CS-512 – Assignment 2 (4%)

Filtering and OpenCV

Due by: October 2, 2018

Review questions

Answer the following questions. Make sure that your answers are concise. In questions requiring explanation, make sure your explanation is brief.

1. Noise and filtering

   (a) Explain how to estimate the signal to noise ratio (SNR) in an image.

   (b) Explain the difference between Gaussian and impulsive noise. Which filter handles better impulsive noise: an averaging filter or a median filter.

   (c) Given an image having the value of 2 in each cell, write the value of the pixels in this image after applying a 3×3 convolution filter having all 1-s in its entries.

   (d) Given that we need the derivative of an image convolved with a filter explain how the operation can be applied more efficiently.

   (e) Explain the three different ways to handle boundaries during convolution.

   (f) Write a basic 3×3 smoothing filter. What is the sum of all entries in this filter? Explain the reason for the sum to be selected as it is.

   (g) Explain how to implement a 2D convolution with a Gaussian using two 1D convolution filters. Which option is more efficient? Is it possible to implement any 2D filter in this way?

   (h) Given a 1D Gaussian filter with $\sigma = 2$, what should be the size of this filter?

   (i) Explain how a Gaussian image pyramid is produced. What is the reason for producing such pyramids? What is the amount of additional processing done in a pyramid compared with a single image?

   (j) Explain how the Laplacian pyramid is produced and its use.

2. Edge detection

   (a) Why is edge detection useful? What are the desired properties of edge detection?

   (b) Explain the basic steps of edge detection and the need for them: smoothing, enhancement, localization.

   (c) Describe two filters for computing the image gradient. What is the meaning of the image gradient? What is it used for?

   (d) Explain how the Sobel filter can be produced from a smoothing and derivative filters.

   (e) Explain how to generate a more accurate derivative filter with an arbitrary σ. Write the elements of a filter for more accurate derivative computation with $\sigma = 2$.

(f) Explain how an edge can be localized using the first or second order derivative of the image.

(g) Let $\sigma = 1$. Write the Laplacian of Gaussian (LOG) filter using this σ. Explain how to use LOG to detect edges.

(h) Explain the main difference between the Canny edge detection algorithm and a standard edge detection that does not use directional derivatives. What is the condition for detecting an edge candidate in Canny?

(i) Explain the non-maximum suppression and hysteresis thresholding parts of the Canny algorithm.

Programming questions

In this part you need to write a program to perform simple image manipulation using openCV. The program should load an image by either reading it from a file or capturing it directly from a camera. When the user presses a key perform the operation corresponding to the key on the original image (not the result of the last processing step). The program should satisfy the following specifications:

1. The image to be processed by the program should be either read from a file or captured directly from a camera. If a file name is specified in the command line, the image should be read from it. Otherwise the program should attempt to capture an image from a camera. When capturing an image from the camera, continue to capture and process images continuously.

2. The read image should be read as a 3 channel color image.

3. The program should work for any size image. Make sure to test it on different size images.

4. Special keys on the keyboard should be used to modify the displayed image as follows:

   (a) 'i' - reload the original image (i.e. cancel any previous processing)

   (b) 'w' - save the current (possibly processed) image into the file 'out.jpg'

   (c) 'g' - convert the image to grayscale using the openCV conversion function.

   (d) 'G' - convert the image to grayscale using your implementation of conversion function.

   (e) 'c' - cycle through the color channels of the image showing a different channel every time the key is pressed.

   (f) 's' - convert the image to grayscale and smooth it using the openCV function. Use a track bar to control the amount of smoothing.

   (g) 'S' - convert the image to grayscale and smooth it using your function which should perform convolution with a suitable filter. You need to implement your own convolution function here. Use a track bar to control the amount of smoothing.

   (h) 'd' - downsample the image by a factor of 2 without smoothing.

   (i) 'D' - downsample the image by a factor of 2 with smoothing.

   (j) 'x' - convert the image to grayscale and perform convolution with an x derivative filter. Normalize the obtained values to the range [0,255].

   (k) 'y' - convert the image to grayscale and perform convolution with a y derivative filter. Normalize the obtained values to the range [0,255].

   (l) 'm' - show the magnitude of the gradient normalized to the range [0,255]. The gradient is computed based on the x and y derivatives of the image.

(m) 'p' - convert the image to grayscale and plot the gradient vectors of the image every N pixels and let the plotted gradient vectors have a length of K. Use a track bar to control N. Plot the vectors as short line segments of length K.

(n) 'r' - convert the image to grayscale and rotate it using an angle of θ degrees. Use a track bar to control the rotation angle. The rotation of the image should be performed using an inverse map so there are no holes in it. Use the cv2.getRotationMatrix2D and cv2.warpAffine functions.

(o) 'h' - Display a short description of the program, its command line arguments, and the keys it supports.

5. In the report you prepare you must summarize the algorithms you used, and evaluate the performance obtained. See provided sample report.

6. Follow the submission instructions of assignment 1.