

CAD-Assignment 1

Luc Nies (s4136748) Tom van de Poll (s4106512)
Harmen Prins (s4132297)

February 2016

Question 1

itkImageFileLoader loads images with the .itk file format while MLIImageloader loads images with the .ml file format.

Question 2

The most suitable overlay option is either **Add** or **Blend**. With Add, you can still see the original image through the overlay, which means you can inspect the highlighted area. When merging, the highlighted area is obscured, which is useful if the area should be masked because it is irrelevant.

See below for a more extensive evaluation.

Replace This overlay method replaces every pixel with the mask pixel. So if the mask has the same size as the image (which it does in our case), the output is equal to the mask and all information of the image is lost. This is a **bad** overlay method in this case.

Add This overlay method adds every pixel of the mask to the corresponding pixel in the original image. This means that the 'empty' part of the mask does not influence its corresponding pixel and those parts therefore equal to the original image. The rest is one color (e.g. green). This is good for highlighting certain parts of the image.

Blend This method is similar to add, except that it seems to normalize the image afterwards so that the parts of the original image become darker.

Reverse Subtract This overlay method should be the opposite of add. It obscures the parts where the mask empty and allows the rest (which is obscured in the add method) to be seen. This helps with removing useless parts of the image so the person/algorithm can focus on the relevant parts. However, in practice it seems to do exactly the same as replace. The same goes for subtract, minimum and maximum.

The documentation also claims that there are two more modules namely *Premultiplied Blend* and *Screen*, which are not available.

Question 3

The alpha factor determines the opacity of the segmentation mask created with the Intervalthreshold module. This means that you can still see the original image through the mask. This means masking an image with the Overlay add option and a lower alpha will be similar to an image overlaid with the merge method.

Question 4

See the attached networks and .cso file. Here's an example:

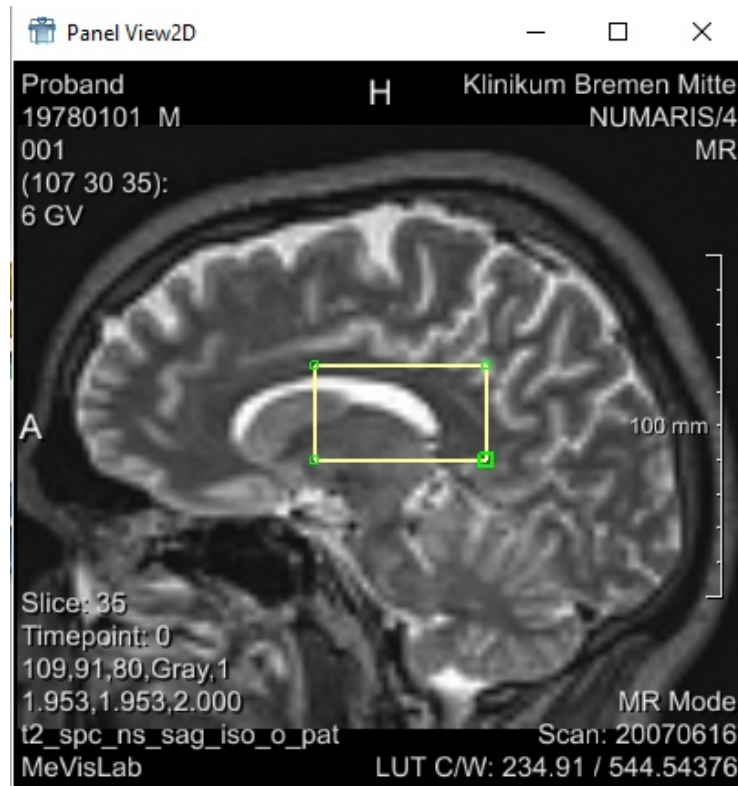


Figure 1: Slice 35 of the ProbandT2 scan overlaid with a contour.

Question 5

The CSOPrimitiveProcessor should be replaced with CSOFreehandProcessor in order to draw freehand contours.

Question 6

In order to save a contour as a binary mask image, the contours needs to be converted to a 3d mask using the `CSOConvertToImage`, then saved using `SaveImage`. In order to only save the boundaries of the contour, you need to uncheck the option 'Fill'.

Question 7

The newly created module has an input and output option for images, but the module itself has no functionality. This is expected as the module simply copies every voxel from the input to the output as can be seen in the follow line of code:

```
*outVoxel = *inVoxel0;
```

Question 8

The change in code necessary to complete the module is rather simple:

```
*outVoxel = *inVoxel0 + 5;
```

All that is needed

Question 9

The changed code in the voxel loop is as follows:

```
if (*inVoxel0 >= threshold)
{
    *outVoxel = *inVoxel0;
}
else
{
    *outVoxel = 0;
}
```

An if statement is used in order to compare the `InVoxel0` value to the threshold parameter, and the `InVoxel0` value is only copied if it is at least as high as the threshold value. Otherwise, the value is replaced with 0.