Lab 3 – Quantization and Color Histogram Equalization

Goals:

- *Getting familiar with Histogram techniques.*
- *Introduction to video tracking using Camshift algorithm.*

Introduction

In this lab we will we will analyze the effects of quantization on image quality. We will get familiar with Histogram techniques which allow us to analyze the distribution of gray levels in image. In this lab we will study histogram, histogram normalization and histogram equalization.

Video tracking - Camshift algorithm

One of the common tasks in digital image processing is tracking an object in a video, that is, finding the location of the moving object in all video frames. The OpenCV suggests a Camshift algorithm [4] for tracking a user-selected object. This algorithm consists of 4 stages:

1) **Initialization**: Create a histogram of the hue (color) component of the selected object. Normalize a histogram to form a probability density function (the "statistical signature" of the object).
2) **Histogram back-projection**: For each pixel in the frame, compare its hue value with the "signature" of the object and find the probability that this pixel belongs to the object. (For instance, if the pixel's hue is equal to 23 and in the "signature" the value 23 corresponds to probability 81%, then the probability of this pixel to belong to the object is equal to 81%.) At the end of this stage we get a "map of probabilities" of pixels to belong to the object.
3) **Mean-shift algorithm** [5]: In order to find the location of the object, we have to find the peak in the "map of probabilities". The mean-shift algorithm performs an iterative search in the rectangular window (the "search window"), locating the center of gravity within the "search window" and then centering the "search window" at this center of gravity. This process stops when we perform a predefined number of iterations or

when the displacement of the "search window" becomes smaller than a predefined threshold. Now the "search window" is centered at the center of gravity of the peak of the "map of probabilities," that is, at the supposed location of the center of object within the frame.

4) **Adaptive adjustment**: Using the statistical moments computed within the "search window", compute:
   - the parameters of the rectangle (size, rotation angle) that is centered at the center of gravity and delineates the peak, that is, the supposed object.
   - the size of the "search window" that will be used for the next frame.

Now display the frame and plot the rectangle on the frame.

For the next frames we repeat the stages (2-4) of this algorithm.

For more information, refer to Ref [4].

Preliminary report

Part 1 – Quantization

- Give the definition of image quantization. What is it used for?
- Write a python function: ***def quant_img(img, N):*** that performs uniform quantization of the *image* to N gray levels (0≤N≤255) and return the quantize image, **use only *Numpy* functions for this implementation**. Check your code on an image and make sure that it works.
- Explain the problem of "false contours" that occurs in image quantization. Suggest a method to reduce this artifact.
- Bonus- Quantization using K-Means algorithm – read about sklearn.cluster.KMeans function. Give a short explanation about K-means algorithm.

Part 2 – Histogram and Histogram Equalization

- Give the definition of contrast stretching. What is contrast stretching used for?
- Give the definition of histogram equalization. What is histogram equalization used for?

- Explain why after histogram equalization the histogram values don't become equal (in other words, they are not lying on a straight line).
- How will contrast stretching and histogram equalization affect the histogram of the image? What are the similarities and differences between them?
- How will contrast stretching and histogram equalization affect the image? What are the similarities and differences between them?
- Suggest a way to perform histogram equalization on a color image. Remember: histogram equalization should improve contrast and not change the distribution of colors. (Hint: in which color space the colors will not change?) Explain your choice.

Find a grayscale image and a color image to be used in the lab and send it to your email. (Hint: in order to demonstrate the contrast-correcting behavior of the histogram equalization, it is recommended to choose a low-contrast color image.)
Send also the python function **quant_img**(*img*, N) that you wrote.

<u>Part 3: Video Tracking</u>

1. As described in the introduction, the tracking algorithm works on the hue component of the HSV color space. Explain why working on this component should give better results than working in the RGB color space.
2. Review the code of the demo **camshiftdemo.py** that demonstrates the use of the Camshift algorithm, and understand how to use the code (i.e. what happened when you press on the 'I' key or 'q' key)

Description of the experiment

<u>Part 1 – Quantization</u>

Open **Lab3_student.ipynb** in Jupyter Notebook, and follow the instructions.

<u>Part 2 – Histogram and Histogram Equalization</u>

- Submit the results of demonstrations from Jupyter Notebook Part 2 with the image of your choice.
- Suppose the histogram of an image has two sharp peaks. The histogram with two sharp peaks is called *bimodal*. Explain how bimodal histogram can be used for binarization.
- Explain the main difference between contrast stretching and histogram equalization. Which problem may occur in histogram equalization and why doesn't it happen in contrast stretching? (*hint*: the problem occurs mostly in smooth background areas).

Answer all the questions appeared in the final report. You can answer the questions using Word, Latex or in your Jupyter notebook - in the end of each section as markdown cell, save the files as PDF format.

<u>Part 3 - Video Tracking</u>

Use the demo **camshiftdemo.py** to track your face as seen in the webcam:

1. Check different sizes of selected area and observe the effect.
2. Observe the quality of tracking when tilting your head and when getting closer and farther away from the camera.
3. Observe what happens when your face get closer to your partner's face.
4. Select an object that isn't a face (for example a notebook) and observe the quality of tracking of this object compared to face tracking.

Describe your observations in the final report; refer to each of the aforementioned subjects.

Final report

Part 1 and 2: Quantization , Histogram and Histogram Equalization

Submit the results of the demonstrations from Jupyter Notebook with the image of your choice. Answer all questions and give the relevant examples as instructed in those parts. Submit the file as PDF format.

Part 3: Tracking

Write your observations from the demo **camshiftdemo.py**, according to the instructions in part 3 of the experiment.

References

1. Install: https://pypi.org/project/opencv-python/
2. Documentation: https://docs.opencv.org/master/index.html
3. Tutorial: https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
4. G. R. Bradski, "Computer video face tracking for use in a perceptual user interface," *Intel Technology Journal*, Q2, 705–740, 1998, available at http://opencv.jp/opencv-1.0.0_org/docs/papers/camshift.pdf
5. D. Comaniciu, P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis Machine Intell.*, Vol. 24, No. 5, 603-619, 2002, available at
    i. https://courses.csail.mit.edu/6.869/handouts/PAMIMeanshift.pdf