COMPRESSION OF A SIGNATURE DATABASE

THESIS

Presented to the Graduate Council of
Texas State University-San Marcos
in Partial Fulfillment
of the Requirements

for the Degree

Master of SCIENCE

by

Edson Ricardo Amboni, B.S.

San Marcos, Texas
December 2011

COMPRESSION OF A SIGNATURE DATABASE

Committee Members Approved:

Dan E. Tamir, Chair

Mina S. Guirguis

Apan Qasem

Approved:

J. Michael Willoughby
Dean of the Graduate College

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

ACKNOWLEDGMENTS

I would like to thank Dr. Mina S. Guirguis and Dr. Apan Qasem for participating on my committee.

In addition, I would like to thank Dr. Dan Tamir for his patience and guidance during this entire process of developing and writing this thesis.

Finally, I would like to thank my family for their support. In specific, I would like to thank my wife Viviane for her understanding and encouragement throughout the nights and weekends I have spent focusing on this project.

This manuscript was submitted on October 18, 2011.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS v

LIST OF TABLES viii

LIST OF FIGURES ix

ABSTRACT xi

CHAPTER

I. INTRODUCTION 1

II. BACKGROUND INFORMATION 4

Digital Image Processing 4
Image Segmentation 5
K-Means Clustering 6
Image Thresholding 6
Connected Component Labeling 7
Contour Tracing 8
Object Representation 9
Chain Code 9
Object Signature 11
Object Recognition 12
Quantization 13
Sum of Absolute Differences 14
The Error Image 14
Signal-to-noise Ratio 14
Lossless Compression 15
Lossy Compression 16
DPCM 17
Differential LPC 17
Compressed Sensing 18
DCT 18

III. RELATED WORK 19

IV. EXPERIMENT SETUP 21

Process 1: Generating a PGM Image from BPM Files 23
Process 2: Creating a Signature Database from a PGM Image 24
Process 3: Reconstructing a PGM Image from a Signature File 27
Process 4: Recognition of a PGM Image Using a Signature Database 30

V. EXPERIMENT RESULTS AND EVALUATION 31

Experiment 1 32
Experiment 2 49
Experiment 3 51
Experiment 4 53
Summary of Results 55

VI. CONCLUSION AND FURTHER RESEARCH 60

WORKS CITED 62

LIST OF TABLES

Table **Page**

Absolute Difference between Non-convex Star Shaped Signatures and the Signature Database 35

Absolute Difference between Rectangle Signatures and the Signature Database 37

Absolute Difference between Square Signatures and the Signature Database 40

Absolute Difference between Ellipse Signatures and the Signature Database 42

Absolute Difference between Triangle Signatures and the Signature Database 44

Absolute Difference between Circle Signatures and the Signature Database 46

Absolute Difference between Comparing Shapes Not in the Database 48

Calculated Compression Ratio 50

SNR Values of Images Reconstructed from Compressed Signatures 51

SNR with Different Bit Allocations 52

Distortion after Image Reconstruction 55

LIST OF FIGURES

Figure **Page**

Pixel Matrix 5

Input and Output of Thresholding Process 7

Example of 4-Connected and 8-Connected Pixels 8

Direction Numbers for 4-Connected and 8-Connected Chain Codes 11

Object Signatures...12

PGM Images Contained in the Signature Database...23

Images Not Stored in the Signature Database...24

The Signature Creation Process 26

Non-convex Star Shaped Signature Recreation Process 28

Ellipse Variations: Normal, Rotation, Rotation and Scale, Scale, and Translation.............33

Comparing Non-convex Star Shaped to the Signature Database...34

Comparing Rectangle to the Signature Database 36

Ellipse and Rectangle Signatures..38

Comparing Square to the Signature Database 39

Comparing Ellipse to the Signature Database 41

Comparing Triangle to the Signature Database..43

Comparing Circle to the Signature Database...45

Comparing Shapes Not in the Database...47

Square and Pentagon Signatures 49

SNR Values of Images Reconstructed from Compressed Signatures 50

SNR with Different Bit Allocations 52

Distortion after Image Reconstruction 54

Signature and Image Representation of Non-convex Star Shaped 57

Signature and Image Representation of Non-convex Star Shaped with DCT 57

Signature and Image Representation of Non-convex Star Shaped with DPCM 58

Signature and Image Representation of Non-convex Star Shaped with DLPC 58

Signature and Image Representation of Non-convex Star Shaped with CS 58

ABSTRACT

COMPRESSION OF A SIGNATURE DATABASE

by

Edson Ricardo Amboni, B.S.

Texas State University-San Marcos

December 2011

SUPERVISING PROFESSOR: DAN TAMIR

Lossy compression algorithms are commonly used to compress multimedia data such as image, video, and audio. This type of compression method is generally applied in streaming media applications since there is a strong motivation for reducing the storage space and/or transmission bandwidth where small visual quality loss is acceptable.

This thesis investigates and evaluates lossless and lossy compression algorithms via the compression and reconstruction of an object representation referred to as signature. Lossy compression is investigated due to the fact that lossless compression does not yield significant compression. An object signature is a sequence of values representing the distance between the object center and its boundary.

Several steps are involved in extracting an object signature. The processes applied to an input image are executed in the following order: binarization, connected component labeling (CCL), contour tracing, and signature calculation. This sequence of events is

applied to a set of synthetic images. The output results are used to create an image signature database.

The lossy compression methods studied in this project are Differential Pulse Code Modulation (DPCM), Differential Linear Predictive Coding (DLPC), Discrete Cosine Transformation (DCT), and Compressed Sensing (CS). The lossless method used in this thesis is the Dictionary based compression method utilized by the UNIX ‘gzip’ compression utility. These algorithms are applied to the object signatures before storing them in a database. The compression quality is evaluated with respect to three different aspects: object recognition using a compressed signature, Signal-to-Noise ratio (SNR) of the original signature compared to the reconstructed compressed signature, and pixel by pixel comparison between the original image and the image reconstructed from a compressed signature.

Our results show that the recognition process is able to identify all input objects for every image and compression method combination. The experiments demonstrate that lossless compression is not a viable method for this application. Furthermore, according to the experimental results DPCM has the best rate distortion performance. On the other hand, the Compressed Sensing method produces higher distortion and requires a higher threshold value in order to accept a valid recognition response. This higher distortion caused by Compressed Sensing is also visible in the lower SNR values obtained when comparing the signature error with other compression methods utilized in this project. Although the quality results for this method are not as good as the DPCM method, Compressed Sensing has an important advantage of requiring less data to represent an image during the acquisition process. This means that CS might enable to capture the data with a smaller number of sensors.

CHAPTER I

INTRODUCTION

The explosion of multimedia content is one of the biggest challenges the Internet faces today. According to the Cisco Visual Networking Index report, in 2010 mobile data traffic was three times the size of the entire global Internet in 2000 (Cisco, 2011).

The desire to reduce the data storage space and the bandwidth necessary to transmit content through the networks is one of the main motivations for data compression.

Data compression algorithms are used to decrease the number of bits necessary to represent images, videos, speech, and music. For example, an uncompressed CD-quality music requires 11 times more storage space compared to amount required by the MP3 digital audio encoding format. This significantly increases the time necessary to download the data into a music player. Compression methods are referred to as lossy or lossless, depending if the method loses information or not (Sayood, 2006).

One of the concerns of computer vision relates to the extraction of information from an image or video. For example, consider an automated quality system for detecting an anomaly in a specific product shape. This process stores the expected shape (boundary) of the product and it is able to detect any significant difference between the manufactured product and the desired one by “visually” inspecting object images at the end of the assembly line.

This research identifies and analyzes compression methods that obtain a high signature data compression rate with negligible effect on the rate of correct recognition of objects. This thesis combines image recognition with different types of lossy data compression. The compression methods are applied to the object border representation. The input images used are basic shapes stored in a plain text image representation format, known as Portable Graymap (PGM). These images are processed by a segmentation method in order to detect and extract the contour of objects. The signature is represented by a set of distances between the object center and its boundary. This representation is translational invariant and it is sufficient to recognize the object. Furthermore, it can be modified in a relatively straight forward way to account for scale and rotation variations.

The hypothesis of this research is that it is possible to devise lossy compression methods for compression of object signature data with high compression rate and low distortion while keeping a low recognition error rate.

Four experiments are done as part of the project. These experiments are related to lossless and lossy compression of an object signature extracted from basic shape images. The first experiment is an image recognition test. A signature database is created from a group of shapes. The signature is stored in its plain format and in five different types of compression methods: UNIX ‘gzip’, DPCM, DLPC, DCT, and CS (Sayood, 2006). Images with similar shapes but with variations in scale, rotation, and translation are compared to this database. Images not present in the signature database are also included in this group. The objective of the first experiment is to analyze image recognition rates with different methods of data compression.

The remaining three experiments are related to the compression rate and reconstruction quality of the compression methods used with the image signatures. All these tests use the compressed object signature in order to reconstruct the original image. SNR and distortion error are used to quantify the reconstruction quality. In addition, manual visual comparison is applied to verify the quality of the reconstructed shapes in comparison to the original input images.

Our results show that the recognition process is able to identify all input objects for every image and compression method combination. The experiments demonstrate that lossless compression is not a viable method for this application. Furthermore, according to the experimental results DPCM has the best rate distortion performance. On the other hand, the Compressed Sensing method produces higher distortion and requires a higher threshold value in order to accept a valid recognition response. This higher distortion caused by Compressed Sensing is also visible in the lower SNR values obtained when comparing the signature error with other compression methods utilized in this project. Although the quality results for this method are not as good as the DPCM method, Compressed Sensing has an important advantage of requiring less data to represent an image during the acquisition process. This means that CS might enable to capture the data with a smaller number of sensors.

CHAPTER II

BACKGROUND INFORMATION

Digital Image Processing

A digital image is a discrete representation of a two-dimensional natural image and it is generally represented by a matrix with coordinates x and y, where x indicates the row location and y indicates the column location of each point in the image. As Figure 1 shows each point in this matrix represents a picture element, referred to as a pixel, and the value of each pixel is called intensity or gray level (Gonzalez & Woods, 2002).

Digital image processing is the use of computer algorithms to perform operations on a digital image. These processes are used to enhance, compress and decompress, or segment an image among several other objectives. They are generally divided into three main levels:

1. Low-level process: the input and output of this process are images. Examples in this category are noise reduction, contrast enhancement, and image sharpening.

2. Mid-level process: the input is an image and the outputs are image attributes. One common example is the partition of an image into regions or objects.

3. High-level process: the input is a recognized object and the output is the result of analysis of this object. It is “making sense” of this visual attribute and an important step for computer vision (Gonzalez & Woods, 2002).

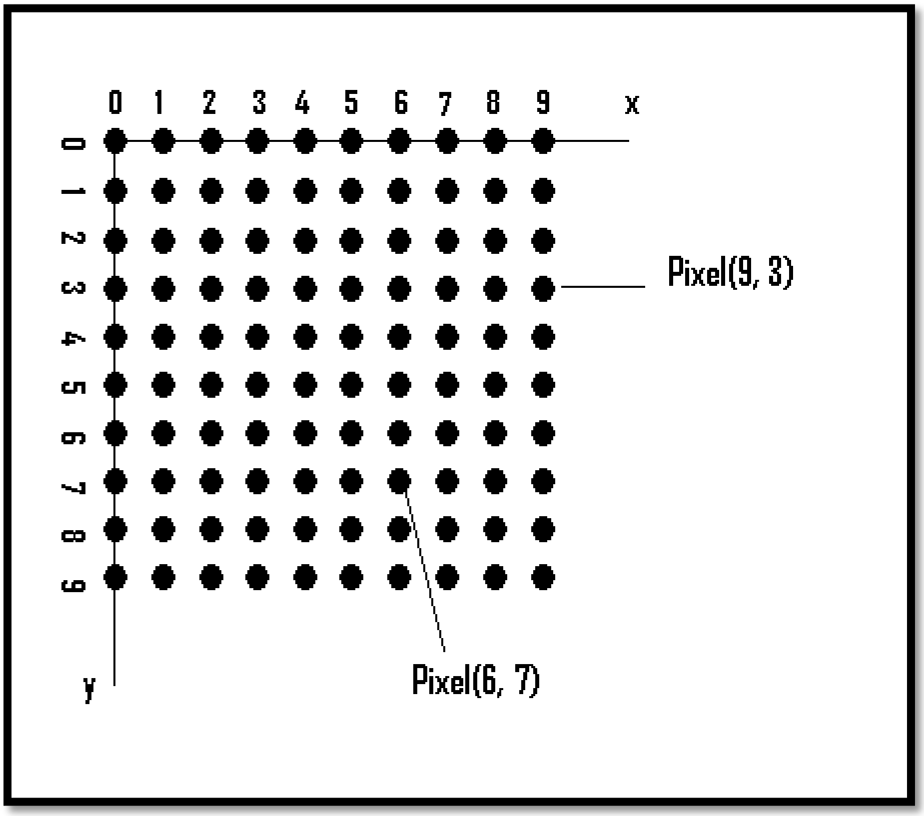


Figure 1 Pixel Matrix

Image Segmentation

Segmentation is the process of breaking an image into sets of pixels. Each set represents an image object. This mid-level method tries to identify a connected set of pixels that has common characteristics. There are two basic characteristics used to find objects in an image: uniformity, and continuity or discontinuity. Uniformity tries to find objects and boundaries in an image by assigning a label to every pixel in a way that pixels with the same label share certain visual characteristics such as low variance. The second property tries to detect objects by abrupt change in the gray level intensity of a pixel, such as point, line or edge detection (Gonzalez & Woods, 2002).

Another method for assessing continuity relates to requiring that the uniformity holds for sets of pixels and their 4 or 8 neighbors.

Applications for the segmentation process include medical imaging (to find pathologies, measure tissue volumes, etc), face and fingerprint recognition, machine vision and several others fields. The next subsections are briefly describing important aspects of image segmentation: K-Means Clustering, Image Thresholding Process, Connected Component Labeling, and Contour Tracing.

K-Means Clustering

K-means clustering is a process that attempts to find the centers of a data set through an iterative refinement approach. The steps for this process are:

1. k initial "means" (in this case k=2) are randomly selected from the data set.
2. k clusters are generated by associating every observation with the closest mean.
3. The center of each of the k clusters becomes the new means.
4. Steps 2 and 3 are repeated until convergence is achieved (MacKay, 2003).

Image Thresholding

The thresholding or binarization process can be used for image segmentation. The basic idea behind this method is to divide the pixels of a digital image into two main categories: object and background pixels. Each pixel is then colored white or black according to a pre-defined pixel threshold value. Figure 2 shows an example of a binary image created by this segmentation process.

Figure 2 Input and Output of Thresholding Process

The main aspect in the binarization process is the choice of the threshold value that best separates the object pixels from the non-object pixels. There are several techniques used to choose a threshold value. Some of these techniques are manual; others calculate the mean or median value. A different method uses K-Means with K=2 to find the two centers of the pixel values and calculate the average of these centers in order to find the threshold value.

Another option is to create an image histogram of the pixel intensities and use the valley point as the threshold. But sometimes an image does not have clearly defined valley points in the histogram and this generally makes the right selection of a threshold difficult. In cases where the gray level intensity of the object is similar to the background, an adaptative method is required to better separate the object (Shapiro & Stockman, 2002).

Connected Component Labeling

The next step in an image segmentation process is to identify different objects in the binary image. An object is represented by a group of interconnected pixels. Connected component labeling (CCL) is a process that identifies similar pixel regions based on connectivity considerations. The algorithm traverses the image and labels the pixels based on the connectivity and value of their neighbors. A pixel is considered connected to another pixel if they are neighbors. Figure 3 shows the two main types of neighbor connectivity: 4-connected or 8-connected pixels.

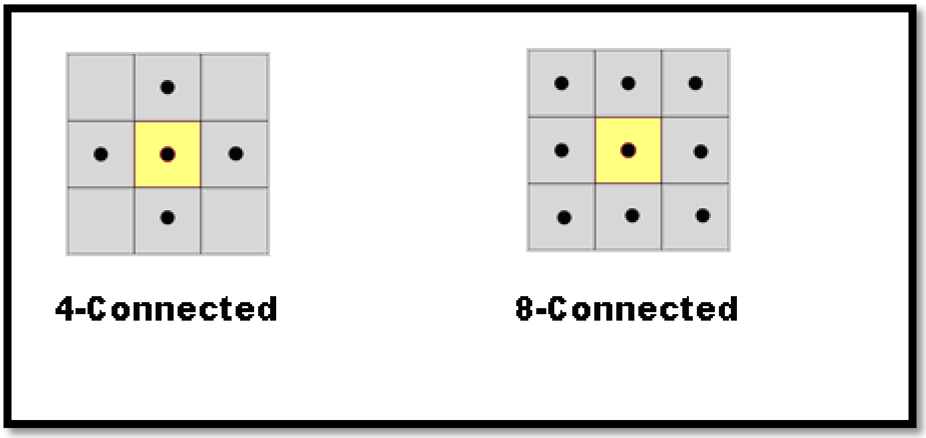


Figure 3 Example of 4-Connected and 8-Connected Pixels

A two-pass algorithm that iterates through 2-dimensional image binary data is a relatively simple example of CCL code. The first pass identifies and labels the pixels in connected regions. The second pass finds equivalent regions and merges these regions together (Shapiro & Stockman, 2002).

Contour Tracing

Contour Tracing is a method applied to digital images in order to extract the set of pixels that defines the object boundary. This technique is also known as border following or boundary following. Thresholding and CCL pre-processing methods are usually necessary before executing the contour algorithm since the pixel set of each object in an image needs to be identified and separated from the background pixels. However, there are exception cases where the contour is used to find the object pixels. This is based on the principle that a component is fully determined by its contours (Chang, Chen, & Lu, 2004).

One well-known contour tracing algorithm is the Moore-Neighbor Tracing. This method uses the 8-connectivity neighborhood (Moore neighborhood) for the boundary extraction. It extracts the contour by going around the pattern in a clockwise direction. Every time the process hits an object pixel P, it marks this point as a boundary pixel and backtracks to the background pixel. It is necessary to switch to a background pixel to guarantee that no edge element is left unvisited. This process continues around pixel P in a clockwise direction, visiting each pixel in its Moore neighborhood, until finding a black pixel again (Pavlidis, 1995).

Object Representation

The next step in the recognition process is to transform a group of related pixels into an abstraction useful for digital processing. An external representation is used to interpret the shape of an object. Boundary, length, and number of concavities in the edge are examples of external representation. An internal representation is used to analyze the internal characteristics of an object. Color and texture are examples of internal representation. Often, both types of representation are required (Gonzalez & Woods, 2002).

Two important types of object representation are described in the next subsections: Chain Code and Object Signature.

Chain Code

Chain code is a method used to represent a boundary in a digital image. This process describes the object boundary by a sequence of 2 or 3 bits long integers, based on a

4-connectivity or 8-connectivity segments respectively. The numbers indicate the change of direction from one pixel to the next pixel in the boundary (Gonzalez & Woods, 2002).

This process starts with a pre-defined pixel in the object contour. From this initial point the encoder moves to the next pixel keeping track of the movement executed. The change of direction is translated into a number representing the direction of the move. The encoder repeats these steps for all boundary pixels.

Figure 4 illustrates this representation method for a 4-connected and 8-connected direction map. Generally, the start pixel is the top left pixel, but for comparison purposes the example described below uses the bottom left pixel as the beginning point. The next element is positioned above the first one and the encoder translates this movement as the number mapped for an up movement in the grid. The third pixel is situated at the up-right side of the second pixel. In this scenario the encoder moves up-right for an 8-connected grid. For the 4-connected grid the encoder only moves up and loses some details in the image. Figure 4 shows the result chain codes for these 4 and 8-connected grids as 11000332222 and 21997664344 respectively (Gonzalez & Woods, 2002).

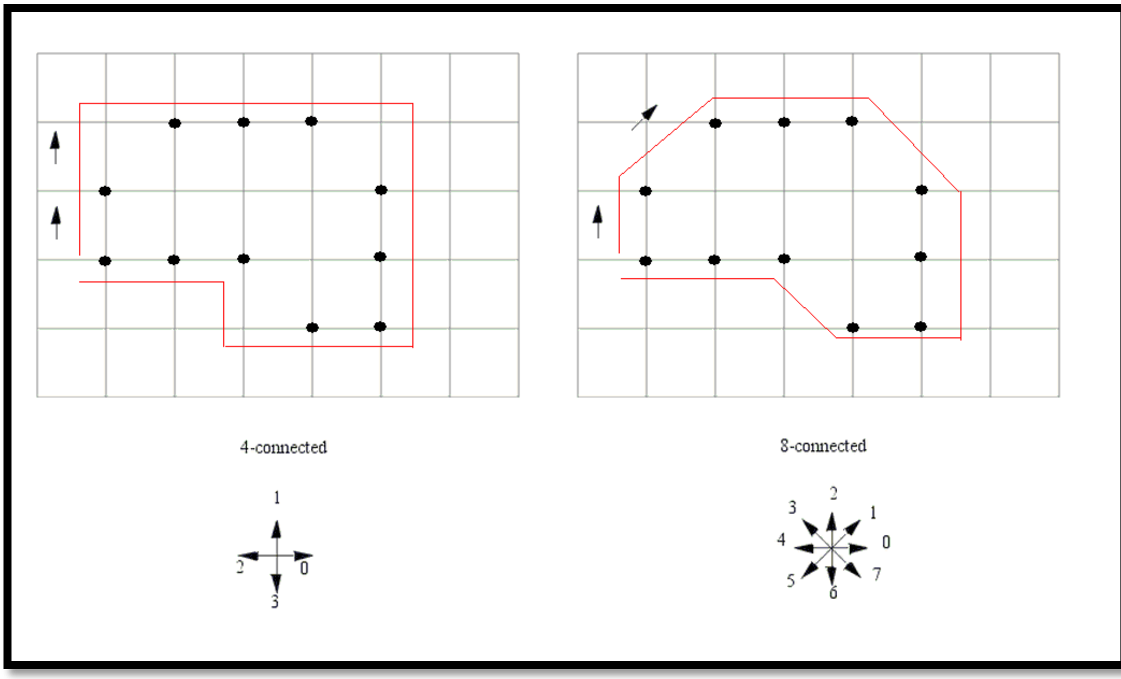


Figure 4 Direction Numbers for 4-Connected and 8-Connected Chain Codes

Object Signature

Object signature is a 1-D representation of an image object and/or its boundary. The main idea is to measure the distance between the object centroid and its boundary in equal angles, or using every pixel in the border. The object boundary can be generated in different methods. Figure 5 displays the signature of a rectangle and circle. Since the distance between the center of the circle and any point in the circumference is always the same (radius), the signature of a circle is a constant line and equal to the radius value.

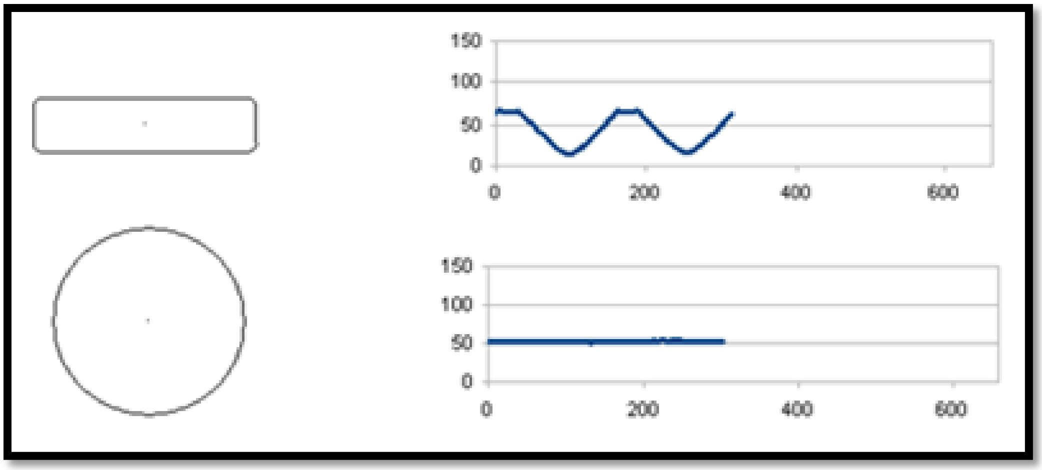


Figure 5 Object Signatures

The object signatures generated using the centroid of an object as the origin of a coordinate system is invariant to translation, but it is still dependent on the object scale and rotation. It is desired to normalize the signature in order to make it invariant to rotation and scale. The goal is to recognize the same instance of an object in another image, but with a different size and rotation. The cyclic autocorrelation method is utilized to minimize the rotation variance (Baggs & Tamir, 2008).

Object Recognition

Object Recognition is concerned with the ability to identify an object in an image. The main aspect of object recognition is the concept of “learning” by samples of objects. An object representation process generates a collection of descriptors. This object feature allows for a classification of these patterns and possible future recognition with as little human intervention as possible.

Matching is a recognition technique based on quantitative descriptors. One simple method is the minimum distance classifier. In this method, a pre-defined class metric is used to find unknown patterns. An object is included in a category if it is the closer Euclidian distance between this verified pattern feature and the known value of this class (Gonzalez & Woods, 2002).

Quantization

The principal of quantization is to convert a large set of input values into a smaller output set. This is a basic step in compression. A simple example is the rounding method from a real number to the closest integer. This operation is performed by a quantizer and since this is a many-to-few mapping, it is not always possible to recover the original input from the quantized input. Therefore, this function is used in the context of lossy compression methods.

Scalar quantization is the most common type and it is executed by a function used to map a scalar (one-dimensional) input value to a scalar output value.

A uniform quantizer has equal size intervals, with the exception of the two external intervals. In the previous example of rounding real numbers to integers, the intervals are equal to one. It is necessary to know the number of levels M expected in the output result in order to design a uniform quantizer. The interval Δ for an input range [$-X_{max}$, X_{max}] is calculated by the formula $\Delta = \frac{2X_{max}}{M}$. Since the quantizer is uniform, the expected quantization error is in the range$[-\frac{\Delta}{2}, \frac{\Delta}{2}]$.

For the rounding number example, the interval Δ = 1 and the error range is from -0.5 to 0.5. If 3.7, 1, -4.345, and 7.3 are the input values, the quantized output values are 4, 1, -4, and 7. The quantized error values, for this example, are -0.3, 0, -0.345, and 0.3 (Sayood, 2006).

Sum of Absolute Differences

Sum of absolute differences (SAD) is a simple algorithm commonly used for evaluating the similarity between image blocks. The process calculates the absolute difference between each pixel value in the original image and the correlated pixel in the image being used for comparison (Richardson, 2004).

SAD is generally applied in object recognition, the creation of disparity maps for stereo images, and motion estimation for video compression. SAD is a fast metric due to its simplicity. SAD can also be parallelized since it analyzes each pixel separately, making it easy to implement with hardware instructions (Richardson, 2004).

The Error Image

Image Subtraction is used to compute the difference between two images f(x, y) and h(x, y). This process generates a third image, called the error image g(x,y), by calculating the absolute value of the pixel difference between all pairs of corresponding pixels from images f() and h(). It is described by the formula:

$g(x, y) = |f(x, y) - h(x, y)|$ (Gonzalez & Woods, 2002).

In lossy image compression, the error image is used to visually compare the difference between the original and reconstructed image. If there is none or almost no difference between these images, the resultant error image appears nearly black.

Signal-to-noise Ratio

Signal-to-noise ratio (SNR) is a measurement used to calculate the ratio between the original signal and the unwanted signal (background noise). A ratio greater than 1:1 indicates that more signal is available than noise. Although this ratio is frequently applied to electrical signals, the concept can be applied to other scientific areas, such as lossy compression.

In data compression, SNR is used to present the difference between the reconstructed data and the original data. The noisy part of the SNR formula is calculated by the square error sum of the original value and the reconstructed value. This difference is referred to as the distortion. An efficient lossy compression algorithm has a higher SNR value. The formulas $\sigma_d^2 = \frac{1}{N} \sum_{n=1}^{N} (X_n - Y_n)^2$ and $SNR(dB) = 10 \; log_{10} \frac{\sigma_x^2}{\sigma\sigma_2^2}$ are used to calculate the SNR, where σ_x is the square sum of all values from the original source and σ_d is the mean square error, that is the sum square difference from the original value and the reconstructed value (Sayood, 2006).

Lossless Compression

Lossless Compression is a data encoding process that exploits data redundancy, in which the original data is reconstructed without any loss of information. This method is generally used where the process cannot accept any difference between the original and the reconstructed data (Sayood, 2006).

In general, the entropy of the source that generates the data is a lower bound on the compression ratio. One type of effective lossless data compression is used in the ZIP file format and in the UNIX utility gzip. This type of compression, known as dictionary techniques, is used in this thesis, due to its good performance and availability (Sayood, 2006).

Lossy Compression

Lossy compression is investigated in this research due to the fact that lossless compression does not yield a significant compression ratio. Lossy compression is a data encoding method in which the compression causes loss of part of the information during the process. Since some information is lost during the compression, it is not possible to recreate the same original data. This type of process is used in applications where it is acceptable to have a reconstruction error, such as voice or video transmission programs. Lossy algorithms are good candidates for compressing the data in these scenarios since a higher compression ratio is required and a small distortion in the transmitted data is tolerated (Sayood, 2006).

Lossy algorithms can be evaluated by the relative complexity, required memory, and speed needed in order to run in a specific hardware configuration. In addition to the performance measurement, there are two other important aspects of code compression: the compression ratio and the reconstruction quality.

A common method used to measure the amount of compression is the ratio between the number of bits required to store the data before and after compression. Another way to represent the compression method is by the amount of bits necessary to store a single sample. This is known as the rate.

The second performance measurement is related to the difference between the original data and the reconstructed compressed data. This difference is known as distortion. Since the quality of a recreated speech or video is a subjective measurement, it is difficult to create a mathematical model to cover the human response for this evaluation. SNR is one of the methods used to overcome this limitation. It is used in this thesis as the distortion measure.

The next part of this section describes four different types of lossy compression methods used in this thesis: DPCM, DLPC, Compressed Sensing, and DCT.

DPCM

Differential pulse-code modulation (DPCM) is a lossy compression method. This method takes advantage of the correlation between sequential data samples and transmits the quantized difference between the current element and the previous one.

A problem with this simple algorithm is that the encoder works with real data to calculate the difference, but the decoder uses the reconstructed data. This means that the quantization error accumulates throughout the reconstruction process. A common strategy to reduce the quantization noise is to enforce the encoder and decoder to work with the reconstructed data sequence. This enhanced encoding algorithm is known as the DPCM and it is mostly used for speech encoding in telephone systems (Sayood, 2006).

Differential LPC

Differential Linear Predictive Coding (DLPC) is a technique mostly used in audio and speech processing. The technique utilizes the information from a linear predictive model. The transmitter analyzes the input signal and creates the filter coefficients that best match the input segment. These coefficients are sent to the receiver and they are used to reconstruct the signal (Sayood, 2006).

Differential LPC is a variation of LPC. The information is compressed using a regular DPCM method, where the encoder quantizes the error between the real value and the predicted value. As in DPCM the encoder mimics the decoded and performs the prediction on reconstructed data (Sayood, 2006).

Compressed Sensing

Compressed Sensing (CS) is a sampling technique used to capture and reconstruct signals at a rate significantly below the Nyquist rate. This rate states that for an accurate reconstruction of a band-limited signal from its samples, the signal needs to be sampled at least twice its bandwidth. CS theory shows that it is possible to recover certain signals and images with zero or small distortion using much less samples than the number implied by the Nyquist theory. The CS method takes advantage of two common characteristics present in many interesting signals: sparsity and incoherence. Sparsity occurs when a signal has several coefficients close or equal to zero when analyzed in a specific domain such as the spectrum domain. Coherence measures the correlation between any two elements in the domain. CS is concerned with elements that have low coherence (correlation), or incoherence (Candès & Wakin, 2008).

DCT

The Discrete Cosine Transformation (DCT) is a lossy compression method based on a sequence of sum of cosine functions. The method is similar to and in fact can be obtained from the Discrete Fourier Transform function (DFT). However, in terms of compression, DCT has superior results (Sayood, 2006).

DCT is the recommended approach for audio (MP3) and images (JPEG/MPEG1 and MPEG2) because its rate-distortion is superior to many other lossy methods and its computational complexity is relatively low (Sayood, 2006).

CHAPTER III

RELATED WORK

He and Chao propose a fast CCL algorithm that requires only one and a half raster scanning. This is quicker than the conventional two passes approach. Their algorithm scans the background pixel once and does not require the relabeling of object pixels (He, Chao, & Suzuki, 2010).

Some CCL algorithms can be implemented at the hardware level. Ito and Nakano examine an implementation of CCL algorithm in Field Programmable Gate Array (FPGA). The advantage of using hardware circuits to implement the CCL code is its low latency. However, one of the problems with hardware implementation is the lack of storage space in the FPGA for large images. It is possible to use external memory such as DRMAs, if available, but this can be too costly in terms of price and power consumption (Ito & Nakano, 2010).

This thesis uses a custom developed Connected Component Labeling (CCL) algorithm to identify the pixels belonging to objects in the image. The algorithm, implemented in software, uses two passes through the image pixels to identify the objects because this approach is not complex to implement and the algorithm performance is not an important concern.

Das and Chande present a lossless adaptive DPCM schema that requires less computation than the conventional DPCM coding used in this project. The method uses Huffman code and it delivers a better compression rate than conventional DPCM techniques (Das & Chande, 2001).

Nikolaou and Papamarkos explore a new technique for retrieval of color images. Their research is based on the construction of a general purpose image database for object recognition. Their method requires the extraction of shape and color information in a set of images. The signature color extraction uses a fractal scanning procedure to create 1-D image signatures corresponding to each one of the image color components. This important feature is related to Discrete Cosine Transform (DCT) and Fourier Descriptor (FD) and is generated from the object signatures. This allows for an effective content-based image retrieval of color images from the database (Nikolaou & Papamarkos, 2002).

Baggs and Tamir investigate image registration using Dynamic Space Warping (DSW) for object recognition of natural and artificial images. DSW is an object spatial transformation technique. The image registration process spatially transforms two images of the same set of objects but of different geometries. The modification generates objects with the same size, shape, position, and orientation of objects from the other image. The method is an adaptation of the process used to align speeches known as Dynamic Time Warping (DTW). The recognition results of this study with natural images are overall successful with a low rate error and a high SNR value (Baggs & Tamir, 2008).

CHAPTER IV

EXPERIMENT SETUP

A custom application developed in the Java language is used to execute the experiments in this thesis. The code is developed using the Object Oriented approach, and the main concepts of the process required for the experiments are translated into classes in the application. The application has four main starting points related to the preparation and execution of the experiments:

1. Process to generate PGM images from bmp files.
2. Process to create a signature database from PGM images.
3. Process to reconstruct PGM images from a signature text file.
4. Process to recognize PGM images using a signature database.

The principal models developed are presented in the following list with a short description:

- PGMImage: represents an entire PGM image.
- ImageObject: represents the pixels of an object extracted from a PGM image.
- ConnectedComponentLabeling: encapsulates the algorithm to find connected pixels in an image. These pixels represent an image object.
- ContourTracing: encapsulates the algorithm to find the pixels only in the border of the object.

- Signature: represents a list of distances from the coordinates of the centroid to the pixels in the border of an object. The signature angle increment is 1 degree. This results in 360 measurements and the signature has the same number of entries. This increment is selected based on analysis of the shapes of objects used in our experiments.
- EqualAngleSignature: encapsulates the algorithm to extract a signature using uniform angle variation from an image object.
- SignatureDatabase: represents the set of signatures stored for future comparison.
- SignatureCompressor: encapsulates the algorithm to encode and decode a signature object to a file in the database.
- SignatureCompressorDCT: this subclass of SignatureCompressor adds DCT compression to the encode and decode process.
- SignatureDiferentialEncoding: this subclass of SignatureCompressor adds DPCM compression to the encode and decode process.
- SignatureCompressorLPC: this subclass of SignatureCompressor adds DLPC compression to the encode and decode process.
- SignatureGenerator: encapsulates the main logic to create a signature from a PGM image file.

The synthetic images are generated using Microsoft Paint. A Java module is responsible for converting these images to PGM format, using the ImageMagick conversion tool. Figure 6 shows the images stored in the signature database. These images are: circle, ellipse, triangle, square, rectangle, and non-convex star shaped object (Tamir, Shaked, Geerts, & Dolev, 2010). In addition to these six basic geometric figures, each individual image generates four variations. The variations are: rotation, rotation and scale, scale, and translation. This gives a total of 30 objects. These instances are used for comparison with the data in the signature database.

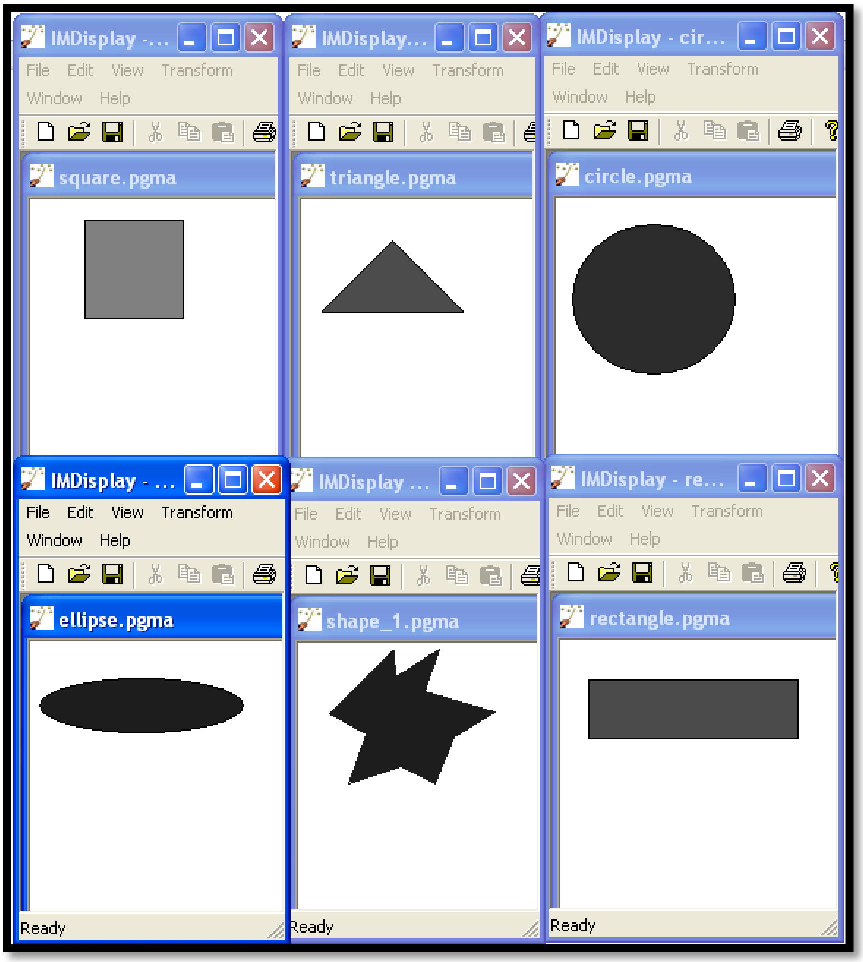


Figure 6 PGM Images Contained in the Signature Database

Process 1: Generating a PGM Image from BPM Files

Each shape used in the experiment is manually created using Microsoft Paint and stored using the BMP format. However, the custom Java application requires an image in a Portable Graymap format (PGM). Therefore, a conversion is required. The simple preparation process is a necessary step in order to convert all BMP images used in the experiments into the PGM plain text format. The final size of the PGM image is 250x250 pixels.

Figure 7 shows two new shapes that are not stored in the signature database: a pentagon and a concave shape. These two transient images have four variations each and they are used to test a negative scenario where a signature that does not exist in the database is compared to the signature entries. The total of images used for the comparison experiment is 40 PGM files.

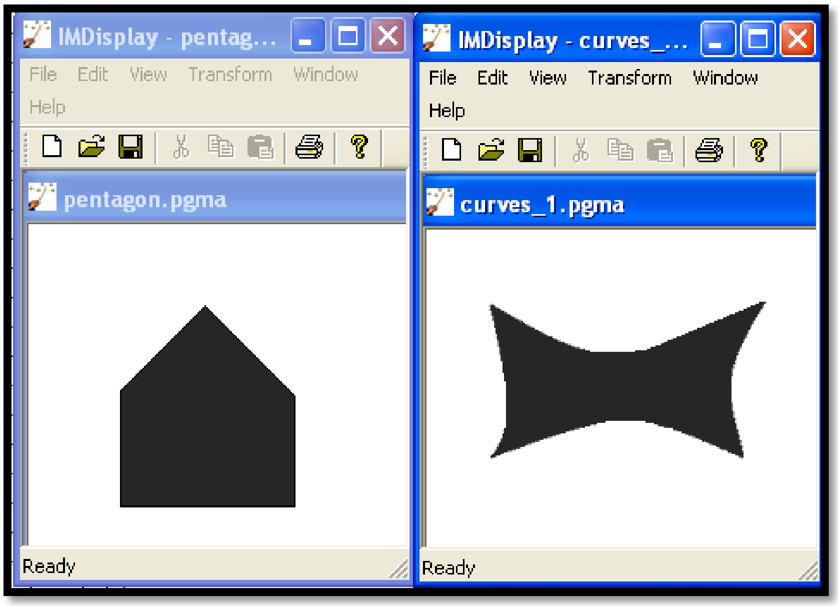


Figure 7 Images Not Stored in the Signature Database

Process 2: Creating a Signature Database from a PGM Image

This process uses a PGM image as an input and creates a signature file in a plain or compressed form as the output. Various steps are necessary in order to reach the final goal. The first step is the binarization or image thresholding process. The process generates a PGM image with two pixel values: black for the object and white for the background.

The second step is to find the pixels belonging to the background and the pixels representing the objects in the input image. This step uses a Connected Component Labeling technique to find all object pixels in the binary PGM image. The output of this method is a map and each entry is a list of coordinates of the object pixels. Although all the input images in this project only have one object, the algorithm is developed in such a way that it is capable to handle images with more than one object.

The third step is the identification of the object boundary. Once the object pixels are mapped, the contour trace algorithm executes the Moore Neighbor Tracing method to find the edge.

The fourth step is to generate the object signature using the geometric center of the object and calculating the distance from this point to selected pixels in the object border. The equal angle approach is used to get a Euclidian distance list with the same number of elements independent of the object size. The technique allows large object signatures to be compared to smaller ones. The total number of elements depends only on the angle selected. It does not depend on the number of pixels in the contour of the object. The value of the angle used in this thesis is one degree (360 measurements in this thesis). This degree value is enough to capture the border details for the executed experiments. Figure 8 shows a graphical representation of the entire process with a rectangle shape. The images representing the results of the threshold processing (rectangle_bin.pgma) and connected component labeling processing (rectangle_ccl.pgma) are the same because the input image (rectangle.pgma) only has one object. The graphical representation of the rectangle signature (rectangle_sig.pgma) also shows the border, but it uses fewer pixels than the real border represented in the rectangle_contour.pgma file because the edge of the object has more than 360 pixels.

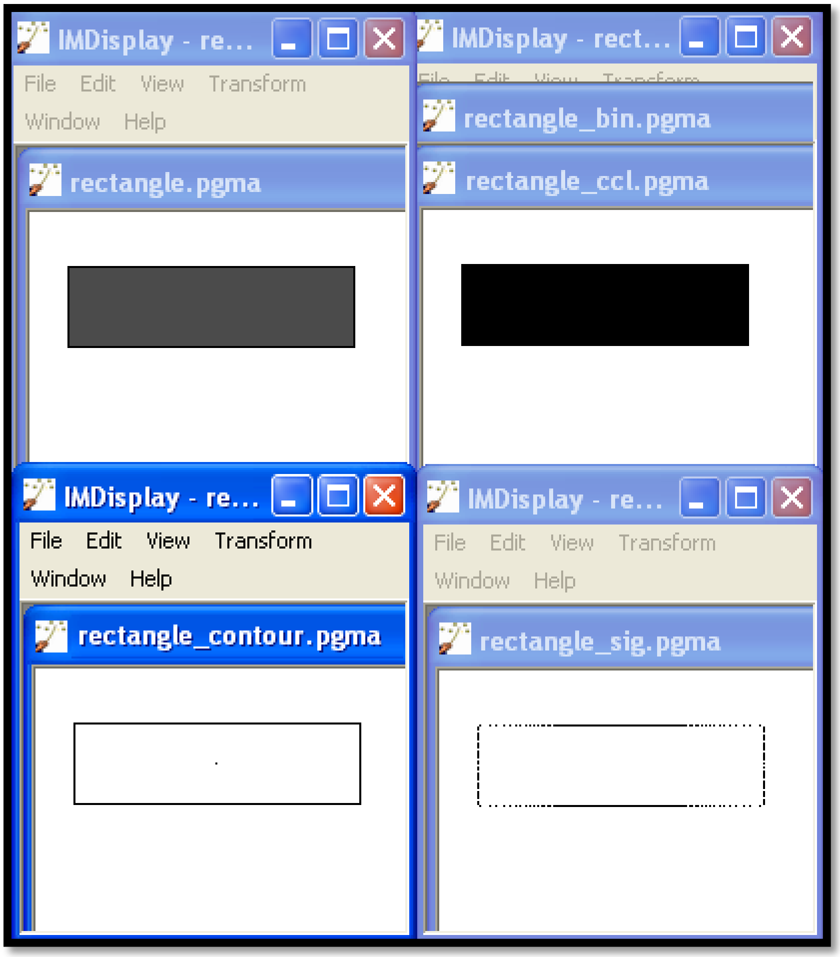


Figure 8 The Signature Creation Process

The final step in creating an object signature representation is the compression method. Since the method used to generate a signature is the one degree equal angle approach, the result list has 360 entries. For reference, a version of the signature stored without compression is furnished. The highest distance value in the signature list is used to normalize all other values. The range here goes from zero to one. Each entry is stored in 16 bits, so the normalized distance is multiplied by 2^{16} - 1. It is also necessary to store the maximum distance value without any modification (normalization) in order to recreate an image from the signature.

The compression methods applied to signatures are DPCM, DCT, DLPC, and Compressive Sensing. This custom developed application creates five output signature files for each input PGM image. Each one is compressed using one of the previously mentioned compression techniques and contains only one signature in its full representation.

The DPCM signature uses a 3 bit quantizer. The DCT stores the first 48 coefficients created by the DCT method. A Java DCT library is used for the actual DCT algorithm.

The Differential LPC method uses 10 LPC coefficients and uses 3 bit differential quantizer.

Compressive Sensing is developed using a Matlab library. The Java application generates a signature file from a PGM image and the Matlab script performs compression and decompression using the Compressive Sensing method. The signature file compressed by this method is used as an input for the Java application in order to recreate a PGM image and calculate the distortion rate.

Process 3: Reconstructing a PGM Image from a Signature File

This process developed in Java is responsible for recreating a PGM image from a signature file which contains either compressed or non-compressed signatures. Since the object signature represents 360 pixels in the object border, the image generated only has a contour with this number of pixels. Furthermore, if the signature is stored with lossy compression, the regenerated image also presents a distortion. Figure 9 shows an example of an image recreated from the non-convex star shaped object. The file shape_1_contour.pgma has the original border of the object and the other three files represent the recreated image from the respective signature file. The two bottom images are created from a compressed signature file and contain a significant distortion in the image recreated from a signature compressed with the Compressive Sensing method.

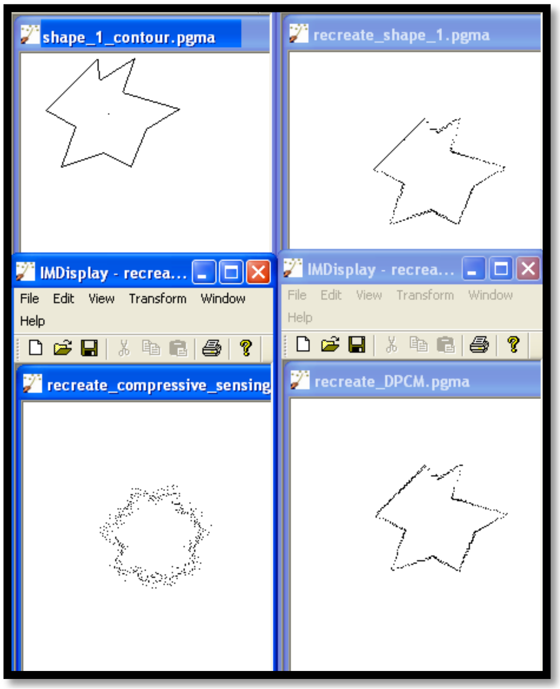


Figure 9 Non-convex Star Shaped Signature Recreation Process

The first step of the process is to read the signature file and decompress it, if necessary. Depending on the compression method, the signature file stores the name of method as the first line. Since Compressive Sensing is created using Matlab, the possible entries are: COMPRESS_NONE, COMPRESS_DCT, COMPRESS_DLPC, and COMPRESS_DIFFERENTIAL_ENCODING.

COMPRESS_NONE only requires that each entry be multiplied by the maximum distance that is stored without modification in the beginning of the file and that the result be divided by $2^{16} - 1$. The decode process is exactly the reverse of the signature encode process described earlier. These COMPRESS_NONE signature files are also used for evaluating the lossless compression rate of signatures. A ZIP application is utilized to compress the signature.

The lossy compression method requires decompressing before mapping the list of distance values from the object center to its border. The DPCM algorithm (COMPRESS_DIFFERENTIAL_ ENCODING) reads the maximum and minimum difference from the signature file to initialize the quantizer. It also retrieves the number of levels the quantizer uses. In this case there are 8 levels (3 bits storage for each difference). The first element in the list is the actual distance. For the subsequent entries, the stored value is the quantized difference from the previous entry.

The DCT method (COMPRESS_DCT) applies the inverse DCT function (iDCT) on the 48 DCT coefficients maintained by the DCT compression method. The Java algorithm developed for DLPC (COMPRESS_DLPC) loads the 10 LPC coefficients and 10 first sample distances from the signature file. The data is used for reconstruction using LPC. The result list is adjusted by the DPCM method by initializing the quantizer with the maximum distance, minimum distance and the number of levels (8 levels again).

The next step, after loading the signature list, is to convert the list of distances into a list of pixels. The first pixel defined is the center. The definition is achieved by reading the center information from the file and creating a pixel with its x and y coordinates. Next, the coordinates of each border pixel are calculated with a line function equation by using the distance stored in the file and the object center point. A rotation matrix is used to rotate the linear function by the correspondent degree.

The SNR is calculated by comparing the non-compressed signatures with the compressed signatures. More details about the results are explained in the Experiment Results and Evaluation section.

Process 4: Recognition of a PGM Image Using a Signature Database

This process compares the six basic shapes (circle, ellipse, triangle, square, rectangle, and non-convex star shaped object) with the four variation types (rotation, rotation and scale, scale, and translation) and the two shapes not in the database (pentagon and concave shape) against the signature in the database. Figure 6 in chapter IV illustrates the six basic shapes and Figure 7 in that same chapter shows the two shapes not in the database. Process 2: Creating a Signature Database from a PGM Image is a required before proceeding with this method.

The first step of this method is to load all signatures from the database to the memory. A similar process to read and decompress the signature (if necessary) is performed using the operations described in the previous section. The next step is to extract the signature from each input PGM image. The recent generated signature is compared to all signatures in the database. The result is a single number for each object in the database and it indicates the difference between the input object signature and the pre-stored signature in the database. If the value is close to zero, then it indicates a possible match between the signatures. A higher result indicates that the comparison of the distances between the center and the border of both objects is too disparate.

CHAPTER V

EXPERIMENT RESULTS AND EVALUATION

This section describes the results of the experiments executed as part of this thesis. The experiments are divided into four groups. The first experiment evaluates the recognition process of image signatures. The results are displayed in distribution graphs and tables for each input shape. The data in the graphs and tables are the sum of the absolute difference between the original and the compressed signature entry.

The second and third experiments are related to the image signature compression and reconstruction. These two experiments use SNR to evaluate the quality of the lossy compression methods. The fourth experiment evaluates the quality in the reconstruction process, but it does not use the object signature. It calculates the image distortion by comparing the difference between pixel values in the original and reconstructed image.

Experiment 1

This experiment consists of an object recognition process that compares the absolute difference of two distinct objects using a 360 entry equal angle signature. It uses process 1and process 2 described before. Process 1 is used to convert the experiment input into PGM images and process 2 is used to populate the signature database with the base shapes. Finally, this experiment executes the comparison method described in process 4, which consists of a comparison between the input image signatures and the signatures in the database.

Figure 6 in chapter IV describes the shape signatures in the image database. The set is composed of the signatures in the plain and compressed forms. For each shape the signature is compressed using one of the following methods: DPCM, DCT, DLPC, and Compressive Sensing (CS). The database has a total of 30 different entries which are used to find a possible match for each input image.

The input PGM images are divided into six different shape groups with five elements each. Among those five elements, one input image is the same as the one previously recorded in the signature database. The other four elements are instances of the same image with translation, scale, rotation and scale, and rotation. Figure 10 shows an example of these variations for an ellipse. The recognition experiment has a total of 32 input images. Figure 7 in chapter IV illustrates one group of two shapes non-existent in the database.

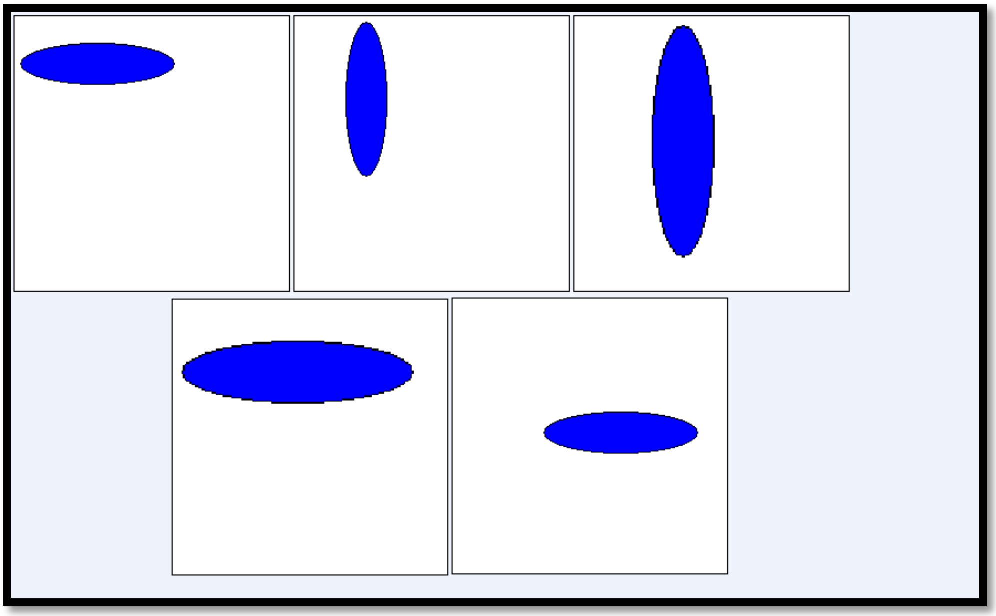
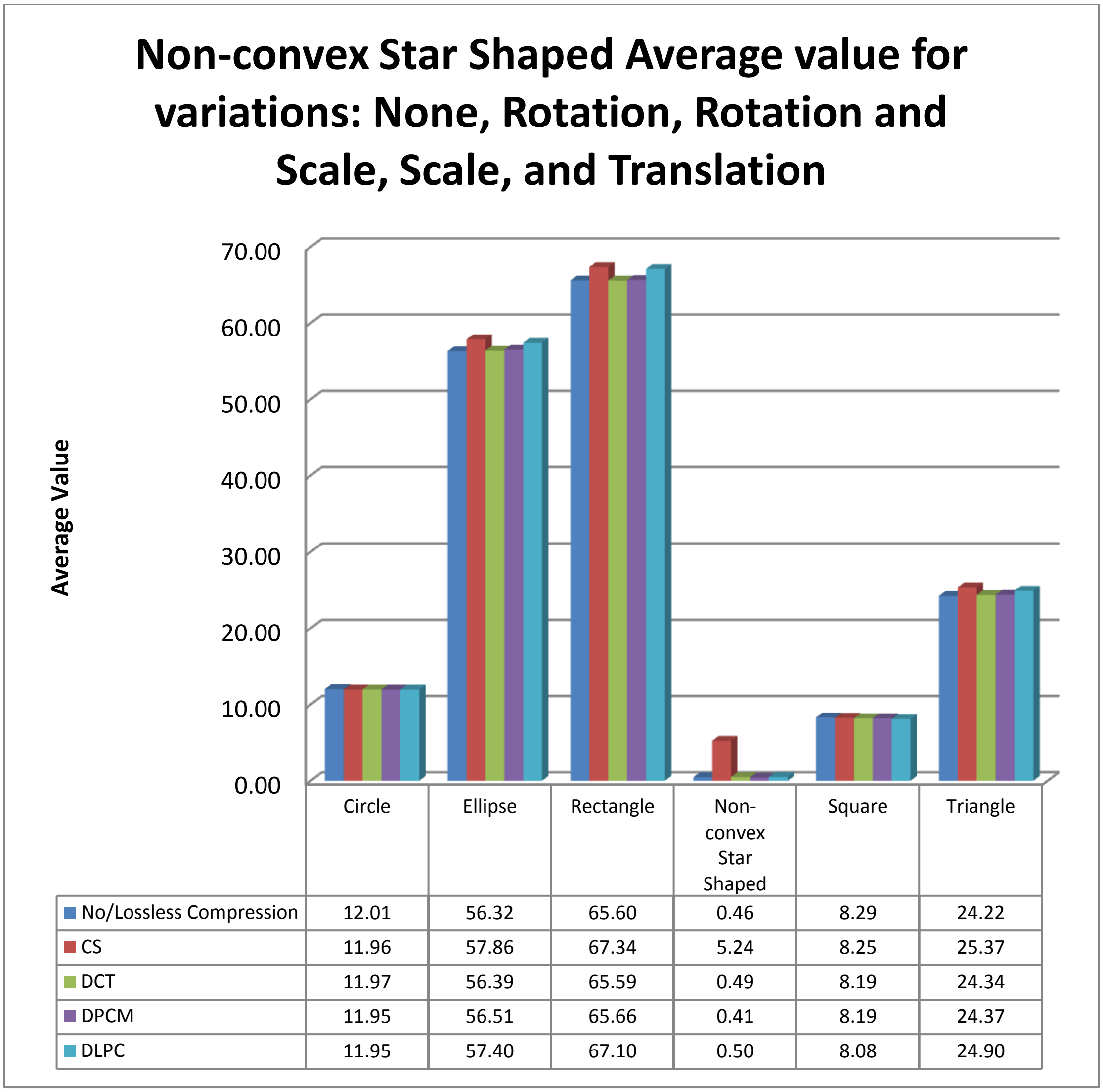


Figure 10 Ellipse Variations: Normal, Rotation, Rotation and Scale, Scale, and Translation

The non-convex star shaped object is the first image compared to the 30 entries in the signature database. Figure 11 shows the results of this test in a graph where the results of all five variations are combined to produce an arithmetic mean value. The average value of the five variations of the non-convex star shaped object has the lowest sum of absolute difference in comparison to a non-convex star shaped object signature stored in the database. The difference between the resultant values of a compressed and non-compressed signature is small, with the exception of the Compressive Sensing technique. This method has a lower recognition rate.



	Circle	Ellipse	Rectangle	Non-convex Star Shaped	Square	Triangle
No/Lossless Compression	12.01	56.32	65.60	0.46	8.29	24.22
CS	11.96	57.86	67.34	5.24	8.25	25.37
DCT	11.97	56.39	65.59	0.49	8.19	24.34
DPCM	11.95	56.51	65.66	0.41	8.19	24.37
DLPC	11.95	57.40	67.10	0.50	8.08	24.90

Figure 11 Comparing Non-convex Star Shaped to the Signature Database

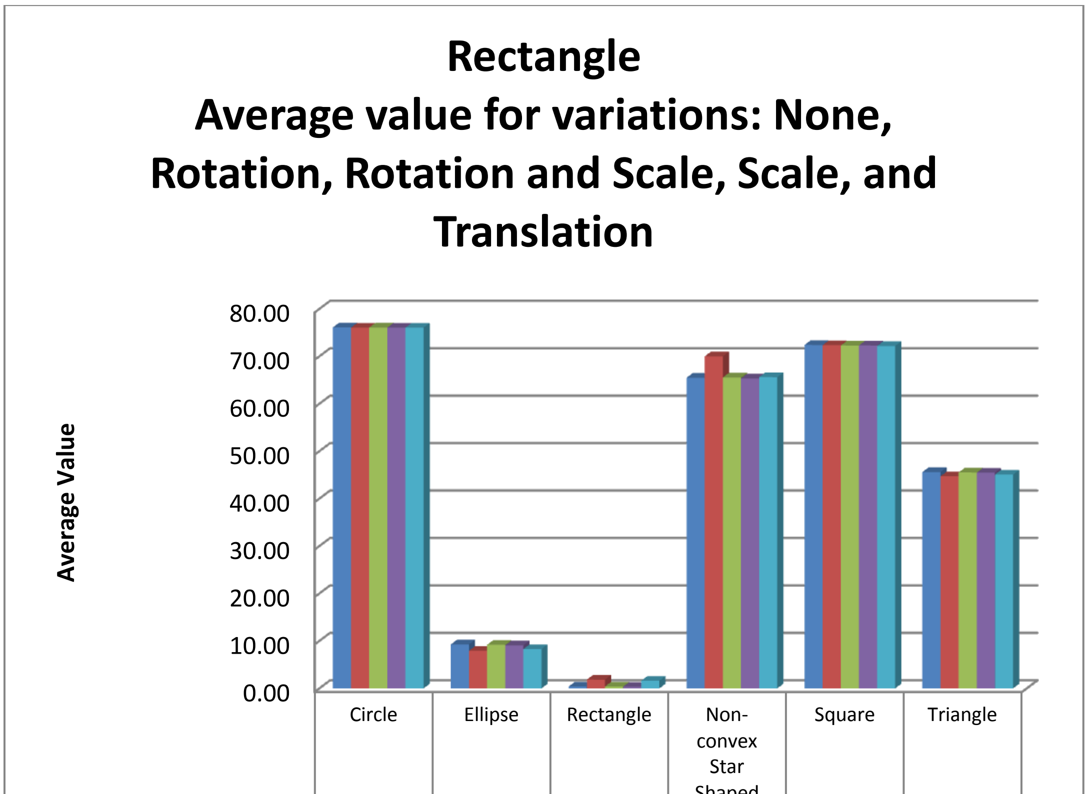
Table 1 shows the data used to build the graph in Figure 11. This table shows that, with the exception of Compressive Sensing, the sum of absolute differences of all signature elements is almost zero when comparing the input image to the non-convex star shaped signatures.

Table 1 Absolute Difference between Non-convex Star Shaped Signatures and the Signature Database

Signature DB	Non-convex Star Shaped	Rotation	Rotation and Scale	Scale	Translation
Circle	12.14	12.03	12.07	11.68	12.14
Circle CS	12.09	11.98	12.02	11.63	12.09
Circle DCT	12.10	11.99	12.03	11.64	12.10
Circle DPCM	12.08	11.98	12.01	11.62	12.08
Circle DLPC	12.07	11.97	12.01	11.62	12.07
Ellipse	56.17	56.44	56.31	56.52	56.17
Ellipse CS	57.71	57.98	57.84	58.06	57.71
Ellipse DCT	56.24	56.51	56.37	56.59	56.24
Ellipse DPCM	56.36	56.63	56.50	56.71	56.36
Ellipse DLPC	57.25	57.52	57.38	57.60	57.25
Rectangle	65.45	65.71	65.58	65.80	65.45
Rectangle CS	67.19	67.45	67.32	67.54	67.19
Rectangle DCT	65.44	65.71	65.58	65.79	65.44
Rectangle DPCM	65.51	65.77	65.64	65.86	65.51
Rectangle DLPC	66.95	67.21	67.08	67.30	66.95
Non-convex Star Shaped	0.20	0.84	0.51	0.57	0.20
Non-convex Star Shaped CS	5.36	5.26	5.29	4.91	5.36
Non-convex Star Shaped DCT	0.25	0.85	0.52	0.56	0.25
Non-convex Star Shaped DPCM	0.03	0.84	0.50	0.63	0.03
Non-convex Star Shaped DLPC	0.32	0.83	0.51	0.53	0.32
Square	8.42	8.31	8.35	7.96	8.42
Square CS	8.38	8.27	8.31	7.92	8.38
Square DCT	8.32	8.21	8.25	7.86	8.32
Square DPCM	8.32	8.21	8.25	7.86	8.32
Square DLPC	8.20	8.10	8.14	7.75	8.20
Triangle	24.09	24.21	24.17	24.56	24.09
Triangle CS	25.24	25.36	25.32	25.70	25.24
Triangle DCT	24.21	24.33	24.29	24.67	24.21
Triangle DPCM	24.24	24.36	24.32	24.70	24.24
Triangle DLPC	24.77	24.89	24.85	25.23	24.77

Although the non-convex star shaped signature stored using Compressive Sensing has a higher value in comparison to other compression methods, it still has a smaller value than other shapes. Based on this characteristic, the automated recognition process acknowledges that the image is in the database if the threshold is around five. This automated recognition process is based on the minimum distance classifier between the signature in the database and the input signature. That is, the sum of absolute differences among the elements in each signature.

The next image compared to the database is a rectangle. Figure 12 shows the results of this comparison. It is possible to observe that the process recognizes this image better than the previous one (non-convex star shaped). Table 2 shows that the Compressive Sensing method has a higher value, but the results are still closer to zero.



	Circle	Ellipse	Rectangle	Non-convex Star Shaped	Square	Triangle
No/Lossless Compression	76.12	9.27	0.40	65.51	72.40	45.62
CS	76.07	7.93	1.85	70.02	72.36	44.72
DCT	76.08	9.20	0.40	65.56	72.30	45.53
DPCM	76.07	9.08	0.35	65.36	72.30	45.49
DLPC	76.06	8.29	1.62	65.62	72.19	45.08

Figure 12 Comparing Rectangle to the Signature Database

The values generated in Table 2 are based on the signatures of each image in the database. The ellipse has the closest absolute sum of absolute differences value in comparison to the recognition results for the rectangle. This is due to the fact that both images have a similar shape.

Table 2 Absolute Difference between Rectangle Signatures and the Signature Database

Signature DB	Rectangle	Rotation	Rotation and Scale	Scale	Translation
Circle	76.33	76.33	75.03	76.58	76.33
Circle CS	76.28	76.28	74.98	76.53	76.28
Circle DCT	76.29	76.29	74.98	76.54	76.29
Circle DPCM	76.28	76.28	74.97	76.53	76.28
Circle DLPC	76.27	76.27	74.96	76.52	76.27
Ellipse	9.46	9.46	8.28	9.70	9.46
Ellipse CS	8.10	8.10	7.07	8.28	8.10
Ellipse DCT	9.38	9.38	8.23	9.62	9.38
Ellipse DPCM	9.26	9.26	8.12	9.49	9.26
Ellipse DLPC	8.47	8.47	7.40	8.66	8.47
Rectangle	0.14	0.14	1.19	0.39	0.14
Rectangle CS	1.64	1.64	2.94	1.39	1.64
Rectangle DCT	0.14	0.14	1.19	0.39	0.14
Rectangle DPCM	0.06	0.06	1.25	0.31	0.06
Rectangle DLPC	1.41	1.41	2.71	1.15	1.41
Non-convex Star Shaped	65.72	65.72	64.43	65.97	65.72
Non-convex Star Shaped CS	70.23	70.23	68.93	70.48	70.23
Non-convex Star Shaped DCT	65.77	65.77	64.48	66.02	65.77
Non-convex Star Shaped DPCM	65.57	65.57	64.27	65.82	65.57
Non-convex Star Shaped DLPC	65.83	65.83	64.53	66.08	65.83
Square	72.61	72.61	71.31	72.86	72.61
Square CS	72.57	72.57	71.26	72.82	72.57
Square DCT	72.51	72.51	71.20	72.76	72.51
Square DPCM	72.51	72.51	71.21	72.76	72.51
Square DLPC	72.40	72.40	71.09	72.65	72.40
Triangle	45.82	45.82	44.58	46.05	45.82
Triangle CS	44.92	44.92	43.69	45.15	44.92
Triangle DCT	45.73	45.73	44.49	45.96	45.73
Triangle DPCM	45.69	45.69	44.46	45.92	45.69
Triangle DLPC	45.28	45.28	44.05	45.51	45.28

Figure 13 demonstrates the similarity between the ellipse and the rectangle signatures. The main difference between these signatures is in the peak regions of the graph line. An ellipse image generates a graph with a round peak and the rectangle has a flat peak.

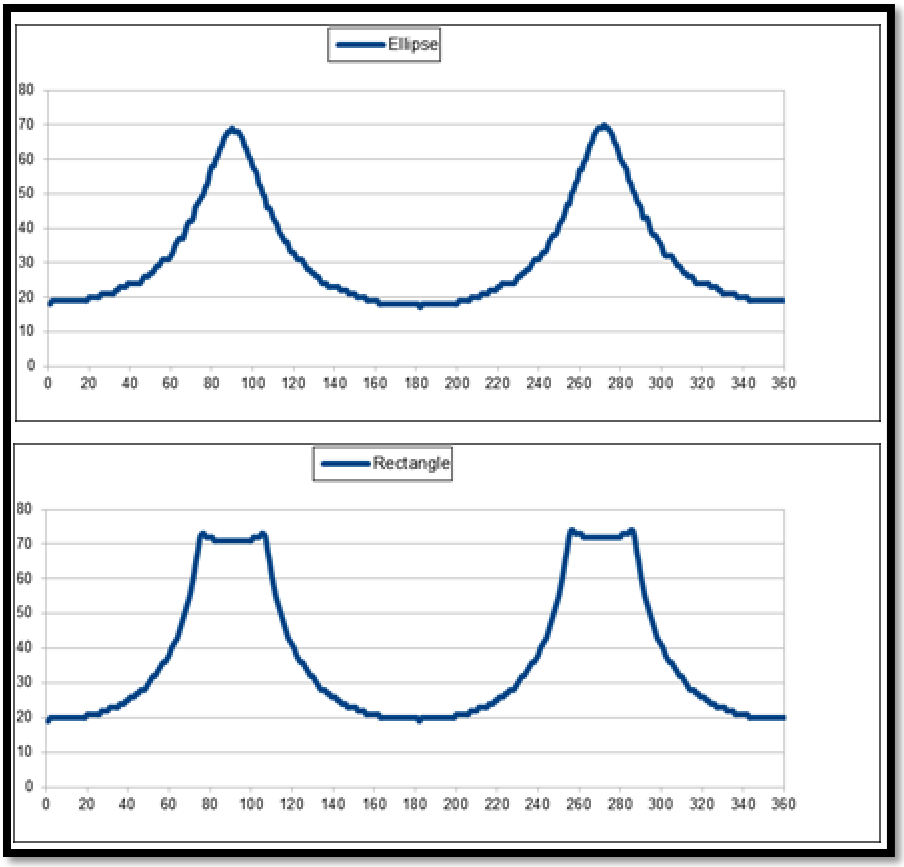
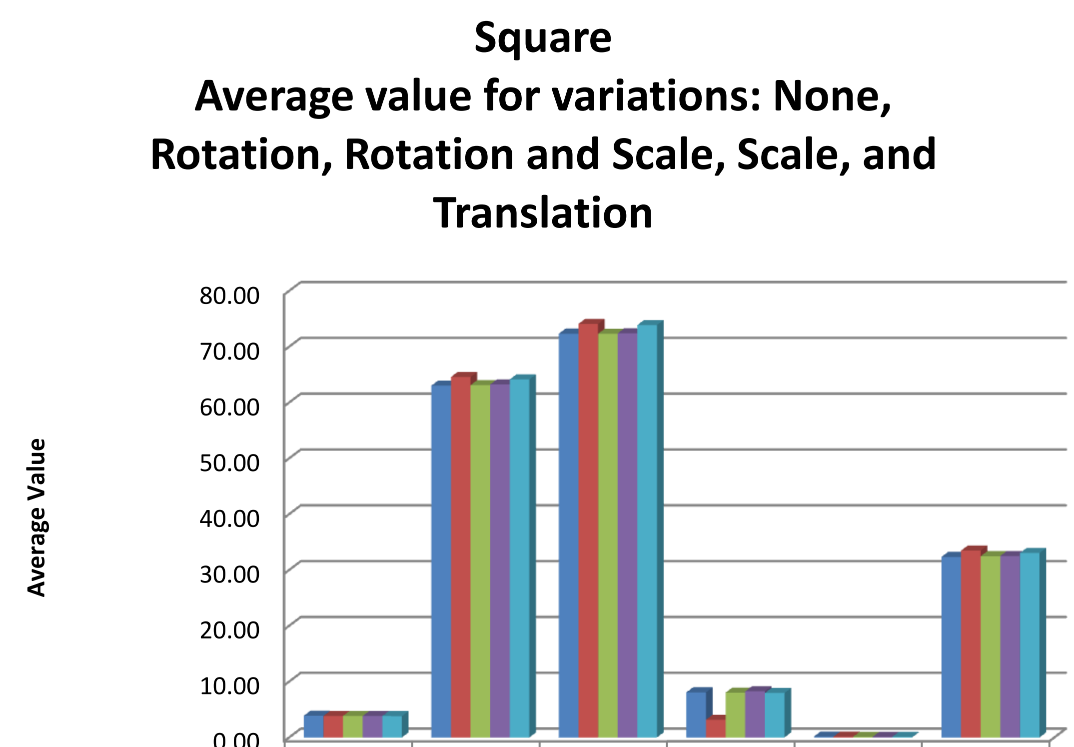


Figure 13 Ellipse and Rectangle Signatures

The square is the next image processed; and Figure 14 illustrates the graph results. The values in the graph are very low for all variations and compression methods for this input shape. This means that a small threshold value is sufficient for automated image recognition.



	Circle	Ellipse	Rectangle	Non-convex Star Shaped	Square	Triangle
No/Lossless Compression	3.89	63.03	72.35	8.09	0.17	32.35
CS	3.85	64.59	74.10	3.18	0.15	33.49
DCT	3.85	63.12	72.34	8.03	0.13	32.48
DPCM	3.84	63.25	72.42	8.27	0.13	32.52
DLPC	3.83	64.16	73.89	7.97	0.12	33.05

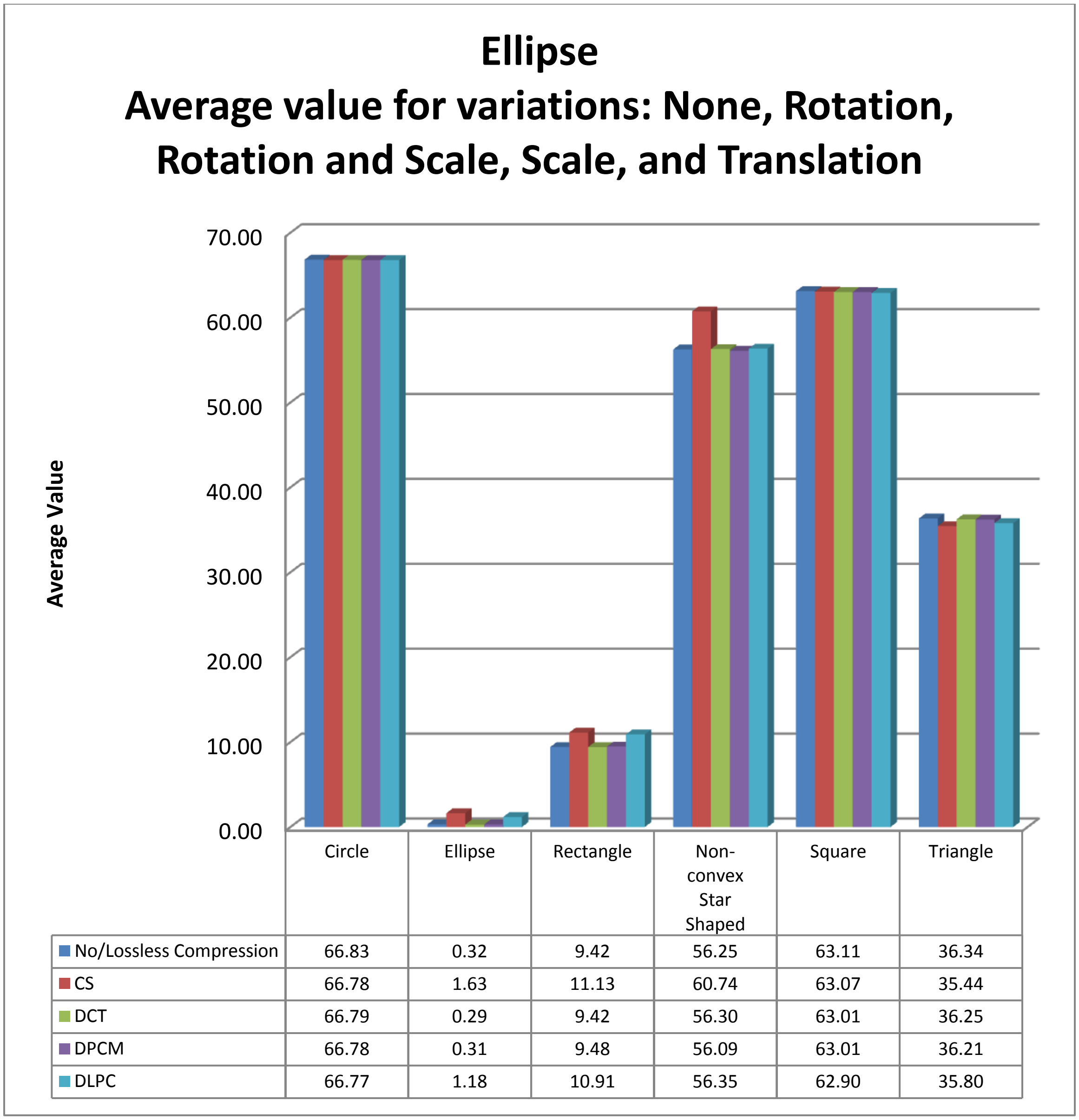
Figure 14 Comparing Square to the Signature Database

Table 3 shows that the values for the Square are closer to zero and this is the ideal result for the recognition process.

Table 3 Absolute Difference between Square Signatures and the Signature Database

Signature DB	Square	Rotation	Rotation and Scale	Scale	Translation
Circle	3.98	3.98	3.75	3.75	3.98
Circle CS	3.94	3.94	3.71	3.71	3.94
Circle DCT	3.94	3.94	3.72	3.72	3.94
Circle DPCM	3.93	3.93	3.70	3.70	3.93
Circle DLPC	3.92	3.92	3.70	3.70	3.92
Ellipse	62.93	62.93	63.17	63.17	62.93
Ellipse CS	64.49	64.49	64.73	64.73	64.49
Ellipse DCT	63.02	63.02	63.26	63.26	63.02
Ellipse DPCM	63.15	63.15	63.39	63.39	63.15
Ellipse DLPC	64.06	64.06	64.30	64.30	64.06
Rectangle	72.25	72.25	72.49	72.49	72.25
Rectangle CS	74.00	74.00	74.24	74.24	74.00
Rectangle DCT	72.24	72.24	72.48	72.48	72.24
Rectangle DPCM	72.32	72.32	72.56	72.56	72.32
Rectangle DLPC	73.79	73.79	74.03	74.03	73.79
Non-convex Star Shaped	7.99	7.99	8.23	8.23	7.99
Non-convex Star Shaped CS	3.13	3.13	3.25	3.25	3.13
Non-convex Star Shaped DCT	7.94	7.94	8.17	8.17	7.94
Non-convex Star Shaped DPCM	8.17	8.17	8.41	8.41	8.17
Non-convex Star Shaped DLPC	7.87	7.87	8.11	8.11	7.87
Square	0.23	0.23	0.07	0.07	0.23
Square CS	0.19	0.19	0.08	0.08	0.19
Square DCT	0.13	0.13	0.12	0.12	0.13
Square DPCM	0.13	0.13	0.12	0.12	0.13
Square DLPC	0.04	0.04	0.23	0.23	0.04
Triangle	32.25	32.25	32.49	32.49	32.25
Triangle CS	33.39	33.39	33.63	33.63	33.39
Triangle DCT	32.38	32.38	32.62	32.62	32.38
Triangle DPCM	32.42	32.42	32.66	32.66	32.42
Triangle DLPC	32.95	32.95	33.19	33.19	32.95

The next image evaluated is the ellipse. Figure 15 shows that the recognition process is capable of identifying all five variations of this shape. Table 4 shows the data results. The rectangle is the closest image to the ellipse. The reason for that is based on the similarity between their signatures.



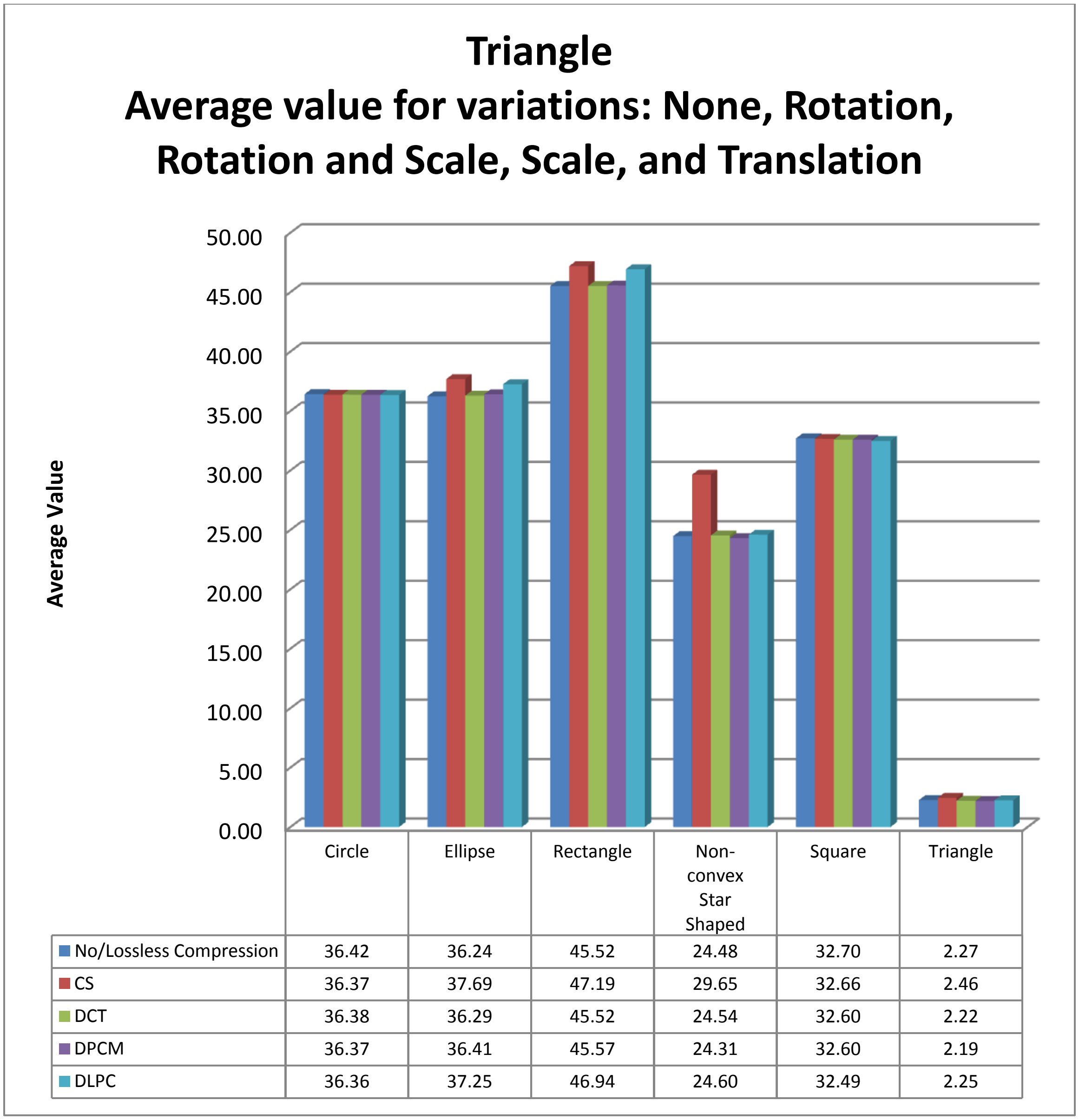
	Circle	Ellipse	Rectangle	Non-convex Star Shaped	Square	Triangle
No/Lossless Compression	66.83	0.32	9.42	56.25	63.11	36.34
CS	66.78	1.63	11.13	60.74	63.07	35.44
DCT	66.79	0.29	9.42	56.30	63.01	36.25
DPCM	66.78	0.31	9.48	56.09	63.01	36.21
DLPC	66.77	1.18	10.91	56.35	62.90	35.80

Figure 15 Comparing Ellipse to the Signature Database

Table 4 Absolute Difference between Ellipse Signatures and the Signature Database

Signature DB	Ellipse	Rotation	Rotation and Scale	Scale	Translation
Circle	67.09	67.10	66.46	66.43	67.09
Circle CS	67.04	67.05	66.41	66.38	67.04
Circle DCT	67.05	67.05	66.42	66.39	67.05
Circle DPCM	67.04	67.04	66.40	66.38	67.04
Circle DLPC	67.03	67.03	66.40	66.37	67.03
Ellipse	0.21	0.21	0.46	0.49	0.21
Ellipse CS	1.38	1.38	2.00	2.02	1.38
Ellipse DCT	0.13	0.13	0.52	0.55	0.13
Ellipse DPCM	0.07	0.07	0.65	0.67	0.07
Ellipse DLPC	0.92	0.92	1.55	1.57	0.92
Rectangle	9.20	9.19	9.75	9.78	9.20
Rectangle CS	10.87	10.87	11.50	11.52	10.87
Rectangle DCT	9.19	9.19	9.75	9.77	9.19
Rectangle DPCM	9.24	9.24	9.82	9.85	9.24
Rectangle DLPC	10.65	10.65	11.29	11.31	10.65
Non-convex Star Shaped	56.50	56.50	55.88	55.86	56.50
Non-convex Star Shaped CS	60.99	61.00	60.37	60.35	60.99
Non-convex Star Shaped DCT	56.55	56.55	55.93	55.91	56.55
Non-convex Star Shaped DPCM	56.34	56.34	55.73	55.70	56.34
Non-convex Star Shaped DLPC	56.60	56.60	55.99	55.96	56.60
Square	63.37	63.38	62.74	62.71	63.37
Square CS	63.33	63.33	62.70	62.67	63.33
Square DCT	63.27	63.27	62.64	62.61	63.27
Square DPCM	63.27	63.27	62.64	62.61	63.27
Square DLPC	63.16	63.16	62.53	62.50	63.16
Triangle	36.57	36.57	36.00	35.98	36.57
Triangle CS	35.67	35.67	35.10	35.08	35.67
Triangle DCT	36.48	36.49	35.91	35.89	36.48
Triangle DPCM	36.45	36.45	35.87	35.85	36.45
Triangle DLPC	36.03	36.03	35.46	35.44	36.03

Figure 16 and Table 5 show that among all evaluated images, the triangle is the most affected by rotation and scale. The sole rotation has the highest result in comparison to other triangle variations, but the value is still low in comparison to other shapes. This is sufficient for the success in a recognition process.



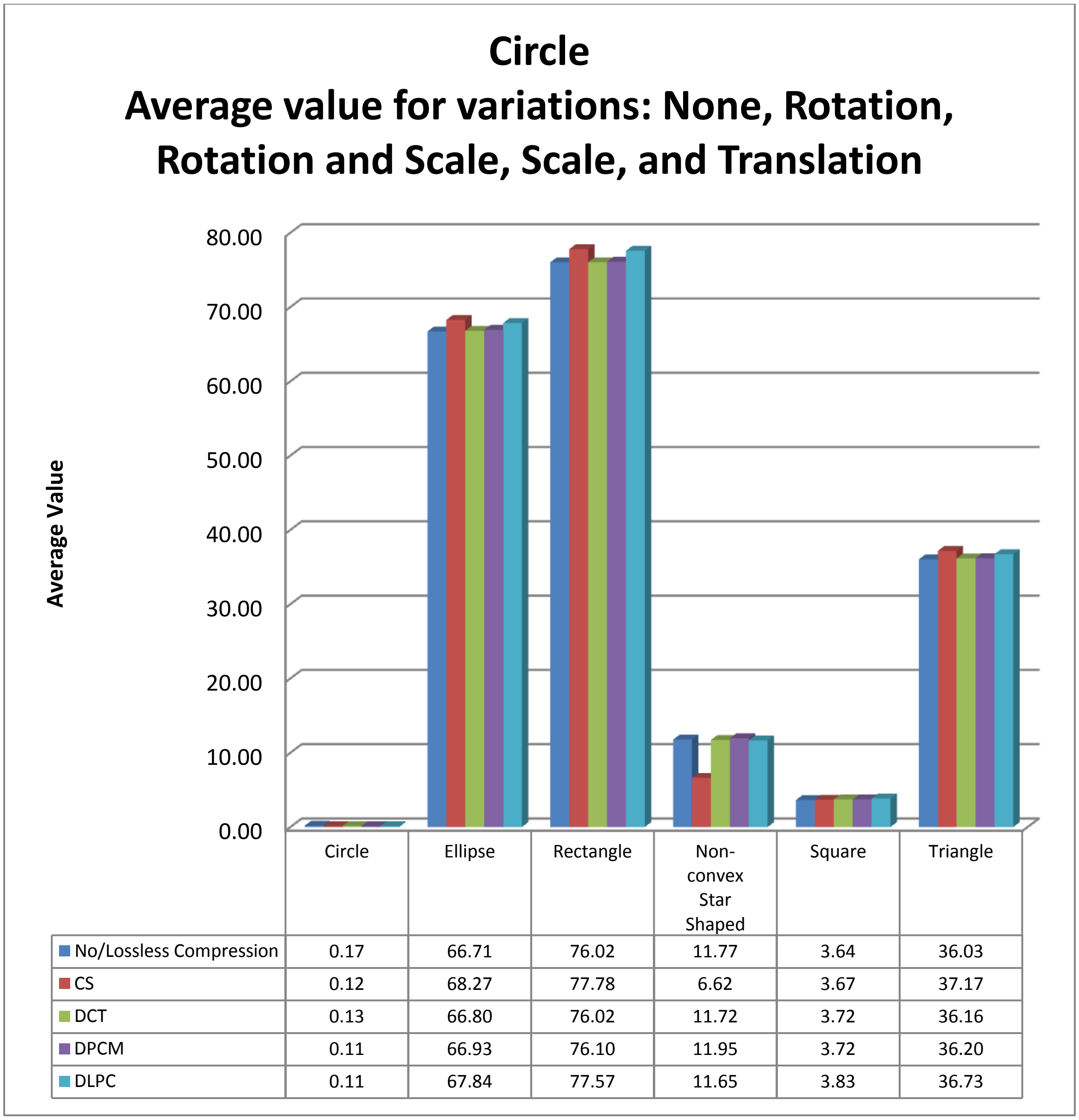
	Circle	Ellipse	Rectangle	Non-convex Star Shaped	Square	Triangle
No/Lossless Compression	36.42	36.24	45.52	24.48	32.70	2.27
CS	36.37	37.69	47.19	29.65	32.66	2.46
DCT	36.38	36.29	45.52	24.54	32.60	2.22
DPCM	36.37	36.41	45.57	24.31	32.60	2.19
DLPC	36.36	37.25	46.94	24.60	32.49	2.25

Figure 16 Comparing Triangle to the Signature Database

Table 5 Absolute Difference between Triangle Signatures and the Signature Database

Signature DB	Triangle	Rotation	Rotation and Scale	Scale	Translation
Circle	36.59	40.76	32.98	35.18	36.59
Circle CS	36.54	40.71	32.93	35.13	36.54
Circle DCT	36.55	40.71	32.94	35.14	36.55
Circle DPCM	36.54	40.70	32.93	35.12	36.54
Circle DLPC	36.53	40.69	32.92	35.12	36.53
Ellipse	36.16	33.84	38.56	36.50	36.16
Ellipse CS	37.61	35.24	40.03	37.97	37.61
Ellipse DCT	36.21	33.89	38.61	36.55	36.21
Ellipse DPCM	36.32	34.00	38.73	36.66	36.32
Ellipse DLPC	37.17	34.81	39.59	37.52	37.17
Rectangle	45.44	43.12	47.84	45.75	45.44
Rectangle CS	47.12	44.75	49.53	47.45	47.12
Rectangle DCT	45.44	43.12	47.83	45.75	45.44
Rectangle DPCM	45.49	43.16	47.89	45.80	45.49
Rectangle DLPC	46.86	44.49	49.28	47.19	46.86
Non-convex Star Shaped	24.65	28.81	21.04	23.24	24.65
Non-convex Star Shaped CS	29.82	33.98	26.21	28.40	29.82
Non-convex Star Shaped DCT	24.71	28.87	21.10	23.30	24.71
Non-convex Star Shaped DPCM	24.48	28.64	20.87	23.06	24.48
Non-convex Star Shaped DLPC	24.77	28.93	21.16	23.36	24.77
Square	32.87	37.04	29.26	31.46	32.87
Square CS	32.83	36.99	29.22	31.42	32.83
Square DCT	32.77	36.93	29.16	31.36	32.77
Square DPCM	32.77	36.93	29.16	31.36	32.77
Square DLPC	32.66	36.82	29.05	31.25	32.66
Triangle	0.39	4.56	3.24	2.75	0.39
Triangle CS	0.81	3.45	4.38	2.85	0.81
Triangle DCT	0.27	4.43	3.36	2.75	0.27
Triangle DPCM	0.23	4.39	3.39	2.71	0.23
Triangle DLPC	0.33	3.85	3.92	2.80	0.33

The circle is the last image analyzed in this experiment. The circle has the best recognition results in comparison to all other shapes. That is due to the constant value of the distance between the center and the boundary of the image. The signature compression has little data loss. Another reason for that is based on the fact that the circle is less affected by rotation or scale variation. Figure 17 and Table 6 show that the recognition results for the circle are almost zero.



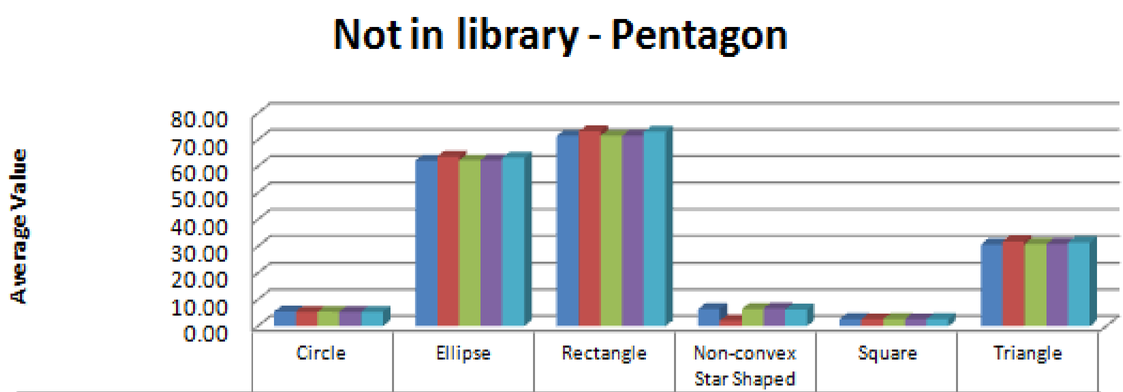
	Circle	Ellipse	Rectangle	Non-convex Star Shaped	Square	Triangle
No/Lossless Compression	0.17	66.71	76.02	11.77	3.64	36.03
CS	0.12	68.27	77.78	6.62	3.67	37.17
DCT	0.13	66.80	76.02	11.72	3.72	36.16
DPCM	0.11	66.93	76.10	11.95	3.72	36.20
DLPC	0.11	67.84	77.57	11.65	3.83	36.73

Figure 17 Comparing Circle to the Signature Database

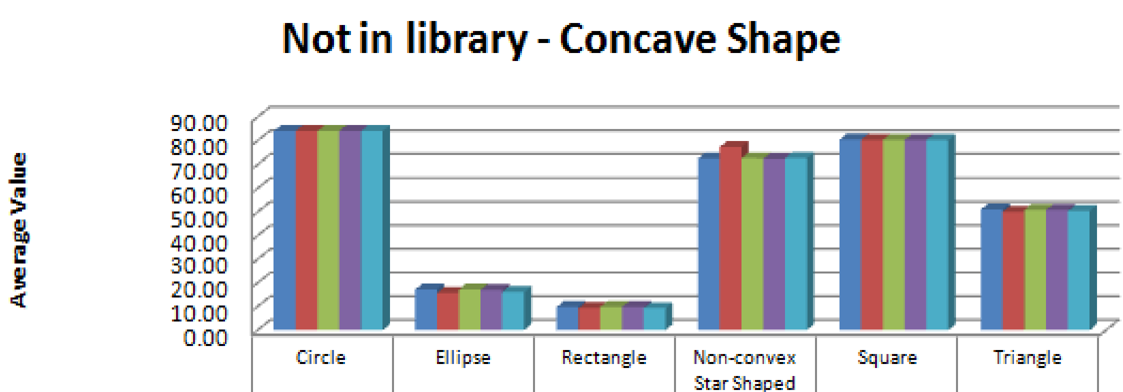
Table 6 Absolute Difference between Circle Signatures and the Signature Database

Signature DB	Circle	Rotation	Rotation and Scale	Scale	Translation
Circle	0.12	0.12	0.09	0.40	0.12
Circle CS	0.07	0.07	0.04	0.35	0.07
Circle DCT	0.08	0.08	0.05	0.36	0.08
Circle DPCM	0.06	0.06	0.03	0.34	0.06
Circle DLPC	0.06	0.06	0.03	0.34	0.06
Ellipse	66.76	66.76	66.79	66.49	66.76
Ellipse CS	68.32	68.32	68.35	68.04	68.32
Ellipse DCT	66.85	66.85	66.88	66.57	66.85
Ellipse DPCM	66.98	66.98	67.01	66.70	66.98
Ellipse DLPC	67.89	67.89	67.92	67.60	67.89
Rectangle	76.07	76.07	76.10	75.80	76.07
Rectangle CS	77.83	77.83	77.86	77.55	77.83
Rectangle DCT	76.07	76.07	76.10	75.79	76.07
Rectangle DPCM	76.15	76.15	76.18	75.87	76.15
Rectangle DLPC	77.62	77.62	77.65	77.34	77.62
Non-convex Star Shaped	11.82	11.82	11.85	11.54	11.82
Non-convex Star Shaped CS	6.67	6.67	6.70	6.39	6.67
Non-convex Star Shaped DCT	11.77	11.77	11.80	11.48	11.77
Non-convex Star Shaped DPCM	12.00	12.00	12.03	11.72	12.00
Non-convex Star Shaped DLPC	11.70	11.70	11.73	11.42	11.70
Square	3.68	3.68	3.70	3.46	3.68
Square CS	3.71	3.71	3.74	3.49	3.71
Square DCT	3.76	3.76	3.79	3.53	3.76
Square DPCM	3.76	3.76	3.79	3.53	3.76
Square DLPC	3.87	3.87	3.90	3.64	3.87
Triangle	36.08	36.08	36.11	35.80	36.08
Triangle CS	37.22	37.22	37.25	36.94	37.22
Triangle DCT	36.21	36.21	36.24	35.92	36.21
Triangle DPCM	36.25	36.25	36.28	35.97	36.25
Triangle DLPC	36.78	36.78	36.81	36.50	36.78

Figure 7 in chapter IV shows two images that are used in the recognition process but are not present in the database. These images are: concave shape and pentagon. Figure 18 and Table 7 demonstrate the results for this test. The square has the closest value in comparison to the pentagon. Figure 19 shows the similarity between both signatures.



	Circle	Ellipse	Rectangle	Non-convex Star Shaped	Square	Triangle
No/Lossless Compression	5.47	62.19	71.48	6.47	2.64	30.73
CS	5.42	63.74	73.23	2.10	2.62	31.87
DCT	5.43	62.26	71.48	6.41	2.63	30.85
DPCM	5.42	62.39	71.55	6.64	2.62	30.89
DLPC	5.41	63.28	73.00	6.35	2.64	31.43

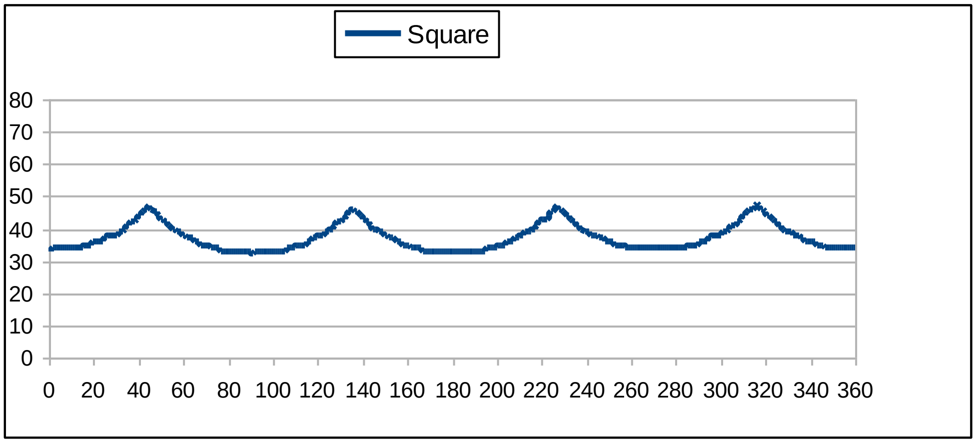


	Circle	Ellipse	Rectangle	Non-convex Star Shaped	Square	Triangle
No/Lossless Compression	84.16	17.27	9.81	72.52	80.44	51.04
CS	84.11	15.71	9.32	77.40	80.39	50.06
DCT	84.11	17.19	9.81	72.57	80.33	50.94
DPCM	84.10	17.05	9.77	72.35	80.33	50.90
DLPC	84.09	16.15	9.30	72.63	80.22	50.46

Figure 18 Comparing Shapes Not in the Database

Table 7 Absolute Difference between Comparing Shapes Not in the Database

Signature DB	Pentagon	Concave Shape
Circle	5.47	84.16
Circle CS	5.42	84.11
Circle DCT	5.43	84.11
Circle DPCM	5.42	84.10
Circle DLPC	5.41	84.09
Ellipse	62.19	17.27
Ellipse CS	63.74	15.71
Ellipse DCT	62.26	17.19
Ellipse DPCM	62.39	17.05
Ellipse DLPC	63.28	16.15
Rectangle	71.48	9.81
Rectangle CS	73.23	9.32
Rectangle DCT	71.48	9.81
Rectangle DPCM	71.55	9.77
Rectangle DLPC	73.00	9.30
Non-convex Star Shaped	6.47	72.52
Non-convex Star Shaped CS	2.10	77.40
Non-convex Star Shaped DCT	6.41	72.57
Non-convex Star Shaped DPCM	6.64	72.35
Non-convex Star Shaped DLPC	6.35	72.63
Square	2.64	80.44
Square CS	2.62	80.39
Square DCT	2.63	80.33
Square DPCM	2.62	80.33
Square DLPC	2.64	80.22
Triangle	30.73	51.04
Triangle CS	31.87	50.06
Triangle DCT	30.85	50.94
Triangle DPCM	30.89	50.90
Triangle DLPC	31.43	50.46



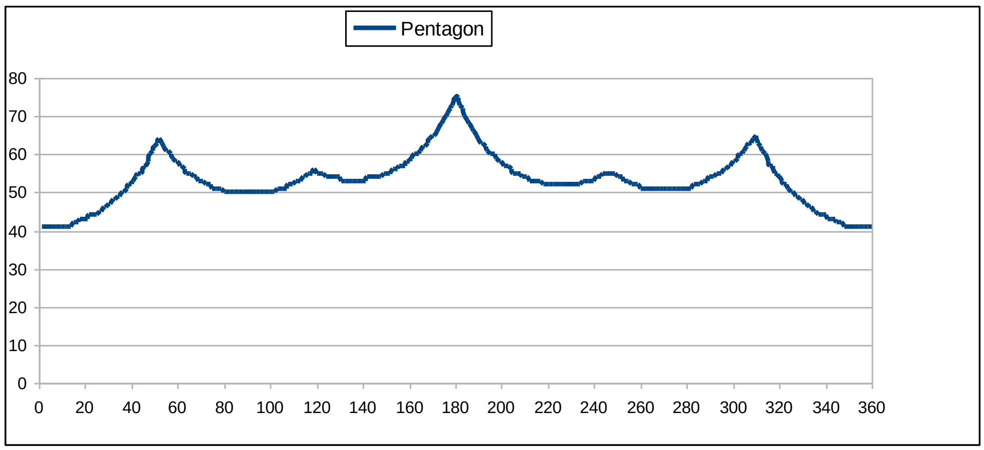


Figure 19 Square and Pentagon Signatures

Experiment 2

The second experiment compares the quality results of the compression methods while trying to reconstruct the shape images. The steps executed in this process are described in a previous section (Process 3: Reconstructing a PGM Image from a Signature File). Table 8 shows a summary of the compression ratio of all methods used in this thesis. As the table demonstrates DPCM yields the best compression ratio. The lossless compression method does not yield significant compression and is omitted from the table.

Table 8 Calculated Compression Ratio

Compression Method	Compression Ratio
DPCM	5
DCT	4
DLPC	3
CS	4

This experiment has six different shapes: rectangle, non-convex star shape, ellipse, triangle, circle, and square. The four types of lossy compression methods are applied to each image. Figure 20 and Table 9 show the results for this experiment. A higher SNR value means that the image is reconstructed with little quality loss. DPCM using 3 bits is the method with best reconstruction results.

The figure and the table demonstrate that the DPCM compression method provides the best SNR. Hence, based on the compression ratio and the SNR, DPCM has the best rate distortion of all the lossy compression methods investigated in this research.

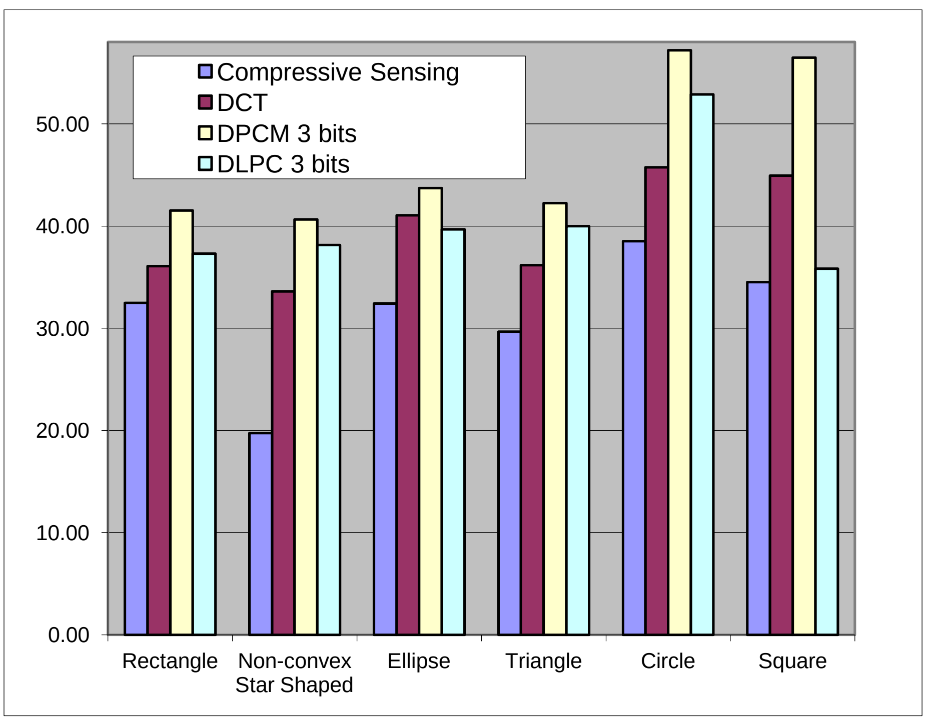


Figure 20 SNR Values of Images Reconstructed from Compressed Signatures

Table 9 SNR Values of Images Reconstructed from Compressed Signatures

Compression Method	Rectangle	Non-convex Star Shaped	Ellipse	Triangle	Circle	Square
Compressive Sensing	32.47	19.73	32.41	29.67	38.53	34.51
DCT	36.06	33.59	41.06	36.17	45.73	44.94
DPCM 3 bits	41.52	40.66	43.72	42.23	57.20	56.47
DLPC 3 bits	37.30	38.13	39.66	39.98	52.87	35.83

Experiment 3

The next experiment is similar to the previous one as it uses the same image reconstruction algorithm in order to generate a PGM file from a signature text file. However, Experiment 3 compares the SNR results using different bit rates in the compression method.

The same six shapes are used in the process, but only three compression techniques are applied in the signature images: DCT, DPCM, and DLPC. Each of these compression methods has two variations of bit rates: one using a low number of bits to store the data and the other using double or more bits than the previous one. The DCT method varies in the number of coefficients stored in the compressing file. One method stores only 32 coefficients and the other stores 90 coefficients. DPCM and DLPC have 3 and 4 bits (8 and 16 levels) used for the quantizer process.

Figure 21 and Table 10 show the results for this experiment. All three compression methods have better results when more bits are used. DPCM with 16 levels is the method with the best results for all images.

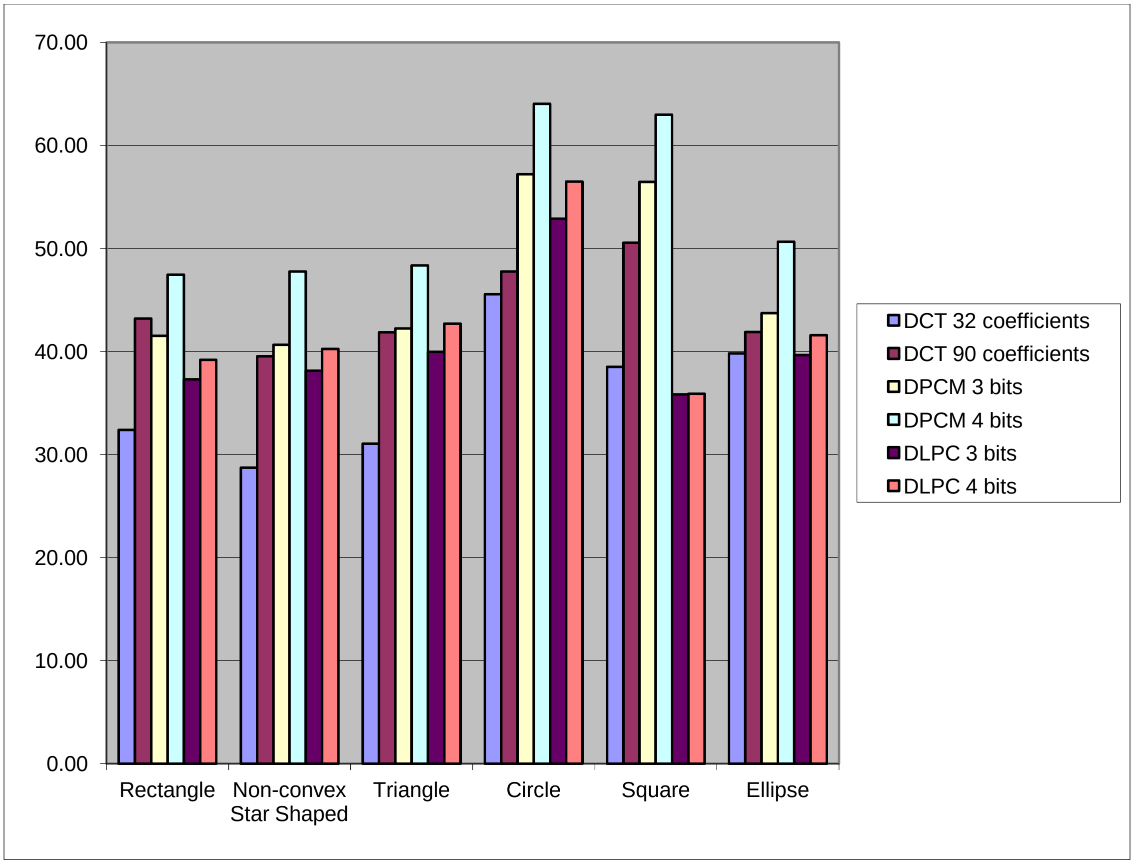


Figure 21 SNR with Different Bit Allocations

Table 10 SNR with Different Bit Allocations

Compression Method	Rectangle	Non-convex Star Shaped	Triangle	Circle	Square	Ellipse
DCT 32 coefficients	32.40	28.73	31.07	45.55	38.50	39.81
DCT 90 coefficients	43.21	39.52	41.85	47.77	50.54	41.90
DPCM 3 bits	41.52	40.66	42.23	57.20	56.47	43.72
DPCM 4 bits	47.44	47.75	48.35	64.03	62.98	50.63
DLPC 3 bits	37.30	38.13	39.98	52.87	35.83	39.66
DLPC 4 bits	39.19	40.24	42.71	56.47	35.91	41.58

Experiment 4

The last experiment in this project is related to the distortion occurred during the image reconstruction process. The process is described in the Experiment Setup section (under Process 3: Reconstructing a PGM Image from a Signature File).

The distortion values for all signature images in the database are calculated by comparing each pixel value from the reconstructed image to the correspondent pixel of an image reconstructed from a signature without compression. Figure 22 and Table 11 display the sum of absolute differences (SAD) between the pixel comparisons. Table 11 shows values close to zero for DPCM with 3 bits for the circle and square.

Figure 22 demonstrates the distortion value for the signature images after the reconstruction process. Compressive Sensing is the method with highest distortion.

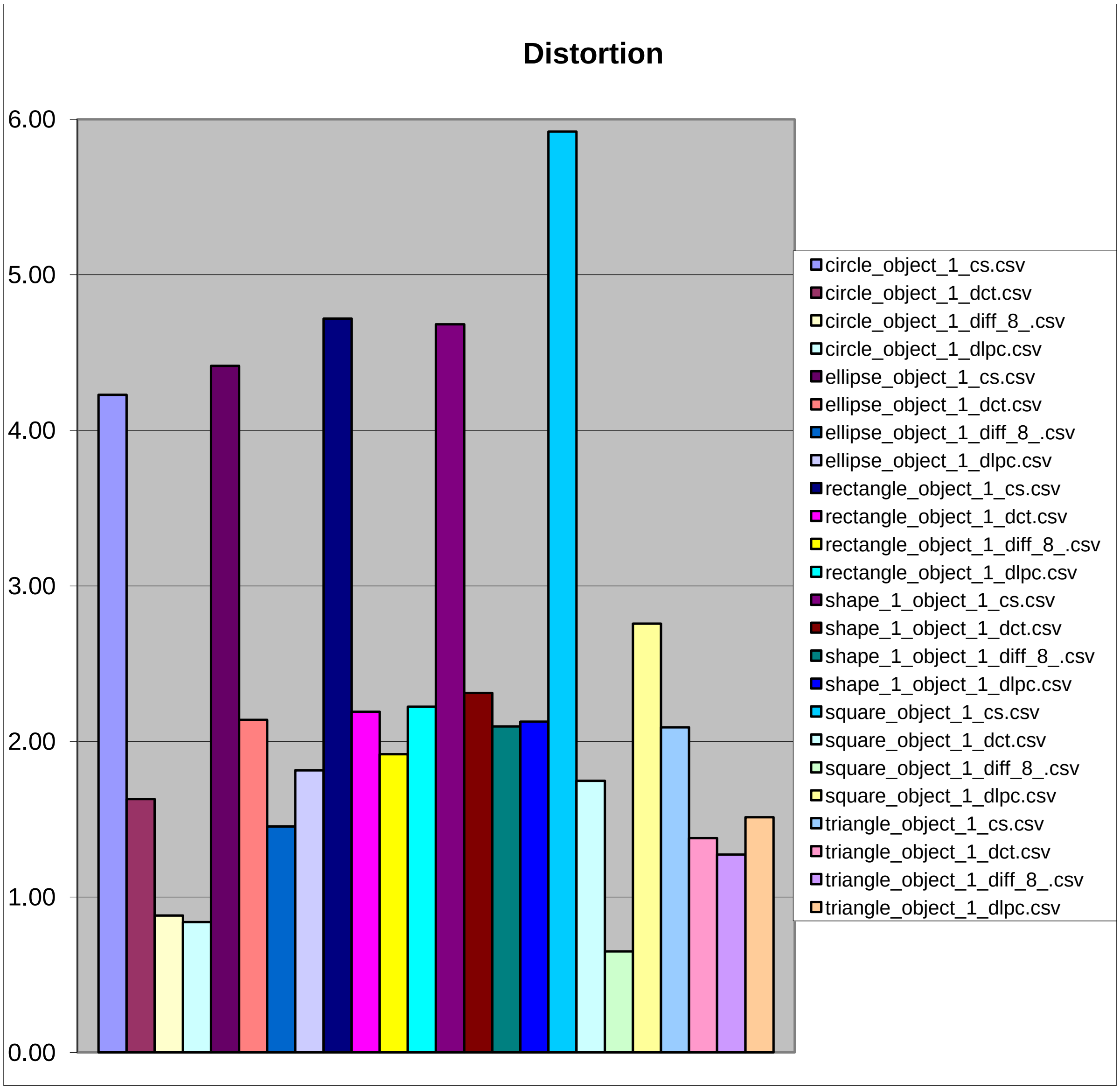


Figure 22 Distortion after Image Reconstruction

Table 11 Distortion after Image Reconstruction

Signature File	Distortion
circle_object_1_compressive_sensing.csv	4.23
circle_object_1_dct.csv	1.63
circle_object_1_diff_8_.csv	0.88
circle_object_1_dlpc.csv	0.84
ellipse_object_1_compressive_sensing.csv	4.41
ellipse_object_1_dct.csv	2.14
ellipse_object_1_diff_8_.csv	1.45
ellipse_object_1_dlpc.csv	1.81
rectangle_object_1_compressive_sensing.csv	4.72
rectangle_object_1_dct.csv	2.19
rectangle_object_1_diff_8_.csv	1.92
rectangle_object_1_dlpc.csv	2.22
shape_1_object_1_compressive_sensing.csv	4.68
shape_1_object_1_dct.csv	2.31
shape_1_object_1_diff_8_.csv	2.10
shape_1_object_1_dlpc.csv	2.13
square_object_1_compressive_sensing.csv	5.92
square_object_1_dct.csv	1.75
square_object_1_diff_8_.csv	0.65
square_object_1_dlpc.csv	2.76
triangle_object_1_compressive_sensing.csv	2.09
triangle_object_1_dct.csv	1.38
triangle_object_1_diff_8_.csv	1.27
triangle_object_1_dlpc.csv	1.51

Summary of Results

The object signature file has 360 entries corresponding to the collected distances from the center to the object border with a difference of one degree separating each value. These entries are normalized by the maximum distance and then quantized to an integer number using 16 bits for storage. The file also stores the maximum distance among all values with the objective of restoring it later. The maximum distance value is stored using a 32-bit floating point representation. This results in a total of 5792 bits per file without using a compression method.

Experiment 1 demonstrates that all four compression methods are able to find a correct match signature for the input image. The Compressive Sensing method needs a higher threshold value to recognize some images, but it is still able to recognize them.

Simple signatures such as circle and square have good recognition results for all different input variations and compression methods. The triangle is an exception because some inputs with rotation variation have results with higher values, which means they almost do not match with the original signature stored in the database.

Experiment 2 shows through the SNR that DPCM is the method with the highest values for all six input objects. Higher value means better reconstruction. Experiment 3 confirms that DPCM has the best quality results with 3 bits quantizer. Once the number of bits increases from 3 to 4 there is an increase in the SNR numbers for all images. Other methods also have an improvement when the compression ratio decreases.

Experiment 4 shows through the distortion that Compressive Sensing has a lower reconstruction quality when the images are compared pixel by pixel. This is also noticeable by visual inspection of the reconstructed images. The following figures graphically demonstrate the reconstruction of a non-convex star shaped image from a signature file using different compression methods. Figure 23 shows the signature without compression and the expected PGM image generated by that. The reconstructed image has almost the same border as the original non-convex star shaped image depicted in Figure 6 (chapter IV). It is important to notice that the contour is recreated using only 360 pixels from the original object edge and this justifies the small border imperfections in Figure 23.

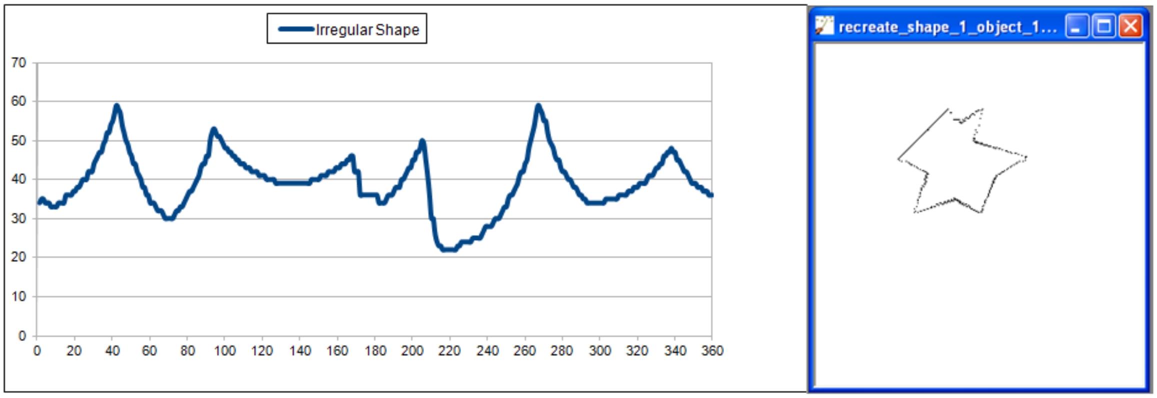


Figure 23 Signature and Image Representation of Non-convex Star Shaped

Subsequent figures reveal different noise quantity in the signature and recreated image in comparison to Figure 23. Figures 24, 25, and 26 have a PGM recreated image similar to the PGM image with no compression method applied in Figure 23. It is possible to visualize a small noise present in the signature graph located at the left side of the PGM image in Figures 24, 25, and 26.

Figure 27 uses Compressive Sensing and it is possible to notice a higher quantity of noise present in the graph. The PGM image is different from images reconstructed using other compression methods.

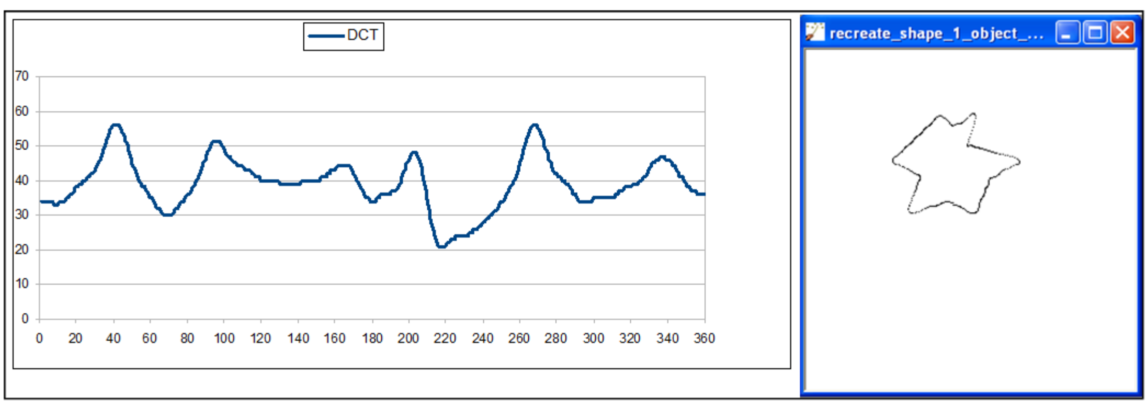


Figure 24 Signature and Image Representation of Non-convex Star Shaped with DCT

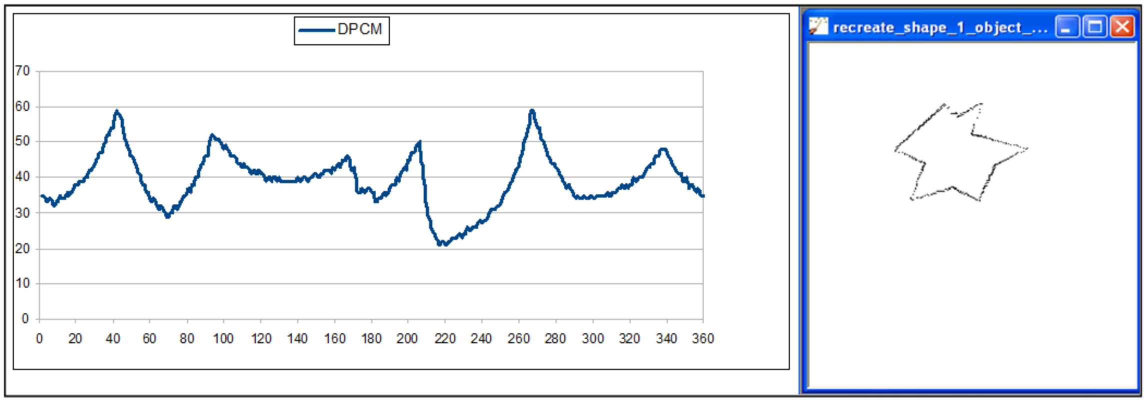


Figure 25 Signature and Image Representation of Non-convex Star Shaped with DPCM

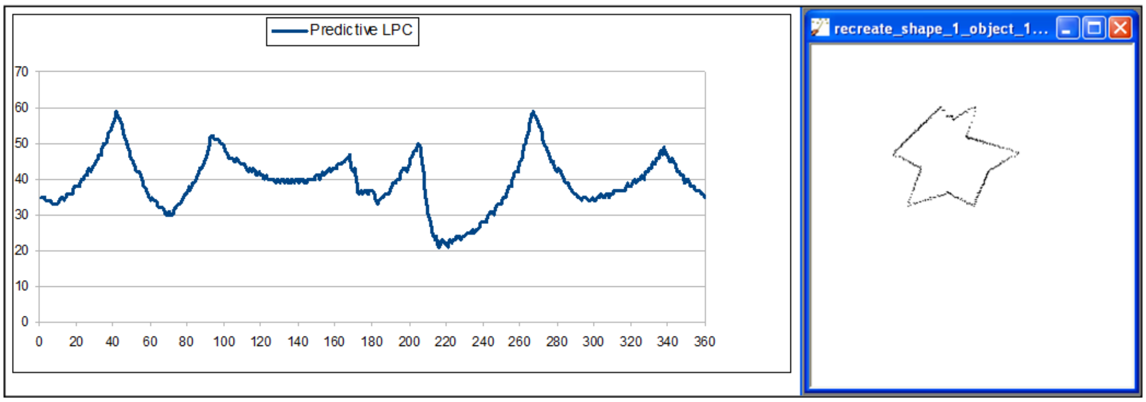


Figure 26 Signature and Image Representation of Non-convex Star Shaped with DLPC

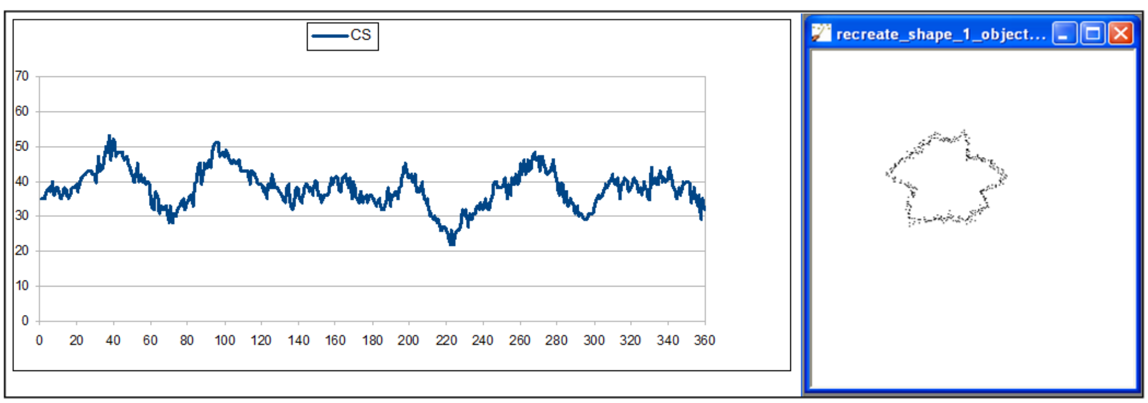


Figure 27 Signature and Image Representation of Non-convex Star Shaped with CS

The results show that the recognition process is able to identify all input objects for every image and compression method combination. The experiments demonstrate that lossless compression is not a viable method for this application. Furthermore, according to the experimental results DPCM has the best rate distortion performance. On the other hand, the Compressed Sensing method produces higher distortion and requires a higher threshold value in order to accept a valid recognition response. Although the quality results for this method are not as good as the DPCM method, Compressed Sensing has an important advantage of requiring less data to represent an image during the acquisition process. This means that CS might enable to capture the data with a smaller number of sensors.

CHAPTER VI

CONCLUSION AND FURTHER RESEARCH

This thesis compares four types of lossy compression methods in the recognition and reconstruction context. The results show that the recognition process is able to identify all input objects for every image and compression method combination. The experiments demonstrate that lossless compression is not a viable method for this application. Furthermore, according to the experimental results DPCM has the best rate distortion performance. On the other hand, the Compressed Sensing method produces higher distortion and requires a higher threshold value in order to accept a valid recognition response. Although the quality results for this method are not as good as the DPCM method, Compressed Sensing has an important advantage of requiring less data to represent an image during the acquisition process. This means that CS might enable to capture the data with a smaller number of sensors.

The analysis calls for further research in the field. For example, the signature extraction and recognition process for natural images can be challenging. The topic requires an enhanced segmentation method in order to compare 3D objects since any small rotation generates completely different signatures.

An interesting topic for future research is the study of the computational complexity of the four compression methods applied in this thesis. Memory requirement for encoding and decoding the object signature is another important aspect to be further evaluated.

Another possible research direction is the study of new types of compression methods in order to combine those with the four methods analyzed in this thesis and evaluate their efficiency using measurements such as compression ratio, quality reconstruction by SNR, speed for encoding and decoding, and memory utilization.

WORKS CITED

Baggs, R. A., & Tamir, D. E. (2008). Non-rigid Image Registration. *Proceedings of the Florida Artificial Intelligence Research Symposium*. Coconut Grove, FL.

Baggs, R. A., & Tamir, D. E. (2008). Image Registration Using Dynamic Space Warping. *Artificial Intelligence and Pattern Recognition*, 128-135. Orlando, FL.

Candès, E. J., & Wakin, M. B. (March 2008). An Introduction To Compressive Sampling. *Signal Processing Magazine, IEEE, 25* (2), 21-30.

Chang, F., Chen, C.-J., & Lu, C.-J. (February 2004). A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique. *Journal of Computer Vision and Image Understanding*, *93* (2), 206-220.

Cisco. (2011). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010–2015.* San Jose: Cisco.

Das, M., & Chande, S. (2001). Efficient lossless image compression using a simple adaptive DPCM model. *Proceedings of the 44th IEEE Midwest Symposium on Circuits and Systems, 1*, 164-167. Dayton, OH.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing.* Upper Saddle River: Prentice Hall.

He, L., Chao, Y., & Suzuki, K. (June 2010). A Run-Based One-And-A-Half-Scan Connected-Component Labeling Algorithm. *International Journal of Pattern Recognition & Artificial Intelligence, 24* (4), 557-579.

Ito, Y., & Nakano, K. (June 2010). Low-Latency Connected Component Labeling Using An FPGA. *International Journal of Foundations of Computer Science*, *21* (3), 405-425.

MacKay, D. (2003). An Example Inference Task: Clustering. In D. MacKay, *Information Theory, Inference, and Learning Algorithms*, 284-292. Cambridge: Cambridge University Press.

Nikolaou, N. N., & Papamarkos, N. N. (February 2002). Color image retrieval using a fractal signature extraction technique. *Engineering Applications of Artificial Intelligence, 15* (1), 81-96.

Pavlidis, T. (1995). *Algorithms for Graphics and Image Processing.* New York: W H Freeman & Co.

Richardson, I. E. (2004). *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia.* Chichester, UK: John Wiley & Sons, Ltd.

Sayood, K. (2006). *Introduction to Data Compression.* San Francisco: Elsevier Inc.

Shapiro, L. G., & Stockman, G. C. (2002). *Computer Vision.* Upper Saddle River: Prentice Hall.

Tamir, D. E., Shaked, N. T., Geerts, W. J., & Dolev, S. (2010). Compressive Sensing of Object-Signature. *Proceedings of the 3nd International Workshop on Optical Super Computing*, *6748*, 63-77. Bertinoro, Italy.

VITA

Edson Ricardo Amboni was born in Florianópolis, Santa Catarina, Brazil on June 26, 1976, the son of Edson Amboni and Maria Aparecida dos Santos Amboni. After graduating high school at Colégio Geração in 1993, he entered Universidade Federal de Santa Catarina (UFSC) in Florianópolis, Santa Catarina. He received the degree of Bachelor of Science in Dentistry from UFSC in October of 1998. During the following years he worked as a Dentist at the Public School Joaquim Santiago in São José, Santa Catarina. In 2000 he attended UFSC again for the degree of Bachelor of Science in Information Systems. He moved to the United States of America in February of 2003. He has been employed as a Software Engineer at SalePoint, Inc., a software company based in San Diego, California since 2003. He entered the Graduate College of Texas State University-San Marcos in August of 2008.

Permanent Email Address: amboni@ig.com.br

This thesis was typed by Edson Ricardo Amboni.