605.427: Project 4 – Coin Detection

1. (Question) Accurate edge detection algorithms are fundamental to most image segmentation and object recognition systems. Given an image Ii, list three algorithms or image operations that can be used to extract an edge map. For each algorithm or operation, provide the formulation and in- depth explanation about how it works. (Assume grayscale images.)

(Answer) There are four popular edge detection techniques: first order derivaties, second order derivatives, multi-filter processing, and canny-edge detection. I’ll focus on examples of the first three below.

* 1. (Answer) A Sobel operator is a first order derivative filter. This means it is less susceptible to noise than a second order derivative, additionally it uses a (larger-than-Roberts) 3x3 filter with the origin at the center to reduce the affect of noise somewhat. A greater weight is given to the derivative of pixels immediately above/below/left/right of the center. This can be seen in the following two kernels used:  which are an approximation of the derivative.
  2. (Answer) A laplacian filter is a second order derivative filter which can be used to detect edges. The formula for this filter is  and it can be approximated by a convolution kernel, typically one of the following: .
  3. (Answer) You can combine filters to produce an even better edge detection algorithm. For example, by taking the laplacian of the Gaussian, we essentially smooth the image first, reducing noise and thus improving the results of a laplacian edge detection algorithm (see answer “b” above). Because laplacian (second order derivative) is especially sensitive to noise, this combination brings out the best that the laplacian can offer by first *reducing* noise in the image before applying the filter. Also, the combined kernel can be calculated in advance, allowing only one convolution to be performed! The equation and kernel are as follows:  . We can see that the origin is in the middle of the kernel and it exhibits the typical drastic Gaussian behavior by emphasizing the central points.

2) (Questions) List and clearly explain two algorithms that can be used to smooth an image. As part of each explanation, provide the formulation or kernel needed to implement each individual approach. (Assume grayscale images.)

(Answer) There are several ways to smooth an image, and several more which remove noise (even if they don’t *smooth* the entire image). The most frequently-used techniques for smoothing an image are as follows: image averaging, spatial filtering, and frequency domain filtering. Furthermore, it’s possible to remove noise using a morphological transformation called erosion.

1. (Answer) The simplest-to-understand method of smoothing is the averaging spatial filter. This filter is a NxN kernel convolution filter where the center pixel becomes the average of all pixels in the kernel. The kernel therefore has homogenous values of 1/N2. The key to this filter is getting the N value, or kernel size correct: too big and the whole image will become a blurry mess but too small and the noise will not be removed. This is also very similar to averaging multiple images of the same scene.
2. (Answer)

1. Simple smoothing spatial filter where a convolution of size NxN averages all the pixels in a kernel...very useful in removing noise...kernel like 1/9 each in a 3x3...note the size of the kernel determines how MUCH smoothing/averaging is performed (too big and it’s all one homogenous blur, too small and you don’t get much smoothing)

2. Weighted smoothing filter....usually priority given to the center, left, right, up, down pixels, then finally outer corners

3. Median filter: take the median of the kernel? Good for salt and pepper....essentially removes it entirely! (Useful for coin counting!?)

4. Gaussian filter: specific form of weighted smoothing filter in the shape of a Gaussian distribution

5. Finally, to smooth an image could be seen as simply using a low pass filter and this can be accomplished by using the Fourier transform and filtering i.e. multiplication in the frequency domain. Applying a low pass filter to a frequency domain image is as simple as AND’ing a mask of all 0’s (black) except for a circle of 1’s in the center of the image...this could also be modeled as a step function but is effectively a circle where the radius corresponds to the cutoff frequency of the filter

a. There is also the butter worth low pass filter which is similar to the Gaussian low pass filter

