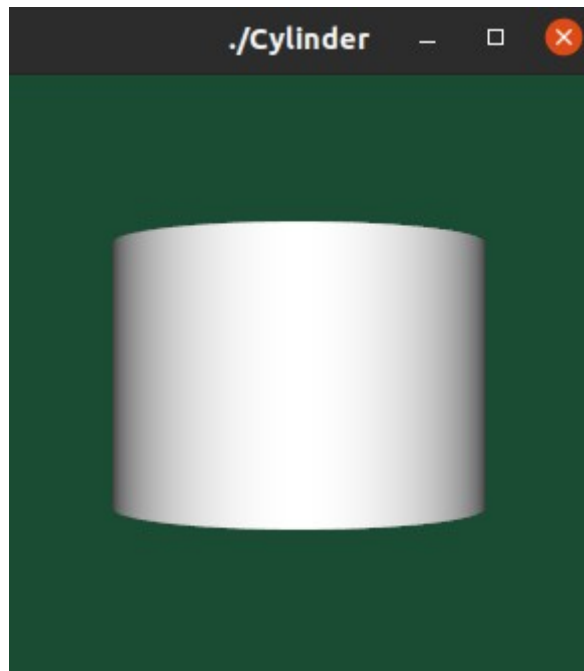


UNIVERSIDAD  
DE GRANADA

## Task 1

**Compile and link the program `cylinder.cxx` included in the `code.zip` file. The important thing here is to read the above paragraphs together with the source code for understanding the two pipelines and the connection between them by means of the `SetMapper()` method.**

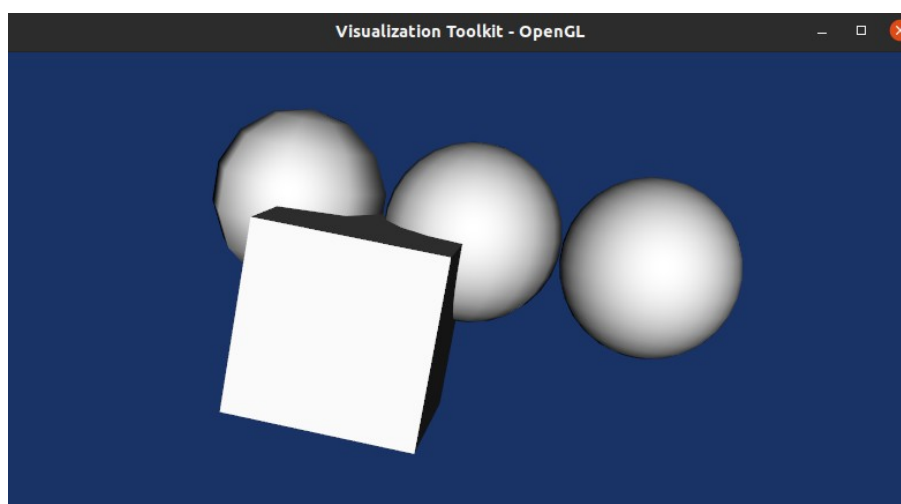
Tras revisar el código y entender su funcionamiento, hicimos el proceso habitual de incluir la conexión a vtk, compilar y ejecutar y este fue el resultado.



## Task 2

**In this task, we have a more elaborated example of using primitive (procedurally generated) models. We use an instance of `vtkSphereSource` class and try out different setups. Next, the program shows a method for manually creating a cube. You have to compile and link the program `helloWorldPrimitives.cpp` and read it carefully, because it presents lots of clues to program the different stages of the CG pipeline. You can see an image rendered from the world of primitives in figure 1.**

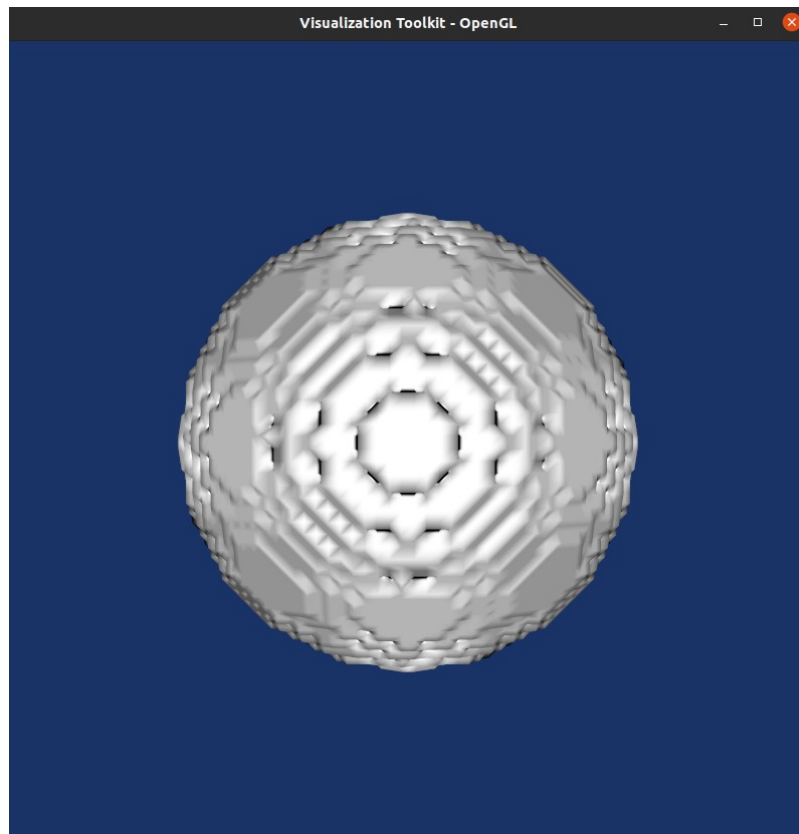
Tras leer el código y entenderlo, siguiendo el procedimiento habitual, ejecutamos el código e interactuamos con él mediante el ratón. Todo funcionó como esperábamos. El resultado fue:



### Task 3

**Compile and link the program `MarchingCubes.cxx` included in the `code.zip` file. The important thing here is to read the above paragraphs together with the source code for understanding the two pipelines and the connection between them by means of the `SetMapper()` method.**

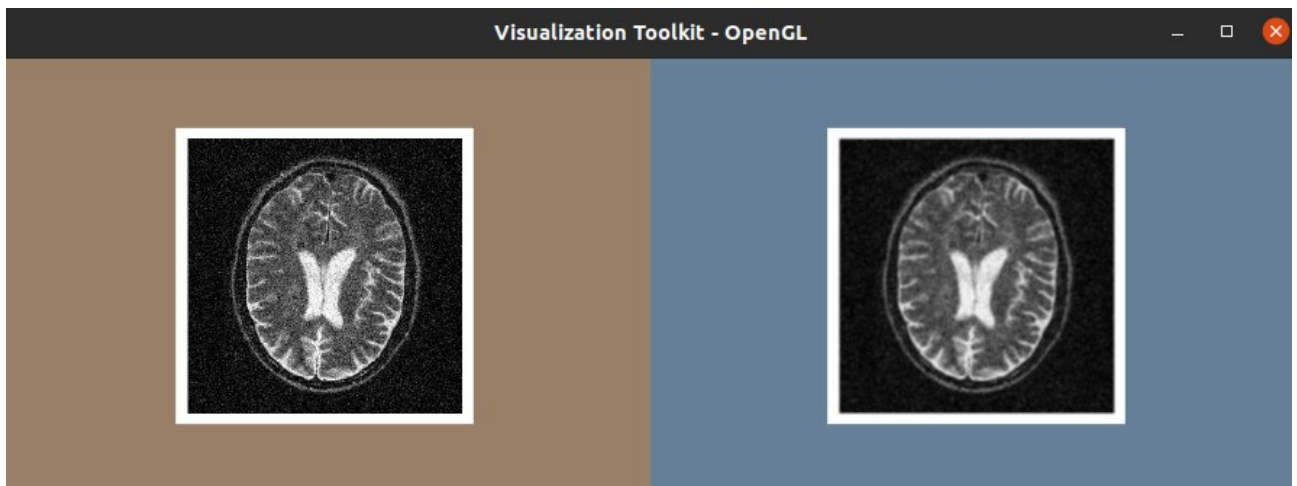
Hemos leído y entendido el código, siguiendo el procedimiento habitual, se ha el y hemos interactuamos con el mediante el ratón. Todo funcionó como esperábamos. El resultado fue:



### Exercise

**Try to visualize, using two different interactor styles of VTK, one of the ITK pipelines for noise reduction that you have programmed in previous sessions. Instead of using a `QuickView` object, we will be using the structure exemplified by `RescaleIntensityImageFilter.cxx` .**

Hemos incluido un filtro de ruido gaussiano al código proporcionado y el resultado es el siguiente:



Se adjunta el código en el que se incluye el filtro.