Multimedia lab 1 – Digital image processing

Intensity transforms, spatial filtering and geometric transformation

In the first session we will explore changing the pixel intensities in an image and filtering in the spatial domain. Both are common methods used in the manipulation of digital images (e.g.: what is behind the buttons in Photoshop). For example, pixel wise intensity transformations can be used to patch up an over or under exposed imagine, while a common application of spatial filtering is the removal of noise. Finally there is also an exercise on geometric transformation, which has an application in the creation hybrid satellite/road map images (e.g.: google maps).

For each exercise you can find example m-files on Toledo (in the same folder as the slides). These files contain examples on the techniques used in each exercise. These are only examples, you write your code in a new separate file.

Preparation

For these exercises we will use the dipum toolbox. You can download the necessary scripts from:

https://beagle.whoi.edu/redmine/projects/ifcb-web/repository/show/trunk/dipum_toolbox_2.0.1

You can put these scripts directly in your working folder, any other scripts and function inside the working folder will then be able to use the dipum functions.

Also place the contents of 'images.zip' into your working folder.

Ex. 1: Intensity transforms

Example m-files: gamma_log_1.m, hist_equal_1.m, hist_match_1.m

1. Load image "Underexposed.jpg" into Matlab and convert it to grayscale and check the type of the image (it should be `double, UINT8`). Convert the image to `double` (see table 2.3 on p26 in the textbook). Show this image. **Don't** use square brackets with `imshow` to rescale the intensities.
   What do you observe? What stands out?
2. **From now on, continue only with the grayscale image**
3. Show the image's histogram together with the cumulative distribution function. How can you link the cumulative to the shown image? Do the histogram and cumulative distribution function look like what you'd expect?
4. Determine a grayscale value in the histogram above which there are almost no more pixels present, except for the white pixels coming from the light source in the ceiling (you can do this manually). Now use the function `imadjust` to spread out the range of intensities (from 0 to your determined value) over the full range. Show the result of the image and the histogram.
5. Try to improve the grayscale image using *gamma correction*. Try it with different values. What are your results and findings?

6. Apply a *contrast-stretch transformation* as described in the textbook on p69. Try this with 2 different values for m. For the first value you can use m = 0.275 (70 in `uint8`), this equals the grayscale value of the sweater of the person in the middle. You can take 2 or 3 as value for E. What do you notice in the image after the contrast stretch (also with regards to the person in the middle)? What happens for high values of E and how do you see this in the image, the histogram and the cumulative distribution function? (tip: you will see this more clearly if you take a lower value for m because the image is under exposed).

Ex. 2: Spatial filtering

Example m-files: lin_filt_sharpning.m, lin_filt_smooth.m, linear_filt.m, nonlinear_filt_1.m.

1. Load the image "Landscape.png" into Matlab and convert it into a grayscale image. Show the grayscale image and histogram. From now on work with the grayscale image.
2. Use the *Laplacian* to sharpen the image. First, create the Laplacian filter and then use `imfilter` to filter the image. (tip: you can use the option 'replicate'). What do you see in the filtered image? Now subtract the filtered image from the original grayscale image to sharpen the original image (or the other way around, which way should it be or doesn't this matter?). Does this produce a good result?
3. Load the image "LandscapeWithNoise.png" into Matlab. This image shows an example of 'salt & pepper'- noise. Show the histogram of the image and compare this to the grayscale histogram of point 1. What do you notice? Try to remove the noise using a *spatial averaging filter*. Also try it with a *median filter*. What are your findings for each filter, and which filter performs the best? Why?

Ex. 3: High frequency emphasis filtering

High frequency emphasis filtering is a technique used to emphasize high frequency components in an image, while still remaining some of the DC energy. This method is described on p138 in the textbook. Use this method to filter the image "X-ray_neck.jpg".

1. Load the image "X-ray_neck.jpg" into Matlab and convert it into a grayscale image. From now on work with the grayscale image.
2. Apply a *Butterworth high-pass filter* to the image. You can create the filter using `hpfilter`. You can choose the cut-off frequency equal to 40 pixels. Discuss your findings. Also show the image in the frequency domain. What is the relation between the distance to the origin in the frequency domain and the suppressed frequencies in the image?
3. Now apply the *high frequency emphasis filtering* on the original grayscale image. a and b are values that describe how much the low and high frequencies respectively are allowed to pass through. Try a few values for a and b and discuss your findings. For each combination of values, compare the output image to the original.
4. Use a = 0.5 and b = 2 and perform *high frequency emphasis filtering* on the original grayscale image. Then apply a *histogram equalization* on the emphasized image. Compare this image to the result of a histogram equalization on the original greyscale image. What are your findings?

Ex. 4: Geometric transformation

You can find information on geometric transformations in the textbook on p182-193.

The goal of this exercise is to match a road map onto a satellite image. The orientation of the satellite image is unknown (and different from the road map). To get good matching, several key points can be used to indicate the same place in both images. Choosing good key points will impact the quality of the transformation.

Example m-files: example_5_12_1.m, example5_14_1_1.m

1. Load the images “GeelMap.png” and “GeelSatellite.png” into Matlab.
2. Use the matlab function `cpselect` to find similar points in both images. Using the function `cp2tform` you can find the transformation between both images.
3. Transform the satellite image using `imtransform` so that it matches the orientation of the road map. Make sure that the output image has the same size as the road map (use options ‘XData’ and ‘YData’ in the function `imtransform`).
4. Show the transformed satellite image and the road map. Also show an image where the satellite image is overlaid onto the road map. Tip: use `hold on` and the option ‘AlphaData’ on the image handle to make it transparent. Make sure the resulting image is a color image.
5. Is the result correct? What happens when you choose key points that are too close together, and why is this?