5-1-2013

Real Time Digital Night Vision Using Nonlinear Contrast Enhancement

Nishikar Sapkota
University of Nevada, Las Vegas, niishiikar@gmail.com

REAL TIME DIGITAL NIGHT VISION

USING NONLINEAR CONTRAST ENHANCEMENT

by

Nishikar Sapkota

Bachelor of Engineering (Computer)
Pokhara University, Nepal
2008

A thesis submitted in partial fulfillment of
the requirements for the

Master of Science in Computer Science

Department of Computer Science
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
May 2013

THE GRADUATE COLLEGE

We recommend the thesis prepared under our supervision by

Nishikar Sapkota

entitled

Real Time Digital Night Vision Using Nonlinear Contrast Enhancement

be accepted in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science
Department of Computer Science

Evangelos Yfantis, Ph.D., Committee Chair

Laxmi Gewali, Ph.D., Committee Member

John Minor, Ph.D., Committee Member

Peter Stubberud, Ph.D., Graduate College Representative

Tom Piechota, Ph.D., Interim Vice President for Research &
Dean of the Graduate College

May 2013

Abstract

This thesis describes a nonlinear contrast enhancement technique to implement night vision in digital video. It is based on the global histogram equalization algorithm. First, the effectiveness of global histogram equalization is examined for images taken in low illumination environments in terms of Peak signal to noise ratio (PSNR) and visual inspection of images. Our analysis establishes the existence of an optimum intensity for which histogram equalization yields the best results in terms of output image quality in the context of night vision. Based on this observation, an incremental approach to histogram equalization is developed which gives better results than the conventional approach in terms of PSNR. This algorithm is also applied to implementing video surveillance in poorly illuminated environments to achieve real time night vision. This involves the application of histogram equalization to digital video frames with data transmission and buffering over a computer network.

Acknowledgements

I am very grateful to my thesis supervisor, Dr. Evangelos Yfantis, Professor, Department of Computer Science, UNLV for his guidance and motivation in completing this thesis. His readiness for consultation at every stage of my work and his invaluable advice and ideas helped me in achieving the best I could.

I am grateful to have Dr. Laxmi Gewali, Dr. John Minor and Dr. Peter Stubberud in this thesis committee and would like to thank them for their support. I would also like to thank my friend and colleague Mr. Dinesh Bajracharya for his help on network programming.

Nishikar Sapkota

University of Nevada, Las Vegas
May 2013

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The goal of any night vision technology is to enable a person to see in the dark. In the past, night vision was implemented by making use of the infrared spectrum of electromagnetic waves and devices such as image intensifiers. With the growing popularity of digital computing, many digital image processing techniques have been proposed([1], [2], [3]) to implement night vision. These techniques can enhance the images captured by ordinary cameras under low light conditions and can be implemented completely in software. They do not require the use of infrared light and special devices. One of the popular and efficient algorithms for adjusting the contrast of an image is a nonlinear contrast enhancement technique called histogram equalization. With respect to night vision, many algorithms based on this technique have been proposed and successfully implemented([4], [5], [6]). In this thesis an improved histogram equalization approach with respect to night vision is presented. This algorithm is applied to video surveillance which involves the capturing of image frames from a camera, video transmission, recording and buffering over a computer network and application of histogram equalization to these frames in real time. Due to these requirements, the efficiency of the technique used is of critical importance. As it is intended to be part of an intelligent video surveillance system, the quality of the images produced is also extremely important. The end result must be suitable for the application of object detection and pattern recognition algorithms which would otherwise not have been possible for images captured in the dark.

1.2 Objective

In most histogram equalization algorithms the cumulative distribution function of the image histogram is normalized such that the maximum intensity value of the equalized image is 255 (the maximum possible intensity value for the commonly used 8-bit encoded grayscale image). This thesis proposes to show that this approach does not always yield the optimum result for night vision. By gradually increasing the maximum intensity level from the maximum intensity of the original image

to 255, the existence of an optimal intensity is shown to give better results in terms of subjective quality of the equalized image and also the Peak Signal to Noise Ratio (PSNR) when compared to the image taken under "well-illuminated" conditions. The application of this technique is evaluated on several images taken under low light conditions and the results in terms of PSNR are compared to support our claim. Based on this observation, a modified approach to implementing histogram equalization is developed which yields better results.

Chapter 2

Background

Vision is the most advanced of human senses and images plays a crucial role in our perception. However, human vision is only limited to a small portion of the electromagnetic spectrum which is called visible light. Due to some morphological and anatomical limitations of the human eye, it is not possible for us to see clearly in the dark or under low light conditions. In contrast, many nocturnal and deep sea animals have far better night vision than humans, due to differences in the biological structure of their eyes. Night vision refers to the ability to see under low light conditions. Night vision is made possible either through biological or technological means. With the invention of night vision technology it has become possible for humans to see in the dark. Although this technology was originally developed for military use, it also has applications in surveillance, security, wildlife observation and navigation. In the area of intelligent video surveillance, night vision can be used as a tool that can reveal objects even under low light conditions which enables us to detect hidden objects.

There are many approaches to implementing night vision. In this chapter some basic concepts behind night vision and how it has been implemented traditionally are discussed. Also, some of the fundamentals of digital images and digital image processing techniques which can be applied to enhance images for the purpose of night vision are introduced.

2.1 Light Wave

To understand night vision, a basic understanding of light and the electromagnetic spectrum is needed. As charged particles having wave-like behavior travel through space they absorb and emit energy in the form of electromagnetic radiation known as light. In 1666, Sir Isaac Newton discovered that when a beam of sunlight is passed through a glass prism, the emerging beam of light is not white but consists instead of a continuous spectrum of colors ranging from violet at one end to red at the other[7]. The range of colors in visible light represents only a tiny portion of the electromagnetic spectrum. Toward the lower end of the spectrum are radio waves which have wavelengths billions

of times longer than those of visible light. Toward the higher end are gamma rays with wavelengths millions of times smaller than those of visible light.

The energy of the electromagnetic radiation depends on its wavelength and frequency. Wavelength is the distance between two consecutive crests or troughs of the wave. Frequency is the number of waves per second. Shorter wavelengths have higher frequency. Among the colors in visible light, violet has the highest frequency and red the lowest. Fig. 2.1 shows an example of a sinusoidal wave highlighting the wavelength.

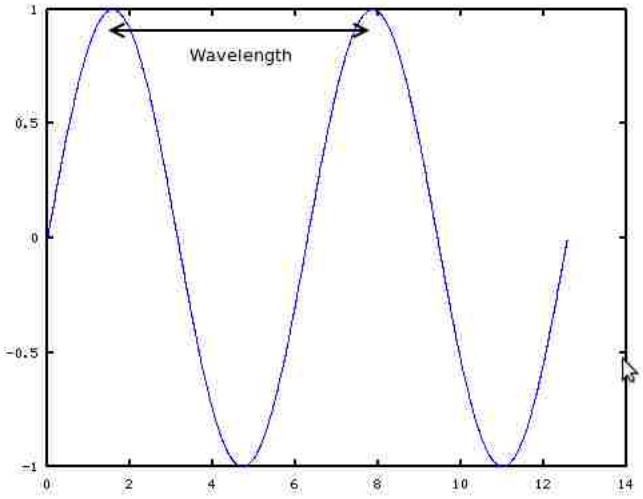

Figure 2.1: A sinusoidal wave

2.1.1 The Infrared Spectrum

Just next to the visible light spectrum and having a longer wavelength is the infrared spectrum. Parts of this infrared spectrum can be used for night vision. The infrared spectrum can be split further into five categories.

1. Near-infrared (NIR). This is closest to visible light and has wavelengths that range from 0.7 to 1.4 micrometers. Image intensifiers are sensitive to this area of the spectrum and this kind of radiation is useful for night vision devices.

2. Short-wavelength infrared (SWIR). These have wavelengths between 1.4 - 3 micrometers.

3. Mid-wavelength infrared (MWIR). These have wavelengths between 3 - 8 micrometers.

4. Long-wavelength infrared (LWIR): These have wavelengths between 8 - 15 micrometers.

5. Far infrared (FIR). These have wavelengths between 15 - 1000 micrometers.

NIR and SWIR are referred to as *reflected infrared* while MWIR and LWIR are referred to as *thermal infrared*. The main difference between the thermal infrared and the reflected infrared is that the thermal infrared is emitted by the object instead of being reflected off it. This makes it possible for sensors to capture the scene based on thermal emissions without any external light or thermal source such as an infrared illuminator.

2.2 Overview of Night Vision Technologies

Night vision has traditionally been made possible by a combination of two approaches - *sufficient spectral range* and *sufficient intensity range*. Spectral range techniques can sense radiation that is invisible to a human observer. It is known that human vision can only perceive a small portion of the electromagnetic spectrum called visible light. Enhanced spectral range allows the viewer to make use of non-visible sources of electromagnetic radiation such as infrared or ultraviolet radiation. Some animals can see better than humans at night using portions of the infrared and/or ultraviolet spectrum. Intensity Range techniques usually make use of an image intensifier technology with low-noise and high sensitivity devices. Based on these approaches, night vision technology can be broadly divided into three main categories.

- Image intensification. Image intensification technologies work on the principle of intensifying the amount of photons received from the available sources of light like starlight or moonlight.
- Active illumination. Active illumination technologies work on the principle of combining imaging intensification technology with an active source of illumination in the near infrared (NIR) or shortwave infrared (SWIR) region of the electromagnetic spectrum.
- Thermal imaging. Thermal imaging technologies work on the principle of detecting the temperature difference between the background and the foreground objects. It captures the portion of the infrared light spectrum which is emitted as heat by objects. Hotter objects emit more of this radiation than cooler objects.

2.2.1 Image Intensification

This method of night vision has been heavily researched by Russia and the United States particularly for military purposes. Image intensification works by magnifying the available light to achieve

better vision. Based on the type of image intensifier tube used, this technology is usually classified into generations 1, 2 and 3. Fig. 2.2 shows the structure of a Gen-1 image intensifier. In a *Gen-1* device, an objective lens focuses whatever available light (photons) there is on the photocathode of an image intensifier tube. The light energy causes electrons to be released from the cathode which are accelerated by an electric field to increase their speed (energy level). The electrons are then made to strike a phosphor screen. The energy of the electrons makes the phosphor glow. This process is repeated multiple times to drastically increase the intensification of the image. The visual light shows the desired view to the user or to an attached photographic camera or video device. Strictly speaking, the light itself is not *amplified*. However, the image is said to become *intensified* because the output visible light is many times brighter than the incoming light. A *Gen-2* tube follows the same general principle as that of the Gen-1 tube, but it differs from Gen-1 tube by employing a special electron amplifier - the micro channel plate (MCP) instead of the multi-stage magnification. The electrons enter holes in this microchannel plate and bounce off the internal specially-coated walls which generate more electrons as the electrons bounce through. This creates a denser cloud of electrons representing an extremely intensified version of the original image. A Gen-3 tube differs from a Gen-2 by the use of a photo cathode based on Gallium Arsenate which has greater sensitivity[8]. Fig. 2.3 shows the structure of a Gen-2 image intensifier. A green phosphor is used in these applications because the human eye can differentiate more shades of green than any other color, allowing for greater differentiation of objects in the picture. A drawback of this technique is that it is not useful when there is absolutely no light because it works on the principle of *intensifying* the available light. This technology is used in Night Vision Devices (NVDs) and Night Vision Goggles.

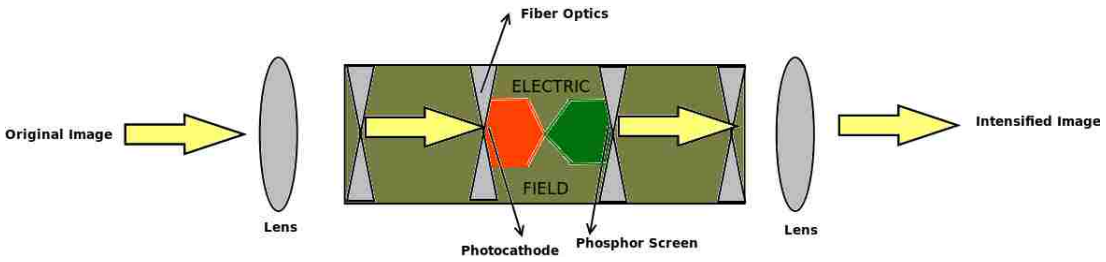

Figure 2.2: A Gen-1 multistage image intensifier.

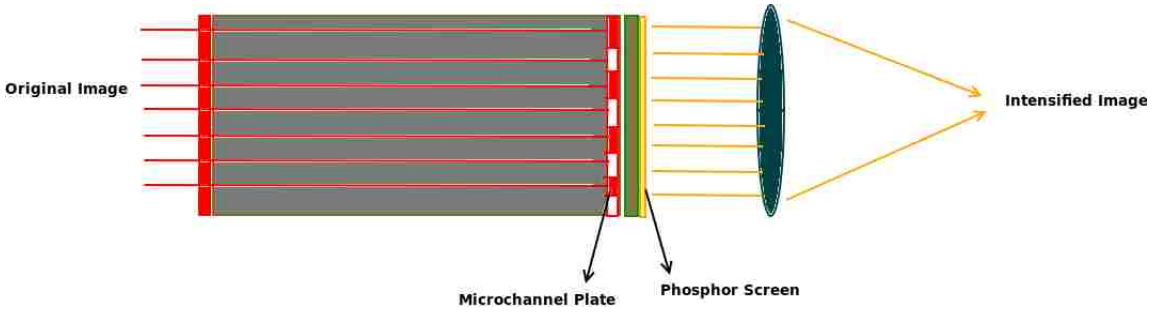

Figure 2.3: A Gen-2 MCP image intensifier.

2.2.2 Active Illumination

Active infrared night vision combines infrared illumination in the NIR or SWIR range with the use of CCD cameras that are sensitive to this light. The resulting scene which would have appeared dark to the naked human eye, then appears as a monochrome image on a display device. Active infrared night vision is now commonly found in many security applications for both commercial and government purposes, where it enables effective night time imaging under poor visibility conditions. A major disadvantage of this kind of technology is that active infrared light can be detected by night vision goggles. Hence, in certain applications like in tactical military operations there can be a risk of being detected and giving away position.

Laser range gated imaging is another form of active illumination night vision technology which make use of a high powered pulsed source of illumination which is also used for imaging. A pulsed laser is used to illuminate the scene while the reflected light is detected by a camera with a short exposure time referred to as a gate. The gate is delayed so imaging occurs at a particular range, thus the image is only from the reflection of objects at that range[9]. Range-gated imaging employs laser sources in the SWIR band. Traditional active illumination technology may not work well under conditions such as dust, smoke, or fog due to the backscattered light. The range-gated imaging system removes these reflections and allows imaging at the desired range only.

2.2.3 Thermal Imaging

Thermal imaging is another technique for night vision. It detects thermal radiation using the thermal infrared portion of the IR spectrum without the need for additional illumination. It operates on the concept that all objects emit infrared energy depending upon their temperature. In general, the hotter an object is, the more radiation it emits. A thermal imager is a device that captures

the infrared radiation from objects in the scene and creates an electronic image. They do not even rely on reflected ambient light. Moreover, they are are able to penetrate obscurants such as smoke, fog and haze. Thermal images are normally black and white in nature, where cold objects appear black and hot objects appear white. Some thermal cameras show images in color. This approach provides a great way of better differentiating between objects at different temperatures. They are widely used in security networks, and for night vision on aircrafts, where they are also referred to as *FLIR* (for *forward-looking infrared.*)

2.3 Digital Image Processing

Digital Image processing is a subfield of digital signal processing. It deals with computer algorithms to process digital images[7]. Digital Image processing techniques provide us with a different approach to implementing night vision. Instead of utilizing the infrared light and special camera or devices, image processing algorithms can be used on images captured by ordinary cameras under low light conditions. These algorithms can be implemented in hardware or software.

2.3.1 Digital Image

An image typically refers to the light intensity variations across a two-dimensional plane[10]. A digital image is essentially a two-dimensional array of light-intensity levels, which can be denoted by $f(x, y)$, where the value or amplitude of f at spatial coordinates (x, y) gives the intensity of the image at the point. The intensity is a measure of the relative *brightness* of each point[7]. For a colored image, this value is usually represented by three numbers pertaining to the decomposition of the color into the three primary components red(R), green(G) and blue(B). Any color visible to human eye can be represented in this manner. The intensity of these colors is represented by a number between 0 and 255. For example, white may be encoded as R = 255, G = 255, B = 255 and black as R = 0, G = 0, B = 0. In other words, a color image is a two dimensional array of color values of pixels, each of which is encoded by 3 bytes, representing the three primary colors. This allows the image to contain a total of $256 \times 256 \times 256 = 16.8$ million different colors. This technique is also known as RGB encoding, and is specifically adapted to human vision. For a monochrome (single color) image each pixel is coded by only 1 byte which represents the intensity value of the pixel, i.e., the possible 256 shades of gray between black and white. These discrete intensity shades are usually referred to as the *gray levels*, with black representing the darkest level and white, the brightest level. For night vision, grayscale images are used instead of color images because the clarity of objects in the dark is more important than the color of objects. The range of 0-255 was agreed upon for two reasons. The first is that the human eye is not sensitive enough to differentiate between

more than 256 levels of intensity. Therefore 256 is usually enough quality for human perception. The second reason for the value of 256 is that it is convenient for computer storage. A byte, which is the computers memory unit, can be coded up to 256 values.

2.3.2 Digital Image Resolution

Image resolution either refers to the *Spatial resolution* or the *Gray-level resolution*. Spatial resolution is the smallest discernible detail in an image[7]. Assume a chart with vertical lines of width W and the space between the lines also having width W. A line pair refers to one line plus its adjacent space. Therefore, the width of a line pair is 2W, and there are 1/2W line pairs per unit distance. A common definition of resolution is the smallest number of discernible line pairs per unit distance. Gray-level resolution similarly refers to the smallest discernible change in gray level[7]. Due to hardware considerations, the number of gray levels is usually an integer power of 2, as mentioned in the previous section. The most common number is 8 bits which gives 256 gray levels. When a strict measure of physical resolution relating pixels and the level of detail they resolve in the original scene are not necessary, it can informally be said that an L-level digital image of size M * N as has a spatial resolution of M * N pixels and a gray-level resolution of L levels.

2.3.3 Image sensor technology: CCD and CMOS

An image sensor is a device that converts an optical image into an electronic signal. It is the key device that determines the quality of the camera's picture[11]. Most of the image sensors used in digital cameras today are either CCD (charge coupled device) or CMOS (complementary metal oxide semiconductor). Both types of imagers convert light into electric charge and process it into electronic signals[12]. In a CCD sensor, every pixel's charge is transferred through just one or a very limited number of output nodes to be converted to voltage. It is then buffered, and sent off-chip in the form of an analog signal. Since all of the pixel can be devoted to light capture, the uniformity in output is high which leads to better image quality. In a CMOS sensor, each pixel has its own charge-to-voltage conversion, and additional circuitry for amplification, noise-correction and digitization. The chip's output is in the form of digital bits. These additional functionality reduce the area available for light capture. Since each pixel is doing its own conversion, it is a massively parallel architecture having high total bandwidth for high speed but the uniformity in output is lower. Due to these major differences, CMOS sensors are more suited for devices that require speed and low power consumption, while CCD sensors are more suited when high image quality is required and/or for low light imaging. Since our work deals with images taken under low light conditions, CCD cameras are used instead of CMOS for all our test images.

2.3.4 Light and Image Characteristics

This section describes some light and image characteristics which will be helpful in understanding the implementation of night vision using digital image processing techniques.

- Radiance: It is the total amount of energy that flows from the light source and is usually measured in Watts (W)[7].

- Luminance: It gives a measure of the amount of energy an observer perceives from a light source[7]. For example, light emanating from a source which operates in the far infrared region of the spectrum could have high radiance, i.e., a lot of energy, but a human observer would hardly perceive it, i.e., its luminance would be almost zero. Luminance is measured in lumens (lm). In image processing terms, luminance can be computed as a properly-weighted sum of *linear-light* red, green, and blue primary components[13]. In contemporary video cameras, the coefficients are

$$Y = 0.2126R + 0.7152G + 0.0722B \tag{2.1}$$

- Luma: Closely related to luminance, luma is the properly-weighted sum of *gamma-compressed* red, green, and blue primary components. In the models used by PAL and NTSC systems the coefficients are

$$Y' = 0.299R' + 0.587G' + 0.114B' \tag{2.2}$$

  In the model used by HDTV, luma is computed as

$$Y' = 0.2126R' + 0.7152G' + 0.0722B' \tag{2.3}$$

- Brightness: It is a subjective descriptor of light perception that is practically impossible to measure[7]. It embodies the notion of intensity or the gray-level of a monochrome image and can be thought of as the overall lightness or darkness of an image.

- Contrast: It can be described as the difference in brightness between objects or regions[10]. For example, a white horse standing in a snowy field has *poor* contrast, while a black dog against the same white background has *good* contrast.

2.3.5 Digital Video

If a series of digital images is displayed in rapid succession and at a constant rate it forms a Digital Video. These individual digital images in the video are often referred to as *frames*. Since every frame is an orthogonal bitmap digital image it is made up of a raster of pixels. If it has a width of W pixels and a height of H pixels, it is said that that the frame size is WxH. The rate at which these frames are displayed is measured in *Frames per Second*(*fps*).

2.4 Image Processing Methods

There are two main methods to process an image depending upon the domain in which the image is processed, namely the spatial domain or the frequency domain. The spatial domain means the image plane, and spatial domain techniques work by directly manipulating the pixels in an image. Frequency domain processing techniques are based on modifying the spatial frequency spectrum of the image given by the Fourier transform of the image. It is also possible to combine different methods in both these categories to obtain the desired enhancement.

1. Spatial Domain Methods.

   The spatial domain refers to the aggregate of pixels composing an image, and spatial domain methods are procedures that operate directly on these pixels. Image processing functions in the spatial domain may be expressed as

   $$g(x, y) = T[f(x, y)] \tag{2.4}$$

   where $f(x, y)$ is the input image data, $g(x, y)$ is the processed image data, and T is an operator on f, defined over some neighborhood of (x, y)[7]. Contrast stretching and histogram equalization are examples of spatial domain techniques.

2. Frequency Domain Methods.

   Frequency domain methods are based on transforming the image to its frequency representation, performing image processing on the frequency representation and then computing the inverse transform back to the spatial domain. The foundation of frequency domain techniques is the convolution theorem. The processed image, $g(x, y)$, is formed by the convolution of an image $f(x, y)$ and a linear, position-invariant operation $h(x, y)$ as

   $$g(x, y) = h(x, y) * f(x, y) \tag{2.5}$$

By the convolution theorem, the following frequency domain relation holds

$$G(u,v) = H(u,v)F(u,v) \tag{2.6}$$

where G, H, and F are the Fourier transforms of g, h and f respectively. $H(u,v)$ is called the transfer function of the process. In a typical image enhancement application, $f(x,y)$ is given and the goal, after computing $F(u,v)$, is to select a $H(u,v)$ so that the desired image $g(x,y)$ exhibits some highlighted feature of $f(x,y)$, i.e.

$$g(x,y) = F^{-1}[H(u,v)F(u,v)] \tag{2.7}$$

Examples of frequency domain methods include highpass and lowpass filters.

Image processing methods can also be classified into *Global* and *Local* methods.

1. Global Methods.

   Image processing methods that use a single transformation operation on the image as a whole are known as global methods. The main advantage of global methods is that they are computationally efficient and relatively simple to implement. The disadvantage is that global methods do not make effective use of local information while working on the overall characteristic of the image which in some scenarios may lead to inferior results. Examples of Global methods include lowpass and highpass filters and histogram transformations.

2. Local Methods.

   A local image processing method is one in which the transformation operation is dependent on the location and the neighborhood of the pixel looked at. These methods are *adaptive* to the local information within the image. Examples of such methods include Adaptive histogram equalization techniques which are effective in enhancing details in local areas of the image. However, because each transformation is done locally and independently it can lead to abrupt changes and boundaries appearing in the image and can lead to artificial appearance of the original image. Also, local image processing methods are relatively more computationally expensive.

2.4.1 Linear and Nonlinear operations

Let H be an operator whose input and output are images. H is said to be a linear operator if, for any two images f and g and any two scalars a and b.

$$H(af + bg) = aH(f) + bH(g) \tag{2.8}$$

In other words, the result of applying a linear operator to the sum of two images (that have been multiplied by the constants shown) is identical to applying the operator to the images individually, multiplying the results by the appropriate constants, and then adding those results[7]. An operator that fails the test of equation.(2.8) is said to be a nonlinear operator. For example, an operator that computes the absolute value of the difference of two images is a nonlinear operator.

2.4.2 Image Histograms

The histogram of a digital image with gray levels in the range $[0, L-1]$ is a discrete function $h(r_k) = n_k$, where r_k is the kth gray level and n_k is the number of pixels in the image having gray level r_k. It is common practice to normalize a histogram by dividing each of its values by the total number of pixels in the image, denoted by n. Thus, a normalized histogram is given by $p(r_k) = n_k/n$ for $k = 0, 1, 2, ..., L-1$[7]. $p(r_k)$ provides an approximate estimation of the probability of occurrence of gray level r_k . The sum of all components of a normalized histogram is equal to 1. Histograms form the basis for a number of spatial domain processing techniques. Histogram manipulation can be used effectively for image enhancement and it also forms the basis for our work on night vision. Histograms are simple and efficient to calculate in both software and hardware implementations which is why they are used extensively for real-time image processing.

Fig. 2.4 shows an example of a dark image and its corresponding histogram. The histogram plot is simply the plot corresponding to the gray level values r_k along the horizontal axis and the $h(r_k) = n_k$ values along the vertical axis. Notice how the values are all lumped together towards the extreme left end of the horizontal axis. Such a histogram is typical of images taken under low light conditions. It indicates that almost all pixels in the image have very low intensity values.

2.5 Image Processing Techniques for Night Vision

Most image processing techniques used for night vision are based on adjusting the contrast of the image to make objects in the scene more easily detectable. These include linear contrast stretching[7], histogram equalizaton[7], wavelet based contrast enhancement[14], retinex[15] and gamma correction[2]. There are many variants of the conventional histogram equalization techniques in use([16], [4], [5]). They can broadly be classified into three categories, namely, *Global*, *Adaptive*

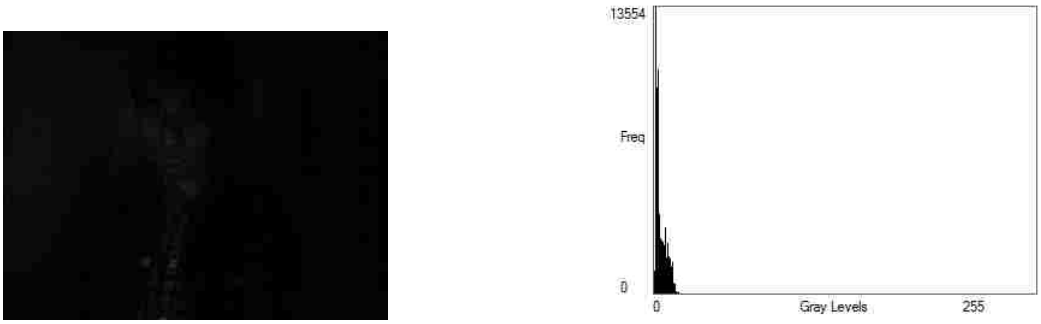

Figure 2.4: A dark image and its corresponding histogram.

and *Block-based*. Global histogram equalization refers to the conventional histogram equalization approach in which the equalization operation is applied to the image as a whole. This is a very efficient and popular approach which is suitable for most images taken under low light. However, this approach may not produce effective results if the illumination in the original scene is uneven. Adaptive histogram equalization first separates the image into small blocks and then applies the conventional histogram equalization technique on each of these blocks. It is a local image processing method. However, this technique can still lead to discontinuities between blocks and noise amplification. Block based histogram equalization basically uses the same approach as the Adaptive version but in this case the blocks are overlapping whereas in Adaptive it was non-overlapping. This is done to solve the problem of block discontinuities. An advantage of global histogram equalization over the adaptive and block based approaches is that it is very efficient to compute and can be applied to real-time video without making any major changes in the algorithm.

The theory behind some of the above mentioned techniques is explained in the next few sections.

2.5.1 Contrast Stretching

Contrast stretching is a typically a linear transformation which can be applied to images taken under poor illumination conditions. The basic idea behind it is to increase the dynamic range of the gray levels in the image being processed[7]. Contrast stretching is also referred to as *normalization*. This operation works as follows. Initially, the upper and lower pixel value limits over which the image is to be normalized is specified. Usually these limits are the minimum and maximum possible pixel values of the original image. For example, for an 8-bit graylevel image the lower and upper limits might be 0 and 255. Let n represent the lower limit and m the upper limit. The normalization procedure then reads every pixel in the image to find the lowest and highest pixel values currently

present. Say these are called l and u respectively. Then each pixel P is scaled using the following function:

$$P_{out} = (P_{in} - l)(\frac{m-n}{u-l}) + n \tag{2.9}$$

Fig. 2.5 shows the result of the application of linear contrast stretching to the image in Fig. 2.4. Observe that the object in the image is now more easily visible and the corresponding histogram is slightly stretched.

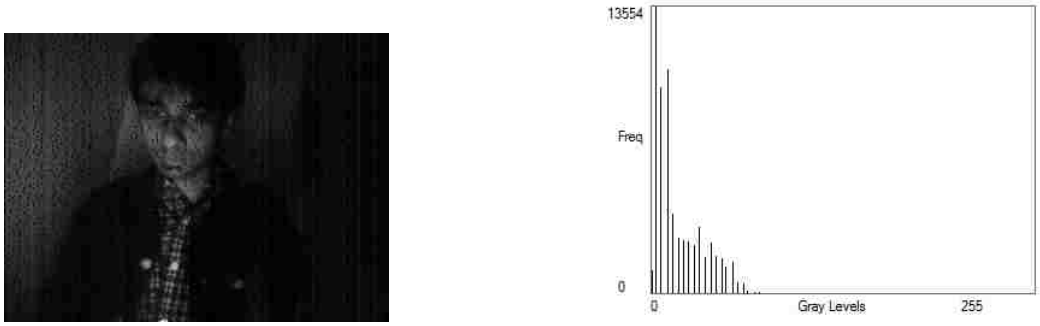

Figure 2.5: A contrast stretched image and its corresponding histogram.

Contrast stretching does not significantly distort relative graylevel intensities and does not increase the contrast too dramatically. Due to this the level of contrast increase may not be sufficient for proper night vision.

2.5.2 Nonlinear Contrast Enhancement

Nonlinear contrast enhancement is a nonlinear operation to adjust the contrast of an image. It typically involves an operation called histogram equalization. This histogram equalization can be of different types as mentioned in section. 2.5. The theory behind this technique is explained in the next section.

2.5.3 Histogram Equalization

Histogram Equalization is an efficient image processing algorithm for contrast adjustment. The histogram equalization transform works by spreading the histogram of the input image uniformly so that the intensity levels of the histogram-equalized image span a larger range of the gray scale. An

image in which the pixels tend to occupy the full range of possible gray levels in a uniform manner has an appearance of high contrast. Such an image shows a great deal of gray-level detail and has high dynamic range. Therefore, histogram equalization has proven to be a very effective tool for night vision.

The probability, p_r of occurrence of gray level r_k in an image is approximated by

$$p_r(r_k) = \frac{n_k}{n}, \quad k = 0, 1, 2, ..., L-1 \tag{2.10}$$

where, n is the total number of pixels in the image, n_k is the number of pixels that have gray level r_k, and L is the total number of possible gray levels in the image (which is 256 for the 8-bit encoded grayscale image). Then, the histogram equalization transformation can be expressed as

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}, \quad k = 0, 1, 2, ..., L-1 \tag{2.11}$$

where, s_k is the level of the output pixel obtained after applying the histogram equalization operation T on input pixel with level r_k. Thus, a processed (output) image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k in the output image[7].

Fig. 2.6 shows the result of the application of global histogram equalization to the image in Fig. 2.4. The original image is on the left and the equalized image is on the right. Notice that the object can now clearly be seen in the image. However, there is an amplification of noise in the image. It is a known fact that global histogram equalization can alter the brightness of the image in a way which may not correspond well with human perception of a good quality image[6]. Fig. 2.7 shows the comparison of histograms between the two images. Observe how the histogram on the right now spans the entire range of the grayscale from 0 to 255.

In the next chapter, it is shown that equalizing the image to include graylevels upto the maximum, i.e., 255 may not always produce optimum results in terms of Peak Signal to Noise ratio. Based on this observation, our own approach to histogram equalization is developed and explained in detail.

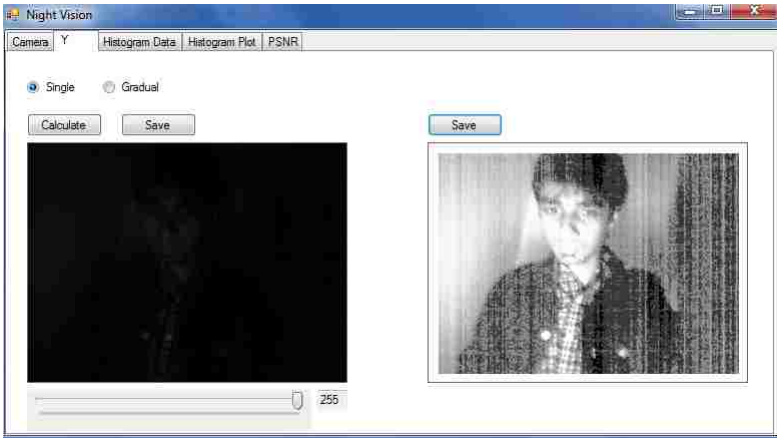

Figure 2.6: An example of Histogram Equalization

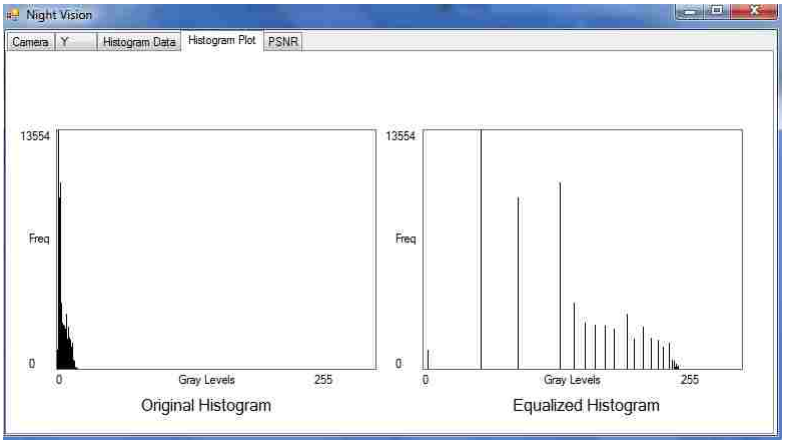

Figure 2.7: Histogram Comparison

2.6 Network programming fundamentals

Our work on image processing for night vision is applied to a video surveillance system. Therefore, it is necessary to understand the basics of network programming which enables us to transmit individual image frames data which are captured at one location to one or more viewers at different remote locations.

2.6.1 The Client Server Network Model

One of the most commonly used models used in computer networking is the *Client Server Model*. In this model the *server* is usually a centrally housed powerful machine which does the bulk of all the processing tasks and/or database storage and is maintained by a system administrator. The *clients*

are usually less powerful machines which access data or request some services from the server. The clients and servers are connected together by a *network*. For example, this network maybe a wired Local Area Network (LAN) of a company in a single building or it can involve a vast number of machines and routers with both wireless and wired communication as in the case of the Internet. The client-server model typically involves two processes, one on the client machine and one on the server machine as shown in Fig. 2.8. Communication takes place by the client process sending a message over the network to the server process. The client process then waits for a reply message. When the server process gets the request, it performs the requested work or looks up the requested data and sends back a reply.

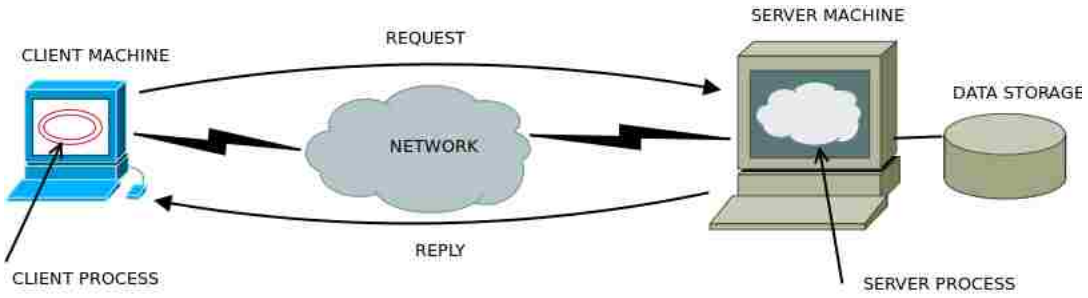

Figure 2.8: Requests and Replies in a Client Server Model

2.6.2 Protocol Hierarchies

In order to minimize the complexity of their design, networks are organized as a stack of layers or levels where each level is built upon the one below it. The number of layers, their names, contents and function vary from network to network. Each layer offers certain services to the higher layers but hides from them the details of how these services are actually implemented. This is a fundamental idea to manage complexity used in computer science in which a particular piece of software or hardware provides a service to its users but keeps the details of its internal state and algorithms hidden from them. The advantage of this approach is that the developer can change the implementation while keeping the interface visible to the outside world consistent. Virtually, layer n on one machine communicates with layer n on another machine. The rules used in this communication are collectively known as the layer n protocol. Basically, a protocol is an agreement between the layers on how communication is to proceed[17]. No data is actually transferred directly from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information

to the layer immediately below it and this process continues until the lowest layer is reached. After the lowest layer is reached this information is passed to the physical medium through which actual communication occurs.

A set of layers and protocols is called a *network architecture*[17]. The specification of an architecture must be detailed enough so that an implementer can write the software or build the hardware for each layer which obeys the appropriate protocol correctly. The architecture does not include the details of the implementation or the specification of the interfaces to be used. These are present within the machines and not visible to the outside. It is also not necessary that the interfaces on all machines in a network be the same. It is sufficient that each machine can correctly use all the protocols no matter how they are implemented. A list of layered protocols used by a certain system is called a protocol stack[17].

Two important network architectures are the OSI reference model and the TCP/IP reference model. Fig. 2.9 shows the layers used in these models. Although the protocols associated with the OSI model are rarely used any more, the model itself acts as a valid and general model for designing other network architectures. The TCP/IP model in itself is not of much use but the protocols associated with TCP/IP are widely used.

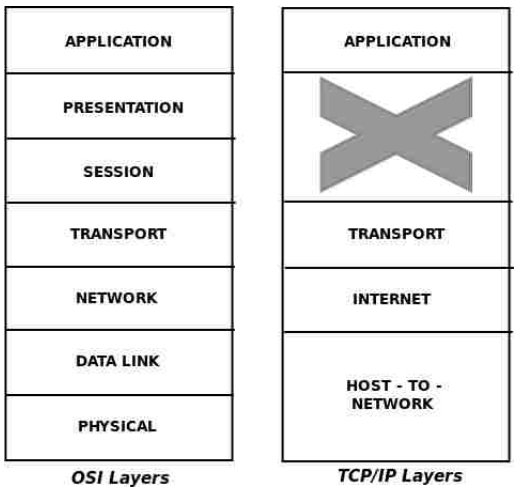

Figure 2.9: The OSI and TCP/IP Reference Models

Each layer of these network models will not be discussed in detail as network programming mainly

deals with the protocols in the *Transport Layer*. A brief summary of the layers is given below. For details please refer to [17].

1. Physical Layer: The physical layer is concerned with transmitting raw bits over a communication channel. This includes dealing with the mechanical, electrical, and timing interfaces and the physical transmission medium.

2. Data Link Layer: The data link layer is concerned with flow regulation and error handling. It usually accomplishes this task by making the sender break up the input data into data frames which may be from a few hundred to few thousand bytes and transmit the frames in sequential order.

3. Network Layer: The network layer is mainly concerned with the *routing* of data packets from source to destination.

4. Transport Layer: The basic function of the transport layer is break the data accepted from the upper layers into smaller units, pass them to the network layer, and ensure that the units arrive correctly and in order at the other end. It must ensure efficiency and must be able to isolate the upper layers in such a way that they are not affected by the changes in hardware technology. The protocols of the transport layer will be discussed in the next subsection (2.6.3) as it forms the basis for networking programming.

5. Application Layer: The application layer contains a variety of protocols that are commonly needed by users. These include the widely used HTTP, FTP and SMTP protocols.

2.6.3 Transport Layer Protocols - TCP and UDP

A connection-oriented service is one in which the service user first establishes a connection, uses the connection, and then releases the connection. The service guarantees that the receiver gets the data in the order in which they were sent. In contrast, a connectionless service is one in which each message carries the full destination address, and each one is routed through the system independent of all the others. Hence, the service itself does not guarantee that the message will arrive in the same order in which it was sent.

A reliable service is one that ensures that messages are not lost. It is usually implemented by having the receiver acknowledge the receipt of each message so the sender knows that it arrived correctly. The acknowledgement process introduces overhead and delays, which are strictly required for some

applications but are sometimes undesirable for other applications. On the other hand, an unreliable service is one in which the receiver does not acknowledge the receipt of messages and hence it does not guarantee that each message sent is received.

Two of the most widely used transport layer protocols are TCP and UDP. They differ from each other in terms of quality of service.

TCP (Transmission Control Protocol), is a *reliable connection-oriented protocol* that delivers a byte stream from one machine to another machine in the internet without error. It breaks the incoming byte stream into discrete messages and sends each one on to the internet layer. At the destination, the receiving TCP process reconstructs the received messages and creates the output stream. TCP also performs flow control to make sure a fast sender cannot swamp a slow receiver by sending more messages than it can handle.

UDP (User Datagram Protocol), is an *unreliable connectionless protocol* for applications that do not require the services provided by TCP like sequencing or flow control and/or wish to provide their own implementation of such services. It is also well suited for one-shot, client-server based request-reply queries and for applications in which prompt delivery is more critical than accurate delivery, such as transmitting speech or video. Since our work deals with the transmission of video frames, the protocol of choice is UDP.

2.6.4 Socket Primitives

Communication across a network works by processes on different computers communicating with each other. A network socket is an endpoint of such a communication. Currently, most communication between computers are Internet Protocol based. Therefore, most network sockets are Internet sockets. Usually, the operating system provides an application programming interface (API) that allows application programs to use network sockets. Such an API is called a socket API. Internet socket APIs are usually based on the Berkeley sockets standard. The combination of an IP address and a port number forms a socket address. Based on this address, internet sockets deliver incoming data packets to the proper application process or thread. The Internet Assigned Numbers Authority (Iana) defines port numbers for common services[18]. Services not on the Iana list can have port numbers in the range 1,024 to 65,535. IP version 4 addresses use 32 bits to represent a network address. For class C addresses using a subnet mask of 255.255.255.0, these bits are separated into four octets. These four octets when expressed in decimal form the commonly used dotted-quad no-

tation, such as 192.168.102.3. The first two octets (192.168 in this example) represents the network number, the third octet (102) defines the subnet, and the final octet (3) denotes the host identifier[19].

As mentioned earlier in subsection (2.6.3), the UDP protocol is used for network programming in our work. The typical socket primitives associated with UDP are

- SOCKET: It is used to create a new end point and allocate table space for it specifying the addressing format to be used, the type of service desired and the protocol.
- BIND: It is used to assign a network address to the socket.
- SEND: It is used for sending some data across the network.
- RECEIVE: It is used for receiving some data from the network.

These primitives are used to send and receive UDP datagrams to and from the network.

2.6.5 Buffering and Flow Control

In network data transmission there is always a possibility that a fast sender can swamp a slow receiver with more data than it can handle. For video frames this could mean frames getting garbled at the receiver's end. Conversely, in the case of a slow sender and fast receiver it can lead to extremely low frame rates at the receiver. Since UDP offers a connectionless service, it does not ensure by itself that the data sent from the server will arrive at the client in the same order. It also does not provide any flow control. Hence, it is upto the programmer to implement the flow control of the data via buffering. The video frames data need to be buffered at both the sender and receiver's end to ensure proper flow control. Our implementation to solve this issue is explained in detail in the next chapter.

Chapter 3

Implementation and Results

This chapter describes the implementation details of the histogram equalization algorithm for night vision and compares the results of the incremental approach of histogram equalization to the conventional one. It also describes the network programming architecture used for video transmission. All test images used for the analysis were taken using a Logitech 2 Megapixels CCD camera with a Carl Zeiss Tessar design lens of focal length 3.7mm and maximum aperture of f/2.0 where f is a reference to the f-number or focal ratio of the optical system. For details on optical systems refer to [20]. The software was implemented in C# using Microsoft Visual Studio 2008.

3.1 Histogram Equalization implementation

Our implementation of night vision is based on an image processing algorithm known as global histogram equalization. The theory behind histogram equalization was given in section (2.5.3). Algorithm. 3.1 gives an outline of the algorithm. The algorithm first converts the RGB input image to a grayscale image. It then finds the minimum and maximum intensity levels in the grayscale image and computes the histogram of the image. Then, for each intensity level ranging from the minimum intensity in the grayscale image to the maximum intensity, the cumulative frequency and cumulative probability are computed. The new intensity for each of these intensity levels is calculated by multiplying the cumulative probability at that level with the maximum possible intensity of 255. Later, it is shown that this approach, i.e., equalizing the image histogram to span the entire grayscale range [0-255] does not lead to the best results in terms of output image quality for night vision and an analysis is performed to find the optimum intensity which will lead to better output image quality.

Fig. 3.1 shows the result of histogram equalization. A simple visual inspection of the original image (on the left) taken under low illumination and the equalized image (on the right) clearly shows that the face can be seen quite distinctly in the equalized image but not in the original image. However, the amplification of striped noise is observed in the equalized image. Fig. 3.2 shows the comparison

Algorithm 3.1 Algorithm for Histogram Equalization

$frequency[] \leftarrow 0$
$cumFrequency[] \leftarrow 0$
$cumProbability[] \leftarrow 0$
$newIntensity[] \leftarrow 0$
$Intensity_{min} \leftarrow 255$
$Intensity_{max} \leftarrow 0$

▷ For each pixel in input image, convert RGB to grayscale.
▷ Calculate Minimum and Maximum Grayscale intensity and find histogram of image.
for $x = MIN_X \rightarrow MAX_X$ **do**
  for $y = MIN_Y \rightarrow MAX_Y$ **do**
    $c \leftarrow Image_{inp}.GetPixel(x, y)$
    $Y \leftarrow (c.Red * 0.3) + (c.Green * 0.59) + (c.Blue * 0.11)$
    $frequency[Y] \leftarrow frequency[Y] + 1$
    if $Y \leq Intensity_{min}$ **then**
      $Intensity_{min} \leftarrow Y$
    end if
    if $Y \geq Intensity_{max}$ **then**
      $Intensity_{max} \leftarrow Y$
    end if
  end for
end for

▷ Calculate cumulative frequency and probability.
▷ Calculate equalized intensity using the cumulative probabilty.
for $i = Intensity_{min} \rightarrow Intensity_{max}$ **do**
  $cumFrequency[i] \leftarrow cumFrequency[i-1] + frequency[i]$
  $cumProbability[i] \leftarrow cumFrequency[i]/TOTALPIXELS$
  $newIntensity[i] \leftarrow cumProbability[i] * 255$
end for

▷ Create the output histogram equalized image using the new intensity.
for $x = MIN_X \rightarrow MAX_X$ **do**
  for $y = MIN_Y \rightarrow MAX_Y$ **do**
    $c \leftarrow Image_{inp}.GetPixel(x, y)$
    $Y \leftarrow (c.Red * 0.3) + (c.Green * 0.59) + (c.Blue * 0.11)$
    $Image_{out}.SetPixel(x, y, newIntensity[Y])$
  end for
end for

of histograms of the two images. It can be seen that in the original image all the graylevels of the image occupy a very small range to the extreme left of the grayscale which is why the image is very dark while in the equalized image the range of graylevels has increased to span the entire range [0-255] which leads to enhancement in the contrast of the image.

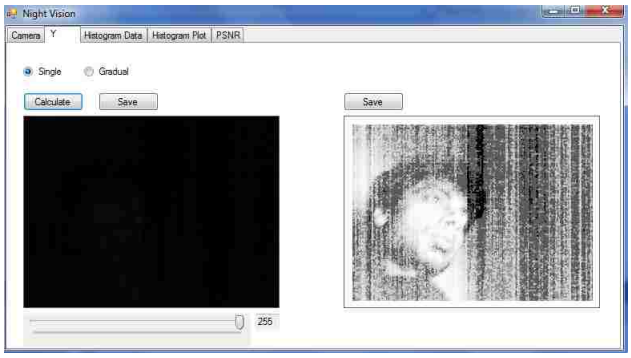

Figure 3.1: Global Histogram Equalization under low illumination

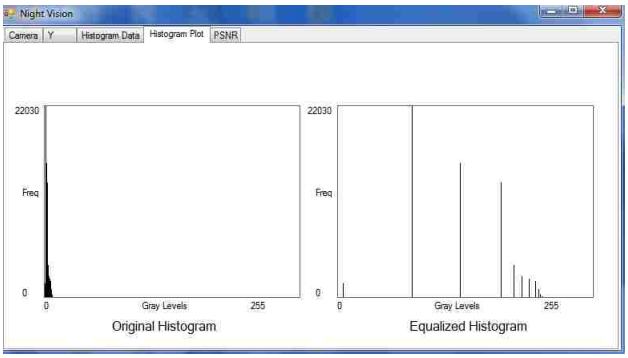

Figure 3.2: Histogram comparison

3.2 Peak Signal to Noise Ratio

In the following sections an assessment of the results of histogram equalization at different equalization intensity values is made. This thesis claims that there exists an optimum intensity which gives the best results in terms of output image quality. It is not sufficient to perform this analysis simply on the basis of visual inspection since it is a subjective measure and may vary from person to person. Therefore, in order to make an objective evaluation of the output image, the metric Peak

Signal to Noise ratio (PSNR) is used.

Peak Signal to Noise Ratio or PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the quality of its representation. Because many signals have a very wide dynamic range, (ratio between the largest and smallest possible values of a changeable quantity) the PSNR is usually expressed in terms of the logarithmic decibel scale[21]. In relation to image processing, the PSNR can be used to systematically compare algorithms by running them on the same set of test images to identify which produces better results. The mathematical representation of the PSNR is as follows

$$PSNR = 20 \log \left(\frac{MAX_f}{\sqrt{MSE}}\right) \tag{3.1}$$

where MAX_f is the maximum possible signal value, which is 255 for an 8 bit monochrome image, and MSE is the mean squared error given by

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [f(i,j) - g(i,j)]^2 \tag{3.2}$$

where f represents the matrix data of the original or "known to be good" image, g represents the matrix data of the degraded image in question (in our case, this is the histogram equalized image), m represents the numbers of rows of pixels of the images and i represents the index of that row, n represents the number of columns of pixels of the image and j represents the index of that column.

3.3 Optimum Intensity for Histogram Equalization

The gray level intensity range for an image taken under low light conditions is typically very small and usually ranges from 0 to 40 or 0 to 60. The basic idea behind histogram equalization is to expand this range. In order to obtain an equalized image of highest possible quality, a test is devised that performs the histogram equalization at each intensity starting from the maximum gray level intensity of the original image upto the maximum possible gray level intensity of 255. At each intensity level an output image is obtained. An objective evaluation of the quality of each of these output images is made using the PSNR metric. In order to do this, for each of our original images taken under low illumination, a corresponding "ideal" image, i.e., a corresponding image

taken under "well illuminated" conditions is required. Although the definition of "well illuminated" can be subjective, care is taken to use an image in which there is no uneven illumination and which in general matches the human perception of an image taken under ideal light conditions. For the PSNR calculation, the equalized image at each intensity level is compared to this ideal image which implies that the obtained PSNR gives a reasonable objective estimate of the quality of the equalized images. Algorithm. 3.2 is an outline for this histogram equalization test procedure. Basically, it is the same algorithm as Algorithm. 3.1 applied to each intensity starting from the maximum intensity of the initial dark image upto the maximum possible intensity of 255. First, the RGB input image is converted to a grayscale image. Then, the minimum and maximum intensity levels in the grayscale image are found and the image histogram is computed. Then, for each intensity level ranging from the minimum intensity in the grayscale image to the maximum intensity, the cumulative frequency and cumulative probability are computed. For each intensity starting from the maximum intensity of the original image upto 255, the new intensities for each intensity level in the original image is calculated and an output image is produced using the new intensities. The PSNR is calculated for each of these output images.

On running this test for several images taken under low illumination, it can be seen that the PSNR follows a general pattern. It increases smoothly starting from the maximum intensity of the original image, eventually reaches a maximum value and then decreases smoothly. The intensity at which the PSNR is maximum gives us the ideal equalized image. These tests support our claim that there exists an optimum intensity value at which the histogram equalization gives better results than at the maximum gray level intensity of 255.

Fig. 3.3(a) shows the image of a book taken under low light conditions and Fig. 3.3(b) its corresponding grayscale image. Fig. 3.3(c) shows the same image taken under ideal light conditions along with Fig. 3.3(d), its corresponding grayscale image. This is used as a control image for calculating the PSNR. The graylevel intensities of the original dark image ranges from 0 to 41. The peak graylevel intensity of the corresponding well illuminated image is 229.

Algorithm 3.2 Algorithm for Gradual Histogram Equalization

$frequency[] \leftarrow 0$
$cumFrequency[] \leftarrow 0$
$cumProbability[] \leftarrow 0$
$Intensity_{min} \leftarrow 255$
$Intensity_{max} \leftarrow 0$

▷ For each pixel in input image, convert RGB to grayscale.
▷ Calculate Minimum and Maximum Grayscale intensity and find histogram of image.
for $x = MIN_X \rightarrow MAX_X$ **do**
  for $y = MIN_Y \rightarrow MAX_Y$ **do**
    $c \leftarrow Image_{inp}.GetPixel(x, y)$
    $Y \leftarrow (c.Red * 0.3) + (c.Green * 0.59) + (c.Blue * 0.11)$
    $frequency[Y] \leftarrow frequency[Y] + 1$
    if $Y \leq Intensity_{min}$ **then**
      $Intensity_{min} \leftarrow Y$
    end if
    if $Y \geq Intensity_{max}$ **then**
      $Intensity_{max} \leftarrow Y$
    end if
  end for
end for

▷ Calculate cumulative frequency and probability.
for $i = Intensity_{min} \rightarrow Intensity_{max}$ **do**
  $cumFrequency[i] \leftarrow cumFrequency[i-1] + frequency[i]$
  $cumProbability[i] \leftarrow cumFrequency[i]/TOTALPIXELS$
end for

▷ Calculate new intensity level for each Equalizing Intensity value
for $eqIntensity = (Intensity_{max} + 1) \rightarrow 255$ **do**
  $newIntensity[] \leftarrow 0$
  for $i = Intensity_{min} \rightarrow Intensity_{max}$ **do**
    $newIntensity[i] \leftarrow cumProbability[i] * eqIntensity$
  end for

▷ Create the output histogram equalized image using the new intensity.
  for $x = MIN_X \rightarrow MAX_X$ **do**
    for $y = MIN_Y \rightarrow MAX_Y$ **do**
      $c \leftarrow Image_{inp}.GetPixel(x, y)$
      $Y \leftarrow (c.Red * 0.3) + (c.Green * 0.59) + (c.Blue * 0.11)$
      $Image_{out}.SetPixel(x, y, newIntensity[Y])$
    end for
  end for
▷ Calculate PSNR using this output image and the ideal image taken under good illumination.
end for

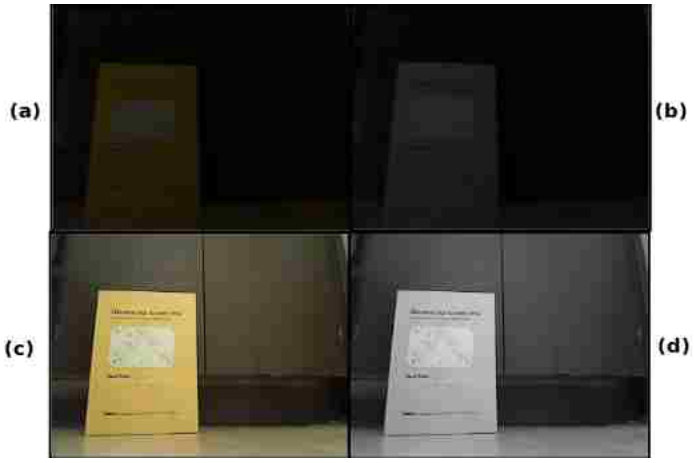

Figure 3.3: Image of book under different conditions:(a)Low light RGB.(b)Low light Grayscale.(c)Well illuminated RGB.(d)Well illuminated Grayscale.

Fig. 3.4 shows the plot of PSNR values obtained after histogram equalization at different intensities. This clearly shows that the PSNR follows a smooth curve starting at a value of 9.363 at intensity 42, peaking at a value of 18.152 at an intensity of 177 and then decreasing gradually. Table. 3.1 shows PSNR values at different intensities. Observe that the PSNR at maximum intensity 255 is only 13.188.

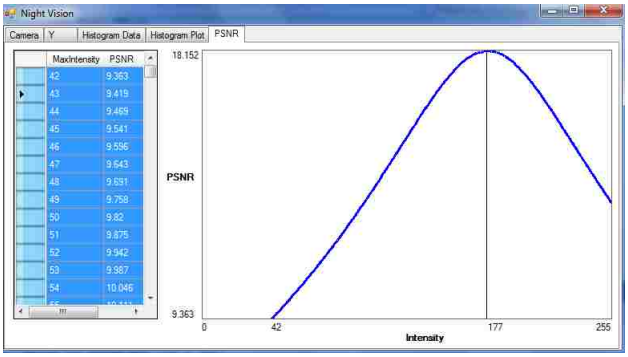

Figure 3.4: PSNR after equalization of book image at different intensities

Fig. 3.5 shows the comparison of the dark image with the equalized image at maximum intensity of 255. Fig. 3.6 shows this same comparison at the peak PSNR intensity of 177. On comparing the equalized images of Fig. 3.5 and Fig. 3.6, by visual inspection, it can be seen that in the case

Intensity	PSNR (dB)
42	9.363
72	11.181
102	13.323
152	17.217
170	18.063
171	18.089
172	18.088
173	18.118
174	18.098
175	18.145
176	18.148
177	18.152
178	18.145
179	18.142
180	18.133
181	18.132
182	18.131
183	18.104
202	17.336
232	15.019
255	13.188

Table 3.1: PSNR values for book image at different intensities

of equalization at intensity 255 there is an uneven brightness in the equalized image. It is known that histogram equalization can lead to uneven brightness and over-enhancement in the produced image([22], [23]) which is quite clear in this case. However, it can be observed that in the case of equalization at intensity 177, the output image has smoother brightness variations and is more comparable to the ideal image of Fig. 3.3(d). Fig. 3.7 and Fig. 3.8 show the histogram comparison after equalization at intensity 255 and 177 respectively.

As another example, consider Fig. 3.9(a) which shows the image of some books and stationery items taken under low illumination and Fig. 3.9(b) its corresponding grayscale image. Fig. 3.9(c) shows the same image taken under well lighted conditions along with Fig. 3.9(d), its corresponding grayscale image. The graylevel intensities of the original dark image ranges from 0 to 23. The peak graylevel intensity of the corresponding well illuminated image is 253.

Histogram equalization is again performed at each intensity level starting from the maximum intensity of original image upto 255. Fig. 3.10 shows the plot of PSNR values obtained. Observe that it follows the same pattern as Fig. 3.4, i.e., it gradually increases, reaches a maximum value and

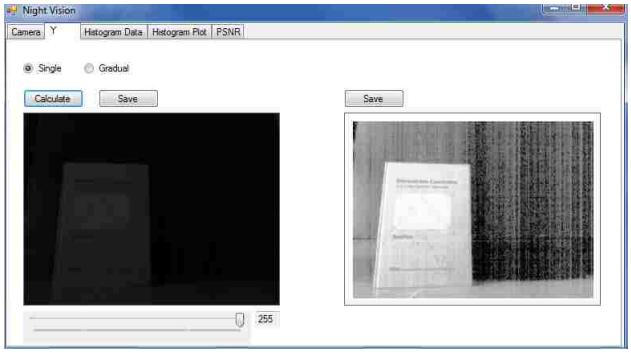

Figure 3.5: Histogram Equalization of book image at Intensity Level 255

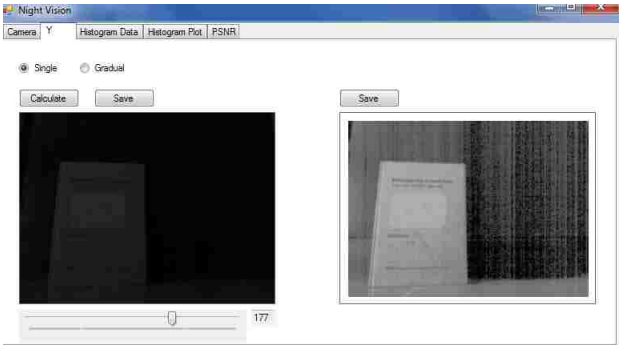

Figure 3.6: Histogram Equalization of book image at Intensity Level 177

then decreases again. The optimum intensity here is 168. Table. 3.2 shows PSNR values at different intensities.

Fig. 3.11 and Fig. 3.12 show the comparison of the original dark image of stationery items with the equalized image at maximum intensity of 255 and optimum intensity of 168 respectively.

3.4 Incremental approach to Histogram Equalization

In the previous section, it was shown that histogram equalization using the maximum possible intensity of 255 does not yield the best results in terms of PSNR of the output image. This was done by performing the conventional histogram equalization at different intensities for several test images taken under poor visibility conditions and comparing the PSNR values. In this section, an incremental approach to histogram equalization is developed. In this approach too, histogram equalization is done starting from the maximum intensity in the original dark image and all the way upto 255. However, for each equalization step, the output image of the previous step acts as

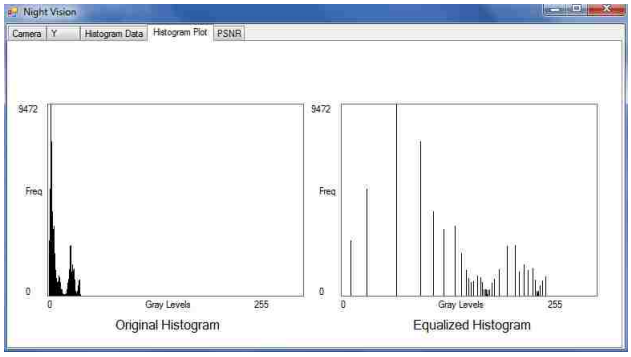

Figure 3.7: Histogram after Equalization of book image at Intensity Level 255

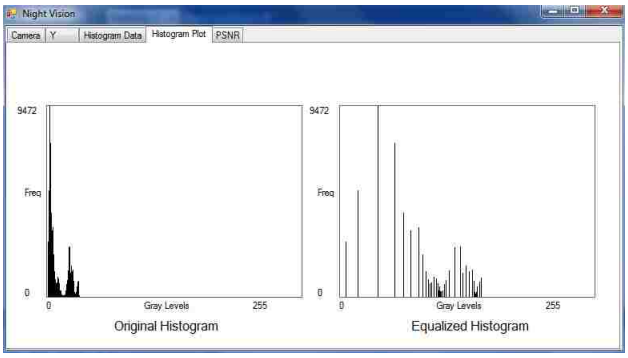

Figure 3.8: Histogram after Equalization of book image at Intensity Level 177

the input. For example, if the dark image has gray level intensities in the range 0 - 20, first an equalization is performed using the original image at gray level intensity of 21. The equalized image obtained is then used as an input image for the next equalization at intensity 22 and so on upto the maximum possible value of 255. Since histogram equalization is a nonlinear operation, better results can be expected using this approach than by using the same dark image for each step. Here again, the PSNR is calculated for each output image obtained at different intensities. The PSNR plot again follows the same general pattern as seen in the previous section and the optimum intensity level (the one at which PSNR is maximum) is obtained somewhere between the maximum intensity level of the original dark image and 255. A comparison is made of the PSNR obtained by the conventional approach and the incremental approach and it is found that the incremental approach yields slightly better results. The outline of the algorithm is given in Algorithm. 3.3. The Histogram Equalization is now presented as a function which takes as its input parameters, an image and a value by which to increment the maximum intensity of that image for the equalization. The output image produced by each histogram equalization step is used as an input for the next step. Refer to Appendix. A for

Figure 3.9: Image of stationery items under different conditions.(a)Low light RGB.(b)Low light Grayscale.(c)Well illuminated RGB.(d)Well illuminated Grayscale.

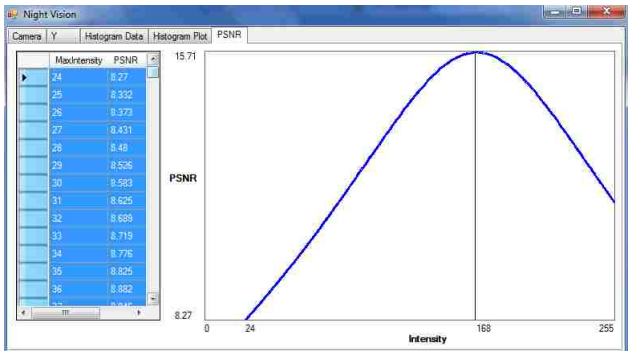

Figure 3.10: PSNR after equalization of stationery image at different intensities

the C# source code for this algorithm.

This algorithm is applied to the stationery items image of Fig. 3.9(b). The PSNR at each step in the equalization process is computed as done previously. Fig. 3.13 shows the PSNR plot for different intensities using the incremental approach.

.

On comparing this PSNR plot with that of the previous approach in Fig. 3.10, it can be seen that the maximum PSNR for the incremental method is 15.735 at intensity 168 which is slightly greater than 15.710 using the conventional method. Table. 3.3 shows the comparison of PSNR values at different intensities for the two methods. PSNR1 represents the PSNR values for the incremental histogram equalization and PSNR2 represents the values for the conventional method. Observe that

Algorithm 3.3 Algorithm for Incremental Histogram Equalization

$THRESHOLD \leftarrow 0$
$Image_{eq} \leftarrow Image_{org}$
while $THRESHOLD < 255$ **do**
  $Image_{eq} \leftarrow$ HistogramEqualize$(Image_{eq}, 1)$
end while

function HistogramEqualize($Image_{inp}$,$incrementVal$)
  $frequency[] \leftarrow 0$
  $cumFrequency[] \leftarrow 0$
  $cumProbability[] \leftarrow 0$
  $Intensity_{min} \leftarrow 255$
  $Intensity_{max} \leftarrow 0$
▷ For each pixel in input image, convert RGB to grayscale.
▷ Calculate Minimum and Maximum Grayscale intensity and find histogram of image.
  for $x = MIN_X \rightarrow MAX_X$ **do**
    for $y = MIN_Y \rightarrow MAX_Y$ **do**
      $c \leftarrow Image_{inp}.GetPixel(x, y)$
      $Y \leftarrow (c.Red * 0.3) + (c.Green * 0.59) + (c.Blue * 0.11)$
      $frequency[Y] \leftarrow frequency[Y] + 1$
      if $Y \leq Intensity_{min}$ **then**
        $Intensity_{min} \leftarrow Y$
      end if
      if $Y \geq Intensity_{max}$ **then**
        $Intensity_{max} \leftarrow Y$
      end if
    end for
  end for
▷ Calculate cumulative frequency and probability.
▷ Calculate new intensity level for the Equalizing Intensity value
  $newIntensity[] \leftarrow 0$
  $eqIntensity \leftarrow Intensity_{max} + incrementVal$
  if $eqIntensity \geq 255$ **then**
    $eqIntensity \leftarrow 255$
  end if
  for $i = Intensity_{min} \rightarrow Intensity_{max}$ **do**
    $cumFrequency[i] \leftarrow cumFrequency[i-1] + frequency[i]$
    $cumProbability[i] \leftarrow cumFrequency[i]/TOTALPIXELS$
    $newIntensity[i] \leftarrow cumProbability[i] * eqIntensity$
  end for
▷ Create the output histogram equalized image using the new intensity.
  for $x = MIN_X \rightarrow MAX_X$ **do**
    for $y = MIN_Y \rightarrow MAX_Y$ **do**
      $c \leftarrow Image_{inp}.GetPixel(x, y)$
      $Y \leftarrow (c.Red * 0.3) + (c.Green * 0.59) + (c.Blue * 0.11)$
      $Image_{out}.SetPixel(x, y, newIntensity[Y])$
    end for
  end for
  $THRESHOLD \leftarrow Intensity_{max}$
▷ Calculate PSNR using this output image and the ideal image taken under good illumination.
  return $Image_{out}$
end function

Intensity	PSNR (dB)
24	8.270
72	10.889
102	12.787
152	15.441
160	15.625
161	15.643
162	15.650
163	15.661
164	15.677
165	15.666
166	15.681
167	15.697
168	15.710
169	15.705
170	15.697
171	15.695
172	15.685
173	15.670
202	14.778
232	13.006
255	11.553

Table 3.2: PSNR values for stationery image at different intensities

the PSNR at intensity 255 is greater for the conventional method than for the incremental one. However, the maximum PSNR which is at intensity 168 for both cases is greater in the incremental approach.

.

Fig. 3.14 shows the stationery image after incremental histogram equalization at different maximum intensities. Observe that as the equalizing intensity increases the image tends to get brighter and the objects become more visible. However, if the images are observed closely for higher intensities at above 200 (see last row of images) it can be seen that the change in brightness is too drastic. According to the calculated PSNR value the central image, i.e., at intensity level 168 gives the image that is most comparable to the ideal well lighted image of Fig. 3.9(d)

.

Fig. 3.15 shows a similar analysis for the book image of Fig. 3.3(b). In this case the third image in the second row at intensity 177 gives the highest PSNR. Notice the drastic change in brightness at intensity 255 (the last image in the third row).

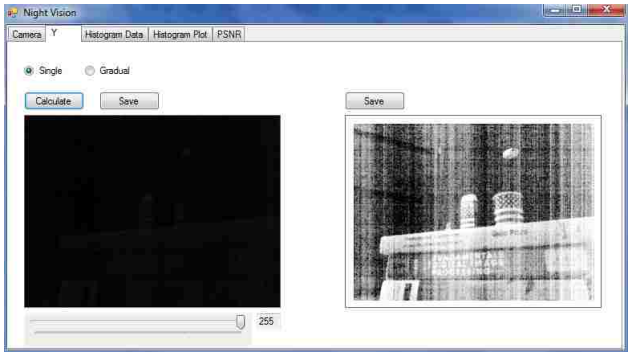

Figure 3.11: Histogram Equalization of stationery image at Intensity Level 255

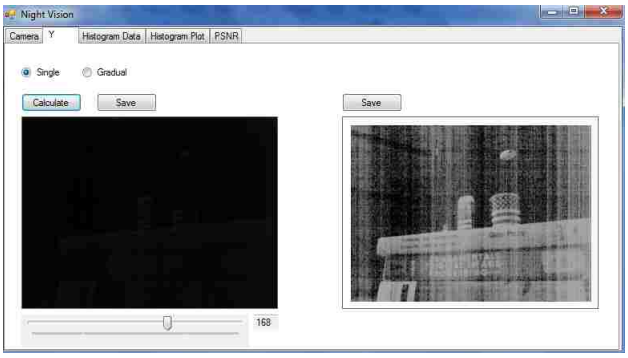

Figure 3.12: Histogram Equalization of stationery image at Intensity Level 168

3.5 Video transmission, Buffering and Flow Control

As our work on histogram equalization for night vision is applied to video surveillance, this section describes the implementation of video capture, recording and transmission over the network.
The network programming basics including the client-server model and UDP data transmission was introduced in the previous chapter. In our implementation a client server network model is used in which the server includes the implementation of video frame capturing, buffering of video frames in files and database storage, the client includes the implementation of viewing the video and night vision. The client can also request the server to send past video frames from the database. At the server end, there is a multithreaded architecture. There are four threads working in parallel, each dedicated to a specific task as follows :

1. Buffering : This thread is dedicated to capturing video frames from the camera, buffering the video frames in files and recording the frames in a database for future use. It captures each frame in a byte array and buffers k of these byte arrays in each video data file. It also saves

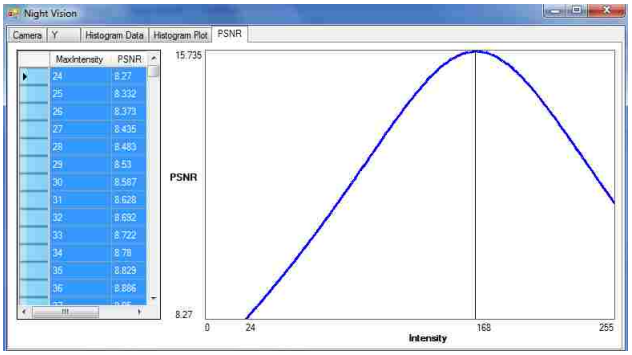

Figure 3.13: PSNR after incremental histogram equalization of stationery image at different intensities

Intensity	PSNR1	PSNR2
24	8.270	8.270
72	10.902	10.889
102	12.807	12.787
152	15.472	15.441
153	15.512	15.482
154	15.528	15.496
155	15.551	15.519
156	15.570	15.538
157	15.603	15.573
158	15.611	15.582
159	15.637	15.608
160	15.657	15.625
161	15.674	15.643
162	15.680	15.650
163	15.690	15.661
164	15.706	15.677
165	15.693	15.666
166	15.708	15.681
167	15.723	15.697
168	15.735	15.710
169	15.729	15.705
170	15.719	15.697
171	15.717	15.695
172	15.706	15.685
173	15.689	15.670
202	14.769	14.778
232	12.977	13.006
255	11.517	11.553

Table 3.3: PSNR values comparison

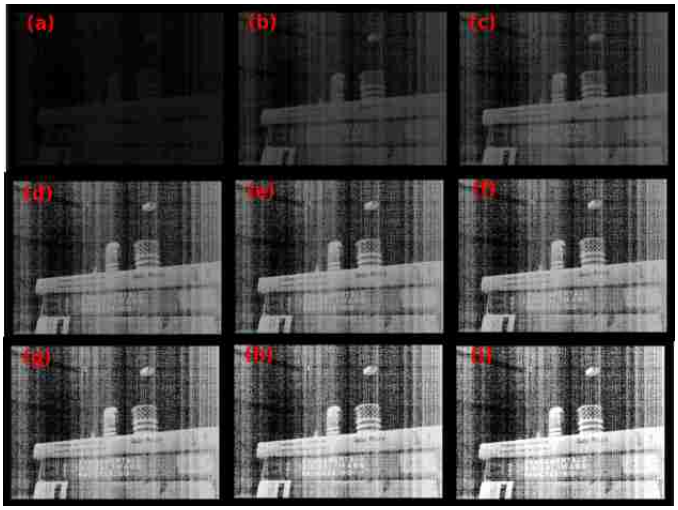

Figure 3.14: Stationery images after incremental histogram equalization at different intensities: (a)24. (b)72. (c)102. (d)152. (e)168. (f)186. (g)202. (h)232. (i)255.

the length of each of these k byte arrays in a corresponding size data file. It follows a circular buffer implementation in which it buffers n files before overwriting the first file. The number of byte arrays (frames) to buffer in each file k and the number of files to buffer before overwriting n can be adjusted based on the speed of the system and network used.

2. Transmitting : This thread is responsible for reading the frames from the frame buffer and sending it to the client over the network. It always transmits if there is atleast one buffered file present in the system. This number can also be adjusted according to the speed of the network used.

3. Listener : This thread is dedicated to listening on a specified port for incoming database data send requests from the client. On receiving such a request it starts the Transmit from database thread to fulfill the client's request.

4. Transmit from database: This thread is responsible for reading the specified frame data out of the database based on the date and time request sent from the client and sending it to the client over the network.

Since the objective here is live data transmission, prompt delivery is needed rather than reliable delivery. Due to this, the UDP protocol is used instead of TCP for all data transmission. UDP and TCP were introduced in the previous chapter. Since UDP does not provide any flow control of its own, it has to be ensured that a slow receiver is not swamped by a fast sender and also that the

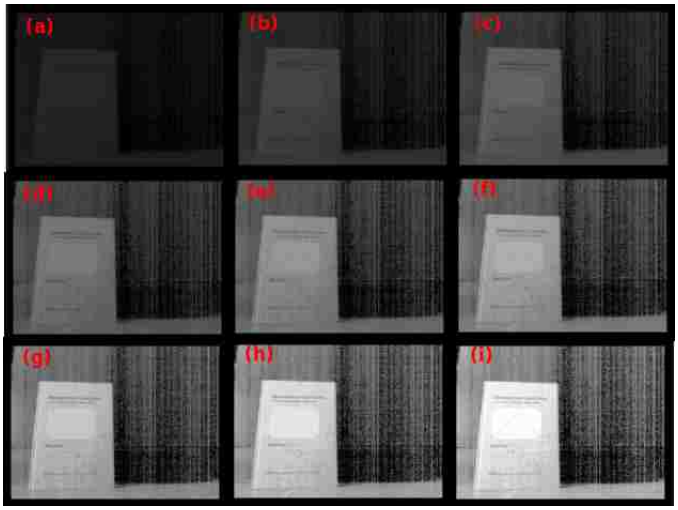

Figure 3.15: Book images after incremental histogram equalization at different intensities: (a)42. (b)72. (c)102. (d)125. (e)152. (f)177. (g)202. (h)232. (i)255.

frames are received in the correct order in which they are sent. This is achieved via proper buffering. Fig. 3.16 shows the diagram of the working of the circular buffer. The buffering and transmitting threads work in parallel.

.

The client end also includes a multithreaded architecture where one thread is dedicated for buffering the frames received from the server into files and another one for reading the frames from the buffer and displaying the video. Similarly, there are another set of two threads in which one is dedicated to buffering frames received from the database of the server and another one to display the video associated with these frames. The buffering threads save each frame in a byte array and buffer k of these byte arrays in each video data file. They also save the length of each of these k byte arrays in a corresponding size data file. Again, a circular buffer implementation is used in which n files are buffered before overwriting the first file. The displaying threads begin reading the frame data out of the buffered files when there are atleast m files present in the buffer. Refer to Appendix. B for the C# source code for circular buffer implementation at the client's end.

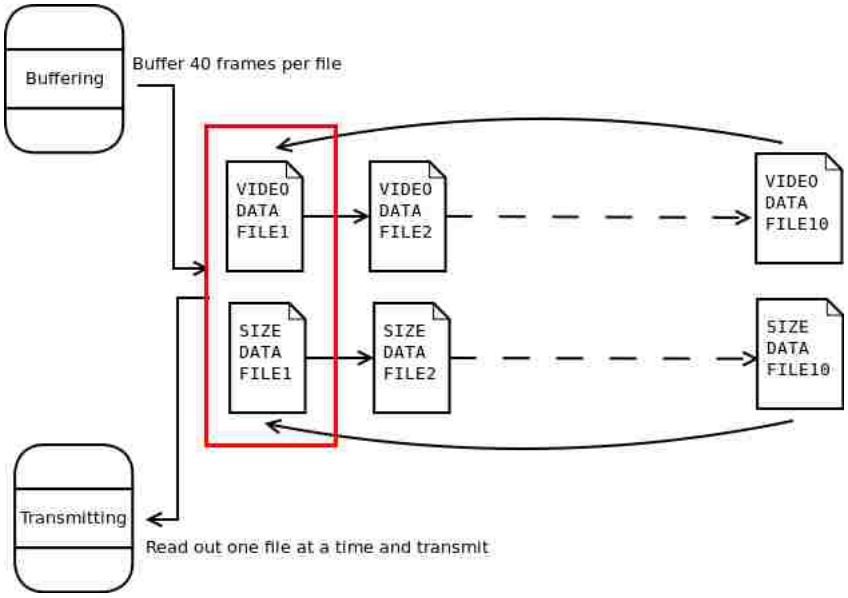

Figure 3.16: Circular File Buffer implementation at Server End

Chapter 4

Conclusion

In this thesis, it was shown that the conventional histogram equalization algorithm in which the image histogram is equalized to span the entire gray level range [0-255] does not yield the best output image quality for night vision in terms of Peak Signal to Noise Ratio and subjective evaluation of the images. This was established experimentally on a number of images taken in low illumination environments. The PSNR for each of these images after histogram equalization over the graylevel intensity values ranging from the maximum intensity level of the original dark image to the maximum possible value of 255 was calculated. The calculated PSNR supported our claim that the equalizing intensity of 255 does not give the best results. The existence of an optimum intensity at which the PSNR is maximum was also established. Based on this analysis, a new incremental histogram equalization approach was developed in which the image histogram was equalized at each intensity level starting from the maximum intensity found in the original image upto 255, but in which, for each step the histogram equalized output image of the previous step was used as input. It was found experimentally that this new incremental approach yields higher maximum PSNR values than the conventional histogram equalization. This algorithm was further used to achieve real time night vision in digital video and was successfully applied in implementing a video surveillance system in which video frames were captured and stored on a server and transmitted over a computer network to a client computer in which it could be viewed. This system achieved good output image quality even for dark video frames enabling a viewer at the client's end to see under poor visibility conditions.

Further work can be done in enhancing the quality of the output images produced. For example, in each of the images, the presence of striped noise of similar pattern was observed, which could be due to the low illumination conditions in which the images were taken. Digital filtering techniques can be applied to identify and remove such noises to further enhance the quality of the output images.

Appendix A

Histogram Equalization in C#

A.1 Histogram Equalization operation

```
private Image equalizeHistogramGradual(Image img, int val)
{
Bitmap bmp = (Bitmap)img;
Bitmap bmpEq = new Bitmap(pbEq.Width, pbEq.Height);
Color c, cEq;
int minIntensity = 256;
int maxIntensity = 0;
int[] orgIntensity = new int[256];
int eqIntensity = 0;

//calculate minimum and maximum intensity
for (int x = MIN_X; x <= MAX_X; x++)
{
for (int y = MIN_Y; y <= MAX_Y; y++)
{
c = bmp.GetPixel(x, y);
if (c.R > maxIntensity)
maxIntensity = c.R;
if (c.R < minIntensity)
minIntensity = c.R;
orgIntensity[c.R]++;
}
}
```

```
if ((maxIntensity + val) > MAX_INTENSITY) eqIntensity = MAX_INTENSITY;
else eqIntensity = (maxIntensity + val);


int[] cumFrequency = new int[maxIntensity + 1];
double[] cumFrequencyProb = new double[maxIntensity + 1];
int[] newIntensity = new int[maxIntensity + 1];
for (int i = 0; i <= maxIntensity; i++)
{

cumFrequency[i] = 0;
cumFrequencyProb[i] = 0;
newIntensity[i] = 0;
}


for (int i = minIntensity; i <= maxIntensity; i++)
{

if (i == 0) { cumFrequency[i] = orgIntensity[i]; }
else cumFrequency[i] = cumFrequency[i - 1] + orgIntensity[i];
cumFrequencyProb[i] = (double)cumFrequency[i] / (double)TOTAL_PIXELS;

newIntensity[i] = (int)(cumFrequencyProb[i] * eqIntensity);

}

//new HE Image
for (int x = MIN_X; x <= MAX_X; x++)
{
for (int y = MIN_Y; y <= MAX_Y; y++)
{
```

```
c = bmp.GetPixel(x, y);
cEq = Color.FromArgb(c.A, newIntensity[c.R], newIntensity[c.R],
    newIntensity[c.R]);
bmpEq.SetPixel(x, y, cEq);

}
}
pbEq.Image = bmpEq;
if (imageSaveHE != null)
imageSaveHE.Dispose();
imageSaveHE = bmpEq;

FileStream fs = new FileStream(eqIntensity.ToString()+".bmp",FileMode.
    OpenOrCreate);
imageSaveHE.Save(fs, System.Drawing.Imaging.ImageFormat.Bmp);
fs.Close();

if(lightedImage != null)
this.calcPSNR(lightedImage, bmpEq, eqIntensity);
thresholdIntenisty = maxIntensity;
return bmpEq;
}
```

A.2 PSNR calculation

```
private void calcPSNR(Image original, Image equalized, int val)
{
Bitmap bmpOrg = (Bitmap)original;
Bitmap bmpEqu = (Bitmap)equalized;
double sum = 0;
double orgIntensity, equIntensity;
double MSE = 0;
double PSNR = 0;
Color cOrg, cEqu;
```

```
for (int x = MIN_X; x <= MAX_X; x++)
{
for (int y = MIN_Y; y <= MAX_Y; y++)
{
cOrg = bmpOrg.GetPixel(x, y);
cEqu = bmpEqu.GetPixel(x, y);
orgIntensity = (double)cOrg.R;
equIntensity = (double)cEqu.R;
sum += Math.Pow((orgIntensity-equIntensity),2);
}
}
MSE = sum / TOTAL_PIXELS;
PSNR = 20 * Math.Log10(MAX_INTENSITY) - 10 * Math.Log10(MSE);
Console.WriteLine(val + ": " + MSE + ": " + PSNR);
DataRow r = PSNRTable.NewRow();
r[MaxIntensityCol] = val;
r[PSNRCol] = Math.Round(PSNR,3);
PSNRTable.Rows.Add(r);
}
```

Appendix B

Video Receiver in C#

B.1 Buffering thread action

```
public void FillBuffer()
{
int max = AppSettingsController.GetAppSetting("BufferFilesLimit", 20);
int maxFrames = AppSettingsController.GetAppSetting("FramesInFileLimit",
    40);
Console.WriteLine("filling...");

while (true)
{

try
{
receiveByteArray = listener.Receive(ref groupEP);
saveByteArray.Add(receiveByteArray);
saveFrameSize.Add(receiveByteArray.Length);

if (saveByteArray.Count >= maxFrames)
{
fileCount++;

if (fileCount > max)
fileCount = 1;

String vidFilename = videoFilePrefix + fileCount.ToString() + ".dat";
```

```
String sizeFilename = sizeFilePrefix + fileCount.ToString() + ".dat";
FileStream vidFileStream = new FileStream(vidFilename, FileMode.Create,
    FileAccess.Write);
StreamWriter sizeFileStream = new StreamWriter(sizeFilename);
try
{

foreach (byte[] byteArrayElement in saveByteArray)
vidFileStream.Write(byteArrayElement, 0, byteArrayElement.Length);



foreach (int sizeArrayElement in saveFrameSize)
sizeFileStream.WriteLine(sizeArrayElement);
}
catch (Exception ex)
{
}
finally
{
saveByteArray.Clear();
saveFrameSize.Clear();
filesBuffered = filesBuffered > 5 ? 5 : filesBuffered + 1;
vidFileStream.Close();
sizeFileStream.Close();

}



}

if(isLive)
Program.f.updateProgress();
else
Program.f.updateProgressDB();
```

```
}
catch (Exception ex)
{
Console.WriteLine("Buffer " + ex.Message);
}
}
}
```

B.2 Client Configuration controller

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="ListenPort" value="11000"/>
    <add key="ListenPortDB" value="11001"/>
    <add key="receiveFromAddress" value="127.0.0.1"/>
    <add key="receiveFromAddressDB" value="127.0.0.1"/>
    <add key ="sendToAddress" value="127.0.0.1"/>
    <add key ="sendToPort" value="11002"/>
    <add key="BufferFilesLimit" value="20"/>
    <add key="BufferLimitForDisplay" value="3"/>
    <add key="FramesInFileLimit" value="40"/>
    <add key="DisplayDelay" value="200"/>
    <add key="WaitForFileDelay" value="200"/>
  </appSettings>
</configuration>
```

Bibliography

[1] Z. Yu, W. Xiqin, and P. Yingning, "New image enhancement algorithm for night vision," *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, vol. 1, pp. 201–203, 1999.

[2] S. Huang, F. Cheng, and Y. Chiu, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *Image Processing, IEEE Transactions on*, vol. 22, pp. 1032–1041, 2013.

[3] Y. Zhao, X. Fu, Z. Wang, and G. Fan, "A new image enhancement algorithm for low illumination environment," *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*, vol. 1, pp. 625–627, 2011.

[4] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 889–896, 2000.

[5] J. Zhao and S. Lei, "Automatic digital image enhancement for dark pictures," *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 2, pp. 2–2, 2006.

[6] H. Ibrahim and N. S. P. Kong, "Brightness preserving dynamic histogram equalization for image contrast enhancement," *Consumer Electronics, IEEE Transactions on*, vol. 53, no. 4, pp. 1752–1758, 2007.

[7] R. C. Gonzalez and E. Richard, *Digital Image Processing.* Upper Saddle River, New Jersey: Pearson Education Inc., second ed., 2002.

[8] R. Sanders, "Night vision optics explained." `http://www.securitysa.com/article.aspx?pklarticleid=5857`, November 2009.

[9] S. U. Inc., "Laser range-gated imaging for imaging at long ranges and through obscurants." `http://www.sensorsinc.com/laserrangegating.html`.

[10] S. W. Smith, *The Scientist and Engineer's guide to Digital Signal Processing.* P.O. Box 502407, San Diego, CA: California Technical Publishing, second ed., 1999.

[11] A. Lustica, "Ccd and cmos image sensors in new hd cameras," *ELMAR, 2011 Proceedings*, pp. 133–136, 2011.

[12] T. D. I. I. K. Center, "Ccd vs. cmos." `http://www.teledynedalsa.com/imaging/knowledge-center/appnotes/ccd-vs-cmos/`.

[13] C. Ponyton, "Frequently asked questions about gamma." `http://www.poynton.com/notes/colour_and_gamma/GammaFAQ.html`.

[14] A. Loza, D. Bull, and A. Achim, "Automatic contrast enhancement of low-light images based on local statistics of wavelet coefficients," *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 3553–3556, 2010.

[15] L. Meylan and S. Süsstrunk, "Color image enhancement using a retinex-based adaptive filter," *Image Processing, IEEE Transactions on*, vol. 4, no. 9, pp. 2820–2830, 2006.

[16] J.-Y. Kim, L.-S. Kim, and S.-H. Hwang, "An advanced contrast enhancement using partially overlapped sub-block histogram equalization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 4, pp. 475–484, 2001.

[17] A. S. Tanenbaum, *Computer Networks*. Upper Saddle River, New Jersey: Pearson Education Inc., fourth ed., 2003.

[18] I. A. N. Authority, "Service name and transport protocol port no. registry." `http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml`, February 2013.

[19] M. M. L. D. Guide, "Using udp services." `http://msdn.microsoft.com/en-us/library/tst0kwb1.aspx`.

[20] W. J. Smith, *Modern Optical Engineering*. McGraw-Hill Professional, third ed., 2000.

[21] N. Instruments, "Peak signal-to-noise ratio as an image quality metric." `http://www.ni.com/white-paper/13306/en`.

[22] T. Kim and J. Paik, "Adaptive contrast enhancement using gain-controllable clipped histogram equalization," *Consumer Electronics, IEEE Transactions on*, vol. 54, pp. 1803–1810, 2008.

[23] S.-H. Yun, J. H. Kim, and S. Kim, "Contrast enhancement using a weighted histogram equalization," *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pp. 203–204, 2011.

Vita

Graduate College
University of Nevada, Las Vegas

Nishikar Sapkota

Degrees:

Bachelor of Computer Engineering 2008
Pokhara University, Nepal

Thesis Title: Real Time Digital Night Vision using Nonlinear Contrast Enhancement

Thesis Examination Committee:

Chairperson, Dr. Evangelos Yfantis, Ph.D.
Committee Member, Dr. Laxmi Gewali, Ph.D.
Committee Member, Dr. John Minor, Ph.D.
Graduate Faculty Representative, Dr. Peter Stubberud, Ph.D.