UNIVERSITY OF
FLORIDA

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

EEL 6562
Image Processing and Computer Vision
Box Filter and Laplacian Filter Implementation

Rajesh Pydipati

Introduction

Image Processing is defined as the analysis, manipulation, storage, and display of graphical images from sources such as photographs, drawings, and video. Image processing spans a sequence of three steps. The input step (image capture and digitizing) converts the differences in coloring and shading in the picture into binary values that a computer can process. The processing step can include image enhancement and data compression. The output step consists of the display or printing of the processed image. Image processing is used in such applications as television and film, medicine, satellite weather mapping, machine vision, and computer–based pattern recognition [1].

Problem statement:

Perform low pass filtering using box filter and high pass filtering using laplacian filter on an image

Approach:

Box Filter:

A spatial averaging filter in which all coefficients are equal is called a box filter. These types of filters are used for blurring and for noise reduction. The output of such a linear smoothing filter is simply the average of the pixels in the neighborhood of the pixel mask.

The idea behind smoothing filters is straight forward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood of the filter mask, the process results in an image with reduced 'sharp transitions' in gray levels. Hence the most obvious application of smoothing is noise reduction. Main disadvantage with such kind of filters is that they blur edges.

Laplacian filter:

The laplacian is a linear operator. Since, it is a derivative operator, its use highlights gray-level discontinuities in an image and deemphasizes regions with slowly varying gray levels. This would tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background. Since the laplacian image contains both positive and negative values, some scaling mechanism is usually necessary for display purposes.

Software Used: Visual C++ 6.0 to run the "C" programs.

Method: For reading the Tiff image and writing the processed image to an output file, the utilities supplied on the class website (tiff_util.c, tiff_util.h) were used. For implementing the box filter in Part 1 and Laplacian mask filter in part 2, programs were written in C.

Results:

Part 1: Box Filter

Original Image

Image after masking with a Box filter

Part 2: Laplacian Filter

Original Image

Image after masking with laplacian filter
(This was scaled for display purposes)

Image after masking with laplacian filter
(This was not scaled)

Questions

Part 1: Box Filter

1) ***What type of filter (low pass, high pass, band pass) does this mask represent?***

The Box filter represents a low pass filter. It is a type of smoothing filter. The idea behind smoothing filters is straight forward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood of the filter mask, the process results in an image with reduced 'sharp transitions' in gray levels.

2) ***Explain the problem with processing the pixels at the edge of the image and give several alternate solutions. Which one did you use in your program and why?***

At the edges of the image, we usually have the problem that the mask area falls outside of the image and hence we need to do padding (that is adding an extra row at the top and bottom and an extra column at the left side and right side of the original). In this way we ensure that the mask always has some pixels in its area. There are several approaches that are followed in padding. The padded rows may be replicated from the edges or they may all be zeros. After the processing is done these padded rows and columns may be removed. This is one approach.

Another approach is to limit the excursions of the center of the filter mask to a distance no less than (n-1)/2, if the size of the mask is n x n. But in this approach the size of the output image would be less than the input image.

In my code, I implemented the zero padding technique. Thus, even though the mask center may approach the edges, filtering is done with the section of the mask that is fully contained in the image. The main advantage was that the size of the output image was the same as the input image size, since

the padding was stripped off at the end. One of the drawbacks in this approach is that, as the size of the mask increases, the values of padding will have an effect near the edges. In this particular problem the size of the mask is small and also the size of the input image was small. So, I implemented this approach. However out of curiosity, the other approach (limiting the excursion of the center of the mask) was also implemented through minor variations in code and no big difference in the output was observed.

Part 2: Laplacian Filter

1) What type of filter (low pass, high pass, band pass) does this mask represent?

The mask given in part 2 of this problem is a type of laplacian mask. It is a type of high pass filter, since it highlights gray-level discontinuities in an image and deemphasizes regions with slowly varying gray levels.

2) How did you process the image so that none of the pixels were negative? Give several alternatives. Which one did you use in your program and why?

There are several methods to deal with the problem of negative values of pixels in the laplacian filtered image. One approach is to use absolute values. This method is not usually correct, since it produces double lines of nearly equal magnitude, which can be confusing.

The other method is to scale the values for display purposes. This was the method I followed in this problem. It is as follows:-

i) Add 255 to every pixel and then divide by 2.

This method does not guarantee that the values will cover the full range of the entire 8-bit range from 0 to 255, but all pixel values will be with in this range. The main advantage of this method is that it is simple and fast. Disadvantage is that full range of the display may not be utilized and truncation inherent in the division by 2 will generally lead to a reduction in accuracy. Due to the simplicity and inherent speed, I used this method.

References:

[1]. Microsoft Computer Dictionary: http://www.microsoft.com/mspress/books/5582.asp

[2]. Digital Image processing by Rafael .C. Gonzalez