

Week 3: Global thresholding, dilation, and overlay of segmentation

Materials

MDSC 689.03 - W2017 Students > Image_Datasets

- Hip.zip → MRI image of the hip. DICOM format.
- Thorax.tar.gz → CT image of the thorax. DICOM format.
- head.nii.tar.gz → MRI image of the head. NIFTI format.

*tar.gz files can be uncompressed in linux using ***tar -xvzf *.tar.gz***

*The WinRAR program can also help to uncompress images in Windows and MacOS.

Assignment

- Implement global thresholding.
- Segment relevant tissues or regions of each image dataset (skin, bone, air, etc). Choose wisely.
- Display the segmentations overlaid on the original images.
- **Dilate** one or more segmentation results using a 3 x 3 x 3 kernel, implemented by you, and not using vtk already implemented filters.
- **Due on: Next monday at noon. Python files and screen captures must be uploaded to your Dropbox directory.**

Important considerations

- Both, **lower and upper** thresholds have to be considered in your implementation.
- Focus on tissues or regions that you can easily isolate by using global thresholding.
- Please implement the thresholding yourself, and do not use the corresponding VTK filters (vtkImageThreshold, or anything that provides a similar level of functionality). However, you may use those additionally for comparison of the results.
- The vtkLookupTable class can help you to map your segmentation to a colourful visualization.

```
lookupTable = vtk.vtkLookupTable()
```

```
lookupTable.SetNumberOfTableValues(2)
```

```
lookupTable.SetRange(0.0,1.0)
```

```
lookupTable.SetTableValue( outRangeValue, 0.0, 0.0, 0.0, 0.0 ) # black, transparent
```

```
lookupTable.SetTableValue( inRangeValue, 0.0, 1.0, 0.0, 1.0 ) # green, opaque
```

```
lookupTable.Build()
```

- Then, the vtkImageMapToColors class can match your binary segmentation with your lookupTable.

```
mapToColors = vtk.vtkImageMapToColors()
```

```
mapToColors.SetLookupTable(lookupTable)  
mapToColors.PassAlphaToOutputOn()  
mapToColors.SetInputData(resultBinaryImage)
```

- You will need to create two actors, one for the original image, and one for the binary segmentation to be overlaid. Both of them should be added to a same renderer.

```
renderer = vtk.vtkRenderer()  
renderer.AddActor(actorOriginal)  
renderer.AddActor(actorSegmentation)
```

Extra ideas

- Tkinter package offers a GUI to select the input file or directory. You can try using it.
- It could be a good idea to apply a Gaussian filter before thresholding, in order to remove some noise.

Some results

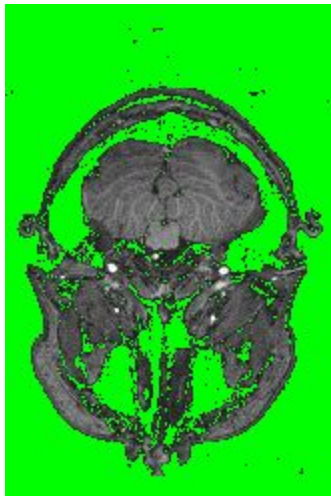


Figure 1. Slice number 100 of the head.nii.tar.gz dataset, with background segmentation overlaid. Window: 1000, level: 500, lower threshold: 0, upper threshold: 100.

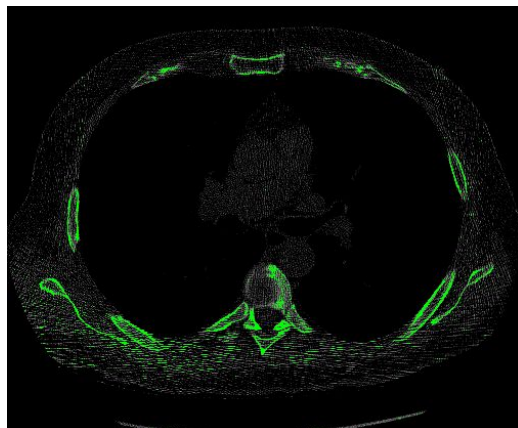


Figure 2. Slice number 100 of the Thorax.tar.gz dataset, with bone segmentation and some noise overlaid. Window: 1000, level: 500, lower threshold: 500, upper threshold: 5000.