ECE 5630: Digital Signal and Image Processing - Two-Dimensional Convolution

Due: October 4, 2018 at midnight

Name (Print):

Objectives

This programming assignment has three objectives:

1. Give you practice in using a program to perform digital signal processing.
2. Help to solidify your understanding of two-dimensional convolution.
3. Observe the effects of three different kinds of image filters.

Instructions

1. The program should be written in C or C++. Only libraries from the C/C++ standard libraries are permitted. No other external libraries should be used – you are writing the entire convolution from scratch. You must write all of the code necessary for the processing.
2. Use the `image.pgm` file found on the Canvas website to test your program. You can learn about the .pgm image format from a Google search. Remember that the pixels are 8-bit `unsigned char` values. You will need to truncate values greater than 255 to 255, and values less than zero to zero in the filtered images.
3. Write your output images in .pgm format. (The output images should be in the *same* binary, `unsigned char` per pixel format as the input image.)
4. Use zero padding to extend the input image for convolution.
5. Crop the output image back to the input image size.
6. Upload a short report and output images to the Canvas website. The report should include an introduction, a section describing your approach and findings, and a conclusion. Please include your C/C++ code as an appendix.

---

1. Write a program that will read in the provided image, filter with an arbitrary-sized 2D filter, and write out the filtered image.
2. Filter the image with a lowpass filter given by the following:

$$\mathbf{H}_1 = \frac{1}{81}\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix} \tag{1}$$

3. A common filter used for finding edges in an image is called a Sobel filter. The Sobel vertical and horizontal filter coefficients are defined as two 3x3 matrices given by

$$\mathbf{S}_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{2}$$

for detecting vertical edges, and

$$\mathbf{S}_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3}$$

for detecting horizontal edges. The two filters are each convolved with the input image $\mathbf{F}$ giving

$$\mathbf{G}_1 = \mathbf{S}_1 * \mathbf{F}, \tag{4}$$

and

$$\mathbf{G}_2 = \mathbf{S}_2 * \mathbf{F}. \tag{5}$$

The absolute value of the resulting images are summed together to create the output image

$$\mathbf{M}(m,n) = |\mathbf{G}_1(m,n)| + |\mathbf{G}_2(m,n)|. \tag{6}$$

Implement the Sobel filter using your program with appropriate modifications.

4. A very useful application of filtering is to use correlation to find features in an image. We will detect a feature using the following steps:

   (a) Read in `image.pgm` and square each of the pixel values (pixel-by-pixel). Write out the resulting image in binary `float`s.

   (b) Create a filter of the same size as the image in the file `filter.pgm` found on the Canvas site. All the values of the filter should be 1.0. Convolve the image from the previous step with your filter and save the result in binary `float`s.

   (c) Convolve the filter in `filter.pgm` with `image.pgm`, and then divide each pixel in the convolved output by the corresponding pixel in the image created in Step 4b above.

   (d) Scale the filtered image from Step 4c so that the maximum value in the image is 255. (Scaling should be done *before* you write out the pixel values.) Save the resulting image in binary `unsigned char`s, one `unsigned char` per pixel.