Lab 4 – CAMSHIFT and Mean Shift

Goal: to provide an introduction to CAMSHIFT face tracking algorithm (based on Mean Shift algorithm) implemented in Computer Vision System Toolbox. In this lab we'll learn to track moving face in video. Also, we'll perform several additional tasks.

1. Median Filtering of Salt-and-Pepper Noise

Use the command **doc medfilt2** in Matlab command window. In the resulting Help Window find **Examples**. You'll see 4 lines of code and explanations. Copy the code and explanations to a new file (don't forget to "comment-out" the explanations with **%**) and save this file. Run this file and observe the results.

1. Describe the salt-and-pepper noise. What may be the physical reason of appearance of this noise in image processing system?
2. Is local mean filter suitable for removal of salt-and-pepper noise? Explain your answer.
3. What is the approximate number of the "bad" pixels (that is, the pixels affected by the noise) in the noisy image **J**?
4. Why local median filter **medfilt2** is very good for removal of salt-and-pepper noise while preserving the edges of the original image? Explain your answer.
5. What window size is used in the command **K = medfilt2(J);**?
6. In the output image **K** we can see black dots in the four corners. Why did this happen?
7. Suggest a simple method for removal of these black dots. (Hint: change input arguments in the command **medfilt2.**)

2. Histograms of R,G,B Layers

Use the command **doc** in Matlab command window. Then select:
Computer Vision System Toolbox→Examples.
Then find the section **Display and Graphics** and choose **Histogram Display**.
Then press the button **Open this Example** (in the top-right corner). Then copy the contents of the file **videohistogram.m** to the file in your local directory.

1. What is the size of the output of the command **cat(3, R_hist,G_hist,B_hist)**?

In Part 1 we computed local medians in running window (that is, computed a median for window placed around each pixel, write this median at the "results" 2D image at the coordinate of the pixel). Now suppose that we need to compute a histogram in running window of size $M \times N$ (M, N are odd integers, e.g., $M = 5, N = 5$). (That is, we compute a histogram for window placed around each pixel, write this histogram in the "results" 3D image at the coordinate of the pixel – the first two dimensions are the spatial dimensions of the image and the third dimension is used for the axis of the gray levels of the histogram of the window around this pixel. That is, in the "result" image for each pixel we attach (in third dimension) the histogram of the window around this pixel. In principle, In principle, histogram can be computed "by definition": for every pixel take the $M \times N$ matrix of

values, sort these values and count the number of pixels with the same value. This "brute-force" approach is correct, but computationally inefficient. If we want to compute in real time the histogram in running window for all pixels, we have to use fast recursive method. Indeed, if we move from the "current" window position to the "next" window position (one pixel to the right), then for each pixel the "$M \times N$-window" is changed by only two columns: the left-most column of "oldest" pixels is removed and the right-most column of "newest" pixels is inserted. So in order to compute the "next" histogram we need to "smartly update" the "current" histogram (and not to compute the histogram of the "next" window "from scratch").

2. Suggest a (simple) fast recursive algorithm for computation of histogram in running window of size $M \times N$.

3. CAMSHIFT and Mean Shift

Use the command **doc vision.HistogramBasedTracker** in Matlab command window. In the resulting Help Window find **Examples** and press on **Track a Face**. You'll see the code and explanations. Copy the code and explanations to a new file (don't forget to "comment-out" the explanations with **%**) and save this file. Run this file and observe the results.

1. Describe the Mean Shift algorithm [1] for estimation of PDF and clustering.
2. Describe the CAMSHIFT algorithm [2] for face tracking.
3. What is the data type of the variable **videoFileReader**?
4. Why the computational speed of the object **vision.HistogramBasedTracker** can be improved by setting the class of the input image to **uint8**?
5. Why the Hue layer (and not R,G,B layers) was chosen for face tracking?
6. How do we have to modify the program in order to change red rectangle to red ellipse? The ellipse will delineate the tracked object, showing its position, size and orientation (angle). (You don't need to attach the modified code because it's not easy to draw ellipse in this case, just provide the answer how the orientation of the tracked object can be found.)

References

[1] 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 http://comaniciu.net/Papers/MsRobustApproach.pdf

[2] G. R. Bradski, "Computer video face tracking for use in a perceptual user interface," *Intel Technology Journal*, Q2, 705–740, 1998, available at http://www.dis.uniroma1.it/~nardi/Didattica/SAI/matdid/tracking/camshift.pdf