A PATTERN RECOGNITION APPROACH TO LOSSLESS IMAGE COMPRESSION

by

SCOTT LARAN JOHNSON

B.S., Kansas State University, 1985

A REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

Approved by:

William Hankley

Major Professor

TABLE OF CONTENTS

Index of Figures

Index of Tables

Acknowledgments

At this time I wish to acknowledge assistance which I have received during the course of this research. First, I wish to note the help I received from employees at CIS. Specifically, Joe and Alex for help in debugging and allowing me to exchange ideas with them. A special thanks goes to Dr. Paul Fisher for his help and guidance throughout the course of the research effort.

I would also like to acknowledge my committee: Dr. William Hankley for helping me to put the results in an understandable form. Also, thank you to the other members: Dr. Virgil Wallentine and Dr. David Schmidt.

Especially, thankyou to my wife Andy for putting up with me during this time.

1.0 Overview

This paper applies the existing pattern recognition technique of Finite Inductive Sequences (FI) to the problem of image compression. FI will be discussed further in Section 1.1. The specific compression problem considered is lossless transmission of images between two points. **Lossless** means that the compressed image is reconstructed as an exact duplicate of the original. The model assumes that reference information is stored at both ends of the transmission.

It is the hypothesis of this paper that the FI technique is directly applicable to the problem of lossless data compression. **Compression**, for purposes of these experiments, will be calculated using a ratio of the size of the data which must be transmitted from station to station and the size of the original data set which would have been transmitted.

Compression is performed in three distinct steps. See Figure 1 for a diagram of the system. Initially the images are pre-processed, via a transform which will be described in Chapter 3, to reduce the effects of differing light intensities between photographs. The next step in the algorithm involves the building of a knowledge base of

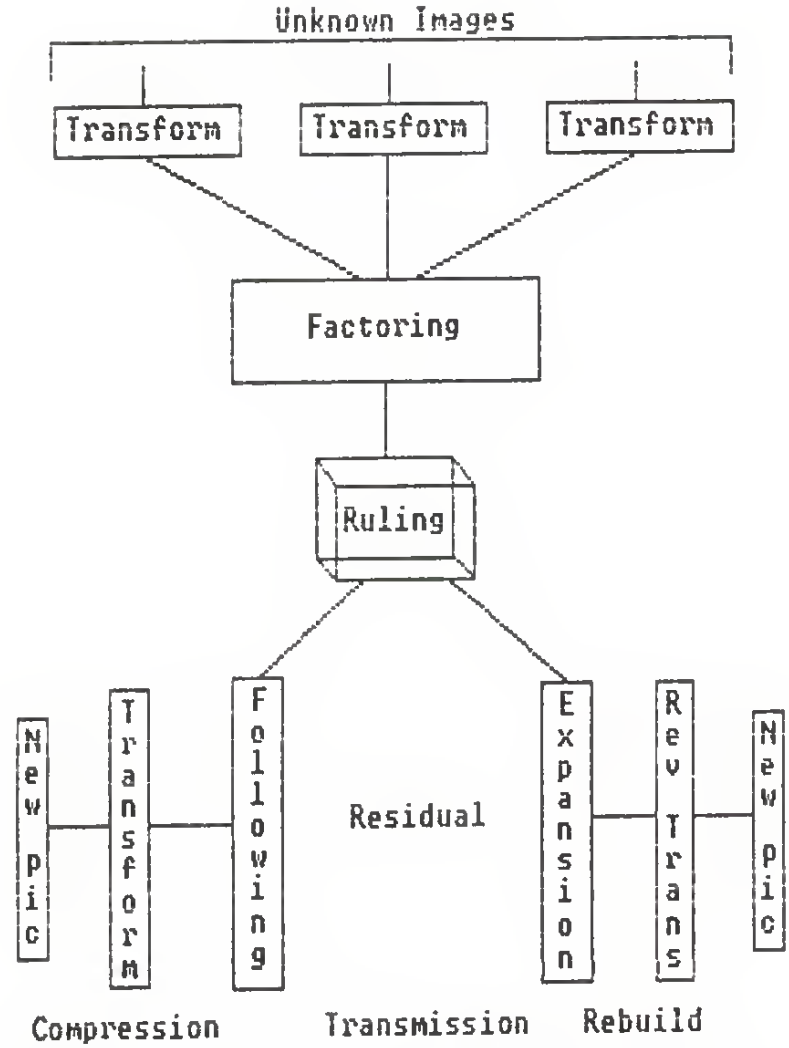

Figure 1. Diagram of FI Compression System

patterns which exist in a aet of photographs. In the final step, the knowledge base is used for compression of new photographs. New pictures are compressed by removing patterns that are contained in the knowledge base.

Experiments were performed to study knowledge base growth and performance compressing new photographs. Knowledge bases of one, two, and four photographs were constructed and applied to new photographs to evaluate the relation between the knowledge base and compression results. See Table 1 for a summary of the results.

Picture #	Pictures used to build knowledge base 2	2,5	1,2,3,5
1	145,306	139,241	2
2	1	2	2
3	179,567	158,126	3
4	166,957	144,414	131,676
5	124,116	2	2
6	112,908	98,862	55,712
7	142,730	120,299	97,032
8	146,372	126,099	101,228
9	147,808	126,156	103,737

Table 1. Number of bytes remaining of the picture after FI was applied using knowledge bases of 1, 2, and 4 pictures. (Original Size = 262,144 bytes)

Table 1 shows the number of pixels which would be transmitted from point to point. To get the full

compression 32,768 bytes must be sdded to the above values for reasons which will be explained in Chspter 3. For comparison, if the entire picture where transmitted there would be 512*512 or 262,144 pixels. It is easy to see that once a picture is sdded into the knowledge base the compression ratio increases dramatically. The table is discussed in more detail in Chapter 3. Depending on similarity between photographs, compression results ranging from .62 through .34 were achieved. These were then compared to results produced by applying a conventional data compression technique to the photographs.

For these experiments the candidate data sets to be compressed consist of digitized high altitude photographs. Each photograph consisted of 512 X 512 eight bit pixels. Although, the technique discussed could be applied to any form of data with slight modifications. Narrative descriptions of the photographs are contained in Table 2.

Picture #	Description	Figure #
1	Largest feature is a river lined with trees and a bridge crossing the river	3
2	Open land with an airport in the center, with some clouds	4
3	Patchy trees and open land, many crossing roads, and a stream	5
4	Almost entire picture is covered by a city	6
5	Mostly open land with patchy trees and a landing strip in the middle	7
6	Open land with rolling hills, white space and buildings in center	8
7	Scattered trees and bushes with open land, roads, and stream	9
8	Entire picture is an airport surrounded by open land	10
9	Most of picture looks like a city with trees in one corner	11

Table 2. Description of pictures

The general structure of the paper is the following. Chapter 1 contains a description of the problem as well as an overview of FI techniques. Chapter 2 contains an review of other approaches to the problem of image compression. Chapter 3 gives a description of the experiments conducted and their results. Finally, Chapter 4 contains the conclusions drawn from the experiments.

1.1 Description of FI

The Finite Inductive Sequences algorithms operate under the assumption that strings of symbols are made up of small frequently occurring subpatterns. The idea is carried further, in that, if the pattern is repeated, only one copy must be saved in a knowledge base. Once a pattern has been recognized it may be ignored in subsequent occurrences since there is nothing new to be learned from it as it simply repeats known information. If a pattern occurs quite often in a data set, the single saved occurrence would permit all subsequent occurrences to be removed in the remaining data or new data sets. When a pattern has not been previously experienced it may then be added to the knowledge base or left in the data set.

A dissertation was recently written by N. Tavakoli entitled "A New Approach to Pattern Recognition" [TAVA86]. In it the technique of Finite Inductive Sequences (FI) is presented and applied to the problem of pattern recognition. This section gives an overview of the FI concept.

First, it will be useful to define some of the hasic terms used in the discussion of FI:

Symbol - The symbol is the smallest unit of information to FI. Ex. A

Alphabet - An alphabet is the set of all possible symbols in a domain. Ex. {A,B,C,D}

String - A string is the list of symbols to be factored. Ex. ABCDABCD...

Data Set - The set of data from which the string is generated

Implicant - Consists of an antecedent and a consequent. Also known as a **rule.** Ex. A -> B

Antecedent - The predicting part of an implicant. Ex. A

Consequent - The predicted part of an implicant. Ex. B

Ruling - The knowledge base of implicants which describe a string. Ex.
A -> B
B -> C
C -> D
D -> A

In an FI pattern recognition system, first a knowledge base is constructed from implicants contained in a sample string. This knowledge hase is the **ruling** and the process of building the ruling is **factoring.** See Algorithm 3.2.3 in [TAVA86] for the factoring algorithm. Once the ruling has been constructed it may be applied to an unknown string and implicants which are contained in the knowledge base may be removed from the unknown string in another process, which is termed **following.**

The process of factoring involves extracting small

predictable patterna from the string known aa implicants. Once an implicant is recognized it may be added to the ruling. The computation effort of the factoring process is proportional to the product of the length of the data set and the total number of implicants. In the worst case of fully random data, the number of implicants will grow exponentially with the length of the data set, so that the computational effort will be an exponential function of the data set size. In practice, data sets include a great deal of pattern replication reducing the computational complexity. This property also directly relates to the size of the ruling. A graph representing the growth of the ruling is shown in Figure 2.

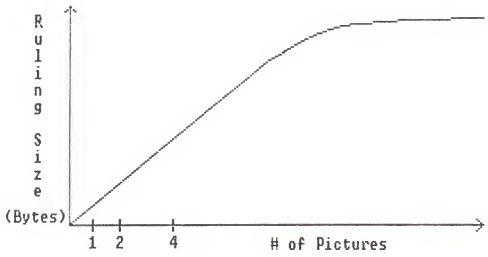

Figure 2. Graph of Ruling Growth

The initial part of the curve has a very large slope showing that the ruling is still gaining experience. As the ruling acquires experience the curve begins to flatten out. If the ruling has acquired every possible pattern which could occur in a domain nothing new could be added to the ruling and the growth curve would be perfectly flat.

As a string is factored all symbols which are predicted by the ruling may be removed from the string. Eventually all but a small number of symbols are predicted. Those which are not predicted are saved as the **residual**. For pattern recognition this residual may then be analyzed to see which

pattern it most closely resembles.

The next phase of FI is **following.** During following the ruling is applied to a new string which may not have been used to build the ruling. This is done by comparing implicants in the ruling to implicants in the new string. As in factoring, if an antecedent is recognized the consequent of the implicant may be removed from the string. Once the entire ruling has been followed over the new string the portion of the string which was not recognized is saved in a residual.

The residual of the following process will contain only information which was not contained in the ruling. As an example, if a string were factored, and the resulting ruling was used to follow a string which is only slightly modified from the original then only the change will remain in the residual.

The following process can be made very fast and may be conducted in line with the application. The speed results from the parallel nature of the following process.

1.2 Grammatical Inference Techniques

This aection notes the relationahip between FI and other techniques of grammatical inference.

FI can be considered a method which allows a grammar to be inferred from a data set and applied to a new data set. Concepts of grammatical inference are presented by K. S. Fu and T. L. Booth in their papers "Grammatical Inference: Introduction and Survey -- Parts I & II" [FU75].

Grammars are used to describe the structural relations of patterns or the syntax of languages. A grammar may also be used to generate all occurrences of a patterns belonging to a set. The process of learning a grammar based on a set of sample data sets is referred to as grammatical inference. It is understood that at least one grammar exists which will describe any given data set.

Inference techniques have been widely applied in the area of pattern recognition. The formulation for a pattern grammar consists of three basic items, as described in [FU75]:

1) A set of primitives or pattern components of which the patterns under consideration are composed.
2) A set of predicates that describe the structure relationships among primitives defined in their

arguments.

3) A set of productions.

However, it is a subjective matter to choose the "best" grammar from a set of grammars.

Fu has proposed a system for grammatical inference which utilizes a man-machine interactive capability to simplify the inference problem. In his system, a human trainer divides each string in the sample into substrings which are again divided. That allows a grammar to be easily constructed as only a small subset of the data is seen at a time.

In comparison, FI will infer a grammar for a data set without a teacher being present. Only one ruling may result from the data set; there is no need to choose the "best" grammar.

2.0 Review of other Compression Techniques

The primary purpose of the literature review contained in this chapter is to introduce some other image compression techniques which have been published.

2.1 Binary Image Compression

This section is to give the reader a brief description of binary image compression techniques. However, these techniques can not be applied directly to a gray scale image because of the added complexity of the image.

In a binary image each pixel has 1 of 2 values. To compress binary images the simplest approach is run-length coding. In run-length coding the image is scanned and the position is saved whenever a new pixel value is found. For compression of images the CCITT recommends a modified Huffman code [SONG86]. A Huffman code assigns symbols to each run-length that requires fewer bits to represent short runs than to long run symbols.

I. Song and S. A. Kassam [SONG86] present a technique which makes use of the fact that all information in a hinary image is contained in the boundaries of the components of the image. Their approach, "Rectangular Edge Coding", is composed of two parts. First, the boundary pixels are

identified and removed leaving only the end pixels. The end pixels are the boundary pixels which form a straight line either horizontally or vertically. The image is then compressed further by reducing the correlation between the coordinates of the remaining pixels. This is done by saving only the coordinates of the edges.

The concept that the boundary contains all information in a binary image is also used by P. A. Maragos and R. W. Schafer [MARA86]. They also suggest isolating the boundaries, skeletons in their description, and then compressing what remains as most of the information has been removed. Their technique is rigorously defined using mathematical morphology theory.

2.2 Filtering Techniques

S. Dravida et. al. [DRAV84] published a paper entitled "A Comparison of Image Filtering Algorithms". In it they discuss four image filtering algorithms. The four algorithms are two-dimensional separable median filters, the Wallis algorithm, the Reduced Update Kalman filter (RUKF), and a combination of the above call multiple model filtering.

Separable median filters are a two-pass operation. First,

the rows of the image are filtered horizontally. Then the columns are filtered vertically. Advantages of this type of filter are that features of the image such as edges are preserved.

Local statistics are used to perform filtering with the Wallis Algorithm. In this approach a local mean and variance are calculated over a (2N+1) X (2N+1) square window centered on the current pixel. The new value for the current pixel is then based on these calculations. This filter is not completely reversible and the amount of error depends on the size of the window. The larger the window the more the error. However, if the window is too small very little information is filtered out.

Reduced Update Kalman filters are based on linear predictive coding strategy. RUKF performs a prediction by employing a dynamic model and follows it with an update of the surrounding pixels. This class of filters performs well on homogeneous images but tends to blur edges in natural images.

To overcome this blurring the Multiple Model filter was developed. This filter is actually made up of five separate filters. The decision of which filter to apply to a given

area is made based on a quadratic decision procedure applied to a 3 X 5 local window of pixels.

E. H. Adelson and P. J. Burt [ADEL81] utilize a Laplacian filter to compress images. In their paper a Laplacian weighting function is used, where each pixel is computed as a local weighted average centered on the pixel. This is to remove the correlation between image pixels. Once the image has been filtered each pixel may be represented with fewer bits. To obtain greater degrees of compression the process may be iterated.

Burge and Wu utilize transforms along with pattern recognition procedures to compress images [BURG81]. The technique is an adaptive method based on a discrete cosine transform. The pattern recognition procedures are applied in order to classify sub-image blocks in order to apply individual treatment to specific areas of the image.

The application of fractals to lossy image compression is approached by E. Walach and E. Karnin [WALA86]. The technique consists of traveling over the image with a fixed length "yardstick" and transmitting a sign bit, for direction and the horizontal distance covered by the yardstick. Upon receiving these values the receiver may

estimate the data values. The authors believe that the algorithm is consistent with the known characteristics of the human visual system. This assumption comes from the fact that the human eye will not view a smooth area with as high of a level of resolution as it will a complicated image. This behavior is also shown by the algorithm.

2.3 Conventional Image Compression Techniques

The following section introduces several universal data compression techniques which are based on the probability of a value appearing. As more of the image is scanned the techniques ability to predict a value increases as the statistics used to calculate the probability are more complete.

J. Ziv and A. Lempel introduce a universal algorithm for compression which utilizes a coding process which learns as it encounters new data [ZIV77]. The algorithm consists of a rule for parsing strings of symbols into substrings and a coding scheme which maps these substrings into unique codewords. The coding scheme utilizes the past history of the occurrence of symbols to generate the codewords. Using their method the authors were able to achieve compression results comparable to results achieved using a technique which had prior knowledge of the contents of the image.

J. Rissanen attempted to improve on the algorithm presented in [ZIV77] by utilizing the "contexts" of a pixel to generate the statistics rather than parsing the data and collecting statistics based on these substrings [RISS83]. These contexts are also subsets of the string but they have varying lengths and overlap. The context is a function of past experience within the string. Using their probabilities the contexts are then encoded into unique values.

Arithmetic coding is another technique which generates coding statistics as the data is encoded. [WITT87] discusses this technique and gives an implementation of the algorithm. This technique is different from those mentioned above as there is no parsing of the data into blocks. Arithmetic coding achieves the theoretical compression limit of entropy for any source. In arithmetic coding, a data set is represented by a range of real numbers between 0 and 1. As the data set becomes longer, the range which represents it becomes smaller, and the number of bits needed to specify that range grows. This occurs because as the range grows smaller more precision is needed to represent it. Symbols which are more likely reduce the range by less than the unlikely symbols so fewer bits are added to the compressed

version of the data.

G. G. Langdon and J. Rissanen give a different algorithm for the process of arithmetic coding binary images in [LANG81]. The primary difference this algorithm has from others is that the divisions between 0 and 1 are done in powers of 2. This allows the elimination of the multiplication process involved in other algorithms and replaces it with a shift.

Other compression techniques use models to predict the next value in a data set. I. H. Witten and J. G. Cleary discuss this problem in [WITT86]. The example they give is, if one computer is sending a stream of numbers to another. If it is possible to predict exactly what the next number is going to be, no numbers need to be sent. This is an ideal situation, in real life if the guesses can be correct most of the time only the wrong guesses need be sent. The models discussed in this paper are a static model and an adaptive model. The static model is constructed separately from the application and doesn't change throughout the execution. This is a very restricted model. An adaptive model is constructed from observation of past events and can change dynamically. This paper talks specifically about the technique of Markov modeling. Markov modeling assumes each symbol in a sequence is affected by a fixed number of

directly preceding symbols. This is much the same premise that FI operates under. However, Markov modeling uses statistical methods to determine which symbol to predict while FI utilizes absolute prediction methods. Arithmetic coding is also considered a predictive technique by the paper.

3.0 Compression Experiments

The following sections detail experiments conducted in applying FI to image compression. First, the sample data is presented, then the filtering and scanning strategies are discussed, and followed by a discussion of alphabet selection strategies. Following these discussions of technique, experimental results concerning ruling growth and compression results are presented. Also, an explanation of the restructured picture is given along with a description of the verification used to prove that the reconstructed picture is identical to the original.

3.1 Description of Data

All experiments were performed using nine high altitude photographs provided by Rome Air Development Center (RADC). The photographs were supplied in a digitized format with a resolution of 512 X 512 pixels. Each pixel represented a unique gray level, one of a possible 256 such levels, requiring eight bits for expression. The value 0 represented black and 255 represented white. The photographs were of a variety of locations showing terrain which varied from a river to a city. See Figures 3 through 11 in the appendix for photographs.

3.2 Filtering and Scanning Strategies

One problem in applying the FI technique to the picture data was that there was differing light intensities between photographs. Because of this, it was decided to transform the raw data by means of a filter. In order to assure lossless compression the filtering transform had to be reversible.

The filter which was chosen utilized a pairwise subtraction strategy performed on a horizontal and vertical pass over the photograph.

3.2.1 Effects of Filtering

This section describes the results yielded by filtering the data. The next section describes in detail the full transformation strategies which were tried during the course of the research.

Transforms were tested by comparing the entropies of the translated data against that of the raw data. It was assumed that lower entropies indicated a better transform. A low entropy indicates that a large portion of the data is redundant. This redundanc gives a higher probability that a pattern will occur more than once.

Entropy is term used to describe the complexity (or randomness) of a data set. The entropy value is the minimum number of bits on average which must be used to uniquely represent each pixel in the photograph. The equation for entropy follows:

$$\text{entropy} = -\sum_{i=1}^{n} P_i \log_2 P_i \quad \text{bits/symbol}$$

Where n is the number of possible gray levels, in our case 256, and P_i is the frequency of occurrence for the value of i [HELD83]. If a photograph has an entropy value of 6 it will take a minimum of 6 bits per pixel to represent the picture.

A side effect of the low entropy was that the pictures appeared smoother to the naked eye and the entropy of the picture was reduced. If the pictures appeared smoother to the naked eye this implied that there were fewer discrete jumps between pixels, making the translated data more homogeneous than the raw data. The entropies of the photographs before and after filtering are given in Table 3.

Picture #	Raw Data	Translated
1	6.122909	2.284749
2	6.662895	2.234960
3	6.635171	4.338287
4	6.496347	2.896313
5	5.639180	1.807386
6	5.879077	1.502678
7	6.451508	2.336874
8	6.387721	2.389371
9	6.318030	2.455133

Table 3: Entropies for the Nine Pictures Transformed with pairwise subtract

Histograms of the occurrences of gray scale values were generated from the raw and translated data and compared. See Figures 12 through 19 for some example distributions of raw, and translated data. As an aside, if the range of pixel values is small enough, each pixel may be represented in fewer bits.

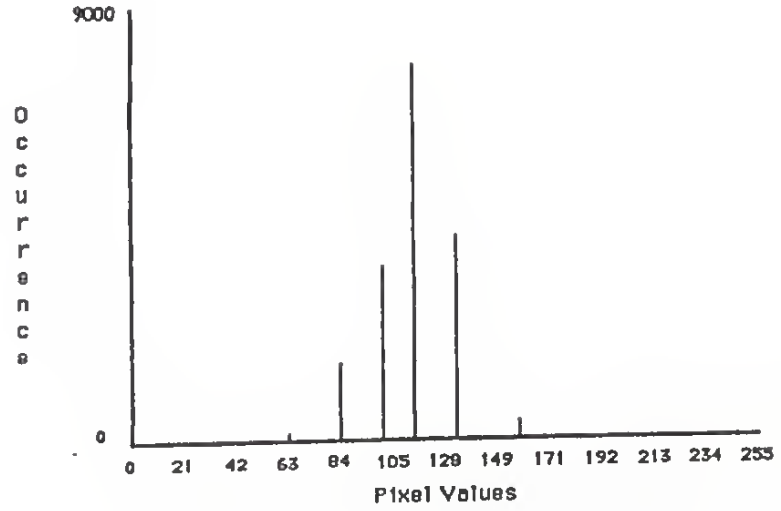

Figure 12 : Distribution of Pixels from Raw Data of Picture 1

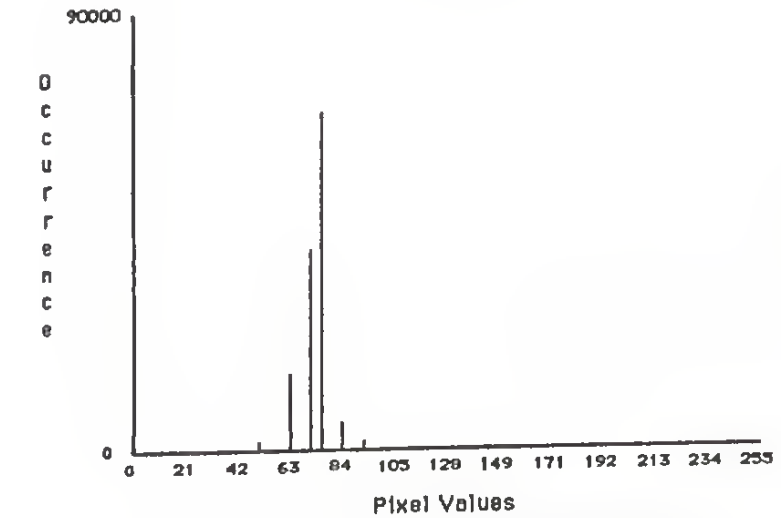

Figure 13 : Distribution of Pixels from Processed Data of Picture 1

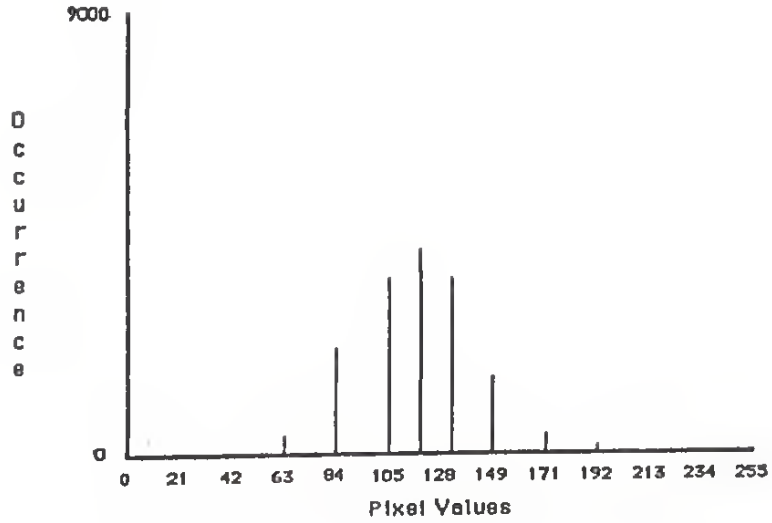

Figure 14 : Distribution of Pixels from Raw Data of Picture 3

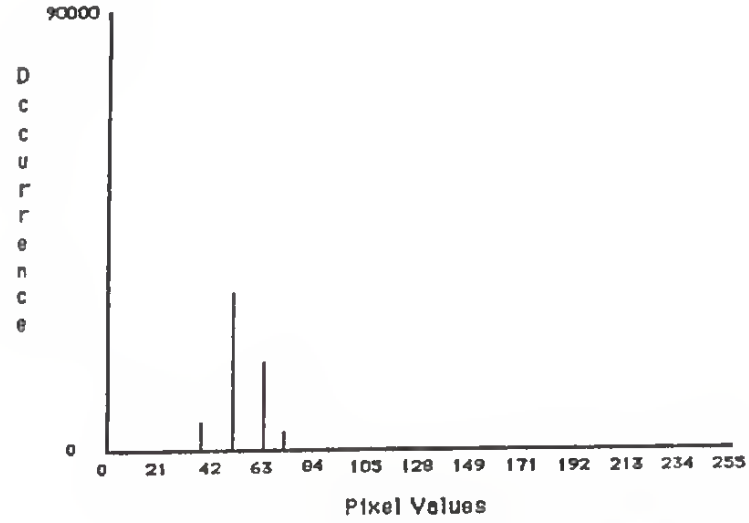

Figure 15 : Distribution of Pixels from Processed Data of Picture 3

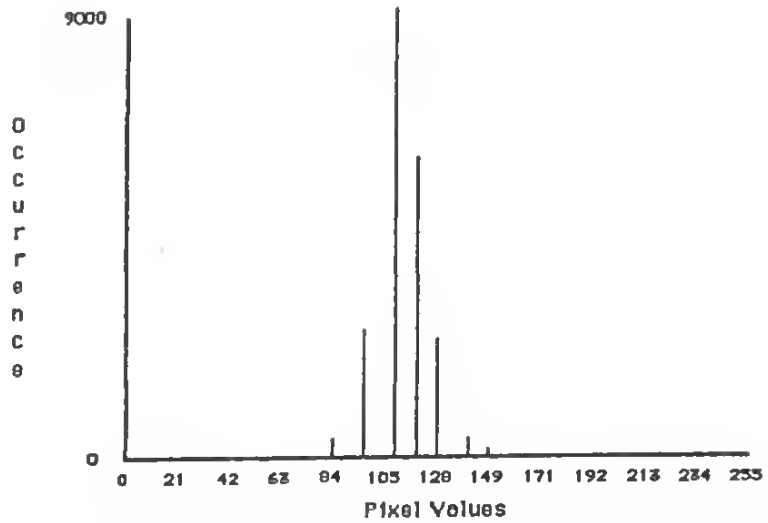

Figure 16 : Distribution of Pixels from Raw Data of Picture 5

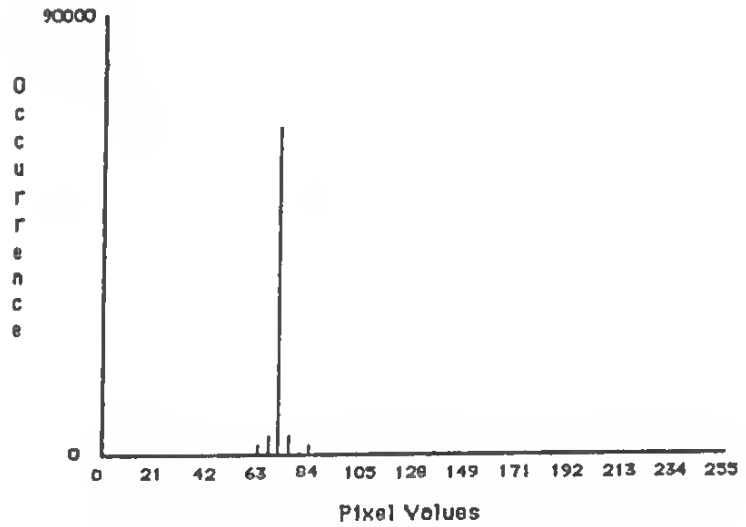

Figure 17: Distribution of Pixels from Processed Data of Picture 5

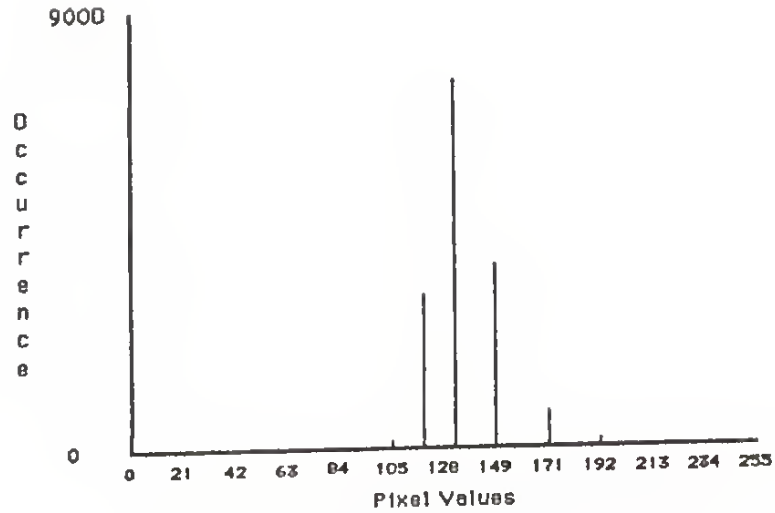

Figure 18: Distribution of Pixels from Raw Data of Picture 6

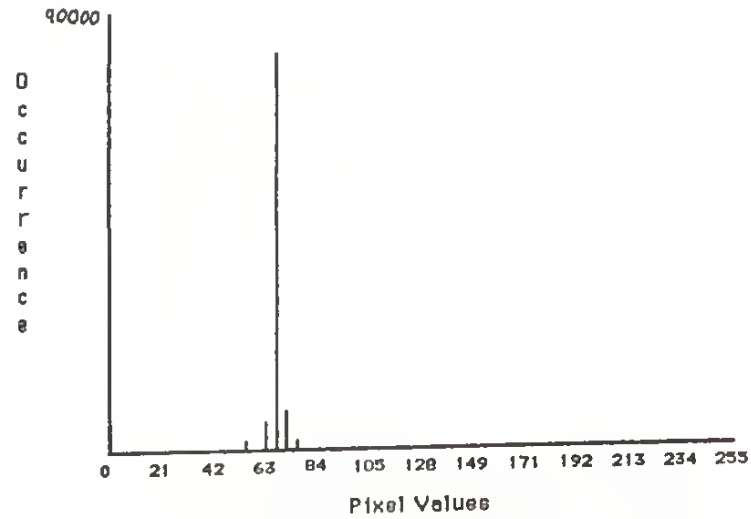

Figure 19 : Distribution of Pixels from Processed Data of Picture 6

3.2.2 Possible Transformation Operations

Two transformations were tried which were fully reversible. These were pairwise subtraction and a bitwise exclusive or (XOR).

Pairwise subtract was attempted using a horizontal raster scan. In the pairwise subtract the value of the preceding pixel in the raster was subtracted from the current pixel with the resulting value being saved in place of the current pixel. The equation follows:

```
Xnew(0,0) = X(0,0)

For i=0 to 511
 For j=0 to 511
  if ( i>0 AND j=0 )
   Xnew(i,j) = X(i,j) - X(i-1,j)
  else
   Xnew(i,j) = X(i,j) - X(i,j-1)
```

Where Xnew is the transformed value and X is the original value. On reconstruction the first pixel gives the starting value. Propagating the inverse operation through the raster results in perfect reconstruction.

After the transformation was performed, the data was linearly shifted relative to the largest negative value to move all of the values into the eight-bit range for viewing.

$$Xout(i,j) = Xnew(i,j) + Low$$

Where Xout is the final value for the pixel and Low is the lowest negative value yielded by the transformation operations.

When the normalized picture was viewed, an overall smoothing of the picture was readily apparent. There were no distinct lighter and darker regions remaining as the entire picture appeared to be a consistent gray level. In appearance the transform seems to cause the pictures to appear almost as a three dimensional view. Topological features such as hills, roads, and buildings seemed to be slightly above the surrounding area. Light clouds in other pictures were also readily apparent as their relative distance from the ground stood out.

A bitwise XOR between adjacent pixels was the next transform which provided very surprising results. In this operation corresponding bits in pixels were exclusive ored giving an 8 bit result. The equation used is given as:

```
Xnew(0,0) = X(0,0)

For i=0 to 511
 For j=0 to 511
  if ( i>0 AND j=0 )
   Xnew(i,j) = X(i,j) XOR X(i-1,j)
  else
   Xnew(i,j) = X(i,j) XOR X(i,j-1)
```

Roads, buildings, and other areas where there was a distinct jump in color intensity between adjacent pixels, appeared as almost white values while the rest of the picture was dark gray or black. This produced an outlining effect which could be used for edge detection. If the picture is observed close enough, light outlines are discernible around individual trees and other comparatively small features. Figure 20, in the appendix, shows picture one after the XOR transform.

The apparent cause of this phenomenon was that if adjacent pixels had more corresponding bits differing, the resulting value had a higher contrast than if the pixels were almost the same. For example, suppose the following are a string of pixels which could be found in a line of the picture (the values are in hexadecimal):

58 59 57 A0 A2 5D

These could represent a sample of a narrow road passing through a field. On either side of the road the values are almost the same. On the road the values are also close. But, the difference between pixels on the road and off the road are significant. The bit pattern for the above values are:

01011000 01011001 01010111 10100000 10100010 01011101

If an XOR operation is performed on the adjacent pixels it yields:

01011000 00000001 00001110 11110111 00000010 11111111

The first pixel value remains unchanged because it was the seed value. The hexadecimal values of the bit strings are:

58 01 0E F7 02 FF

As can be seen, the location where a distinct jump between pixel values exists, the transformed values are significantly higher. This causes a distinct white line to appear along the edge of the road.

Another transform equation which was attempted was to take the current pixel, suhtract the pixel value above it and the pixel value to the left of it, and add the pixel value up and to the left of the current pixel. If a pixel was in row R and column C, then the pixel in X(R,C-1) was subtracted. Likewise the pixel value above it X(R-1,C) was suhtracted from it. The pixel value in X(R-1,C-1) was added to the

X(R,C) pixel. In equation form:

```
For R=1 to 511
 For C=1 to 511
  Xnew(R,C) = X(R,C) - X(R,C-1) - X(R-1,C) + X(R-1,C-1)
```

This yielded almost the same results as making two passes with the subtraction, one horizontal and one vertical. But, the peak value at the mean was not quite as great as other transforms so it was not pursued further.

The transform which was finally decided upon was the pairwise subtraction because of the low entropy and smooth appearance of the data after it was transformed.

As the pairwise subtract was the chosen transform, when the word **transform** appears it is in reference to the pairwise subtract operation.

3.2.3 Scanning Strategies

Other operations performed on the data involved different scanning strategies. Where a scanning strategy is the pattern in which the data is transformed. The first was to scan the data in a raster format. This was rejected because the pixels in column 1 were processed using the pixels in column 512. These two pixels had no relation to each other, so in some cases they created an anomaly in the transformed data.

Another scan, shown in Figure 21, was termed the "Snake scan." With it the data in the first row was scanned left to right, the second row was right to left, and so on in an alternating fashion. Pixels on the edges were transformed using the pixel above it. This scan was better because it yielded a smaller entropy value than the raster scan, but still its new results did not meet our level of expectation.

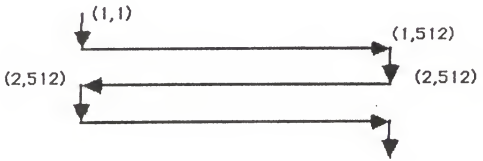

Figure 21 ; Snake Scan

The scan, which was adopted, is a "modified raster scan" shown in Figure 22. The data was scanned left to right with the scan terminating at column 512. Pixels in column 1 were transformed using the corresponding pixel in the row above it.

Figure 22 : Modified Raster Scan

More than one pass of the same transform was also attempted with the philosophy that if the first transform produced better results, then multiple applications of the transform would be better. This did not prove to be true. Transforms which caused the raw data to converge on a single pass began to cause the data to diverge on multiple passes. That is, the normal distribution representing the pixels disappeared after multiple passes.

Vertical scans of the same type as the horizontal scans were also evaluated. These yielded many of the same desirable properties as the horizontal scans; the main difference being the orientation of the data they transformed best. For example, horizontal scans worked best on vertical lines, while vertical scans performed better on horizontal lines. In this case performing better means removing more of the discontinuity between the adjacent pixels, thus making the

data more homogeneous in that direction.

The scan pattern adopted for the pictures was the modified raster scan shown in Figure 22, performed horizontally first, then on a second pass, performed vertically. This preserved the relations between pixels since a single pixel not only relates to the pixel immediately to the left or below it but also to the pixels above and to the left of it. This scan seemed to reflect that relation, and smooth the data well.

3.3 Alphabet Selection

For FI to work well an alphabet must be defined. Desirable properties of the alphabet and methods of it's generation are described below.

In selecting the alphabet, different size symbols could be extracted from the transformed data. Symbols of size 2, 4, 8, 16, and 32 bits could be used. To do this the pictures would be treated as long bit strings and separated into symbols of the desired bit length. Symbol sizes of 2 and 4 bits would yield a smaller alphabet, but would be rejected because adjacent symbols of each 2 bit symbol have no observable relationship to one another. The same consideration likewise applies to 4 bit symbols. Symbol

sizes of 16 and 32 bits would yield a huge alphabet; 65,536 members for 16 bits and 4,294,967,296 members for 32 bit symbols. With these large alphabets, the probability of all symbols being unique over all the pictures is very high. That is, that each symbol may occur only once in the entire collection of all pictures in the data set. If these symbols were factored almost all of the data would be removed in the first level, giving a large ruling of very few levels and a very small residual. If another picture were followed through this ruling, because of the uniqueness of the symbols, almost nothing would be removed. These observations left consideration for only eight-bit pixels. These eight-bit pixels were factorable and dependent upon each other, but were still not very effective when used for following because of the light intensity problem.

3.4 Ruling Design and Construction

The next step was to create a ruling. To generate this ruling, pictures 1, 2, 3, and 5 were chosen. These particular pictures were chosen to give a wide experience base for the ruling over the most similar terrain. In order to provide for this experience base, a variety of topological variations were selected for the following reasons:

-- Picture one includes a river as a major portion of the terrain. Adding this experience to the ruling ahould enable the ruling to remove large portiona of lakes and rivers in other pictures.

-- The second picture chosen, picture two, is largely open land. In the center is a landing strip with a few buildings. Also, there are some buildings clustered around the edge. A few small clouda are also visible.

-- The next picture chosen, picture three, contained the most varied data. It consisted of scattered trees, open fields, and many criss-croasing roada and paths. Some of these roads appeared to be dirt while others were paved. Because the roads contained many curves, it provided the ruling the opportunity to gain experience on roads which travel in many different directions, as well as very broken terrain.

-- The final picture, picture five, is largely open terrain with a few straight-line roads, scattered trees and buildings, and a landing strip. This added a variety of open terrain to the experience base.

To test ruling growth three rulings were constructed from

just picture 2, then from pictures 2 and 5, and finally from all four pictures. Table 4 shows the final sizes of the rulings.

Picture Numbers Used to build the ruling	Original data size (bytes)	Ruling size (bytes)
2	262,144	1,382,341
2, 5	524,288	2,788,747
1, 2, 3, 5	1,048,576	5,621,187

Table 4. Ruling Sizes for data sets of 1, 2, and 4 photographs.

As anticipated the ruling was several times larger than the original data set. However, the ruling seemed to be growing slightly greater than linearly. This could be caused by contradictions of implicants which existed in smaller data sets. Due to the large number of patterns contained in the pictures the ruling continued to grow at an almost constant rate. This shows that the ruling growth is still at the bottom of the curve in Figure 2. To get the ruling up on the curve to the point were it begins to flatten out may mean factoring hundreds or thousands of pictures. This would be necessary because of the huge number of patterns which occur in nature.

If a large number of photographs are taken of similar terrain there is the possibility that not as many photographs would be needed. This is because patterns would be almost consistent across the pictures.

3.5 Compression Experiments (Following)

This section details experiments concerning the effectiveness of using the rulings for compression on the photographs. For the purposes of this paper, the more the ruling removes from a photograph the more effective it is at compressing that photograph. As an aside, this also implies that the relative entropy of the photograph to the ruling is also lower. Table 1 is included here once again for ease of reference.

Picture #	Pictures used to build ruling 2	2,5	1,2,3,5
1	145,306	139,241	2
2	1	2	2
3	179,567	158,126	3
4	166,957	144,414	131,676
5	124,116	2	2
6	112,908	98,862	55,712
7	142,730	120,299	97,032
8	146,372	126,099	101,228
9	147,808	126,156	103,737

Table 1. Number of pixels in the residual using rulings of 1, 2, and 4 pictures. (Original Size = 262,144)

All nine of the photographs were followed through each of the three rulings described in the above section. It is easy to see that once a photograph ia added to the ruling almost none of the information remains in the residual. This supports the theory that the more experience the ruling has gained the better the compression results.

When FI is used for pattern recognition reconstruction of the string is not a concern. However, to assure perfect reconstruction extra information in the form of a bitmap must be retained. How the hitmap is used for reconstruction of the picture is explained in Section 3.6. The bitmap contains information as to which symbols were removed from the string, a 1 if removed otherwise a 0. On a worst case basis the size of this bitmap would be 1 bit for each symbol or 32,768 bytes. However, if a string were contained in the ruling no bitmap would be needed. Even if the bitmap were utilized the entropy should be very low allowing it to be compressed further with binary compression techniques. To allow for the bitmap size, compression will be calculated by:

Compression = (Residual Size + 32768) / 262,144

As a comparison to existing techniques, Table 5 shows the

results from compressing the photographs using the Unix 'pack' function on the transformed data. The FI compression values are from using the four picture ruling.

Picture #	FI compression	Pack compression
1	.13	.45
2	.13	.45
3	.13	.55
4	.63	.52
5	.13	.40
6	.34	.34
7	.50	.46
8	.51	.47
9	.52	.47

Table 5. Comparison between FI results and Unix pack function results

The average compression ratio achieved by FI is .34 while the average compression for the pack function is .46. It is very unlikely that an exact duplicate of a photograph in the ruling will be encountered again. Therefore, if pictures which were contained in the ruling are disregarded the FI compression ratio drops to .50. It must be kept in mind that no effort has been made to compress the bitmap which was added to the FI results. Also, the photographs were highly dissimilar and the ruling was very limited in it's experience.

3.6 Reconstruction of the Picture

The factoring proceas is eaaily reversed. The residual of the factoring process and the bitmap are used as a starting point. The following pseudo code explains the process:

> Scan the bitmap for a 1 also scan the residual to the same number of bits.
>
> Find a matching implicant in the ruling for the current antecedent.
>
> Add the consequent of the implicant to the residual.
>
> Repeat the process until the atring is reconstructed.

This process is referred to as **expansion.** A 1 in the bitmap indicates that a symbol has been removed from that position in the string.

Because the data which was factored had been transformed, upon completion of the expansion process the picture will be in it's transformed form. The remaining step required to complete the lossless regeneration is to reverse the transformation. This is done simply by reversing the order of the tranaformation passes and performing additions to move the tranalated data to it's original values.

3.7 Comparison of Original with Reconstructed Picture

To verify that no data has been lost, a comparison of the

original and the restructured picture must be performed. The easiest and least forgiving method of comparison is to perform a subtraction between corresponding pixels in the data sets. If a non-zero result from the subtraction is encountered, a discrepancy exists and is reported.

Through this method of verification, it was shown that the restructured picture is an exact duplicate of the original.

4.0 Conclusions

This chapter provides a summary of the results obtained during the experiments of the preceding chapter. Also in this section are conclusions which may be drawn from the results and auggestions for further research.

The first conclusion which may be drawn is from the experiments with transforms. It was shown that the entropy of the photograph could be substantially reduced without any loss of data. This is significant for conventional compression techniques, because of entropy is interpreted as the limit of possible compression.

The experiments with ruling growth showed that with data as varied as in this study the rulings experience level was still at the bottom of the curve in Figure 2. For the ruling

to gain enough experience to move to where the curve flattens out hundreds or thousands of pictures may have to be factored into the ruling.

The compression experiments (Following) ahowed that the larger a ruling ia, i.e. the more picture information it contains, the greater the capability for compression of new pictures. This is because the larger ruling contains more experience and may remove more patterns. Theae results appear in Table 1.

The compression results achieved in this paper using FI indicate that this technique may hold promise for use in the area of compreasion for transmisaion. If the compression were not for the purposes of transmission many pictures would have to be stored to show much of a savings because of the size of the ruling. However, applications such as transmission of pictures from a satellite to a ground station would benefit greatly. The ruling would have to be transmitted once, or could be stored in memory of the satellite before liftoff. Once the ruling was on board many pictures could be transmitted back in a much smaller bandwidth and expanded using the ruling at the ground station.

Subjects which were not aufficiently atudied due to time conatraints involve the combinations of pictures. Would different combinationa of pictures give better results? It was apparent that pictures which appeared more alike were compressed to a higher degree than pictures which differed greatly. However, ia there a combination of pictures which would give optimum compression across all of the photographs?

To conduct further experiments on ruling growth a larger data set of pictures would be needed along with a large amount of computer time.

Studies also need to be conducted into the area of the bitmaps needed to expand the picture. Methods which would reduce the size or eliminate completely the need for these would greatly enhance the compression performance of FI.

Bibliography of Referenced Materials

[ADEL81] Adelaon, E. H. and Burt, P. J., "Image Data Compression with the Laplacian Pyramid", Proceedings IEEE 1981, 1981, p. 218-223.

[BURG81] Burge, R. E. and Wu, J. K., "An Adaptive Tranaform Image Data Compreaaion Scheme Incorporating Pattern Recognition Procedures", Proceedings IEEE 1981, 1981, p. 230-236.

[DRAV84] Dravida, S., Wooda, J. W., and Shen, W. C., "A Comparison of Image Filtering Algorithms", proceedinga IEEE 1984, 1984, p. 23.3.1-23.3.4.

[FU75] Fu, K. S. and Booth, T. L., "Grammatical Inference: Introduction and Survey -- Part I", IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1975, p. 343-359.

Fu, K. S. and Booth, T. L., "Grammatical Inference: Introduction and Survey -- Part II", IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1975, p. 360-375.

[HELD83] Held, G., Data Compression, Wiley Heyden, p. 59.

[LANG81] Langdon, G. G., Jr. and Riasanen, J., "Compression of Black-White Images with Arithmetic Coding", IEEE Transactions on Communications, June 1981, p. 858-867.

[MARA86] Maragoa, P. A. and Schafer, R. W., "Morphological Skeleton Representation and Coding of Binary Images",IEEE Transactions on Acoustics, Speech, and Signal Processing, Oct. 1986, p. 1228-1244.

[RISS83] Rissanen, J., "A Univeraal Data Compreasion System", IEEE Transactions on Information Theory, Sept. 1983, p. 656-664.

[SONG86] Song, I. and Kasaam, S. A., "A New Noiseleas Coding Technique for Binary Images", IEEE Transactions on Acoustics, Speech, and Signal Proceaaing, Aug. 1986, p.944-951.

[TAVA86] Tavakoli, N., "A New Approach to Pattern Recognition", PH.D. Dissertation, Kansaa State Univeraity, 1986.

[WALA86] Walach, E. and Karnin, E., "A Fractal Based Approach to Image Compression", IEEE ICASSP 86, Tokyo, 1986, p. 11A.3.1-11A.3.4.

[WITT86] Witten, I. H., Radford, M. N., and Cleary, J. G., "Arithmetic Coding for Data Compreasion", Communications of the ACM, June 1987, p. 520-540.

[WITT87] Witten, I. H. and Cleary, J. G., "Foretelling the Future by Adaptive Modeling", ABACUS, Spring 1986, p. 16-36+.

[ZIV77] Ziv, J. and Lempel, A., "A Univeraal Algorithm for Sequential Data Compression", IEEE Transactions on Information Theory, May 1977, p. 337-343.

APPENDIX

Figure 3. Picture 1 - Raw Data on Left,
Transformed with Subtraction on Right

Figure 4. Picture 2. Raw data on Left,
Transformed with Subtraction on Right

Figure 5. Picture 3 - Raw Data on Left,
Transformed with Subtraction on Right

Figure 6. Picture 4 - Raw Data on Left,
Transformed with Subtraction on Right

Figure 7. Picture 5 - Raw Data on Left,
Transformed with Subtraction on Right

Figure 8. Picture 6 - Raw Data on Left,
Transformed with Subtraction on Right

Figure 9. Picture 7 - Raw Data on Left,
Transformed with Subtraction on Right

Figure 10. Picture 8 - Raw Data on Left,
Transformed with Subtraction on Right

Figure 11. Picture 9 - Raw Data on Left,
Transformed with Subtraction on Right

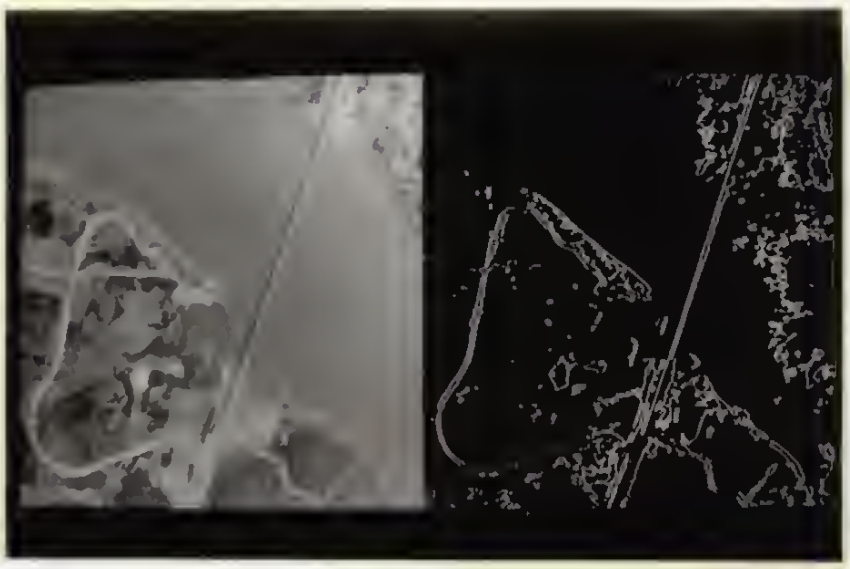

Figure 20. Picture 1 - Raw Data on Left,
Tranaformed with XOR on Right

A PATTERN RECOGNITION APPROACH TO LOSSLESS IMAGE COMPRESSION

by

SCOTT LARAN JOHNSON

B.S., Kansas State Univeraity, 1985

AN ABSTRACT OF A REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988

This paper applies an existing pattern recognition technique to a problem of image compression. The specific compression problem considered is transmission of images between two points, assuming that reference information is stored at both ends of the transmission.

Compression is performed in two distinct steps. As part of the first step, some pre-processing of the images is performed to reduce the effects of differing light intensities. Then, sample photographs are used to build a knowledge base of patterns. In the second step, the knowledge base is used for compression of new photographs. New pictures are compressed by removing patterns that are contained in the knowledge base.

Knowledge bases of one, two, and four photographs were constructed and applied to a test photograph to evaluate the relation between the knowledge base and compression results. Depending on similarity between the test photograph and the knowledge base, compression results ranging from none to 45,000 to 1 were achieved. These were then compared to results produced by applying a conventional data compression technique to the photographs.