Fuzzy Algorithm for the Enhancement of Noise degraded Images

by

Holger Findling

Bachelor of Science
in Electrical Engineering
University of Central Florida
1986

Master of Science
Management / Contract Management
Florida Institute of Technology
1989

A thesis
submitted to the Graduate School of
Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Melbourne, Florida
December, 1996

We the undersigned committee
hereby approve the attached thesis

Fuzzy Algorithm
for the
Enhancement
of
Noise degraded Images

by

Holger Findling

L. Healy, Ph.D., P. E.
Adjunct Professor, Computer Science
Thesis Advisor

D. E. Clapp, Ph.D., P. E.
Associate Professor, Academic Chair,
Business

R. H. Cofer, Ph.D.
Associate Professor,
Computer Engineering

W. Shoaff, Ph.D
Associate Professor, Program Chair,
Computer Science

Abstract

Fuzzy Algorithm for the Enhancement of
Noise degraded Images

by

Holger Findling

Major Advisor: L. Healy, Ph.D.

The design and analysis of a fuzzy algorithm for the enhancement of noise degraded images are presented in this thesis. The fuzzy operator implements the mean and median filter using a powerful rule base. The strength of each filter is adjusted at the individual pixel level, and this method protects the high frequency components of image details from attenuation. The theoretical and practical results using the fuzzy algorithm are compared to classical filtering methods. A model simulating non-uniformity problems caused by camera optics, and temporal and spatial noise generated by system electronics is developed to serve as a base line and validation of this research.

Included in this paper is an explanation of the digital imaging system, classical filtering methods, fuzzy logic, and the software Imager. Imager provides for the analysis and evaluation of imaging enhancement techniques.

Table of Contents

List of Figures vi

List of Tables ix

Acknowledgment x

Chapter 1 Introduction 1
1.1 Statement of Problem 2
1.2 Proposed Solution 3
1.3 Organization of the Thesis 5

Chapter 2 The Imaging System 6
2.1 The RGB Color Space 8
2.2 Noise 11
2.2.1 Random Noise 12
2.2.2 CCD And Noise Related Problems 16
2.2.3 Frame Grabber And Noise Related Problems 17

Chapter 3 Image Processing 18
3.1 Frame Averaging 21
3.2 Pixel Gain Correction 23
3.3 Edge Detection Algorithm 28
3.4 Box Filter 31
3.5 Binomial Filter 37
3.6 Median Filter 40
3.7 Uniformity Response 43

Chapter 4 Fuzzy Expert System 46
4.1 A Brief Explanation of Fuzzy Logic 47
4.1.1 Fuzzy Sets 48

4.1.2 Fuzzy Expert System 51
4.1.3 The Fuzzy Centroid 53
4.2 The Proposed Fuzzy Algorithm 55
4.2.1 Previous Research 58
4.2.2 Gain Correction 60
4.2.3 Fuzzification Process 63
4.2.4 Inference Process 66
4.2.5 Composition And Defuzzification 70
4.2.6 Filter Analysis 72
4.3 Fuzzy Filter 76

Chapter 5 Performance Evaluation 85

Chapter 6 Summary 98

Reference 100

Appendix 102

List of Figures

Figure 2.1	Digital Imaging System	6
Figure 2.2	CIE Color Matching Functions	9
Figure 2.3	RGB Color Space	10
Figure 2.4	Scan Line 120, Random Noise of the Red Component	14
Figure 2.5	Scan Line 120, Random Noise of the Green Component	15
Figure 2.6	Scan Line 120, Random Noise of the Blue Component	15
Figure 2.7	Gray Scale 5.5	17
Figure 3.1	Test Image #1, Squares Computer Generated	20
Figure 3.2	Test Image #2, Squares Computer Simulated	20
Figure 3.3	Improvement of SNR with Frame Averaging	23
Figure 3.4	Response	24
Figure 3.5	Video Row 120, Red Pixel Gain Coefficients	25
Figure 3.6	Video Row 120, Green Pixel Gain Coefficients	26
Figure 3.7	Video Row 120, Blue Pixel Gain Coefficients	26
Figure 3.8	Test Image #2, Gain Corrected	27
Figure 3.9	Laplacian Operator	28
Figure 3.10	Laplacian Operator applied to Image Matrix	29
Figure 3.11	Test Image #1, Laplacian Operator and Normalization	30
Figure 3.12	Test Image #1, Laplacian Operator and Contrast Adjustment	31
Figure 3.13	Test Image #1, Box Filter	33
Figure 3.14	Laplacian Operator applied to Figure 3.13	33
Figure 3.15	Box Filter applied to Fixed Pattern	34
Figure 3.16	Test Image #3, 10 % Exponential Noise	35
Figure 3.17	Test Image #3, Laplacian Operator	36
Figure 3.18	Test Image #3, Box Filter and Laplacian Operator	37
Figure 3.19	Binomial Filter applied to Fixed Pattern	39
Figure 3.20	Test Image #3, Binomial Filter and Laplacian Operator	40

Figure 3.21 Median Filter 41
Figure 3.22 Test Image #3, Median Filter 42
Figure 3.23 Test Image #3, Median Filter and Laplacian Operator 42
Figure 3.24 Gray Scale 5.5 43
Figure 3.25 Gray Scale 5.5, Gain Adj., Median Filter and Box Filter 45

Figure 4.1 FAM Matrix 52
Figure 4.2 FAM System 54
Figure 4.3 Flow Diagram of Fuzzy Algorithm 58
Figure 4.4 Luminance Values For A Single Scan Line 61
Figure 4.5 Test Image #4 62
Figure 4.6 Test Image #4, Gain Corrected 62
Figure 4.7 Fuzzification Process 64
Figure 4.8 The Slope And Range Provides Filter Control 65
Figure 4.9 FAM Cube 66
Figure 4.10 Fuzzy Rules 67
Figure 4.11 Arbitrary Image Matrix #1 72
Figure 4.12 Arbitrary Image Matrix #2 73
Figure 4.13 Arbitrary Image Matrix #3 75
Figure 4.14 Arbitrary Image Matrix #4 76
Figure 4.15 Test Image #1, Fuzzy Filter 77
Figure 4.16 Edge Detection Algorithm Applied To Figure 4.15 78
Figure 4.17 Test Image #3, Fuzzy Filter 78
Figure 4.18 Edge Detection Algorithm Applied To Figure 4.17 79
Figure 4.19 Computer Simulated Image, Low Contrast 80
Figure 4.20 Edge Detection Algorithm Applied To Figure 4.19 80
Figure 4.21 Fuzzy Algorithm Applied To Figure 4.19 81
Figure 4.22 Edge Detection Algorithm Applied To Figure 4.21 81
Figure 4.23 Test Image #4, Fuzzy Algorithm 82
Figure 4.24 Scan Line 120, Red 83
Figure 4.25 Scan Line 120, Green..... 83
Figure 4.26 Scan Line 120, Blue 84

Figure 5.1 Test Image #5, Gray Scale 5.5 86
Figure 5.2 Test Image #5. Fuzzy Algorithm 86
Figure 5.3 Test Image #6, Satellite Mounted In The Cargo Bay 88
Figure 5.4 Scan Line 124, Red 88
Figure 5.5 Scan Line 124, Green 89
Figure 5.6 Scan Line 124, Blue 89
Figure 5.7 Test Image #6, Fuzzy Filter 90
Figure 5.8 Fuzzy Filter, Scan Line 124, Red 90
Figure 5.9 Fuzzy Filter, Scan Line 124, Green 91
Figure 5.10 Fuzzy Filter, Scan Line 124, Blue 91
Figure 5.11 Gain Corrected, Median Filter And Binomial Filter 92
Figure 5.12 Scan Line 124, Gain Correction, Median And Binomial Filter, Red 93
Figure 5.13 Scan Line 124, Gain Correction, Median And Binomial Filter, Green 93
Figure 5.14 Scan Line 124, Gain Correction, Median And Binomial Filter, Blue 94
Figure 5.15 Fuzzy Filter - Test Image #6 95
Figure 5.16 Median and Binomial Filter - Test Image #6 95
Figure 5.17 Differences Between Classical Filter And Fuzzy Filter 96
Figure 5.18 Test Image #6, Applying Four Iterations Of Fuzzy Filter 97

Figure A.1 Imager Bitmap Menu 104
Figure A.2 Imager Filter Menu 106
Figure A.3 Imager Process Menu 108
Figure A.4 Imager Target Window 110

List of Tables

Table 2.1 Gray Scales 11

Table 3.1 RGB Values vs. Reflectance 21
Table 3.2 Binomial Distribution 38
Table 3.3 Original Image, Frame Averaging 44
Table 3.4 Gain Correction 44
Table 3.5 Gain Correction, Median Filter and Box Filter 45

Table 4.1 Gray Scales and Linearity 71

Table 5.1 Fuzzy Algorithm Applied To Uniform Gray Scale Image 87

Acknowledgment

I would like to acknowledge the assistance and support that I received while working on this Thesis and the required course work for the degree of Master of Science in Computer Science. Especially, I would like to thank my advisor, Dr. L. Healy for his direction and support. I would also like to thank Dr. D. Clapp and staff for their support and encouragement.

Finally, I am grateful for the patience and encouragement given to me by my wife Linda and two children, Melissa and Timothy.

Chapter 1

Introduction

Limitations in image acquisition and processing can result in numerous visible defects and lack of image quality. Factors contributing to the loss of image quality include increases in color artifacts, random noise, spatial noise, and losses in contrast, sharpness, and uniformity. The degradation of images can be separated into two categories, spatial degradation and point degradation. A common source of point degradation is random noise generated by the imaging electronics. Spatial degradation may be caused by camera optics, atmospheric interference, and camera motion. Image problems created by spatial degradation is most difficult to reverse in the presence of additive noise [V. Z. Mesarovic, N. P. Galatsanos, and A. K. Katsaggelos] [1]. Relevant information becomes less visible to an observer with the increase of noise and the presence of image abnormalities.

Numerous image processing techniques have been developed to restore and improve various aspects of image quality. The attenuation of the noise signal is one of the first steps required to restore degraded images. Some of the classical approaches include frame averaging, gain and offset normalization, histogram equalization and filtering. The classical filtering methods can be divided into two categories, linear and non-linear. Examples of linear filters include box filters (smoothing), binomial filters, Wiener filters, FIR, and IIR. The general classification for non-linear filters is centered

around median-related filters. Included in this group are rank-order filters, order statistic filters, weighted median filters, and stack filters.

Non-linear filters are better suited for noise reductions than linear filters. In [2] Zeng states that "Optimal linear filters can hardly be robust as their performances depend heavily on the characteristics of the signal as well as the type of noise that corrupts the signal." He explains that the preservations of relevant information in images and the elimination of noise are two contradictory aspects in signal and image processing. Trying to preserve signal features limits the ability to reduce noise. The different types of noise present in an image may require the application of several filters; and, the sequence of applying the filters is critical. Median filters are optimal in performance when filtering double-exponential noise and impulse noise; whereas, mean filters perform better than median filters eliminating Gaussian noise, but fail to reduce impulse noise. Neither of the two types of filters amplify the signal of interest; *i.e.*, the signal-to-noise-ratio is improved only by the ability of the filter to effectively reduce noise.

1.1 Statement of Problem

Classical filtering methods are optimal in performance only when characteristics of the relevant signal and the noise signal are known. In [3] W. Ho Pun and B. D. Jeff state "Any given restoration method usually performs best for a particular class of images only. This limitation usually arises from the implicit or explicit assumptions regarding image information and degradation which are associated with a particular algorithm."

Using classical methods to provide a reduction of the noise signal requires that all video data becomes subject to the filtering process. The ability to eliminate the noise signal is highly limited when the high frequency component of the relevant signal requires preservation. The strength of linear and non-linear filters is not adjustable on a pixel by pixel basis; *i.e.*, the process limits the substitution of the corrupted pixel value with a rigid replacement. This constraint results in the attenuation of the high frequency component of the relevant signal. A loss of image detail and severe blurring of edges may result in this process.

The problems encountered using classical filtering methods suggest that a noise reduction algorithm must have the ability to protect image details by determining the type of filtering necessary at each pixel level. The algorithm must be flexible to adapt to a great variety of signal characteristics adjusting the strength of the filter to preserve relevant image details. Classical filtering methods fail in this regard.

1.2 Proposed Solution

This research involves the analysis and development of an image correction algorithm using a fuzzy controller to apply the principles of approximate reasoning. The objectives include the preservation of the high frequency signal components and the reduction of temporal and spatial noise. Knowledge of the digitizing system and the ability to obtain several frames of data shall be considered in solving the problem.

Applying fuzzy logic to the field of imaging is a relative new approach. The decision to implement fuzzy logic in this proposed algorithm is based on information stated in [4] and [5]. In [4] a video camera image stabilizer has been developed which limits the effects of operator movement in the video. The fuzzy set theory was successfully applied to discriminate between operator and object movement. In [5] F. Russo and G. Ramponi design and evaluate a rule-based fuzzy operator that performs detail sharpening of objects in images. Sharpening of the images was performed by discriminating between detail and noise based on the amplitude of luminance changes and correlation between adjacent pixels.

In these studies the concept of approximate reasoning was applied to digital image processing. Fuzzy rules consisting of IF-THEN statements are expressed in linguistic terms to provide for human like reasoning. A mathematical model defining the relationship between inputs and outputs of the system controller is not required. The relationship between the input variables and the output variables is provided by the set of fuzzy rules. The results show that fuzzy logic can be highly instrumental in solving complex image restoration problems. Fuzzy logic provides for human reasoning, and the fuzzy algorithm can be designed to evaluate data at the individual pixel level. The design of the proposed algorithm differs from [5] with the emphasis placed on improving the signal-to-noise-ratio instead of sharpening edges.

1.3 Organization of the Thesis

The remaining sections of this Thesis present the results of the theoretical and practical research conducted. Chapter 2 describes the imaging system used for this study and identifies noise related problems. An image model with temporal and spatial noise, and image non-uniformity is defined in Chapter 3. A description and evaluation of classical filtering methods is included in this section. Chapter 4 provides an explanation of fuzzy logic and presents the design of the proposed algorithm. A discussion of results obtained using the fuzzy algorithm is included. A comparison between classical filtering methods and the fuzzy algorithm is provided in Chapter 5. A conclusion is derived by applying the filters to a natural filmed image.

A short explanation of the software Imager is provided in the Appendix. Imager was designed to work in Windows 95, and is an integral part of this research to provide for the evaluation of the new fuzzy algorithm and the classical filtering methods. Images used for the evaluation can be viewed and stored in a 320 x 240 bitmap, and the pixel data can be saved to a file compatible with the Excel Spreadsheet. To validate this research, experimental results using Imager are presented throughout this report.

Chapter 2

The Imaging System

The fundamental purpose of the digital imaging system is to provide for the digital recording and transmission of information obtained from the visible spectrum. Digital images captured to computer memory can be further processed and analyzed by software methods or directly viewed on the display monitor. Figure 2.1 displays a digital imaging system consisting of four major components which are the camera, the frame grabber, the computer, and the display monitor.

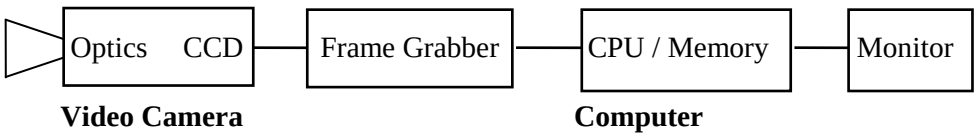


Figure 2.1 Digital Imaging System

The camera captures the scene and converts the visible signal into an analog video signal. Inside the camera head resides a charge coupled device (CCD) sensor providing an array of video pixels. The CCD array comprises of light-sensitive

silicon cells called photodiodes. Each photodiode produces an electrical charge that is proportional to the number of photons falling incident to the sensor plane. The charge generated by the photoelectric effect is transferred from each pixel by vertical registers and one horizontal register to an electronic amplifier. The output of the amplifier combined with timing signals generates the composite video signal.

The video signal contains information that creates video frames consisting of two interlaced fields. The interlaced signal provides for an evenly illuminated image when displayed on a television picture tube. Each frame consists of 525 horizontal scan lines with 640 pixels in each line. The time duration of one horizontal scan line measures 63.556 microseconds, and it includes eight microseconds required for the horizontal retrace pulse. This allows for approximately 86.8 nanoseconds of processing time for each pixel. In [6] C. Poynton provides additional details of technical parameters of the composite NTSC and PAL.

The horizontal and vertical synchronization pulses that are imbedded in the analog video signal provide the frame grabber with the necessary timing to lock on and uniformly sample the continuous analog signal $x_a(t)$ every T second to produce a discrete signal x(n), as defined in equation (2-1). The frame grabber quantizes the samples to a discrete set of amplitude levels matching the dynamic range of the RGB color space. The digitizing accuracy of the frame grabber affects the image quality and the repeatability of measuring image details. In this thesis the resolution of the final digitized image is 320 x 240 pixels.

$$x(n) = x_a(nT) \tag{2-1}$$

The discrete video signal is stored in the computer memory to an audio-video-interleaved file (AVI). The AVI format is an extension of the Resource Interchange File Format (RIFF), which provides for the storage and synchronization of video and audio data streams. The simplest AVI format comprises of uncompressed video data and no audio data. Either the AVI file or the real time video signal from the frame grabber can be viewed on the computer display monitor at a rate of 30 frames per second. The frame rate is limited by the data transfer rate of the interface between the video board and the computer. Images can also be captured directly to the hard drive, but the process requires that the media is not fragmented. A fragmented hard drive can result in a lower frame rate and dropped video frames. Capturing an image directly to computer memory provides for the highest image quality.

2.1 RGB Color Space

The trichromatic nature of human vision allows any color to be defined by red, green, and blue intensity components. The intensity components are also known as the X, Y, and Z tristimulus values, which can be derived from the CIE chromaticity diagram and the color matching functions. Figure 2.2 displays the color matching functions, and equation (2-2) shows the RGB transform. It should be noted that only human perception associates' color with the electromagnetic waves in the visible spectrum. The visible spectrum comprises of radiant energy with the wavelength among 380 and 720 nanometers.

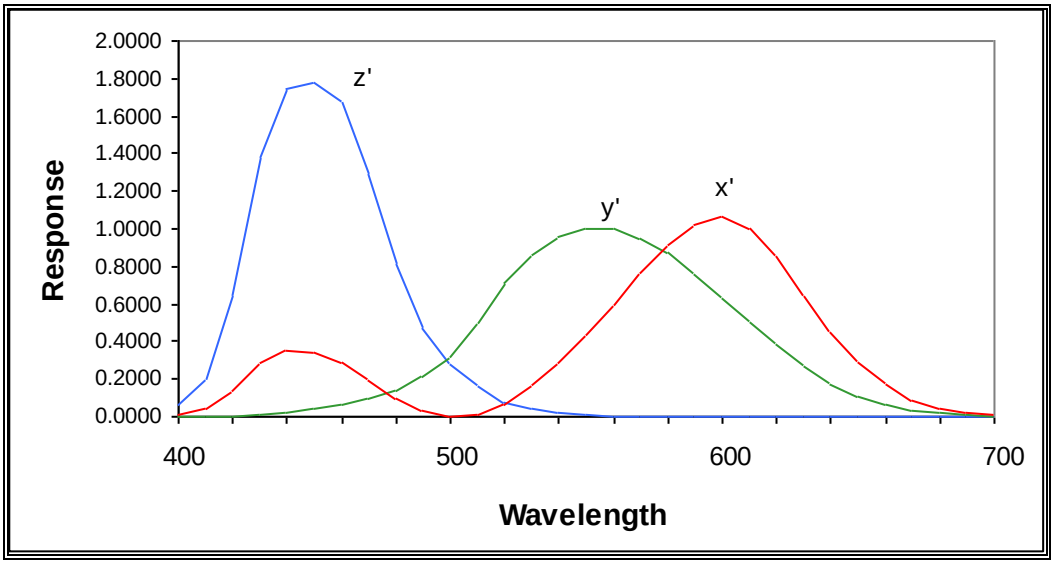


Figure 2.2 CIE Color Matching Functions

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 2.179151 & -0.946884 & -0.263777 \\ -1.382685 & 2.327499 & 0.045336 \\ 0.007989 & -0.015138 & 1.333346 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad (2\text{-}2)$$

The RGB color space can be defined by mapping the red, green, and blue intensity components into the Cartesian coordinate system. The dynamic range of the intensity values is scaled from 0 to 255 counts, and each primary color is represented by eight bits. The RGB color space shown in Figure 2.3 displays 16.77 million discrete colors. The red, green, and blue corners of the cube indicate 100 percent color saturation.

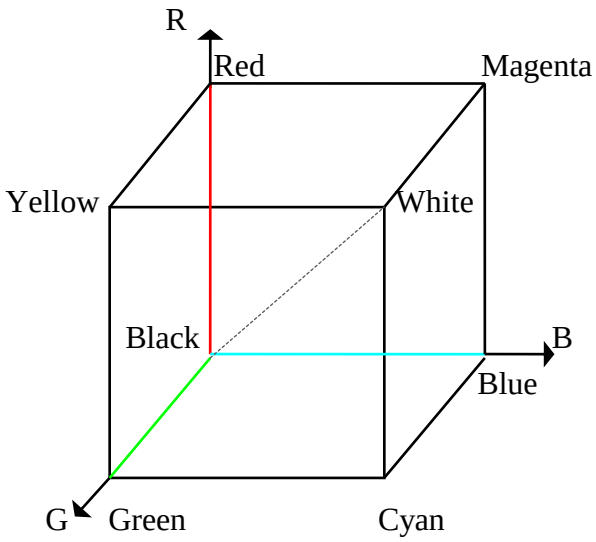


Figure 2.3 RGB Color Space

An imaginary line can be drawn from the origin of the cube to the furthest opposite corner. Along this line are 256 achromatic colors representing possible shades of gray. Black resides at the origin of the color cube, and white is at the opposite corner. The fourteen gray scales recorded in Table 2.1 are located on this color line. The RGB system enables the reproduction of any color within the color space by using an additive mixture of the primary colors. For an example, White is the sum of 255 counts of red, green, and blue, and the function is usually expressed by RGB(255, 255, 255).

Gray Scale	Reflectance	Red	Green	Blue
Dark Current	NA	45	45	45
Gray 0.5	1 %	56	56	56
Gray 1.0	3 %	66	66	66
Gray 1.5	5 %	77	77	77
Gray 2.0	7 %	88	88	88
Gray 2.5	10 %	98	98	98
Gray 3.0	13 %	108	108	108
Gray 3.5	16 %	132	132	132
Gray 4.0	18 %	145	145	145
Gray 4.5	23 %	148	148	148
Gray 5.0	27 %	156	156	156
Gray 5.5	32 %	161	161	161
Gray 6.0	36 %	170	170	170
Gray 6.5	41 %	177	177	177
Gray 7.0	45 %	182	182	182
White 10.0	90 %	ND	ND	ND

Table 2.1 Gray Scales

2.2 Noise

System noise comprises of noise currents and voltages in the electronic circuits that degrade or obscure the signal of interest. The electronic noise may result in a random point degradation or generate fixed patterns in the image. The noise signal ultimately limits the detection and processing of the relevant signal and decreases the quality of the image. Noise sources external to the imaging system further reduce the ability to obtain a perfect image.

The quality of the imaging system can be expressed in terms of resolution and the signal-to-noise-ratio (SNR). The signal-to-noise-ratio is a measure of the ratio of the average signal power to the average noise power. It is common to express the SNR in decibels, where it reflects the ratio of the mean-square signal to the mean-square noise, as defined in equation (2-3) and (2-4). The maximum SNR for an eight bit digitizing system is limited to 48.16 dB. It is assumed that the least significant bit is toggled by pixel jitter and noise.

$$S/N\,(dB) = 10 \log \left[\overline{S^2(t)} \,/\, \overline{N^2(t)} \right] \tag{2-3}$$

$$S/N\,(dB) = 20 \log \frac{S(t)}{N(t)} \tag{2-4}$$

2.2.1 Random Noise

The generation of random noise is inherent in all electronic circuits. The vertical and horizontal registers in the CCD, electronic amplifiers, and cabling are examples of major contributors in generating electronic noise. Two significant components of electronic noise is Thermal noise or Johnson noise, and Shot noise. Shot noise is generated in semiconductor devices due to leakage currents and statistical fluctuations in charge carriers. The leakage currents or dark currents are

currents flowing without the presence of a signal. One important characteristic of Shot noise is that the noise is Gaussian distributed with a zero mean value.

Thermal noise is the electrical noise generated from the random motion of electrons in a conductor. The noise is also Gaussian distributed. The attenuation of Gaussian noise is possible through low pass filtering and limiting the bandwidth of the imaging system. Decreasing the electrical bandwidth of the imaging system reduces the Shot noise and Thermal noise. The minimum video bandwidth required for the National Television System is 4.2 MHz.

The average random noise of the imaging system can be measured by filming a uniform scene and capturing the individual frames to bitmaps. The imaging statistics are calculated from the pixel data contained in each bitmap. Random noise of a pixel (RN_P) is the standard deviation of the pixel output signal (S_{pi}) obtained over a specified number of frames (F), as defined in equation (2-5). The average random noise is the mean value of the RN_P for all the pixels in the image, as defined in equation (2-6).

$$RN_P = \left[\frac{\left[F * \sum_{i=0}^{i=F-1} S_{Pi}^{\,2} \right] - \left[\sum_{i=0}^{i=F-1} S_{Pi} \right]^2}{F\,(F-1)} \right]^{1/2} \qquad (2\text{-}5)$$

$$\text{Average RN} = \frac{1}{F} \sum_{i=1}^{i=F} RN_{Pi} \qquad (2\text{-}6)$$

Figure 2.4 through Figure 2.6 graphically display one scan line of random noise. The noise was recorded by filming the uniform gray scale 5.5, defined in Table 2.1. The average SNR measured for the red, green and blue component is 41.23 dB, 38.30 dB, and 39.28 dB respectively.

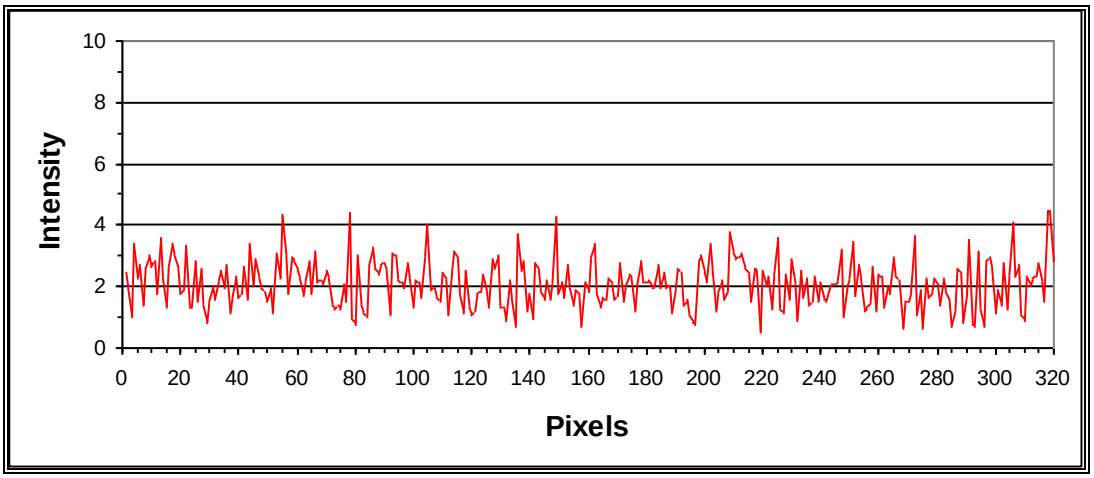


Figure 2.4 Scan Line 120, Random Noise Of The Red Component

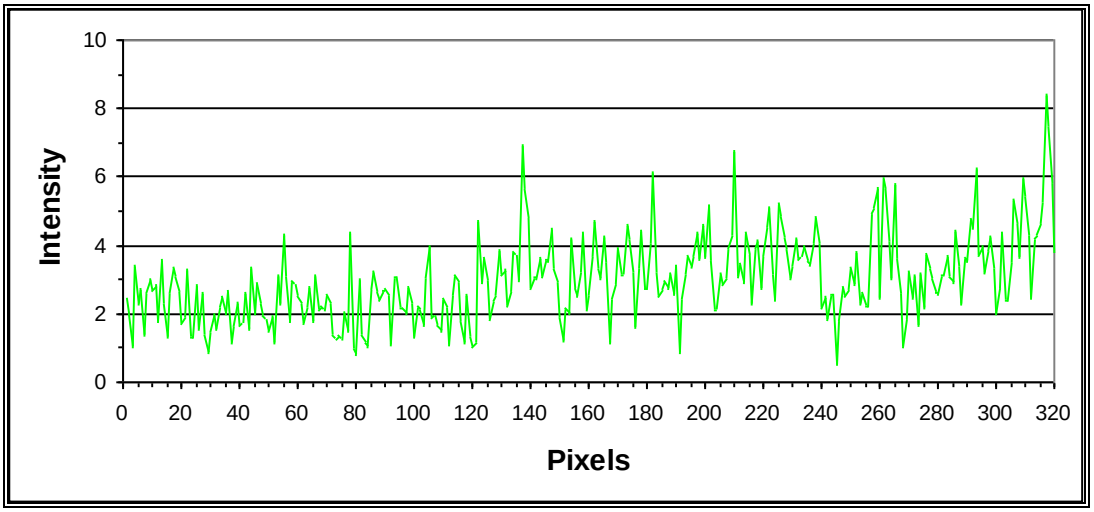


Figure 2.5 Scan Line 120, Random Noise Of The Green Component

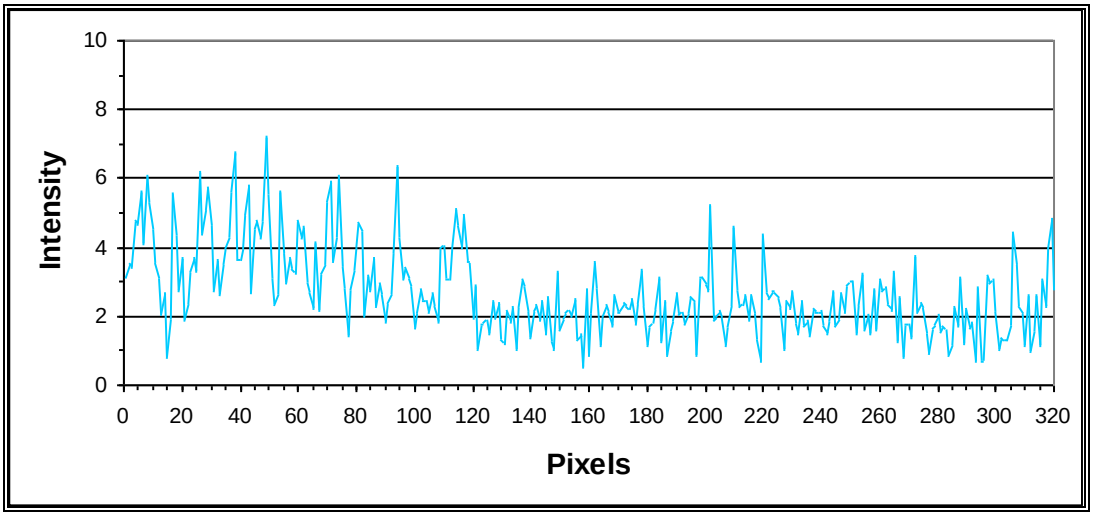


Figure 2.6 Scan Line 120, Random Noise Of The Blue Component

2.2.2 CCD And Noise Related Problems

The quantum efficiency (QE) of photocells in a CCD array greatly varies between the individual elements and the wavelength of light. The quantum efficiency of the average CCD is approximately 16 % at 650 nm, 22 % at 550 nm, and 14 % at 450 nm. Figure 3.4 shows the average response of a camcorder filming gray scales. The gray scales and their associated reflectance is listed in Table 2.1. Detectors engaging in front side illumination show a greater surface reflectivity at the lower wavelength, and this characteristic results in a decreasing quantum efficiency in the green and blue wavelength. Backside illuminated detectors provide for better quantum efficiency, but the device requires the manufacturing of a thinner substrate near ten microns. Maintaining the flatness of a substrate with less than 10 micron thickness is a manufacturing problem. The non-flatness of a substrate results in poor focus and the non-uniform response of individual cells. Figure 2.7 illustrates the non-uniformity problems caused by the camera optics and the CCD. In this image the gray scale is showing a green tint due to the lower response towards the blue color.

Noise generated by the CCD can be categorized into Read Noise and Dark Current Noise. Read noise is a combination of thermal noise and deviation in sensitivity at the sense node. The sense node at the end of the horizontal register is the final collecting point for the electrical charges created by the received photons. At this node the charges are converted to signal voltages, and any deviation in sensitivity results in point degradation.

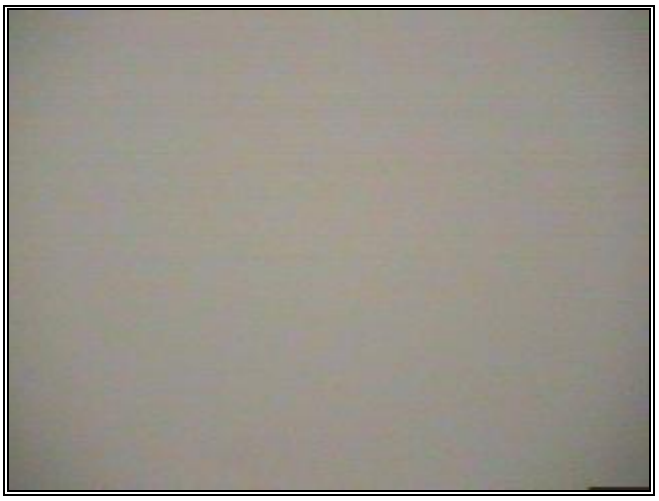


Figure 2.7 Gray Scale 5.5

2.2.3 Frame Grabber And Noise Related Problems

The random noise introduced in the digital video signal is partially caused by the frame grabber due to phase shifts, frequency errors, and pixel jitter. Slight variations in the time of sampling, see equation (2-1), can result in the inaccurate digitization of intensity values. Pixel jitter limits how precisely a frame grabber quantizes the gray scale intensities along each horizontal scan line. A typical performance of a phase locked loop timing circuit creates jitter up to $\pm$ 20 nanoseconds. Digital pixel clocks and programmable timing circuits have achieved $\pm$ 5 nanoseconds; however, the implementation drastically increases the cost of the system.

Chapter 3

Image Processing

Digital image processing serves the purpose of enhancing the visual appearance of images, and preparing the images for an evaluation of features and structures present. The evaluation of image quality includes the measurement of signal to noise ratios, dynamic range, contrast, sharpness, and uniformity. The restoration of images degraded by noise must consider two aspects of signal processing, the attenuation of temporal and spatial noise and the preservation of the signal. Classical methods and algorithms available include regularized constrained total least squares [1], optimal median-type filtering [2], histogram equalization of color images [7] [8], and numerous filtering methods of Gaussian noise.

One concern met in processing noise degraded images are that classical methods and algorithms target a particular noise or image problem. This requires an analysis of the image problem before selecting a correction method, and the analysis may be extremely difficult to perform. For an example, a space probe traveling to a distant star becomes subject to many external interferences, and the nature of the interference may be impossible to predict accurately. Filtering methods such as the least squares restoration algorithm may be optimal in the presence of Gaussian noise, but the algorithm fails in filtering double exponential noise. Testing a particular algorithm requires that a specific noise degraded image or model is

available. Here a real time noise model is difficult to emulate, and it is often replaced with a simpler model.

For this Thesis a real time noise model is developed to serve for the evaluation and comparison of classical filtering methods and the proposed fuzzy algorithm. The model is derived from filming and digitizing noise currents of the imaging system and fourteen achromatic color sheets. The noise currents of the imaging system were measured without light energy entering the camera. The color sheets representing the fourteen gray scales were filmed in daylight, while maintaining the same camera setup for each film clip. The film clips were captured directly to the computer memory and than recorded to an uncompressed audio-video-interleaved file (AVI).

Data sets of various signal-to-noise-ratios can be generated from the AVI files. Each data set consists of fifteen 320 x 240 bitmaps, which were created using the same number of frame averages. The bitmaps provide a source for generating test images comprising of non-uniformity problems and video noise. The signal-to-noise-ratio of a test image is controlled by selecting a data set that was developed with a different frame average. For an example Test Image #1 displayed in Figure 3.1 is a computer generated image without defects and noise present, and Test Image #2 displayed in Figure 3.2 is a computer simulated image that includes non-uniformity problems and real time noise. Test Image #2 was developed from the data set displayed in Table 3.1. The fifteen bitmaps of this data set are the result of one hundred frame averages. Both test images shall serve for the digital image processing and analysis in this Thesis.

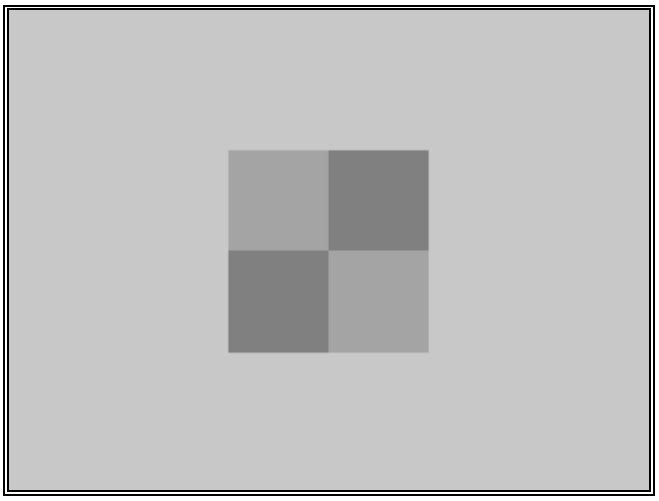


Figure 3.1 Test Image #1, Squares Computer Generated

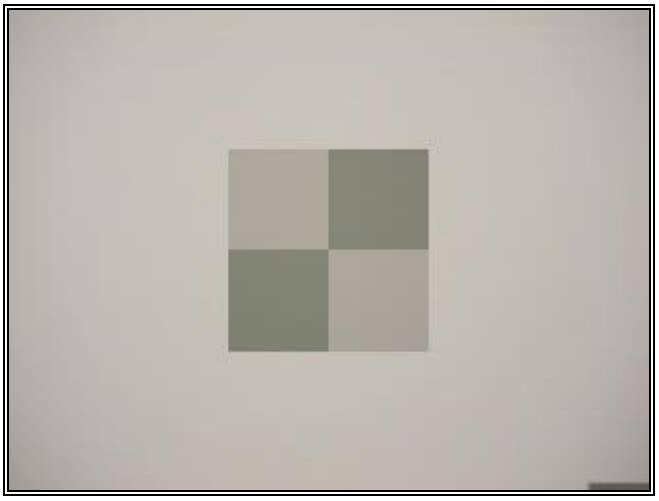


Figure 3.2 Test Image #2, Squares Computer Simulated

Gray Scale	Reflectance	Red	Green	Blue
Dark Current	NA	46	51	42
Gray 0.5	1 %	55	60	50
Gray 1.0	3 %	70	70	60
Gray 1.5	5 %	80	82	71
Gray 2.0	7 %	91	93	81
Gray 2.5	10 %	100	101	89
Gray 3.0	13 %	112	114	102
Gray 3.5	16 %	135	138	123
Gray 4.0	18 %	149	147	131
Gray 4.5	23 %	155	149	138
Gray 5.0	27 %	161	155	142
Gray 5.5	32 %	167	160	149
Gray 6.0	36 %	175	168	158
Gray 6.5	41 %	180	172	163
Gray 7.0	45 %	185	178	167
White 10.0	90 %	ND	ND	ND

Table 3.1 RGB values vs. Reflectance

3.1 **Frame Averaging**

The most effective approach to improve the image quality is to select and design the best imaging hardware possible. Noise generated by the camera, cabling, and the digitizing electronics should be limited and corrected at the hardware level. Collecting more of the relevant signal is one of the best methods to resolve issues

with noise degraded images. Unfortunately, there are many instances were technical limits and cost renders further hardware improvements not possible or practical. Typically, commercially available cameras and camcorders do not provide for a manually controlled signal integration time. Once all of the hardware improvements are exploited, further attenuation of noise and corrections for image non-uniformity are possible through the implementation of algorithms in software.

Frame averaging is one of the best software solutions available to reduce temporal and spatial noise. Many image digitizing circuits and frame grabbers are capable of performing the frame averaging real time; however, post processing the captured frames is not a serious problem providing that enough computer memory is available. One hundred frames of color images with a resolution 320 x 240 pixels and true color (24 bits) requires 23.040 Megabytes of memory. Improvement in the signal-to-noise-ratio is proportional to the square root of the number of frames added together [9]. The SNR can be improved by approximately 10 dB when averaging 100 frames. Figure 3.3 illustrates the relationship between improvements of the SNR and frame averaging.

Limitation for applying frame averaging includes that scenes must be static for the number of frames averaged, and the lighting conditions must be fixed. Blurring of image details and the loss of contrast may be observed when frame averaging non-static scenes. Frame averaging becomes a powerful imaging enhancement tool when it is combined with pixel gain and offset correction.

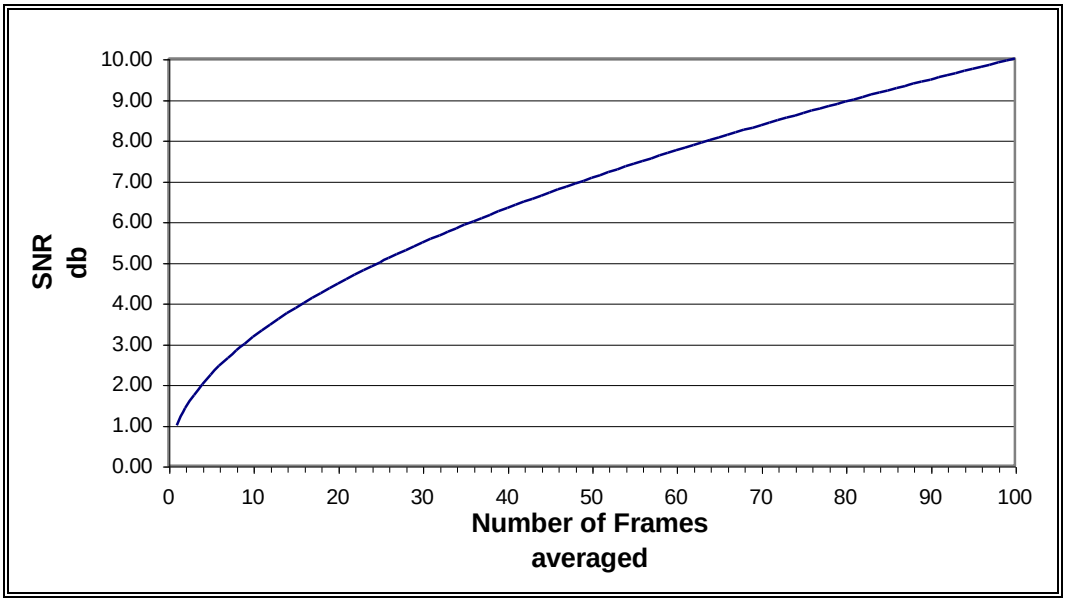


Figure 3.3 Improvement of SNR with Frame Averaging

3.2 Pixel Gain Correction

The response of the image system used for this research was established by filming achromatic color sheets with a matte finish. The sheets measured 9" x 12" in size and are devoid of chromaticness. The data in Table 3.1 represents the reflectance of the gray scales, and the average RGB values measured for the colors red, green and blue. The tolerance of the reflectance for each gray scale is 0.5 percent. The basic gray scale consists of 10 steps, gray scale 1 through 10, and intermediate grays in 0.5 increments are also available for fine tuning the scale. Figure 3.4 shows the gray scales and the average response of the three color

components of the image system. Gray scales greater than 7.0 resulted in saturation and were eliminated from the graph.

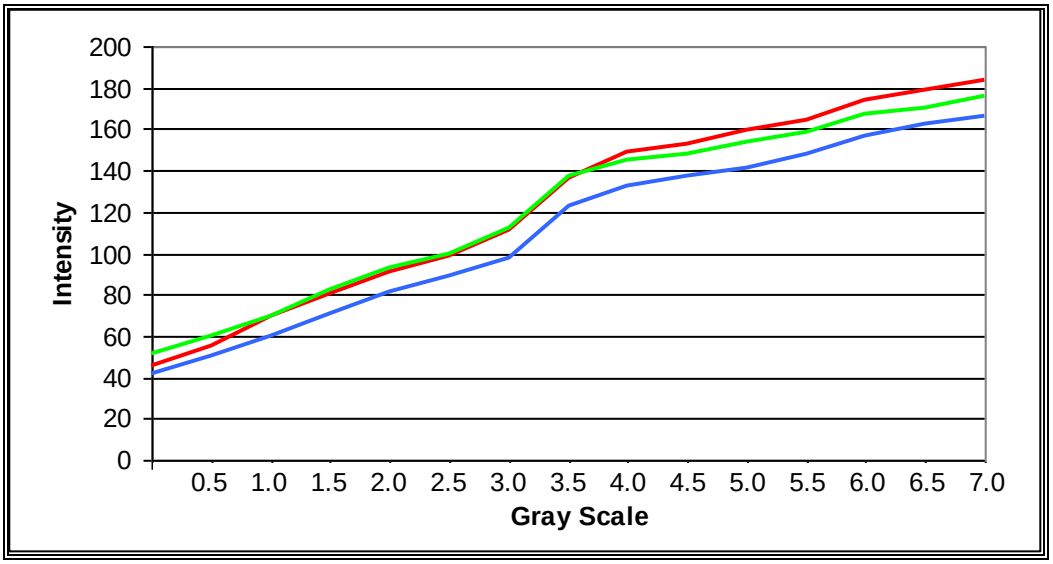


Figure 3.4 Response

In Figure 3.4 it is observed that the response is not linear over the dynamic range and varies as much as fifteen counts in intensity between the colors red, green, and blue. Video signals below 45 counts cannot be displayed due to the cutoff levels of the digitizing system. In addition to the non-linear response images show uniformity problems. The corners of the test image displayed in Figure 3.2 are noticeable darker than the center of the image. This effect is caused by non-uniformity problems in the camera optics and the CCD. The correction for pixel

response and non-uniformity requires that a gain and offset calibration is performed for each individual pixel.

Figure 3.5 through Figure 3.7 display a set of response curves for the colors red, green, and blue respectively. These response curves are the result of filming the gray scales and performing 100 frame averaging. The response measured for each pixel serves as a calibration gain coefficient. The gain coefficients are stored in twelve lookup tables, and they are unique for one specific zoom setting of the camera lens. The pixel gain calibration is implemented adjusting the pixel luminance values by finding the two closest values in the lookup tables and interpolating between them. The pixel correction is performed separately for each of the three colors red, green, and blue. A perfect pixel gain correction requires a set of lookup tables for the different zoom settings.

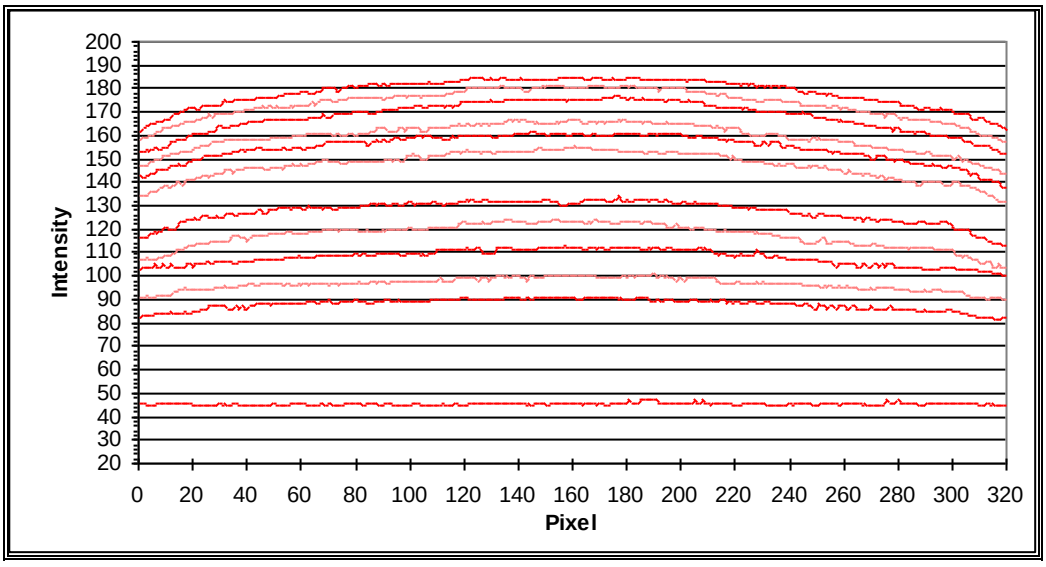


Figure 3.5 Video Row 120, Red Pixel Gain Coefficients

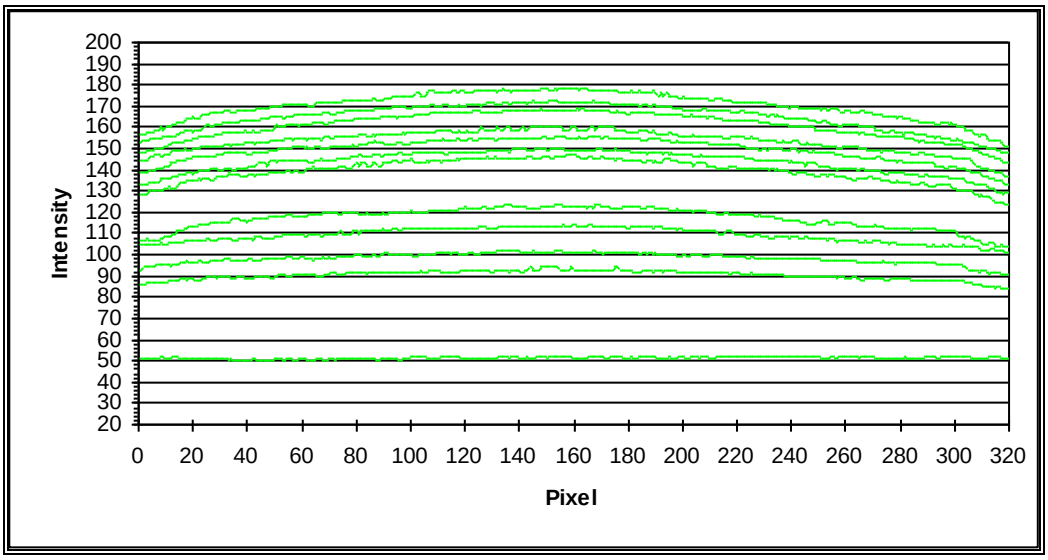


Figure 3.6 Video Row 120, Green Pixel Gain Coefficients

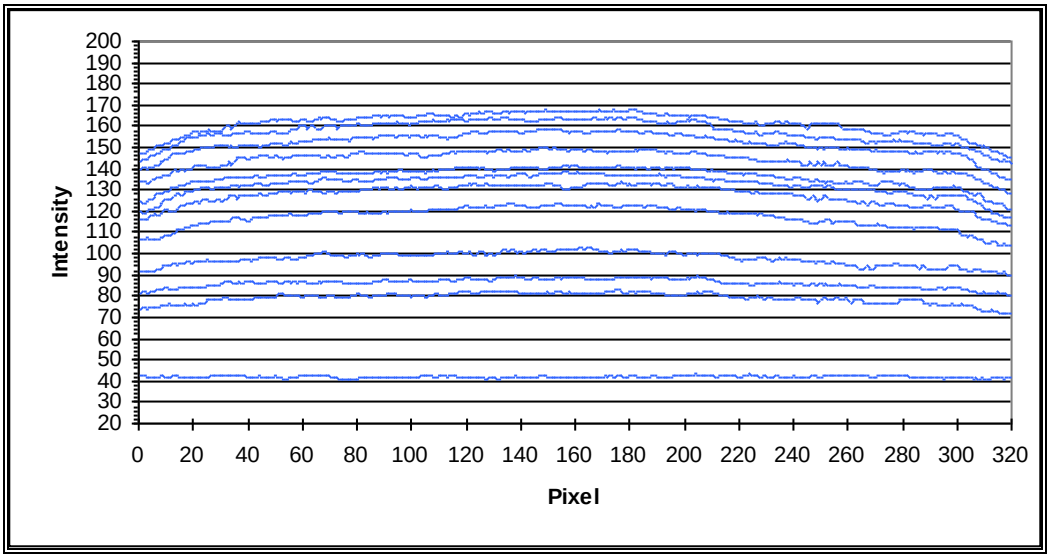


Figure 3.7 Video Row 120, Blue Pixel Gain Coefficients

Figure 3.8 illustrates the result of applying the pixel gain correction to the Test Image #2 displayed in Figure 3.2. The bitmap is restored to an achromatic image and corrected for non-uniformity. Details of the image and the overall sharpness is maintained.

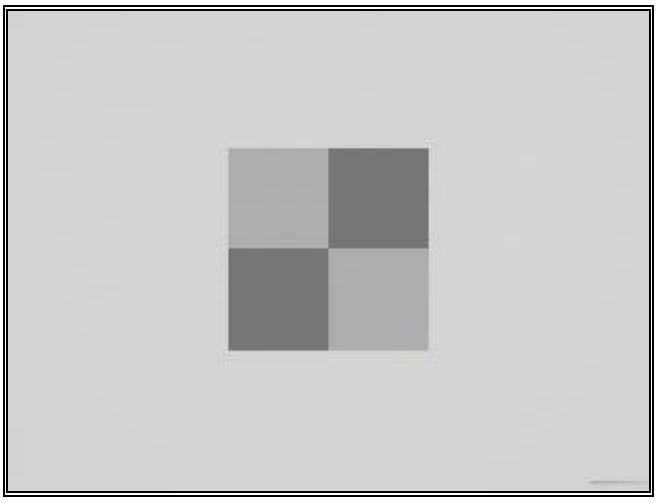


Figure 3.8 Test Image #2, Gain Corrected

3.3 Edge Detection Algorithm

One method to detect non-uniform regions and edges in an image is to measure the gradient of pixel luminance values. The Laplace operator is a convolution mask capable of detecting changes in brightness and discriminates uniform areas. The 3x3 kernel shown in Figure 3.9 is superimposed onto the image matrix. As the window slides across the matrix new luminance values are calculated for each pixel using equation (3-1).

-1	-1	-1
-1	+8	-1
-1	-1	-1

Figure 3.9 Laplacian Operator

$$P_{x,\,y} = \sum_{i,\,j\,=\,-m}^{+\,m} W_{i,\,j} * P_{x\,+\,i,\,y\,+\,j} \qquad (3\text{-}1)$$

The Laplacian operator assigns a new luminance value of zero (black) to a pixel, when the particular image section under the convolution mask is uniform and void of edges. Any luminance differences between the center pixel and the

surrounding pixel results in a non-zero value. It is possible for the Laplace operator to generate negative values, which will be displayed on the CRT as black. Since negative luminance values result in the loss of image when displayed, it is customary to normalize the image by adding 128 counts to each pixel value. This method allows all differences and edges to be visible.

60	60	60	80	80
60	60	60	80	80
60	60	60	80	80

A

0	0	- 60	60	0
0	0	- 60	60	0
0	0	- 60	60	0

B

128	128	68	188	128
128	128	68	188	128
128	128	68	188	128

C

255	255	0	0	255
255	255	0	0	255
255	255	0	0	255

D

Figure 3.10 Laplacian Operator applied to Image Matrix

An arbitrary section of an image matrix is shown in Figure 3.10 before (A) and after (B) the Laplacian operator is applied. The original image matrix (A) comprises of two uniform areas separated by a vertical edge. The result shows that the uniform areas received the RGB value 0, but the edge was assigned a negative and positive value. Figure 3.10 C shows the result after the operator is applied and the image

matrix is normalized. The pixel values are biased to a brightness value 128, which is the center of the gray scale. For inspection and evaluation purposes it is sometimes desirable to highlight local luminance differences by increasing the contrast to a maximum. This can be accomplished by reassigning a luminance value 255 (white) to pixels with current values of 128, and all other pixels receive a new value 0 (black). An example is shown in Figure 3.10 D.

This method is used for the visual evaluation of several images in this study. Figure 3.11 shows the results applying the Laplacian operator to the Test Image #1 displayed in Figure 3.1 and normalizing it to a 128 count. Figure 3.12 illustrates the same image scaled to zero and 255.

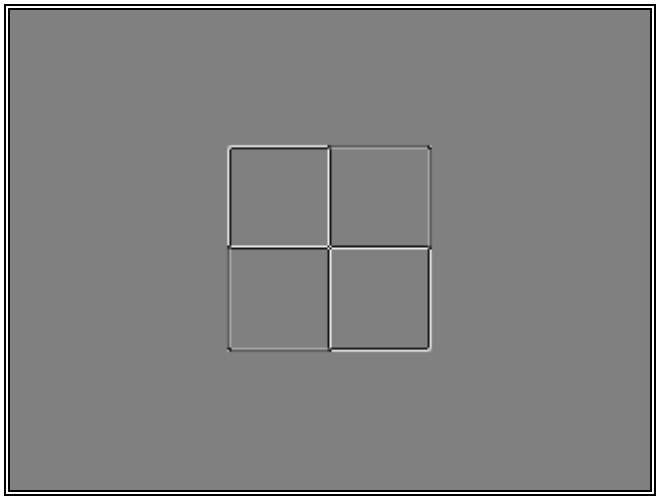


Figure 3.11 Test Image #1, Laplacian Operator and Normalization

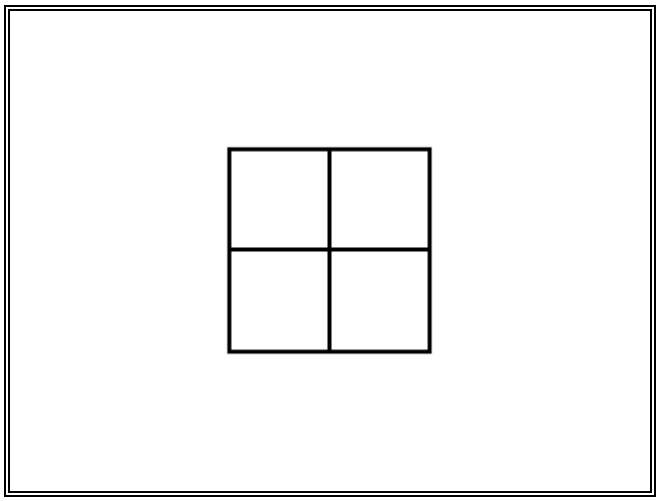


Figure 3.12 Test Image #1, Laplacian Operator and Contrast Adjustment

3.4 Box Filter

The suppression of spatial noise can be achieved by applying a low pass filter to an image matrix. One widely used filter in the imaging field is the box filter. The implementation of the box filter can be accomplished using a convolution mask. In equation (3-2) the operator sums the pixel values confined within the mask and scales the result by the total weight. Scaling the result after applying the mask ensures that unity gain is maintained.

The box filter provides smoothing of local image sections by averaging luminance values within the mask. In this process the high frequency components of spatial noise and image details are eliminated. The filter cannot discriminate

between relevant information and the noise signal. Canceling high frequency components associated with edges results in the loss of sharp transitions between the image details. Human visions interprets the resulting image out of focus and blurred.

$$P_{x,y} = \frac{\sum_{i,j=-m}^{+m} W_{i,j} * P_{x+i,y+j}}{\sum_{i,j=-m}^{+m} W_{i,j}} \qquad (3\text{-}2)$$

Equation (3-3) defines the 3x3 kernel for the box filter. Figure 3.13 illustrates the results of applying the box filter to the Test Image #1. The result of applying the Laplacian operator to Figure 3.13 is displayed in Figure 3.14. It is noted that the outline of the four squares consist of double lines instead of single lines. The dual lines indicate that high frequency components of the edges of the squares were attenuated by the filter. The smoothing operation transformed the slope of the edge from one pixel to a ramp of three pixels wide.

$$R = \frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad (3\text{-}3)$$

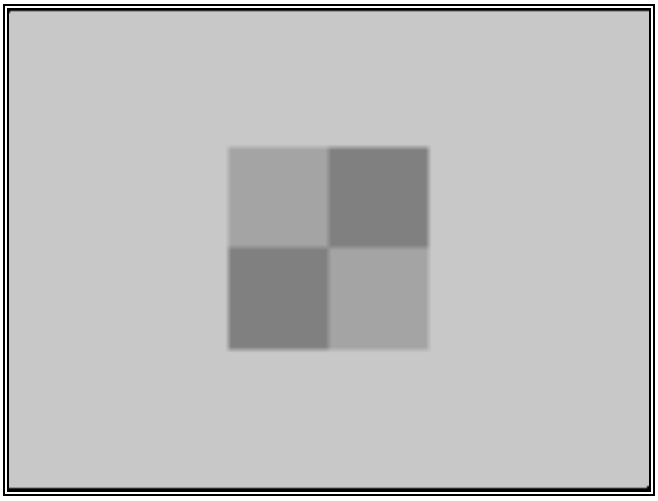


Figure 3.13 Test Image #1, Box Filter

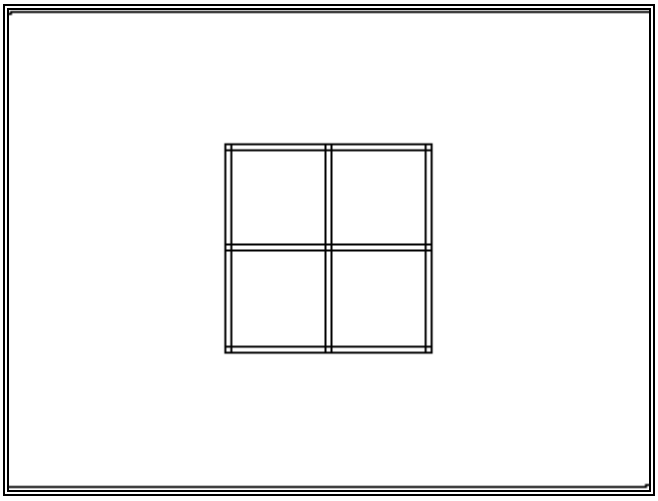


Figure 3.14 Laplacian Operator applied to Figure 3.13

The analysis of the box filter given by B. Jahne in [10] explains that the filter does not meet all of the characteristics of a good smoothing filter. The box filter is not a good low pass filter, and it fails to attenuate fixed patterns proportionally with an increase in the pattern's frequency. Figure 3.15 illustrates this characteristic of the box filter. Figure 3.15 (A) and (B) show an arbitrary image section consisting of fixed patterns repeating every second and third pixel respectively. The result of applying the box filter to these images is displayed in Figure 3.15 (C) and (D). The fixed pattern displayed in (A) consist of a higher repetitive frequency than the pattern in (B), and the filter was less effective in smoothing the image. Figure 3.15 (D) shows the result of applying the box filter to (B); and, the filter successfully attenuated the fixed pattern.

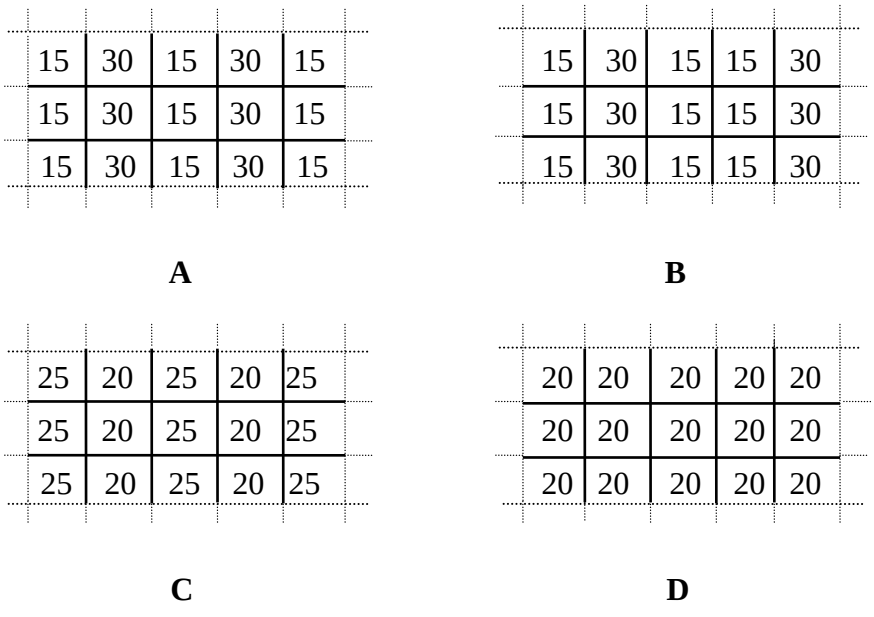


Figure 3.15 Box Filter applied to Fixed Pattern

Test Image #3 displayed in Figure 3.16 is used to evaluate the effectiveness of the filter to attenuate exponential noise. The test image is the result of degrading Test Image #1 with ten percent exponential noise. The random digitizing noise was introduced at the individual pixel level of varying degrees of luminance values. A random generator of Park and Miller with Bays-Durham shuffle defined in [11] was used to ensure that the digitizing noise and its luminance values are completely randomly distributed. The point degradation in the Test Image #3 is barely visible, because the resolution of the color printer is such that each point degradation is equivalent to approximately 3.3 printer dots. Each pixel of the image is 0.01 inches wide. The result of applying the Laplacian operator to Test Image #3 is shown in Figure 3.17.

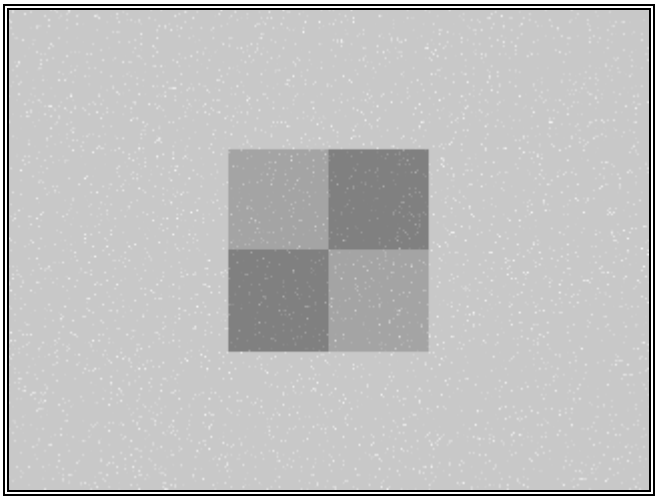


Figure 3.16 Test Image #3, 10 % Exponential Noise

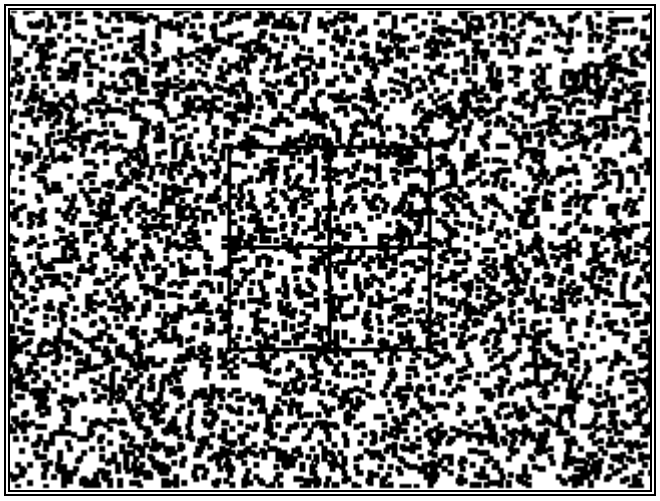


Figure 3.17 Test Image #3, Laplacian Operator

The result of applying the box filter and the Laplacian operator to the degraded test image is shown in Figure 3.18. It can be noted that the smoothing function of the filter reduced the intensity of the point degradation and affected adjacent pixels. In Figure 3.18 the Laplacian operator outlined the edges of the squares and the noise. In comparison to Figure 3.17 it is noticeable that after applying the filter to the image the outline of the noise has become several pixels wide. The box filter slightly blurred details in the image and is not effective in attenuating exponential noise. Applying additional iterations of the box filter or using a larger convolution mask will further attenuate the noise; however, blurring of image details would become significant.

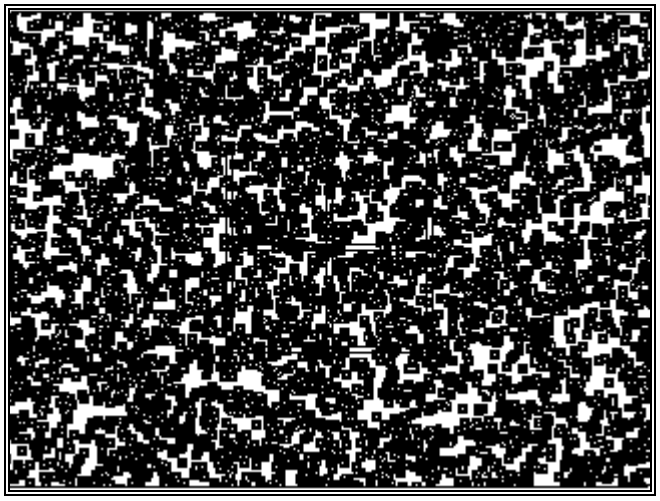


Figure 3.18 Test Image #3, Box Filter and Laplacian Operator

3.5 Binomial Filter

The binomial filter can serve as an alternative solution in attenuating Gaussian noise. The blurring effect is greatly reduced by exerting greater control over the neighborhood averaging using weight values other than one. This allows for the application of larger kernels, which also provides for the implementation of a more effective low pass filter. One artifact caused by using a larger kernel is called pseudo resolution [9]. Pseudo resolution is the result of averaging together two nearby structures in an image, which can result in an apparent feature between these two structures.

The convolution mask for the binomial filter contains values of the discrete binomial distribution shown in Table 3.2. The order of the binomial is denoted by n, f is the scaling factor, and σ^2 is the variance or the effective width of the mask. The convolution mask for a 3x3 and 5x5 kernel are derived in equation (3-4) and (3-5) from the binomial distribution shown in Table 3.2.

n	f	Convolution Mask	σ^2
0	1	1	0
1	1/2	1 1	1/4
2	1/4	1 2 1	1/2
3	1/8	1 3 3 1	3/4
4	1/16	1 4 6 4 1	1
5	1/32	1 5 10 10 5 1	5/4
6	1/64	1 6 15 20 15 6 1	3/2
7	1/128	1 7 21 35 35 21 7 1	7/4
8	1/256	1 8 28 56 70 56 28 8 1	2

Table 3.2 Binomial Distribution

$$B = \frac{1}{4}[1\ 2\ 1] * \frac{1}{4}\begin{bmatrix}1\\2\\1\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\2 & 4 & 2\\1 & 2 & 1\end{bmatrix} \qquad (3\text{-}4)$$

$$B = \frac{1}{16}\,[\,1\;4\;6\;4\;1\,] * \frac{1}{16}\begin{bmatrix}1\\4\\6\\4\\1\end{bmatrix} = \frac{1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\4 & 16 & 24 & 16 & 4\\6 & 24 & 36 & 24 & 6\\4 & 16 & 24 & 16 & 4\\1 & 4 & 6 & 4 & 1\end{bmatrix} \qquad (3\text{-}5)$$

The analysis of the binomial filter given in [10] shows that the filter is a good low pass filter. Figure 3.19 (A) and (B) show an arbitrary image section consisting of fixed vertical patterns. The result of applying the binomial filter to these image sections is displayed in (C) and (D). The fixed pattern displayed in (A) consist of a higher repetitive frequency than the pattern in (B). The filter was effective in smoothing the image. Figure 3.19 (D) shows the result of applying the binomial filter to (B). The filter was less effective due to the lower repetitive frequency of the fixed pattern. These results show considerable improvements in respect to the box filter.

15	30	15	30	15
15	30	15	30	15
15	30	15	30	15

A

15	30	15	15	30
15	30	15	15	30
15	30	15	15	30

B

23	23	23	23	23
23	23	23	23	23
23	23	23	23	23

C

19	23	19	19	23
19	23	19	19	23
19	23	19	19	23

D

Figure 3.19 Binomial Filter applied to Fixed Pattern

Figure 3.20 displays the result of applying the binomial filter with a 5x5 kernel and the Laplacian operator to the Test Image #3. It can be observed that the filter is effective in attenuating spatial noise, but slightly blurs the image details. The filter is not effective in attenuating exponential noise.

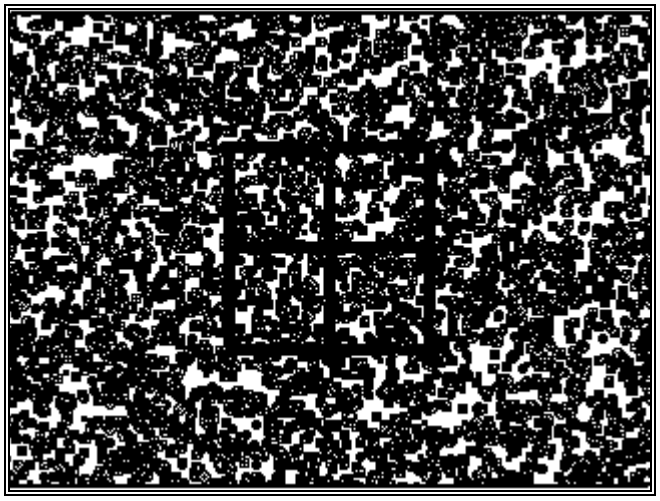


Figure 3.20 Test Image #3, Binomial Filter and Laplacian Operator

3.6 Median Filter

The median filter falls into the category of rank value filters. Rank value filters are non-linear filters and prove extremely effective in attenuating exponential noise. The operator sorts the values within the filter mask in ascending order, and replaces

the center pixel with the median of the sorted values. Figure 3.21 illustrates the implementation of the median filter. The arbitrary section of the image matrix consists of a uniform background with one noise spike of 200 digitizing counts present. The median filter replaces the noisy pixel with the medium value 128.

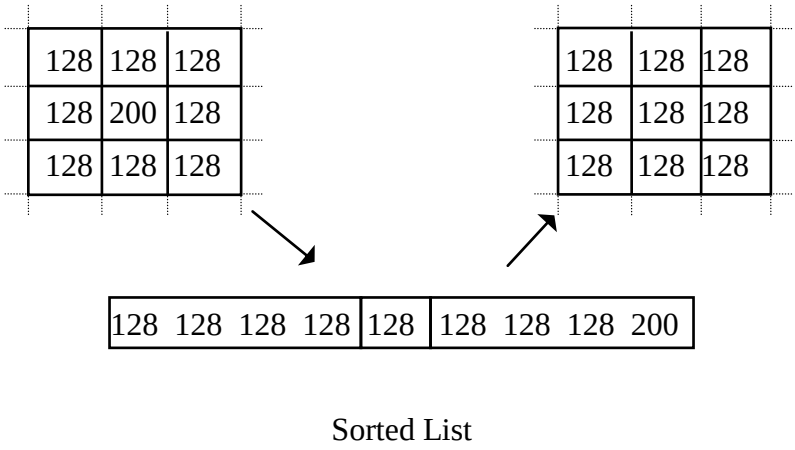


Figure 3.21 Median Filter

Figure 3.22 and Figure 3.23 show the results of applying the median filter to the Test Image #3. Blurring of image details is not noticeable as when it was realized using the linear shift invariant filters discussed in the previous sections. However; one image abnormality observed due to the median filter is the deformation of corners in image details.

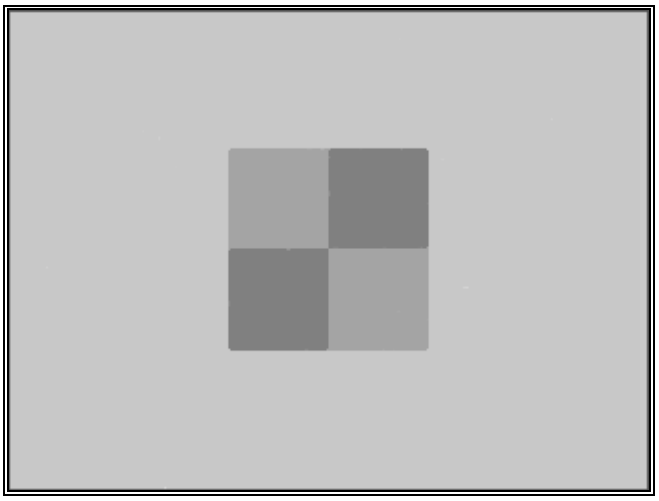


Figure 3.22 Test Image #3, Median Filter

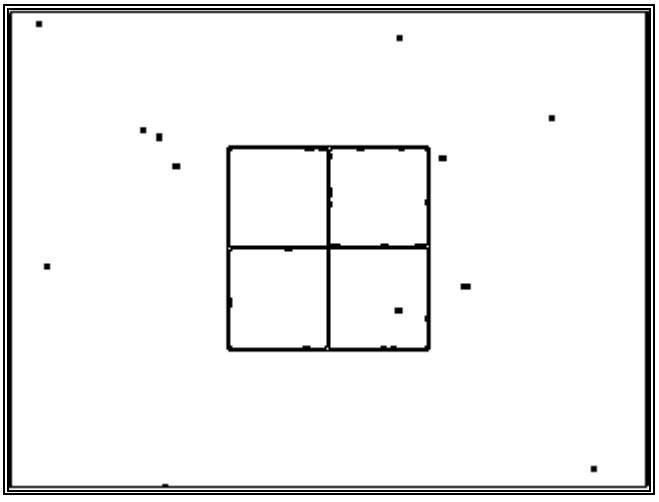


Figure 3.23 Test Image #3, Median Filter and Laplacian Operator

3.7 Uniformity Response

Figure 3.24 displays one frame of the gray scale 5.5. The bitmap shows that the imaging system used to capture the image has a greater response at the red and green wavelength, and the corners of the image is slightly darker than the center. The effectiveness of combining frame averaging and filtering using the median filter and the box filter is summarized in Table 3.3, Table 3.4, and Table 3.5.

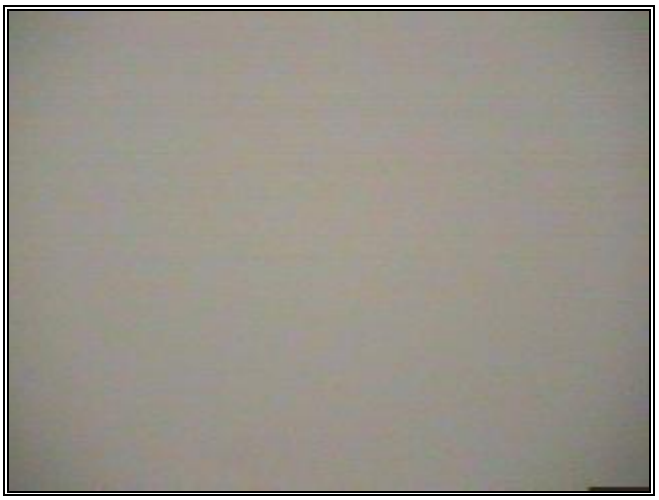


Figure 3.24 Gray Scale 5.5

Table 3.3 displays the data for averaging 1, 9, and 100 frames without gain correction or filtering. The standard deviation indicates that the pixel response deviates approximately eight counts. Table 3.4 shows the data for the gain corrected images. Averaging 100 frames and performing the gain adjustment for each pixel

produced a uniform image with zero digitizing counts of standard deviation. The image is completely void of temporal and spatial noise. Table 3.5 reflects the data for gain corrected and filtered images. Averaging nine image frames and applying pixel gain correction and filtering produced an acceptable uniform image, see Figure 3.25. The average pixel noise is approximately two counts. The advantage of this image processing technique is that good results are achieved using static or non-static images, and the limitations involved in obtaining one hundred frames is avoided.

Frame	Mean			Standard Deviation		
Averaging	Red	Green	Blue	Red	Green	Blue
1	153.02	148.00	138.97	8.94	7.85	8.13
9	152.46	147.63	138.54	8.42	7.58	7.18
100	155.50	150.35	141.05	8.60	7.80	7.28

Table 3.3 Original Image, Frame Averaging

Frame	Mean			Standard Deviation		
Averaging	Red	Green	Blue	Red	Green	Blue
1	156.98	157.00	159.30	5.76	5.10	6.41
9	155.79	156.20	158.88	3.19	2.85	2.72
100	162.00	162.00	162.00	0.04	0.00	0.00

Table 3.4 Gain Correction

Frame Averaging	Mean			Standard Deviation		
	Red	Green	Blue	Red	Green	Blue
1	156.60	156.66	158.98	4.33	3.30	4.84
9	155.31	155.75	158.46	2.26	1.85	2.02
100	162.00	162.00	162.00	0.16	0.45	0.25

Table 3.5 Gain Correction, Median Filter, And Box Filter

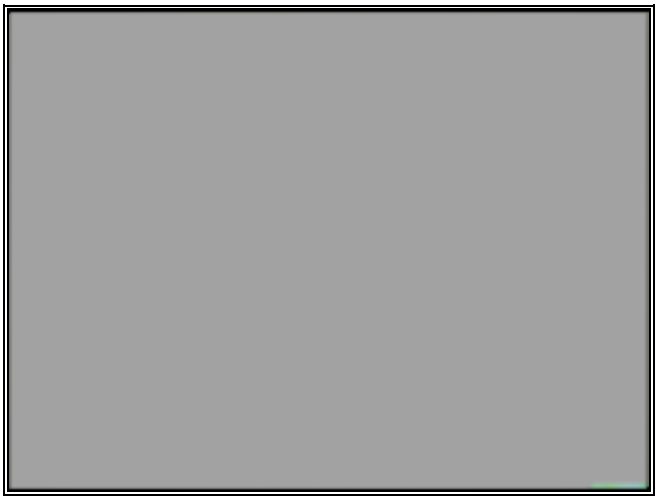


Figure 3.25 Gray Scale 5.5, Gain Adj., Median Filter, and Box Filter

Chapter 4

Fuzzy Expert System

Fuzzy expert systems are relative new systems in the field of Artificial Intelligence. The fuzzy systems are knowledge based systems, and their performance can be optimized by an adoptive learning process. Lotfi Zadeh first introduced the concept of Fuzzy Sets in 1965, but the concept of fuzziness and multivalued logic was not immediately accepted by the western scientific community [12]. The eastern countries adopted the fuzzy set theory and are very successful in applying the technology to every aspect of their lifes. Fuzzy systems prove highly efficient in linear and nonlinear control systems, data analysis, forensic science, pattern recognition, and image processing. One successful implementation of fuzzy technology in image processing is found in [4], where Yo Egusa, Hiroshi Akahori, Atsushi Morimura, and Noboru Wakami develop an electronic video camera with an image stabilizer which eliminates the instability caused by movement of the camera operator. Fuzzy set theory was applied to discriminate between movements of the operator and objects in the image.

Recent advances in computer systems and microprocessors provided for a new interest and advances in fuzzy technology. The implementation of fuzzy designs can be realized by embedding it directly into VLSI and ASIC chips. This method greatly improves the processing speed and provides for real time processing of live video. In this Thesis a fuzzy expert system is designed and evaluated to correct

digital images for noise, linearity, and non-uniformity. The images are captured in an AVI file, and then each frame is converted to a 320 x 240 bitmap. The fuzzy algorithm is applied to the bitmap to enhance the image. The ability of fuzzy logic to discriminate between noise and detail in images proves essential in solving complex image problems.

4.1 A Brief Explanation Of Fuzzy Logic

In 1965 Lotfi Zadeh, a professor for computer science at the University of California in Berkeley, published the first paper defining the theory of fuzzy sets [13]. Previously in 1937, Max Black introduced concepts of fuzzy sets by defining the membership of objects in terms of degree of usage. The multivalued logic was introduced as a result of pondering logical paradoxes. Zadeh realized that objects can partially belong to different subsets, and their membership may be expressed in degrees of truth. Fuzzy set theory defines the extent to which an object belongs to a subset. The algorithm for the enhancement of noisy and non-uniform images developed and evaluated in this thesis is based on the fuzzy set theory originated by Zadeh [12] and explained in detail by Kosko [14].

4.1.1 Fuzzy Sets

In this thesis capital letters denote sets, and the symbol $\in$ denote membership in a set. For bivalent logic the indicator function I_S of subset S of U is defined below in (4-1) .

$$I_S(x) = \begin{cases} 1 \text{ if } x \in S \\ 0 \text{ if } x \notin S \end{cases} \qquad (4\text{-}1)$$

The value 1 indicates that x is a member of subset S, and the value 0 indicates that x is not a member of subset S. U is an arbitrary set referred to the Universal Set or the Universe of Discourse. In classical set theory the membership function is expressed in (4-2).

$$S : U \rightarrow (0, 1) \qquad (4\text{-}2)$$

For multivalued logic the indicator function I_S extends to the membership function m_A and measures the degree to which the element x belongs to the subset S. The set U is the Universe of Discourse of all values between the interval 0 and 1.

$$S : U \rightarrow [0, 1] \qquad (4\text{-}3)$$

Discrete mathematics defines P(S) the power set of S and provides for binary and unary operations on the set P(S). The definition for the Union of Sets is given in (4-4).

Let A, B $\in$ P(S). Then

$$A \cup B = \{ x \mid x \in A \text{ or } x \in B\} \quad (4\text{-}4)$$

The definition for the Cartesian Product of Sets is giving in (4-5) .

Let $A, B \in P(S)$. Then

$$A \cap B = \{(x, y) \mid x \in A \text{ and } y \in B\} \quad (4\text{-}5)$$

The definition for the Complement of a Set is giving in (4-6) .

Let $A \in P(S)$. Then

$$A' = \{x \mid x \in S \text{ and } x \notin A\} \quad (4\text{-}6)$$

The three operations stated in (4-4), (4-5), and (4-6) can also be extended to fuzzy logic. The membership function of the Union of fuzzy sets A and B is defined in (4-7).

Let m_A and m_B be membership functions of fuzzy set A and B.

$$m_A \cup m_B = \max(m_A , m_B) \quad (4\text{-}7)$$

The membership function of the Intersection of fuzzy sets A and B is defined in (4-8).

Let m_A and m_B be membership functions of fuzzy set A and B.

$$m_A \cap m_B = \min(m_A , m_B) \quad (4\text{-}8)$$

The membership function of the Complement of fuzzy set A is defined in (4-9).

Let m_A be a membership function of fuzzy set A .

$m_A' = 1 - m_A$ (4-9)

The following set identities are common to classical set theory and fuzzy set theory.

Associate law

$(A \cap B) \cap C = A \cap (B \cap C)$

$(A \cup B) \cup C = A \cup (B \cup C)$

Commutative law

$A \cap B = B \cap A$

$A \cup B = B \cup A$

Distributive law

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

De Morgans law

$(A \cap B)' = A' \cap B'$

$(A \cup B)' = A' \cup B'$

4.1.2 Fuzzy Expert System

The design of Fuzzy Expert Systems does not require that the engineer mathematically expresses how output variables depend on input variables. Fuzzy controllers generate solutions from approximate information instead of bivalent propositional associations. Experts provide and define the knowledge in human terms for the system. This is a great advantage, because mathematical models are often difficult to define and it severely limits the application of control systems. Fuzzy systems are model free estimators and consist of collections of rules and associations (A_i ,B_i). Human reasoning can be programmed into the system using a collection of fuzzy membership functions and rules. The rules are expressed in the form of **IF** condition true **THEN** response.

In the fuzzy rule "**IF** x = small positive **AND** y = large positive **THEN** z = medium positive ", the small positive, large positive, and medium positive are membership functions defined on the variables x, y, and z respectively. Small, large, and medium are referred to as hedges, and provide for the fuzziness of the expert system. Hedges define the degree of membership in human terms. The input variables applied to the fuzzy logic are not limited to discrete values, instead the variables can assume any value in the interval of zero and one. The collection of rules is referred to the knowledge base or rule base of the system. Fuzzy systems S map input fuzzy sets I^n to output fuzzy sets I^P and is characterized as associative memory.

$$S : I^n \rightarrow I^P \qquad (4\text{-}10)$$

In Fuzzy Systems the Fuzzy Associative Memory (FAM) rules are processed in parallel to produce the output fuzzy set B. Figure 4.1 shows a FAM matrix for a small fuzzy controller. The controller provides for two input variables x and y, and determines the degree of membership in five possible input fuzzy sets. Each input variable can hold partial or full membership in large negative (LN), small negative (SN), zero (ZE), small positive (SP), and large positive (LP). The controller output B defined in (4-11) also comprises of five possible fuzzy sets, which is medium positive (MP), small positive (SP), zero (ZE), small negative (SN), and medium negative (MN). Variable w_m in (4-11) is the weight of the rules, and expresses the degree of membership in the output fuzzy sets. Output B of the fuzzy controller is the result of all activated output fuzzy sets.

$$B = w_1B_1 + \ldots + w_mB_m \qquad (4\text{-}11)$$

		x				
		LN	SN	ZE	SP	LP
y	LP	MP	MP	MP	MP	MP
	SP	MN	SP	SP	SP	MP
	ZE	MN	SN	ZE	SP	MP
	SN	MN	SN	SN	SP	MP
	LN	MN	MN	MN	MN	MP

Figure 4.1 FAM Matrix

Equation (4-10) shows that each entry into the FAM matrix maps to at least one fuzzy set output w_iB_i. For example, the input variable x may have full membership in input fuzzy set ZE, and the input variable y may have partial membership in input fuzzy sets SN and LN. The output of the fuzzy controller would comprise of the output fuzzy sets SN and MN. For a time variant system the input entry x = 0, and y = 0 produces the final steady state output ZE = 0.

The output fuzzy set B defined in (4-11) requires defuzzification to provide for one crisp output value. Calculating the fuzzy centroid of the output is one of many defuzzification methods available. The process of defuzzification may be combined directly with the composition process. Composition is the process of generating the output fuzzy set B.

4.1.3 The Fuzzy Centroid

In this thesis the technique used to calculate the fuzzy centroid is obtained from Kosko in [14]. Calculating the fuzzy centroid is relative simple to implement in software, but forces several constraints on the inference machine. The discrete fuzzy centroid $\bar{B}$ can be calculated only if the correlation-product operation is used for the inference process. The additional requirement is that output fuzzy sets are unimodal and symmetric. The fuzzy centroid defined in (4-12) creates the single numerical output y_j in the universe of discourse $Y = \{ y_1, ..., y_p \}$.

$$\bar{B} = \frac{\sum_{j=1}^{p} y_j \, m_B(y_j)}{\sum_{j=1}^{p} m_B(y_j)} \qquad (4\text{-}12)$$

Figure 4.2 displays a FAM system mapping the input fuzzy set A to the output fuzzy set B. The defuzzification process generates element y of the output set B. It should be noted that fuzzy systems process the FAM rules in parallel and sums the outputs.

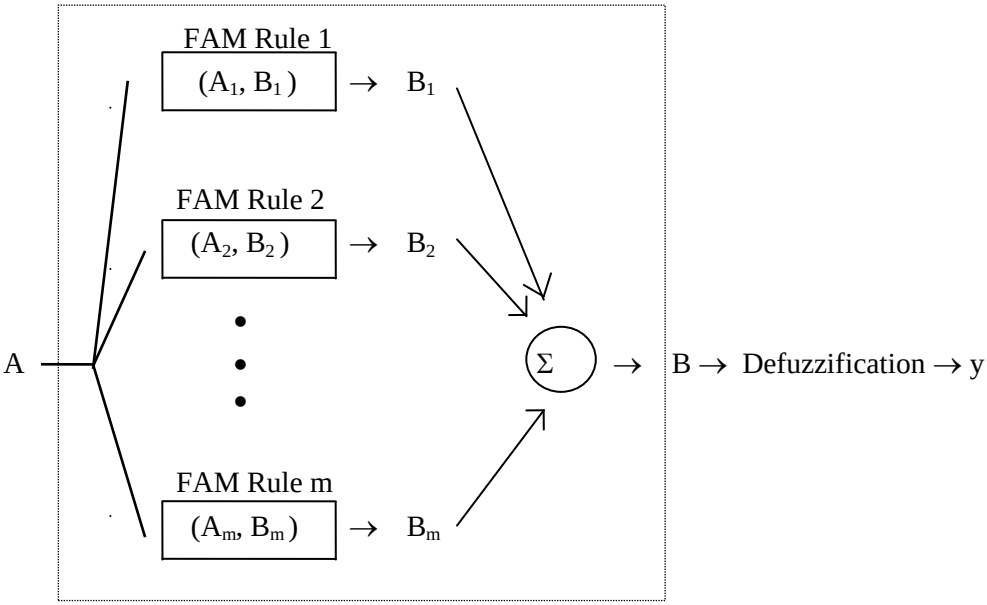


Figure 4.2 FAM System

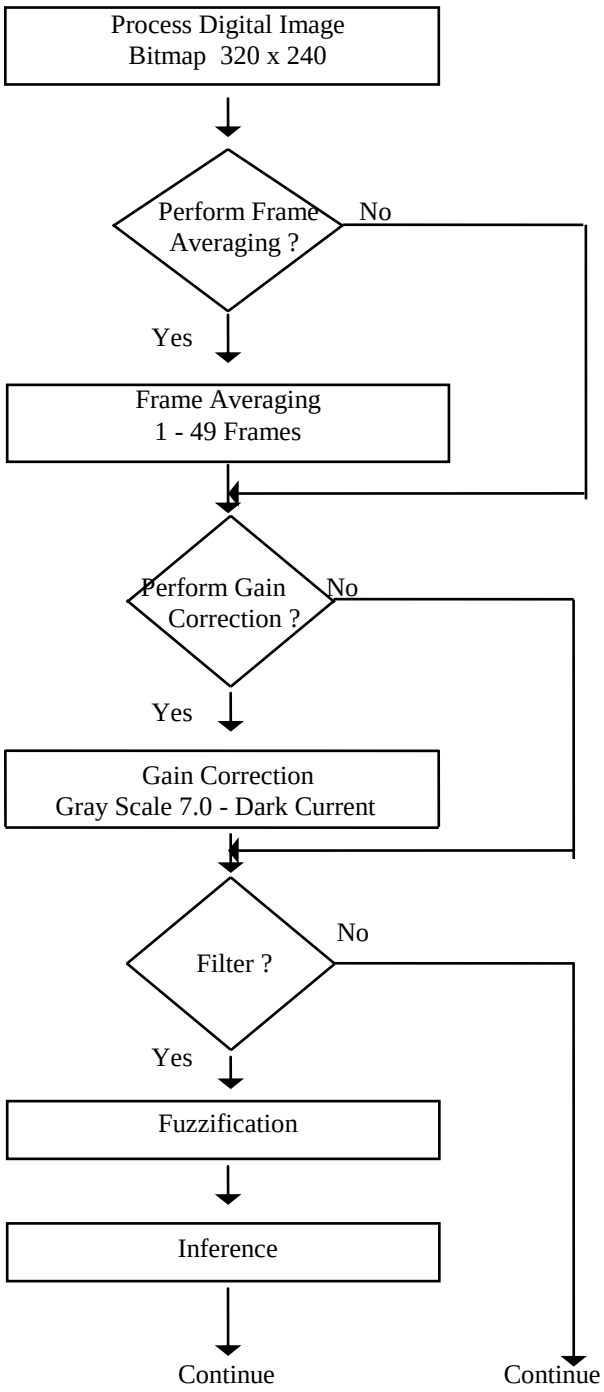
4.2 The Proposed Fuzzy Algorithm

Improving the quality and clarity of an image requires increasing the degree of visibility of relevant information. Image processing may be accomplished through frame averaging, enhancing the dynamic range or contrast, correcting image uniformity and pixel linearity, and filtering noise. The effectiveness of filtering noise using classical methods depends on a priori knowledge of the signal and the type of noise present. In [3] Bing Zeng presents a method for the design of median type filters with an optimum noise attenuation. Zeng evaluates the design of median type filters with structural constraints imposed to preserve specific signal and image features. The preservation of relevant information in an image and the reduction of noise are two opposing criteria when processing the digital image. The classical filtering methods may require the application of different types of filters to reduce noise, and the sequence of applying the filters is critical. Median filters are optimal in filtering double exponential noise and impulse noise, but fail in filtering Gaussian noise. Mean filters are better suited for filtering Gaussian noise, but fail filtering exponential noise. The results in [3] show that the standard median filter has superior noise suppression in respect to weighted median filters.

The design of the proposed fuzzy algorithm incorporates the capabilities of the mean and median filter. The strength of the fuzzy filtering is continuously adjusted to adopt to the different features present in the image. The inference process uses a set of heuristic rules to determine the type filtering necessary, and detects the presence of edges and corners. This feature provides for an effective noise reduction and an optimal preservation of image details. Blurring of edges and altering corners is avoided using the fuzzy logic. Prior knowledge of the signal and noise present in the image is not required, since the inference process controls the type of filtering applied to each pixel. In addition to filtering noise each pixel is

corrected for gain, offset, and linearity. The response to each color red, green, and blue is balanced to achieve a true gray scale.

Figure 4.3 shows the flow diagram for the fuzzy algorithm. The video is captured to an AVI file, and the individual frames are converted to a 320 x 240 bitmap. At the beginning of the digital image process the user can set appropriate flags in the software to control which type of correction is applied to each image. The selections include frame averaging, gain correction, filtering, and the correction of pixel linearity. Frame averaging can be in the range of 1 and 49 frames. The gain and offset correction is performed for each pixel using two distinct data points. One point is derived from a filtered Gray Scale 7.0, and the other point is the dark current of the image system. The user can also select specific video rows to which filtering is applied. The fuzzy filtering process consist of fuzzification, inference, composition, and defuzzification. The defuzzification stage includes correction for pixel linearity. The user can deselect this feature. The corrected image can be stored to a user defined bitmap, and the RGB values of the pixels is stored to a data file compatible with the EXCEL spreadsheet.


Process Digital Image
Bitmap 320 x 240
Perform Frame Averaging ?
No
Yes
Frame Averaging
1 - 49 Frames
Perform Gain Correction ?
No
Yes
Gain Correction
Gray Scale 7.0 - Dark Current
Filter ?
No
Yes
Fuzzification
Inference
Continue
Continue

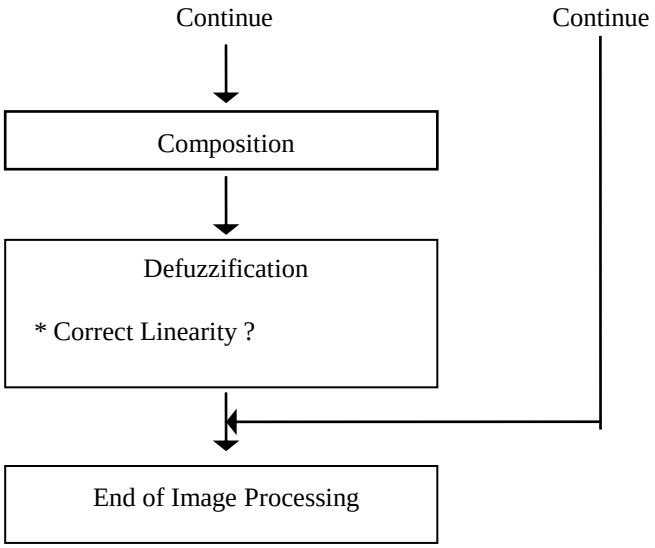


Figure 4.3 Flow Diagram of Fuzzy Algorithm

4.2.1 Previous Research

Applying fuzzy technology to digital image processing is a relative new approach to solving noise related problems. Ching-Yu Tyan and Paul P.Wang in [15] developed a fuzzy filtering technique capable of providing an improved restoration of an image degraded by additive random noise. The inference engine of the fuzzy filter applies a set of fuzzy rules to each pixel and corrects the luminance values of the noisy pixels. The signal to noise ratio is defined in (4-13), and the model for the degraded image is given in (4-14). The fuzzy variable f (p_1 , p_2) represents the intensity values of the pixels, and n (n_1 , n_2) represents the additive random noise. Var[.] is the variance.

$$\text{SNR (db)} = 10 \log * \frac{\text{Var}\,[\, f(p_1 , p_2)\,]}{\text{Var}\,[\, n(n_1 , n_2)\,]} \qquad (4\text{-}13)$$

$$f_{pn}(pn_1 , pn_2) = f(p_1 , p_2) + n(n_1 , n_2) \qquad (4\text{-}14)$$

The result of the research in [15] demonstrates that fuzzy filters can provide for greater improvements in the signal to noise ratio than the traditional low pass filters. The process of removing noise from a digital image involves filtering high frequency components. This presents a problem in enhancing the noisy image, because the different details in the image comprises of many low and high frequency components. Conventional low pass filters don't discriminate between high frequency components of the relevant video signal and the noise component. The fuzzy algorithm must protect the high frequency components of the video signal to prevent blurring the image by altering edges and corners.

Fabrizio Russo and Giovanni Ramponi in [5] developed a new class of fuzzy operator for the enhancement of blurred and noisy images. The fuzzy operator is designed to sharpen details in images and proved insensitive to noise. The results obtained from the study show that the edge enhancements performed on the noise degraded image compares similar to the results of using a conventional filter. Sharpening of the edges was successfully performed without amplifying the noise signal. Fabrizio Russo extends the research to the smoothing of images with the preservation of edges. A detailed explanation of the design and implementation of the fuzzy operator is provided in [16]. The fuzzy controller processes rules and membership functions directly defined on pixel P at location (p_1, p_2). Variables that

provide information to the fuzzy operator are local luminance differences v_i found between the center pixel P_c and the surrounding pixel P_i . The knowledge base defines a set of rules applicable to two different size windows, 3x3 and 5x5, and centers each on pixel P_c. The window slides over the image matrix, and the inference engine processes each pixel non-recursively. Variable qB in equation (4-16) is the luminance value required to correct P_c . The scaling factor q is optional.

$$v_i = P_i - P_c \qquad (4\text{-}15)$$

$$P'_c = P_c + qB \qquad (4\text{-}16)$$

In this Thesis the filtering technique used to attenuate temporal and spatial noise is based on this new class of fuzzy operators developed by Fabrizio Russo and Giovanni Ramponi. The fuzzy operator and the rules are modified to optimize the overall performance and provide noise attenuation by incorporating characteristics of the mean and median filter.

4.2.2 Gain Correction

The gain correction algorithm performs the function of adjusting the gain for each individual pixel, and compensates the image for non-uniformity. Each pixel response is balanced for the colors red, green, and blue using interpolation and two lookup tables. The first lookup table holds values for the dark currents, and the second lookup table comprises of the pixel responses at gray level 7.0. Figure 4.4

shows the dark currents at the bottom of the figure and the pixel responses for scan line10 at the top. Each scan line consist of 320 pixels. It is noticeable that the outside perimeter of the image is darker than the center, and the response decreases from the color red to blue. All pixel luminance values are adjusted to RGB(190, 190, 190) at gray scale 7.0 .

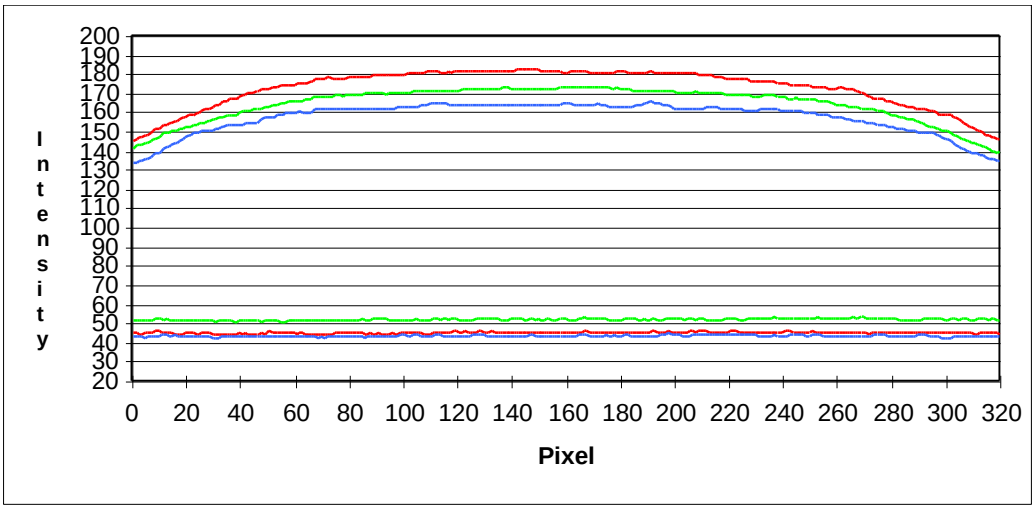


Figure 4.4 Luminance Values For A Single Scan Line

The gain correction is tested with Test Image #4 shown in Figure 4.5. The 320 x 240 bitmap is the average of nine frames. The average signal-to-noise-ratio for the colors red, green, and blue are 42.68 dB, 40.10 dB, and 41.27 dB respectively. Figure 4.6 shows the corrected image. The average signal-to-noise-ratio improved by 0.37 dB, 0.46 dB, and 0.29 dB for the colors red, green, and blue respectively.

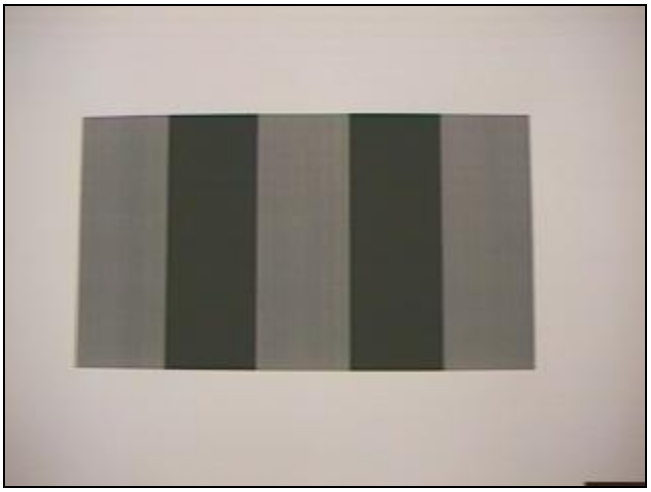


Figure 4.5 Test Image #4

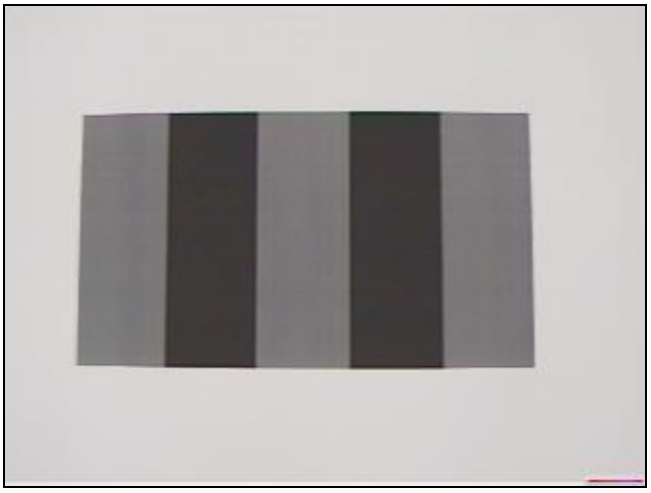


Figure 4.6 Test Image #4, Gain Corrected

4.2.3 Fuzzification Process

During the fuzzification process membership functions (4-17), (4-18), and (4-19) defined on the input variables are applied to their actual values. The premise of each rule is determined; and, if the premise is not zero, then the rule is activated and fires. The process starts by superimposing a 3x3 window on the image matrix and determining the luminance differences between the center pixel P_c and the surrounding pixels P_i as defined in equation (4-15). The luminance differences are classified into four categories, large negative (LN), small negative (SN), small positive (SP), and large positive (LP). Two groups, SN and SP, are combined to form the category small difference (SD). The fuzzification process is performed on each pixel for the colors red, green, and blue. Figure 4.7 shows the four categories and their membership relationships. The three groups are symmetric and trapezoidal in shape.

$$LN(P) = \begin{cases} +1.0, \text{ if } P \leq -10.0 \\ \\ \dfrac{P + 6.0}{4.0}, \text{ if } -10.0 < P < -6.0 \end{cases} \qquad (4\text{-}17)$$

$$SD(P) = \begin{cases} \dfrac{-10.0 - P}{4.0}, & \text{if } -10.0 < P < -6.0 \\ +1.0, & \text{if } -6.0 \leq P < 0.0 \\ +1.0, & \text{if } 0.0 \leq P \leq 6.0 \\ \dfrac{10.0 - P}{4.0}, & \text{if } 6.0 < P \leq 10.0 \end{cases} \qquad (4\text{-}18)$$

$$LP(P) = \begin{cases} \dfrac{P - 6.0}{4.0}, & \text{if } 6.0 < P < 10.0 \\ +1.0, & \text{if } P \geq +10.0 \end{cases} \qquad (4\text{-}19)$$

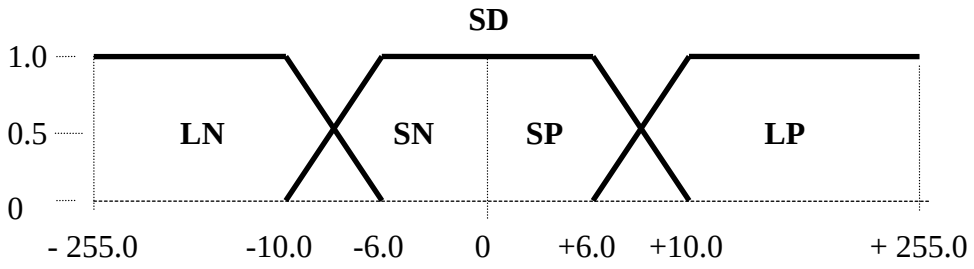


Figure 4.7 Fuzzification Process

In the design of the filter the luminance differences are not normalized into the interval [0,1], instead they can assume any value between - 255 and + 255

digitizing counts. The extend to which a luminance difference belongs to either LN, SN, SP, or LP is defined in the interval [0,1], and expresses the degree of membership. When the luminance differences between the center pixel and it's surrounding pixels are less than -10.0 or greater than + 10.0, the degree of membership in LN and LP is 1.0. A membership of 1.0 in LN or LP indicates that the center pixel was altered by exponential noise and requires filtering with the median filter. Luminance differences in the range from -6.0 to +6.0 counts indicate that the center pixel is most likely corrupted by Gaussian noise and requires correction using the mean filter. The range between - 6.0 and - 10.0 indicates a partial membership in both LN and SN, and the range between +6.0 and +10.0 indicates partial membership in SP and LP. A partial membership in either SN and LN or SP and LP implies that both mean and median filters will be applied. The strength of each filter is determined by the degree of membership in LN, SD, and LP.

The strength of the mean and median filter can be adjusted by changing the number of digitizing counts that defines the membership in LN(P), SD(P), and LP(P). Adjusting the overlap and the slope of the three groups provides additional control in the design. If the slopes are not 45 degrees then the degree of membership may sum to values other than 1.0. In Figure 4.8 the membership at -25.0 counts is 0.75 LN and 0.75 SD. This may result in a nonlinear correction of the corrupted pixel luminance values.

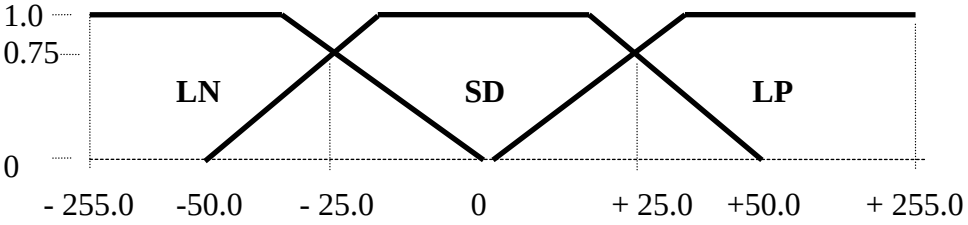


Figure 4.8 The Slope And Range Provides Filter Control.

4.2.4 Inference Process

In the fuzzification process the degree of membership in LN, SD, and LP was determined for eight luminance differences for each pixel in the image matrix. The degrees of membership are expressed in the interval [0,1], and they can be mapped into a FAM cube as illustrated in figure 4.9. For each group LN, SD, and LP there are eight cells forming three rows and three columns. Cell P represents the pixel that is processed and receives no value. This cell is not included to form rows and columns. Row 0 consist of cell 0, 1, and 2, Row 1 consist of cell 3 and 4, and Row 2 consist of cell 5, 6, and 7. Column 0 consist of cell 0, 3, and 5, column 1 consist of cell 1 and 6, and column 2 consist of cell 2, 4, and 7.

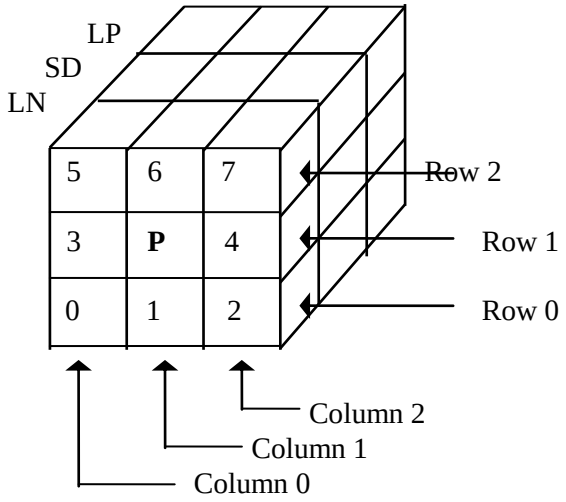
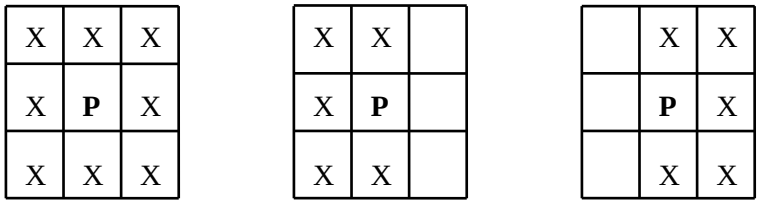
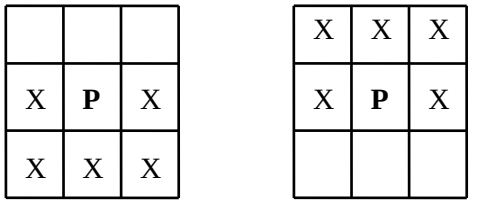


Figure 4.9 FAM Cube

In the inference process fuzzy rules from the rule base are applied to the FAM cube to determine the fuzzy subsets $w_m B_m$ defined in (4-11). The fuzzy knowledge base comprises of 15 rules, and they are stated below in the rule base. Figure 4.10 graphically displays the conditions that have to be met in order for a rule to fire. An X indicates membership due to the result of the luminance difference between pixel P_c and the surrounding pixel P_j . The fuzzy rules are designed such that any edge in the 3x3 window is detected and protected from filtering. The rows and columns must form an orthogonal pattern within each group LN, SD, and LP in order for the rules to activate. If the partial membership in LN, SD, and LP are distributed into the cube such that none of the fifteen rules apply, then the luminance value of pixel P is not adjusted. Such a distribution implies that the local image detail consist of high frequency components and the noise component cannot be isolated and removed without loosing image detail.



Rule 1, 6, 11 Rule 2, 7, 12 Rule 3, 8, 13



Rule 4, 9, 14 Rule 5, 10, 15

Figure 4.10 Fuzzy Rules

Rule Base :

Rule 1 - If the degrees of membership map into the eight cells of LN, then w_i assumes the maximum value of cell 0 through 7. B is assigned the median value of the pixel luminance differences. $w_m B_m$ reduces the luminance value of P.

Rule 2 - If the degrees of membership map into LN cells 0, 1, 3, 5, and 6, then w_i assumes the maximum value of cell 0, 1, 3, 5, and 6. B is assigned the median value of the pixel luminance differences associated with cell 0, 1, 3, 5, and 6. $w_m B_m$ reduces the luminance value of P.

Rule 3 - If the degrees of membership map into LN cells 1, 2, 4, 6, and 7, then w_i assumes the maximum value of cell 1, 2, 4, 6, and 7. B is assigned the median value of the pixel luminance differences associated with these cells. $w_m B_m$ reduces the luminance value of P.

Rule 4 - If the degrees of membership map into LN cells 0, 1, 2, 3, and 4, then w_i assumes the maximum value of cell 0, 1, 2, 3, and 4. B is assigned the median value of the pixel luminance differences associated with these cells. $w_m B_m$ reduces the luminance value of P.

Rule 5 - If the degrees of membership map into LN cells 3, 4, 5, 6, and 7, then w_i assumes the maximum value of cell 3, 4, 5, 6, and 7. B is assigned the median value of the pixel luminance differences associated with these cells. $w_m B_m$ reduces the luminance value of P.

Rule 6 - If the degrees of membership map into the eight cells of SD, then w_i assumes the maximum value of cell 0 through 7. B is assigned the mean value of

the pixel luminance differences. $w_m B_m$ either increases or decreases the luminance value of P.

Rule 7 - If the degrees of membership map into SD cells 0, 1, 3, 5, and 6, then w_i assumes the maximum value of cell 0, 1, 3, 5, and 6. B is assigned the mean value of the pixel luminance differences associated with cell 0, 1, 3, 5, and 6. $w_m B_m$ either increases or decreases the luminance value of P.

Rule 8 - If the degrees of membership map into SD cells 1, 2, 4, 6, and 7, then w_i assumes the maximum value of cell 1, 2, 4, 6, and 7. B is assigned the mean value of the pixel luminance differences associated with these cells. $w_m B_m$ either increases or decreases the luminance value of P.

Rule 9 - If the degrees of membership map into SD cells 0, 1, 2, 3, and 4, then w_i assumes the maximum value of cell 0, 1, 2, 3, and 4. B is assigned the mean value of the pixel luminance differences associated with these cells. $w_m B_m$ either increases or decreases the luminance value of P.

Rule 10 - If the degrees of membership map into SD cells 3, 4, 5, 6, and 7, then w_i assumes the maximum value of cell 3, 4, 5, 6, and 7. B is assigned the mean value of the pixel luminance differences associated with these cells. $w_m B_m$ either increases or decreases the luminance value of P.

Rule 11 - If the degrees of membership map into the eight cells of LP, then w_i assumes the maximum value of cell 0 through 7. B is assigned the median value of the pixel luminance differences. $w_m B_m$ increases the luminance value of P.

Rule 12 - If the degrees of membership map into LP cells 0, 1, 3, 5, and 6, then w_i assumes the maximum value of cell 0, 1, 3, 5, and 6. B is assigned the median value

of the pixel luminance differences associated with cell 0, 1, 3, 5, and 6. $w_m B_m$ increases the luminance value of P.

Rule 13 - If the degrees of membership map into LP cells 1, 2, 4, 6, and 7, then w_i assumes the maximum value of cell 1, 2, 4, 6, and 7. B is assigned the median value of the pixel luminance differences associated with these cells. $w_m B_m$ increases the luminance value of P.

Rule 14 - If the degrees of membership map into LP cells 0, 1, 2, 3, and 4, then w_i assumes the maximum value of cell 0, 1, 2, 3, and 4. B is assigned the median value of the pixel luminance differences associated with these cells. $w_m B_m$ increases the luminance value of P.

Rule 15 - If the degrees of membership map into LP cells 3, 4, 5, 6, and 7, then w_i assumes the maximum value of cell 3, 4, 5, 6, and 7. B is assigned the median value of the pixel luminance differences associated with these cells. $w_m B_m$ increases the luminance value of P.

4.2.5 Composition And Defuzzification

In the composition process the subsets $w_m B_m$ are combined to create the output fuzzy set B. This process is illustrated in Figure 4.2. Once fuzzy set B is defined the defuzzification process determines one crisp value by calculating the fuzzy centroid. The fuzzy centroid is defined in (4-12) and may be simplified as in (4-20) below. The final value y is applied to pixel P_c to obtain the corrected pixel P defined in (4-21).

$$y = \frac{\sum_{m=1}^{15} w_m B_m}{\sum_{m=1}^{15} w_m} \qquad (4\text{-}20)$$

$$P = P_c - y \qquad (4\text{-}21)$$

The defuzzification process includes the option to correct each pixel for a linear response for each color red, green, and blue. The user can select / deselect this option in the software. The gray scale is divided into seven ranges to provide for a suitable adjustment. Table 4.1 shows the gray scales, reflectance, and response in digitizing counts. The pixel response is adjusted to approximately 20.1 counts per gray scale.

Gray Scales	Reflectance (%)	RGB Values (counts)	Adjusted RGB Values (counts)
0.0 - 2.0	0 - 7	45 - 91	45 - 94
2.0 - 3.0	7 - 13	91 - 114	94 - 118
3.0 - 4.0	13 - 18	114 - 149	118 - 142
4.0 - 5.0	18 - 27	149 - 160	142 - 166
5.0 - 6.0	27 - 36	160 - 179	166 - 191
6.0 - 7.0	36 - 45	179 - 190	191 - 215
7.0 -10.0	45 - 90	190 - 255	215 - 255

Table 4.1 Gray Scales and Linearity

4.2.6 Filter Analysis

Figure 4.11 displays the luminance values of a section in an arbitrary image matrix before and after the fuzzy filter is applied to the center pixel. The values were chosen to present local image deviations without edges or exponential noise present. The luminance differences between the surrounding pixels and the center pixel are mapped into **A**. Applying the differences to the membership functions graphically shown in Figure 4.7 cell 0, 3, and 5 indicate membership in LN, displayed in **B,** and cell 1, 2, 4, 6 and 7 indicate membership in SD shown in **C**. There is no membership in LP, and it is indicated by the zero entries in **D**. During the inference process rule eight is activated to generate w_8B_8 , where w_8 is equal to 1, and B_8 is equal to the mean of cell 1, 2, 4, 6, and 7. The centroid defined in (4-20) and the new pixel luminance value is calculated below.

90	102	106
87	100	105
88	102	102

Original matrix

90	102	106
87	103	105
88	102	102

Filter applied to P_c

-10	2	6
-13	**P**	5
-12	2	2

A

1	0	0
1	**LN**	0
1	0	0

B

0	1	1
0	**SD**	1
0	1	1

C

0	0	0
0	**LP**	0
0	0	0

D

Figure 4.11 Arbitrary Image Matrix #1

$$y = \frac{w_8 B_8}{w_8}$$

$$y = \frac{1 * 3.4}{1}$$

$$P = 100 + 3$$

Figure 4.12 displays the luminance values of an arbitrary image matrix where all of the pixel differences hold membership in SD. In this section of the image matrix exponential noise and edges are not present. During the inference process rules 6, 7, 8, 9, and 10 are activated to provide filtering. The centroid for the pixel correction and the new pixel value is calculated below.

102	102	106
99	100	105
100	102	102

Original matrix

102	102	106
99	102	105
100	102	102

Filter applied to P_c

2	2	6
-1	**P**	5
0	2	2

A

0	0	0
0	**LN**	0
0	0	0

B

1	1	1
1	**SD**	1
1	1	1

C

0	0	0
0	**LP**	0
0	0	0

D

Figure 4.12 Arbitrary Image Matrix #2

$$y = \frac{(1 * 2.25) + (1 * 1.40) + (1 * 2.80) + (1 * 1.00) + (1 * 3.40)}{1 + 1 + 1 + 1 + 1}$$

$y = 2.17$

$P = 100 + 2.17$

$P = 102$

Figure 4.13 displays the luminance values of a section of an arbitrary image matrix. All of the differences hold membership in LN, and the rules 1, 2, 3, 4, and 5 are activated during the inference process. This indicates that only the median filtering method is used to correct P. The centroid for the luminance correction and the new pixel value is calculated below.

$$y = \frac{(1 * -33) + (1 * -33) + (1 * -33) + (1 * -33) + (1 * -33)}{1 + 1 + 1 + 1 + 1}$$

$y = -33$

$P = 135 - 33$

$P = 102$

102	102	106
99	135	105
100	102	102

Original matrix

102	102	106
99	102	105
100	102	102

Filter applied to P_c

-33	-33	-29
-36	**P**	-30
-35	-33	-33

A

1 1	1	0
1	**LN**	1
1	1	1

B

0	0	0
0	**SD**	0
0	0	0

C

0	0	0
0	**LP**	0
0	0	0

D

Figure 4.13 Arbitrary Image Matrix #3

Figure 4.14 displays the luminance values of a section of an arbitrary image matrix where local pixel differences hold partial membership in SD and LP. During the inference process the rules 6, 7, 8, 9, 10, and 12 are activated. Mean filtering and median filtering are applied to correct the luminance value of P. The centroid for the luminance correction and the new pixel value is calculated below.

$$y = \frac{(1 * 6.75) + (0.75 * 8.2) + (1 * 6.0) + (1 * 6.6) + (1 * 6.8) + (0.75 * 8.0)}{1.0 + 0.75 + 1.0 + 1.0 + 1.0 + 0.75}$$

$y = 6.96$

$P = 100 + 6.96$

$P = 106$

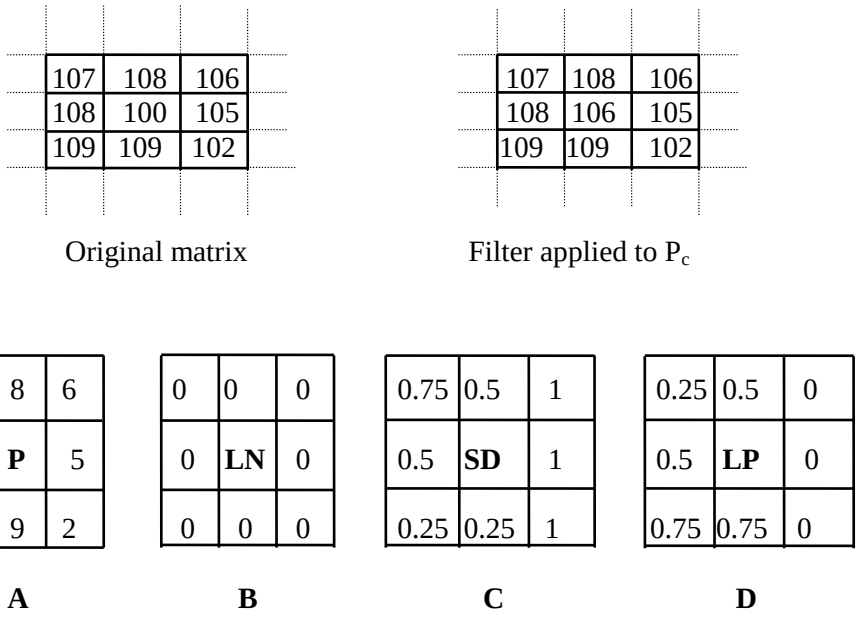


Figure 4.14 Arbitrary Image Matrix #4

4.3 Fuzzy Filter

Figure 4.15 displays the result of applying the fuzzy filter to the Test Image #1 displayed in Figure 3.1. The computer generated test image is a perfect image without noise and defects. The software options for frame averaging, gain adjustment, and linearity correction was disabled to determine if the fuzzy filter creates any image abnormalities. Figure 4.16 illustrates the result of applying the Laplacian operator shown in Figure 3.9 to Figure 4.15. The result shows that the edges and corners of the four squares are fully preserved. The fuzzy filter did not

alter any of the image details as it was observed using the classical mean and median filter.

Figure 4.17 displays the result of applying the fuzzy filter to the exponential noise corrupted Test Image #3 (see Figure 3.16). Figure 4.18 shows the effects of applying the Laplacian operator to the image that was corrected with the fuzzy filter. The fuzzy filter is not quite as effective as the median filter, but shows a significant reduction of exponential noise. All relevant image information is preserved.

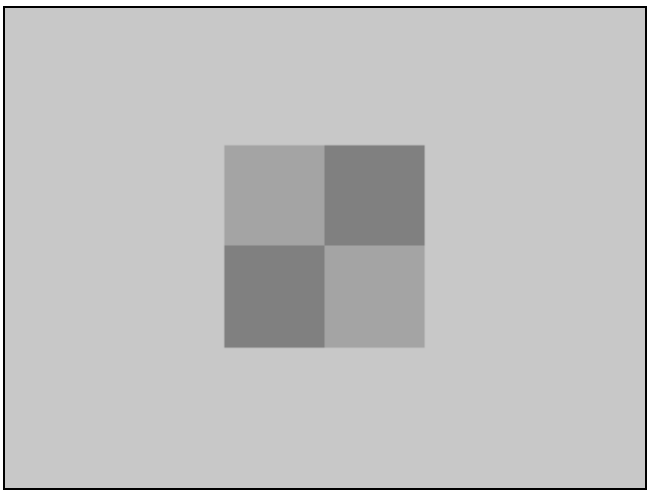


Figure 4.15 Test Image #1, Fuzzy Filter

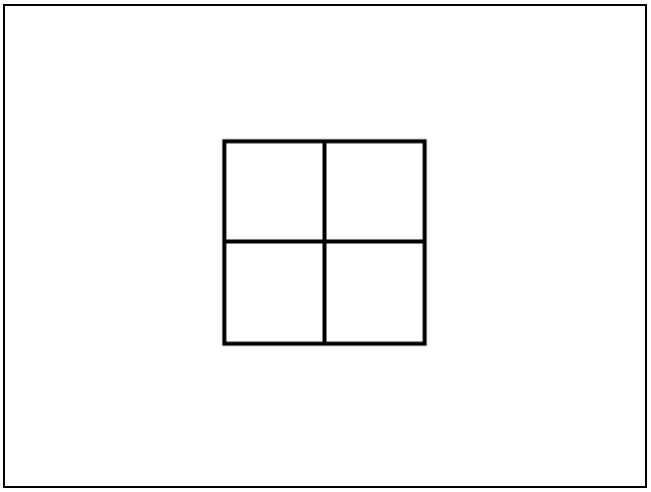


Figure 4.16 Edge Detection Algorithm Applied To Figure 4.15

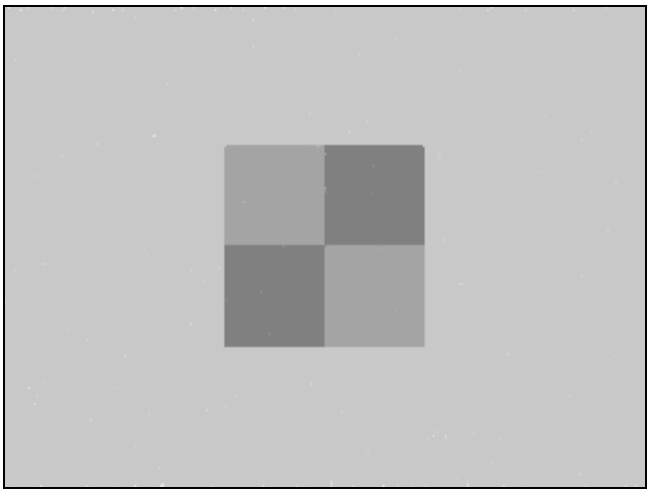


Figure 4.17 Test Image #3, Fuzzy Filter

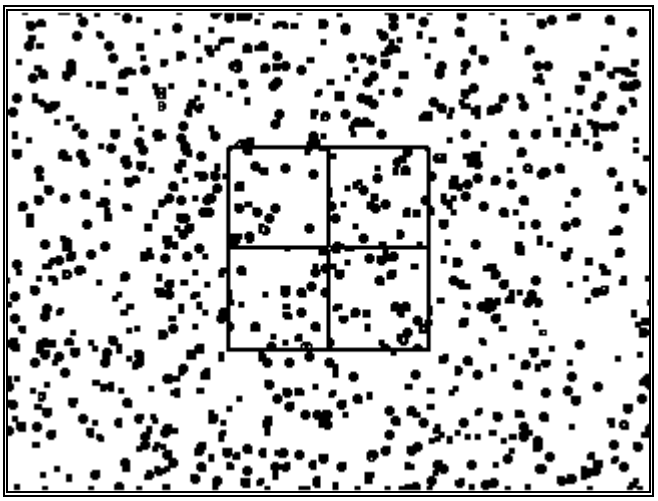


Figure 4.18 Edge Detection Algorithm Applied To Figure 4.17

Figure 4.19 displays a computer simulated image containing objects of low contrast. The simulated image differs from the computer generated image, since all the noise components and non-uniformity are present in the image. The result of applying the Laplacian operator to the image is also shown in Figure 4.20. Here the edges of the four bars are hidden by the non-uniformity patterns. The patterns are denser at the outside perimeter of the image, which implies a greater gradient in pixel luminance values. Figure 4.21 reflects the result of applying the fuzzy algorithm to Figure 4.19. The frame averaging was disabled because the computer simulated image consist of one frame only. Figure 4.22 shows that the corrected image consists of a greater uniform background than the original.

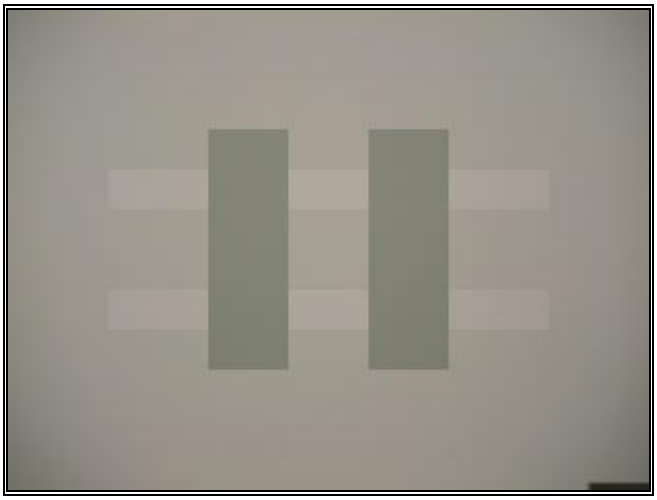


Figure 4.19 Computer Simulated Image, Low Contrast

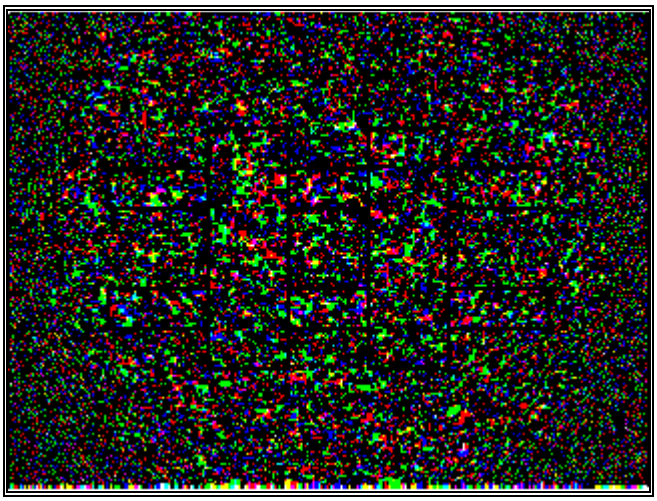


Figure 4.20 Edge Detection Algorithm Applied To Figure 4.19

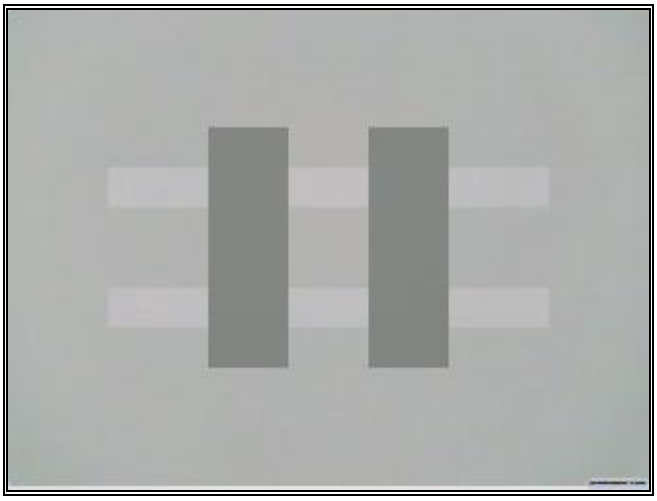


Figure 4.21 Fuzzy Algorithm Applied To Figure 4.19

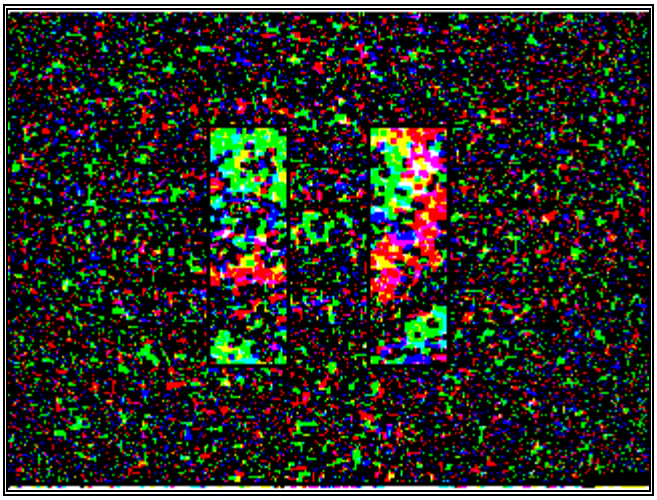


Figure 4.22 Edge Detection Algorithm Applied to Figure 4.21

Figure 4.23 shows the result of applying the fuzzy algorithm to the Test Image #4. Each pixel was gain corrected and adjusted to provide a linear response at all fourteen gray scales, defined in Figure 3.4. Figure 4.24 through Figure 4.26 display the 120th horizontal scan line of the bitmap displayed in Figure 4.23 for the color red, green, and blue respectively. The red, green, and blue scan lines display the video signal after the image was filtered, and the black scan lines display the signal before filtering. In Figure 4.26 it is noticeable that the pixel gain for the blue response is increased, and the high frequency components of the noise signal is attenuated. The same results can be observed in Figure 4.24 and Figure 4.25.

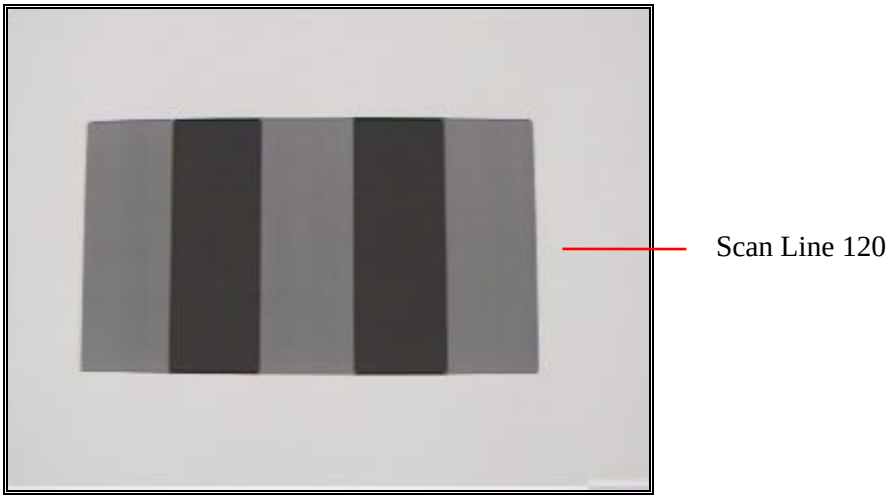


Figure 4.23 Test Image #4, Fuzzy Algorithm

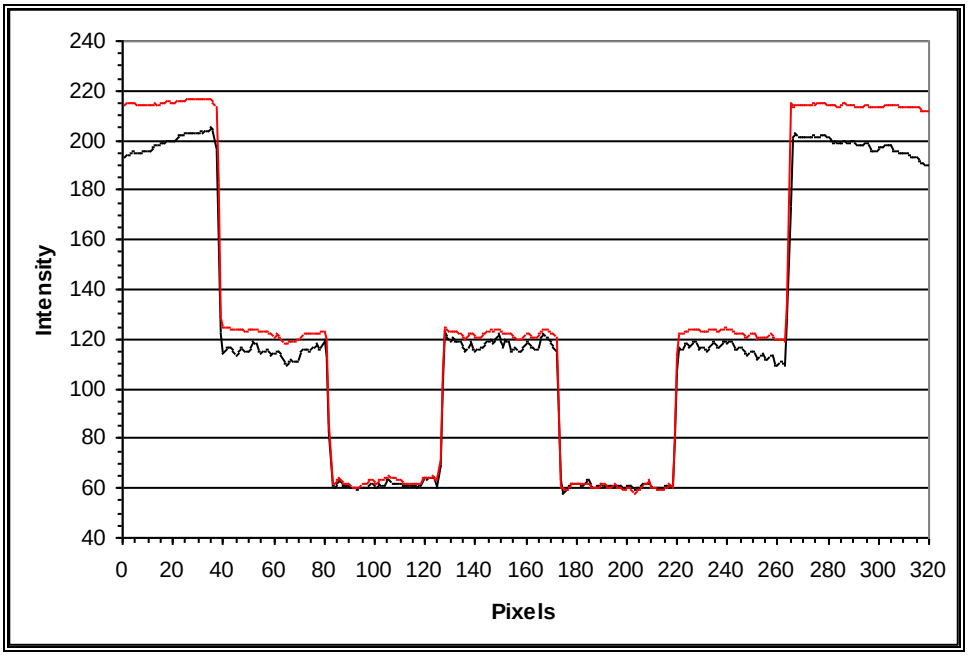


Figure 4.24 Scan Line 120, Red

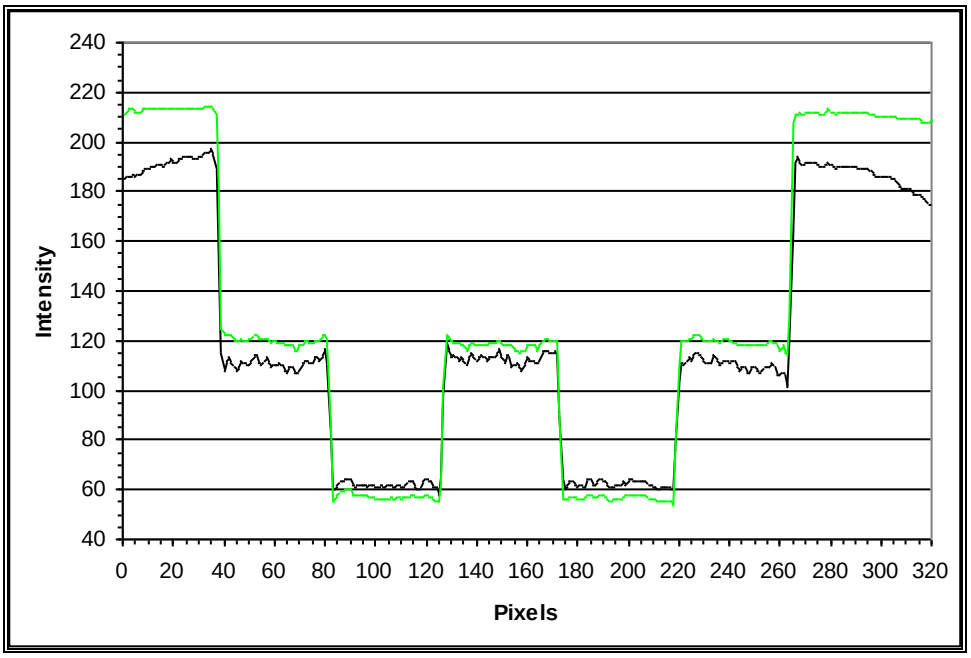


Figure 4.25 Scan Line 120, Green

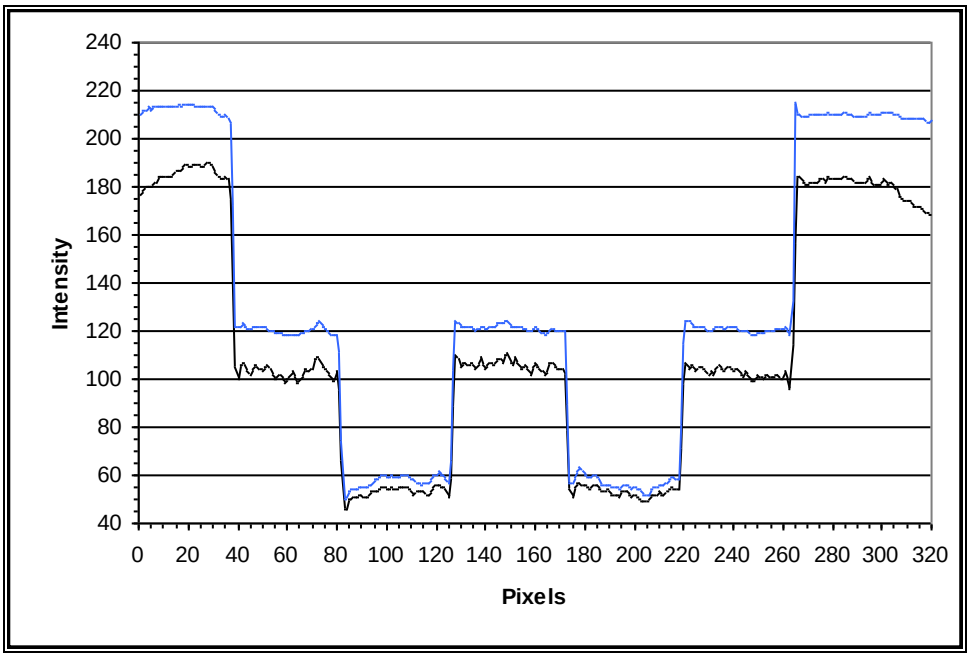


Figure 4.26 Scan Line 120, Blue

Chapter 5

Performance Evaluation

The evaluation of the fuzzy algorithm was performed using two different types of test images. Test Image #5 was obtained by filming the gray scale 5.5 in a controlled studio environment, and Test Image #6 was obtained capturing a natural live scene. Test Image #6 was first recorded to an eight mm film; and, afterwards the images were captured to an AVI file and converted to the bitmaps. Each test image comprises of different uniformity problems and video signal characteristics. During the evaluation the fuzzy algorithm proved robust and performed well in the enhancement of either test image.

Figure 5.1 displays Test Image #5. The image comprises nine frame averages and is severely degraded by noise and non-uniformity. The result of applying the fuzzy filter to the image is displayed in Figure 5.2. Each pixel was gain corrected and adjusted for linearity to achieve a uniform response for the colors red, green, and blue. Table 5.1 shows quantitative results of applying the fuzzy algorithm to the uniform gray scale 5.5. The statistical information for frame averages of one, nine, and one hundred frames is displayed for a further comparison of the fuzzy algorithm and the classical filters. See Table 3.3 through Table 3.5 for data obtained using the classical methods.

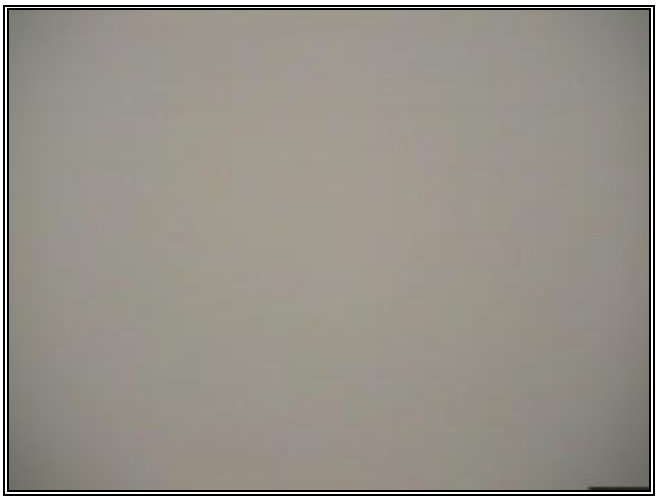


Figure 5.1 Test Image #5, Gray Scale 5.5

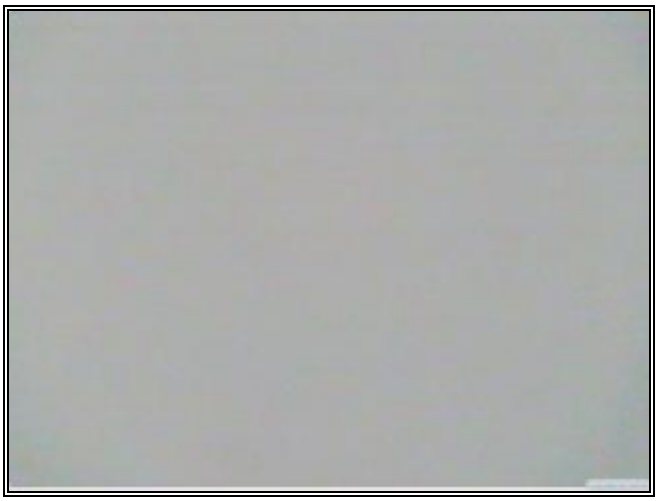


Figure 5.2 Test Image #5, Fuzzy Algorithm

Frame Averaging	Mean			Standard Deviation		
	Red	Green	Blue	Red	Green	Blue
1	173.00	174.68	173.19	5.05	3.47	7.04
9	172.15	173.97	172.58	3.35	2.32	3.38
100	176.09	177.80	175.99	2.86	2.22	2.93

Table 5.1 Fuzzy Algorithm Applied To Uniform Gray Scale Image

The fuzzy algorithm reduced the non-uniformity of the image by 5.07, 5.26, and 3.80 counts of standard deviation for the colors red, green and blue respectively. Applying the gain correction, median filter, and box filter to the Test Image #5 reduced the non-uniformity by 6.16, 5.73, and 5.16 counts of standard deviation for the color red, green, and blue respectively. The classical filtering methods provide slightly better results in smoothing the image; however, high frequency components of the relevant signal are not protected. Applying one additional iteration of the fuzzy filter to the Test Image #5 further reduced the non-uniformity by 0.25 counts.

Figure 5.3 displays one video frame of the Test Image #6. The image details consist of a satellite mounted inside the space shuttle's cargo bay. Figure 5.4 through Figure 5.6 show the 124th horizontal scan line for the color components red, green, and blue. Figure 5.7 shows the results of applying the fuzzy filter to the Test Image #6. The 124th horizontal scan line of the fuzzy filtered image is displayed in Figure 5.8 through Figure 5.10. The red, green, and blue video signals show that the rising and falling edges of the sharp image transitions are maintained. Altering these edges would result in blurring image details. Filtering of higher frequency components is clearly noticeable at pixel location 220 through 270.

Figure 5.3 Test Image #6, Satellite Mounted In The Cargo Bay

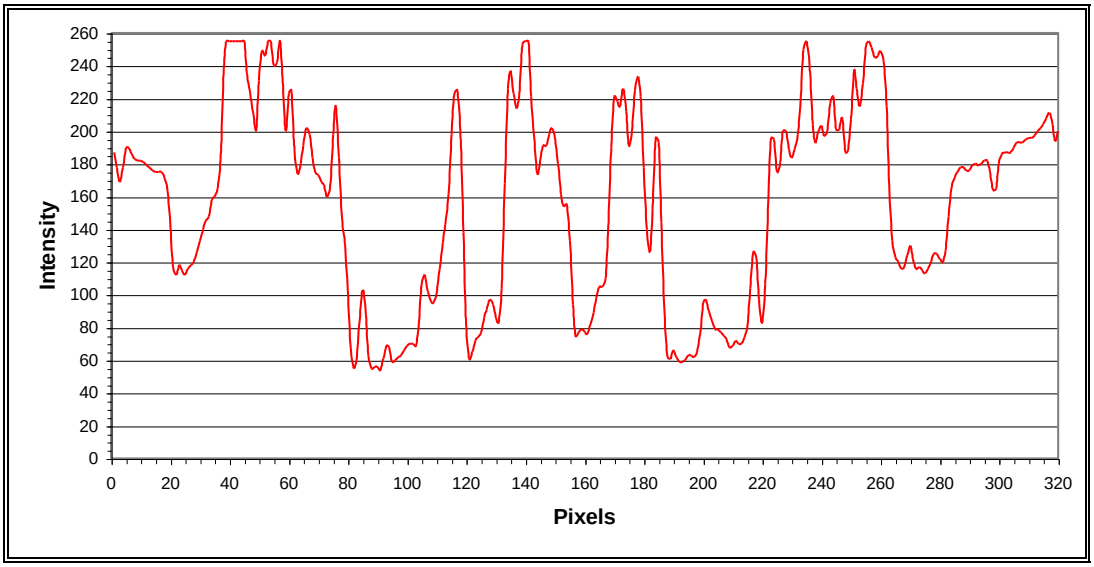


Figure 5.4 Scan Line 124, Red

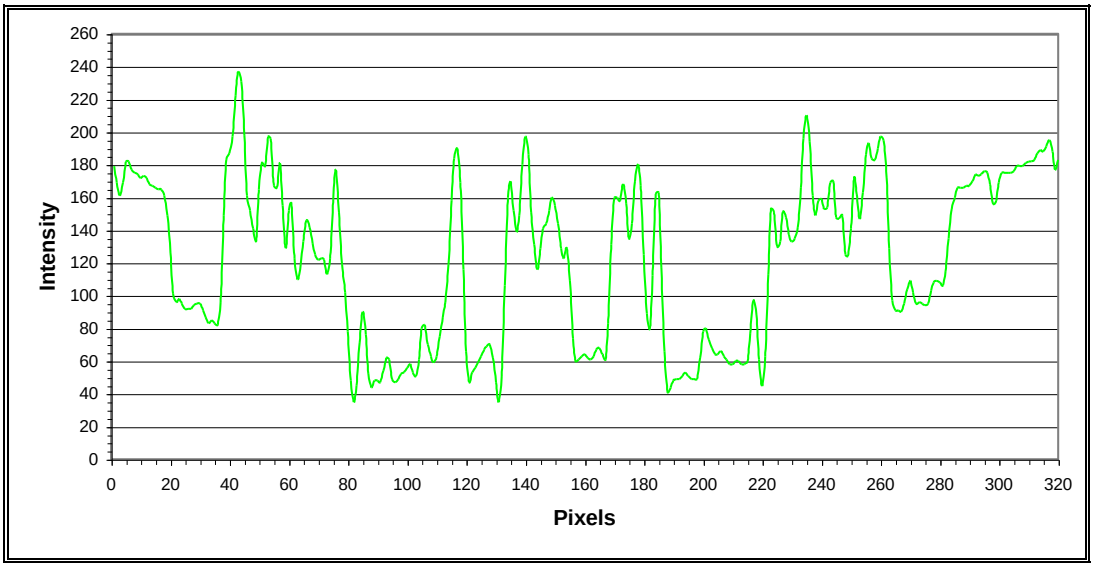


Figure 5.5 Scan Line 124, Green

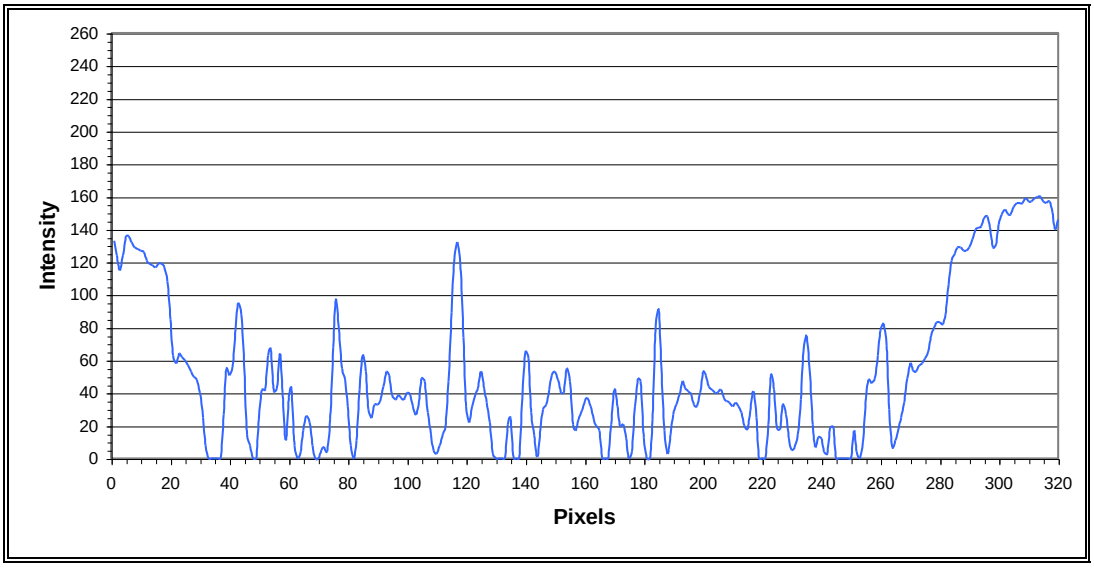


Figure 5.6 Scan Line 124, Blue

Figure 5.7 Test Image #6, Fuzzy Filter

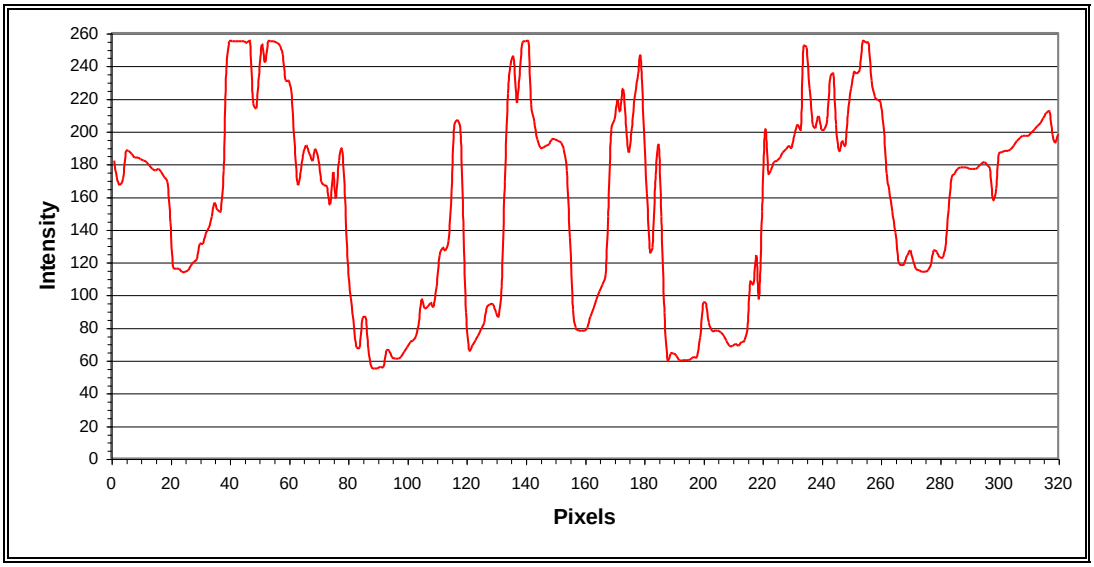


Figure 5.8 Fuzzy Filter, Scan Line 124, Red

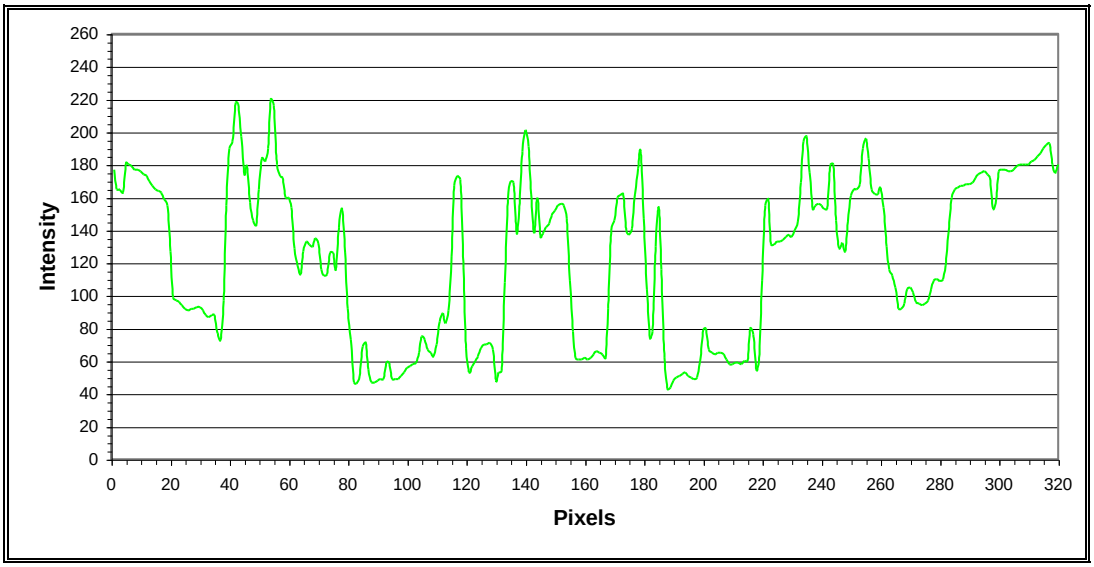


Figure 5.9 Fuzzy Filter, Scan Line 124, Green

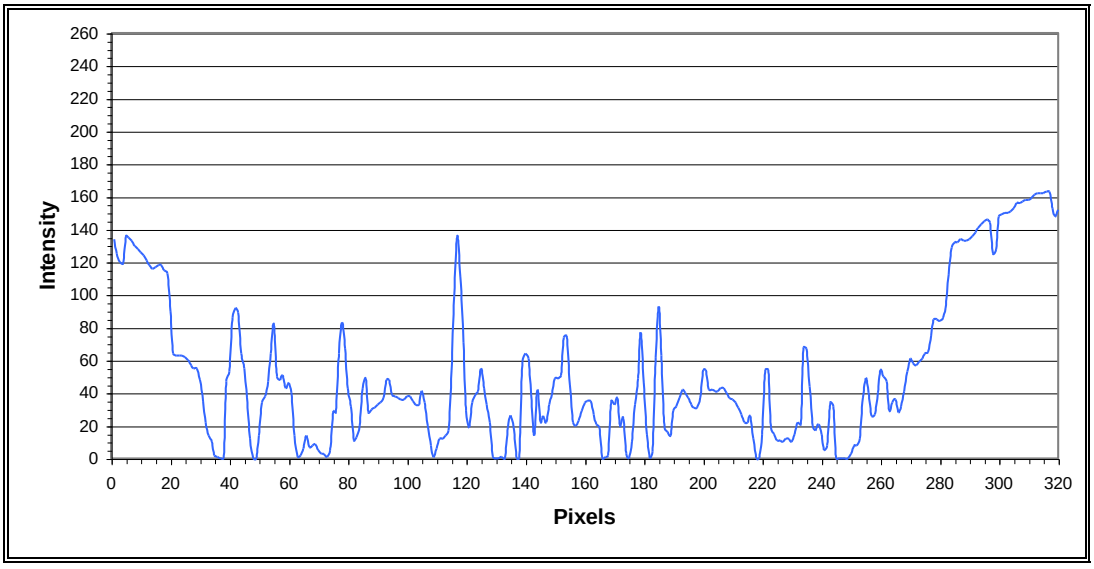


Figure 5.10 Fuzzy Filter, Scan Line 124, Blue

Figure 5.11 shows the result of applying the gain correction, median filter, and binomial filter to the Test Image #6. The image appears slightly blurred and several image details are fading. The crisp reflections in the protective shielding of the satellite are lost due to the attenuation of the high frequency components. Classical filtering methods cannot determine differences between the relevant information and the noise signal. The fuzzy algorithm performs better in this regard of image processing. A comparison between Figure 5.11 and the fuzzy filtered image shown in Figure 5.7 indicates the differences.

Figure 5.12 through Figure 5.14 display the red, green, and blue video signal of the 124th horizontal scan line. The sharp leading and trailing edges of the image details are rounded indicating that the high frequency components are severely attenuated.

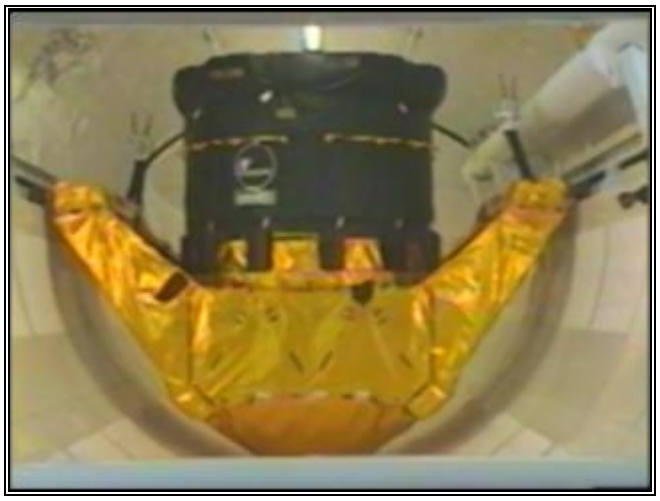


Figure 5.11 Gain Correction, Median Filter And Binomial Filter

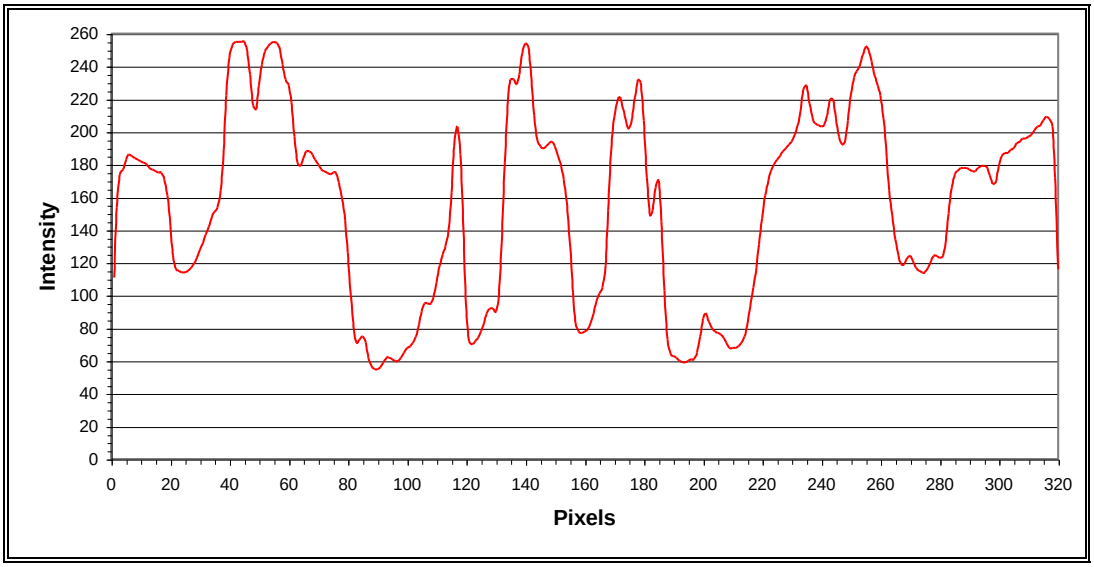


Figure 5.12 Scan Line 124, Gain Correction, Median And Binomial Filter, Red

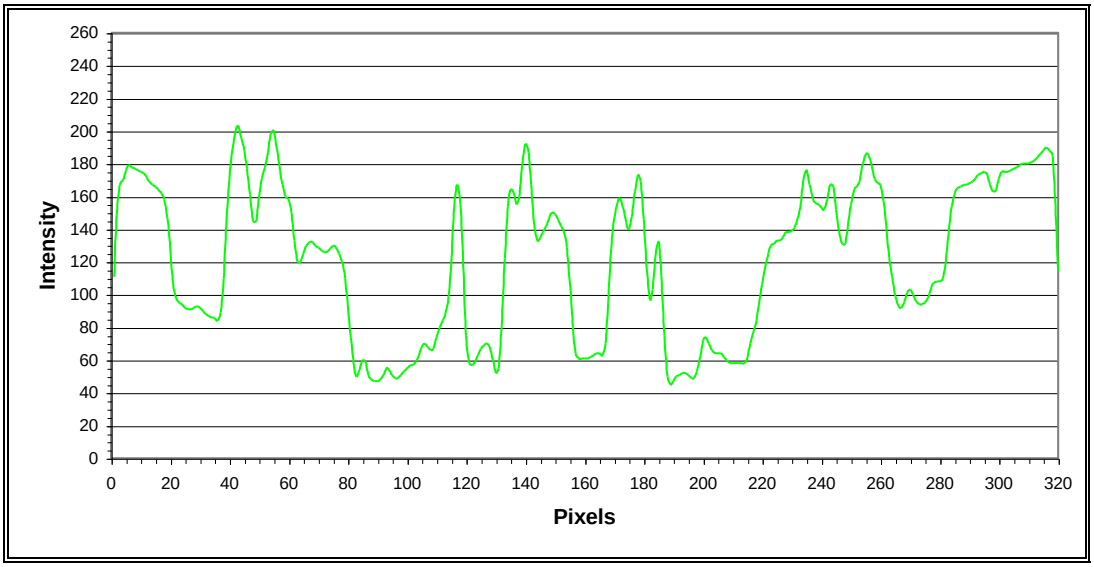


Figure 5.13 Scan Line 124, Gain Correction, Median And Binomial Filter, Green

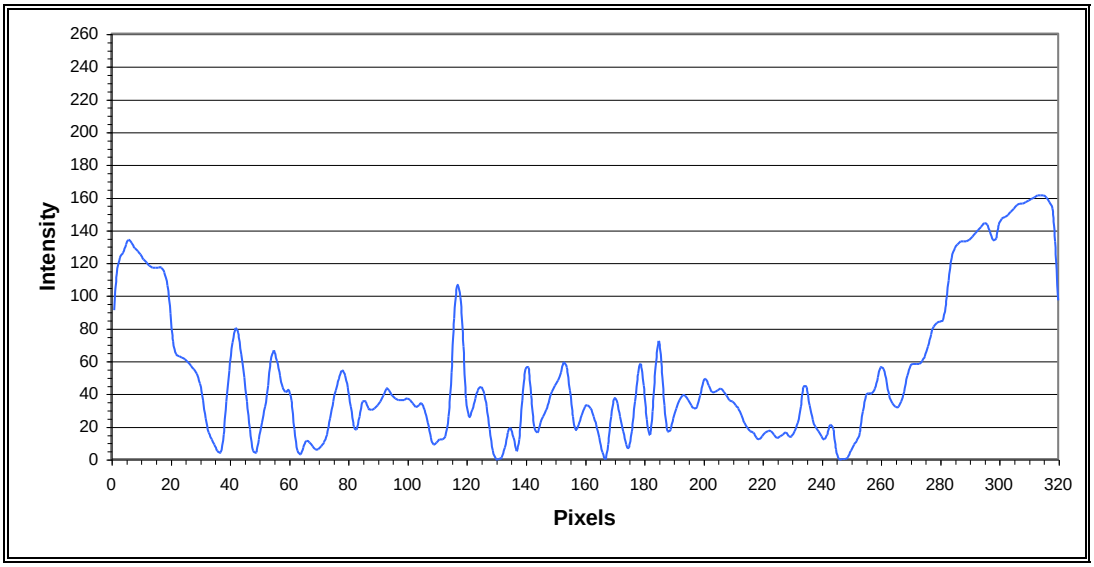


Figure 5.14 Scan Line 124, Gain Correction, Median And Binomial Filter, Blue

Figure 5.15 shows the differences that are generated by subtracting the Test Image #6 from the fuzzy filtered image, and normalizing the result to RGB(128, 128, 128). Only the noise component present in the original image should be visible if the algorithm is performing perfectly. Figure 5.16 shows the differences between the image filtered by the classical method and the original Test Image #6. The differences between Figure 5.15 and Figure 5.16 are illustrated in Figure 5.17. It can be observed that both filtering methods successfully attenuated the noise signal; but, the classical filtering method is more aggressive towards transitions of image details.

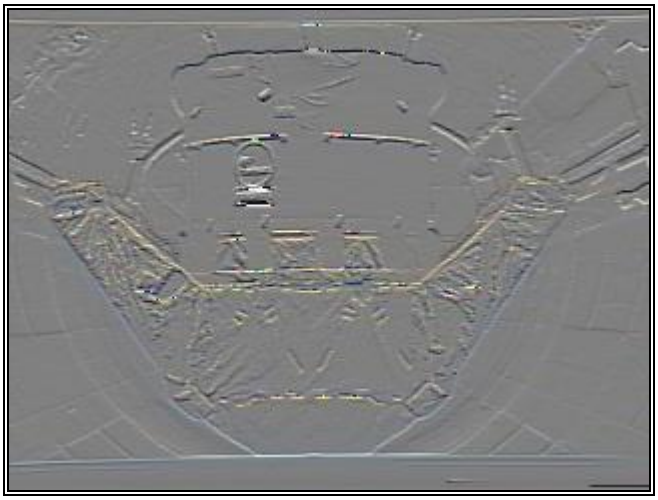


Figure 5.15 Fuzzy Filter - Test Image #6

Figure 5.16 Median And Binomial Filter - Test Image #6

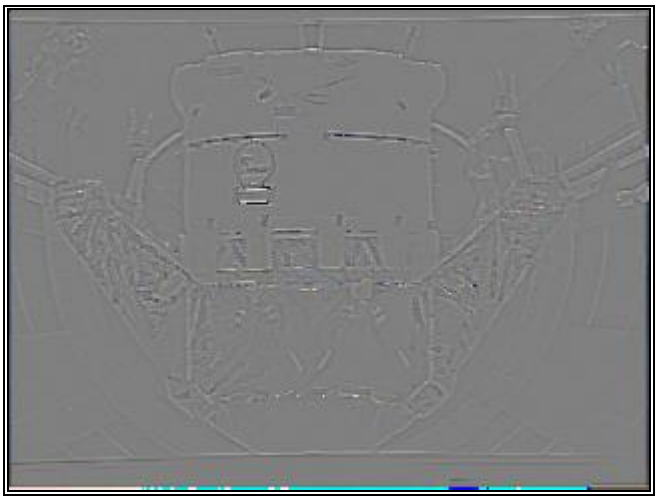


Figure 5.17 Differences between Classical Filter And Fuzzy Filter

Figure 5.18 illustrates the ability of the fuzzy controller to protect image details from filtering. The image displayed in Figure 5.18 is the result of applying four iterations of the fuzzy filter to the Test Image #6. The image does not indicate blurring and image details are fully preserved. This characteristic of the fuzzy algorithm presents a strong advantage over classical filtering methods. Applying several iterations of the classical filters to an image would result in blurring and the loss of image details. The image displayed in Figure 5.11 is the result of applying one iteration of the median filter and binomial filter to the Test Image #6, and it appears less sharp than the image displayed in Figure 5.18.

Figure 5.18 Test Image #6, Applying Four Iterations Of Fuzzy Filter

Chapter 6

Summary

This Thesis presents the design of a fuzzy algorithm for the correction and enhancement of noise degraded images. The design criteria for the algorithm involves the removal of the noise signal from the degraded image without altering the relevant signal features. Protecting relevant information and filtering noise requires an automated decision making process at the individual pixel level. The results of this study show that fuzzy logic proves highly effective in solving this critical image processing problem.

The approach of the design of the fuzzy controller involved the determination of the rule base and the membership functions. The rule base and the inference machine replaced the need for mathematically defining the relationship between the input and output variables. The fuzzy controller processes the individual pixel data, and determines the strength and type of filtering necessary to attenuate the noise components. If none of the fuzzy rules become activated while processing information of a particular pixel, then the pixel data remains unchanged. The advantage of this process is that the fuzzy operator protects the high frequency components of image details and successfully filters exponential noise and Gaussian noise.

The experimental results show that the fuzzy algorithm performs better than the classical filtering methods. The fuzzy filter can be applied repetitively to an image to obtain additional smoothing and filtering, which is a feature that is not possible using the box filter or binomial filter. Regardless how many iterations the fuzzy filter is applied to an image, the image details remain preserved. Blurring of image details is not observed using the fuzzy algorithm.

The research conducted for this Thesis included the evaluation of noise reduction through frame averaging and adjusting individual pixel gains. This method provided the best results; but, it required numerous gain correction curves and the capturing of at least one hundred frames of static images. The alternative method of using the fuzzy algorithm and selecting nine frame averages provided good results. An improvement in the signal-to-noise-ratio of approximately 8.2 dB was obtained.

Future research should expand the fuzzy algorithm to provide for three dimensional image processing. Additional research should also explore processing noise degraded images in a different color space. In the studies conducted for this Thesis the image processing was limited to correcting the intensity component of the video signal.

References

[1] Vladimir Z. Mesarovic, Nilolas P. Galatsanos, and Aggelos K. Katsaggelos, "Regularized Constrained Total Least Squares Image Restoration," IEEE Transactions on Image Processing, Vol. 4, No. 8, (August 1995) .

[2] Bing Zeng, "Optimal Median-Type Filtering Under Structural Constraints," IEEE Transactions on Image Processing, Vol. 4, No. 7, (July 1995) .

[3] Wai Ho Pun, and Brian D. Jeffs, "Adoptive Image Restoration Using a Generalized Gaussian Model for Unknown Noise," IEEE Transaction on Image Processing, Vol. 4, No. 10, (October 1995) .

[4] Y. Egusa, H. Akahori, A. Morimura, N. Wakami, "An Appplication of Fuzzy Set Theory for an Electronic Video Camera Image Stabilizer," IEEE Transaction on Fuzzy Systems, Vol. 3, No. 3, (August 1995) .

[5] Fabrizio Russo , and Giovanni Ramponi, "A Fuzzy Operator for the Enhancement of Blurred and Noisy Images," IEEE Transactions on Image Processing, Vol.4, NO. 8, (August 1995) .

[6] Charles A. Poynton, "A Technical Introduction to Digital Video," John Wiley & Sons, Inc., 1996 .

[7] Arthur R. Weeks, G. Eric Hague, and Harley R. Myler, "Histogram Equalization of 24-bit Color Images in the Color Difference (C-Y) Color Space," Journal of Electronic Imaging 4(1), 15-22 (January 1995) .

[8] George Eric Hague II, "Color Image Enhancement Using Both Chromatic and Luminance Components," Thesis, University of Central Florida College of Engineering, (April 1994) .

[9] John C. Russ, "The Image Processing Handbook," CRC Press, Inc., (1994) .

[10] Bernd Jahne, "Digital Image Processing," Springer-Verlag, Second Edition, 1993).

[11] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, "Numerical Recipes in C," Cambride University Press, Second Edition (1992) .

[12] Daniel McNeill and Paul Freiberger, " Fuzzy Logic," Touchstone, 1993 .

[13] Lotfi Zadeh," Fuzzy Sets," Information and Control, Vol. 8, No. 3, June 1965, pp. 338 - 353.

[14] Bart Kosko, "Neural Networks and Fuzzy Systems", Prentice Hall, 1992 .

[15] Ching-Yu Tyan and Paul P. Wang, " Image Processing - Enhancement, Filtering and Edge Detection Using the Fuzzy Logic Approach," Proc. Second IEEE Int. Conf. Fuzzy Systems, San Francisco, CA, Mar, 28 - Apr. 1, 1993 .

[16] Fabrizio Russo, "A new Class of Fuzzy Operators for Image Processing: Design and Implementation," Proc. Second IEEE Int. Conf. Fuzzy Systems, San Francisco, CA, Mar, 28 - Apr. 1, 1993, pp. 815 - 820.

[17] Rangaraj M. Rangayyan, and Salaheddin G. Elkadiki, "Algorithm for the Computation of Region-Based Image Edge Profile Acutance," Journal of Electronic Imaging 4(1), 62-70 (January 1995) .

[18] C.Wayne Brown and Barry J. Shepherd, "Graphics File Formats," Manning Publications Co. , (1995) .

[19] Ronald N. Bracewell, " Two-Dimensional Imaging, "Prentice Hall Inc, (1995) .

[20] John G. Proakis, Charles M. Radar, Fuyun Ling, and C. Nikias, "Advanced Digital Signal Processing," Macmillian Publishing Company, (1992) .

[21] Irwin Miller, and John E. Freund, "Probability and Statistics for Engineers," Prentice Hall Inc , (1985) .

Appendix A

Imager

Imager was designed to provide for the enhancement of noise degraded images using either classical filtering methods or the fuzzy algorithm developed for this Thesis. All bitmaps imported to Imager must have a resolution of 320 x 240 pixels and display 24-Bit true color. The program was developed to operate within the Microsoft Windows 95 environment. The minimum system requirements include an Intel 80386 or faster PC with at least 8 megabytes (MB) of application random-access memory (RAM), and a color SVGA display adapter and compatible monitor. The video display adapter must be capable of displaying 24-Bits at resolutions of 640 x 480. The video adapter should use the Cirrus Logic CL-GD5426 Controller or equivalent.

Installation And Execution

Imager and the supporting files must be installed on the C Drive with the directory structure listed below. To execute the program from Windows 95 select **START**, **RUN**, and enter at the prompt : **C:\IMAGER\IMAGER** .

Directory and required files :

C:\IMAGER\IMAGER.EXE
C:\IMAGER\IMAGER.WND
C:\IMAGER\ANALYSIS.WND

C:\IMAGER\DATA\STDEV.BLU
C:\IMAGER\DATA\STDEV.GRN
C:\IMAGER\DATA\STDEV.RED

C:\IMAGER\GAIN\DARK.BMP
C:\IMAGER\GAIN\DK.BMP
C:\IMAGER\GAIN\GR70.BMP
C:\IMAGER\GAIN\GRAY20.BMP
C:\IMAGER\GAIN\GRAY25.BMP
C:\IMAGER\GAIN\GRAY30.BMP
C:\IMAGER\GAIN\GRAY35.BMP
C:\IMAGER\GAIN\GRAY40.BMP
C:\IMAGER\GAIN\GRAY45.BMP
C:\IMAGER\GAIN\GRAY50.BMP
C:\IMAGER\GAIN\GRAY55.BMP
C:\IMAGER\GAIN\GRAY60.BMP
C:\IMAGER\GAIN\GRAY65.BMP
C:\IMAGER\GAIN\GRAY70.BMP

C:\IMAGER\VIDEO\FR0.BMP
C:\IMAGER\VIDEO\FR1.BMP
C:\IMAGER\VIDEO\FR2.BMP
C:\IMAGER\VIDEO\FR3.BMP
C:\IMAGER\VIDEO\FR4.BMP
C:\IMAGER\VIDEO\FR5.BMP
C:\IMAGER\VIDEO\FR6.BMP
C:\IMAGER\VIDEO\FR7.BMP
C:\IMAGER\VIDEO\FR8.BMP
C:\IMAGER\VIDEO\FR9.BMP

Software Functions

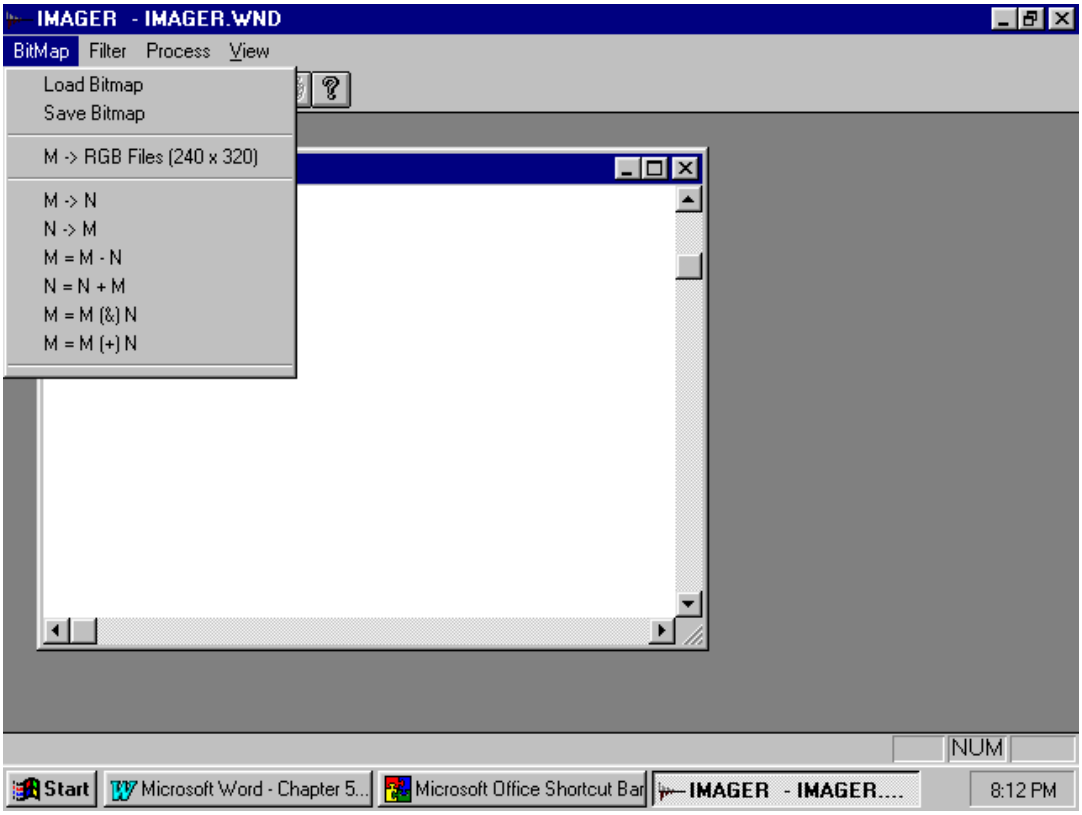


Figure A.1 Imager Bitmap Menu

Figure A.1 shows the Imager pull down menu Bitmap. The nine options are defined below.

Load Bitmap :

The user can select a 320 x 240 Bitmap and display it in the Imager window. The pixel data is stored in the matrix m_bm for further processing.

Save Bitmap :

Save the pixel data stored in matrix m_bm to a user defined 320 x 240 Bitmap.

M -> RGB Files (240 x 320) :

Save the pixel data stored in matrix m_bm to a user defined data file compatible with the EXCEL spreadsheet. The 320 x 240 image format is translated into a 240 x 320 format to provide for the graphical display in EXCEL.

M -> N :

Transfer pixel data from matrix m_bm to matrix n_bm. Image data stored in matrix n_bm is not displayed in the Imager window.

N -> M :

Transfer pixel data from matrix n_bm to matrix m_bm. Image data transfered to matrix m_bm is automatically displayed in the Imager window.

M = M - N :

Subtract pixel data stored in matrix n_bm from pixel data stored in matrix m_bm. The result is automatically displayed in the Imager window.

N = N + M :

Add pixel data stored in matrix m_bm to pixel data stored in matrix n_bm. The result stored in matrix n_bm is not displayed.

M = M (&) N :

Logical AND pixel data stored in matrix m_bm and matrix n_bm. The result stored in matrix m_bm is automatically displayed in the Imager window.

M = M (+) N :

Logical OR pixel data stored in matrix m_bm and matrix n_bm. The result stored in matrix m_bm is automatically displayed in the Imager window.

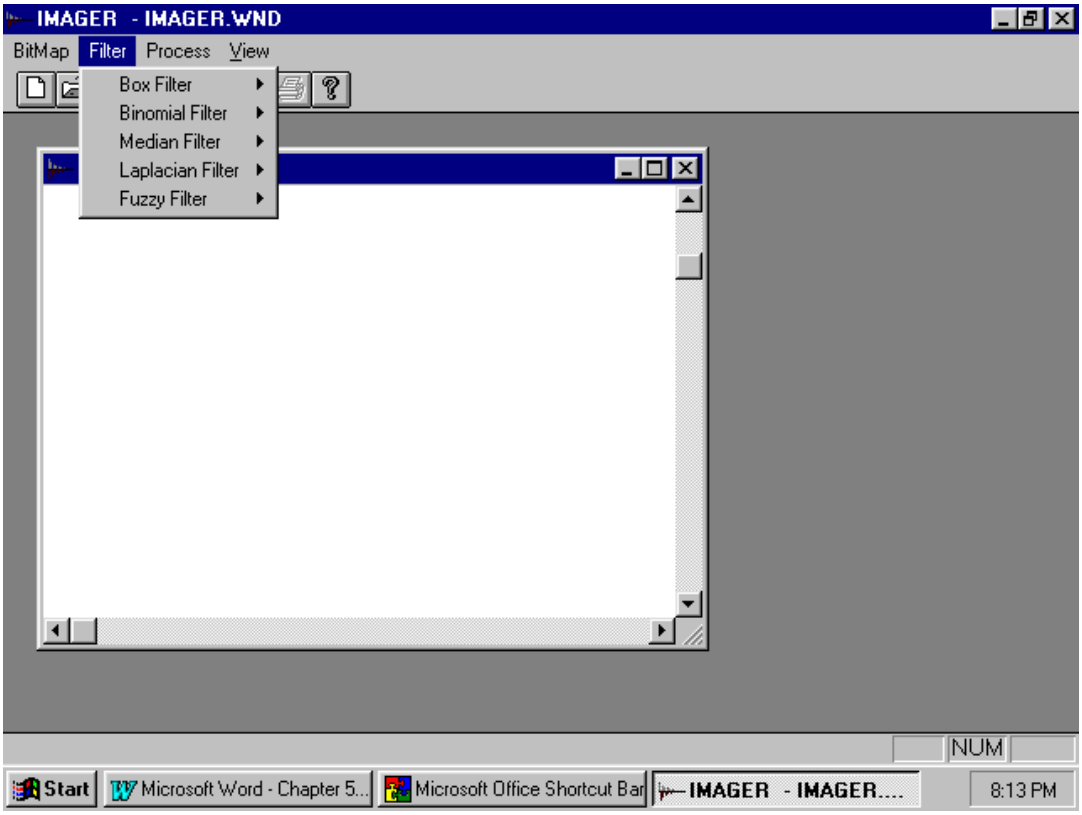


Figure A.2 Imager Filter Menu

Figure A.2 shows the Imager pull down menu Filter. The five categories and fourteen options are defined below.

Box Filter :

Applies a box filter to the pixel data stored in matrix m_bm, The result is stored in matrix m_bm and is automatically displayed in the Imager window. The user can select one of three convolution masks of size 3 x3, 5 x 5, and 7 x7.

Binomial Filter :

Applies a binomial filter to the pixel data stored in matrix m_bm, The result is stored in matrix m_bm and is automatically displayed in the Imager window. The user can select one of three convolution masks of size 3 x3, 5 x 5, and 7 x7.

Median Filter :

Applies the median filter to the pixel data stored in matrix m_bm, The result is stored in matrix m_bm and is automatically displayed in the Imager window. The user can select one of three different kernels of size 3 x3, 5 x 5, and 7 x7.

Laplacian Filter :

The user can select three different options of applying the Laplacian operator to the pixel data stored in the image matrix m_bm. The options include the standard kernel size 3x3, normalization to RGB(128,128,128), and contrast enhancement that results in a white uniform image with the image details outlined in black. The result is stored in matrix m_bm and is automatically displayed in the Imager window.

Fuzzy Filter :

Applies fuzzy algorithm to the pixel data stored in matrix m_bm, The result is stored in matrix m_bm and is automatically displayed in the Imager window. The matrix n_bm is used during the image enhancement process, and previously stored pixel data in matrix n_bm will be lost. The user can de/select frame averaging, pixel gain adjustment, fuzzy filtering, and linearity adjustment. In addition the range of filtering can be limited to user specified image rows.

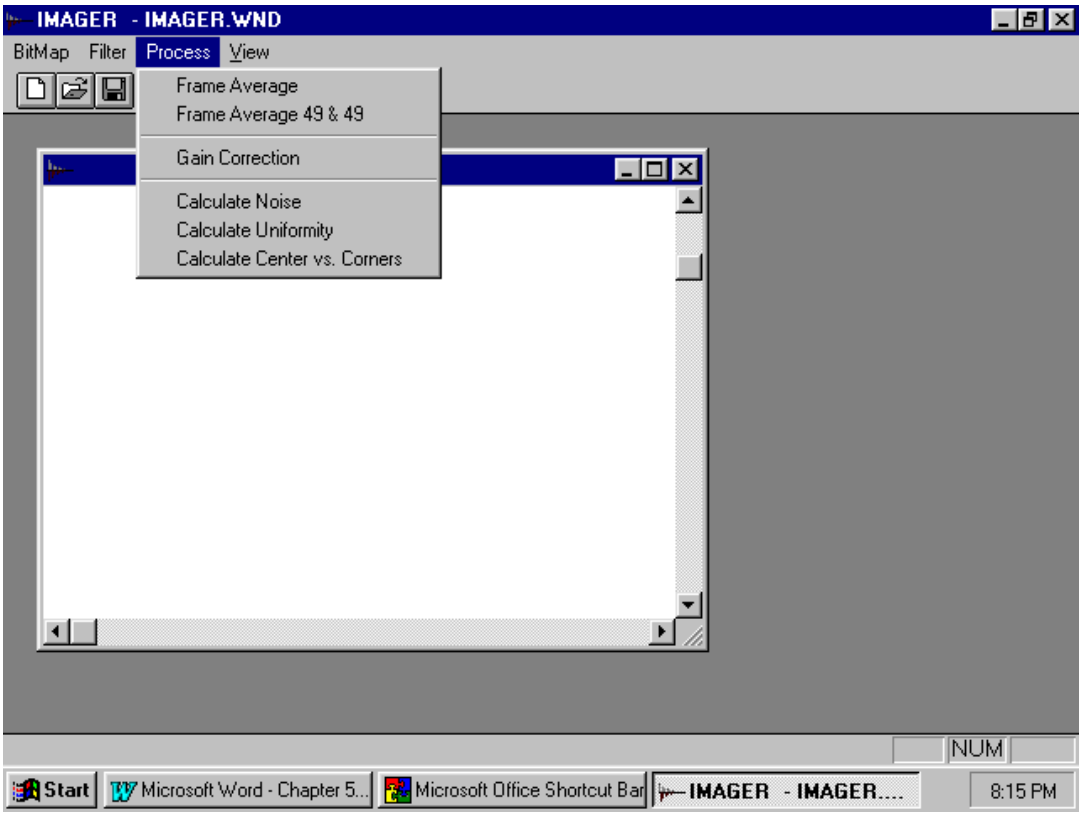


Figure A.3 Imager Process Menu

Figure A.3 shows the Imager pull down menu Process. The user can select from five options, and they are defined below.

Frame Averaging :

Frame averages a user defined number of frames (1 - 49 frames). The result is stored in matrix m_bm and is automatically displayed in the Imager window. The user must copy the selected frames into the directory C:\Imager\Video prior to performing frame averaging. The captured frames must be stored in the bitmap format with the filename starting at FR0.BMP and continue with FR1.BMP ..... to FR48.BMP.

Frame Average 49 & 49 :

Frame averages an existing user defined bitmap consisting of 49 frame averages with 49 additional frames (FR0.BMP ... to FR48.BMP) stored in the directory C:\Imager\Video. The result is stored in the matrix m_bm and is automatically displayed in the Imager window.

Gain Correction :

Performs pixel gain and offset correction for the image data stored in matrix m_bm. The process uses 12 lookup tables derived from the bitmaps stored in the directory C:\IMAGER\GAIN. The set of 11 bitmaps, GRAY20.BMP through GRAY70.BMP, may be replaced with a set of bitmaps reflecting the different camera zoom settings. The bitmaps DK.BMP and GR70.BMP are used by the fuzzy enhancement algorithm, and they can also be replaced to accommodate different camera settings.

Calculate Noise :

Calculates the average standard deviation of pixel digitizing counts using FR0.BMP through FR9.BMP in the directory C:\IMAGER\VIDEO. The user must copy the image data to these bitmaps prior to selecting this option. The random noise data for each pixel is stored in the files STDEV.RED, STDEV.GRN, and STDEV.BLU. Previous stored data in these files will be lost. The files reside in the directory C:\IMAGER\DATA.

Calculate Uniformity :

Calculates the standard deviation of all pixel luminance values stored in matrix m_bm. The pixel data stored in matrix m_bm should reflect the result of capturing a uniform scene. The result is displayed in the Imager window, and it includes the mean and standard deviation for each color red, green, and blue.

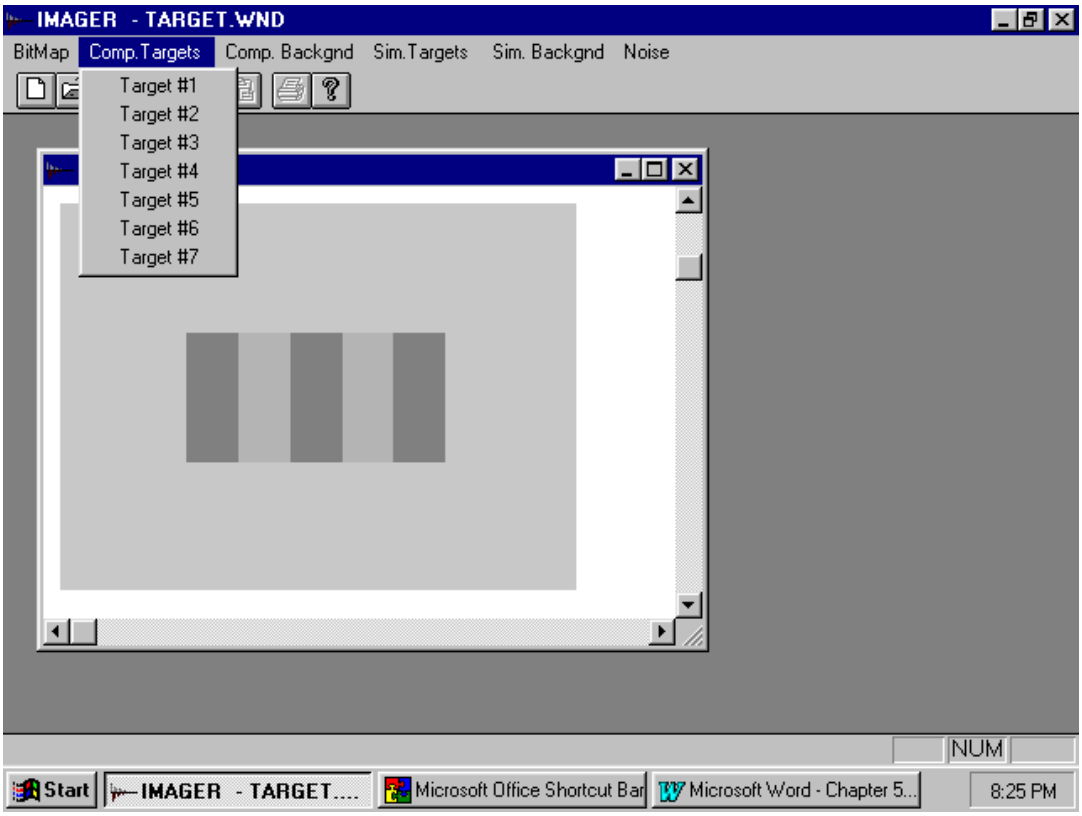


Figure A.4 Imager Target Window

Figure A.4 displays the Target window and the associated pull down menu. The user can select from 6 categories and 26 options which are defined below.

Bitmap :

The user can select from the options **Load Bitmap**, **Save Bitmap**, and **M->RGB Files (240 x 320)**. These options perform the same function available in the Imager window.

Comp. Targets :

The user can select from 7 different computer generated targets of various shapes and uniform backgrounds. Each image is void of noise and image abnormalities.

Comp. Backgnd :

The user can select from four uniform image backgrounds. The luminance values are RGB(80,80,80), RGB(128,128,128), RGB(160, 160, 160), and RGB(200, 200, 200). Each background is void of noise. The pixel data is stored in matrix m_bm and is automatically displayed in the Target window.

Sim. Targets :

The user can select from seven computer simulated images. The images are degraded by non-uniformity problems caused by camera optics, and temporal and spatial noise. The images are created from the 12 bitmaps DARK.Bmp, and GRAY20.Bmp through GRAY70.Bmp stored in the directory C:\IMAGER\GAIN. The shape and sizes of the targets are identical to the computer generated images. The pixel data is stored in matrix m_bm and is automatically displayed in the Target window.

Sim. Backgnd :

The user can select from 4 computer simulated images. The uniform image backgrounds are degraded by non-uniformity problems, temporal noise and spatial noise. The pixel data is stored in matrix m_bm and is automatically displayed in the Target window.

Noise :

This function randomly selects ten percent of the pixels in matrix m_bm and degrades their intensity values with exponential noise. The additive noise is Gaussian distributed and passes all statistical tests. The result is stored in matrix m_bm and is automatically displayed in the Target window.