ISBN 0-315-54813-4

ADAPTIVE HISTOGRAM REGRADING

FOR

REAL-TIME IMAGE ENHANCEMENT

by

John Craig Muller

A thesis

presented to the University of Manitoba

in partial fulfillment for the degree of

Master of Science

in

Industrial Engineering

Winnipeg, Manitoba

May 1989

ADAPTIVE HISTOGRAM REGRADING FOR REAL-TIME IMAGE ENHANCEMENT

BY

JOHN CRAIG MULLER

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ABSTRACT:

Human visual perception of grey-level image information lacks sufficient sensitivity to adequately interpret images in environments with severe lighting conditions. It has been shown that image enhancement can be accomplished by grey-level regrading. Present non-spatial image regrading techniques are image dependent and real-time implementation requires a priori knowledge about the input image. On the other hand, present spatial techniques are more successful in image independent applications but due to computational intensity these techniques are not suited to real-time applications. The aim of this research work was to develop a real-time image independent image enhancement technique.

In this thesis a new technique called Adaptive Histogram Regrading (AHR) is presented which has the high speed and low computational cost of non-spatial techniques while remaining adaptive to handle a wide variety of input images without a priori knowledge of them. This is achieved by examining the grey-level histogram of an image and performing feature characterization (fingerprinting). The image is then indirectly segmented based on this characterization and histogram regrading is applied to each region using local criteria.

This AHR technique appears to work on all images, independent of their histogram attributes. Its adaptive ability gives excellent performance for general purpose

image enhancement applications where there is no prior knowledge of the image. In a worst case scenario, there is no degradation of the output image because because no regrading of the the input image is necessary. In other cases, specifically where the input image has a skewed multi-modal histogram, there appears to be a significant improvement resulting in a more natural appearance to the output image. The speed and elegant simplicity of this technique easily lends itself to real-time implementation on standard PC type computer equipment.

ACKNOWLEDGMENTS

Thomas Edison didn't fail 17000 times, he found 17000 ways the experiment didn't work. Special thanks to my wife, Deyanira, for all her moral support through the all the frustration of methods that didn't work. This was central in helping me to achieve my goal of finding the one way it did.

I would like to thank my advisors John D. Jenness and A.W. DeGroot for their ideas and support in technical direction of this thesis and for providing the necessary equipment to realize the objectives.

My appreciation is also extended to all my friends, fellow students and staff for their help when it was needed, especially my office mate John Kaye for all his time and valuable comments.

TABLE OF CONTENTS

LIST OF FIGURES:

1. INTRODUCTION

The human vision system is a complex combination of image translation and image understanding. Through study of the theories of perception it has been possible to isolate characteristics of perception, but separating the eye and the brain functions is difficult because they are both essential in the vision process. Many of the vision functions are learned processes which are unconscious and based on how the brain judges the incoming information. For example, the color yellow can be seen identically as both a single (monochromatic) frequency of light or as a mixture of red and green light. Lightness is also a judgmental process, our perception of illumination of some object, relative to the illumination of the whole viewing area. If the entire viewing area is considered relative to our knowledge of bright and dark, then it is possible to make a judgment about lightness of the entire scene. These mental calculations have been explained as cognitive explanations because they assume perception is based on unconscious processes similar to conscious reasoning.

When our vision is put to the test of determining illumination of various levels in a single image, it would seem that our visual perception would be capable of distinguishing a large number of distinct brightness levels in a single image. However, the result is exactly the opposite[6]. In a single image, the human vision system can only perceive 16 to 32 brightness levels, or grey-levels.

This may be due to the way in which the brain processes the visual information, or the way the eyes' receptors respond to the light. Hochberg [7], in 1978, explained that the receptors are not independent from one another, but rather, the excitation of one receptor causes a negative excitation of its neighbors. This was demonstrated in 1978 by Gonzalez and Wintz [6] in the illusion of Mach bands, named after Ernst Mach who first described the illusion in 1865. This effect may also be involved in grey-level resolution.

1.1 Problem Statement:

Since the human vision system has poor sensitivity to grey-level information, increasing the contrast between consecutive grey-levels in a digitized image also increases the visual quality of that image. Histogram modification techniques, which stretch certain ranges of the image histogram, are widely used as a standard technique in digital image enhancement.

The basic problem with all non-spatial techniques is that they are dependent on the input image. What method to apply, and how to apply it all depend on the type of original image. If the range of image types is narrow and known in advance, such as parts on a conveyor belt, then the optimal method can be chosen prior to processing. As long as the images remain in this narrow range of types, the enhancement will always be satisfactory.

In 1988, McCollum and Bowman [12] presented a hardware system which implements histogram modification in real-time using a linear equalization. Although the example images are greatly improved, they fall into a narrow category of image types. The authors acknowledged in their conclusions that the most effective type of histogram modification is image dependent, and suggested that the process could be implemented as an interactive system. For general purpose applications however, image types are not constrained, and optimal histogram modification can only be performed on an interactive basis. This precludes these techniques for interactive use only, which is unsatisfactory for real-time general purpose applications.

Real-time image enhancement must be free from any interactive requirement, since human response time is long. Non-spatial techniques, such as Adaptive Histogram Equalization, can perform without human interaction but because they are spatial, they cannot be performed in real-time without an expensive computing engine.

An image histogram is actually the sum of many distributions of varied shapes and sizes [1]. Each of these distributions corresponds to some object or group of objects and/or regions in the image. Sometimes this is a single object such as a satellite in space or sometimes it may be due to a texture created by many objects such as a forest or rivers and lakes viewed from space. A single image histogram will often consist of a number of peaks and valleys of varied size and shape. Each peak will usually result from a single distribution and each valley corresponds to a transition from one distribution to the next. Each distribution is referred to as a mode of the histogram and may be divided from one or more regions in the image with pixels of similar grey-scale intensities.

The modes of a histogram may vary in size, but size does not always indicate the relative importance of the data in the mode. If an object in a scene is small and the contrasting background is relatively uniform, then the object will correspond to a small mode in the histogram next to a large mode, corresponding to the background. Histogram modification expands or compresses grey scales based solely on their magnitude and therefore will redistribute a mode based only on its relative size. As a result, the object is compressed at the expense of the background, which is the reverse of what is desired.

For most kinds of images, histogram modification is successful in improving image quality. There are, however, some classes of images which exhibit degraded or destroyed quality as a result of applying traditional histogram equalization techniques. Such is the case for certain kinds of strongly multi-modal images which contain both large and small modes in the same histogram. An example of this is shown in Figure 1. In this image the portion of the image occupied by satellite is small in comparison to the background. The Histogram Equalized form of this image is shown in figure 2. Using conventional techniques the background tends to affect the tranformation much more than the satellite creating undesireable results. The problem is to enhance information embedded in one region of an image as indicated by a local mode of the histogram, without damaging information in another mode of the image.

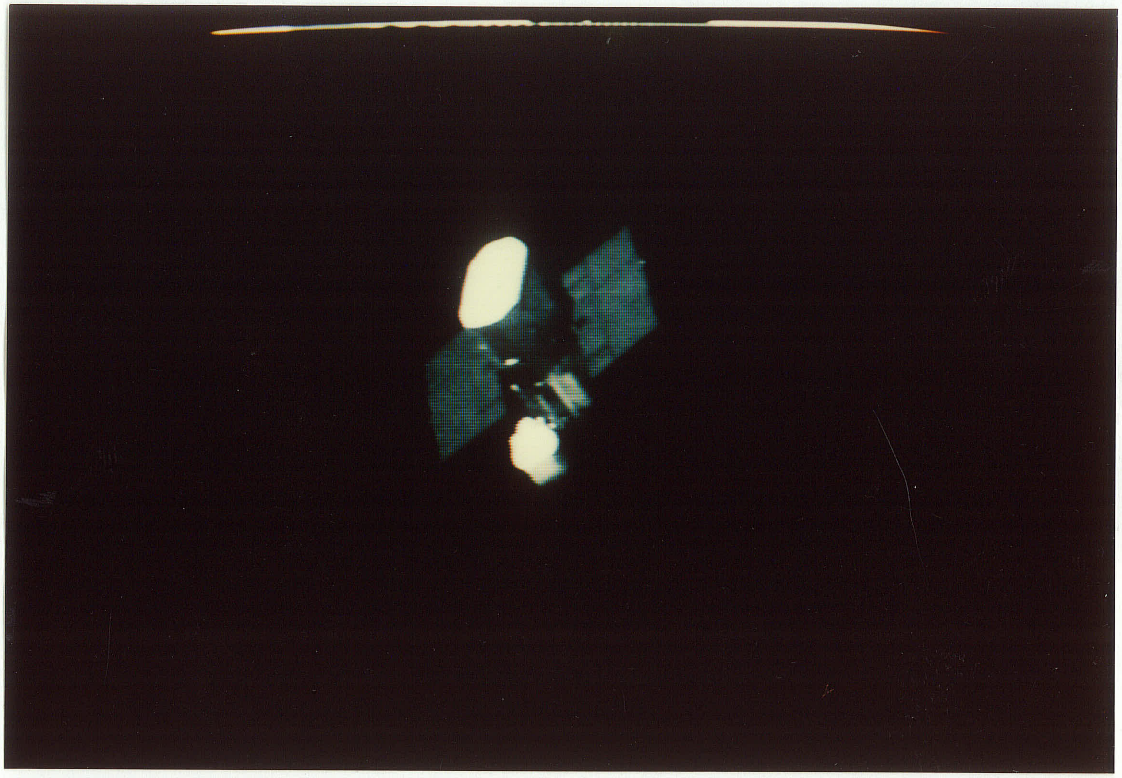


Figure 1: Original Satellite Image

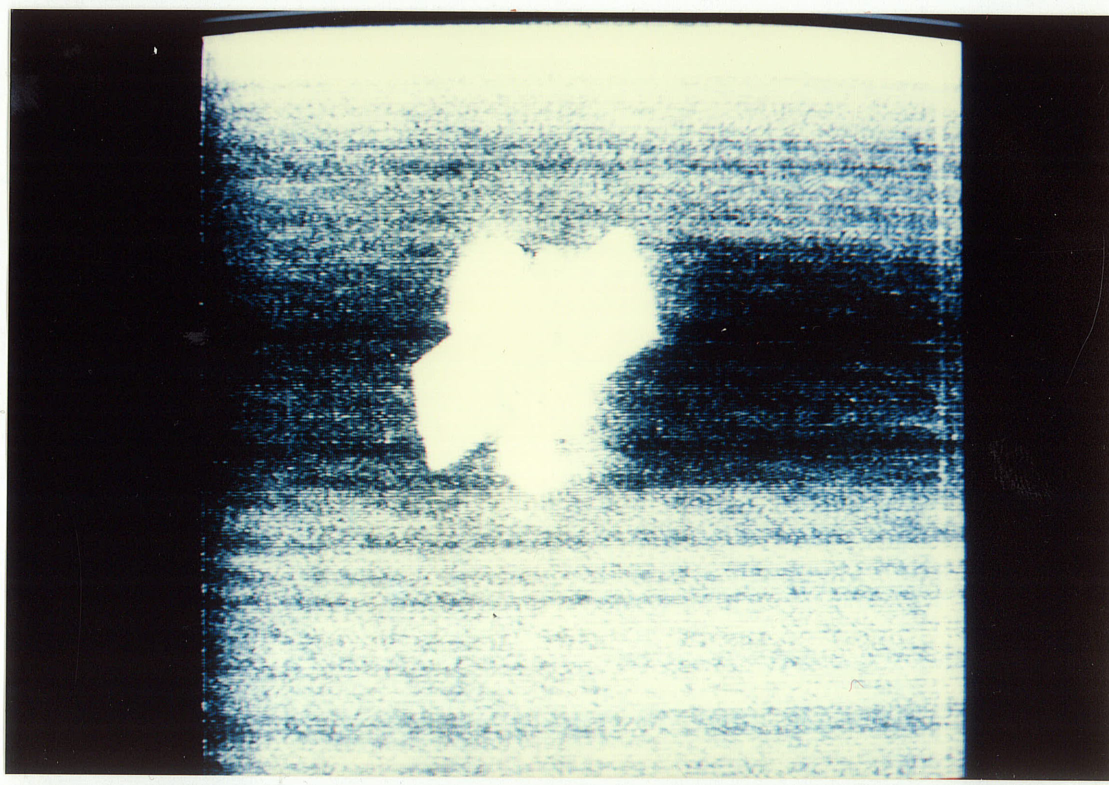


Figure 2. Histogram Equalized Satellite Image.

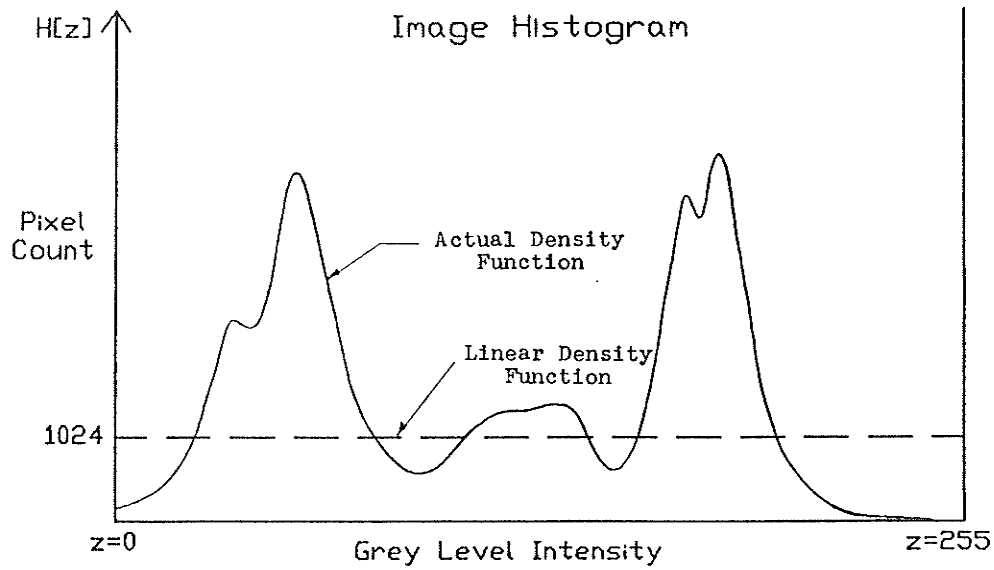
1.2 Research Goals:

When histogram modification in performed, the modes in a histogram are redistributed to approximate a desired function. The reasons for failure of this technique under some conditions give clues as to how it could be improved. The research presented here addresses the shortcomings of standard techniques.

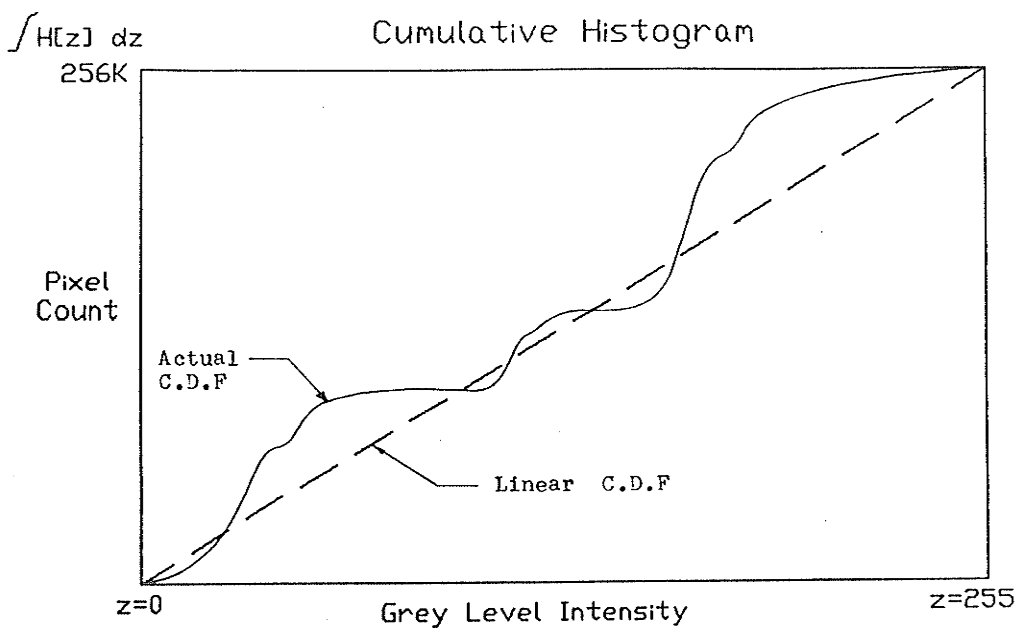
In strongly modal image histograms, such as in Figure 3(a), the modes can often be directly equated to an object in the image space. Multi-modal histograms often result from images of objects on an uniform brightness background where the object represents only a small portion of the image area. The goal of the research is to create a general procedure for isolating the modes of the histogram, which will correspond to objects in the image, and regrade them to produce an improved image. This approach would be superior to standard methods because it would have the ability to adapt to any kind of input histogram, and therefore any image.

Very often the object viewed by the camera occupies only a fraction of the image space, leaving the rest as a dark (or light) background. Conventional histogram modification attempts to improve image quality by redistributing grey-levels of the image such that the end result is a <u>Cumulative</u> <u>Distribution</u> <u>Function</u> (CDF) which is linear, corresponding to a histogram with equal percentage of the total area in each quantization level, as shown in Figure 3(b). Occasionally, the term Histogram Equalization is used loosely to include other CDF's which are non-linear. This is the case with the DI-IRIS Image Processing software [19]. Usually these include CDF's of Bell, Cubic, Logarithmic or Exponential type distributions. A linear function is most often used because it equalizes the image without emphasizing any particular grey-scale range.

Each CDF accents the histogram in a different way and the choice is largely a subjective one. A particular function may be used if it is known in advance what parts of grey-scale range need to be accented, but this is not usually the case. The technique integrates the histogram; wherever the cumulative histogram is less than the desired function the grey-scale is compressed, and wherever the cumulative histogram exceeds the desired function, the grey-scale is expanded.



(a) Histogram



(b) Cumulative

Figure 3. Density Functions.

In this thesis the histogram will be considered as an important source of input information. The research is directed at using this knowledge to guide the modification process as well as infer basic attributes of the input image. Inferences are made about the spatial nature of the image from non-spatial histogram data.

The goal was to create an adaptive general purpose technique for image enhancement in real-time applications. A new method, which we shall refer to as Adaptive Histogram Regrading (AHR), was created to address both the multimodal images as well as images which can be processed by conventional methods.

In AHR, each distinct mode is identified in advance and the range of grey-levels corresponding to each mode is separated. This allows us to indirectly carve an image into pieces, or segment the image, and then treat each piece as a smaller image with its own histogram. By characterizing the modes in the histogram, the AHR technique can provide an adaptive expansion to the image, and small modes which correspond to regions such as that of the satellite, are considered independently, thereby conserving vital information.

Once the basic rules or heuristics are established, then the goal is to implement these procedures in a computer program on a personal computer. If reasonable speed is attained using a high level computer program then implementation in dedicated hardware would be highly feasible since dedicated hardware is many times faster.

1.3 Thesis outline:

Global histogram modification techniques treat the image as a single entity, but because of the dependency on the input image, the results are sometimes undesirable, or they are not optimal. Current non-spatial techniques do not extract any characteristics from the input histogram, and unpredictable results may occur if these histograms are very different from one another. Alternatives to non-spatial techniques have segmented the image into regions and perform a modification process on each region. Image segmentation can be performed in a number of ways. The image can be broken down into a number of smaller rectangular cells and then each cell can be processed separately [8], or edge detection can be used to determine boundaries of regions and each region can then be equalized [9]. Both techniques use spatial information to perform the task, and thus then require an extensive amount of CPU time, which eliminates the ability to do real-time processing on anything but powerful special purpose hardware. If this segmentation process can be carried out based only on the histogram then real-time processing is possible on inexpensive general purpose hardware.

Chapter three explains how the histogram can be broken down and analyzed using maxima and minima. An algorithm is presented for eliminating histogram noise using a 1 dimensional implementation of local averaging. The histogram analysis give important information for segmentation of the histogram into several parts. A method for classifying each mode based on a vector representation scheme is presented. Heuristics to condense all the segmented pieces into a few distinct modes are also presented.

Chapter four describes the regrading process of each mode based on the information collected from histogram analysis. Each mode is stretched and then fitted in the histogram using both standard expansion methods and a new rubber banding technique.

2. RELATED RESEARCH

2.1 Histogram Modification

Human visual perception has been a subject of much study, yet while the mechanism behind image transformation and translation, the eye, is well understood, the way in which this information is used by the brain is unclear. Optical illusions are evidence that human vision is strongly linked to the brain's interpretation of the visual data. The human eye is a matrix of discrete elements which provide analog information to the brain. This would indicate that visual information is limited in its spatial resolution but has an infinite number of brightness levels, however it has been argued that brightness sensitivity is limited to the context of the overall brightness. Human vision is an adaptive process that should be considered as an integral part of the overall image processing system [3].

Due to the nature of the human visual system, it is possible to modify the grey level information in an image in order to enhance its visual perceptibility. Certain types of spatial operations have shown to improve image quality, but their computational requirements put them out of reach for economical real-time image enhancement. Other forms of operations which deal only with the grey level data, or histogram, have proven to be effective for economical real-time applications [12]. These non-spatial operations are referred to as histogram modification techniques or sometimes as histogram regrading.

The early work from 1975 to 1977 in histogram modification techniques is detailed by Hummel [8,9] . He pointed out that since human visual perception has difficulty in distinguishing low contrast areas (possible sensitivity may only be 16-32 grey levels at any one time), a grey level transformation may be used to increase visual contrast. This increase in the contrast is considered to be an artificial enhancement because it provides no enhancement in the theoretic sense, but rather an improvement in the interpretation of the image by the human vision system [9]. Hummel showed that modifying the histogram to produce a new histogram which is flat will achieve this improvement, in many images this enhancement can be quite dramatic.

Since real grey-level histograms are discrete, and the transformation is to be single valued, the simplest histogram transformation involved merging of grey-levels together but no grey-levels could be broken up. This technique, called <u>Histogram Equalization</u> (HE) [9], approximates a flat histogram function. In theory there is only one way to equalize a histogram; the image histogram must be modified so that it forms a completely flat distribution. In reality the image and its histogram consists of discrete points. Unless some heuristics are implemented so that one grey-level may be distributed into many, an equalized histogram can only be an approximation.

Subsequently, techniques to produce an even flatter histogram were introduced. By introducing an algorithm which can break up the previous unbroken bins, and map one grey-level into many, an extremely flat histogram can be realized. The method works by examining the immediate neighborhood of a pixel in question and based on these neighbors decide which bin to place it in. This technique is a special form of histogram equalization referred to as Histogram Flattening (HF) [9]. This method proved to have little practical merit for several reasons: the results of histogram flattening look almost identical to histogram equalization; no information is added to the new image; it has a tendency to filter high frequency components in the image; and it requires much more computation than histogram equalization.

In the implementation of either histogram equalization or flattening there are several ways in which the pixels can be mapped into the target histogram. Usually some regions will be expanded while others are compressed. Mapping several grey-levels into one grey-level presents no difficulty, however, when one must be mapped into many rules must be introduced in order to decide which grey-level a pixel should be assigned. There are three rules for mapping one grey-level into many, as outlined by Pavlidis [14] in 1982:

1) Always map each pixel into the midpoint of its new grey-level range. When one grey level must be mapped into three grey-levels then it would be mapped into the middle grey-level and the other two would be left unused. The resulting histogram still has an appearance of peaks and valleys, but when it is integrated it approximates a linear plot as desired (due to the discrete nature of the histogram the plot is a staircase which approximates the linear function). This method has the advantage of simplicity, speed and it makes use of the full dynamic range. The disadvantage is that not all of the quantization levels are properly utilized.

2) Always map each pixel randomly into its new grey scale range. The result of this process is to produce a flat histogram which appears like the mathematical ideal. Since these pixels are assigned at random this process offers only minor improvement over the first rule. The advantage of the random assignment is it uses all the quantization levels and avoids any systematic errors resulting from the need to make a choice. However, it requires about four times as much computation as the first rule.

3) Examine the image in the immediate neighborhood of each pixel, calculate the average value of a 3x3 matrix and then assign the pixel to the grey-level which is closest to this value. The result of this equalization is a high quality

equalization since it uses spatial information to help guide the process. The major drawback of this operation is the introduction of a 2 dimensional image operation, requiring at least ten times as much computation as the first rule. Another side effect is a certain amount of edge smearing. When a 3x3 matrix is averaged it imitates a low pass filter, removing some of the high frequency information in the image.

The drawback of these last two types of histogram modification is that any improvement due to the grey-scale transformation depends on the input image. It is possible that some input images may be degraded by these transformations. In order to get an optimal output image a subjective decision must be made by the observer. For this reason, most image processing systems provide an interactive approach to histogram modification.

Some reports have argued that if the human retinal system is included in the overall imaging system, then an image which produces a flat distribution as the output of human retinal receptors will provide the best visual representation. Human brightness perception is non-linear function. Therefore, applying linear histogram equalization to an image cannot achieve a linear perceived brightness, since the cascade of linear and non-linear functions is non-linear. Human perception is generally understood to be logarithmic. In 1977 it was suggested by Frei [5] that a transformation which includes this brightness perception would yield superior visual quality than that of a linear method. The problem with modeling human perception is its dependency on the viewing conditions, which are not known in advance. An expression which most closely matches human perception was chosen and included, which resulted in a histogram similar to a hyperbolic function. This method is referred to as <u>Histogram Hyperbolization</u> (HH) [5]. Results showed that hyperbolization produced images which were consistently better than their histogram equalized counterparts. The hyberbolic function results in a better presentation of the image to the eye, but since HH is simply HE with a different CDF, it suffers from all the same shortcomings associated with HE.

The histogram modification techniques described up to this point are considered to be optimum when the error between the output CDF and the desired CDF is minimized. Each of these methods have a cost in terms of the amount of information lost (when grey-levels are compressed) and the degree of contrast enhancement (when grey-levels are stretched). There are some classes of images in which the amount of information is lost disproportionately to the degree of contrast enhancement. This occurs when a portion of the histogram is stretched beyond what is visually required for sufficient contrast. Any image which has a large uniform background will create such a result.

A technique for balancing the error against the information lost by a specified transformation was introduced by Kautsky et al [10] in 1984. A Smoothed Histogram Modification (SHM) was achieved by balancing the grey-level modification between linear regrading, which is also called contrast stretching or compression, and a transformation which matches a specified reference histogram. Smoothed regradings were demonstrated between two cases: linear regrading and equalization. A padding parameter was introduced, measured from 0 to 1, designed to regulate the regrading between equalization (padding = 0) and linear regrading (padding = 1). The procedure was tested on images of a forest from LANDSAT data, resulting in dramatically improved images over both linear regrading and histogram equalization.

The selection of the padding factor for SHM is chosen interactively by the user. This means that a subjective decision must be made by the user for each image in order to achieve the best results. This precludes SHM to interactive use only, and cannot be implemented in real time without some form of additional decision making built in to the computer.

2.2 Adaptive Histogram Equalization

Another method which is spatial in nature has shown that breaking the image into many smaller images is beneficial in histogram equalization, allowing the equalization to be context sensitive rather than global. This method, which is referred to as Adaptive Histogram Equalization (AHE), is described in its basic form by Hummel [9] in 1977. Each pixel in question is mapped based on the pixels in a region surrounding it, which is referred to as its contextural region. This technique, while providing some advantages over conventional non-spatial methods, is slow and can over-enhance noise in homogeneous regions. Modified forms of adaptive histogram equalization were introduced by Pizer et al [8] called Weighted AHE (WAHE) and Contrast Limited (CLipped) AHE (CLAHE). WAHE offered little improvement over standard AHE, but CLAHE produced improved results and was less time consuming to perform. These time factors, however, are relative to standard AHE which is extremely slow compared to non-spatial techniques. When applied to an image data set like those used in this thesis, processing time on a VAX 11/780 in C language was about 2 minutes. This is much improved over previous implementations which needed about 20 minutes.

Another form of AHE was recently introduced by Vossepoel et al [17] (1988), capable of addressing the problem of over-enhancement of noise. This method was called Variable Region Adaptive Histogram Equalization (VRAHE) and worked by combining several histograms in neighboring regions, rather than averaging it with a uniform distribution. This form of AHE is said to produce preferable results over other forms of AHE with approximately the same computational requirements.

In a variation of AHE, the histogram of the contextural region is used to perform different enhancement tasks. These enhancements were presented by Chocia [3] in 1988 and he described how the histogram of a region can be used to detect edges, perform smoothing and enhance images. This work demonstrates that local criteria can be a very powerful tool for enhancement of images.

AHE is a powerful technique for image enhancement, but it is also slow and does not fit the criteria of high speed required for real-time applications. To acheive the goal of speed and image independence a technique must be both non-spatial and capable to extract enough image information to be adaptive.

2.3 Histogram Fingerprinting

All related research in adaptive histogram equalization has shown that local criteria is very effective for image enhancement. <u>Iterative Histogram Modification</u> (IHM) presented by Rosenfeld and Davis [16] in 1978 used histogram peaks to rapidly and inexpensively segment an image into regions of similar grey level intensities. This was achieved by applying a one dimensional equivalent of the curve thinning algorithm to an image's histogram. After only a few iterations, the process converges, sharpening the peaks of a histogram into spikes and segmenting the image into distinct regions. In AHR presented in this thesis, it will be shown how this same segmentation can be achieved without actually modifying the histogram. Rather than compressing all the grey-level information, the information is retained and segmented regions are represented by pseudo-coloring using output lookup tables.

Even though the histogram of an image contains no spatial information, iterative histogram modification proved to be a fast and effective means of segmenting the image using only the histogram. This shows that there is an intimate correspondence between regions in 2-D space and peaks of the histogram in 1-D space.

In some studies, it has been observed that zero-crossings in 1-D histogram space corresponded to contours in 2-D space, referred to as Scale-Space. Based on this principle, Carlotto [1] in 1987 showed that a histogram could be approximated by a sum of normal distributions, where each of these components corresponded to objects and/or textures present in the image. This process is called histogram fingerprinting. A histogram can be approximated in increasing or decreasing accuracy based on the number of components (modes) detected, where each mode corresponds to a pair of zero-crossings. The histogram is approximated by summing a normal distribution, of varying width and magnitude, for every mode detected. In his experimental results the histogram was analyzed at a scale which detected 5 modes, which was considered low enough to determine the components of interest. Since many of the modes are very close to one another, the tails on either side of the normal distribution overlap one another. Carlotto's approach is able to resolve modes which have a high degree of overlap by using iterative estimation and convergence testing. However, when the image is to be segmented, a threshold point must be chosen and the overlap in the modes cannot be resolved in 2-D space.

For practical image segmentation the process of histogram fingerprinting is useful when modes are reasonably distinct. Usually a histogram will contain at most 5 large modes which are distinct. If we consider two overlapping modes as one distinct mode, then image segmentation by a simple form of histogram fingerprinting can be effective and economically achieved. Subsequent histogram modification using local histogram criteria of these segmented regions can be used for efficient real-time image enhancement.

3. HISTOGRAM ANALYSIS

3.1 The Grey-Level Histogram

A digitized image is stored as a 2-D array of discrete data points indicating the grey-level intensities at that point in the image. The indices for each of the dimensions in the array are the cartesian coordinates of the point in the image space. If all the data points in the array are sorted based on their value (grey-level) and then tabulated, a new array results which is one dimension whose index is grey-level intensities. An image which has 256 grey-levels would create an array of 256 elements. Each of these elements is often referred to as a bin because the process is much like sorting different objects into discrete bins. The new 1-D array indicates the frequency distribution of grey levels in the image and is called the <u>Grey-Level Histogram</u>. Figure 4 is the example image which will be used to decribe the various aspects of AHR. Figure 5 is a trimodal histogram of the image in Figure 4. Each of the peaks corresponds to a feature in the input image such as the foam pad, the ball, the binder, or the white background on which all the objects are resting.

Figure 4. Original Example Image

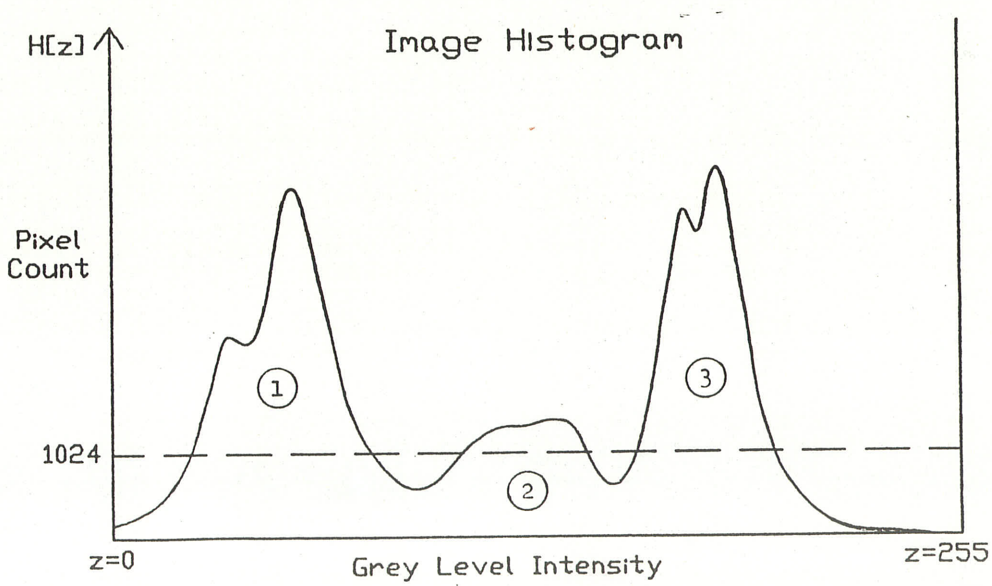


Figure 5. Example Image Histogram.

Theoretically, an infinite number of images can resolve to this histogram, but since our image has a finite resolution, this number is also finite. For purposes of explanation it is clearer to show example histograms in the continuous case. Wherever the transformation must be approximated as in the discrete case, the method of approximation is explained and the difference with the continuous case will be described as the quantization error.

Let us consider some important facts about image histograms in the following items:

1) When an image is reduced to a histogram all spatial information about the image is lost.

2) There is a unique histogram for any particular image but many images may resolve to the same histogram. For example, if an object is moved around in a uniform background the histogram will remain unchanged even though the images are quite different.

3) Most of the previous research papers discuss the histogram in terms of continuous-to-continuous transformations because it applies to the theory of distributions and statistics.

4) When the theory is applied using computer technology, the transformation is adapted to the discrete-to-discrete case.

5) The grey-level histogram gives us information about the image's contrast, which is described by the overall spread of the histogram data. If an image histogram is narrow then pixels are grouped in similar intensities and the image has low contrast. Conversely, if a histogram plot is wide then the image has high contrast.

6) A histogram which has most of the pixels grouped at either end of the grey scale is skewed. Corresponding images will appear either overexposed (white skewed) or underexposed (black skewed).

7) Histograms often appear uneven and have peaks due to regions in the image which have low contrast locally, but with high contrast between the regions.

3.2 General Approach to Histogram Analysis

Adaptive Histogram Regrading (AHR) is a multi-step process which first performs a simple histogram finger-printing procedure, and then uses the extracted information to guide the regrading process. The entire process is given by the thirteen steps shown below. Steps 1 and 2 are pre-processing performed by the DT-IRIS hardware and software. Steps 3 thru 6 describe the histogram analysis stage of AHR. The general procedure for simple histogram finger-printing is a three step process, given by steps 3-5 below. The regrading process, which consists of steps 7 thru 11 is described in the following chapter. Finally the image is modified in the post-processing of steps 12 and 13, again performed by DT-IRIS.

1) Image capture and digitization.
2) Histogram evaluation.
3) Histogram smoothing to remove noise.
4) Determination of histogram maxima and minima.
5) Determination of histogram modes.
6) Merging of non-distinct modes.
7) Segment the image using scale-space method.
8) Mode regrading by local grey-level stretching.
9) Determine the new width of the stretched mode.
10) Perform optimal fitting of new modes.
11) Fit non-selected ranges.
12) Program output Look-Up Table (LUT).
13) Apply output LUT to the input image.

STEP 1: The image capture step is the procedure where an image recorded by the camera is converted from an analog signal and digitized into a discrete matrix of discrete values which can be process by the computer system. This stage of the process is performed by the on board circuitry of the DT-IRIS Frame Grabber Board (or any other similar board). The image is stored on the board in memory buffers as a two dimensional array of integers representing grey-level values of the capture image [19,20]. The display converts these numbers contiuously into grey level intensities onto the image monitor. Figure 6 is a photograph of the image monitor display.

STEP 2: In the second stage of the process the 2-D array of grey_levels must be evaluated and converted into a one dimensional array which describes the density of pixels at each grey-level. This task is performed by the DT-IRIS Programming library using a fast low-level software routine [20]. The histogram is displayed on the computer screen in black and white, as shown in Figure 7. For real-time applications this stage is usually implemented in hardware so that this task is performed very fast. Adaptive Histogram Regrading does not attempt to address the image capture and histogram evaluation processes. Technology to perform real-time histogram evaluation is available and relatively inexpensive.

3.3 Simple Histogram Fingerprinting

Although the process of reducing an image to its histogram discards all spatial information, there is a great deal of image information which can be inferred from the shape of the distribution, and often this information is not used to its full potential. If the modes of a histogram are considered in histogram modification, it is not necessary to have knowledge of all the modes which make up the histogram, but rather to determine only the major trends given by the peaks and valleys. Even if the normal distributions which make up the histogram are known, only those which do not overlap other modes can be treated independently. Modes which overlap beyond a certain threshold cannot be considered independently since there is no spatial information to separate them.

If the independent modes, or distinct modes, of the histogram are used to guide the modification process, the process will become knowledge based and be able to adapt to the entire range of possible histograms. Images which are strongly multi-modal contain several large and/or small peaks which are the result of many distinct modes. The individual modes which make up the image in Figure 5 are shown in Figure 6. Each of these modes can result from one or more regions in the image which have compressed grey scales, and its size indicates the total area of these regions.

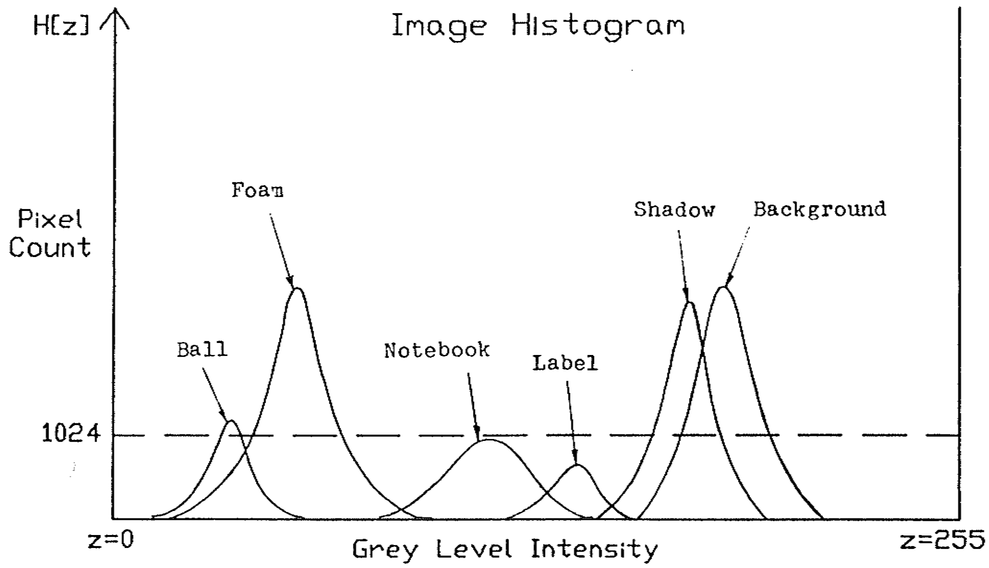


Figure 6. Fingerprint of the Trimodal Image.

There is no way to determine from the histogram if a given mode contains useful information or not. For this reason, all modes must be considered in AHR. For example, a satellite which is pictured in space (see Figure 1) will have a histogram consisting of two modes. One mode is very large and close to zero, corresponding to the large black background. The other is small, only a fraction the size of the large one, corresponding to the satellite. Since the satellite is the part of the image we are interested in, the small mode is the important one in the histogram. Conventional histogram modification fails under these conditions because it does not consider the importance of the small mode. Rather, it attempts to spread the entire black background across the 256 grey levels and compresses the satellite information out of existence.

Before the histogram can be modified using local criteria, it must be segmented into smaller pieces. This, in turn, will also segment the image based on grey scale ranges. The first step towards extracting the features from a histogram is to categorize each of its modes. By segmenting the histogram into distinct ranges of grey-levels corresponding to each mode, these ranges can be expanded individually using the local criteria. This represents the adaptive part of the histogram expansion. The algorithm can give equal consideration to the smaller modes as well as the larger ones, and expansion can be based on properties of the mode itself, not the global average.

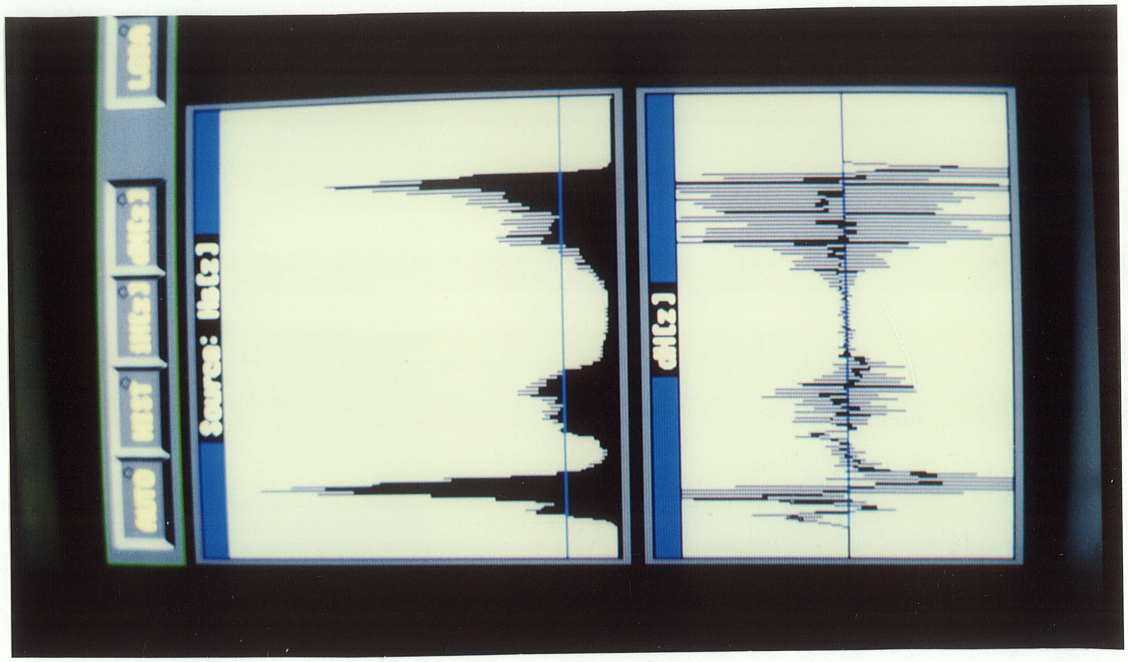
STEP 3: The first procedure in simple histogram fingerprinting, after capture and evaluation, is to filter the histogram data with a 1-D equivalent of median filtering which will be referred to as <u>Local Grey Scale Averaging</u> (LGSA). A peculiarity was observed in the enumeration of the histogram when using our equipment. The histogram array is very "noisy" and not a smooth function as expected. Studying the histogram data revealed that the odd and even grey-levels contained the same pattern but that one always contained more pixels than the other. This is unexpected, since there seems to be no difference in signifigance between odd and even grey-levels. This problem is eliminated once LGSA is applied.

An algorithm was designed to take the original histogram and smooth the grey-level plot so that only the larger trends remain. Instead of using a 3x3 element, a single 3 element filter is used because the histogram is one dimensional. The algorithm takes each grey level in the source histogram and averages it with its nearest neighbor on each side. There is one modification to this however. While a 2-D filtering process contributes to the next operation, the new data set for LGSA is stored in a temporary array before it is copied back in to the original array, preventing the new value from affecting the next grey level. If not, the results may become skewed, especially if more than one pass is performed. The following calculation is performed for each grey-level in the histogram:

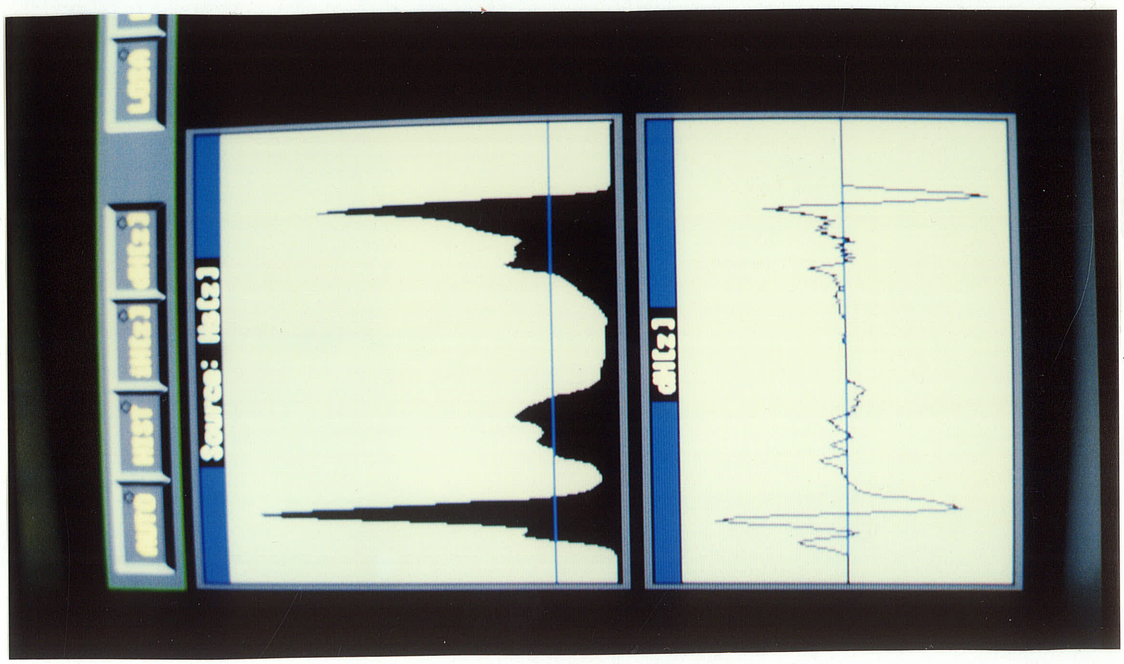
$$\text{new } H[z] = (H[z-1] + H[z] + H[z+1]) / 3$$

Only the major trends in the histogram are of interest to the regrading process, so smaller perturbations are first removed by a filtering procedure. The procedure is designed to perform a single pass averaging. If the "noise" component is large then LGSA can be applied iteratively until only the major components remain. LGSA converges rapidly after three iterations so the number of passes required, even for very noisy histograms, is less than five.

When the first derivative array is plotted on a graph the effect of noise in the histogram is very evident, as shown in Figure 7(a). The spikes in the plot cause difficulty in finding maxima and minima. As the trend plot approaches zero the spikes cause false zero crossings which are interpreted to be maxima or minima in the histogram. When LGSA is applied and the derivative array is then recalculated, the graph is much smoother as shown in Figure 7(b). LGSA is applied until the trend array has no spikes and only smooth transitions from positive to negative remain. The new derivative can now be used to identify the major trends in the histogram by finding the zero crossings. These zero crossings will mark all peaks and valleys in the histogram.



a) Before LGSA



b) After LGSA

Figure 7. Histogram of example image

STEP 4: The next step is to determine the location of the maxima and minima in the distribution. The first derivative of the grey-level histogram is calculated. In the continuous case the maxima will correspond to a zero point where the derivative switches from positive to negative and a minimum will correspond to a zero point where the derivative changes from negative to positive. Each pair of minima will define the beginning and end points of a mode in the histogram.

In the discrete case it is not feasible to simply detect the zero points because they may fall between quantization levels. After the histogram has been smoothed by several passes of LGSA, the first derivative of the histogram is recalculated. The algorithm is modified to detect the change of sign instead of zero points, and the maxima and minima are approximated to the nearest quantization level. There are three possible states for the first derivative. A change from one state to another indicates the presence of a zero crossing.

1) if (H[z+1] > H[z]) dH[z] > 0

2) if (H[z+1] = H[z]) dH[z] = 0

3) if (H[z+1] < H[z]) dH[z] < 0

Where H[z] is the histogram array and dH[z] = H[z+1] - H[z] and is an approximation for the derivative of H[z].

In a continuous function the maxima and minima are given at the points where the first derivative is zero. Since we do not have a continuous function the zero points may fall between quantization levels. Therefore, rather than searching for zeroes in the data, the algorithm finds the maxima and minima by determining where the derivative changes sign (zero crossings). The minima are grouped into pairs where each minimum represents the end of one mode and the start of the next, characterizing all modes, no matter how large or small they may be. The enumerated modes can then be subjected to a set of rules or heuristics to decide which of them need to be merged.

3.4 Trend Analysis

STEP 5: Not all modes will contain useful image information. Areas such as a uniform background will appear as a mode but contain no significant information. Modes of this type appear as very tall sharp spikes in the histogram which indicates a large number of pixels with very few grey levels. As a mode becomes narrower and more acute the amount of information which can be enhanced becomes less. Conversely, if a mode becomes increasingly wider and flatter, the contrast information increases, but it begins to approach an equalized state. Less enhancement is possible because less is required. Clearly, as the characteristics of a mode approaches either extreme case, the less it can be enhanced by expansion.

There is a size limitation on the smallest modes allowed, because otherwise the algorithm would admit, those which are only a single pixel in size. A mode can not be included unless it meets minimum requirements for magnitude and bandwidth (BW), for example, a magnitude of 100 pixels and a BW of 32 grey levels. All modes meeting these requirements will be considered for grey-level regrading.

To form a description of the basic characteristics of a mode and how it relates to the regrading process, each mode is approximated by a triangle whose points are represented by the two minima forming the base, and the one maximum forming the tip of the triangle. The slope and magnitude of each line can be represented by a vector quantity as shown in Figure 8. The two sides are represented by the vectors V1 and V3 corresponding to the rise and fall of each side. The base of the triangle, formed by the vector V3 is simply the sum of V1 and V2. Only vectors V1 and V2 are required to represent an single mode since the third (V3) can be calculated.

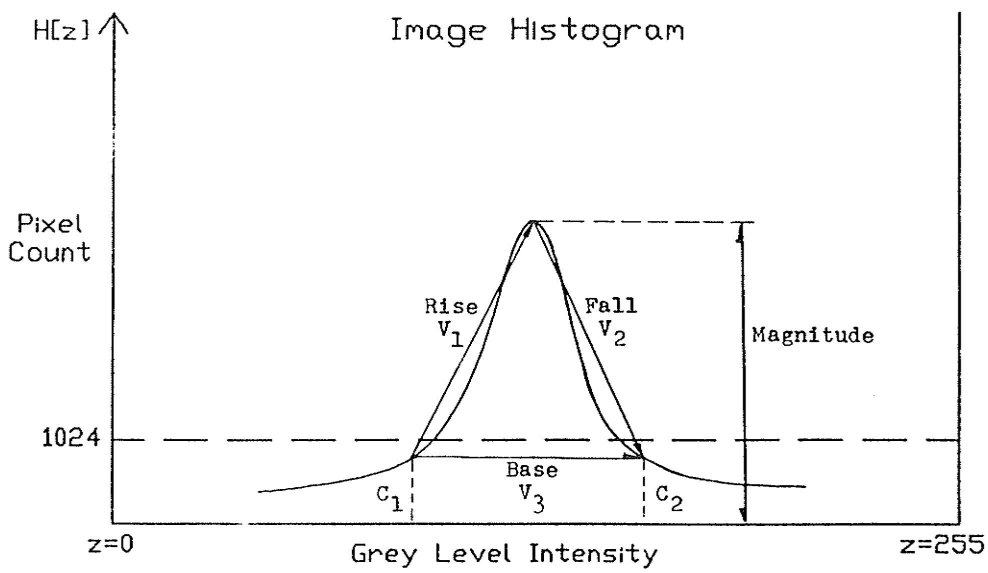


Figure 8. Vector Representation of a Single Mode.

If the dH/dz is positive the grey level is on the rising edge of a mode, and if dH/dz is negative then z is on the falling edge of a mode. In most cases this will work well, but occasionally the slopes of the edges are so gradual near the outer edges of the mode that the algorithm may continue including grey levels in the mode further than is necessary (due to the normal distribution of the mode). The ends contain very few pixels and occupy a large range of grey levels. If they are included in the expansion they will severely limit the available quantization levels in the histogram. An example of characterization error is shown in Figure 9 below. The first mode in the histogram is improperly fingerprinted. Rather than searching for a change in the sign of the slope, it is preferable to mark the point where the slope falls below a certain threshold.

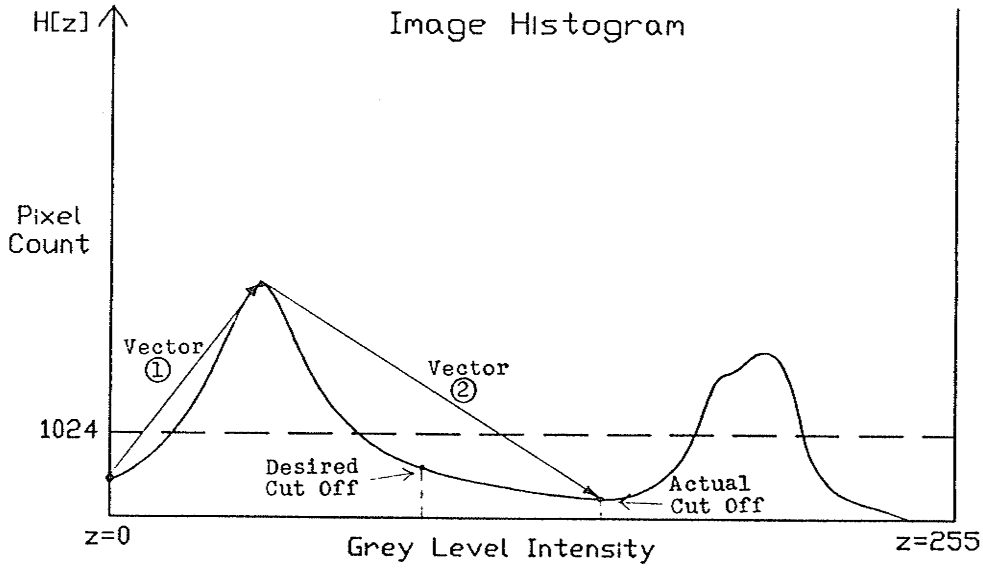


Figure 9. Mode Characterization Error.

In addition to finding maxima and minima the algorithm must be able to deal with these situations. The algorithm has a threshold which creates a dead band around zero. This gives the algorithm a degree of hysteresis so that only significant slopes can trigger the selection process. The algorithm was modified to calculate the first derivative of H[z] and compare its magnitude with a threshold value. This results in a more accurate marking of the modes at the base.

Thresholding can also cause a problem if noise in the histogram forces dH[z] to momentarily fall below the threshold value. The largest component of noise tends to occur when the pixel counts are also large, at the top of the peaks. Even with LGSA there can still be some degree of perturbation near the peak of the mode. On this point of the histogram the modes stop sharply with little or no flat area, so applying a threshold here would have little value. Also, modes in this area have a large degree of intersection. When Heuristics, described in the next section, are applied, these modes will always be merged with another mode. The threshold value must change depending on the height of the histogram.

This is eliminated by creating a threshold which is proportional to the magnitude of the grey level. As the magnitude of the grey level increases the requirement for a mode to be distinct also increases. If the threshold is proportional to the magnitude of the peak then the algorithm can automatically adjust for different cases, and the resulting reliability of the algorithm is greatly improved.

The conditions for thresholding are:

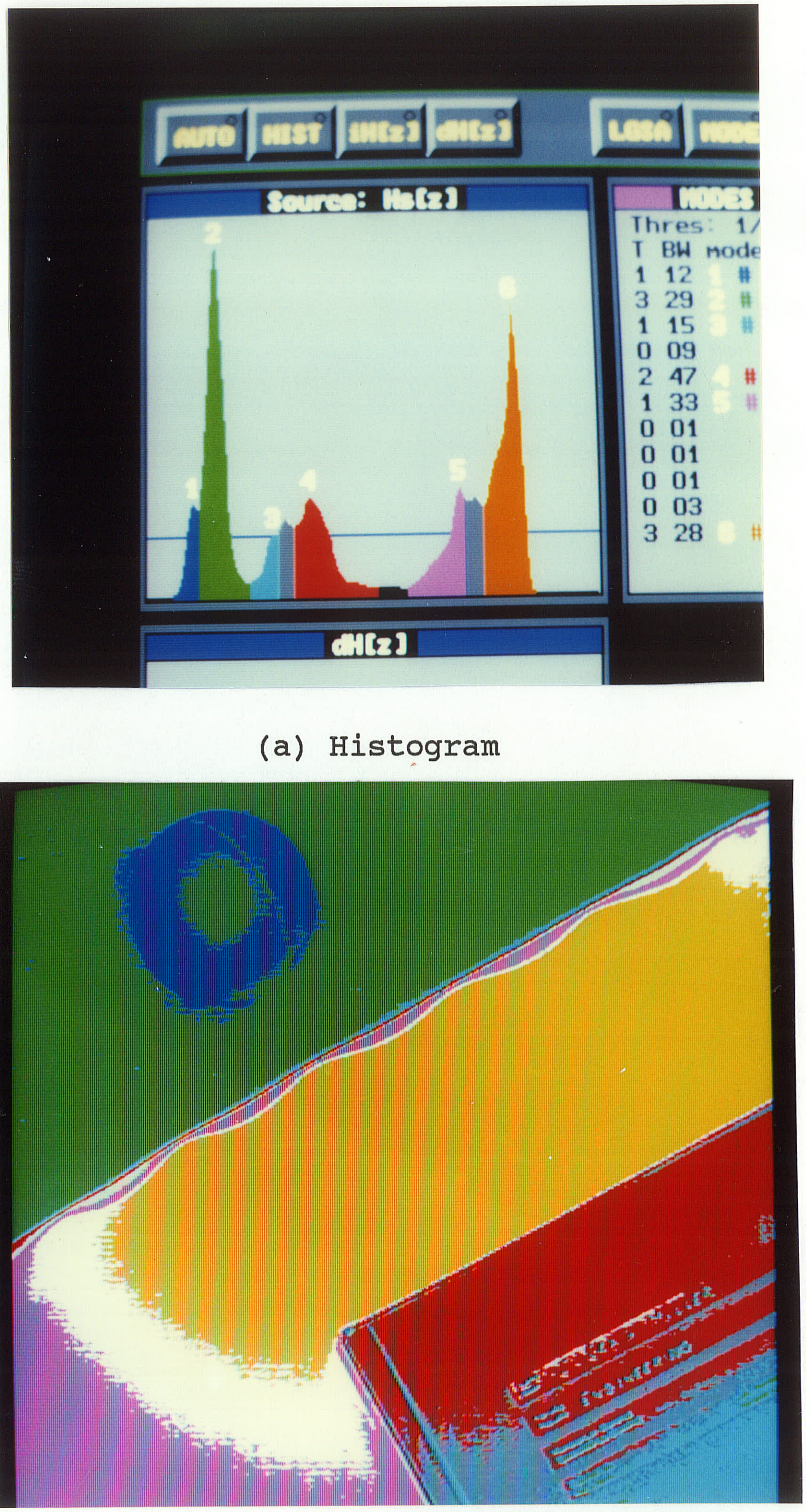
IF H[z] > HAVG THEN threshold = k * HAVG

IF H[z] <=HAVG THEN threshold = 0

Where HAVG = global average = 256K/256 = 1K

and k is a constant percent fraction.

Figure 10(a) and 10(b) shows mode characterization on the PC and the corresponding pseudocolor image.



(a) Histogram

(b) Corresponding pseudocolor image.

Figure 10. Mode Chracterization

3.5 Mode Merging Using Heuristics

STEP 6: Depending on conditions it may not be desirable to classify the range between two minima as a distinct mode. At some minima the number of pixels may still be relatively large. This indicates that there are a large number of pixels which have the same grey levels in two separate modes. In order for a mode to be distinct, it must have a maximum which is sufficiently larger than the neighboring minima on either side. Otherwise the degree of intersection between modes, as discussed earlier, is too large for them to be considered for independent expansion. A set of heuristics must be included in the algorithm so that modes which do not meet certain criteria can be eliminated. Also, those modes which intersect one another to the degree that they cannot be considered separately, must be merged to form a new mode. Figure 11 is an example of a complex mode.

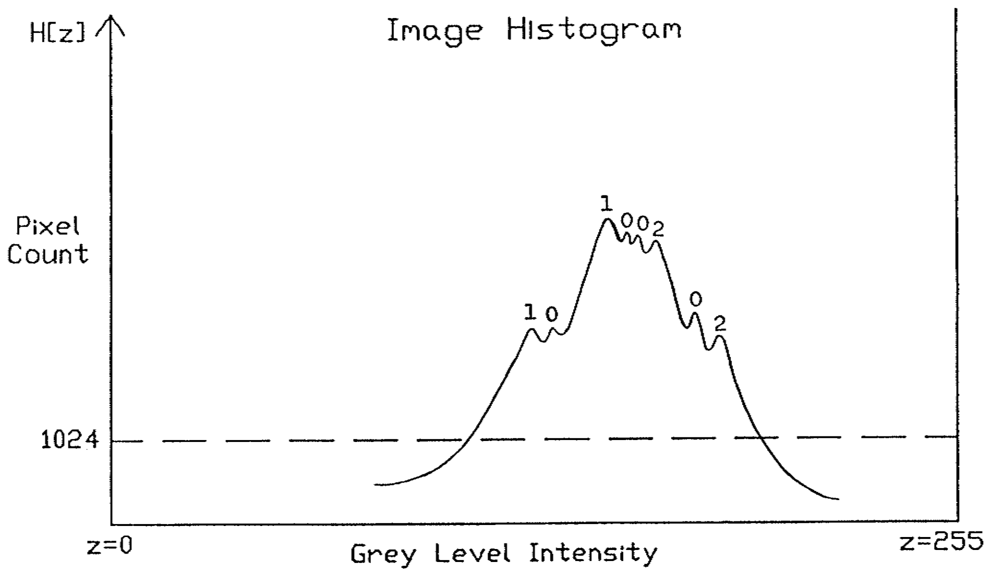


Figure 11. Example of a Multi-Peak Mode.

To characterize each mode, the vertical component of the two vectors is compared to a single threshold value, determined subjectively through experimentation. The result of this comparison is a binary value indicating greater than or less than the threshold. The total number of combinations resulting from these comparisons is 2 x 2! = 4 possible states. These possible states become the four rules which classify the modes:

RULE 1: (Below Threshold)

IF mode n has a rise < the threshold
AND mode n has a fall < the threshold
THEN mode n is classified TYPE 0.

RULE 2: (Black Skewed)

IF mode n has a rise > the threshold
AND mode n has a fall < the threshold
THEN mode n is classified TYPE 1.

RULE 3: (White Skewed)

IF mode n has a rise < the threshold
AND mode n has a fall > the threshold
THEN mode n is classified TYPE 2.

RULE 4: (Distinct)

IF mode n has a rise > the threshold
AND mode n has a fall > the threshold
THEN mode n is classified TYPE 3.

Only a type 3 mode is considered to be distinct. To separate the distinct modes from the mode data, a set of heuristics must be applied to each mode. The outcome should be one of the following.

1) Accept it as a mode.

2) Discard it as insignificant

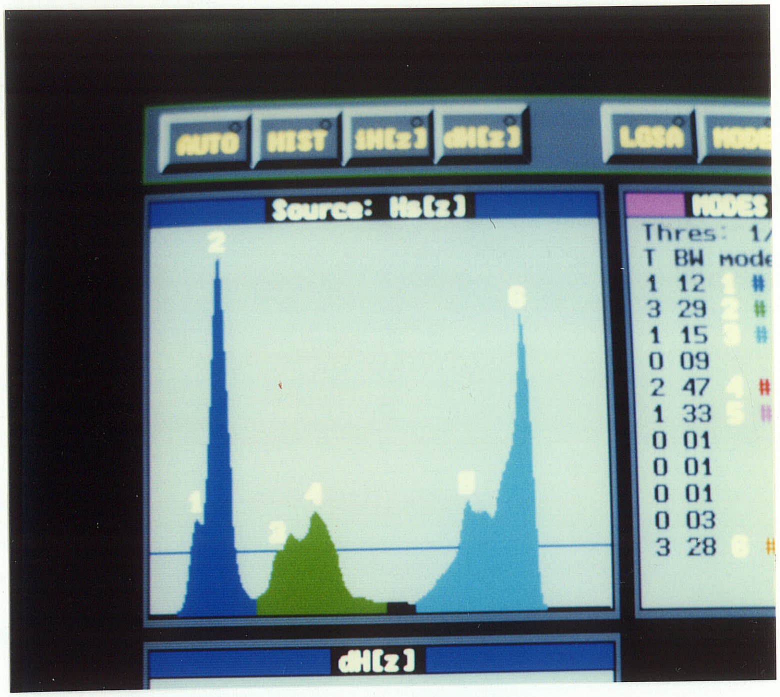
3) Merge it with mode [n-1]

4) Merge it with mode [n+1]

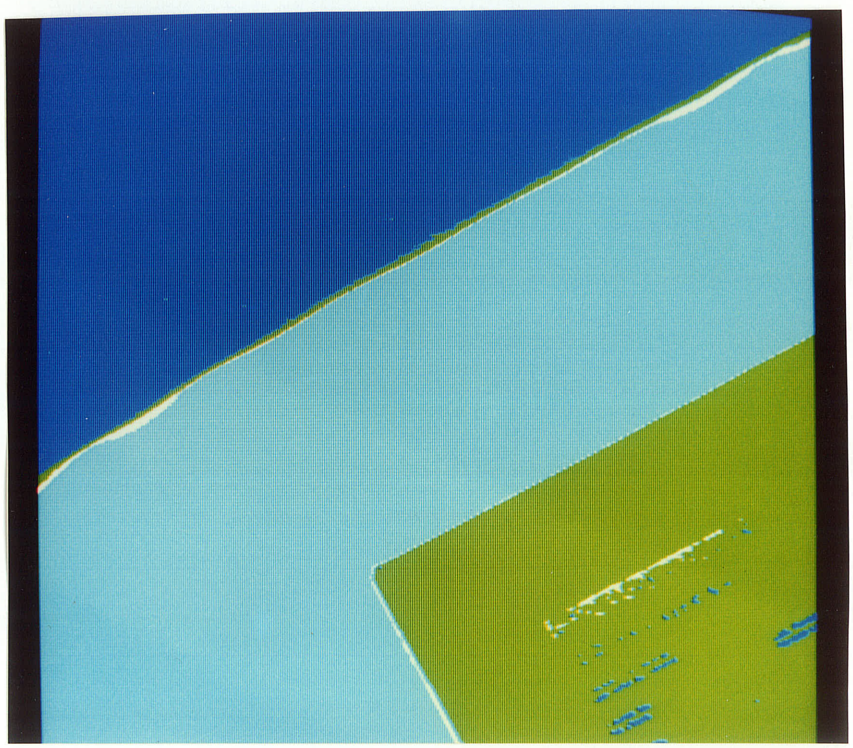
The example shown in Fig 11, characterizes almost all of the merging conditions. Type 0 modes are absorbed in the merging process. The first mode (type 1) is merged forward until it reaches a type 2 or type 3 mode. All type 0 and type 1 modes are absorbed in the merging process regardless of their number until a type 2 is reached, creating a type 3. The merging process is not completed in the first step because a type 2 mode is found in the next 2 iterations, so the process merges backward until it finds a type 1 or type 3 mode. When this is complete it begins to iterate forward again. The entire process is shown below. The arrow represents the current element in the list that the algorithm is processing.

```
1) 10100202
   ^
2)  1100202
    ^
3)   100202
     ^
4)    10202
      ^
5)     1202
       ^
6)      302
        ^
7)      302
         ^
8)      302
          ^
9)      32
         ^
10)     3
        ^
```

Figure 12(a) shows mode merging of Figure 10(a) and 12(b) is the corresponding pseudocolor image.



(a) Histogram



(b) Pseudocolor image.

Figure 12. Mode Merging

3.6 Q FACTOR

Modes are characterized using the starting point (origin) and two vector quantities, one describing the rising edge and the other describing the falling edge of the mode. The relative "sharpness" of a mode can be describe using a term adopted from filter theory called Q factor. The Q Factor can be calculated by comparing the average value of the magnitude of the vertical components of vectors V1 and V2 with the magnitude of horizontal component of V1+V2.

$$Q\ Factor = |V1y| + |V2y| \ / \ |V1x| + |V2x|$$

If a filter had a sharp cutoff point it is said to have a high Q factor. However, if the cutoff is mild it has a low Q factor. For the purposes of mode characterization it is used to describe the sharpness of the mode. The Q factor for a mode can be calculated as the magnitude of the mode (peak height) divided by the width of the mode in quantization levels. An example of two modes, one with a high Q Factor and one with a low Q Factor are shown in Figure 13. If Q is a small number then that describes a mode which is very flat. This kind of mode would not be affected by histogram redistribution since the pixels are already distributed more or less equally. Looking at the other end of the scale, if Q is a very large number, the mode may be expanded but probably would not result in a great improvement since there are very few grey levels to expand.

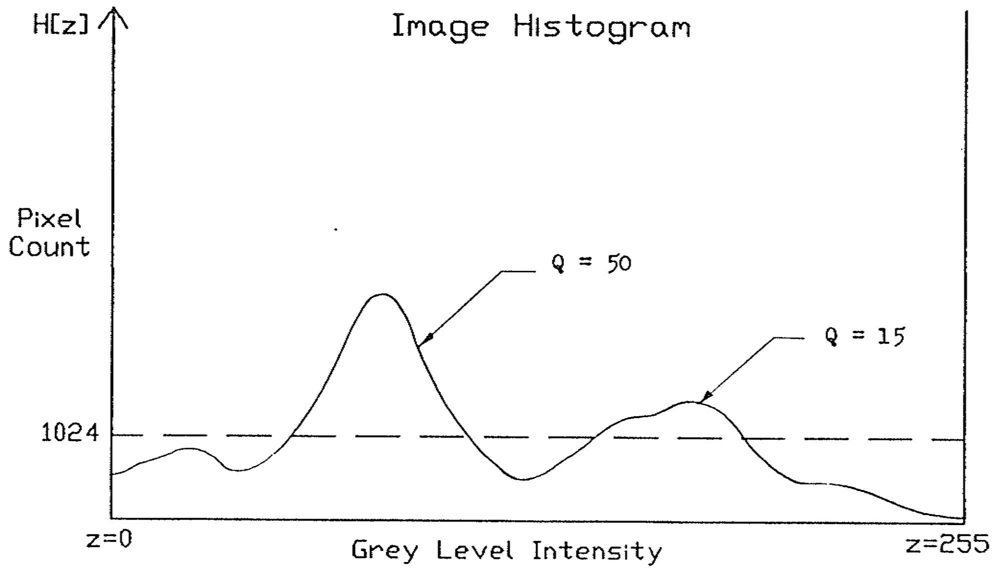


Figure 13. Example of Modes with Large and Small Q Factor.

The Q Factor therefore indicates how much enhancement is possible in a particular mode. Modes with Q Factors at either extreme are poor candidates for expansion, whereas those whose Q Factors are in the medium range, are good candidates. Although this concept was not fully implemented in this thesis presentation, it could be a topic for further improvement of the heuristics. For example, a mode which has a very high Q Factor near zero grey level, for images in space, is primarily the result of large black backgrounds. Heuristics may be implemented to instruct the procedure to leave this mode unchanged since it cannot offer any improvement to the image while the expansion of other modes may be optimized instead.

4. HISTOGRAM REGRADING

Extracting knowledge from the input image histogram was explained in the previous chapter. In the mode merging process, knowledge about the knowledge (meta-knowledge) was used to determine what knowledge will be used to guide the regrading process. In this chapter it will be shown how this knowledge can be used to indirectly segment the image into distinct modes by the scale-space approach. The modes are then regraded, using a local contrast stretching scheme based on local criteria. After regrading, the new stretched modes are fitted to the output histogram which is then applied to the image, and results in a new output image.

In conventional non-spatial histogram modification techniques the histogram of the input image is regraded so that it is forced to an arbitrary abstract distribution. The transformation is usually performed using the integral of the histogram called the cumulative histogram or Cumulative Density Function (CDF). In standard histogram equalization this CDF is a straight line corresponding to a linear transformation. As discussed in the literature, other forms of CDF which are non-linear, such as hyperbolic functions, also give good results. In more elegant non-spatial techniques a combination of linear regrading (contrast stretching) and a linear transformation (histogram equalization) were used to alleviate some of the problems associated with the use of histogram equalization.

The Adaptive Histogram Regrading (AHR) technique presented here does not perform any spatial segmentation, but rather segments the histogram into separate modes which in turn indirectly segments the image by the scale-space approach [1]. It is possible to start with the entire histogram and, to a certain extent, separate it into histograms of the disjoint regions. These regional histograms can now be treated as separate entities and expanded independently, based on local characteristics. Then the equalized regions can be recombined to create a new image of improved quality over the conventional histogram modification.

The great advantage obtained from the AHR technique is that a histogram can be expanded differently depending on the characteristics of the modes. Using heuristics it is possible to make inferences about the image based on the quantity, location, size and shape information extracted from the histogram. These inferences can be used to guide the expansion process so as to optimize the mapping of pixels no matter what type of input image is used. The general approach to histogram regrading has seven steps:

1) Segment the image using scale-space method.
2) Regrade each mode using local grey-level stretching.
3) Determine the new width of the stretched mode.
4) Perform optimal fitting of new modes.
5) Fit non-selected ranges.
6) Program output Look-Up Table (LUT).
7) Apply output LUT to the input image.

4.1 Image Segmentation using Scale-space

It has been observed in the literature that a single point in the 1-D scale corresponds to a contour or set of contours in the 2-D space. A relief map is an example of this correspondence, where a single value on the elevation scale traces out a contour on the map. The effect, referred to as scale-space, has also been successfully used to predict the finger-print of a histogram in terms of normal distributions by Carlotto [1]. If the 1-D point is extended to a range of sequential points, then the corresponding 2-D contours become regions. On the relief map this would be the same as filling in the region between two contours, or the region within the innermost contours (such as hilltops). This correspondence was successfully exploited by Rosenfeld and Davis [16] to segment an image into distinct regions. The resulting image is a map of the original image using single grey-levels.

For Adaptive Histogram Regrading, image segmentation is done using scale-space in a manner similar to the Rosenfeld and Davis approach [16]. The difference is that no thinning is applied to the histogram. Instead, information about the selected modes is used to mark each range of grey-levels which will be segmented. The corresponding image will be indirectly segmented as a result. When the image segmentation is coloured using pseudo-colour some regions appear very fragmented. However,the segmentation is exact because no one pixel can belong to more than one selected mode. The mode characterization process is an example of scale-space segmention. If mode characterization is applied to the original satellite image in Figure 1 an image with pseudo-colour is produced as shown Figure 14.

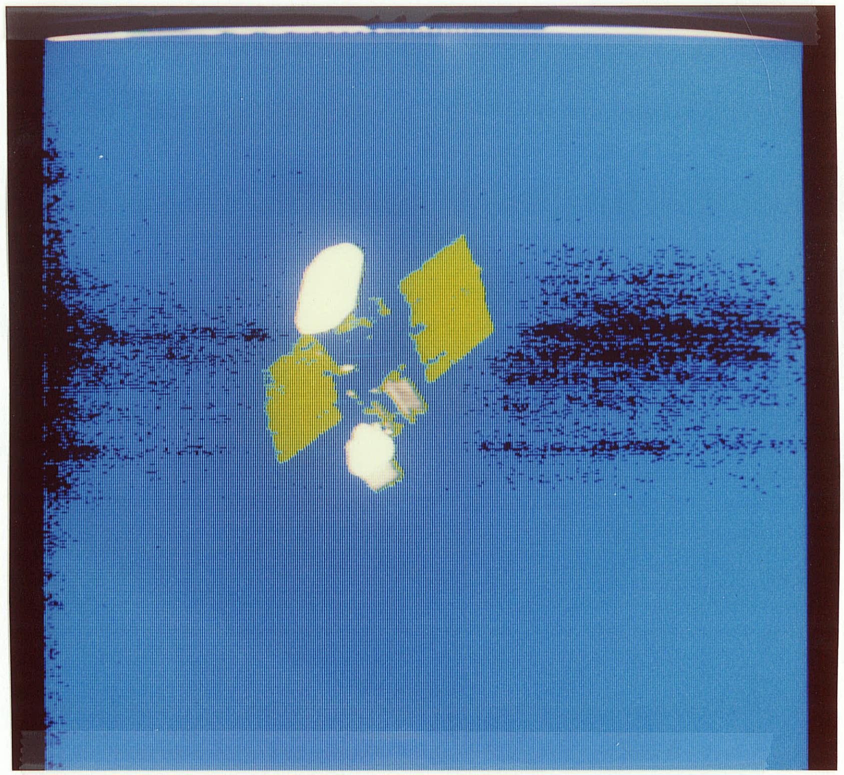


Figure 14. Scale-Space Segmented Image in Pseudocolor.

4.2 Local Histogram Stretching

At this point the selected modes are part of the knowledge base for regrading. How to go about the individual regrading however, is a different matter. At this stage there are a number of different ways that the selected modes can be regraded. Many of these methods came to mind during implementation and testing. However, the scope of this thesis could only allow for implementation and testing of a single method. It is of value to mention some of the other methods because they could be good topics for extended study of Adaptive Histogram Regrading:

1) Treat each selected mode like a small histogram. Then apply standard histogram equalization to the range of grey-levels corresponding to the mode. The equalized value used is the average pixel count within that range. This would be the simplest method to implement on the computer because it is simply a series of linear histogram modifications. Since both stretching and compression is applied, the new mode has the same quantization width as before and no fitting is required. The main advantages of this technique are simplicity and speed. Output images, however, would not be optimal because compression is not necessarily performed on grey levels of lowest performance and information may be lost.

2) Perform a growing operation on the range of each selected mode so that the entire histogram is segmented. Then apply histogram equalization to each piece as described in method 1. This would have the advantage of utilizing the full dynamic range, but may not work well if the histogram is highly skewed.

3) Regrade each mode using grey-level stretching, but no compression. Then perform a fit on each stretched mode to ensure no two modes are overlapping. If compression is required to prevent clipping, scan the histogram for unselected ranges and perform compression. This has the advantage of optimizing the stretching and compression so that information loss is minimized. The drawback is its complexity of implementation, particularly for the mode fitting process and compression.

In the work presented here, the third implementation of AHR was used. Although it had high complexity, it offered the most optimal image quality, and the original goal of the work was to minimize information loss.

Histogram equalization expands certain ranges of the histogram and compresses others. The compression is performed to maintain the same number of total quantization levels, otherwise the new histogram may have a greater dynamic range than the hardware can realize. Without compression many images will be clipped at either end of the grey scale range. If, for example, an expanded histogram has a total dynamic range of 270 grey levels and the hardware is capable of only 256 grey levels, then parts of the histogram must be compressed by 14 grey levels to avoid clipping.

Local grey-level stretching is performed using the same one-to-many mapping as in histogram equalization, because of its simplicity and efficiency. In AHR, only stretching of the modes is performed initially. The goal is to perform minimal compression of grey levels and minimize information loss. In some cases the overall dynamic range may exceed 256, making it necessary to perform some compression to prevent the histogram from clipping at 255. Since we are treating each mode in the histogram separately, it is not known how many grey levels must be compressed until all of the stretching has been calculated. The compression of grey-levels to recover quantization levels, normally carried by conventional methods, must be performed after mode stretching and after the regraded modes are fitted to the output histogram. This is because it is not known which or how many grey-levels must be compressed until all the modes have been expanded.

Since the regrading is now a local procedure, each mode is treated like a small histogram. Each of the modes shown in the Figure 15 can now be expanded independently, enhancing their grey scale ranges. Within the boundaries of the mode, the average grey scale value is calculated, which is then used to guide the regrading of the mode. The number of grey-levels for stretching is given by:

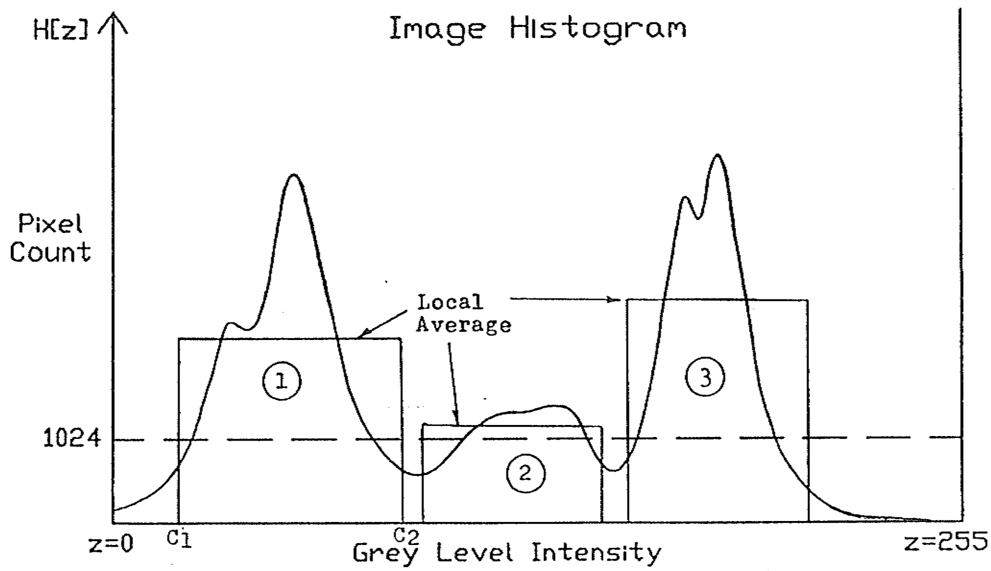


Figure 15. Block Representation of Distinct Modes.

$$\text{new \# of grey-levels} = \frac{\text{grey-level value}}{\text{local average}}$$

where the new number is rounded to the nearest integer.

Figure 15 shows a trimodal image with boxes overlaid on each selected mode. These boxes represent the equalized equivalent of each mode, indicating the current quantization width and local average. The local average is used to stretch the grey-levels with high pixel counts, as in histogram equalization. Unlike histogram equalization, however, the grey-levels with low pixel counts are not compressed, hence the overall quantization range increases.

Each mode is stretched individually until all modes of the histogram are complete. Usually the number of modes is no more than two or three, and very rarely will this number exceed five.

4.3 Expansion Factor

With conventional histogram equalization the stretching and compression of grey-levels is performed on a global basis, and there is no flexibility to adjust these processes. The equalization is controlled by the global average and, if changed, will result in either under-utilization of the dynamic range or clipping at either end. In AHE presented here, the regrading is carried out first and followed by compression. The amount of compression required is determined by the total dynamic range of the expanded histogram and is adjusted accordingly. This provides flexibility in modifying the degree of regrading of each mode. Rather than expand each mode by its local average, the grey-level stretching can be controlled by a proportion of this average.

Normally the proportional factor is 100%, which corresponds to the local average. This can be changed, for example to 90%, which would result in a greater grey-level stretching, or 110% which would result in a smaller grey-level stretching. This factor will be defined as the expansion factor. The relationship between the expansion factor and the expansion is inverse, since the smaller the expansion factor the greater the grey-level stretching and vice versa. It can have any positive value, but practical values fall within the range from 60% to 110%. In some cases, after the expanded modes have been calculated, the sum of all blocks is greater than the total number of levels in the histogram. In that instance the expansion factor must be adjusted to ensure that all the modes will fit in the available space.

4.4 Shifting Stretched Modes

A problem arises when modes are characterized, merged and then expanded by the criteria we have specified. At present the grey-level stretching is centered about the midpoint of the mode. In the process of grey level stretching the total number of quantization levels occupied by the mode is increased (the mode becomes wider). When two modes located in immediate proximity to one another are expanded then the extremities, and sometimes an entire mode can overlap another mode. The blocks, representing the area of the modes occupied in Figure 15 become wider after stretching. Figure 16 shows the block representation of the modes in Figure 15 after grey-level stretching occurs and demonstrates the overlap that occurs with two adjacent modes. The modes are stretched using an expansion factor of

100%, which is equivalent to expansion by the local average. It is clear from the diagram the the blocks are no longer separate from one another.

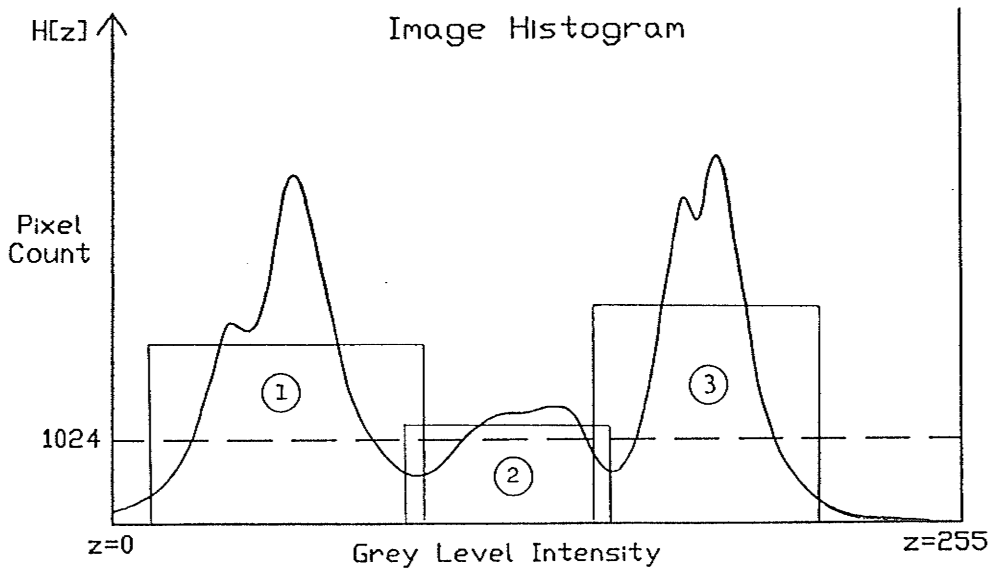


Figure 16. Example of Overlapping Modes.

Overlap conditions are not desirable because they create transitions of grey-levels from one mode to the next, which then appears in the resulting image as a contour line, corresponding to the scale-space relationship between 1-D and 2-D data sets. This transition adds artifacts to the image which would be interpreted by post processing as real data. When the histogram is expanded using the current procedure, the very large peak due to the black background almost completely envelopes the small mode created by the satellite. The resulting image appears confused because of the transition-created artifact in the image around the satellite.

To eliminate this transition point, the expanded modes need to be shifted so that when mapped, there is a smooth transition from one mode to the next. By shifting the second mode to the right, the intensity of the brightest pixels in mode 1 is now a little less than the darkest pixels in mode 2, eliminating the transition.

Example: Shifting Modes in a Bi-modal Image.

Figure 17 shows a histogram for the image of a satellite in space. The histogram for this image is distinctly bimodal with one large mode near zero and another, but much smaller one, immediately next to it. When the system was tested using conventional histogram equalization, it failed to enhance the satellite and expanded the background instead. The image resulting from this grey-level stretching has the background expanded and the satellite untouched. This is the opposite of the desired result and clearly indicates that the size of each mode does not dictate the importance of the mode to the image. It is unsatisfactory to use the global average, such as in conventional histogram equalization to control the grey-level stretching.

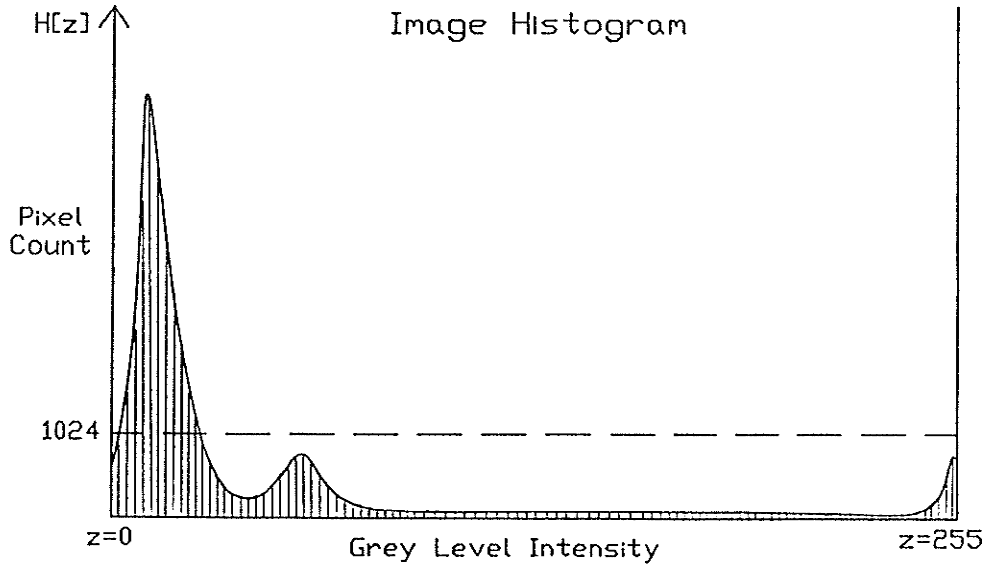
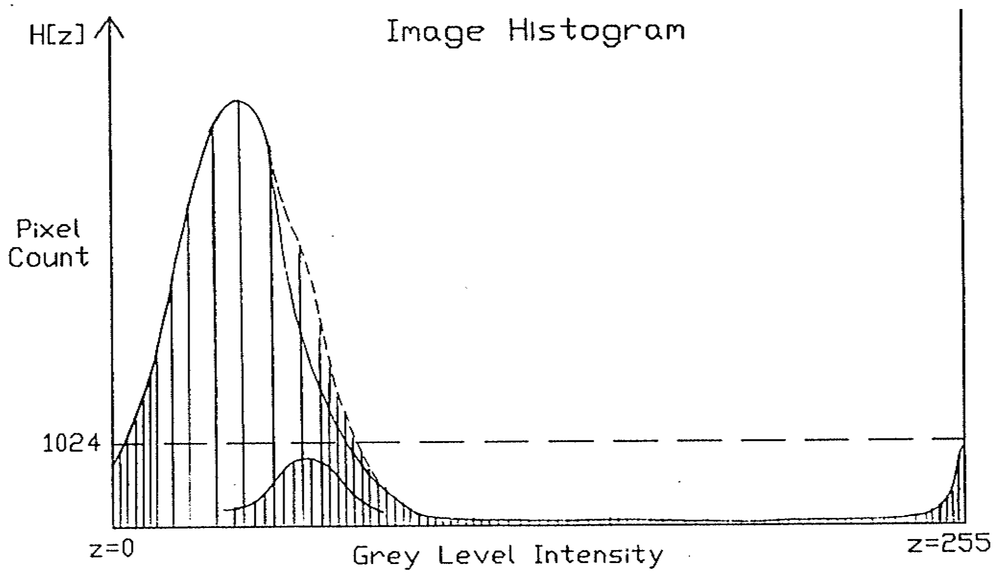
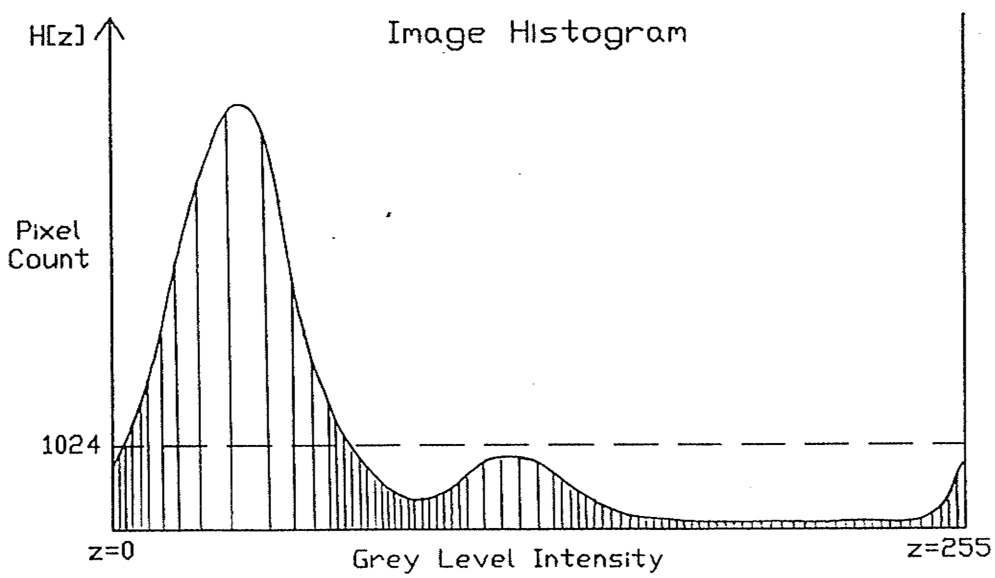


Figure 17. Histogram of the Image of a Satellite in Space.

When the image was tested using AHR without any mode shifting, the grey-level stretching of the larger mode near zero ended up completely enveloping the smaller mode situated next to it as shown in Fig. 18(a). The overall grey-level stretching of the large mode was less than that of histogram equalization but still enough to create a strong transition from one to the other. The only way to maintain the integrity of the image is to shift mode 2 up to end of mode 1 as shown in Fig. 18(b). Point A and B have been shifted to the right (white shift) to prevent mode 1 from enveloping mode 2.



(a) Without Shifting.



(b) With Shifting.

Figure 18. Expanded Histogram of Satellite.

The general approach to mode regrading is:

1) Calculate the amount of grey-level stretching that will occur in each mode (i.e. how wide it will be after grey-level stretching)

2) Sum all the expanded modes and those regions which are to be expanded in order to determine the overall dynamic range.

3) Compare this sum with the number of quantization levels available (in this case 256) and determine if it is more or less. If it is found to be larger, then we have a guaranteed overlay condition and the grey-level stretching of each mode must then be adjusted so that the sum is less than 256. This can be done by increasing the stretching number. If it is smaller, then there is sufficient space to support the grey-level stretching and no changes need to be made.

4) Stack each stretched mode starting at zero on the output histogram so that they are all contiguous and no overlap exists.

5) Apply an iterative relaxation algorithm until each mode reaches an equilibrium and is as close to its original location as possible without any overlap.

Handling conflicting conditions where the grey-level stretching of one mode overlaps that of another mode requires heuristics to guide the shifting process. The modes must be juggled until they all fit in the space without any overlapping. The sum of all stretched grey-level ranges must be less than 256 grey levels, otherwise there is no way to achieve non overlapping conditions without clipping the histogram. Once this condition is met, an algorithm must be applied to the histogram data to shift each mode so that none overlap.

The simplest way to prevent any overlap would be to position the first mode starting at zero, and then stack all other modes consecutively, as shown in Fig 19. Study of the problem reveals a number of conditions that can arise. If, for example, there is a large gap between two modes, simple stacking would unnecessarily close this gap and radically alter the image.

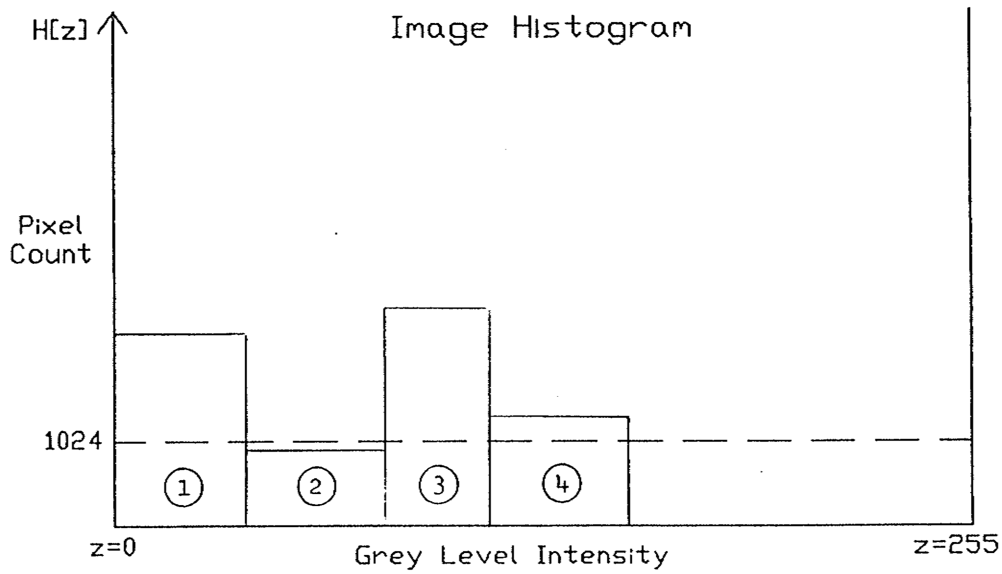


Figure 19. Blocks Representing Expanded Modes Stacked at 0.

Conditions limit how the modes (represented by blocks as shown in Fig. 18 can be shifted, and isolating all special cases can be difficult. Some of them are listed below:

1) A mode is restricted when it has been moved to either limit of the histogram. This occurs when the stop point is equal to the upper limit or the start point is at the lower limit.

2) A conflict condition is reached when the stop point of mode[n] is greater than the start point of mode[n+1]. When this occurs one or both must be shifted.

3) If two modes are in conflict and one of them is

restricted then the other one must be moved.

4) If one of the modes is to be shifted and there is insufficient space to do so then it must be shifted as far as possible.

5) If both modes are possible candidates for shifting, then the algorithm must decide which of the two should be moved.

One method which was considered to resolve which mode should be shifted was to use the cumulative distribution function iH[z]. If the value of iH[z] at the starting point is greater than the theoretical average, then the remapped mode should be shifted right. This would push iH[z] toward a linear distribution as in histogram equalization. This method works in some cases, but falls short when the mode in question is very small. If the mode is small, it has a low total number of pixels. This contributes little to the cumulative histogram which may result in iH[z] having a value lower than the average. This would cause the mode to be pushed in the opposite direction, increasing the overlap in order to increase iH[z] to its normal value.

These are just some of the considerations involved in optimizing the distribution. The problem quickly becomes unwieldy because there are so many special cases to consider. This is not incorporated easily into an algorithm for the computer.

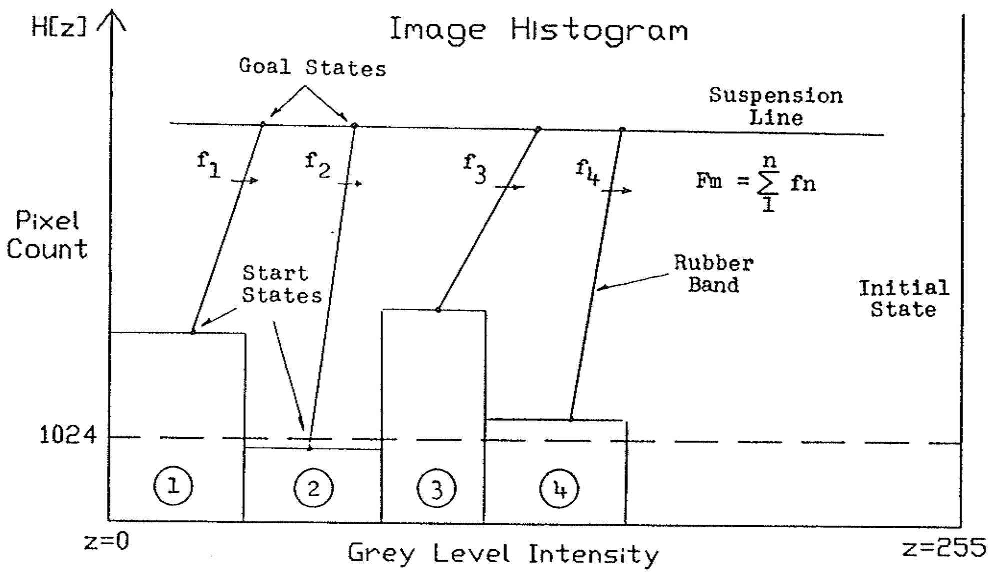
This problem should be approached from another viewpoint. Rather than trying to isolate all the special cases which occur, it would be better to define the goals of the shifting process. These are summed up by just two conditions:

1) The histogram must have no overlapping modes.

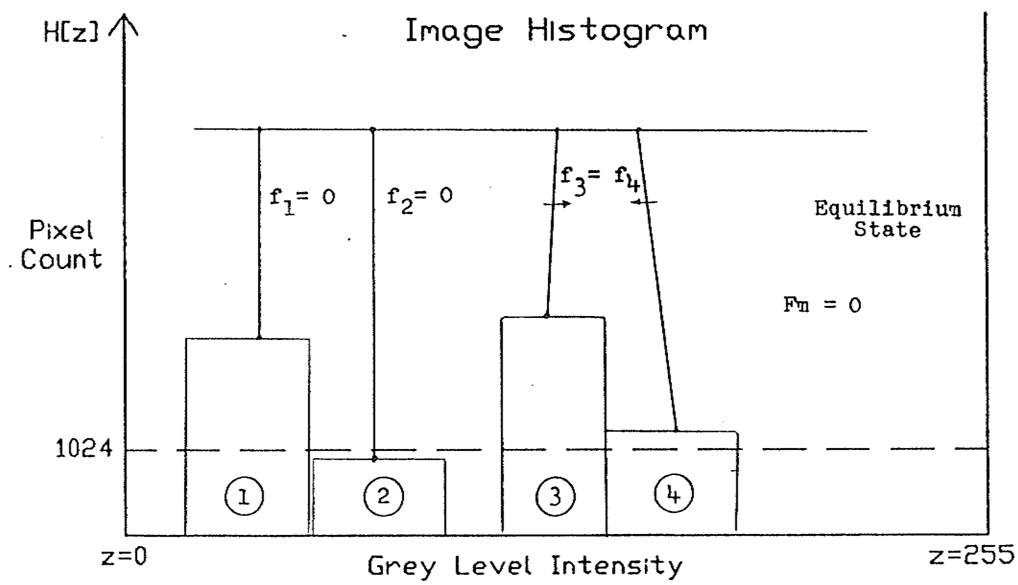
2) Modes should be located as near as possible to their original location.

Modes cannot overlap and they also cannot extend beyond either end of the histogram. These conditions can be simulated by blocks in the physical world. The goal of the algorithm is to place the center of gravity of each block as closely as possible to its original location. This is not always possible since each mode becomes wider when it is expanded. Two modes which are located side by side cannot overlap so it is impossible for both to maintain their location unchanged. An algorithm must place the blocks as closely as possible to the ideal and still avoid an overlap condition.

If we imagine each block being pulled to the ideal by some potential force then this force will be a minimum when the blocks are in equilibrium. This force could pull the block along a frictionless floor until the force dropped to zero as the block approached its ideal point, or until some opposing force from a block pulling in the opposite direction canceled it out. In the real world this could be simulated by attaching a rubber band to each block and releasing it. Since each block contains a different number of pixels, each has a greater or lesser importance to the overall image. This weighting is expressed in the spring constant of each rubber band. This way the bands connected to the larger blocks could generate a greater force than those attached to smaller ones. A representation of the system would be as shown in Fig. 20(a).



(a) Before Relaxation.



(b) After Relaxation.

Figure 20. Representation of Rubber Bands and Blocks

The blocks in Figure 20(b) represent the new modes after they are expanded. The modes have all been stacked starting at zero. The lines attached to each block represent hypothetical rubber bands which pull on each block. The spring constant of each rubber band is given by the mass of each block (given by the area of mode). Each rubber band is attached to the center of gravity of each block at one end and to the target center of gravity at the other end.

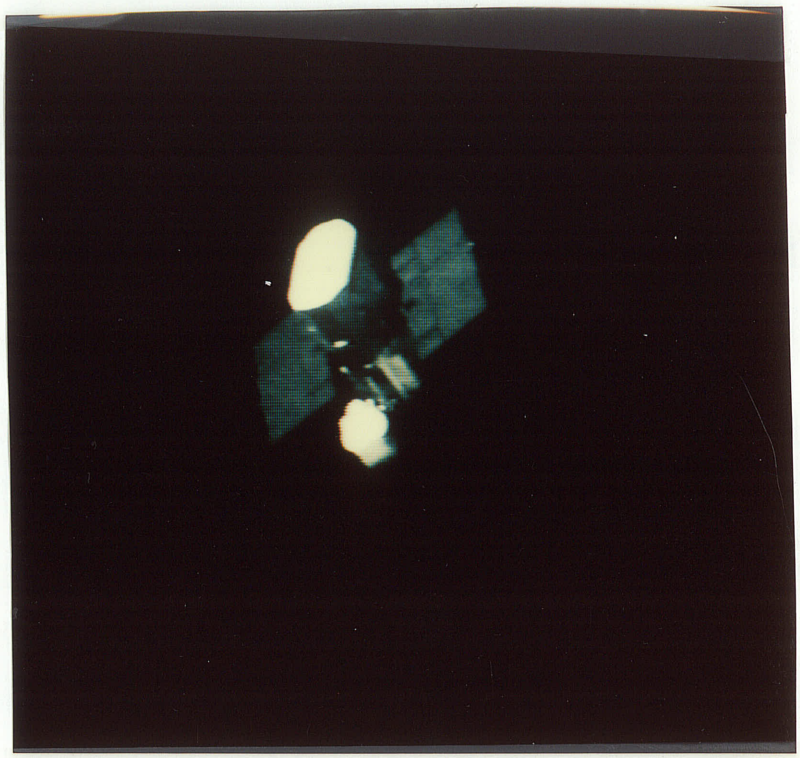
In this system only the horizontal component of the force, and the displacement are used to calculate the force on each block. Blocks would then be stacked starting at z=0 in the histogram and then released. As the blocks slide across the frictionless surface, the rubber bands are relaxed until the system reaches an equilibrium state. In this algorithm only two rules apply.

1) When iterating from first to last, the algorithm determines how many blocks will slide by checking to ensure that the total force Fm is always > 0.
2) If Fm <= 0 at any point then all the blocks to the left of, and including the current iteration are fully relaxed and will not slide any further.

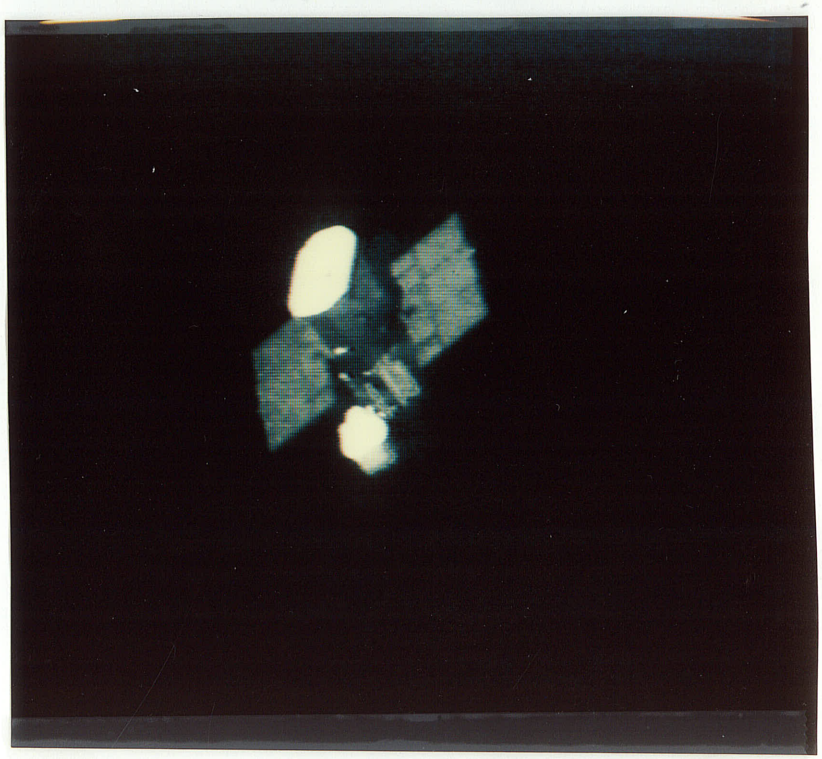
In order to perform the relaxation process in the computer, the algorithm needs a starting point. If the blocks are started at their original position, the algorithm would need to perform a lot of jumping forward and backward. This would not only slow the process but would be difficult to implement. To streamline the process, it would be preferable to stack all the blocks initially at the bottom of the histogram. This way the iteration process could progress in one direction only. The algorithm stops when all the blocks have reached an equilibrium state as shown in figure 20(b).

Once the new locations for the modes have been determined, the next step is to map all the modes into these spaces. Each mode is expanded starting at the new start point as determined by <u>Rubber Band Relaxation</u> (RBR). The pixels are mapped by programming each entry of the input lookup table with the value of its new location. After each mode has been expanded using the input LUT, a feedback operation is performed on the image using the input LUT (or input palette as it is sometimes referred to). This performs the transformation and the new equalized image appears in the buffer and on the display screen.

Figures 21(a) show the image of a satellite in space in its orignal form. Figure 21(b) shows the same image with Adaptive Histogram Regrading applied to the image. The results speak for themselves. Rather than destroying the image as in conventional methods, the image has been enhanced as much as possible without degradation.



(a) Before AHR.



(b) After AHR

Figure 21. Satellite Image.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

1) Although many non-spatial histogram modification techniques have been shown to improve the quality of images, the improvement is highly image dependent. Unless the image characteristics are known prior to the histogram modification, a subjective decision must be made as to the quality of the output image. This precludes these techniques for real-time general purpose applications.

2) AHR requires no a priori knowledge of image characteristics. It produces output results that are in the worst case equivalent to the input image, and in the best case significantly improved over other histogram modification techniques.

3) Currently Adaptive Histogram Regrading is an effective method for enhancement applications in a general purpose environment.

4) Regrading of modes based on local criteria is successful in preventing the problems associated with over-stretching of grey-levels that often occurrs when using conventional histogram equalization.

5) Simple histogram fingerprinting allows the process to

successfully identify and enhance modes which were previously too small and often compressed when using conventional techniques.

6) Although current computational requirements do not permit it, if implemented in dedicated hardware, both histogram evaluation and AHR can be performed in a real-time environment.

5.2 Recommendations

At this stage there are a number of different ways that the selected modes can be regraded. Many of these methods came to mind during implementation and testing. However, the scope of this thesis could only allow for implementation and testing of a single method. It is of value to mention two other methods because they could be good topics for extended study of Adaptive Histogram Regrading:

1) Treat each selected mode like a small histogram. Then apply standard histogram equalization to the range of grey-levels corresponding to the mode. The equalized value used is the average pixel count within that range. This would be the simplest method to implement on the computer because it is simply a series of linear histogram modifications. Since both stretching and compression is applied, the new mode has the same quantization width as before and no fitting is

required. The main advantages of this technique are simplicity and speed. Output images, however, would not be optimal because compression is not necessarily performed on grey levels of lowest performance and information may be lost.

2) Perform a growing operation on the range of each selected mode so that the entire histogram is segmented. Then apply histogram equalization to each piece as described in method 1. This would have the advantage of utilizing the full dynamic range, but may not work well if the histogram is highly skewed.

DI-IRIS provides software routines for histogram evaluation but there are two slow for real-time applications. Since the routine requires to much time (about 1 second to process), this would need to be changed to implement AHR in real-time, but for demonstration it is sufficient.

BIBLIOGRAPHY

1. Carlotto, Mark J. (1987). "Histogram Analysis Using a Scale-Space Approach", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, 1: 121-129.

2. Castleman, K.R. (1979) "Digital Image Processing", Englewood Cliffs NJ: Prentice-Hall.

3. Chochia, Pavel A. (1988). "Image Enhancement Using Sliding Histograms", Computer Vision, Graphics, and Image Processing, 44: 211-229.

4. Ekstrom, M. P. (1984). "Digital Image Processing Techniques", Academic Press, Orlando, Florida.

5. Frei, Werner (1977). "Image Enhancement by Histogram Hyperbolization", Computer Graphics and Image Processing, 6: 286-294.

6. Gonzalez, R. C. and Wintz, P. (1978), "Digital Image Processing" (Applied Mathematics and Computation; no. 13), Addison-Wesley, Reading, Mass.

7. Hochberg, Julian E. (1978), "Perception", Prentice-Hall, Englewood Cliffs, N. J. pp: 105-110.

8. Hummel, Robert A. (1975). "Histogram Modification Techniques", Computer Graphics and Image Processing, 4: 209-224.

9. Hummel, Robert A. (1977). "Image Enhancement by Histogram Transformation", Computer Graphics and Image Processing, 6: 184-195.

10. Kautsky, Jaroslav, Nichols, Nancy K. and Jupp, David L.B. (1984). "Smoothed Histogram Modification for Image Processing", Computer Vision, Graphics, and Image Processing, 26: 271-291.

11. Knoll, Thomas F. and Delp, Edward J. (1986). "Adaptive Grey Scale Mapping to Reduce Registration Noise in Difference Images", Computer Vision, Graphics, and Image Processing, 33: 129-137.

12. McCollum, A.J., Bowman, C.C., Daniels, P.A. and Batchelor, B.G. (1988). "A Histogram Modification Unit for Real-Time Image Enhancement", Computer Vision, Graphics, and Image Processing, 42: 387-398.

13. O'Gorman, Lawrence (1988). "A Note on Histogram Equalization for Optimal Intensity Range Utilization", Computer Vision, Graphics, and Image Processing, 41: 229-232.

14. Pavlidis, T. (1982) "Algorithms for Graphics and Image Processing", Computer Science, Rockville, MD.

15. Pizer, Stephen M., Amburn, E. Philip, Austin, John D., Cromartie, Robert, Geselowitz, Ari, Greer, Trey, Romeny, Bart Ter, Haar, Zimmerman, John B. and Zuiderveld, Karel (1987). "Adaptive Histogram Equalization and Its Variations", Computer Vision, Graphics, and Image Processing, 39: 355-368.

16. Rosenfeld, A. and Davis, L.S. (1978). "Iterative Histogram Modification", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-8, 4: 300-302.

17. Vossepoel, Albert M., Stoel, Berend C. and Meershoek, A. Peter (1988). "Adaptive Histogram Equalization Using Variable Regions", IEEE : 351-353.

18. Wang, David C.C., Vagnucci, Anthony, H. and Li, C.C. (1983). "Digital Image Enhancement: A Survey", Computer Vision, Graphics, and Image Processing, 24: 363-381.

19. DT/Image-Pro Interactive Image Processing System. Version 1.6 (1986). Data Translation Inc. User Manual.

20. DT-IRIS SP0210 Version V01.00 (1986) Data Translation Inc. User Manual.

GLOSSARY OF TERMINOLOGY:

A/D CONVERTER:
Analog/Digital converter which changes analogue signals into digital signals.

BINARY SYSTEM:
A numerical system with only two digits: 0 and 1. Also called the two digit system.

BIT: A unit of information consisting of a single binary digit.

BYTE:
Information consisting of 8 bites.

COMPUTER:
An electronic unit capable of performing substantial computation and data processing.

FRAME BUFFER:
Memory storage for digitized image data.

GREY SCALE:
A single level of image brightness or greyness, described by a binary number.

HARDWARE:
The physical part of a computer.

IMAGE SENSING:
Electronic recognition of patterns.

K:
"Kilo" - the symbol for 1000 (in computer terminology it refers to a quantity of 1024 or 2^10). A store capacity of 2K bits thus contains 2048 bits.

M:
"Mega"- the symbol for 1,000,000. (in computer terminology it refers to a quantity of 1024 x 1024 or 2^20).

MICROPROCESSOR:
A mass-produced microprocessor manufactured for a range of different areas of application.

PIXEL:
Short form for "Picture Element". The single smallest unit of a digitized image frame, whose value is represented in a single byte.

RAM:
Random-Access-Memory. A memory in which information

can be stored and later erased. Normally used for data storage in a processor system.

SOFTWARE:
Programs, procedures and data pertaining to the operation of a computer system.

WORD:
Information consisting of 2 bytes or 16 bits.