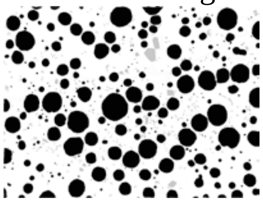
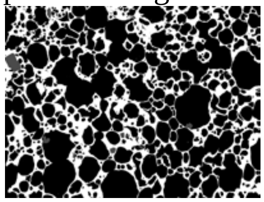
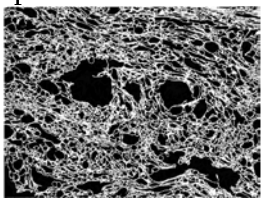
Homework Assignment #12
Image Processing – Part 1
Due date: Tuesday November 15, 2016, 11:59 pm

*This homework is based on the computer lab exercise and has three parts. Please cut and paste some results into this Word file and submit it to the bSpace drop box along with some *.m files that we ask you to write.*

Part 1 – More Bubbles in Pumice

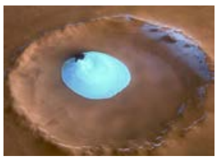
Download the following three pumice images from bSpace:

Compute the how many bubbles are in each image and determine the 2D porosity (ratio of voids and total area). **Enter your results** for each image here:

Now analyze the shape of the bubbles. Select three properties from the results of the *regionprop* function that best illustrate the differences between the three images. **Insert the results** for the three images **in numerical or graphical form** (e.g. table, histogram, or plot) and give a brief **explanation**:

Part 2 – Ice Deposits on Mars

ESA's Mars Express took this picture  that shows ice deposits inside a crater in northern plain on Mars. More details can be found at here. Download the image and start writing a new Matlab script file to analyze it. We want to determine the area of the ice deposits with two different methods:

a) First we want to use a grey scale threshold so that the ice deposit stands out as isolated grain. Read the image and convert it to a grey scale image, G. Use `imshow(G); impixelinfo;` and scan the resulting image with your cursor to determine a good grey scale threshold. Then use the command `bw=im2bw(I,threshold)` to obtain a black&white image. Decide if you need to add `bw=1-bw` before you use `bwconncomp` and `regionprops` to determine the size of the biggest grain in pixels.

b) Now we want to repeat the exercise in part a, but after having performed some smoothing.

Pick a reasonable size for your neighborhood and smooth using a median filter:

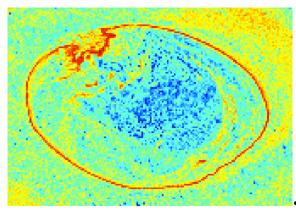
```
fSize = ???;
Sm = medfilt2(G, [fSize fSize]);
figure; imshow(Sm);
title('median smoothing');
```

As before, use the command `bw=im2bw(I,threshold)` to obtain a black&white image, and use `bwconncomp` and `regionprops` to determine the size of the biggest grain in pixels.

Write and submit a Matlab script that displays a black and white image where the ice stands out. As final step, your program should print the area of the ice deposit in pixels. You will need to fiddle with the threshold for a while to obtain a reasonable estimate to the size of the deposits. You may find that there is no single value that works perfectly and that is why this exercise has more parts.

c) Now we want to use an *entropy threshold* to the ice-rock boundary. Pick a reasonable size for your neighborhood and analyze the entropy landscape using

```
fSize = ???;
nhood = true(fSize,fSize);
E = entropyfilt(G,nhood);
figure; imagesc(E); impixelinfo
```

You should obtain an image where the ice boundary stands out . Again hover over the image with your cursor to determine an entropy threshold so that the ice boundary stands out as one *connected* ring.

```
thresh = ???;
L = E>thresh;
figure; imshow(L);
```

Now we analyze the different grains in the image

```
cc = bwconncomp(L, 8);
graindata = regionprops(cc, 'all');
```

Before we can obtain the area of ice deposit, we face two minor challenges. First the ice boundary is not the largest grain but the second largest. So the command, `[max_area, idx] = max(grain_areas)` does no longer work because it returns the index, `idx`, of the largest grain. Please use the `sort` function to obtain the grain index the represents the ice boundary.

Secondly, the grain contains the boundary of the ice, not the whole area of the ice deposit. Consequently, you need to determine the area that the ice boundary encloses. Luckily `regionprops` already provides the function you need.

Write and submit a Matlab script the prints the ice area based on the entropy and display it graphically. Compare with your results from section (a).

Part 3 – Size of Your Dream Lake

This is the part where you can be creative. Please use GoogleEarth, Google maps, or even a simple image search to download an aerial image of a lake of your choice. Alternatively look for a aerial photograph of a glacier and determine its size. Write a Matlab program that determines its size in units of pixels and possibly in km^2 also. **Please submit your image and Matlab code.** The best submission will be used the next time this course will be offered.