%% Exercise 19 - Working with images

%% 1. Digitization from screen
%% One of the common tasks in working with data is digitization of data from
%% images. Here is an example of doing this task in Matlab. Digitize the
%% temperature profile as a function of depth in Lake Kivu (East Africa).
%% First, download the image KivuT.jpg and load it into workspace. Replace
%% the image path below by the directory to which you downloaded the image.
I=imread('\Users\sergeikatsev/Documents/KivuT.jpg');
imshow(I)
%% The command 'imread' reads the .jpg format and parses it into a 3D array
%% where the first two dimensions are width and height, and the third one is
%% the color depth.
%% Now you need to relate the pixel coordinates to the data coordinates. You
%% can use the 'ginput' command to get the pixel coordinates of the corner
%% with the minimum x- and y- values (upper left in this example) and the corner
%% with the maximum values (lower right here).
a=size(I,2); b=size(I,1);
disp('Click on the corner of the graph with (xmin,ymin) then (xmax, ymax), then
<return>')
[xcr ycr]=ginput;
%% Define the limits for the axes in the image. These are the data
%% coordinates of the points that you just clicked on.
xmin=22; xmax=27;
ymin=0; ymax=480;
%% Now you can digitize the data. Click on the data points that you would
%% like to read the coordinates from. The data values will be in pixel
%% coordinates.
[xdata ydata]=ginput;
%% Show the dataset that you have created along with the image.
imshow(I)
hold on
plot(xdata,ydata,'o--')
hold off
%% Convert the points to the real coordinates and verify the graph.
figure
XDATA = xmin + (xmax-xmin)/(xcr(2)-xcr(1))*(xdata-xcr(1));
YDATA = ymin + (ymax-ymin)/(ycr(2)-ycr(1))*(ydata-ycr(1));
plot(XDATA,YDATA,'o-');axis ij
axis([xmin xmax ymin ymax])
%% If you find that you would like to add more points to your profile, how
%% would you do that?

%% 2. Digitization of color levels in images
%% In many areas, one needs the ability to process images such as maps or
%% microscopy pictures. Here is the image of the sediment core from the
%% Western arm of Lake Superior. Download the file CoreImage.jpg and load it
%% into the workspace.
CoreImage=imread('\Users\sergeikatsev/Documents/CoreImage.jpg');
imshow(CoreImage)
%% In cases where the information is contained mainly in the brightness of
%% the color, it may be beneficial to work with grayscale images. Let's make
%% a grayscale copy:
CoreImageGray=rgb2gray(CoreImage);
imshow(CoreImageGray)
%% If you worked with digital photos, you may be familiar with the concept
%% of an image histogram. This is the distribution of grayscale levels in
%% the image.
imhist(CoreImageGray)
Many common image enhancement techniques are based on modifying the histogram. Here is the result of histogram equilization. Note that the contrast is enhanced.

```matlab
CoreImageEq = histeq(CoreImageGray);
```

```
subplot(3,1,1)
imshow(CoreImage)
subplot(3,1,2)
imshow(CoreImageGray)
subplot(3,1,3)
imshow(CoreImageEq)
```

```
% imhist(CoreImageEq)
```

You may also convert the original image from true color to indexed color (fewer colors), although this is not particularly useful in this exercise.

```matlab
[x map] = rgb2ind(CoreImage, 16);
imshow(x, map)
```

```
%% Color intensity transect
% Let's digitize the color levels.
imshow(CoreImageEq)
axis on  % shows the image dimensions in pixels
%%
% Use the ginput command to convert the scaling from pixels to cm.
% Click on the scale in the image two times, at a 10 cm interval, and hit Enter.
[x, y] = ginput
% Now the scaling factor should be 10/(x(2)-x(1))
%%
% Show the image with the new scaling factor. Compare your scale against the ruler in the image. What can be done to improve the accuracy?
ix = 10/(x(2)-x(1)) * size(CoreImageEq, 2)
iy = 10/(x(2)-x(1)) * size(CoreImageEq, 1)
imshow(CoreImageEq, 'Xdata', [0 ix], 'Ydata', [0 iy]), axis on
%%
% Now determine the color intensity along a transect.
% The function 'improfile' determines the RGB pixel values C along line segments defined by coordinates [CX, CY].
[CX, CY, C] = improfile;
% Click on the image to define the transect and hit Enter;
%%
% Show the transect and the color values
% First, on the grayscale image
subplot(2, 1, 1)
imshow(CoreImageEq, 'Xdata', [0 ix], 'Ydata', [0 iy]), hold on
plot(CX, CY),
axis on
plot(CX, C/30, 'k', 'LineWidth', 2);
xlim([0 71])
hold off
% The grayscale levels have been scaled to fit within the image.
```

```
%% Second, on the colored image (the R G B values)
%C = improfile(CoreImage, [CX(1) CX(size(CX, 1))], [CY(1) CY(size(CY, 1))]);
imshow(CoreImage, 'Xdata', [0 ix], 'Ydata', [0 iy]), axis on
[CX, CY, C] = improfile;
%%
subplot(2, 1, 2)
imshow(CoreImage, 'Xdata', [0 ix], 'Ydata', [0 iy]), hold on
plot(CX, CY),
axis on
plot(CX, C(:, 1)/10, 'r', CX, C(:, 2)/10, 'g', CX, C(:, 3)/10, 'b')
hold off
```