Comparison of Digital Image Filtering Techniques

Kevin Liu

Thomas Jefferson High School for Science and Technology

Computer Systems Lab 2006-2007

January 18, 2007

Abstract

This project explores digital image filtering techniques by comparing the median and frequency filters. By testing the filters with images varying in object type (people, landscapes, or objects) and noise composition, the project determines the advantages and disadvantages of each in specific situations. The median filter implementation is developed as a typical three-by-three median filter. The frequency filter will utilize the Fourier Transformation to process the image in the frequency domain. Noise is introduced manually and is representative of the different kinds of noise experienced in data transmission.

Keywords: digital image, image filter, median filter, frequency filter, frequency domain, Fourier transformation.

Introduction

The field of image processing has wide and important uses. Any field of work that involves images or videos has uses for research in this area. The results of this project will influence how images are processed and enhanced. Applications stemming from the results of this project will be important to image and video enhancement applications because this research project provides insights on the best techniques in filtering and enhancing each kind of image.

This study will involve two filtering techniques: median filtering, and frequency filtering. First of all, algorithms will be developed for each of the two methods. After writing the code implementing these methods and conducting necessary testing, a variety of images will be put through the two filtering techniques to determine which filtering techniques work best for which types of images. The variety of images used will include images of people, objects, and landscapes. The quality, or the amount of noise, of the images used will be another independent variable.

The quality of filtered images will determined in a subjective manner through surveying a pool of people. The effects of human errors and subjectivity does not detract from this study because image filtering is done for subjective aesthetic qualities in the first place.

Background

Digital image processing such as filtering was first developed in the 1960's. As computers became cheaper and faster, real-time image processing became available and its applications boomed. Digital filtering attempts to clear out noise, or useless and distracting information, in pictures. Examples of noise include missing pixels and wrong pixels. Noise is inevitable when converting analog information into a digital form. Such a conversion occurs inside a digital camera, when the camera takes the analog picture from the lens and stores it as a digital file.

To produce high quality images, for both aesthetic and additional high level processing purposes such as edge detection, noise reduction is very helpful and often required. The median and frequency filters are two such filters.

The median filter deals with each pixel and assures it fits with the pixels around it. Therefore, it is very useful in filtering out missing or damaged pixels. The frequency filter implements the Fourier transformation to smooth out images. It is much more demanding in terms of computational complexity and processing time. Because of the sheer volume of data that normally needs to be filtered, the main problem in designing median and frequency filters is efficiency and time consumption.

This project will implement the median and frequency filters and evaluate them for their effectiveness in filtering different kinds of pictures. The results of this project may help determine the best filtering techniques to be used for an image that needs to be processed.

Development

One. The Interface Module

The interface module is the module that provides the interface between the user and the program. This module also serves the purpose of processing the input image by representing the image in a format that is easy to manipulate. In the case of this program, the two dimensional integer array was chosen.

The interface between the user and program is currently very simple. The user accesses the module through the command prompt window and specifies the image to be filtered in the command prompt.

The conversion to a two dimensional integer array structure is done by reading the image in its pgm format and filling in a two dimensional array during that process. The code for this process is reproduced below.

```
FileInputStream fstream = new
                FileInputStream(args[0]);

            // Convert our input stream to a
            // DataInputStream
                DataInputStream in =
                  new DataInputStream(fstream);

            // Continue to read lines while
            // there are still some left to read

                String type = in.readLine();

                String size = in.readLine();

                size = in.readLine(); //skip the second line

                int numCol = Integer.valueOf( size.substring(0,
size.indexOf(" ")) ).intValue();
```

```
				int numRow = Integer.valueOf(
	size.substring(size.indexOf(" ")+1) ).intValue();

				System.out.println(numRow);
				System.out.println(numCol);

				String maxValue = in.readLine();

				if( type.equals("P2") ) //if file is pgm image
				{

					int[][] data = new int[numRow][numCol];

					int row;

					for(int column = 0; column < numCol;
	column++)
					{
						for(row = 0; row < numRow; row++)
						{
							data[row][column] =
	Integer.valueOf(in.readLine()).intValue();
						}
					}
```

After the image is in that appropriate format, it calls the median and frequency filters to filter the image.

Two. The Median Filter

The median filter module is the module that encompasses all the internal processing for the median filter. Due to its relative simplicity, the median filter may be encapsulated into one module without any problems of over-complexity or size. The median filter is a Gaussian filter that slides a window of a certain size across each pixel of the image. The size of the window in this program is three by three. At each position of the window, the nine pixels values inside that window are copied and sorted. The value of the center pixel of the window is replaced with the median value of the nine pixels in the window. The implementation of this program does not do anything with the pixels on the edges.

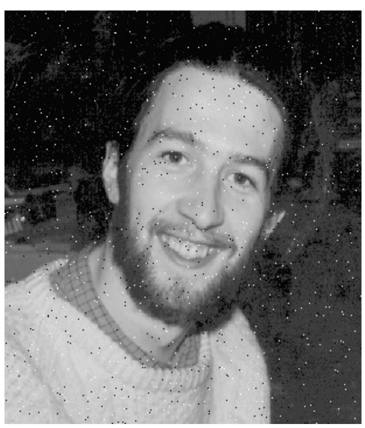
A problem was encountered during the development of the median filter. The filter is eliminating outlier pixels such as salt and pepper noise as intended, The filter seems to be shifting pixels
A piece of the code for this process is reproduced below.

```
	// slides the window across the image and copies the nine pixels inside into
	an array called "surround"
		for(row = 0; row < numRow; row++)
		{
			for(column = 0; column < numCol; column++)
```

```
			{
				if(row == 0 || row == numRow-1 || column == 0 ||
column == numCol-1)
				{
					result[row][column] = data[row][column];
					continue;
				}
				iterator = 0;
				for(r=row-1; r < row+2; r++)
					for(c=column-1; c < column+2; c++)
					{
						surround[iterator] = data[r][c];
						iterator++;
					}
				result[row][column] = insertionSortMedian(surround, 9);
	//calls the sorting method and enters the median value

			}
		}
```

During the course of developing the median filter, a problem was encountered in which the filter eliminates the the salt and pepper noise as intended, but the pixels were shifted as well.



The output produced is placed next to the original. As obviously seen, even though the filter eliminated the noise, it shifted pixels to destroy the original image.

Bibliography

Davies, E. R. (2005). Shifts Introduced by Median Filters. In E. R. Davies (Ed.), *Machine Vision* (3rd ed., pp. 66 - 78). San Francisco, CA: Morgan Kaufmann Publishers.

The median filter is used more often than its counterparts, including the mean and mode filter because the median filter suppresses noise without blurring the image. The median filter, however, may shift the edges of objects in certain situations. Straight edges are not shifted as a result of median filtering, but curved edges are shifted inward (toward the center of the curve) and bumps tend to be smoothed out as a result. The worst case scenario that produces the maximum amount of filtering shift is a circular object.

The median filter may also remove smalls details that the filter mistakes for noise. A line one pixel wide, for example, would be completely wiped out by a one-by-one median filter.

Fourier Transforms & the Frequency Domain. (n.d.). Retrieved January 11, 2007, from University of California at Berkeley Web site: http://grus.berkeley.edu/~jrg/ngst/fft/concepts.html

Signals can be expressed in two different domains, spatial and frequency domains. Considering a graph of a signal in either domain, the horizontal variable in the spatial domain is time; and the vertical variable is amplitude. For a graph in the frequency domain, the horizontal variable is frequency; and the vertical variable is amplitude. Signals are expressed in either of these domains to make processing and calculations easier. To convert a representation from the spatial to the frequency domain or vice versa, we use the Fourier Transformation. The

Fourier Transformation has applications in communications, astrology, geology, and optics. The Fourier Transformation is needed for my project because the frequency filter works on images on the frequency domain, therefore incoming images must be converted from its usual spatial domain representation to the frequency domain.