Sehbau: Segregation RGB

The use of the program for segmenting an image patch into two groups of regions is explained, named **sgrRGB**. The program takes as input a chromatic RGB triplet that represents the segmentation target, ie. a foreground object. The program then generates two groups, one containing the foreground region(s), and the other the background region(s). This is done with a single pass determining the distances for the target triplet and an automatically calculated distractor triplet. The purpose of the program is to obtain a more precise outline of an object or scene part as rapidly as possible.

Read	still needs to be written.
Repository	https://github.com/Sehbau/SgrRGB
Docu	overview: https://github.com/Sehbau/Docu/blob/main/overview.pdf

The directories in the repository contain the following:

- /**Imgs** sample images for immediate testing
- /**UtilMb** Matlab scripts to read the output data files

The program is available as:
- **sgrRGB.exe**, Windows 10 [x64]
- **sgrRGB_ubu**, Ubuntu 20.04.3 LTS (Focal Fossa) [x64]

The program **sgrRGB** takes a (color) image as input and if the approximate location of the target object (or scene part) is known, then it should be cropped before application for improved segmentation results. The program will determine a distractor triplet using the image border pixels. There are different initializations possible.

The example script `run1img.m` demonstrates how to run the program and how to load the output files. The script `compareInit.m` shows the output for the different initialization techniques.

1 Program Use

An image path and a RGB triplet needs to be specified. In the following example the average gray-scale value is selected (128, 128, 128):

```
> sgrRGB Imgs/img2.jpg 128 128 128
```

This will generate the following output:
- **BonFore.bonPix** boundary pixels for the target/foreground
- **CrvPrt.cvp** curvature partitions (arcs and straigther segments)
- **Mlab.mpu** black-white map, with white the foreground
- **Ifore.png** the foreground regions in average color
- **BonBack.bonPix** boundary pixels for the distractor/background

How those are loaded will be explained in Section 2. In the following it is explained what options are available.

1.1 Options

The following long option is available, to be specified with a double dash '--'; single letter optioni is not in use.

--init: the initialization technique that calculates the distractor triplet, specified as digit 0, 1 or 2:

-**0** (default): uses the (image) patch borders to calculate the average RGB value. The complexity is corrsponds to the number of border pixels: $O(\texttt{nPxBorder})$.

-**1**: determines the farthest distance between target and each border pixel. The complexity is the same as for `init=0`: $O(\texttt{nPxBorder})$.

-**2**: measures the distance of the target triplet to black (0,0,0) and white (255,255,255), and takes the farther of either as the distractor. This has lowest complexity as it calculates only two distances: $O(2)$.

2 Output

The **.bonPix** files can be loaded as explained in repository 'demotree': https://github.com/Sehbau/DemoTree

The **.cvp** file loads the arc and straighter segments as explained for the descriptor extraction program **dscx**: https://github.com/Sehbau/DescExtraction

The **.mpu** file holds the size of the map in the first two integer values, height and width. The remaining values are of type `unsigned char` and describe the map values. It is loaded with Matlab script `LoadMapUch.m`.

The plotting routines are taken partially from the repository for descriptor extraction: https://github.com/Sehbau/DescExtraction