Image Processing lab session 1: image sketching and image mosaïcking

N.B. for this labsession, you have to deliver a report with your main results and the code you have developed. Please name the different files with the last names of the pairs and download both files on chamilo, Travaux directory one week after the session at the latest.

1. Image sketching

The goal of this exercise is to build a sketch from a given image. Look below at an example of this kind of process:

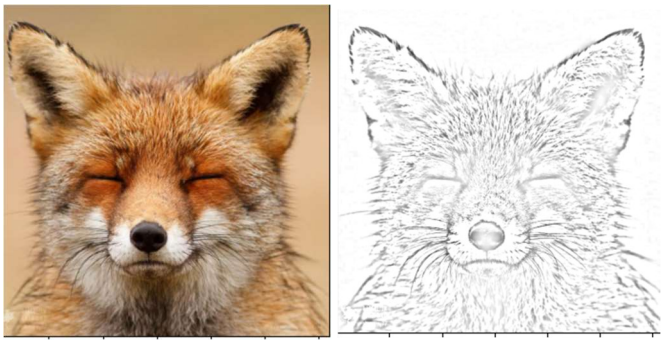

Image sketching involves the following steps:

- Luminance computation (if the original image is a color image)
- Image video inversion (I=255-I)
- Inverted image blurring with a Gaussian filter
- Final sketch computation in the following way:

  sketch = (inverted_blurr_img*255)/inverted_img

  where all the values higher or equal to 255 are casted to 255

Develop a code in order to sketch an image and apply it on a picture of your own face. Display the obtained image after each step of the algorithm.

Test different values of the Gaussian filter bandpass (sigma parameter). Explain the evolution of the obtained sketch wrt the sigma value.

Deliver the python code and put your sketched face for three different sigma values in your report.

2. Image mosaicking

The main goal in this part is to create a photo mosaic from a dataset of small images called patches. The main principle is to split the image to be mosaicked into small square blocks and to replace each square block by the most similar patch of the dataset. The following figure presents an example of the final result to obtain. The global image looks like an eagle in front of the American flag and when zooming around the nose, the patch images are appearing.

Mosaic image: final goal

Zoom around the eagle nose

There are three main steps in the mosaicking process:

- Split the target image in small blocks and compute the mean color of each channel over each block. Generate a new image in which the color of all the pixels in a given block are replaced by the mean color value.
- For each patch in the dataset, compute the mean color on each R, G and B channel.
- Select in the patch dataset, the patch that is the most similar to the target block to be replaced. The Euclidean distance will be used in order to compute the color similarity between each patch and a given block. Generate a second image in which each block has been replaced by the most similar patch that has been resized to fit the block size.

Rmk: the size of the blocks to be replaced is an input parameter of the program.

The directory ***Patches*** contains more than 2000 images to be used in order to build the mosaic image. The directory ***Target*** contains several images to be used as the image to be mosaicked. It is not mandatory to deal with all the proposed images of the Target directory.

Develop a well-organized code in order to build an image mosaic from the proposed patch images. Each step has to be clearly identified in your code. The code has to be delivered on chamilo.

Rmk: when developing your code, do not consider the whole patch dataset other while the running time is too long. Put as a parameter the number of images to take into account in the patch dataset. 100 patches could be enough during the code development step.

2.1 Block size influence

Run your program on several images of the ***Target*** directory with different values of the block size. Comment the results. According to your experiments, what seems to be an appropriate block size value? Take this value as default block size value for the second part of the exercise.

2.2 Patch number influence

Reduce to 5 images only the number of patch images to be taken into account in the ***Patches*** directory and run the program on the same images as in part 2.1. Comment the results.