INVESTIGATION OF THE EFFECTS OF IMAGE COMPRESSION ON THE GEOMETRIC QUALITY OF DIGITAL PHOTOGRAMMETRIC IMAGERY

MSc Thesis

By

KWABENA-FORKUO,ERIC

January 1997

Thesis submitted in partial fulfilment of the requirements for the Degree of Master of Science in Engineering

Department of Surveying & Geodetic Engineering
University of Cape Town

University of Cape Town

The University of Cape Town has been given the right to reproduce this thesis in whole or in part. Copyright is held by the author.

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

University of Cape Town

DECLARATION

I, Kwabena-Forkuo Eric, submit this thesis in partial fulfilment of the requirements for the degree of master of science in engineering. I hereby declare that this thesis is my original work and has not been submitted in any form to another university.

Acknowledgements

I would like to express my appreciation and indebtedness to my supervisor, Dr. Scott Mason for his valuable suggestions towards the successful accomplishment of this research work. I am grateful to Prof. Ruther, Head of Department, Surveying and Geodetic Engineering, for providing the research topic, and for his guidance.

A particular note of thanks is given to Nsafoa Boateng Joseph, Yeboah Druyer Collins and Gloria Ntow for their love and their financial support. I would also like to acknowledge the suggestions, comments and technical assistance received from Mr M. Barry (GIS Course Convenor), Mr D. Wilson and Mrs S. Binedell, Julian Smith, Malcolm Dingle, Ulrike Bruessler, Henty Walker, Sidney Smith and Michael Haywood.

Mufalo Mbinji, Li Jun and Mtshatsha Bandile not only encouraged me but helped me by reading the draft of the manuscript, and offering very helpful critical evaluations. To them I am particularly grateful. Many thanks are also extended to all friends and members of staff in the Surveying and Geodetic Engineering Department, University of Cape Town, for their valuable suggestions and comments on the draft of the manuscript.

Last but not least, I want to thank the following colleagues and friends, for their assistance and co-operation during my stay in Cape Town: Mr D. Padkin, Mufaro Chivasa, Danso Antwi Adjei, Osei Samuel, Yirenkyi Samuel, Edem Ankude, Addae-Manu R, Sophie Wamono, Mrs A Danso, and Rockson Ansah. Since the list is too long to mention, it is hoped that those who have given me help in this manner will accept this anonymous recognition.

Abstract

We are living in a decade, where the use of digital images is becoming increasingly important. Photographs are now converted into digital form, and direct acquisition of digital images is becoming increasing important as sensors and associated electronics improve. Unlike images in analogue form, digital representation of images allows visual information to be easily manipulated in useful ways. One practical problem of the digital image representation is that, it requires a very large number of bits and hence one encounters a fairly large volume of data in a digital production environment if they are stored uncompressed on the disk.

With the rapid advances in sensor technology and digital electronics, the number of bits grow larger in softcopy photogrammetry, remote sensing and multimedia GIS. As a result, it is desirable to find efficient representation for digital images in order to reduce the memory required for storage, improve the data access rate from storage devices, and reduce the time required for transfer across communication channels. The component of digital image processing that deals with this problem is called image compression. Image compression is a necessity for the utilisation of large digital images in softcopy photogrammetry, remote sensing, and multimedia GIS.

Numerous image compression standards exist today with the common goal of reducing the number of bits needed to store images, and to facilitate the interchange of compressed image data between various devices and applications. JPEG image compression standard is one alternative for carrying out the image compression task. This standard was formed under the auspices of ISO and CCITT for the purpose of developing an international standard for the compression and decompression of continuous-tone, still-frame, monochrome and color images. The JPEG standard algorithm falls into three general categories: the baseline sequential process that provides a simple and efficient algorithm for most image coding applications, the extended DCT-based process that allows the baseline system to satisfy a broader range of applications, and an independent lossless process for application demanding that type of compression.

This thesis experimentally investigates the geometric degradations resulting from lossy JPEG compression on photogrammetric imagery at various levels of quality factors. The effects and the suitability of JPEG lossy image compression on industrial photogrammetric imagery are investigated. Examples are drawn from the extraction of targets in close-range photogrammetric imagery. In the experiments, the JPEG was used to compress and decompress a set of test images. The algorithm has been tested on digital images containing various levels of entropy (a measure of information content of an image) with different image capture capabilities. Residual data was obtained by taking the pixel by pixel difference between the original data and the reconstructed data. The image quality measure, root mean square (rms) error of the residual was used as a quality measure to judge the quality of images produced by JPEG(DCT-based) image compression technique.

Two techniques, TIFF (LZW) compression and JPEG(DCT-based) compression are compared with respect to compression ratios achieved. JPEG(DCT-based) yields better compression ratios and it seems to be a good choice for image compression. Further in the investigation, it is found out that, for gray-scale images, the best compression ratios were obtained when the quality factors between 60 and 90 were used (i.e., at a compression ratios of 1:10 to 1:20). At these quality factors the reconstructed data has virtually no degradation in the visual and geometric quality for the application at hand.

Recently, many fast and efficient image file formats have also been developed to store, organise and display images in an efficient way. Almost every image file format incorporates some kind of compression method to manage data within common place networks and storage devices. The current major file formats used in softcopy photogrammetry, remote sensing and multimedia GIS were also investigated. It was also found out that the choice of a particular image file format for a given application generally involves several interdependent considerations including quality; flexibility; computation; storage, or transmission. The suitability of a file format for a given purpose is best determined by knowing its original purpose. Some of these are widely used (e.g. TIFF, JPEG) and serve as exchange formats. Others are adapted to the needs of particular applications or particular operating systems.

Table of Contents

List of Illustrations

List of Tables

Glossary and List of Abbreviations

Animation: a sequence of two or more images displayed in a rapid sequence so as to provide the illusion of continuous motion. Animations are typically played back at a rate of 12 to 15 frames per second.

Array of pixels: An ordered set of colored display elements on an output device. This term is used loosely to refer to an array of numerical values used by an application program to specify colored elements on an output device.

Bandwidth: The maximum frequency a monitor can accept without degrading the video signal.

Bilevel: Any image that contains one background color and one foreground color (e.g., black and white).

Binary: A numerical system using base two.

Bit-endian: Refers to systems or machines that store the most-significant byte (MSB) at the lowest address in a word, usually referred to as byte 0.

Bit depth: The size of a value used to represent a pixel in bitmap graphics data. This is usually stated as the number of bits compressing the individual data value, or sometimes the number of bytes. The number 2 raised to the power *bit depth* specifies the maximum number of values the pixel can assume. Same as *pixel depth*.

Bitmap: A set of numerical values specifying the colors of pixel on an output device. In older usage, the term referred to data intended for display on output device capable of displaying only two levels. It is used in this thesis as a synonym for *raster*

Bitmap data: The portion of a bitmap file containing information associated with the actual image.

Bitmap image: A representation of a graphics work on a raster device or in a bitmap file.

Bit order: The order of this bits within a byte. The first bit in a byte may be either the most-significant or lest-significant bit .

Bit plane: A two-dimensional array of bits one bit deep. A bitmap containing pixels with a depth of eight bits may be said to contain eight planes. A monochrome image (one bit per pixel) is usually stored as a single bit plane.

Byte: A unit of measurement used to rate storage capacity of display and system memories and disks. A byte is composed of 8 bits of information processed as a unit.

Calibration: conversion of data values from arbitrary engineering units (such as pixel values) to meaningful geophysical quantities such as temperature.

CAD (Computer-aided design): The use of applications, usually vector-based, for the design and rendering of graphical data of architectural and mechanical drawings, electronic schematics, and three-dimensional models.

CCD(Charge Couple Device) Camera:. A solid-state camera that uses a CCD to transform a light image into a digitized image.

CCITT: Comte Consultatif International Telephonique et Telegraphique.

CD-ROM (Compact Disk Read Only): A format standard for placing any kind of digital data. on a compact disc.

Chrominance: The color portion of an image. It is the mixture of hue and saturation, or the combination of three primary colors, such as red, green, and blue.

CMY: Acronym for Cyan, Magenta, Yellow. A subtractive color sysrtem based on the primary colors cyan, magenta, and yellow.

Color space: A set of parameters that describes color values such as RGB or CMYK.

Data compression: Various techniques that reduce the data content and storage needed to represent an image. Compression can be lossy or lossless. It also reduces the time it takes to display and manipulate files and images.

Data encoding: A generic term for the process of converting data from one format to another. Data compression is a form of data encoding

Decompression: The restoration of a compressed image or data file.

Digital image: An image composed of discrete pixels of digitally quantized brightness.

FGDC: Federal Geographical Data Committee

File: It is a collection of records treated as a units. Files can contain programs or data (or both intermixed).

File identifier value: A specific value, or set of values, used to positively identify a file as being of a particular file format. File ID values may be an integer or a string of ASCII characters. They usually appear in the first field of a file header.

Format: It is the structure of each record in the file

Frame: The total amount of instantaneous information (as perceived by the viewer) presented by a display.

Grey scale: A term used when referring to an image. A gray shade is any color whose three primary colors are the same value. Grey shades only have intensity(luminance) and no color (chrominance).

Grey level: The brightness value assigned to a pixel. A value can range from black, through the grays, to white

GIS (Geographic Information Systems): Tools for collecting, storing, manipulating and retrieving data from the real world for a particular purpose.

HIS: Acronym for Hue, Intensity, Saturation. An additive color system based on the attributes of color(hue), percentage of white(saturation), and brightness(intensity).

Histogram: The graphical representation of grey scale occupancy of an image. With the horizontal axis representing the grey level and the vertical axis representing number of pixels, the histogram presents an-easy-to read indication of image contrast and brightness dynamic range

Interframe compression: The creation of encoded data from two of more image frames. MPEG encoding is an interframe encoding method.

Intraframe compression: The creation of encoded data from a single image frame. JPEG encoding is an intraframe encoding method.

Joint Photographic Expert Group (JPEG): An international standard for compression and decompression of continuous-tone, still-frame, monochrome and color images.

Lossless encoding: A data compression or encoding algorithm that does not lose or discard any input data during the encoding process.

Lossy encoding: A data compression or encoding algorithm that loses, or purposely throws away, input data during the encoding process to gain a better compression ratio. JPEG is an example of a lossy encoding method.

Luminance: The brihtness or intensity of a color. The pixels in a monochrome image have a luminance of either 100 percent or 0 percent.

LWZ : Lempel Ziv Welch hybrid coding system

Metadata: It is comprised of attributes, parameters, notebooks, and other types of miscellaneous complex data aggregates associated with primary scientific data.

Metafile: A file format capable of storing two or more types of image data, usually vector and bitmap, in the same file.

MIDI: Musical Instrument Digital Interface. A standard for digital signals used to control electronic musical instruments. MIDI information may be stored as a data file and is found in many multimedia file formats.

Motion Picture Experts Group (MPEG): An international standard for real-time compression and decompression of video or motion images.

Multimedia: The concept of creating, storing, and playing back two or more forms of electronic information simultaneously. such information includes still- images, motion-video, animations, digitized sound, and control information such as midi codes.

Operating system: Software procedures that tell the computer how to operate programs. MS-DOS and UNIX are two example of operating systems.

Palette: A collection of colors from which a single color or multiple colors can be chosen. The palette is the maximum number of colors or shades possible by all combination of brightness level of the three primary color(RGB) output. The palette size is found by taking the base two value of the total number of bits of the outputs.

Pixel: In traditional usage , short for "picture element." These are irreducible elements of color created by an output device on its display surface. The term is sometimes used loosely to refer to values of bitmap data elements used by an application to order the display of color elements on an output device.

Pixel values: Numerical data items in an image file indicating the color or other information associated with an individual pixel.

Planar files: Image files with image data stored as bit planes or color planes rather than as pixels.

Predictive encoding: An algorithm that has certain prior knowledge about the format of the data it is encoding. Huffman is a predictive encoding algorithm.

Quick-Time movie: Video footage that can be assembled on a Macintosh, stored on a disk drive, incorporated in any Macintosh application, and played back without any external video equipment.

Raster: An image represented by color values at points, which taken together describe the display on an output device.

RGB: Acronym for Red, Green, Blue. An additive color system based on the primary colors red, green, and blue. The RGB model is loosely patterned after human eyes, which have a peak sensitive to the colors red, green, and bue light.

Remote Sensing: The use of electromagnetic radiation sensors to record images of the environment which can be interpreted to yield useful information: i.e obtaining information about the physical features without any physical contacts with the features.

Resolution: The number of pixels, represented by bits in the display memory. A display memory can be organised by pixels in the x axis(pixel per line) by the number of pixels in the y axis(lines), and by the number of memory planes in the z axis. The higher the resolution, the greater the detail and the more information that can be stored.

Run-Length Coding(RLE): A data compression technique that records repeated data elements with the same value, which is encoded once along with a count of the number of times it occurs.

Sampling: The process of reading an analog signal at specific increments in time (sample rate) and storing the data as digital values. Sampling is the basic process used to create digital audio and video.

Scaling: A function that adjusts an image to form within a window.

Scan line: One row of digital image, generally in a cross-trac direction.

Scene: The area on the ground that is covered by an image or photograph.

SDTS: Spatial Data Transfer Standard

Stream: Data with no fixed position in a file, composed of sub-elements with a known structure.

Strip: A collection of one or more contiguous scan lines in a bitmap. Scan lines are often grouped in strips to buffer them in memory more efficiently. It is also called bands in some file format specifications

Tag : A data structure in a file which can vary in both size and position.

Tag Image File Format(TIFF): A standard image file format developed by Aldus for the storage of high-resolution scanned images.

USGS: United States Geographical Survey

Vector: Refers to image data composed mainly of representation of lines and outlines of objects, which can be compactly represented by specifying sets of key points.

YUV: Acronym for Y-signal, U-signal, and V-signal, which is based on early color television terminology. A luminance -chrominance base color model (Y specifies gray-scale or luminance, U and V chrominace) are used by many video compression algorithms, such as MPEG.

Chapter 1

Introduction

1.1 Introduction

Since the advent of digital photogrammetric workstations, satellite imagery and raster- to- vector conversions of digitised maps, Digital Image Processing has become increasingly important in Softcopy Photogrammetry, Remote Sensing and Geographical Information Systems (GIS). Apart from these applications, digital images are essential to many professions and are commonly used in film and video production, meteorological, medicine, and publishing. However, the later areas have different requirements and tend to have their own preferred image representation and operations.

Unlike images in analogue form, digital images can be compressed, enhanced, analysed, and modified on a computer. It has been also possible to store and transmit digital images across network lines. As the resolution of scanners and digital sensors improves in spatially and in radiometrically , the resulting data volume of the image increases and the consequent demand for image storage and transmission increases. Due to the large amounts of data used to represent an image, the processing of digital images is often very difficult. That is the amount of data storage media needed for archiving of such images is enormous. However, the advances in digital networks for mass storage and digital image processing have paved the way for improved image data compression technologies and have improved the efficiency of storage and transmission of images (Jain, 1981).

The contents of this thesis is structured into seven chapters. Chapter 1 is an introduction to the study area. Chapter 2 presents background information on the fundamental concepts of image file formats relevant in softcopy photogrammetry and remote sensing. This chapter also gives a brief description of how digital images are stored. Chapter 3 addresses the fundamentals of image compression techniques.

Lossless and lossy compression techniques, as well as the algorithms and transformation involved are discussed in this chapter. Chapter 4 deals with various stages of JPEG (DCT-based) image compression for still images. Chapter 5 is concerned with the practical aspects of JPEG (DCT-based) image compression. It describes the design of the experiments and the tests. In chapter 6, the results given in the previous chapter are interpreted and inferences are drawn. The accuracy achieved by the target measuring algorithm and accuracy due to JPEG compression are also discussed. Chapter 7 outlines conclusions and recommendations of this study. In Appendix A, graphs of quality compression ratios versus distortions for the test images are presented.

1.2 Area of Study

This thesis generates experimental results of the effects of image compression techniques on photogrammetric imagery. The main focus is to review the compression techniques and to investigate the quantitative effects of the JPEG (DCT- based) compression techniques on the geometric quality of digital images. The primary motivation in selecting JPEG was to maintain the theme of standards compliance. While other compression algorithms e.g., wavelet and fractal, allow greater levels of compression, there are no agreed standards to allow for interoperability among users (Hoffman, 1995). Even though JPEG does not absolutely assure this condition, it is more probable that a user at a location different from where the image data was compressed will be able to expand it for display and for subsequent use. JPEG has already been widely embraced as the computing industry's standard encoding method for high-quality grey scale and color continuos-tone images.

JPEG (DCT- based) is primarily a lossy (with loss of information) method of compression. That is, it was designed specifically to discard information that the human eye cannot easily see. It offers a gain in the file size against the quality of the reconstructed image. Therefore the "art" of JPEG image compression is finding the lowest quality factor that produces an image that is visually accepted and preferably as identical to the original as possible.

There are two main concerns to phogrammetrists related to the use of JPEG(DCT-based) image compression. The first is the degree of variation in grey scale values which can result in apparent shift of features in the digital image accompanying the removal of spatial frequency data and the second is the possibility of introducing artefacts (or false edge effects) as a consequence of the 8 x 8 pixel blocking strategy employed (Fraser & Edmundson, 1996). The research therefore aims at investigating the quantitative effects of JPEG compression on the geometric quality of digital images. It focuses on industrial applications of photogrammetry in which the key object features to be measured are signalised with circular targets.

Recently, many image file formats have been developed to store, organise and display images. Almost every image or graphic file format incorporates some kind of compression method (Murray & VanRyper, 1994) and hence the choice of image file format is also of relevance. In this thesis, the current major file formats used in softcopy photogrammetry, remote sensing and multimedia GIS are also described.

1.3 Background to the Study

The advancements in softcopy photogrammetry and the development of digital photogrammetric and satellite workstations have led to a tremendous demand for digital aerial and satellite imagery. These images are directly acquired by using solid state cameras, and satellite sensors or indirectly by scanning or digitising existing photographs (Novak and Fayez, 1996). However, digital representation of these images requires a very large number of bits and hence one encounters a large volume of data in a digital production environment if they are stored uncompressed on the disk. That is digital images "consume" large amount of bytes. For instance, in photogrammetric application, if a standard pixel size of 15μm is used, a digitised 23cm x 23cm aerial photograph has a data volume of 250 MB as an 8-bit black and white image, and of 750 MB as a color image respectively (Krzysteck, 1995). Similarly, in remote sensing applications, a single satellite (SPOT P mode) scenes contains 6 000 by 6 000 pixels or 36 MB.

Considering the number of images that are produced by these applications, it is desirable to represent the information in the image with considerable fewer number of bits per pixel or with a reduction in the volume of image data. Thus, there is a consequent need to limit the resulting data volume associated with the applications and one possible way to decrease the necessary amount of storage, is to work with compressed image data (Sonka, Hlavac & Boyle, 1993).

Working with compressed digital images means that the amount of data used to represent the image has been controlled by making adjustments to the components of digital image quality (i.e., resolution, size, color depth, and the amount of compression). These factors can be adjusted to varying degrees, and combinations, in order to find the most appropriate compromise between image storage and image quality. The techniques for reducing the memory space requirements of bitmapped image data are (Jaakkola & Orava , 1994) :

- larger pixel size;
- less bits per pixel;
- digressive sampling;
- image patches; and;
- image compression.

Because optimising the number of grey values does not produce enough reduction, and digressive sampling and the usage of image patches lead to complicated systems, the reasonable alternatives to reduce the bitmapped data volume are to use larger pixels and image compression (Jaakkola & Orava, 1994). Image data compression technique is the science of removing redundant information from image data to minimise the number of bits that need to be stored on a disk transfer through a network. That is, image compression is one of the methods which can considerably contributes to smaller file sizes and minimise transfer rates and the required disk space. The basic idea of image compression is to remove redundancy present in the image data in a way which makes image reconstruction possible.

Image compression can be classified into two basic methods: Lossy (information preserving) and lossless (with loss of information) compression methods. A compression method is lossless when it reduces the number of bits required to represent an image such that the reconstructed image is numerically identical to the original image on a pixel- by - pixel basis. On the other hand, lossy compression schemes allow degradation in the reconstructed image in exchange for a reduced bit rate (Novak & Shahin, 1996).

Improved quality can be always be achieved at the expense of higher bit rates, but good compression techniques must provide a good trade-off between image quality and bit rate (Oehler & Gray, 1995). In order to reduce image size, many image file formats compress the pixel data before storage. Several compression techniques such as Run-Length Encoding, LZW coding, predictive coding, sub-band coding, vector quantization, block-truncation coding, wavelets and transform-based coding, are currently applied in softcopy photogrammetry. There are many compression methods which employ several of these techniques in combination. One of the most popular and widely used method is the Baseline JPEG compression algorithm. This algorithm provides a lossy image compression based on a Discrete Cosine Transformation (DCT) to convert image data into rate-of-change information and Huffman Coding to produce the compressed image. A detailed overview of the algorithms used in image compression techniques is discussed in Chapter 3.

1.4 Applications of Image Compression

The usefulness of image compression arises in applications where storage of large quantities of digital image data (i.e. scanned aerial photographs or document, satellite and medical images) have to be archived in a limited storage space or where digital images are transmitted over a narrow channel (Lammi & Sarjakoski, 1995). Early applications of data compression can be found in the fields of telecommunication, data storage and printing (Wallace, 1991).

Major users of image data compression are:

- high resolution television;
- satellite remote sensing;
- military communication via aircraft, radar and sonar;
- teleconferencing;
- computer communications; and;
- facsimile transmission.

Other applications of compressed images includes time critical data processing, in which case the speed of the image analysis algorithms is improved by applying them to the compressed data.

1.5 Objectives of image compression

In attempting to obtain an efficient and accurate representation of an original image for a reduced number of bits, a fundamental goal of image compression is to extract the information in an image which is important for a particular application (Oehler & Gray, 1995). The objectives of image compression include the following:

- Compression reduces the volume of image data. It transforms the image to a compressed form so that the information content of the image is preserved as far as possible;

- Compression techniques permit progressive transmission, producing an increasingly good reconstruction image as bits arrive. This is especially useful for telebrowsing and selection of important parts of the image (Storer, 1992); and;

- Compression conserves storage space which allows larger volume of data to be stored especially in remote sensing, photogrammetry and GIS. Conservation of storage space also allows for larger inventories in some applications such as medical imaging.

1.6 Objectives of the Study

Image processing plays a central role in remote sensing and digital photogrammetry. It is also involved in the raster-to-vector conversion of digitised maps. As the resolution of scanners (now down to 7 microns) and digital sensors improves, the size of the images increase and naturally with it the importance of image compression also increases. The choice of image compression technique and hence image format used is therefore of relevance. Therefore, the goal of this thesis is to:

1) Study and summarise the main image compression techniques currently used in image processing;

2) Examine the importance of image compression to various photogrammetric and multimedia GIS applications;

3) Identify experimentally the effects of JPEG lossy image compression techniques on photogrammetric. The emphasis is on both geometric and visual quality of images;

4) Identify the current major image formats used in photogrammetry and remote sensing, their structure and their header information; and;

5) Make recommendations on choice of image compression technique for various photogrammetric applications and requirements (e.g. speed and the amount of tolerable loss).

1.7 Research methodology and experimental procedure

This section includes a description of the test process, procedures and methodologies used. The general methodology for the practical testing was as follows:

1. Design and arrangement of the test images for the investigation. These images were real grey scale images of objects with signalised with circular targets;

2. Compression and decompression of 8-bit imagery utilising quality factor values in the range of 40 to 95 (approximate compression ratios of 2:1 to 40:1) with the independent JPEG Group's (IJP) JPEG software;

3. Automatic detection and approximate locations of targets in the image using a program which extracts the targets centres with an appropriate selection of grey level threshold value; and;

4 Statistical analysis: Quantitative measures were employed to compare original imagery with JPEG compressed and decompressed data set by:
   a) Computing variances, standard deviations, maximum and minimum values and root mean square(rms) errors of the data sets; and;
   (b) Plotting graphs of quality factors against rms error of different image data sets

1.8 Limitations of the study

- Lossy schemes cause degradations of the reconstructed image in exchange for larger compression ratios. These degradations of the reconstructed lead to radiometric (i.e images become blurred) and geometric (i.e., shifting objects in line or column or both directions) distortion. The study concentrated on the geometric distortions caused by JPEG lossy schemes.

- The computing time required to compress and decompressed images using JPEG image compression standard was not investigated, although computing time plays a major role in selecting a compression scheme as the sizes of digital images used in softcopy photogrammetry and remote sensing become very large.

- The study concentrated on the extraction of circular targets in close-range photogrammetric imagery. Geometric information from features such as edges, corner lines, and crosses was not extracted.

Chapter 2

Primary Image File Formats relevant to Digital Photogrammetry, Remote Sensing and Multimedia GIS

2.1. Introduction

An image file format is the format in which image data (data describing an image) is stored in a file. In practical sense, every major application in digital photogrammetry, remote sensing, and multimedia GIS creates an enormous amount of data that needs to be stored, organised and retrieved in an efficient and logical way. Image file format is an important transport mechanism that allows the interchange of visual data between software applications and computer systems. However, there is no standard method of storing images in files. Instead, there are a vast array of file formats. Some of these are widely used (e.g. GIF, TIFF) and serve as exchange formats. Others are adapted to the needs of a particular applications or particular operating systems. Many commonly used formats are described in literature published by Murray and vanRyper (1994). There are two principal yet very important different methods of representing image data: bitmaps (raster or array of pixels) and vectors (David & Levine, 1994).

In most photogrammetric and remote sensing applications, image data is stored in the former method. In this method (i.e., bitmap representation), an image data which is formed from a set of numerical values is broken down into a grid specifying the colors of individual pixel or picture elements. The light value (lightness, darkness or color) of each pixel of the grid is recorded individually. On the other hand, vector representation usually refers to a means of describing an image as a series of lines, polygons and curves by numerically specifying key points (Murray & VanRyper, 1994). Vector data is always associated with attribute information (such as color and line thickness) and a set of conventions allowing a program to draw the designed objects.

The choice of a particular image file format for a given application generally involves several interdependent considerations including quality; flexibility; computation; storage, or transmission (David & Levine ,1994). The suitability of a file format for a given purpose is best determined by knowing its original purpose. For the purpose of this thesis, file formats such as page description language files, output device language files, and fax files are not described. The following sections look at the major image file formats used in softcopy photogrammetry, remote sensing and multimedia GIS, their structure and their header information. Image file formats covered in this thesis are: TIFF, GIFF, SUNRASTER, JPEG and MPEG.

2.2 Bitmap Versus Vector files

This section gives a general consideration of both bitmap and vector files: their applications, advantages and disadvantages in photogrammetric, remote sensing and GIS applications. Bitmap files formats are especially suited for the storage of real-world images; images with complex variations in colors, shades or shapes such as photographs , paintings and digitized video frames. Bitmap representation can record just about any conceivable image, since every image can be broken up into a grid as far as the human eye is concerned (David & Levine, 1994). Furthermore, the advantages of bitmap files include the following (Murray & VanRyper, 1994):

- Bitmap files may be easily created from existing pixel data stored in an array in memory;
- Retrieving pixel data stored in a bitmap file may often be accomplished by using a set of coordinates that allows the data to be conceptualised as a grid;
- Pixel values may be modified individually or as large groups by altering a palette if present; and;
- Bitmap files translate well to dot-format output devices such as CRTs and printers.

Bitmap files, however, do have the following drawbacks (Murray & VanRyper, 1994):

- They can be very large, particularly if the image contains a large number of colors. Data compression can shrink the size of pixel data, but the data must be expanded before it can be used, and this can slow down the reading and rendering process considerably. Also, the more complex a bitmap image (large number of colors and minute detail), the less efficient the compression process will be; and;

- They typically do not scale very well. Shrinking an image by decimation (throwing away pixels) can change the image in an unacceptable manner, as can expanding the image through pixel replication. Because of this, bitmap files must usually be printed at the resolution in which they were originally stored.

Vector files contain, instead, mathematical descriptions of one or more image elements, which are used by rendering application to construct a final image. Vector files are made up of descriptions of image elements, rather than pixel values. They work well for line art such as computer-aided design (CAD) drawings and images with simple shapes, shadings, and coloration. Charts, graphs and certain types of freehand illustrations and typically recorded as vector files (David & levine, 1994). Vector representation has more limitations than bitmap in the practical point of view, but it has the advantage of being far more efficient and flexible than bitmap for many application (David & Levine, 1994). Among other things, the advantages of vector files includes the following (Murray & VanRyper, 1994):

- Vector files are useful for storing image composed of line-based elements such as lines and polygons, or those that can be decomposed into a simple geometrical object, such as text;

- Many vectors files containing only ASCII format data can be modified with text editing tools. Individual element may be added, removed, or changed without affecting the other objects in the image; and;

- It is usually easy to render vector data and save it into a bitmap format file, or, alternatively, to convert the data to another vector format, with good results.

Some drawbacks of vector files include the following (Murray & VanRyper, 1994):

- Vector files cannot easily be used to store extremely complex images, such as some photographs, where color information is paramount and may vary on a pixel-by-pixel basis;

- The appearance of vector images can vary considerably depending upon the application interpreting the image. Factors include the rendering application's compatibility with the creator application and the sophisticated of its toolkit of geometric primitives and drawing operations;

- Vector data also displays best on vectored output devices such as plotters and random scan displays. High resolution raster displays are needed to display vector graphics as effectively; and;

- Reconstruction of vector data may take considerably longer than that contained in a bitmap file of equivalent complexity, because each image element must be drawn individually and in sequence.

2.3 General consideration of TIFF file format

The acronym TIFF stands for the Tag Image File Format as discussed in section 2.3.2.2. TIFF is a format for storage, transfer, display, and printing of bitmap (raster) images, such as clipart, logotypes, scanned documents. TIFF imagery file format can also be used for storing and transferring digital satellite imagery, scanned aerial photograph, elevation models, and scanned maps or the results of many types of geographic analysis. The largest possible Tiff file is $2^{32} - 1$ bytes in length. TIFF uses byte offsets (pointers) which makes it much easier to read Tiff files from a random access device such as a hard disk or flexible diskette.

It is also possible to read and write TIFF files on a magnetic tapes. TIFF is independent of specific operating medium (MS-DOS, UNIX,etc), filling system, compilers, and processor. The recommended MS-DOS and UNIX file extension for TIFF is dot Tiff (.Tiff). Macintosh file type recommends 'TIFF' as its file extension.

2.3.1 Major enhancement in TIFF

TIFF is the only full-featured format in the public domain, capable of supporting compression, tiling, and the extension to include geographic metadata. Its main strength is a highly flexible and platform independent format which is supported by numerous image processing applications. TIFF is designed to provide and promote a rich environment in which interchange of digital image data can be accomplished. It serves as an interface to several scanners and graphics arts packages. TIFF can be read and incorporated into a document or publication such as a desktop publishing package.

Another feature of TIFF which is also useful is the ability to decompose an image by tile rather than scanlines. This permits much more efficient access to very large imagery which has been compressed (since one does not have to decompress an entire scanline).
The following types of images can be stored and displayed by Tiff file:

- Black and white (one bit);
- Grayscale with (4, 8 , 16, 32 bits);
- Pseudocolor image (4, 8 bits);
- Palette color images; and;
- Multiband images with eight bits per pixel.

TIFF file offers no restrictions on the number of bands in the images. Both single and multiband images are supported by TIFF file.

The following compression schemes are supported by TIFF file for better disk space utilisation:

- Raw uncompressed;
- LZW compression scheme for grayscale, mapped color and full color images;
- PackBits compression for one bit images;
- CCITT Fax 3 & 4; and;
- JPEG.

TIFF image colorimetry tags are used to define the characteristics of full color RGB data, and thereby potentially improve the quality of color image reproduction (Aldus Microsoft Technical Memorandum, 1988).

6.3.2 An overview of TIFF File Structure

TIFF files are organised into a three-level of hierarchy and these are :

- the Image File Header (IFH);
- the Image File Directory (IFD); and;
- Bitmap data .

The subsequent sections describe each level in detail

2.3.2.1 Image File Header

A TIFF file begins with an eight-byte image file header, which contains the following information (Aldus Microsoft Technical Memorandoum, 1988):

- **Bytes -order Identifier :** It indicates whether the data in the TIFF file is written in Intel format (i.e. 4949 or II) or Motorola-format (i.e 4D4D or MM);

- **TIFF version number:** The number is 42 (2A in hex). This number does not change; and;

- **The byte offset of the first image file directory:** The directory may be at any location in the file after the header. The byte offset refers to the location of bytes with respect to the beginning of the file. The first byte of the file has an offset of zero.

2.3.2.2 Image field Directory (IFD)

Image file directory is a collection of information similar to a header, and it is used to describe the bitmap image data to which it is attached. There can be more than one image file directory. In this case, each subdirectory defines a subfile. The Image file directory contains information on:

- the height of the image;
- the width of the image;
- the depth of the image;
- the number of color planes of the image; and;
- the type of data compression used on the bitmapped data.

Figure 2.3 shows three possible arrangements of the internal data structure and the Image File Directory of a TIFF file. In each example, the Image File Header (IFH) appears first in the TIFF file. In the first example, each of the IFDs follows after the IFH and it is the most efficient arrangement for reading the IFD data quickly (Murray & VanRyper, 1994). In the second example, each IFD is written, followed by its bitmap data. This is perhaps the most common internal format of a multi-image TIFF file. In the last example, it is seen that the bitmapped data has been written first followed by the IFDs. This seemingly unusual arrangement might occur if the bitmapped data is available to be written before the information that appears in the IFDs.

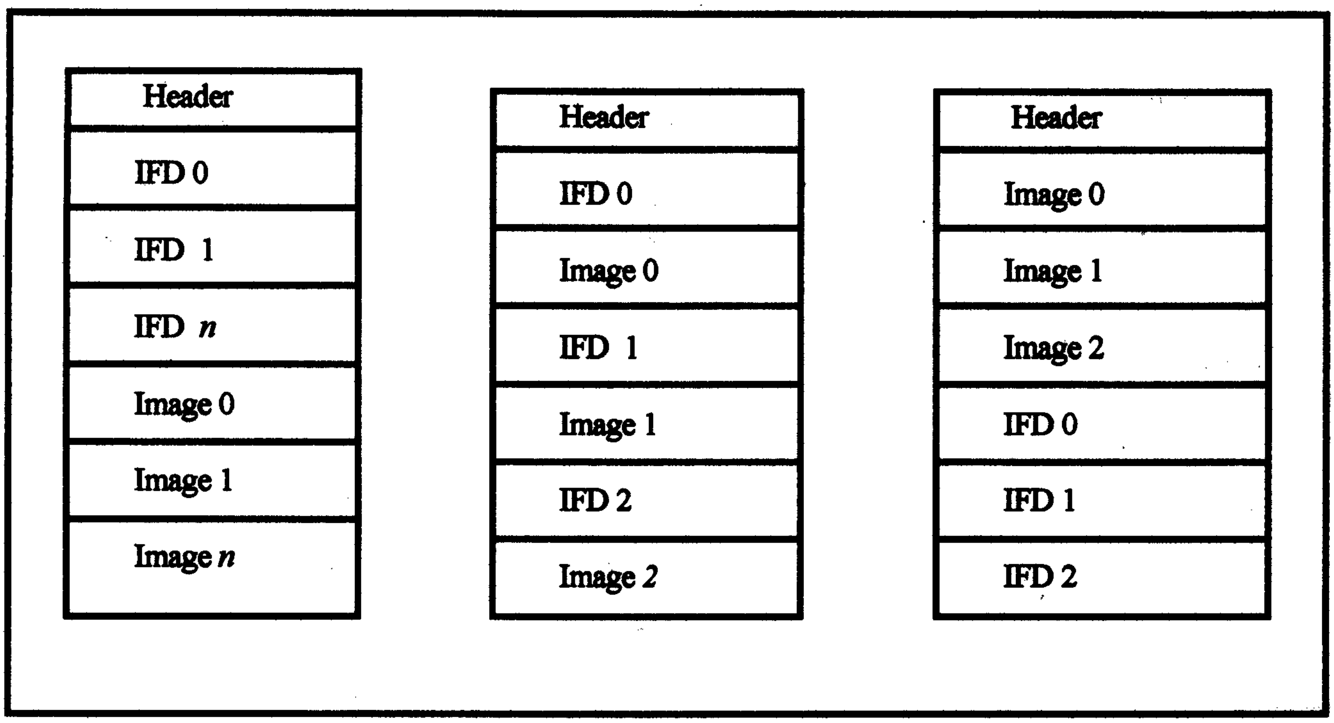


***FIG 2.1*: The three possible arrangement of data in a TIFF file**

An image file directory may vary in size because it may contain a variable number of data records called tags. Each tag is a 12-byte record that contains a unique piece of information about the bitmapped data (imagery). These tags are the only way that a TIFF reading file can obtain fundamental information about the TIFF image. That is, the tags support the image with information that the TIFF reader needs to know in order to control the appearance of image on the user's display or printer. TIFF tags are categorised into the following:

- **TIFF Public Tags**: These are tags defined by TIFF specification and may not be modified outside of the parameters in the latest TIFF specification. They contain information about the pixel architecture. Information such as the number of rows, number of columns, and bits per sample are conveyed by the TIFF public tags to the user of TIFF file; and;

- **Private TIFF tags (user-definable tags) :** Private tags are registered to specific users. The users who register for these tags are allowed by the designers to write information on them to support their software community.

Special functionality has been added to the tags in such a way that incompatibility problems with older TIFF packages is minimised. For example, TIFF version five file will be readable by older applications such as TIFF version four.

2.3.2.3 Bitmap data in the TIFF file

TIFF file contains only bitmap data which is organised into either a strip (i.e., one dimensional objects that has only a length) or tiles (i.e., two dimensional strip that has both width and length) Three tags are necessary to define the strips of a bitmapped data within a TIFF file. These are:

- RowPerStrip tag to indicate the number of rows of compressed, bitmapped data in each strip;
- StripOffsets tag to indicate the position of the first byte of each strip within the TIFF file; and;
- StripByteCount tag to maintain an array of values that indicate the size of each strip in bytes.

TIFF version 6 introduced the concept of tiles rather than strips of bitmapped data. In the tiles format, image compression algorithms such as JPEG which compresses data as two-dimensional tiles can be incorporated into its compression schemes. Storing the compressed data as tiles optimises the decompressions of the data quite significantly.

2.3.2.4 Compression Summary

TFF Image File Format support the following types of data compression:

- Packbits Run-Length Encoding (RLE);
- CCITT (Huffman) encoding;
- Lempel-Ziv-Welch (LZW); and;
- JPEG compression techniques.

Packbits RLE in TIFF is a byte-wise compression technique and most efficient at encoding runs of bytes. TIFF, among other file formats applies LZW compression techniques for image data files. In TIFF, the pixel data is packed (i.e., pixel packing is an efficient way to store data in contiguous bytes of memory) into bytes before being presented to LZW, so an LZW source byte might be a pixel value, part of pixel value, or several pixel values, depending on the image's bit depth and number of color channels (Murray & VanRyper, 1994). This approach of TIFF does not work very well for odd-size pixels, because packing into bytes creates byte sequence that do not match the original pixel sequences, so any pattern in the pixels are obscured.

If the pixel boundaries and byte boundaries agree (e.g., 4-bit pixel per byte, or one 16-bit pixel per every two bytes), then TIFF'S method work well. The CCITT defines three algorithms for the encoding of bilevel image data:
Group 3 one-dimensional (G3 ID);
Group3 two dimensional (G3 2D); and;
Group4 two dimensional G4 2D).

TIFF compression schemes follows the G3 ID and G4 2D specification. Group 3 encoding was designed specifically for bilevel, black-and-white image data . Group 4 is far more efficient form of bilevel compression.

2.3.3 GEOTIFF

GeoTIFF is an implementation of TIFF files which defines a set of TIFF tags provided to describe all cartographic information associated with TIFF imagery that originates from satellite imaging systems, scanned aerial photography, scanned maps, digital elevation models, or as a result of geographic analyses. The purpose of GeoTIFF implementation is to have a public domain, multi-vendor, non-proprietary, open and platform interoperable format for transfer and utilisation of any TIFF geographic image. GeoTIFF does not intend to become a replacement for existing geographic data interchange standards, such as the USGS SDTS or the FGDC metadata standard. Rather, it aims to augment an existing popular raster-data format to support georeferencing and geocoding information.

The GeoTIFF format provides enough information that the software can spatially display an image without requirement of any user intervention, such as typing in coordinates, digitising points, or other labour intensive and technical actions. That is, it reduces the need of users to be geographic experts in order to load a map-projected image or scanned map. The TIFF 6.0 specification is fully supported by GeoTIFF format. That is, GeoTIFF images fully comply with the TIFF 6.0 specification, and its extensions conform in every way to the scope of raster data supported by TFF version 6.0. The GeoTIFF tags and definitions are considered completely orthogonal to the raster-data descriptions of the TIFF specification and impose no restrictions on how the standard TIFF tags (baseline and extended) are to be interpreted, which color spaces, or compression types are to be used.

GeoTIFF uses a small set of reserved TIFF tags to store a broad range of georeferencing information, catering to geographic as well as projected coordinate systems needs. It is possible to customise geographic projections using GeoTIFF. Projections includes: UTM, US State Plane and National Grids, as well as the underlying projection types such as Transverse Mercator, Lambert Conformal Conic and Albert Equal Area. GeoTIFF uses numerical codes to describe the projection types, coordinates systems, datums and ellipsoid. These codes are derived from the EPSG list compiled by the Petrotechnical Open Software Corporation (POSC), and mechanisms for adding further international projections, datum and ellipsoid has been established. The GeoTIFF information content is designed to be compatible with the data decomposition approach used by the National Spatial Data Infrastructure (NSDI) of the U.S.Federal Geographic Data Committee (FGDC).

GeoTIFF uses a Meta Tag (GeoKey) approach to encode dozens of information elements into just six tags, taking advantage of TIFF platform-independent data format representation to avoid cross-platform interchange difficulties. These keys are designed in a manner parallel to standard TIFF tags, and closely follow the TIFF discipline in their structure and layout. New keys may be defined as needs arise. While GeoTIFF provides a robust framework for specifying a broad class of existing projection coordinate systems, it is fully extensible, permitting internal, private or proprietary information storage.

2.4 Graphic Interchange Format (GIF)

Graphic Interchange Format is a bitmap format created by CompuServe. The GIF format is capable of storing bitmap data with pixel depths of 1 to 8 bits (i.e., maximum of 256 colors) GIF images are always stored using the RGB color model and palette data. GIF is also capable of storing multiple images per file. The vast majority of GIF file contains 16-color or 265-color near- photographic quality images.

Gray-scale images such as those produced by scanners, are stored using GIF. GIF features includes:

- LZW compression;
- the sequencing or overlay of multiple images;
- interlaced screen printing; and;
- text overlays.

There are two versions of GIF specifications, both of which have been widely distributed. The original version was GIF 87a and many images were created in this format. The current version (GIF 89a) adds several capabilities including the ability to store text and graphic data in the same file. GIF was designed as an image communications protol used for the interactive viewing of online images.

2.4.1 An overview of GIF File Structure.

This section gives an overview of GIF 89a file structure which is the most recent revision of GIF file format. Figure 5.2 illustrates the basic layout of a GIF 89A image file. Just as with the version 87a, each image in the 89a version also begins with a Header, a Logical Screen Descriptor, and optional Global Color Table, Local Image Descriptor, an optional Local Color Table and , a block of image data. The sequence and the contents of the features common to both versions are described in the subsequent sections.

2.4.1.1 The Header and the Logical Screen Descriptor

- **The Header:** It is used only to identify the data stream as GIF. That is, it indicates the version of GIF decoder (87a or 89a) required to interpret the data that follows.

- **The Logical Screen Descriptor:** It contains information describing the screen and the color information used to create and display the GIF file image. It defines the size, pixel aspect ratio, and color depth of an image plane that encompasses the image or images that follows. It is analogous to the monitor screen on which the images originated. It also indicates whether a Global Color Table follows.

2.4.1.2 The Global Color and the Local Color Tables

- **The Global Color Table :** It is the color map used to index the pixel color data contained within the image data. If a Global Color Table is not present, each image stored in the GIF file contains a Local Color Table that it uses in place of a Global Color Table.

Global Color Table data always follows the Logical Screen Descriptor information and varies in size depending upon the number of enttries in the table.

- **The Local Color Table :** If a Local Color Table is present, it immediately follows the Local Image Descriptor and precedes the image data with which it is associated. A Local Color Table only affects the image it is associated with and, if it is present, its data supersedes that of the Global Color Table. Each image may have no more than one Local Color Table.

Header
Logical Screen Descriptor
Global Color Table
comment Extension
Application Extension
Graphic Control Extension
Local Image Descriptor
Local Color Table
Image Data
comment Extension
Plain Text Extension
Trailer

FIG 2.2: **The basic Layout of a GIF 89a file**

2.4.1.3 GIF Image Data and the Trailer

GIF image data is always stored by scanline and by pixel. It does not have the capability to store image data as planes, so when GIF files are displayed using pane-oriented display adapters, quite a bit of buffering, shifting, and masking of image data must occur before the GIF image can be displayed (Murray & VanRyper, 1994). The scan lines making up of GIF bitmap image data are normally stored in consecutive order, starting with the first row and ending with last row. The GIF format also supports an alternative way to store rows of bitmap data in an interlaced order. Interlaced images are stored as alternating row of bitmap data.

- **The Trailer:** The Trailer is a single byte of data that occurs as the last character in the file. It indicates the end of the GIF data stream and it appears in every GIF file.

2.4.1.4 GIF Control Extensions

This is a new feature which has been added in the version 89a (the most recent version). In the older version (GIF 87a), specialised blocks of information are used to control the rendering of the graphical data stored within a GIF image file. The design of GIF 87a only allowed the display of images one at a time in a " slide show" fashion. Through the interpretation and use of Control Extension data, GIF 89a allows both textual and bitmap-based graphical data to be displayed, overlaid, and deleted as in an animated multimedia presentation. The Control Extensions introduced by GIF 89a are:

- *The Graphic Control Extension blocks:* They control how bitmap or plain text data found in a Graphic Rendering block is displayed. Such control information includes whether the graphic is to be overlaid in a transparent or opaque fashion over another graphic, whether the graphic is to be restored or deleted, and whether user input is expected before continuing with the display of the GIF file data;

- *The Plain Text Extension blocks*: They allow the mixing of plain text ASCII graphics with bitmapped image data. Many GIF images contain human-readable text that is actually part of the bitmap data itself. Using the Plain Text Extension, captions that are not actually part of the bitmapped image may be overlaid onto the image. This can invaluable when it is necessary to display textual data over an image, but it is inconvenient to alter the bitmap to include this information. It is even possible to construct an 89a file that contains only plain-text data and no bitmap image data at all;

- *Application Extension blocks:* They allow the storage of data that is understood only by the software application reading the GIF file. This data could be additional information used to help display the image data or to coordinate the way the image data is displayed with other GIF imagefiles; and;

- *Comment Extension block*: It is used to insert a human-readable ASCII text into a GIF file or data stream.

2.4.2 Data Compression method in GIF

GIF support only LZW lossless encoding scheme. GIF requires each LZW input symbol to be a pixel value. Because GIF allows 1-to 8-bit deep images, there are between two and 256 LZW input symbols and the LZW dictionary is initialised accordingly. The LZW-encoded image data is stored as a continuous stream of data that is read from the beginning to the end, but in GIF, the encoded image data is stored as a series of data sub-blocks. The LZW decoder reads each sub-block for the decoding processing. It is irrelevant how the pixel might have been packed into storage originally; LZW will deal with them as a sequence of symbols. Unlike TIFF which does not work well for odd-size bit pixels, the GIF approach of data encoding works well for odd-size bit depths.

GIF files cannot be significantly compressed further when stored using file archivers such as PKZIP and ZOO. This is because the image data found in every GIFF file is always compressed using the LZW encoding scheme; the same compression algorithm used by most file archivers. Compressing a GIF file is therefore a redundant operation, which rarely results in smaller files and usually not worth the time and effort involved in the attempt.

2.5 Sun Raster File Format

The Sun Raster image file format is a bitmap format which is capable of storing black-and-white, gray-scale, and color bitmapped data of any pixel depth. It uses SunOs System Platform and it is supported by most UNIX imaging applications. The use of color maps and a simple Run-Length data compression are also supported.

2.5.1 Organisation of Sun Raster Files

The basic layout of a Sun Raster file is a Header, followed by an optional color map, and then by the bitmapped image data.

2.5.1.1 Sun Raster Header file

It is 32 bytes in length and has the following information:

- magic (identification) number which identifies a file as a Sun Raster image;
- width and height of image in pixels;
- number of bits per pixel;
- size of image data in bytes;
- type of raster file;
- type of color map; and;
- size of the color map in bytes.

2.5.1.2 Sun Raster ColorMap Type file

It indicates the type of color map included in the Sun Raster file, or whether a color map is included at all. The following three color formats can be found in the ColorMap Type file:

- No color map;
- RGB color map; and;
- Raw color map.

Typically, a black-and-white or gray-scale images do not need a color map. For the color images (RGB color map), the colors are separated into three planes, and are stored in RGB order with the first being the red values, the second values being green and the third being the blues values. A raw color map is any other type of color map not defined by Sun Raster files format and is stored as individual bytes values.

2.5.1.3 Sun Raster Image Data

The pixels in the image come after the color map. They are written left to right within a row, and by row from top to bottom. Bitmap files with a depth of 1 contains 2-color image data. These images do not have color map and each bit in the bitmap represents a pixel, with a value of 0 representing black and a value of 1 representing white. Raster files with a depth of 8 may contain either color or gray-scale image data.

Images with pixels eight or fewer bits in depth do not include a color map. If a color map is present in an 8-bit bitmapped file, then the pixel values are index pointers into a color map. Such an image contains only a maximun of 256 colors. Raster files with a depth of 24 or 32 normally do not have color maps. Intead, the color values are stored directory in the image data itself (true color bitmap). If a 24-bit image has a color map, it is either a raw color map, or an RGB color map that contains more than 256 elements.

2.5.2 Sun Raster Compression

Sun Raster image data is compressed using Simple Run-Length Encoding. This encoding method is found in any Sun Raster file regardless of the type of image data it contains. Sun Raster file uses the RLE schemes that uses three bytes in size, rather than two, to represent a *run*. The first byte is a flag value indicating that the following two bytes are part of an encoded packet. The second byte is the count value, and the third byte is the run value. The encoding of pixel runs does not stop at the end of each scan line because the Sun Raster bitmap is read as if it is a single stream of data.

2.6 JPEG file formats

JPEG is supported in a variety of formats and software, although each typically has its own peculiarities. TIFF version 6, for instance, support JPEG encoding, but the prospective developer should obtain the detailed TIFF specification to ensure proper handing of details. Similarly, the Photo Compressor in the QuickTime portion of Apple's System 7 operating system supports the standard baseline JPEG bitstream. This section takes a general look at JPEG file format and associated file formats. For an overview of other common image formats, including JFIF, JPEG-TIFF formats.

2.6.1 JFIF File Format

JFIF stands for JPEG File Interchange Format. It is the current standard for exchanging JPEG compressed files across hardware platforms. JFIF file is basically a JPEG data stream with a few restrictions and an identifying marker. JPEG data in general is stored as a stream of blocks, and each block is by a marker. Each block starts with a two-byte (Start Of Image) marker that identifies the type of segment. All JPEG data are stored with 16-bit word values in big-endian format, with the highest-order part of the value stored first.

A JFIF file contains a single image typically encoded using baseline DCT compression and Huffman encoding. A JFIF image may be either gray-scale or in YCbCr color representation. This is the format most adept at working on the Internet.

2.6.2 JPEG- TIFF

There is a variation of TIFF (Tagged Image File Format) that includes JPEG compressed images. A TIFF file can contain either a straight DCT or lossless JPEG image, or a series of JPEG-encoded strips or tiles, allowing pieces of the image to be recovered without having to read through its entirety.

2.7 Multimedia File Formats

Multimedia refers to the integration of text, images, audio and video in a variety of application environments. In addition to bitmap and vector file formats, there are also animation, multimedia, meta and scene file formats. Storage of multimedia data and information in a disk is similar to image file formats. In Multimedia formats, the following types of data is stored: text, image data, audio and video data, computer animation and binary data such as Musical Instrument Digital Interface (MIDI), control information and graphical fonts.

Multimedia file formats offer the ability to store these types of data in one or more existing data formats that are already in general use. For example, a multimedia format may allow text to be stored as Postscript or Rich Text Format (RTF) data rather than in conventional ASCII plain text format. Still-Image bitmap data may be stored as BMP or TIFF files rather than as raw bitmaps. Similarly, audio, video, and animation data can be stored using industry-recognised formats specified as being supported by multimedia file format.

Multimedia formats are also optimised for the types of data they store and the format of the medium on which they are stored. Multimedia information are stored on CD-ROM. In order for multimedia applications to work together and realise the benefits of distributed computing, a common interchange format for multimedia information has been developed. Most formats are designed for file-based interchange and a real-time delivery of multimedia materials from disk or across a network. The most current multimedia interchange file formats are:

- QuickTime Movie File (QMF) Format: It is a published file format for storing multimedia content for quicktime presentation. It presents a model for storage and interchange of time-related media that is independent of a system's built-in timing and synchronisation (Buford, 1994);

- Open Media Framework (OMF): It is an industry standardization effort led by Avid Technology to define a common framework and multimedia interchange format. It's primary concern is in representing time-based media such as video and audio; and;

- Multimedia and Hypermedia Information Encoding Expert Group (MHEG): This is an ISO working group that is defining an object-oriented model for multimedia and hypermedia interchange.

2.8 Animation File Formats

The storage of various types of multimedia data that have listed in section 4.8 cannot discussed in detail in this thesis. They have been discussed in detail in the book : Encyclopaedia of Graphics File Formats written by James D. Murray & William VanRyper. This section seeks to describe how animation data is stored using multimedia formats. Computerised animation is a combination of still and motion imaging. Each frame or cell of an animation is a still image that requires compression and storage. Animation files offer the ability for storing data of animation frame and for providing the information necessary to play back the frames using the proper display mode and frame rate.

Most animation files use delta compression schemes, which is a form of Run-Length Encoding that stores and compresses only the information that is difference between two images. Storing animation using multimedia format also produces the benefit of adding sound to animation. Most animation formats cannot store sound directly in their files and must rely on storing the sound in a separate disk file which is read by the application that is playing back the animation. Animated sequences are used by CAD programmers to rotate three dimensional (3D) objects so they can be observed from different perspectives; complex, ray-traced objects may be sent into movement so their behaviour can be seen; mathematical data collected by an aircraft of satellite may be rendered into animated fly-by sequence; and movie special effects benefits greatly by computer animation.

2.9 Summary

Several of the image file formats which have been described , especially GeoTIFF are under design, and the description given here could change with time. This section summarizes and compares the major contributions of each image file formats. There are two different methods of representing image data: Bitmaps and vectors. Vector data refers to a means of representing lines, polygons or curves whereas bitmap data is an array of pixels. In most applications in softcopy photogrammetry and remote sensing, image data is usually stored as bitmaps. One practical problem in bitmap representation of image data is image size. A high resolution color image file requires more bytes for storage and more bytes for memory, processing and display than lower resolution images. Many file formats such as TIFF, GIFF, SUNRASTER and others have been developed to store, organise and retrieve image data in an efficient and logical way.

TIFF is a format for storage, interchange, display and printing of bitmap images. Its main strength is a highly flexible and platform-independent format which is supported by numerous image processing applications. TIFF is a full-featured format in the public domain, capable of supporting compression, tilling, colorimetry calibration, and the extension to include geographic metadata.

The TIFF file extension, GeoTIFF, is embedded in TIFF file format as tags to handle cartographic and geographic information. Another feature of TIFF which is also useful is the ability to decompose an image by tiles rather scanlines.

GIF is a bitmap format to store bitmap data with a pixel depths of 1 to 8 bits. GIF images are always stored using RGB color model and pallete data. Sunraster file formats is capable of storing black-and -white, grey-scale, and color bitmapped data of any pixel depth. JFIF is the JPEG interchange file format which handles JPEG compressed files across hardware platforms. In addition to bitmap and vector file formats, there are also animation, multimedia, meta and scene file formats. Storage of multimedia data and information in a disk is similar to image file formats.

Chapter 3

Fundamental issues in Image Compression Techniques

3.1 *Introduction*

This chapter review the general techniques for still and motion image compression. This thesis will not review all of the techniques for compression described in the literature, since most of them are not implemented in standard software packages available for dedicated image processing. Image compression techniques attempts to reconstruct image data into a different, more compact representation conveying the same information. Reducing the amount of data needed to reconstruct images results in some compression but with accompanying distortions or degradation in the reconstructed image. Measures of compression techniques performance are basically composed of three entities (Luther, 1994 & Schalkoff, 1989):

- A quantitative or qualitative assessment of the degradation (if any) of the image data;
- A quantitative measure of the amount of data reduction expressed in terms of memory bits per image or bits per pixel; and;
- A measure of the algorithm complexity, particularly with respect to the speed of compression or decompression.

3.2 *Lossless Versus Lossy Compression Techniques*

One of the most fundamental concepts in image compression is the difference between lossless and lossy compression techniques. In general terms, lossless techniques create compressed files that decompress into exactly the same file as the original, bit for bit. That is, when a chunk of data is compressed and then decompressed, the original information contained in the data is preserved (i.e. no data is lost or discarded). The most popular lossless compression techniques are Huffman, adaptive Huffman, Ziv-Lempel-Welch (ZLW) and arithmetic encoding.

Lossy compression techniques on the other hand, create compressed files that decompress into images that look similar to the original but different in digital make-up. This means that lossy compression techniques discard some of the data in the image in order to achieve compression ratios better than that of most lossless compression methods. JPEG and fractal compression techniques are suited to lossy compression of large, full 24 bit images. The storage space saved by lossy compression techniques can be very large. Some of the common compression techniques are predictive coding, transform, and scalar quantization. Figure 3.1 gives a summary classification of various image compression techniques (Luther, 1994). The succeeding sections take a closer look at lossless and lossy compression techniques, as well as the algorithm and transformations involved. The review of these techniques are based on publications by Luther (1994), Jain (1989) and Storer (1988).

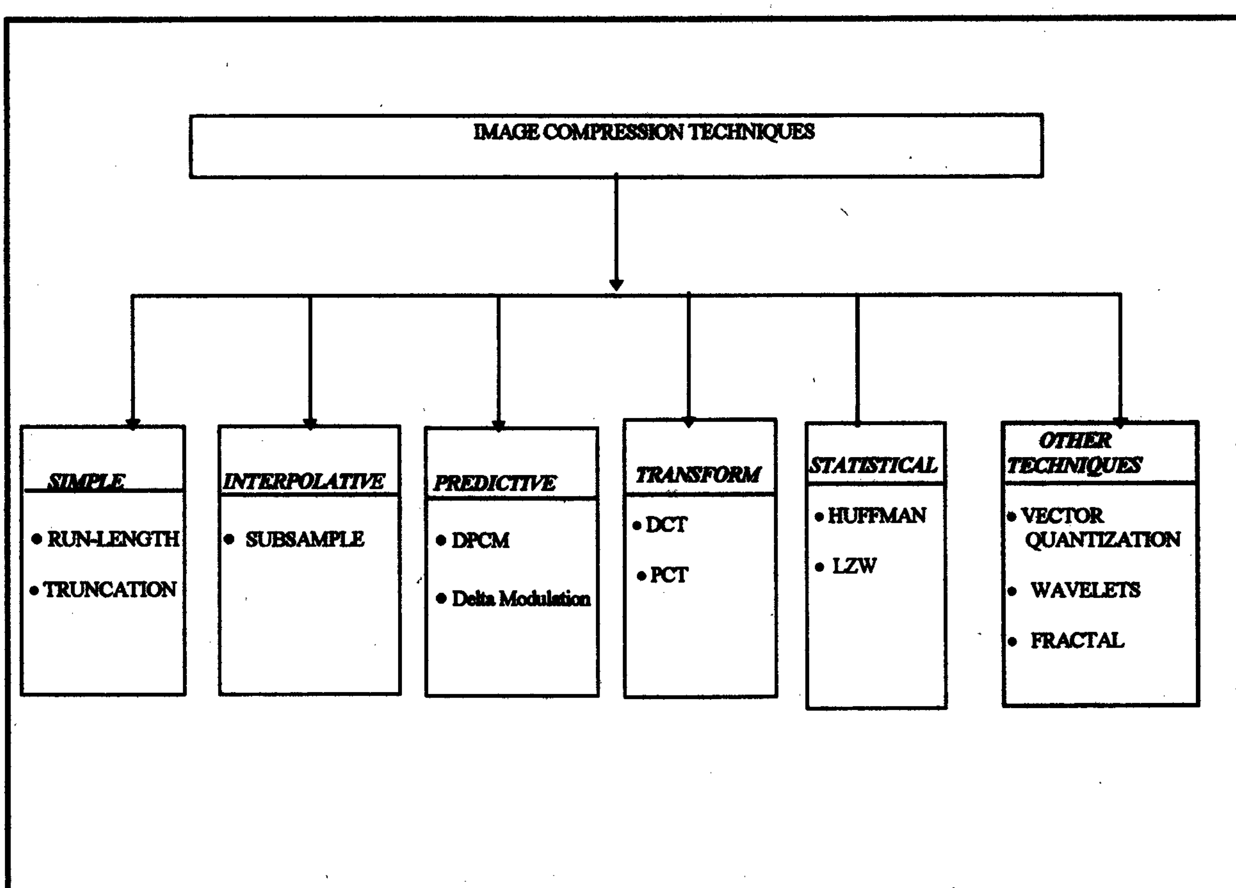


FIG 3.1: **Image Compression Techniques (after Luther 1994)**

3.3 Simple Image Compression Techniques

Image compression techniques such as Run-length Encoding (RLE) coding and Truncation are classified as Simple image compression techniques (Luther, 1994). Each of these techniques are discussed in terms of their compression abilities and working principles.

3.3.1 The Run-Length Encoding

Run-length coding capitalises on the high pixel-to-pixel correlation in the image, and then recodes the data to eliminate the redundancies. That is, blocks of repeated pixels are replaced with a single value and a count of how many times to repeat that value. To achieve a high degree of compression, the run-length data is recoded into digital codes of varying length, with shorter codes assigned to the more likely original code values (Cavigioli, 1990). This encoding technique is lossless and supported by most bitmap or raster file formats such as TIFF, BMP and PCX. It is suited for compressing any type of data regardless of its information content but the contents of the data will affect the compression ratio achieve by RLE.

The RLE compression efficiency depends on the type of image data being encoded. A black-and-white image for example will encode very well, this being due to the large amount of continuous data with the same color. It also works well with computer-generated images, cartoons and CLUT images (Luther, 1994). RLE is both easy to implement and quick to execute.

3.3.2 Truncation Technique

Truncation is a simple and fast lossy compression technique for grey scale images. The key idea of truncation is to reduce image data through arbitrary lowering of bits per pixel so that the quality of the image will remain acceptable and at the same time the demand for the storage space will decrease. This is done by throwing away some of the least significant bits for every pixel.

The greatest deficiency of truncation is a relatively high bit rate as compared to other coding techniques like Discrete Cosine Transform (DCT). Other significant drawbacks are ragged edges (staircase effects) in the reproduced images and blocky appearance in some cases (Franti *et al*, 1994). However, truncation is attractive because its processing is extremely simple.

3.4 Interpolative Techniques

Interpolative compression at the pixel level consists of transmitting a subset of the pixels and using interpolation to reconstruct the intervening pixels. Luther (1994) argues that, in the real meaning of compression, this is not a valid technique for use on the entire pixels because it effectively reduces the number of independent pixels contained in the output. The interpolation in that case is simply a means for reducing the visibility of pixellation, but the output pixel count is still equal to the subset. However, there is one case where interpolation is a valid compression technique. It can be used in coding color and multispectral images.

In practical image coding schemes for color imagery, the chrominance components of the image are sampled at correspondingly lower rates. This process is called color subsampling, and it is most valuable with luminance- chrominance component images (YUV, YIQ). For example, in a digital system starting with 8 bits each of YUV (24 bpp), the U and V components can be subsampled by a factor of four both horizontally and vertically(a ratio of 16:1). This technique gives excellent reproduction of real pictures. Interpolation can also be applied between frames of a motion sequence.

3.5 Predictive Techniques

The predictive compression technique exploits the correlation between neighbouring pixel values to determine the grey values of a digital image. This means that, it predicts the next sample (i.e. line, pixel or frame) from the previous sample.

The technique removes mutual redundancy between successive pixels and encodes only the new information. For example, an image of constant grey levels is fully predictable once the grey level of the first pixel is known. On the other hand, a white noise random field is totally unpredictable and every pixel has to be stored to produce the image (Jain, 1989). Techniques such as Delta Modulation and Differential Pulse Code Modulation (DPCM) fall into this category. Only DPCM is fully discussed below in the succeeding sections, as it is the most common approach to predictive techniques.

3.5.1 The Differential Pulse Code Modulation

The Differential Pulse Code Modulation (DPCM) is the simplest form of predictive techniques which operates at the pixel level. In DPCM, the predictive image is computed on a pixel-by-pixel basis by investigating the correlations between the gray values of a pixel and all its neighbours. That is, adjacent pixels are compared and only the difference between them are transmitted. The predicted image is subtracted from the original image to form a differential image in which pixel values are less correlated than in the original. This is possible, because adjacent pixels are often similar, and the difference values have a high probability of being small, and they can safely be transmitted with fewer bits than it would take to send a whole new pixel.

The difference image is quantized and encoded (Novak & Shahin, 1996). In decompression, the difference information is used to modify the previous pixel to get the new pixel. Normally the difference bits would represent only a portion of the amplitude range of an entire pixel, and any call for a full-amplitude change would cause a slope overload effect. This effect causes smearing of high contrast edges in the image. The distortion from the slope overload may be reduced by Adaptive Differential Pulse Code Modulation (ADPCM). Adaptive prediction in this respect usually reduces the prediction error prior to quantization, because prediction is dynamically modified according to the existing configuration of grey values (Novak & Shahin, 1996). Adaptive prediction results in less quantization errors and a better quality of the reconstructed image. Generally, non-adaptive predictors perform poorly at grey level edges. The independent lossless coding in JPEG is based on DPCM. This coding is provided for situations when no image degradation is allowed.

3.6 Transform Coding Techniques

Transform coding applies a mathematical transform to the pixel data to put the resulting data values into another domain. That is, it is a process that converts a bundle of data (a group of pixels, usually a two dimensional array of pixel from the image) into an alternate form which is more convenient for some particular purpose. A transform is chosen which consolidates pixel information into a more compact form. By this, the information in the image can be transformed or stored using less data (Cavigioli, 1990). Once in this form, other techniques are employed to compress the information even further. For example, in a lossy system, thresholding and quantization can be used to select the more significant values from the less significant ones, and then transmit only the most important ones.

For lossless systems, entropy coding (such as Huffman coding) and run-length encoding can be used to reduce the bit rate even further. In general, transform coding becomes more effective with larger block sizes, but the calculations also become more difficult with larger blocks. Due to this, it involves subdividing a digital image into smaller blocks and performs a unitary transform on each block (Jain, 1989). The transform is ordinarily designed to be reversible which means that there exists an inverse transform which can restore the original data. The most common transforms used in image compression are the Principal Component Transform (PCT), the Discrete Cosine Transform (DCT), and the Waslsh-Hadamard Transform. In this thesis, only the Discrete Cosine Transform is discussed.

3.6.1 The Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is relatively computer-intensive, and it is virtually impossible to perform a DCT on a large image. The amount of calculation needed to perform a DCT transformation on 256 by 256 grey scale block is prohibitively large (Nelson, 1991). For practical implementation, JPEG group has selected an 8 x 8 block for the size of DCT calculations. Thus, 64 pixels values at a time are processed by the transform.

The output is 64 new values, representing amplitudes of two-dimensional spatial frequency components of the 64-pixel block. The coefficient for zero spatial frequency is called the DC coefficient, and it is the average value of all the pixels in the block. The remaining 63 coefficients are the AC coefficients and they represent the amplitudes of progressively higher horizontal and vertical spatial frequencies in the block.

Since adjacent pixel values tend to be similar or vary slowly from one to another, the DCT processing provides opportunity for compression by forcing most of the signal energy into the lower spatial frequency components (Luther, 1994). In most cases, many of the higher-frequency coefficients will have zero or near zero values and can be ignored. A DCT decoder performs the reverse process where the spatial frequency coefficients are converted back to pixel values. JPEG provides a lossy compression based on the DCT. The Mathematical formulae of DCT are presented in section 4.3.3 of chapter 4.

3.7 *Statistical Coding Techniques*

Statistical coding techniques (sometimes called entropy coding) take advantage of the statistical distribution of the pixel values of an image or of the statistics of the data created from one of the techniques already discussed above. Statistical coding may be either in the compression algorithm itself or applied separately as part of the bit assignment following another compression technique. Statistical coding works on the principles that some image data values will occur more frequently than other data values and the coding techniques is set up to code the more frequently occurring with words using fewer bits, and less frequently occurring values will be coded with longer words (Buford, 1994). This results in a reduced number of bits in the final bitstream, and it is a lossless technique. The most widely used form of this coding are Huffman and Lempel-Ziv-Welch (LZW).

3.7.1 Huffman Coding Techniques

Huffman coding relies on a statistical model to exploit predictability which gives a reduction in the average code length used to represent the pixel values. The basic scheme is to assign a binary code unique value, with the codes varying in length. That is, the Huffman method results in a variable length code. The length of a code is inversely proportional to the probability of the symbol. This means that, long code are used for less frequently occurring symbols whereas short codes are used for more frequently occurring values. These assignments are stored in a conversion table, which is sent to the decoding software before the codes are sent (Kay & Levine, 1995). The compression efficiency of Huffman encoding varies with the type of image and different formats, but few get any better than 8:1 compression.

Huffman coding tends to perform less well for files containing long runs of identical pixel values, which can be better compressed using run-length or other coding. Huffman encoding also needs accurate statistics on how frequently each value occurs in the original life. Without accurate statistics, the final file is not much smaller, and may even turn out longer than the original. So to work properly, Huffman is usually done in two passes. In the first pass, the statistical model is created; in the second, the data is encoded. As a result, and because variable-length codes requires a lot of processing to decode, Huffman compression and decompression are relatively slow processes.

Another deficiency in Huffman coding is sensitivity to dropped or added bits. Since all the bits are jammed together with regard to byte boundaries, the decoder's only way of knowing when a code is finished by reaching the end of a branch. If a bit is dropped or added, the decoder starts in the middle of a code and the rest of the data becomes redundant (Kay & Levine, 1995). Another technique which efficiently represents shorter codes for frequently occurring sequences of pixels values (more-probable values) and longer codes for infrequently occurring sequences of pixels values is Arithmetic coding.

3.7.2 Lempel-Ziv-Welch coding techniques

Lempel-Ziv-Welch (LZW) coding technique exploits redundancies in the patterns it finds. Unlike Huffman encoding, LZW does not construct a table of codes in advance of coding. It is adaptive in that it builds a more effective table of codes as it goes along. LZW is referred to as a *substitutional* or *dictionary-based* encoding technique which builds a *data dictionary* (also called a *translation* or *string table*) of data occurring in an uncompressed data stream (Murray & VanRyper, 1994). Patterns of data (*substrings*) are identified in the data stream and are matched to the entries in the dictionary. If the substring is not present in the dictionary, a code phrase is created based on the data content of the substring , and is stored in the dictionary.

LZW encoding is reversible . It is capable of working on almost any type of data and provides compression ratios between 1:1 and 3:1. Noisy images, i.e. those with random variations in data values are hard to compress with LZW. Its implementations do not generally use shorter codes for more frequently occurring values as in Huffman coding. LZW is a lossless technique and its principles are applied in several image file formats, such as GIF and TIFF.

3.8 Other Image Compression Techniques

3.8.1 Vector Quantization

Like DCT, Vector Quantization (VQ), also divides the images into 8 x 8 blocks, but the information quantized is completely different. Vector Quantization is a recursive (multistep), algorithm with inherently self-correcting features. The first step is separating similar blocks into categories and building a reference block for each category. The original blocks are all discarded. During decompression, the single reference block will replace all of the original blocks in the category (Ozer, 1995). After the first set of reference blocks have been selected, the image is then decompressed and compared to the original.

Typically there will be many differences, so an additional set of reference blocks is created to fill in the gaps created during the first estimation. This is the self-correcting aspect of the algorithm. The process is then repeated to find a third set of reference blocks to fill in the remaining gaps (more self -correction). These reference blocks are all posted in a look-up table to be used during decompression. The final step is to use lossless techniques such as Run-Length Encoding to further compress the remaining information. Vector Quantization is obviously computationally intensive. However, decompression, which simply involves pulling values from the look-up table is extremely simple and fast. Vector Quantization is a public-domain algorithm.

3.8.2 Wavelets

Wavelets combine completely different methods of image analysis with the best features of JPEG and Vector Quantization. The first step is to filter the image with a high-pass and low-pass filter, essentially creating multiple views of the image. This clearly identifies the location of low-frequency and high- amplitude information and high-frequency and low-amplitude information. A tree-based encoder works through the different views of the image, starting with low-frequency data. After each encoding run, the codec checks the encoded information agaist the next image view. This is similar to the self-correcting aspects of Vector Quantization. If the higher-frequency information does not accurately describe the text view, the difference data is encoded. Otherwise, no additional information is encoded and the codec begins searching the next level, checking for accuracy at each level.

Once the image is completely encoded, a simple quantization method similar to JPEG's is used to compress the data. Once again, the high-frequency and the low-amplitude information is compressed more than the low-frequency and high-amplitude information. This preserves edges and targets the bulk of the compression towards areas not readily noticed by the human eye (Ozer, 1995). The final step is lossless compression through Huffman Encoding. Wavelets is a symmetric technology that compresses and decompresses very quickly. It is a public-domain algorithm.

3.8.3 *Fractal Compression*

Like JPEG and Vector Quantization, fractal compression starts by breaking the image into blocks. The next step, similar to Vector Quantization, is comparing blocks to other larger blocks to find similar bocks. However, rather than storing this information in a look-up table, the fractal process converts these self-similar regions into equations that can be used to create the image. In essence, the process converts the original pixel-related information into mathematical model of the image. This mathematical model is technically resolution-independent, meaning that fractal images can be zoomed to any resolution, irrespective of the original size. This characteristic, called scalability, is the primary advantage of fractal techniques (Ozer, 1995). Searching for self-similar regions is a lengthy process, and fractal compression is extremely time-consuming. However, the technology is asymmetric, and decompression is extremely simple and fast.

3.9 *Summary*

The main aim of image compression techniques is to minimise image data volume with insignificant loss of information, and all basic image compression groups have advantages and disadvantages. The primary difference between lossy and lossless compression techniques is the inclusion of quantization in lossy techniques. The reduction of the number of output symbols at the quantization step leads to distortions in the reconstructed image in exchange for higher compression ratios. The distortions introduced by the simple compression (i.e. truncation) and interpolative (i.e. subsampling) techniques for a given level of compression are generally much larger than the more sophisticated methods available for image compression.

Transform based (e.g DCT) techniques better preserve subjective image quality, and are less sensitive to statistical image property changes both inside a single image and between images. Prediction techniques (DPCM), on the other hand, can achieve larger compression ratios in a much less expensive way, tend to be much faster than transform based or Vector Quantization compression schemes, and are easily realised in hardware.

Adaptive techniques improve performance substantially. For sequences of images, an additional compression technique, motion compensation, based on similarity between successive frames has been developed to enable large data reduction in motion compression. Most image compression standards or applications actually employ several of these techniques in combination. One of these is JPEG image compression for still and motion images. The next chapter reviews the algorithm structure of JPEG image compression standard.

Chapter 4

JPEG Image Compression Standard for Still Images

4.1 Introduction

Image compression techniques which have been described in the preceding chapter are available to create algorithms (thus, algorithm consists of one or more techniques which operates on the raw image to create a compressed bitstream) to reduce image data to the point of manageability within common place networks and storage devices. However, practical applications require that all users who wish to interchange compressed digital still images or motion images must use exactly the same algorithm choice (Luther, 1994). This, therefore, expresses the need for an international standard that will enable both interoperability and economics of scale. Standardisation of image data compression algorithm will also allow the orderly growth of markets and multimedia technology which use digital image technology.

In addition to digital photogrammetry, remote sensing and multimedia GIS, application fields such as desktop publishing, graphic arts, color facsimile, medical imaging, computer multimedia have a need for a continuous-tone image compression standard. Driven by the needs of the latter market, there has been an increasing effort to develop an international image compression algorithm standard for still images by the International Organisation for Standardisation (ISO) and the International Electrotechnical Commission (ICE). There are two working parties for algorithm standardisation in a joint ISO/ICE committee. These working parties are the Joint Photographic Expert Group (JPEG), which deals with an international standard for a general purpose, continuous tone, still images compression, and the Motion Picture Coding Experts Group (MPEG) which defines a compression algorithms for motion compression. The latter (i.e., MPEG) is not discussed in this thesis.

Standardisation of image compression algorithms can be done at several levels. It can be based on hardware or software. The review of JPEG compression standard is based on publications by Luther (1994), Murray and VanRyper (1994), ISO/IEC JTI Draft International Standard (1991), and Wallace (1991).

4.2 The JPEG Image Compression Standard

JPEG provides a compression method that is capable of compressing color or grey scale continuous-tone image data with a pixel depth of 6 to 24 bits at a reasonable speed and efficiency. It is a lossy compression standard, meaning that some aspect of the image data is discarded. It does not necessarily reconstruct the original image bit-by-bit. Things which are discarded in JPEG lossy compression include things the eye does not need for perception in a given situation, such as unnecessary sharp edges or high spatial resolution of color (Kay & Levine, 1995). It is this effect which is of concern here since edges are often features for measurement. JPEG lossy compression makes a difference in disk space and transmission time. It can reconstruct a full-color picture nearly indistinguishable from the original using one bit per pixel for storage. However, the amount of compression achieved depends upon the content of the image data.

Another useful property of JPEG is that the degree of lossiness can be varied by adjusting compression parameter called a "quality setting or a "quality factor". This means that an end user can trade off file size against output image quality. Although different implementations have varying scales of quality factors, a range of 1 to 100 is typical. A factor of 1 produces the smallest , worst quality images; a factor of 100 produces the largest, best quality (i.e., lossless). The optimal quality factor depends in each case the image content and is therefore different for every image. JPEG compresses very well with the following real world subjects: photographs, video stills or any complex graphics that resemble natural objects. Animation, ray tracing, line art, black-and white documents and typical vector graphics do not on the other hand compress very well under JPEG.

There are lossless image compression algorithms, but JPEG archives much greater compression than is possible with the lossless method. Lossy JPEG compression is increasingly popular for computer video and multimedia applications, and applications such as indexing image archives. The most commonly cited JPEG lossy compression algorithm is the baseline algorithm of JPEG which uses Discrete Cosine Transform (DCT) to encode images. The Baseline mode of operation is discussed in more detail in subsection 3.2.3 of this chapter. JPEG itself does not define a standard image file format, several image file formats have been modified to fill the needs of JPEG data storage. The most common are JFIF files and JPEG-TIFF files.

4.2.1 The Purpose of JPEG

The goal of JPEG is to develop an image compression method for still-frame, continuous-tone images (both for color and grey scale). Generally, it is used to cover as wide range of applications in communications and image-based computer applications as is feasible for a single standard (Sarjakoski & Lammi, 1994). The scope of this goal is best seen by listing them in detail (Luther, 1994 & Wallace, 1991):

- To be at or near the state of the art, for degree of compression versus image quality;
- To be parameterizable so that the user can select the desired compression versus quality tradeoff;
- To be applicable to practically any kind of source image, without regard to dimension, image content, and aspect ratio;
- To have computational requirements that are reasonable for both hardware implementation; and;

- To support four different modes of operations:
  - sequential encoding, where the image component is encoded in the same order that it was scanned;
  - progressive encoding, where the image is encoded in multiple passes so that a coarse image is presented rapidly, followed by repeated images showing greater and greater detail;
  - lossless encoding, where the encoding guarantees exact reproduction of all the data in the source image; and;
  - hierarchical encoding, where the image is encoded at a multiple resolutions.

4.2.2 The Algorithm Structure of JPEG

JPEG is not a single, fixed algorithm, instead, it may be thought of as a family or a toolkit of several image compression techniques that may be altered to fit the needs of the user. Fig 4.1 shows the relationships of all current JPEG algorithms (Cavigioli, 1990).

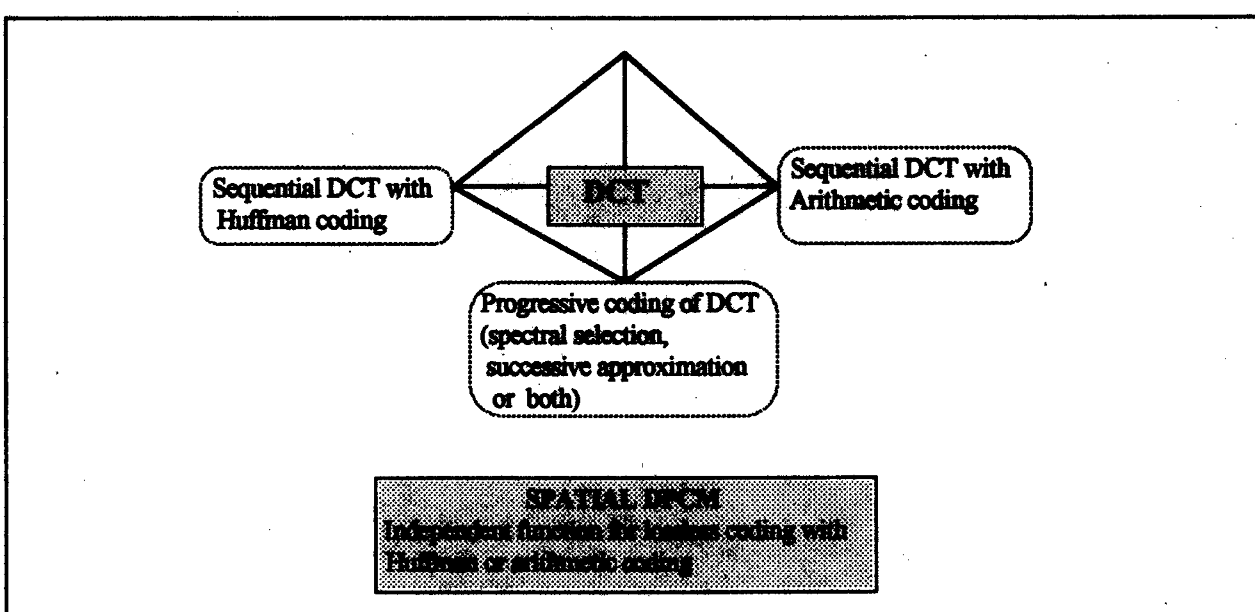


FIG 4.1 **: Diagram showing a family of JPEG algorithms**

The JPEG standard algorithm can be divided into the following modes of operations:

(a) Baseline (sequential) encoding mode;

(b) Progressive encoding mode;

(c) Hierarchical encoding mode; and;

(d) Lossless encoding mode.

The lossy modes of operation (baseline, progressive and hierarchical) are implemented with DCT encoding of 8 x 8 pixel blocks, followed by one of the two statistical (i.e Huffman or arithmetic) coding methods, while the lossless mode is implemented with simple prediction coding followed by statistical coding. The succeeding sections provide an overview of the JPEG standard. The discussions concentrate on the Baseline encoding, because the Baseline encoding is sophisticated or sufficient enough for many applications and it explains the common idea behind most of the JPEG modes. Besides, the JPEG implementations currently on the market typically support the baseline encoding mode (Sarjakoski & Lammi, 1991).

4.3 The stages of Baseline JPEG Compression

The ISO JPEG standard for still images is based on the 8 x 8 (two-dimensional) Discrete Cosine Transform (DCT), followed by quantization and statistical coding. Although not strictly part of the JPEG compression, most implementations start by converting the RGB image into luminance-chrominance color space. In practice, JPEG works well only on images with depth of at least four or five bits per color channel. The baseline standard specifies eight bits per input sample.

The baseline JPEG operates through the following process (Murray & VanRyper, 1994):

i) Preliminary scaling and color transform of the image into an optimal color space;

ii) Color subsampling chrominance components by averaging groups of pixel together;

iii) Applying a Discrete Cosine Transform (DCT) to blocks of pixels, thus removing redundant image data;

iv) Quantizing each block of DCT coefficients using weighting functions empirically optimised for the human eye; and;

v) Encoding the resulting coefficient (image data) using a Huffman variable word-length algorithm to remove redundancies in the coefficients.

Figure 4.2 below is a diagrammatic representation of the steps in the process of Baseline JPEG compression (adapted from Murray and VanRyper (1994)).

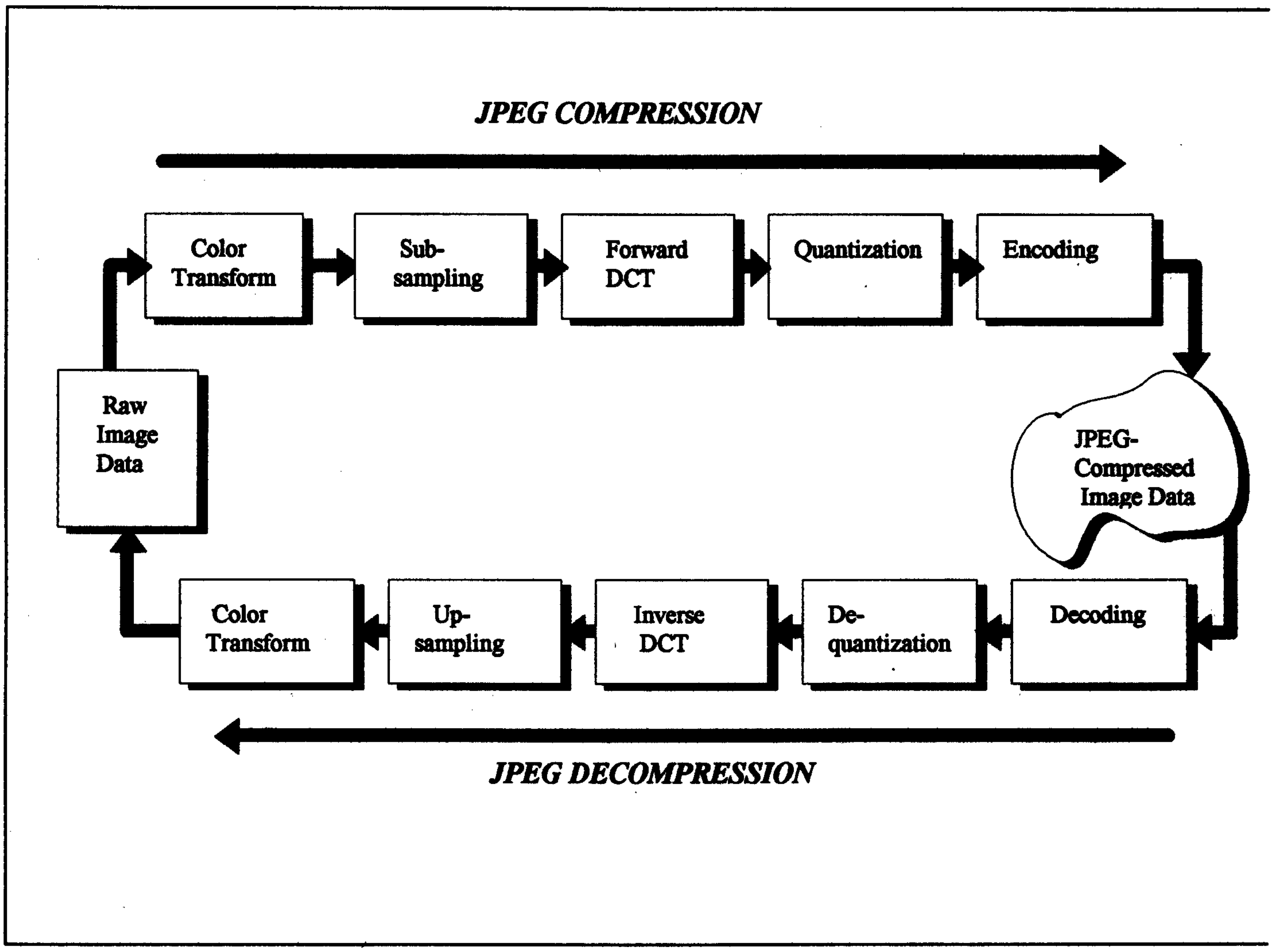


***FIG 4.2*: Diagram showing JPEG Compression and Decompression**

4.3.1 Preliminary Scaling and Transformation into the YUV Color Space

Although grey level imagery is the subject of this discussion, for completness a short overview of the application of JPEG to color imagery is given. JPEG algorithm is capable of encoding images that use any color space. It compresses each color component separately and it is completely independent of any color-space model (such as RGB, HIS, CMY). Although it is possible to compress the usual Red, Green and Blue components, JPEG compression works better when applied to color data expressed as luminance (brightness) and chrominance color space such YUV or Ycbcr.

The lumimance (Y) of YUV color space contains the high-frequency brightness information whereas the two chrominance components(U and V) contain the high-frequency "color" information. This is helpful because the human eye is more sensitive in the high-frequecy, luminance component (Y), and less sensitive to changes in color (UV), the chrominance components can be coded with more loss than the luminance channels. Most of the information in UV can effectively be discarded and an acceptable image still reconstructed using many fever U and V samples than Y samples. Subsampling phase which is discussed in the next section takes advantage of using YUV color space in JPEG compression.

4.3.2 Color Subsampling of Chrominance Components (U and V)

JPEG allows different components of the color space to be sampled at different rates and the techniques of sampling some components at lower rates than others is known as subsampling (Sarjakoski & Lammi, 1991). Because chrominance values need not to be considered as frequently as luminance, the spatial resolution of the U and V components can be decreased. This stage is a cause for information loss because part of the data is systematically deleted. Thus, for color imagery, the sampling rate parameter together with the quantization parameter (discuss in section 4.3.4), determine the compression rate and reconstruction image quality.

In a standard JPEG file, all these parameters are included in the file header so that a decoder can reconstruct the quality correctly.

4.3.3 Discrete Cosine Transform

The Discrete Cosine Transform converts an array of intensity data into an array of frequency data that describe the variations in image. JPEG applies the DCT to 8 x 8 blocks of pixel data. The data points in each block are numbered from (0,0) at the upper left to (7,7) at the lower right, with f(x, y) being the data value at (x, y) as shown in the equation below. The DCT then converts the colors and the pixels into frequency space by describing each block in terms of the number of color shifts (frequency) and the extent of the change (amplitudes) which produces a new 8 x 8 block of transformed data using the formula:

$$f(u, v) = \frac{1}{4} C(u)C(v)\left[\sum_{x=0}^{7} \sum_{y=0}^{7} f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}\right]$$

Where:

x,y are spatial coordinates in the block domain;

$f(x,y)$ are gray values of the pixels of the block;

u, v are coordinates in the frequency domain; and

$C(u), C(v) = 1/\sqrt{2}$ for $u, v = 0$
$C(u), C(v) = 1$ otherwise

The DCT is reversible, using the inverse DCT:

$$f(x, y) = \frac{1}{4}\left[\sum_{x=0}^{7} \sum_{y=0}^{7} C(u)C(v) f(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}\right]$$

The DCT itself is lossless except for round-off errors which will generally make the calculated values slightly different from the original ones. The DCT calculation is fairly complex and the point of changing from intensities to frequencies is that slow changes are much more noticeable than fast ones (Kay & Levine, 1995), so the low-frequency data is more important than high-frequency data for reconstructing the image. High-frequency data information can be easily discarded without losing low-frequency information. Quantization takes advantage of this fact.

4.3.4 Quantization

Quantization sets the precision to which each of the values resulting from the DCT is stored. In other words, the goal of this processing step is to discard information which is not visually significant. To discard an appropriate amount of information, the JPEG compressor uses linear quantization which means that each of the DCT values is divided by a predetermined quantization factor and rounds that results to the nearest integer:

$$F(u, v) = \text{Integer Round } \frac{f(u, v)}{Q(u, v)}$$

where:

F(u, v) is the coefficient after DCT at pixel (u, v); and;

Q(u, v) is the quantization factor from the quantization table.

An 8 x 8 table of quantization factors is used for each output term. That is, each of the 64 positions of the DCT output block has its own "quantization coefficient". From the above formula, it could be seen that, the larger the quantization coefficient (Q(u,v)), the more data is lost. JPEG exploits the eye's differing sensitivity to the luminance and chrominance with the chrominance data being quantized more heavily than the luminance data. However, because most of the picture is categorised in the high-frequency and low-amplitude range, most of the "loss" occurs among subtle shifts that are largely indiscernible to the human eye. This concentrates the bulk of compressed file information on low-frequency and high-amplitude changes such as edges and corners (Ozer, 1995).

Since the adjacent DC components tend to be similar, each DC component is stored as the difference from the DC component in the preceding block. After the quantization, the values are further compressed through Run-Length Encoding using a special zig-zag pattern (figure 4.5) designed to optimise the compression of like regions within the image. At extremely high compression ratios, more high-frequency and low-amplitude changes get averaged, which can cause an entire pixel block to adopt the same color. This causes a blockiness that is characteristic of JPEG compressed images and is one of the disadvantages of JPEG image compression standard.

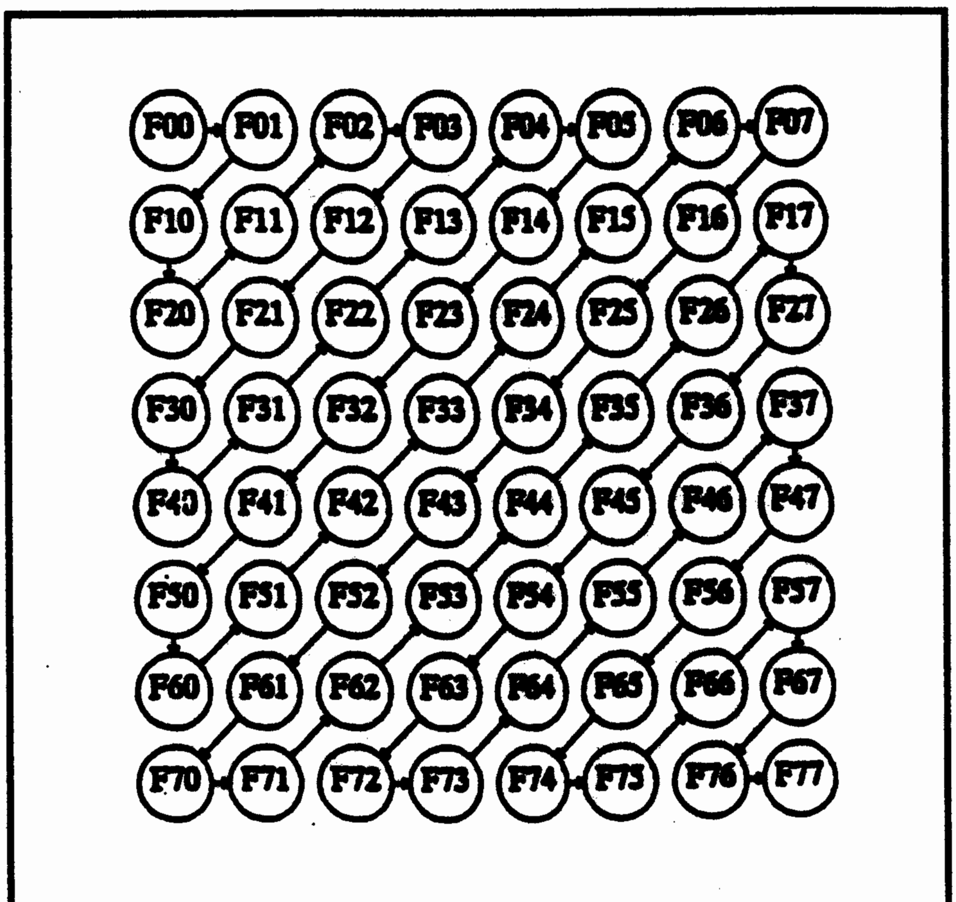


FIG 4.3: A zig-zag order from DCT output array

4.3.4.1 Quality factor

Within the quantization process a user-specified scaling factor, the "quality factor", governs the degree of removal of spatial frequency data and therefore lossiness. The JPEG compression algorithm uses a built-in quantization table that is appropriate for a medium to high quality setting. The quality factor allows the JPEG user to trade off compressed file size against the quality of the reconstructed image: the higher the quality factor, the larger the JPEG file, and the closer the output image will to the original output.

The quality factor is implementation specific and refers to a scaling factor for the quantization table used. This factors define an array of values that determine the quantization step size. In the Independent JPEG Group software, the quality factor ranges from 0 to 100. A factor of 0 produces the smallest, worst quality images; a factor of 100 produces the largest, best quality images. There is one quantization table in the Independent JPEG Group software. The user can override the default and specify his or her own. There are 64 elements of this table, each one being a quantization interval for a different cosine spatial frequency.

To alter the quality factor, each member of the table is scaled up or down. A higher Quality factor image results by making each quantization formula smaller and a lower quality image by making each interval larger. The formula (i.e., (200-2*QF)%) converts the quality factor between 0 to 100 into a scaling factor for the quantization table. QF represents the quality factor. For example, the scaling factor for the quality factor of 75 is (200-2*75) = 50%. This means that each member of the default quantization table is multiplied by 0.5. This table is recorded in the compressed file so that the decompression algorithm will know how to approximately reconstruct the DCT coefficients. Figure 4.4 shows an example of the JPEG quantization table (Q(u, v)). The optimal quality factor depends on the image content and is therefore different for every image.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

FIG 4.4: The JPEG Quantization Table (Q(u, v))

4.3.5 Coding the Reduced Coefficients

The final step in the baseline JPEG compression process is the process whereby the resulting coefficients which contain a significant amount of redundant data are encoded. JPEG uses either a modified Huffman or arithmetic coding which losslessly removes the redundancies to make JPEG data smaller. Huffman encodes the DC components as a 4-bit length followed by a signed integer of the given length. The AC components are stored as an 8-bit token followed by a variable length integer.

Due to patent restrictions, the baseline system uses Huffman coding while the arithmetic coding is an optional extension (Kay & Levine, 1995). JPEG's relatively simple mechanics make it extremely fast. It is also a symmetric algorithm, meaning that decompression is exactly the opposite process of compression and it occurs just as fast: Huffman decoding, Dequantization, Inverse DCT, Resampling and YUV to RGB-transform.

4.4 Progressive modes of operation

The progressive mode of JPEG consist of two techniques: spectral selection and successive approximation techniques. Spectral selection technique sends the lower frequency components of each 8 x 8 block first, then adds detail by sending some missing higher frequency components. When all the coefficients have been sent, the final image results. The successive approximation technique sends the approximate value of the DCT coefficients by truncating their least significant bits in the first stage. The missing low order magnitude bits are sent in the latter stages, one bit plane at a time. Hybrid algorithms employing a combination of both techniques are allowed(Kay & Lavine, 1995). Progressive encoding is useful in applications such as image database browsing.

4.5 Hierarchical mode of operation

In the hierarchical mode, an image is stored at several increasing resolutions. Each image is stored as the difference from the preceding smaller version and the processing is added ahead of the DCT encoder to filter and subsample the source image before encoding. Subsampling and encoding are done repeatedly, with progressively less subsampling to transmit images of increasing resolution one after the other (Luther, 1994). Hierarchical encoding is useful in applications in which a very high resolution image must be accessed by a lower-resolution display. An example is an image scanned and compressed at a high resolution for a very high -quality printer, where the image must also be displayed on a low-resolution PC video screen (Wallace, 1991).

4.6 Independent function for lossless mode of operation

A lossless compression which is a separate and independent function (not associated with DCT system) is specified for JPEG users who requires exact bit-for-bit reconstruction of their images. Lossless compression cannot achieve high compression ratios like the lossy technique, but lossless compression may be an end application requirement where pixel data is qualitatively important.

For example, medical images usually are not allowed to be lossy compressed. Other images where the pixel data is quantitatively important may be scientific data, remote sensed or computer-generated visualisation images. The lossless mode uses a spatial algorithm which is based on Differential Pulse Code Modulation coding model (DPCM). The DCT phase is not used; nor are the subsampling and the color space conversion as these would introduce loss (Kay & Lavine, 1995).

4.7 Is JPEG an ideal compression solution ?

The fact that JPEG is a lossy technique and works only on a select type of image data the question "why then bother to use JPEG ? ". This question had been looked into in the introductory section of this chapter. The importance of JPEG and applications have been discussed in terms of storage space and transmission time. This section deals with the above question by looking in the strength and weakness of JPEG compression standards. JPEG compression standard offers the following advantages and strengths:

- JPEG is an excellent way to store 24-bit color images of natural scenes. In contrast, GIF only supports 8-bit images;
- The compressed image size and image quality tradeoff can be user determined;
- It is ideally suited to images of real world scenes, or computer generated images which are complex.; and;
- It is platform independent for storing 24 bit images. It is currently the most widely adhered to standard, with the algorithm, source code implementations and public domain viewers readily available.

JPEG compression standard, however, do have the following drawbacks:

- JPEG compression is a tradeoff between degree of compression, resultant image quality and time required for compression and decompression. Blockiness results at high image compression ratios;

- JPEG does not fit every compression need. Images containing large areas of a single color (Animations, ray tracing, line art, black-and-white images or documents, and typical vector graphics) do not compress well. Infact, JPEG introduces "artifacts" or "blockiness" into such images that are visible against a flat background, making them considerably worse in appearance than if you used a conventional lossless compression method (Murray & VanRyper, 1994); and

- Baseline JPEG is not resolution independent. It does not provide for scalability, where the image is displayed optimally depending on the resolution of the viewing device.

4.8 Summary

The goal of JPEG is to develop an image compression method for still-frame, continuous tone image (both for color, and grey-scale). The JPEG standard contains four modes of operations: sequential encoding, progressive encoding, lossless encoding, and hierarchical encoding. The sequential and progressive modes refer to the order in which the compressed information is sent from the encoder to the decoder and subsequently displayed. They are based on the Discrete Cosine Transform and both of them are lossy in nature.

In sequential mode, each image component is handled in a single left to right across the rows, and top to bottom down the columns. With the same decoder in progressive mode, the image component is handled in multiple scans. The image progressively improves in quality until it is indistinguishable from the sequentially-generated image. A separate, independent JPEG function is specified by JPEG for lossless compression. It is based on the DPCM coding model.

The hierarchical process method encodes the image at a multiple spatial resolutions using either the DCT-based compression or the lossless mode. The baseline encoding method of JPEG contains the following processing steps: Colorspace conversion, Subsampling, Discrete Cosine Transform, Quantization, Huffman encoding, and Decompression. Although not strictly part of JPEG compression, most JPEG implementations start by converting the RGB image into luminance - chrominance color space. The image data may be subsampled, combing adjacent pixels into a single value. Then a Discrete Cosine Transform is applied to convert the image data into spatial frequency information. Quantization truncates the results of the DCT coding to a smaller range of values. Finally, the results of the quantization are compressed using either Huffman or arithmetic coding to produce the final output.

Chapter 5

Geometric Effect of JPEG(DCT-based) Image Compression

5.1 Introduction

The previous chapters gave an overview of the image compression techniques and in particular, the JPEG image compression algorithm. There has already been some mention in the previous chapter about lossless and lossy compression. After compressing an image with a lossless scheme, the reconstructed image is identical to the original image on a pixel to pixel basis. Lossy schemes cause degradation of the reconstructed image in exchange for larger compression ratios. These degradations of the reconstructed image lead to radiometric and geometric distortions and the question, of course, is what consequences such compression has on the geometric quality of images as it may be relevant to industrial photogrammetric applications.

Some of the artefacts and loss of data which do not detract from visual examination of the images, or their printed appearance, may be objectionable when images are to be measured. Changes in the sharpness or location of edges, the contrast across boundaries, or the absolute brightness of regions might have serious effects on image analysis procedures in industrial photogrammetry. Much is known about the visual impact of JPEG compression, but there are still only a limited number of published accounts in the photogrammetric literature of the impact of JPEG on the geometric quality of digital images.

This chapter therefore quantitatively investigates the geometric degradations of JPEG (DCT-based) compression at various levels of quality factors. It illustrates the specific defects with practical results that JPEG (DCT-based) compression can introduce on digital images used in industrial photogrammetry.

Radiometric distortions were not investigated because geometric distortions are most important for softcopy photogrammetry as they change the locations of the image points and, consequently, influence the accuracy of objects reconstructed by photogrammetric procedures (Novak & Shahin, 1996). The section below is the review of the previous investigative on geometric quality evaluation of JPEG compression.

5.2 Review of the previous experimental work on JPEG (DCT-based)

Mikhail *et al* (1994) studied the geometric effect of a Discrete Cosine Transform (DCT) image compression. In their work, aerial digital images with artificial targets in the form of crosses of different dimensions and orientations was used. It was found that DCT-based compression moved the measured cross targets as much as 0.5 pixels at compression ratios of about 1:16 on 8-bit images. Based on the finding they concluded that, geometric accuracy is progressively degraded as the original image is subjected to increasing compression. The cross location is shifted by increasing amounts as the number of bits is decreased in the coding scheme.

Jaakola and Orava (1994) investigated the effects of lossy image compression on the metric quality of scanned aerial images. The behaviour of different point- and area-like objects were examined by using different measuring method such as semiautomatic and automatic measuring of digital images, and manual measuring on analog images. They concluded from their study that JPEG (DCT-based) image compression does not have a significant effect on the geometric quality when compression ratios of 10:1 or less are used for measurements such as point location and area calculation.

Lamini and Sarjakoski (1995) also address the effects of JPEG image compression on color images by using an aerial color diapositive scanned image. Homogenous linear features were measured using manual pointing and the root mean square error for perpendicular differences between the end points of the linear features were calculated.

They found that baseline JPEG image compression does not have a geometric effect on the image geometry when compression ratios of up to 1:10 are used. In other words, they found out that the pointing precision was not affected by compression of up to 1:10. But some geometric degradation effect may occur when higher compression ratio is used.

Further, Novak and Shahin (1996) tested the geometric quality of baseline JPEG. The JPEG algorithm was tried out on images containing various levels of entropy. A scanned aerial image, an image captured by a digital camera, and an image captured during the test-field calibration of a digital camera were used for their investigations. They reported that geometric distortions of aerial images compressed at 5.9:1, 12.5:1 and 42:3 are in the order of 0.01 of a pixel. They conclude that larger compression ratios may distort the images considerably and hence are not recommended for aerial softcopy photogrammetry.

The research work of Robinson *et al* (1995) investigates the effects of JPEG image compression on automated digital terrain model extraction via the approach of feature-based matching using aerial photographs. Their study indicates that lossy compression with JPEG at a ratio of 8:1 degrades the accuracy of terrain models derived from the compressed images by about 10% for a pixel size of 15 microns.

5.3 Experimental Design and Results

The goal of this thesis is to quantitatively evaluate the geometric effects of baseline JPEG image compression on grey scale images and to determine the optimal quality factor that can produce an image which is visible accepted and with minimum degradation in quality. The test was based on the assumption that baseline JPEG image compression does not affect image geometry if small compression ratios are used. This assumption too is based on the review of experimental work by most researchers discussed in section 4.2. Notwithstanding, the investigation focuses on what happens when the amount of compression becomes very large.

The test images with retroreflective targets (figures 4.1(a) to 4.1(c)) were created to evaluate the effects of JPEG (DCT-based) compression on geometric image quality.

5.3.1 Test Images and Target Considerations

The test images consist of features and lines whose boundaries are sharply defined. For industrial photogrammetric purposes, retroreflective circular targets are commonly used as object points for the analysis. The object points co-ordinates refer to centre of the circular targets. The target size on the image is controlled by the physical size of the target and the scale of the imagery. Circles have the property of becoming ellipses under a perspective transformation and thus circular features (circles and ellipses) can be identified by their elliptical shape on an image. Retroreflective circular targets, if properly lit, provide a high contrast with the background, which eases the detection task and improves the centring precision (Vander Der Vlugt, 1995). The following set of images that were used in the compression and the decompression processes are grouped into three categories according to the image capture procedure:

- **Scanned Image:** Figure5.1(a) shows 8-bit scanned images. The images, *new* and *paw* were scanned for calibration and measurements. The sizes of these images are (*new* and *paw*) 384 rows by 256 columns and 1536 rows and 1024 columns respectively. The *mbh13* is a scanned image of an original medium format transparent with 5000 columns and 4730 rows. The number of bytes to store the original image 23 650 057.

- **Image capture by CCD Video Camera** (figure 5.1 (b)): These images (i.e. *left image*, *prop image*, *rock image* and *frame image*) are 8-bit images captured during the test-field calibration of the CCD Video camera. The sizes of these images are 512 rows by 512 columns, and the number of bytes to store each of the original image is 262 144. The images *left* and *frame* depict a calibration frame with targets. The images *rock* and *prop* are images with a rock and a propeller respectively mounted on calibration frames.

- **Image captured by DSC 420 Digital Camera** (figure 5.1(c)): There are two 8-bit target image captured DSC 420 digital camera for calibration and measurement purposes. The sizes of these images are 1012 rows by 1524 columns, and the number of bytes to the original of each image is 15 724 864. The *door* image is a car door which is mounted on a frame whereas *cal1* is special calibrated frame.

The main concern related to the use of JPEG (DCT-based) compression on these test images is the possibility of introducing artifacts (or false edge effects) as a consequence of the 8 x 8 pixel blocking strategy employed. This effect has the potential of adversely affecting the positions of targets in the digital images for photogrammetric measurements and analysis.

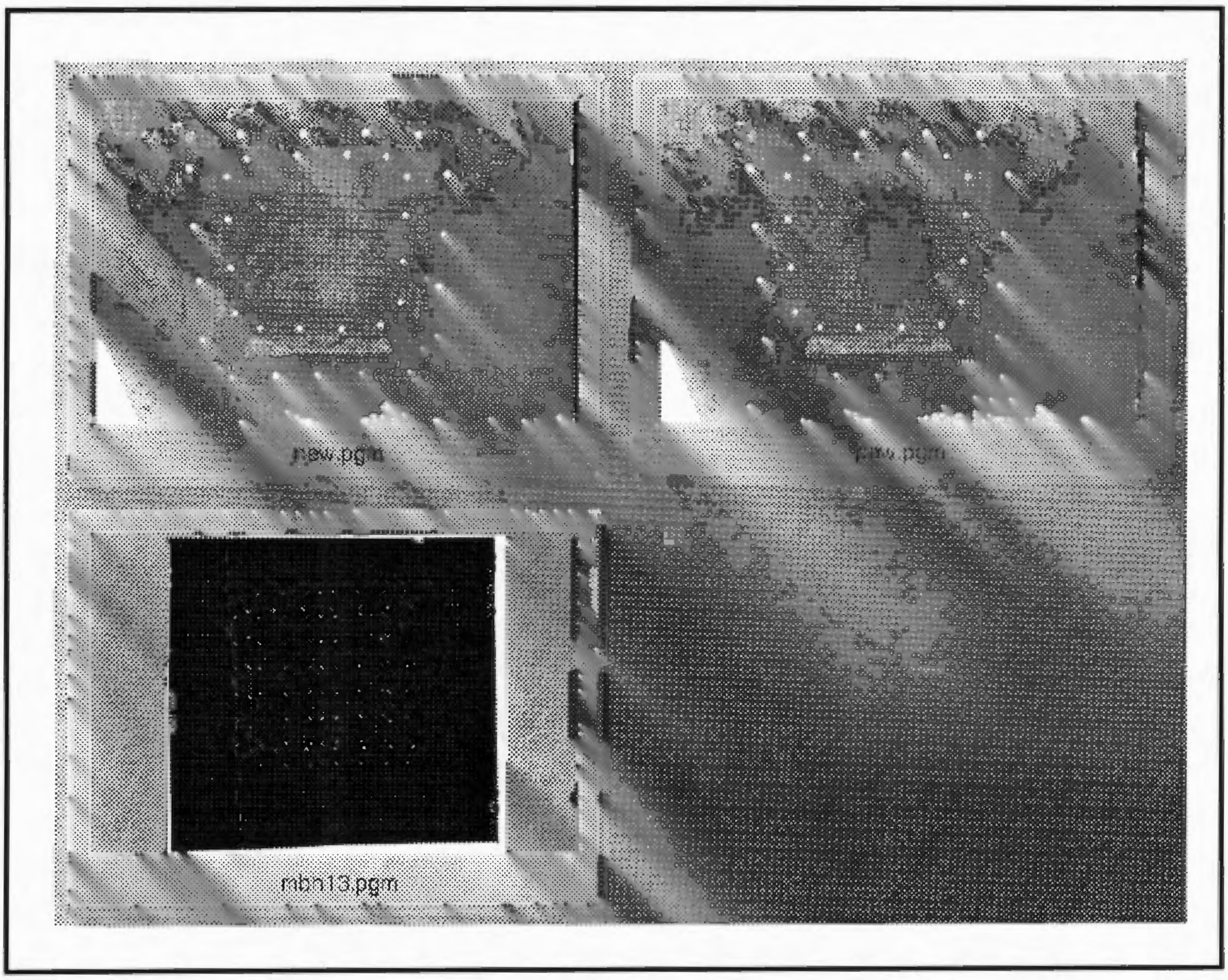


FIG 5.1(a): Scanned Images

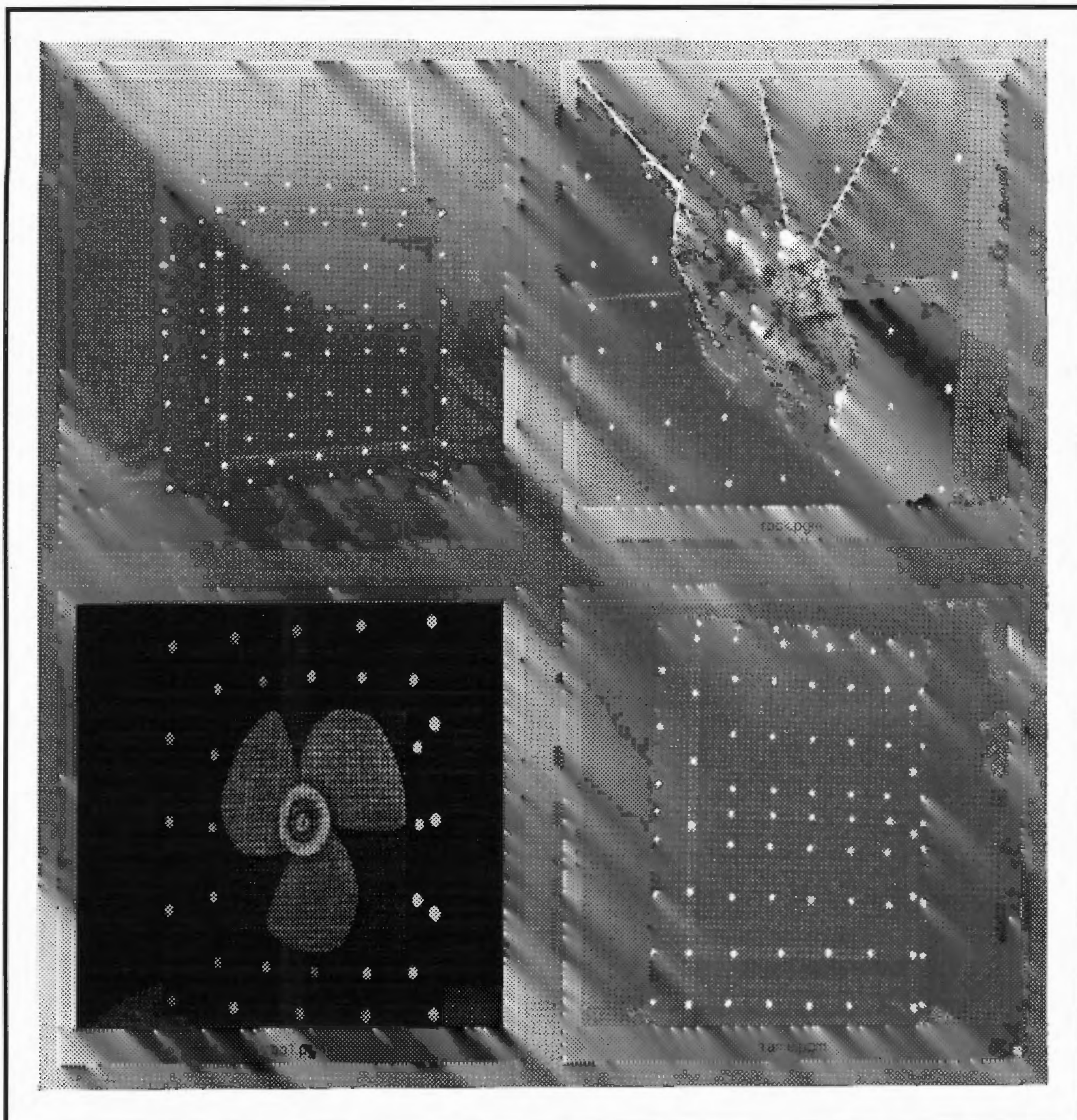


FIG 5.1(b): CCD Video Camera Images

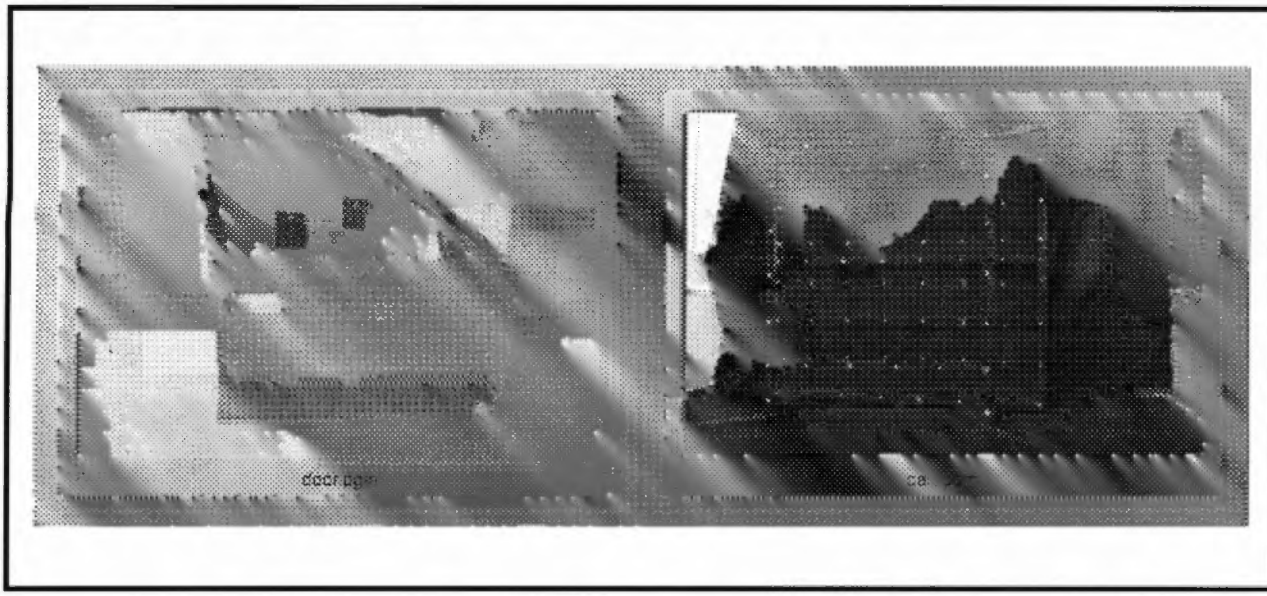


FIG 5.1(c): DCS 420 Digital Camera Images

5.3.2 Compression Method

The compression and decompression of the test images were performed using software implementation of the JPEG algorithm on a UNIX based Personal Computer (PC). It is Independent JPEG Group's JPEG software. The test images described in section 5.3.1 were compressed using JPEG (DCT-based) algorithm at different quantization levels (i.e., between 30 and 95 quality factors). Since one of the objectives of the research is to investigate the effects of JPEG lossy compression on digital images used in softcopy photogrammetry, the JPEG algorithm was used on images containing various levels of entropy (a measure of information content of an image) with different image capture capabilities. The influence of the image contents on JPEG (DCT-based) are shown in Table 5.1 and Figure 5.4. Typical quality factors for photogrammetric and image analysis applications are in the 50 to 100 range although the range extends between 1 and 100 (Huffman, 1994). The numeric value of the quality setting are different in the various implementations. These results clearly show that a large amount of compression can be achieved when using lossy compression schemes.

5.3.3 Semi-Automatic Extraction of Circular Target Points

This section explains the procedure and algorithm (discuss in the next section) used to extract the target points in the test digital images. After compression and decompression of the test images, the centres of targets in the images were detected by using a target location and identification program (i.e., search program). This in-house program which is semi-automatic uses grey level thresholding to search a digital image to locate the positions and perimeters of bright targets and find their centres. Figure 5.1 shows a digital representation of a bright target in a digital image. Target centre detection is the process in which the such targets centres are automatically detected and their position on the image is determined.

The target centre detection process is based on the identification of distinguishing properties of targets images such as grey level differences between targets and the background, target size or target shape. Figure 5.2 is an example of a target window in the digital imagery. The input to this program is a digital image and the output is a list of targets positions.

80	80	80	81	88	84	83	81	80	80	80	80
80	80	81	85	95	108	116	110	98	80	80	80
80	81	87	110	151	187	202	189	150	106	85	80
80	85	111	149	228	250	253	251	223	151	97	81
81	96	131	228	254	255	255	255	251	193	112	83
98	111	155	255	255	255	255	255	252	190	112	85
85	99	142	251	255	255	255	255	249	189	109	83
84	89	133	230	255	255	255	255	223	147	89	83
84	86	90	150	200	250	254	249	220	111	84	81
80	83	85	101	118	150	150	122	101	89	82	82
80	83	84	96	101	121	130	126	111	86	83	82
81	83	83	83	84	84	84	83	83	83	83	82

FIG 5.2: Digital representation of target in digital image

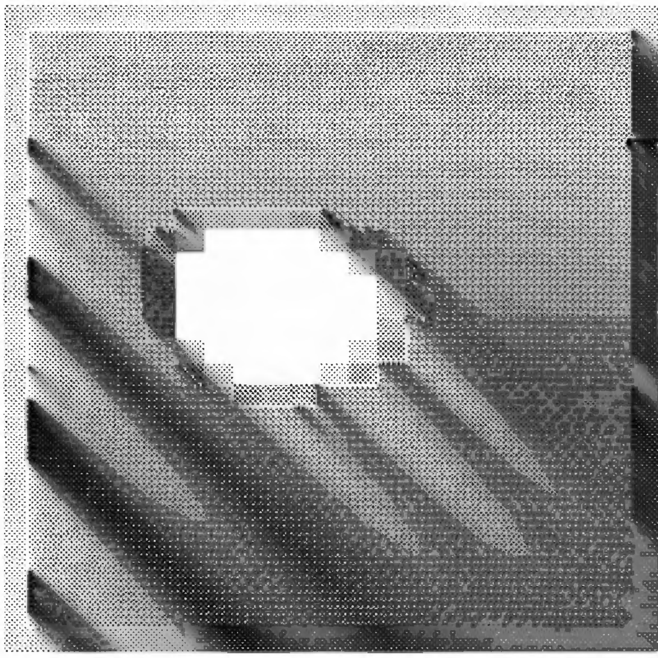


FIG 5.3: Plan View of a circular Target Window

5.3.3.1 Target Detecting and Centring Algorithm

Firstly an in-house program was used to extract the centre of the target to subpixel accuracy in a grey level digital imagery. Target detecting and centring algorithm is used to determine the precise location of target in each of the nine resulting image files. The user is prompted for the interactive selection of an appropriate grey-level threshold level at which to extract the target locality. Target centres are calculated using a weighted centre-of-gravity of the pixel grey levels in a window defined around the target by a thresholding process. The equations for the centre of gravity using the grey level value as weight are:

$$x_0 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} g^2_{ij} x_i}{\sum_{i=1}^{n}\sum_{j=1}^{m} g^2_{ij}}$$

$$y_e = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} g^2{}_{ij} y_j}{\sum_{i=1}^{n}\sum_{j=1}^{m} g^2{}_{ij}}$$

Where g_{ij} is the grey value located in row i and column j of the digital image.

Using the grey value squared as weight, as opposed to grey value, reduces the influence of the window position on the computed centre co-ordinates. However it is more sensitive to noisy pixels. The grey scale distribution has its influence on accuracy when the grey scale distribution in not uniform. That is this method results in accuracy being dependent on the image quality. Improvement in accuracy can be obtained by only including those pixels with grey level values greater than the background level as observations. This removes the effects of window position if the whole target is within this window. The accuracy of this algorithm is discussed in chapter 6.

5.4 Performance Evaluation for JPEG Image Compression

Image compression decreases the amount of image data, but depending on the technique used, the image may be degraded. For the purpose of investigating geometric distortions of the reconstructed images, 100 to 200 points were defined in the original images. The positions of these points in the original images were pre-determined before the compression and decompression processes. In the reconstructed images their positions were found using in-house software already described in section 5.3 of this chapter. The evaluation of algorithmic performance of JPEG image compression is based on measuring the:

- compression ratio which quantitatively measures of the amount of data reduction expressed in terms of memory bits per image or bits per pixel; and;

- root mean square (rms) error which gives a quantitative assessment of the geometric degradation (if any) of the image data.

Each of these quality measures is discussed and resulting experimental results are presented in the subsequent sections.

5.4.1 Compression Ratios

The compression ratio is defined as the ratio of the number of bytes of the original image before compression and the number of bytes of the compressed image. Compression ratio can serve the general purpose of measuring how efficiently the images can be compressed according to the needs of the user. The compression ratios for the test images were calculated using both JPEG(DCT-based) and LZW compression techniques.

Table 5.1 shows the resulting compression ratios for the nine image set. As it can be seen, the range of values in this table represent the compression rate with degree of quality factors ranging from 95 to 40. For example, a value of 20 for quality factor of 90 for *left* image represents the compression ratio achieved by JPEG (DCT-based) lossy compression for the *left* image. This can be written as 20:1 which therefore means that the *left* image with original size of 262 144 for storage will be compressed to the size of 12 932 (i.e., one twentieth its original size). Comparisons are made between the JPEG (DCT-based) compression and LZW lossless compression in TIFF file format. With the exception of the scanned photographs (i.e., new, paw, and mbh13) the compression rate for JPEG quality factor of 90 or less was greater than lossless LZW. These data are plotted in figure 5.2 with compression rate versus quality factor. The results are interpreted and inferences are drawn in the next chapter.

	CCD VIDEO IMAGES				DCS-40 IMAGES		SCANNED IMAGES		
Quality Factor	**Left**	**Prop**	**Frame**	**Rock**	**Door**	**Call**	**New**	**Paw**	**Mbh13**
95	3.8	8.2	3.0	2.6	4.2	5.9	1.9	2.1	10.9
90	5.9	11.5	4.7	3.9	6.5	9.9	2.7	3.0	21.1
85	7.6	13.9	6.1	5.0	8.4	14.0	3.3	3.9	35.3
80	9.2	15.9	7.4	6.0	9.9	17.9	4.0	4.6	48.2
75	10.8	17.6	8.7	6.9	11.3	21.8	4.6	5.3	56.6
70	12.1	18.9	9.8	7.7	12.5	24.9	5.1	5.9	60.3
65	13.5	20.3	11.0	8.6	13.7	28.0	5.7	6.5	63.0
60	14.9	21.7	12.4	9.5	15.0	30.1	6.3	7.1	64.8
55	16.2	22.9	13.8	10.4	16.2	33.1	6.8	7.7	66.4
50	17.4	24.0	15.0	11.2	17.3	35.0	7.3	8.2	66.9
40	20.3	29.3	18.5	13.6	20.1	39.5	8.5	9.6	69.5
30	23.9	30.0	23.6	17.3	24.2	44.5	10.4	11.7	71.7
LZW	1.4	3.7	1.3	1.1	4.8	6.6	1.0	3.3	40.3

Table 5.1: Compression ratio for test data set for JPEG with varying quality factor and LZW lossless compression.

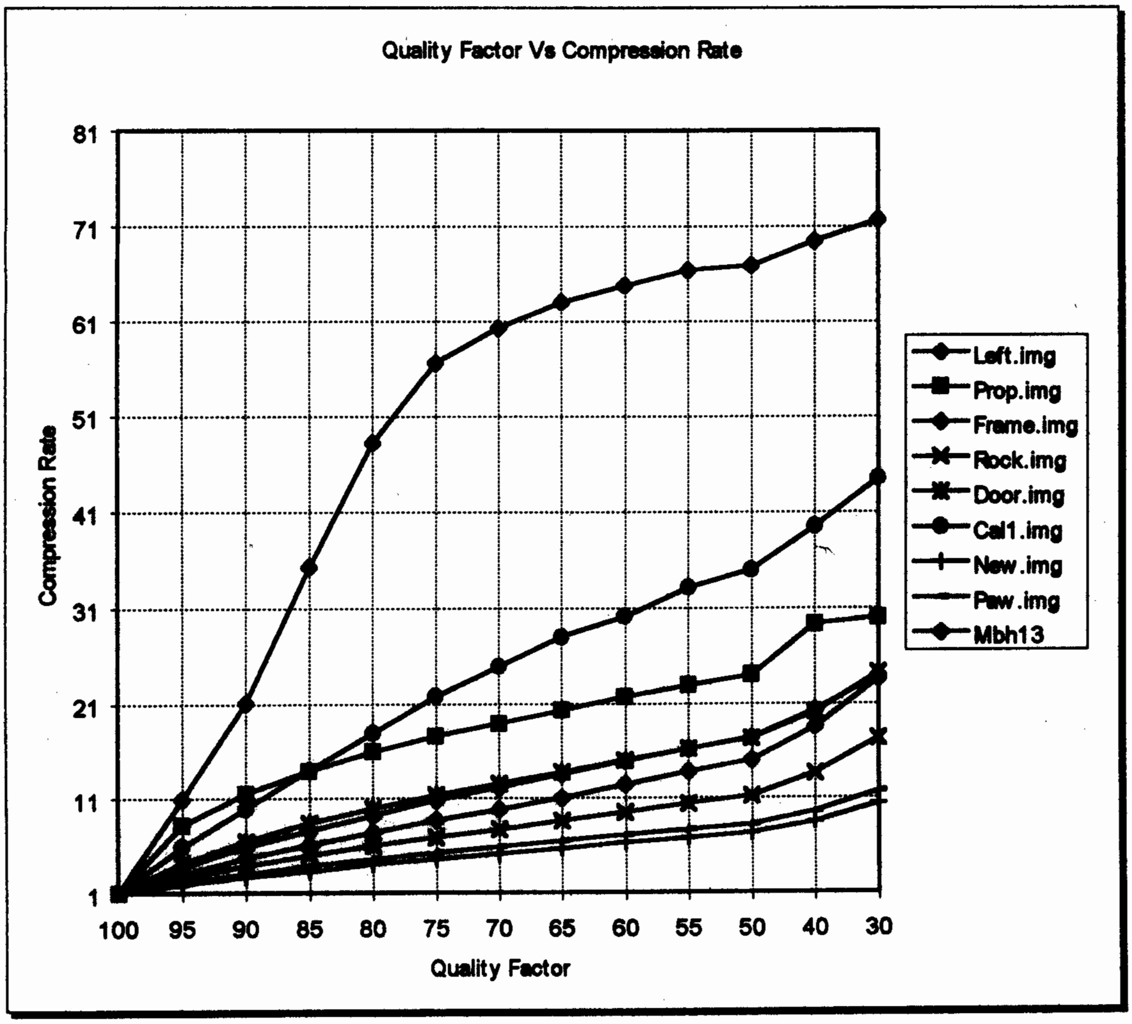


FIG 5.4 The influence of the image contents on JPEG (DCT-based) algorithm.

5.4.2 Root Mean Square (RMS) Error Between the Original and Reconstructed Targets

The distance in pixels between the original location of a target, and the reconstructed location for the same target is used to identify how much distortion in geometric terms JPEG (DCT-based) was introduce by the compression process. Evaluating the performance of the image compression algorithm is achieved through visual judgement of the reconstructed image.

Although visual judgement of the reconstructed image is probably the natural criterion for evaluating the quality of reconstructed image, such judgement is subjective and difficult to model and represent. Mathematically, we the rms error over all targets in an image:

$$\text{rms error} = \left(\frac{1}{N} \sum_{i=1}^{N} \left(X_i - X_i^1 \right)^2 \right)^{1/2}$$

where:

N = the number of pixel in the image

X_i = original image location

X_i^1 = reconstructed (decompressed) image location

Figure 5.5 illustrates the relationship between the residuals (shifts) and the error per target.

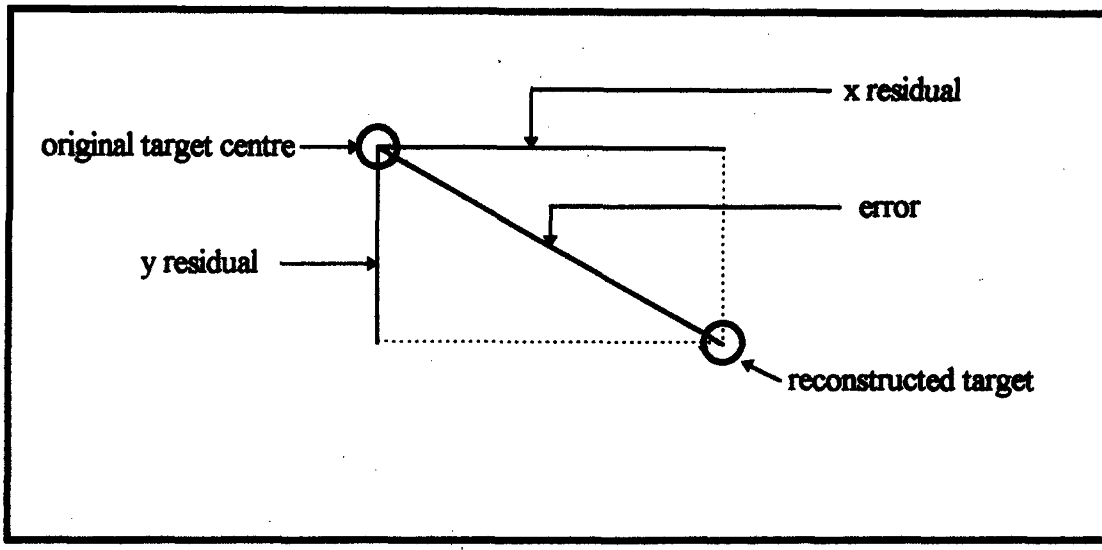


FIG 5.5: Residuals and error per target

Tables 5.2 to 5.10 represent the geometric distortions (in pixel) for different quantization levels of different images. The **Vx** and **Vy** columns are the maximum shift in x and y coordinates respectively. The ∇x and ∇ y columns are the root mean squares of the x and y coordinate residuals of entire data set in each image and ∇xy column is the root mean square residual occurring in both x and y coordinates. The rms errors of targets centres in each image are presented. For example the rms error obtained with *left* image using JPEG (DCT-based) compression under the quality factor of 75 is 0.05. This means that the centre of the reconstructed target pixel is on average 0.05 pixels away from the centre of the original target pixel target.

Under the same quality factor the amount of shift in x and y coordinates is 0.06 and 0.05 respectively. For purposes of illustration, the overall difference in pixel values across the whole image was computed. Three image data sets were chosen for the analysis and the results are summarised in tables 5.11 to 5.13. These results reflects the global change in the image due to JPEG(DCT-based) image compression. As can be seen, the trend for the JPEG(DCT-based) image compression is also apparent and consistent amongst the data sets.

On average, the algorithm produces the poorest results as the original image is subject to more compression. On the hand, figures 5.6 to 5.32 illustrate the relationships between the rms error (distortion) and quality factors, maximum shift in x and quality factor, and maximum shift in y and quality factor for grey scale images (i.e nine test images) using JPEG(DCT-based) compression. The results in tables 5.2 to 5.10 are discussed in chapter 6. Furthermore, the graphs of compression ratio versus rms error are plotted in appendix A. The next chapter further discusses and interprets the results.

Left image	**Maximum Shift**		**root mean square (rms) error**		
Quality Factor	**Vx**	**Vy**	**∇x**	**∇y**	**∇xy**
95	0.06	0.05	0.014	0.002	0.02
90	0.03	0.03	0.080	0.009	0.01
85	0.05	0.04	0.010	0.009	0.01
80	0.04	0.05	0.010	0.011	0.02
75	0.06	0.05	0.012	0.012	0.02
70	0.08	0.05	0.018	0 .012	0.02
65	0.07	0.08	0.018	0.016	0.02
60	0.08	0.08	0.018	0.016	0.02
55	0.09	0.07	0.021	0.017	0.03
50	0.10	0.09	0.019	0.019	0.03
40	0.12	0.10	0.025	0.023	0.04

Table 5.2 : Geometric distortions of the reconstructed image (i.e left image) different for varying JPEG quality factor.. (Units: Pixels)

Prop image	Maximum Shift		root mean square (rms) error		
Quality Factor	**Vx**	**Vy**	**∇x**	**∇y**	**∇xy**
95	0.12	0.14	0.033	0.039	0.05
90	0.06	0.07	0.012	0.014	0.02
85	0.04	0.09	0.012	0.018	0.02
80	0.07	0.07	0.017	0.019	0.02
75	0.08	0.13	0.018	0.031	0.03
70	0.06	0.10	0.016	0.026	0.03
65	0.10	0.13	0.025	0.024	0.03
60	0.10	0.07	0.026	0.020	0.03
55	0.10	0.17	0.022	0.038	0.04
50	0.14	0.16	0.034	0.036	0.04
40	0.11	0.19	0.024	0.058	0.06

Table 5.3 : Geometric distortions of the reconstructed image (i.e prop image) for varying JPEG quality factor. (Units: Pixels)

Frame Image	**Maximum Shift**		**root mean square (rmse) error**		
Quality Factor	**Vy**	**Vy**	**∇x**	**∇y**	**∇xy**
95	0.06	0.05	0.010	0.010	0.01
90	0.03	0.03	0.008	0.010	0.01
85	0.03	0.06	0.009	0.011	0.01
80	0.04	0.05	0.010	0.010	0.01
75	0.07	0.05	0.013	0.013	0.02
70	0.06	0.08	0.013	0.015	0.02
65	0.06	0.08	0.015	0.019	0.02
60	0.06	0.09	0.016	0.019	0.02
55	0.07	0.09	0.016	0.022	0.03
50	0.07	0.09	0.017	0.021	0.03
40	0.11	0.10	0.022	0.025	0.03

Table 5.4 : Geometric distortions of the reconstructed image (i.e prop image) for varying JPEG quality factor. (Units: Pixels)

Rock image	Maximum Shift		root mean square (rms) error		
Quality Factor	**Vx**	**Vy**	**∇x**	**∇y**	**∇xy**
95	0.06	0.07	0.020	0.017	0.02
90	0.04	0.04	0.011	0.009	0.01
85	0.02	0.03	0.006	0.010	0.01
80	0.02	0.03	0.008	0.008	0.01
75	0.03	0.03	0.010	0.010	0.01
70	0.02	0.03	0.007	0.010	0.01
65	0.04	0.04	0.013	0.010	0.01
60	0.06	0.05	0.015	0.017	0.02
55	0.06	0.04	0.012	0.017	0.02
50	0.04	0.05	0.013	0.017	0.02
40	0.04	0.05	0.013	0.015	0.02

Table 5.5 : Geometric distortions of the reconstructed image (i.e rock image) for varying JPEG quality factor. (Units: Pixels)

Door image	Maximum Shift		root mean square (rms) errors		
Quality Factor	Vx	Vy	∇x	∇x	∇xy
95	0.13	0.14	0.036	0.037	0.05
90	0.03	0.04	0.008	0.008	0.01
85	0.06	0.03	0.012	0.008	0.01
80	0.06	0.05	0.012	0.011	0.02
75	0.04	0.07	0.011	0.015	0.02
70	0.07	0.04	0.014	0.011	0.02
65	0.01	0.09	0.017	0.018	0.02
60	0.07	0.07	0.016	0.018	0.02
55	0.08	0.09	0.020	0.020	0.03
50	0.14	0.12	0.024	0.023	0.03
40	0.15	0.35	0.025	0.057	0.06

Table 5.6 : Geometric distortions of the reconstructed image (i.e door image) for varying JPEG quality factor. (Units: Pixels)

Call Image	Maximum Shift		root mean square (rms) error		
Quality Factor	Vx	Vy	∇x	∇x	∇xy
95	0.09	0.08	0.022	0.025	0.03
90	0.03	0.06	0.007	0.009	0.01
85	0.04	0.03	0.011	0.007	0.01
80	0.05	0.04	0.008	0.009	0.01
75	0.06	0.05	0.011	0.011	0.02
70	0.06	0.06	0.013	0.010	0.02
65	0.07	0.08	0.012	0.013	0.02
60	0.09	0.09	0.015	0.015	0.02
55	0.09	0.10	0.017	0.016	0.02
50	0.06	0.10	0.011	0.015	0.02
40	0.14	0.06	0.022	0.015	0.03

Table 5.7 : Geometric distortions of the reconstructed image (i.e call image) for varying JPEG quality factor. (Units: Pixels)

New Image Quality Factor	Maximum Shift		root mean square (rms) error		
	Vx	Vy	∇x	∇y	∇xy
95	0.34	0.50	0.069	0.120	0.14
90	0.38	0.35	0.091	0.087	0.13
85	0.37	0.34	0.089	0.090	0.13
80	0.40	0.40	0.087	0.091	0.13
75	0.58	0.52	0.122	0.124	0.17
70	0.48	0.52	0.105	0.140	0.18
65	0.34	0.65	0.096	0.178	0.20
60	0.45	0.68	0.109	0.179	0.21
55	0.73	0.57	0.178	0.166	0.24
50	0.83	0.78	0.149	0.199	0.25
40	0.34	0.87	0.090	0.197	0.22

Table 5.8 : Geometric distortions of the reconstructed image (i.e new image) for varying JPEG quality factor. (Units: Pixels)

Paw image	Maximum Shift		root mean square (rms) error		
Quality Factor	**Vx**	**Vy**	**∇x**	**∇y**	**∇xy**
95	0.28	0.43	0.077	0.106	0.11
90	0.31	0.35	0.090	0.110	0.14
85	0.41	0.27	0.091	0.068	0.11
80	0.32	0.27	0.079	0.270	0.28
75	0.46	0.34	0.122	0.093	0.15
70	0.32	0.43	0.091	0.105	0.14
65	0.39	0.32	0.085	0.71	0.11
60	0.44	0.31	0.110	0.087	0.14
55	0.31	0.54	0.086	0.143	0.17
50	0.39	0.41	0.105	0.107	0.15
40	0.60	0.50	0.164	0.140	0.22

Table 5.9 : Geometric distortions of the reconstructed image (i.e paw image) for varying JPEG quality factor. (Units: Pixels)

Mbh13 Image	Maximum Shift		root mean square (rms) error		
Quality Factor	Vx	Vy	∇y	∇y	∇xy
95	0.18	0.78	0.033	0.259	0.26
90	0.19	0.67	0.022	0.196	0.20
85	0.14	0.66	0.022	0.166	0.17
80	0.16	0.65	0.023	0.205	0.21
75	0.08	0.67	0.019	0.242	0.24
70	0.11	0.68	0.022	0.239	0.24
65	0.19	0.67	0.032	0.247	0.25
60	0.23	0.82	0.039	0.246	0.25
55	0.25	0.69	0.040	0.265	0.27
50	0.18	0.68	0.038	0.241	0.244
40	0.24	0.71	0.049	0.281	0.29

Table 5.10 : Geometric distortions of the reconstructed image (i.e mbh13 image) for varying JPEG quality factor (Units: Pixels)

Quality Factor	95	90	85	80	75	70	65	60	55	50	40
rms error (grey values)	1.8	1.8	2.2	2.4	2.5	2.6	2.7	2.9	2.9	3.2	3.4

Table 5.11: Global change in intensity values in the left image due to JPEG(DCT-based) at various quality factors.

Quality Factor	95	90	85	80	75	70	65	60	55	50	40
rms error (grey values)	1.3	1.3	1.4	1.5	1.6	1.6	1.7	1.8	1.8	1.8	2.0

Table 5.12: Global change in intensity values in the call image due to JPEG(DCT-based) at various quality factors.

Quality Factor	95	90	85	80	75	70	65	60	55	50	40
rms error (grey values)	2.3	3.5	4.7	5.6	6.3	6.9	7.3	7.7	8.0	8.3	8.9

Table 5.12: Global change in intensity values in the new image due to JPEG(DCT-based) at various quality factors.

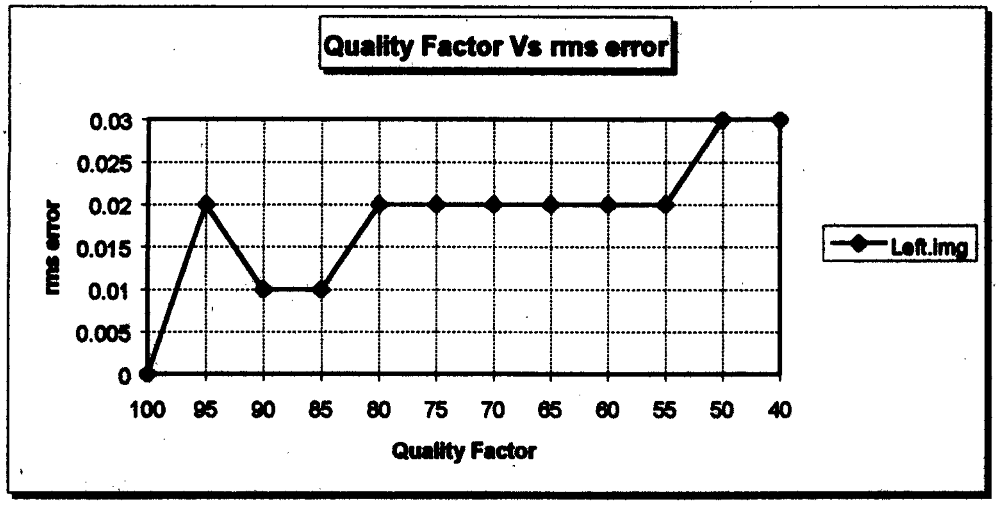


FIG5.6: Quality factor versus distortion for left image using JPEG (DCT-based) compression

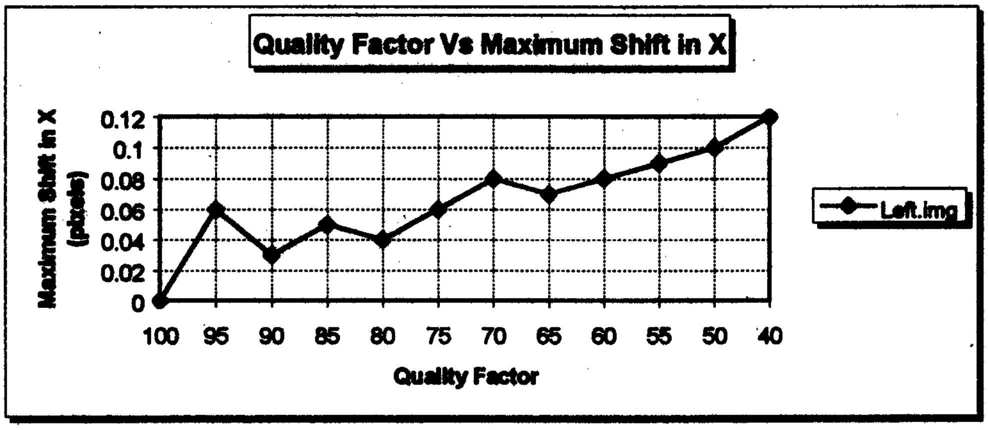


FIG5.7: Quality factor versus maximum shift in X for left image using JPEG (DCT-based) compression

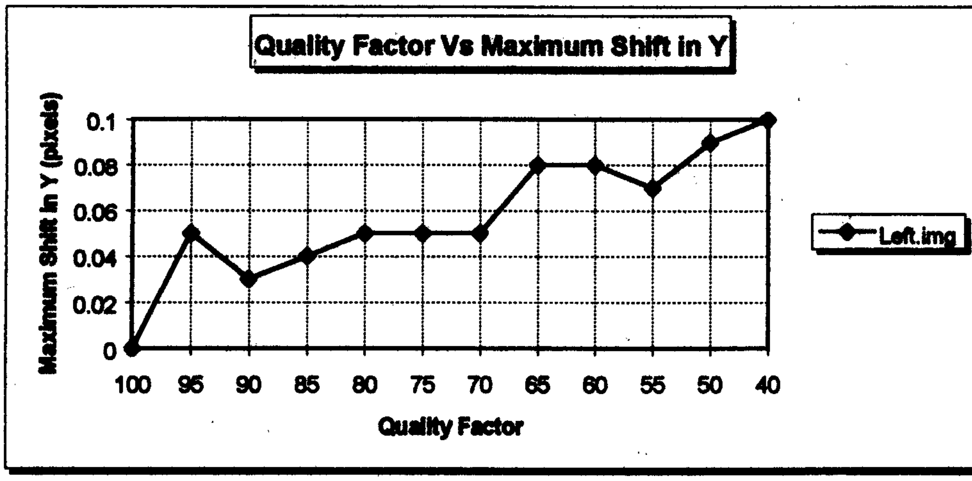


FIG5.8: Quality factor versus maximum shift in Y for left image using JPEG (DCT-based) compression

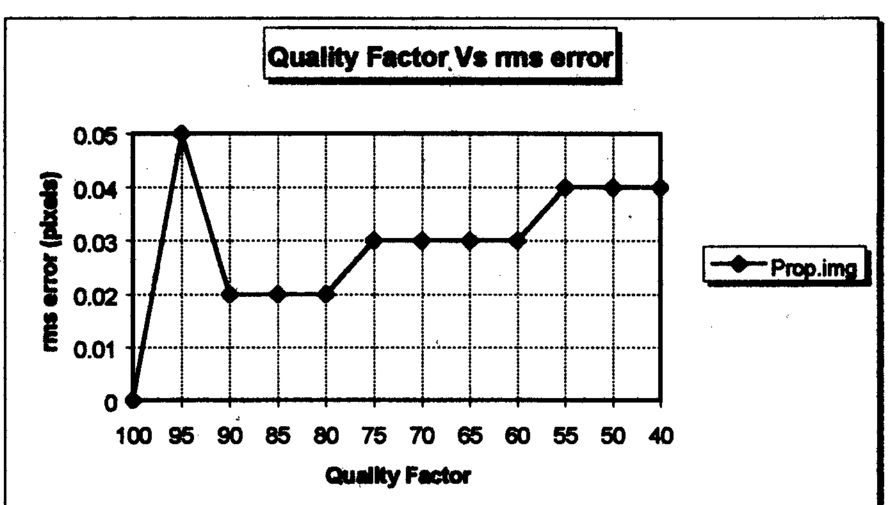


FIG 5.9 : Quality factor versus distortion for prop image using JPEG (DCT-based) compression

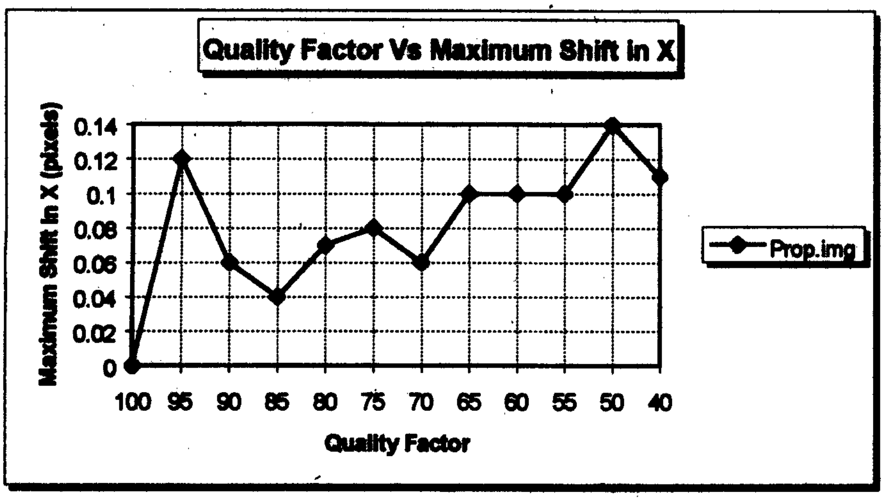


FIG 5.10: ***Quality factor versus maximum shift in X for prop image using JPEG (DCT-based) compression***

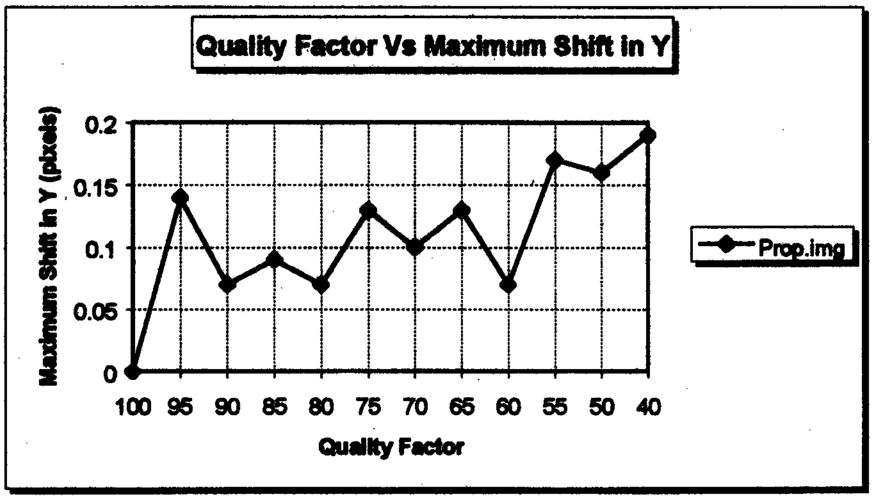


FIG 5.11: ***Quality factor versus maximum shift in Y for prop image using JPEG (DCT-based) Compression***

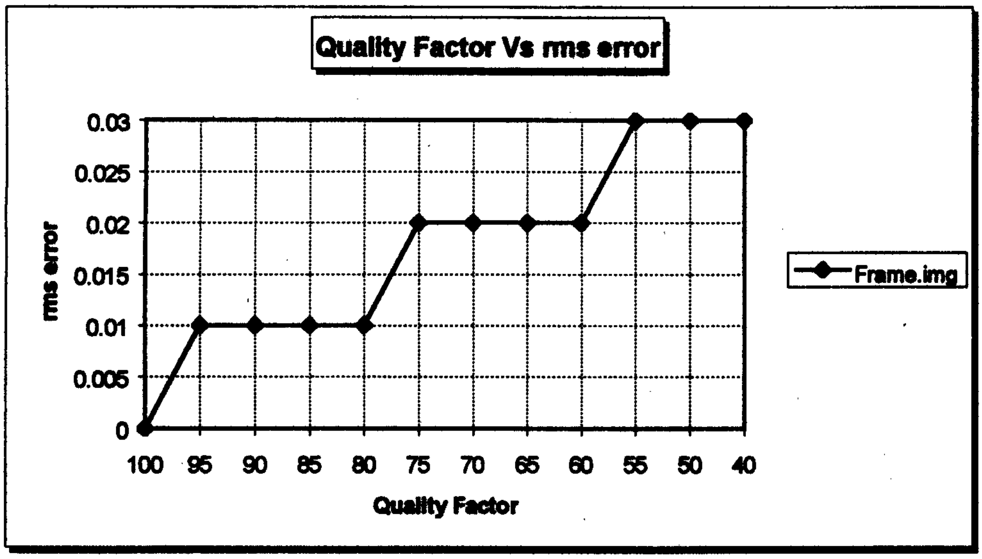


FIG 5.12: Quality Factor Versus Distortion for frame image using JPEG (DCT-based) Compression

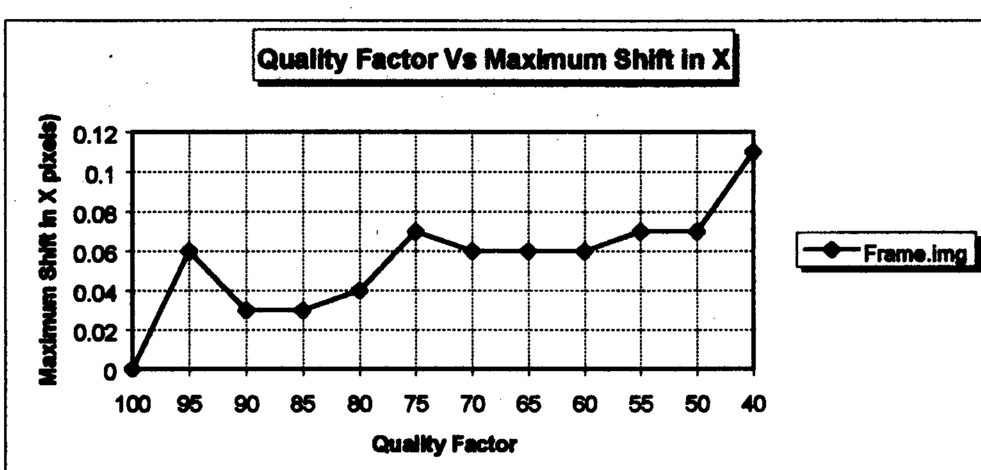


FIG 5.13: Quality Factor Versus Maximum Shift in X for frame image using JPEG (DCT-based) Compression

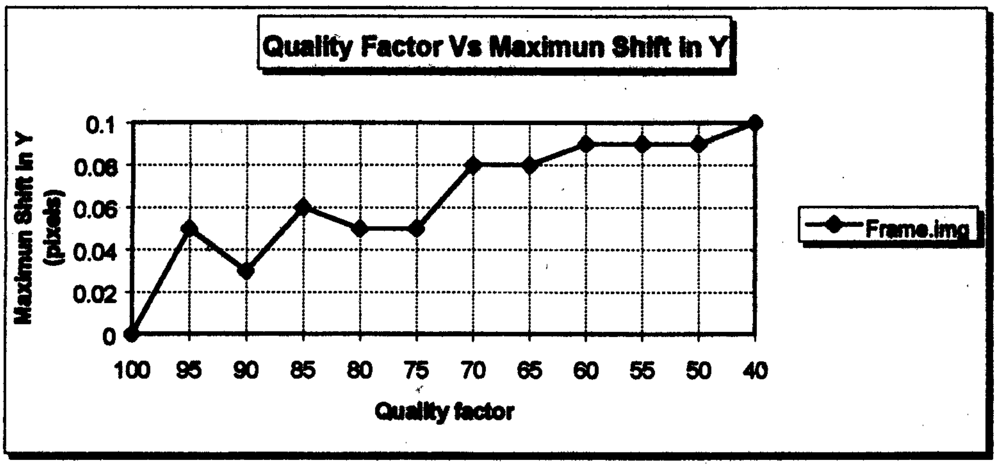


FIG 5.14: Quality Factor Versus Maximum Shift in Y for frame image using JPEG (DCT-based) Compression

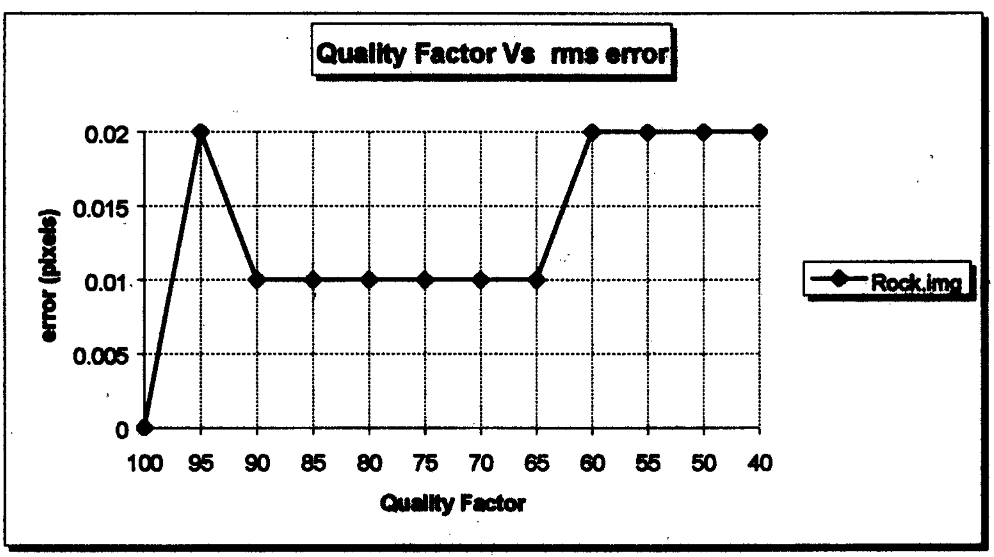


FIG 5.15: Quality Factor Versus Distortion for rock image using JPEG (DCT-based) Compression

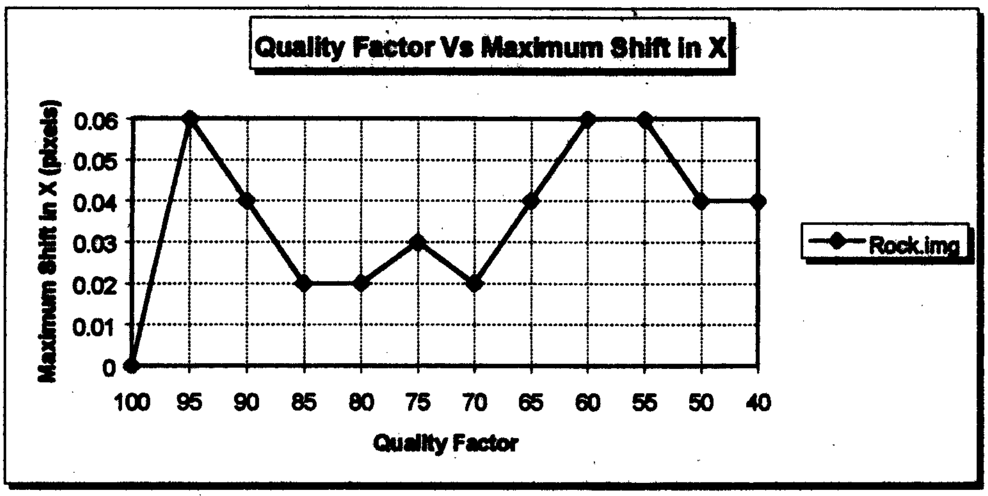


FIG 5.16: Quality Factor Versus Maximum Shift in X for rock image using JPEG (DCT-based) Compression

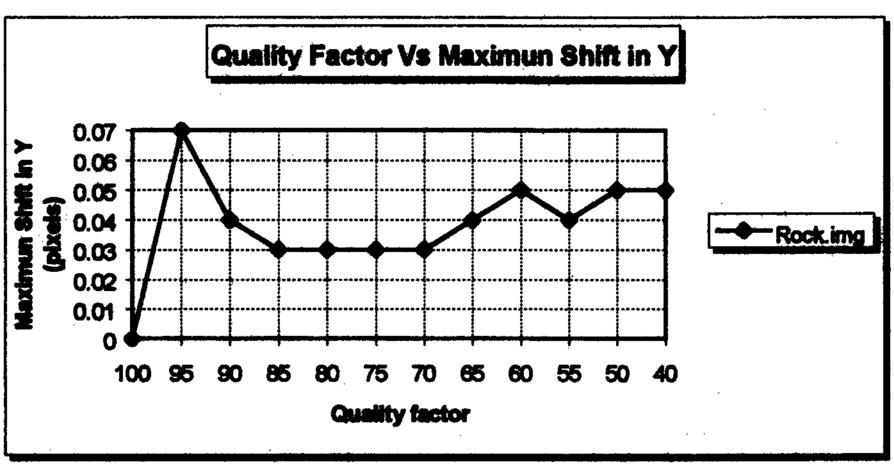


FIG 5.17: Quality Factor Versus Maximum Shift in Y for rock image using JPEG (DCT-based) Compression

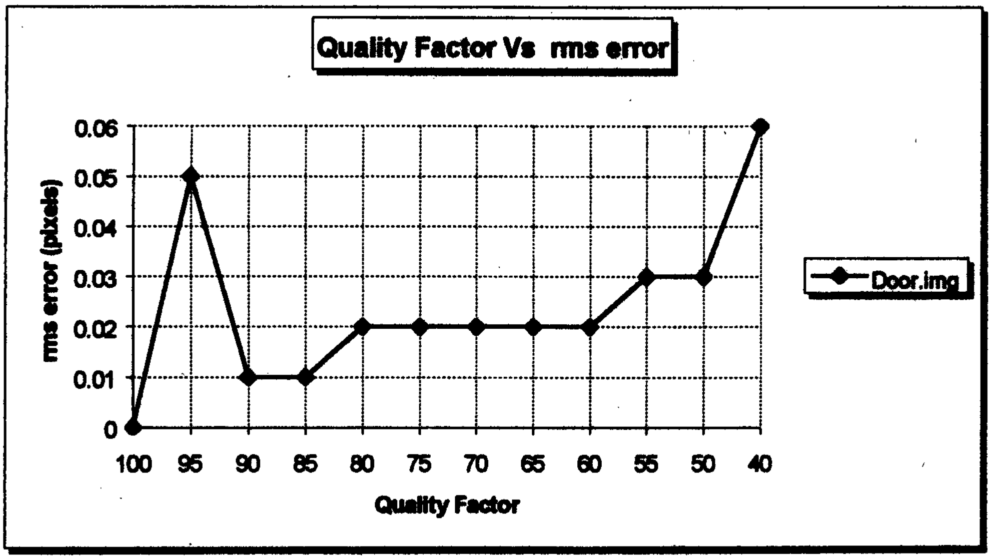


FIG 5.18: Quality Factor Versus Distortion for door image using JPEG (DCT-based) Compression

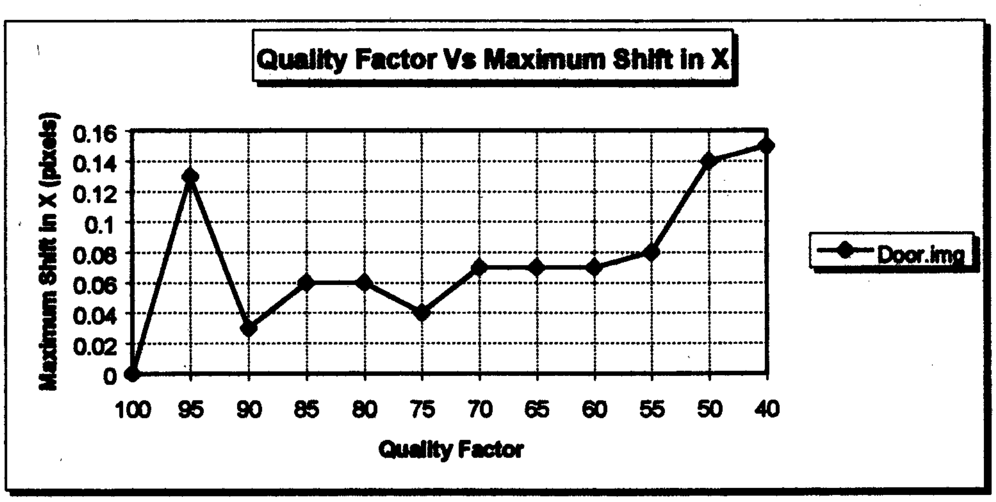


FIG 5.19: Quality Factor Versus Maximum Shift in X for door image using JPEG (DCT-based) Compression

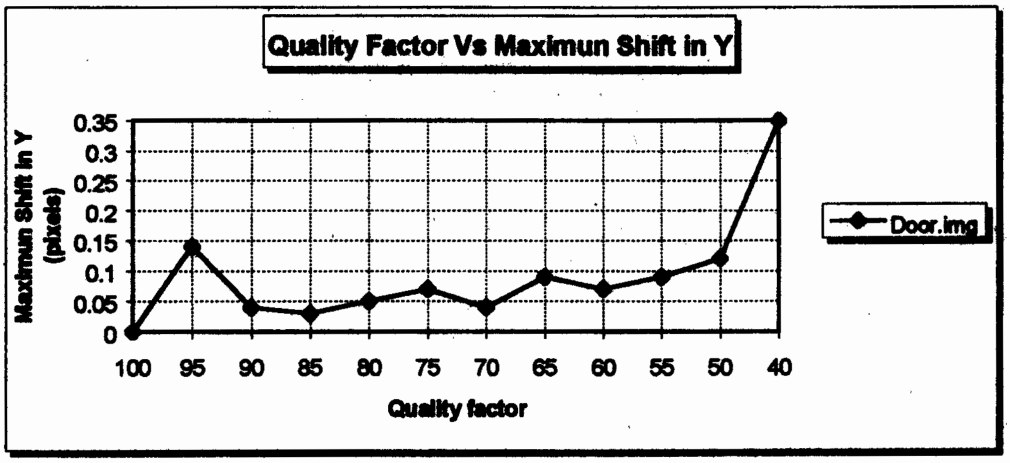


FIG 5.20: Quality Factor Versus Maximum Shift in Y for door image using JPEG (DCT-based) Compression

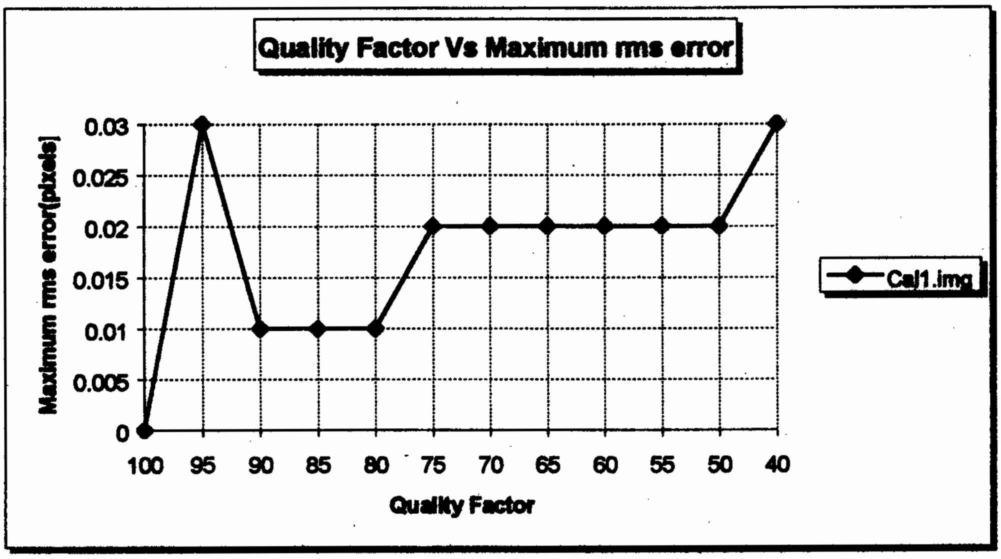


FIG 5.21: Quality Factor Versus Distortion for cal1 image using JPEG (DCT-based) Compression

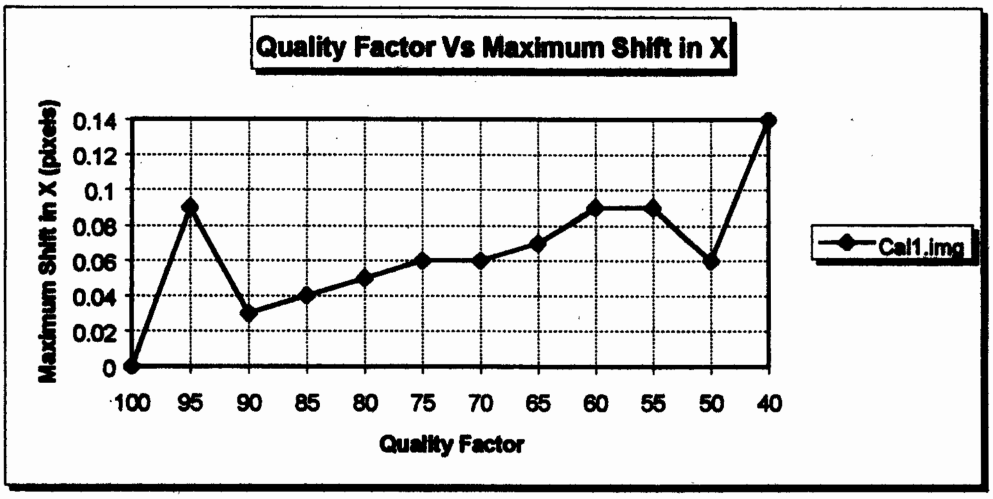


FIG 22: Quality Factor Versus Maximum Shift in X for call image using JPEG (DCT-based) Compression

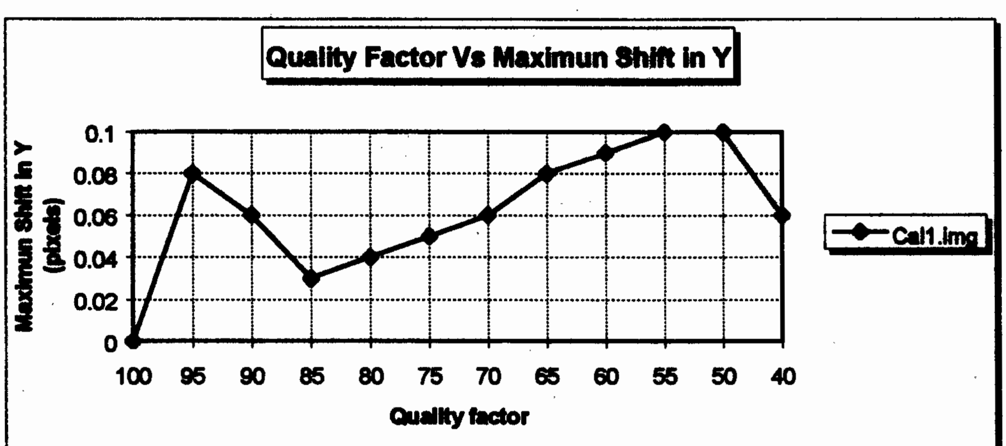


FIG 5.23: Quality Factor Versus Maximum Shift in Y for call image using JPEG (DCT-based) Compression

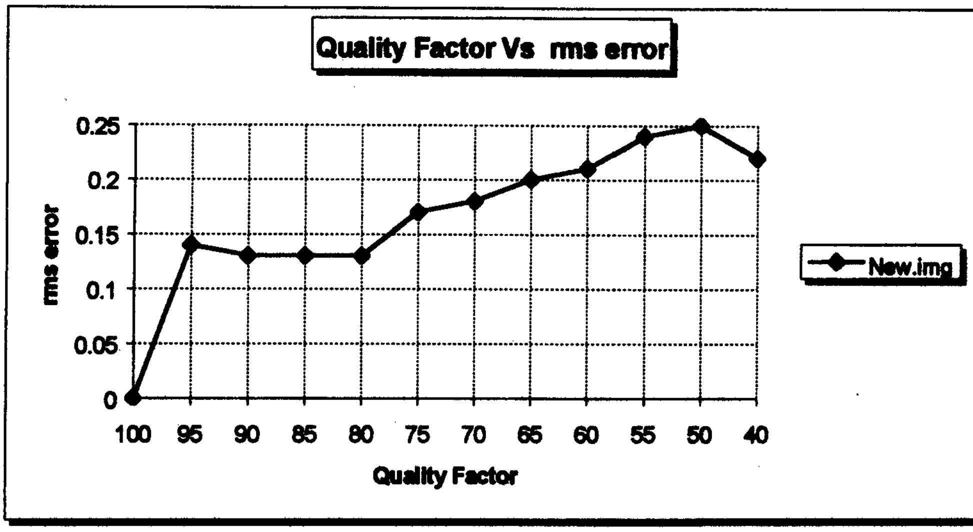


FIG 5.24*: *Quality Factor Versus Distortion for new image using JPEG (DCT-based) Compression

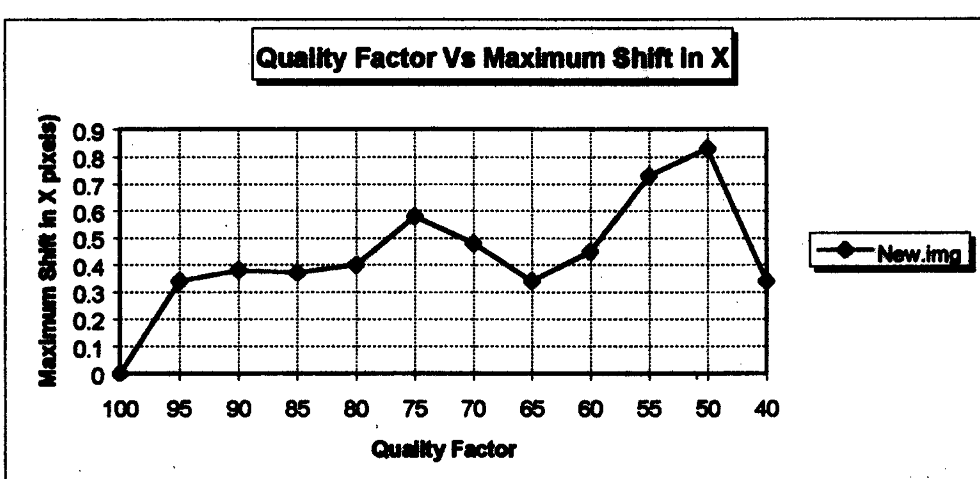


FIG 5.25*: *Quality Factor Versus Maximum Shift in X for new image using JPEG (DCT-based) Compression

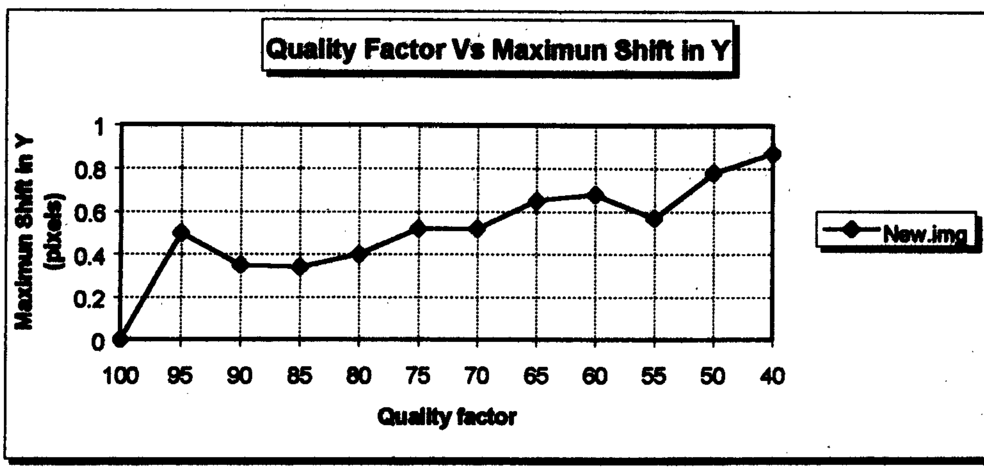


FIG 5.26: Quality Factor Versus Maximum Shift in Y for new image using JPEG (DCT-based) Compression

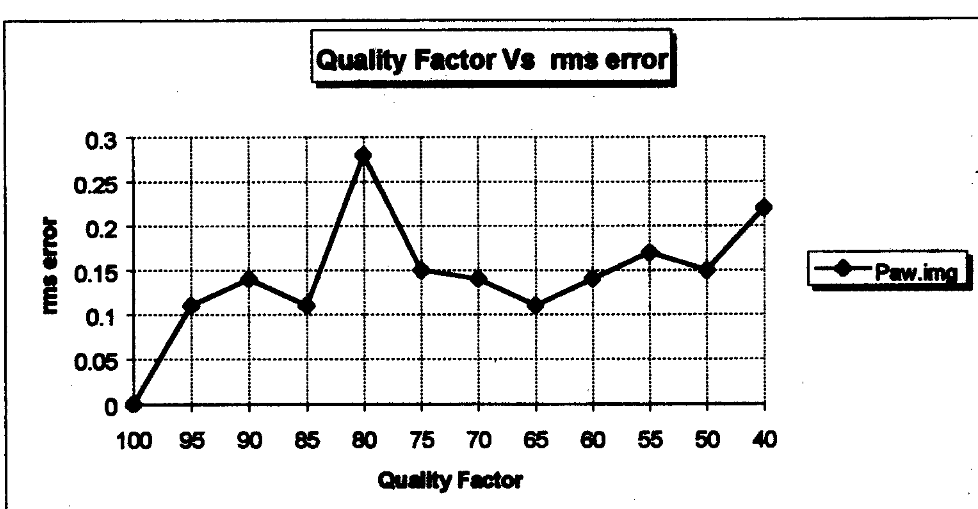


FIG 5.27: Quality Factor Versus Distortion for paw image using JPEG (DCT-based) Compression

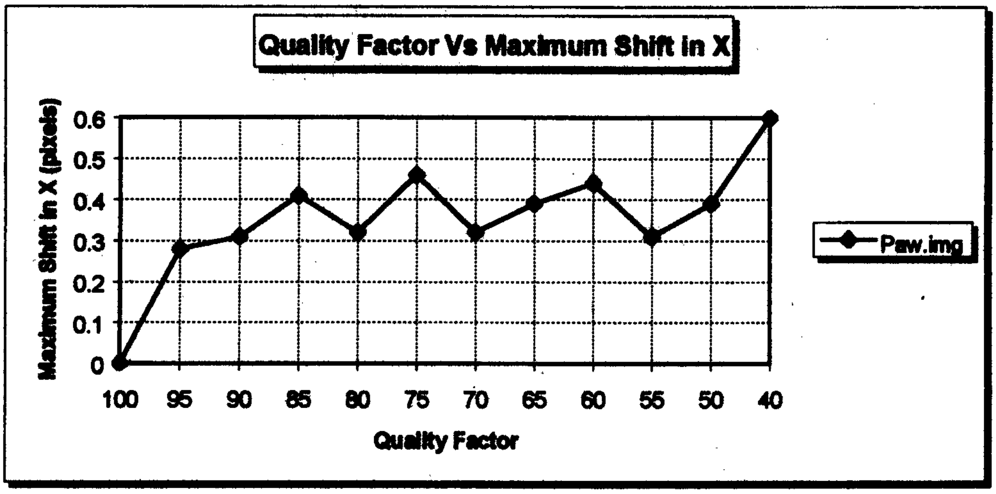


FIG 5.28: Quality Factor Versus Maximum Shift in X for paw image using JPEG (DCT-based) Compression

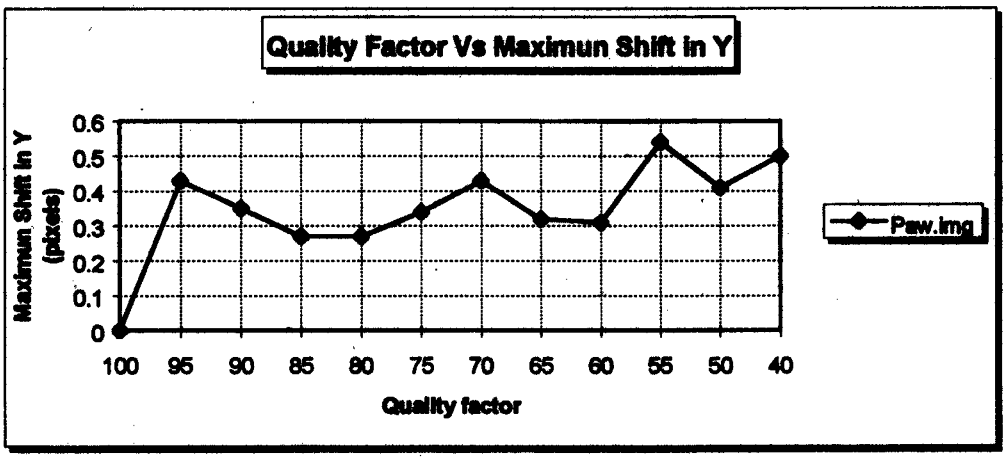


FIG 5.29: Quality Factor Versus Maximum Shift in Y for paw image using JPEG (DCT-based) Compression

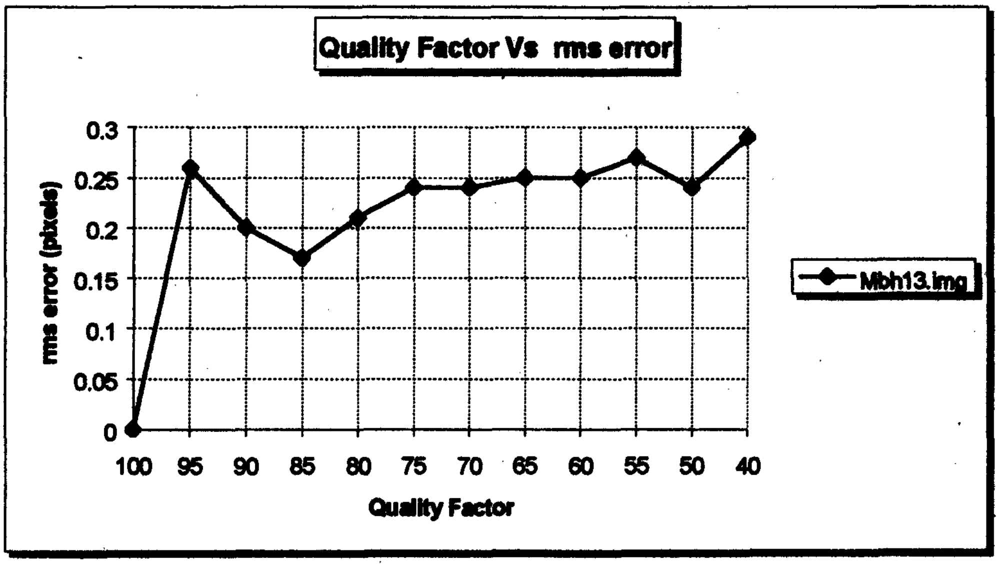


FIG 5.30: Quality Factor Versus Distortion for mbh13 image using JPEG (DCT-based) Compression

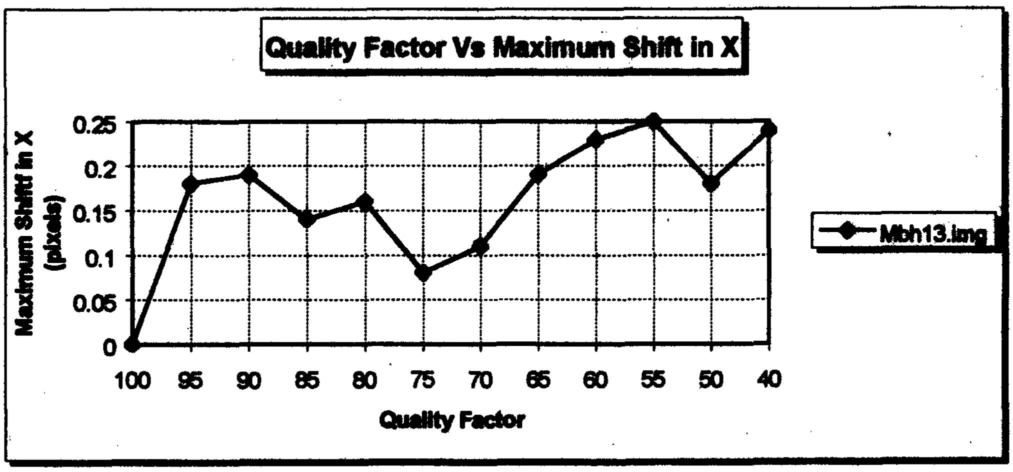


FIG 5.31: Quality Factor Versus Maximum Shift in X for mbh13 image using JPEG (DCT-based) Compression

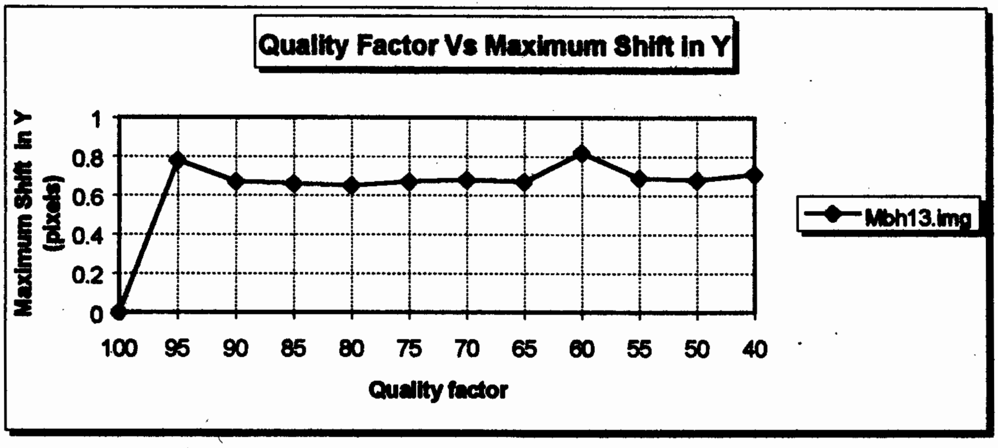


FIG 5.32: Quality Factor Versus Maximum Shift in Y for mbh13 image using JPEG (DCT-based) Compression

Chapter 6

Discussion

This chapter discusses the accuracy achieved by the target centring algorithm (discussed in section 5.3.3) and the precision due to JPEG image compression. The experimental results presented in sections 5.4.1 and 5.4.2 are interpreted and inferences are drawn.

6.1 Measures of Accuracy of Target Location

The accuracy of target locations in this thesis is considered as the same as the precision. That is, for repeated pointing on circular targets, the mean position of the target is the correct centre within the precision of pointing (Trinder, 1989). The output of the semi-automatic extraction of circular targets in the test images described in section 5.3.3 is a list of target positions and sizes. In section 5.4.2 results were presented. The estimated maximum shift in both x and y coordinates, their standard deviations, as well as the root mean square errors were also presented. This section therefore explains the accuracy of the target location and identification algorithm. It is also of importance to estimate how well the calculated positions of the targets in the reconstructed images fit to the corresponding position of the original images. The results of the test are also summarised and presented.

6.1.1 Repeatability Test of Targets in Real Image

Rubinstein (1990) and Van Der Vlugt (1995) describe a repeatability test of targets in real images using the target location and identification algorithm (i.e., discussed in section 4.3.3). The purpose of these test was to investigate and evaluate the shortcomings of the algorithm and to determine its maximal possible accuracy in the presence of noise. Rubeistein (1990) performed his test on three images (i.e, an image was taken of real target) having targets with known centre coordinates.

For each image, the deviation of all the targets of the algorithm from the known centre is used as a measure of accuracy. The results indicated that:

- the centring algorithm gives coordinates with accuracy which is equal to 0.01 of a pixel; and;
- the magnification of the target window does not improve centring accuracy.

Van Der Vlugt (1995), on the other used a flat wooden surface with a regular 10 by 10 grid of retroreflective targets was made for centring repeatability test of targets in real images. These targets, 8mm in diameter were spaced at approximately 3cm intervals. The wooden surface was painted with a mat black paint (blackboard paint) of low reflectance. The test involved the capturing of ten sequential images of target field and then performing centring on all the images for all 100 targets using weighted centre of gravity (WCG) with grey squared as weight technique. Table 6.1 shows the results of the repeatability test. The RMS Vx and RMS Vy columns are the root mean squares of the x and y coordinate residuals of every point in each image for WCG with grey squared as weight centring method. The max Vx and Vy columns are the maximum residual occurring this centring method. The results indicate that repeatability reaches 12/1000 th of a pixel in X and 8/1000 th of a pixel in Y.

Repeatability, units: pixels				
Method	**RMS Vx**	**RMS Vy**	**max Vx**	**max Vy**
WCG with grey squared as weight	0.012	0.008	0.052	0.027

Table 6.1: Repeatability of target centring method (WCG with grey squared as weight) method using 10 images(after Van Der Vlugt, 1995).

The agreement between these two test is very good and one can see that the centring algorithm offers a good method of obtaining reliable estimates for the centring process. However the resulting accuracy is dependent on the type of camera used.(Van Der Vlugt, 1995). Trinder (1989) in his investigation of precision of digital target location, also concluded that under ideal circumstances the precision of pointing can approach 0.01 of a pixel size and that variations in image quality has no significant effect on precision.

6.1.2 Effect of Target Size on the Accuracy of the Target Centre

The target size on the image is controlled by the physical size of the target and the distance from the lens (i.e., perspective). The weighted centre of gravity algorithm (discussed in section 4.3.3) is used to evaluate the effects of increasing the number of the pixels (Rubinstein, 1990). The graph (i.e., figure 6.1) illustrates the effects of increasing the number of pixels on the accuracy of the target centre. The accuracy of all target centring algorithm is primarily a function of the number pixels (those pixels with a grey level between saturation and background). Increasing the target size results in more pixel (and sub-pixels) defining the target and thus a higher precision target centre is expected. Increasing the number of sub-pixels is evaluated per implies a more accurate perimeter pixel grey value from which a higher precision target centre is expected.

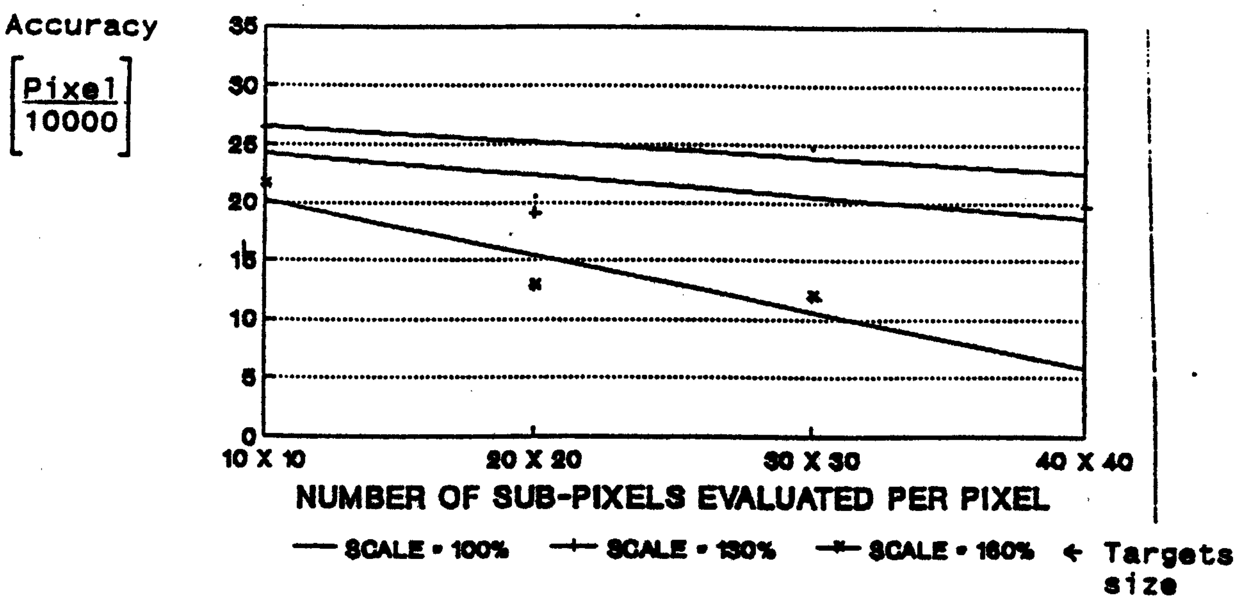


FIG 6.1: Effects of magnifying the target and evaluating increasing number of sub-pixel per pixel (after Rubinstein, 1990)

6.1.3 Accuracy of Centre of Gravity Algorithm Versus Threshold Levels

Improvement in accuracy can be obtained by only including those pixels grey values greater than the background level as observations. That is a rectangular window of suitable size, so that it covers the complete target and the surrounding area is approximately centred on a target and then thresholding within the window is carried out. This therefore calls for the interactive selection of an appropriate grey level threshold to identify the area of the target.

To overcome this problem, thresholding has been used to good effect, as in Trinder (1989), Rubinstein (1990) and Van Der Vlugt (1995). Van Der Vlugt used various techniques for determining a threshold value which separates background from target. One simple and fairly fast way is to use an interactive approach where the user changes the threshold value until a suitable binary is found by inspection. A further test was carried out by Van Der Vlugt (1995) for the weighted centre of gravity, with grey value as weight technique using the actual background threshold value. The target location error obtained under this technique had a root mean square error of 3/1000 pixels in X and 4/1000 pixels in Y.

Similarly, Rubinstein (1990) theoretically investigated the effects of the variation of the threshold levels (subtracted from pixel grey values) using synthetic symmetric targets. Figure 6.2 illustrates the accuracy of the centre of gravity algorithm versus the entire range of threshold levels subtracted from the pixel grey scale value of the targets. In the case of threshold level being greater than the grey level, the new grey level is set to zero.

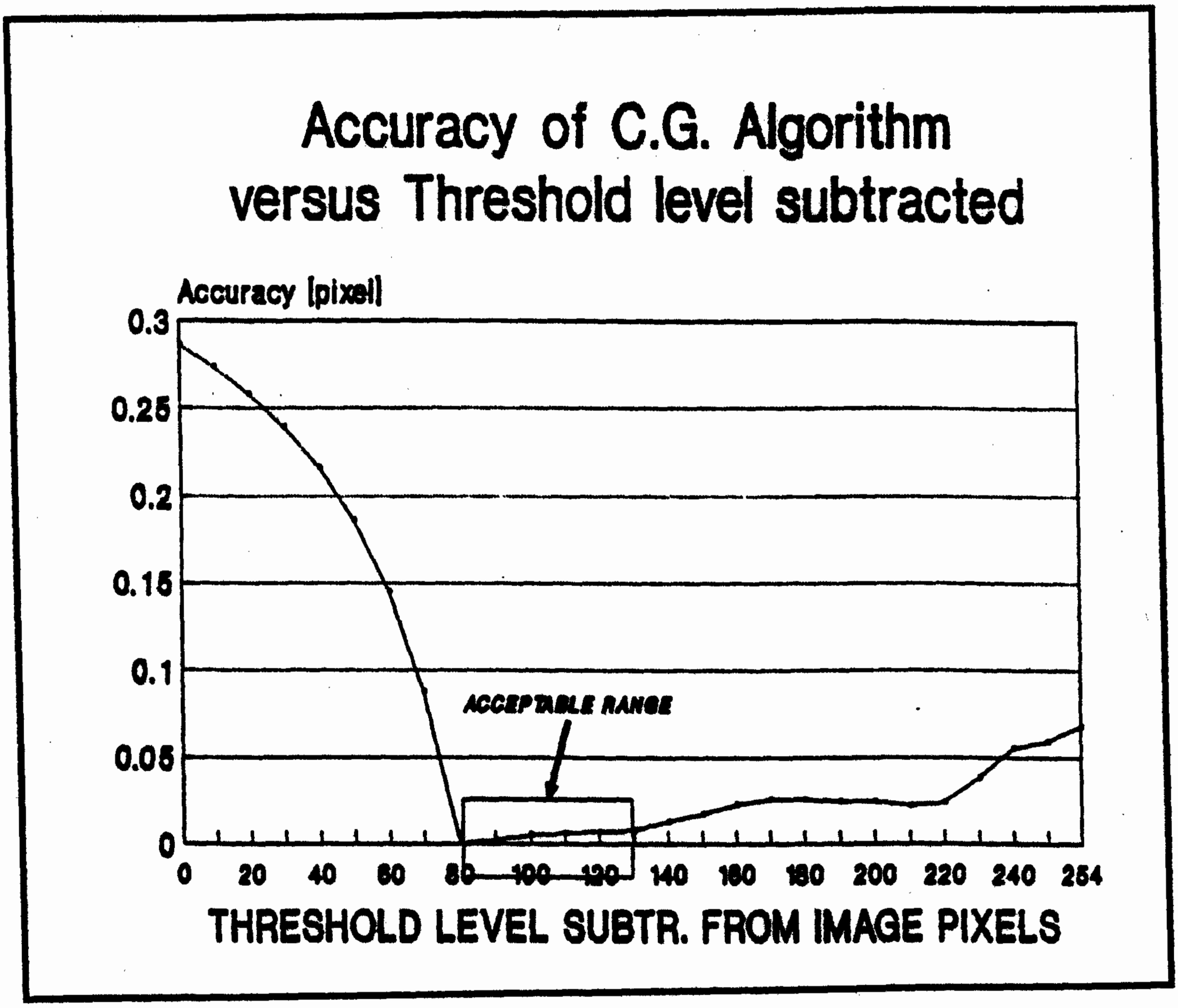


FIG 6.2: **Accuracy of centre of gravity versus the threshold level subtracted from the grey scale pixel value (after Rubinstein, 1990)**

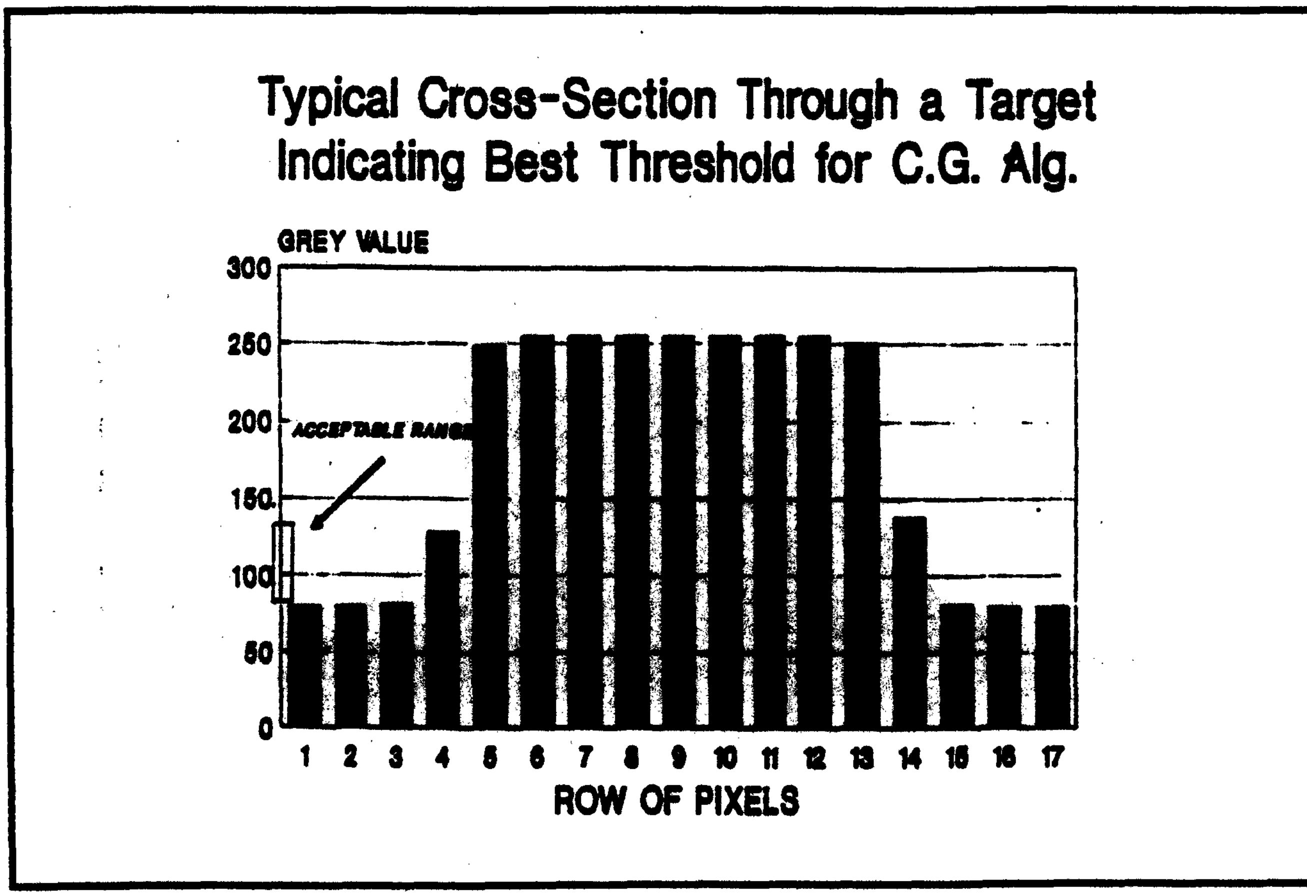


***FIG 6.3 :* Acceptable values of threshold superimposed on the target profile(after Rubinstein, 1990).**

From figure 5.2, the centre determination :

(i) is poor (accuracy of ± 0.01 pixel) for threshold levels below the background grey level;

(ii) is best for the threshold in a band above the background level (accuracy of centre between 1/1000 and 1/200 of a pixel); and;

(iii) gets progressively worse as the threshold level approaches a grey level of saturation (1/150 to 1/ 15 of a pixel centring accuracy).

Thus the choice of a threshold level of just above the background level in a target window is essential for good performance of the centre of gravity algorithm (figure 6.3).

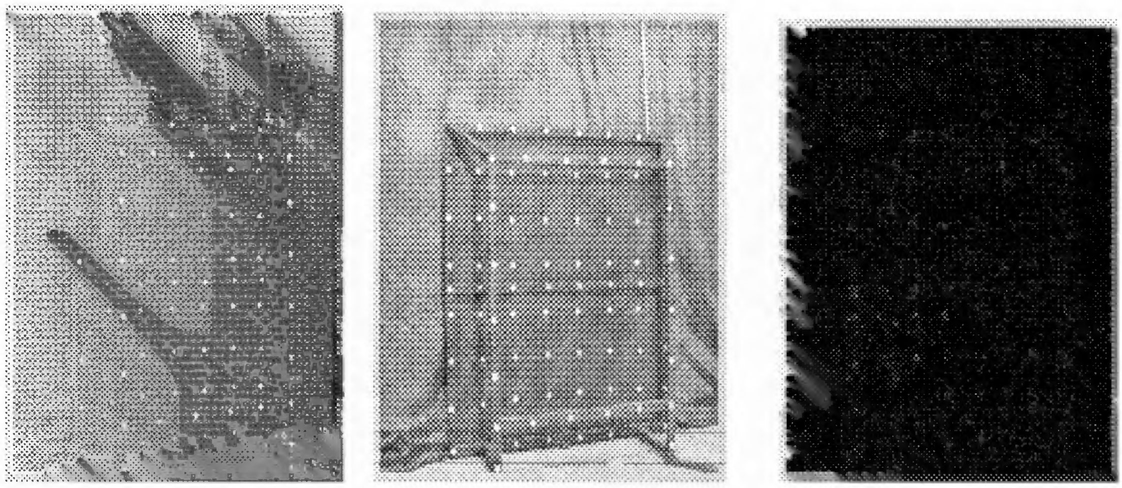
6.2 Degree of Compression

Table 5.1 and figure 5.2 show the compression ratios for the nine test images using JPEG(DCT-based) compression. Comparatively, results show that a large amount of compression can be achieved when using JPEG(DCT-based) lossy compression algorithm. Compression ratios obtained by LZW (in TIFF format) lossless compression is not as effective as compared to the compression ratio obtained by JPEG(DCT-based). For example, in table 5.1, the compression ratio achieved by JPEG (DCT-based) lossy compression for the *left* image is 23.9 whereas TIFF (LZW) lossless achieved only 1.4. This example shows that the compression ratio obtained by lossless schemes is limited by the entropy (a measure of information content of an image) of the image.

The results also show that the compression ratios vary for different images at the same quality factor with JPEG (DCT-based) algorithm. For example, in table 5.1, at JPEG quantization level 75, the *rock* image (captured by CCD Video Camera) was compressed only 6.94 times, while the *door* image (captured by DCS Digital Camera) was compressed 23.51 times. This difference is due to the amount of redundancy available in the *door* image and the large information content in the *rock* image. With the exception of the scanned photographs the compression rate for JPEG quality factor of 90 or less was greater than lossless LZW. The lower compression rate for *new* and *paw* images confirms the fact that JPEG (DCT-based) algorithm does not compress scanned images well.

In the case of this *Mbh13* image, JPEG quality factor of 30 produces a compression ratio of 70:1 which saves about 98% of the bytes used to store the original image. Even a higher compression ratios of 100:1 can be achieved by JPEG (DCT-based) with a very low quality factor. This high compression ratio makes JPEG a time and space trade off which is potentially capable of providing a high performance image compression for database, archiving, transmission and visual applications. It should be pointed out, that at these high compression rates (i.e., with quality factors below 60) there is detectable loss of detail and sharpness. These quality factors produce very small files with low image quality which will be objectionable for photogrammetric measurement purposes.

Figure 6.4 is an example of a JPEG-compressed image. Due to the limitations of the printing process, it may be difficult to see any differences between the two images (an example of visually lossless compression). However, as shown by the difference image (figure 6.4 (c)), there are changes in pixel values which might be important in any subsequent use or analysis of the image in industrial photogrammetry. The image in this figure also shows that even substantial compression does not necessarily degrade the visual appearance of the image as printed. In figure 6.4 (b), typical artifacts of blocks-based DCT compression can be seen (e.g., the obvious block boundaries and edge degradation). The visibility of these artifacts is dependent on the particular image and the conditions under which it is viewed.



(a): original image

(b): JPEG-compressed image at QF of 30 (24:1 compression ratio)

(c): The difference between (a) and (b)

FIG 6.4: Original and JPEG-compressed images at quality factor of 30(24:1 compression ratio)

Furthermore, it is noted from the shapes of the graphs in figure 5.3 that compression ratios are monotonically increasing with decreasing JPEG quality factors. Another interesting observation is that JPEG is a variable rate compression algorithm. This means that compression rate varies from image to image and cannot be determined *a prior*, although it is possible to increase or decrease compression.

6.3 Geometric Distortions

Tables 5.2 to 5.10 and figures 5.3 to 5.11 present the geometric distortions (in pixels) for different quantization levels of the nine test images. As can be seen, the range of rms errors under investigation represents a cross-section from lightly degraded (i.e., rms error =0.01) to heavily degraded (rms errors =0.06) cases. With images captured by CCD Video camera, the range in rms errors for the *left* image is from 0.01 to 0.04, with *prop* image is from 0.02 to 0.06, and for *frame* and *rock* images, they range from 0.01 to 0.03 and 0.01 to 0.02 respectively.

DCS images, *call* and *door* yield rms errors in the range of 0.01 to 0.03 and 0.01 to 0.06. Similarly, the range in rms errors for all the scanned images are 0.1 to 0.27. Generally, these range of rms errors revealed that there is no consistent reduction in accuracy with increasing level of compression ratio. It can also be seen from table 5.6 that the maximum shifts caused by JPEG(DCT-based) image compression in x and y coordinate under quality factor of 55 for the *door* image are 0.08 and 0.09 respectively. The repeatability of the target location algorithm (discussed 6.1.1) is 9/1000th of a pixel and the accuracy due to thresholding (in section 6.1) is 0.005. Accepting these levels of accuracy in determining the location of targets, then the shifts of this magnitude (0.08 and 0.09) are beyond the limits of the repeatability of the target centroiding algorithm and therefore are significant in precise industrial photogrammetric applications. But it is interesting to note that the maximum rms error due to JPEG (DCT-based) image compression of the *door* image is 0.02 pixel at the compression rate of 16:1 which is within the limits of the repeatability of the target centroiding algorithm and therefore are quite insignificant and can be neglected in even precise industrial photogrammetric measurements.

The results of the rms errors and the maximum shifts in both directions seem to contradict each other in the performance evaluation of JPEG algorithm. A possible explanation for this inconsistency is that the rms errors seem to compensate each other in the computation of the rms errors and this confirms the fact that the rms error is not a reliable and consistent measure for determining the magnitude of distortion of the reconstructed image cause by JPEG lossy compression algorithm. In other words, the rms error is not a good indicator of the visualisation of a reconstructed image. However, it remains as the standard quantitative measure of image quality for lossy image compression such as JPEG.

Generally, the highest rms errors of all the test images (i.e., nine images) are obtained, when the quality factors are 40 and 95. These high rms errors substantiate the warnings associated with the documentation supplied with JPEG compression and decompression source code (Lane *et al*, 1994). Part of these high rms errors can be attributed to information loss in quantization step, subsampling, as well as round off.

The high rms error for the quality factor of 95 seems to agree with the expectation that the quality 95 is mainly of interest for experimental purposes (Lane *et al,* 1994) and is not therefore recommended for normal use in industrial photogrammetric measurements.

In general, one can conclude that geometric distortions for quality factors between 60 and 90 (rms error in the range of 0.01 to 0.03 pixels) are within the limits of the repeatability of the target centroiding algorithm and therefore are quite insignificant and can be neglected. In other words, industrial photogrammetric imagery wherein circular signalized targets are to be measured can be compressed using the JPEG baseline algorithm down to quality factors of 60. Thus, it is safe to use these images for precise photogrammetric measurement, analysis and point determination. Reconstructed images with quality factors below 60 and above 90 (rms errors in the range of 0.05 to 0.06 pixels) are significant and can be used for tasks that do not require precise point determination, such as the extraction and interpretation of features. This is because at high compression ratios (i.e., compression ratio above 20), more high-frequency and low-amplitude changes get averaged, which can cause an entire pixel block to adopt the same color , and subsequently cause blockiness that is characteristics of JPEG-compressed images. Quality factors around 5 to 10 might be useful in preparing an index of a large image library and for images intended for visual examination.

6.3.1 Target Size and Position

Another interesting feature to be considered in analysing the possible causes of distortions by JPEG(DCT-based) compression are the position of targets in the digital images. From table 4.1, the targets with maximum shifts in x and y coordinates were investigated with respect to their positions in the image. Figure 6.1 is an example showing the positions of the most degraded targets in the digital images. It was found that most of the targets in the sharp boundaries, edges and corners of images show the greatest degradation.

A possible explanation of this phenomena is the nature of JPEG compression process which keeps the larger terms and erase the smaller ones in the discrete cosine transform of each block. This is likely to remove more of the small high frequency terms in images with more detail present. Furthermore, the influence of the relative dimensions of the pixel in relation to the target size is investigated. Figure 6.5 shows the relative sizes of target in the image. Targets in (a) to (c) are relatively small as compared to targets in (d) to (e). It was found during the analysis that the sizes of some of the most degraded targets as discussed in section 6.3 were relatively small.

The poorer rms errors associated with these targets may be due to the fact that the accuracy of the target centring algorithm (discussed in section 6.1.2,) increases with increasing the number of pixels per target. It can be generally concluded that, if the target sizes are relatively small, pointing precision will deteriorate and this will consequently affect the rms errors. The precision also improves and the magnitude of the rms error will be small as the target size increases. Due to the way JPEG (DCT-based) subdivides the whole image data into 8 x 8 pixel blocks, it has not been fully understood whether the physical size of the targets contributes to the geometric distortions discussed in section 6.3. A further studies need to be performed to assess how JPEG compression affects targets size in digital images for close-range photogrammetric purposes.

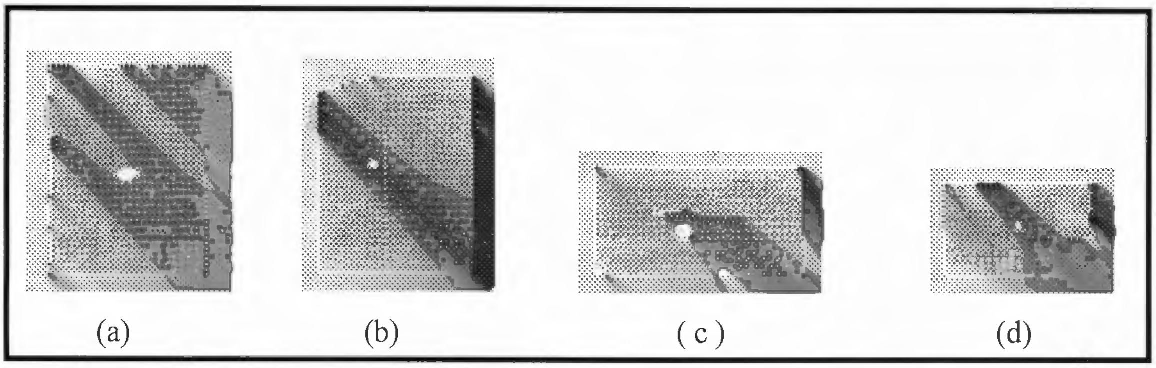


Figure 6.5: Positions of the most degraded targets

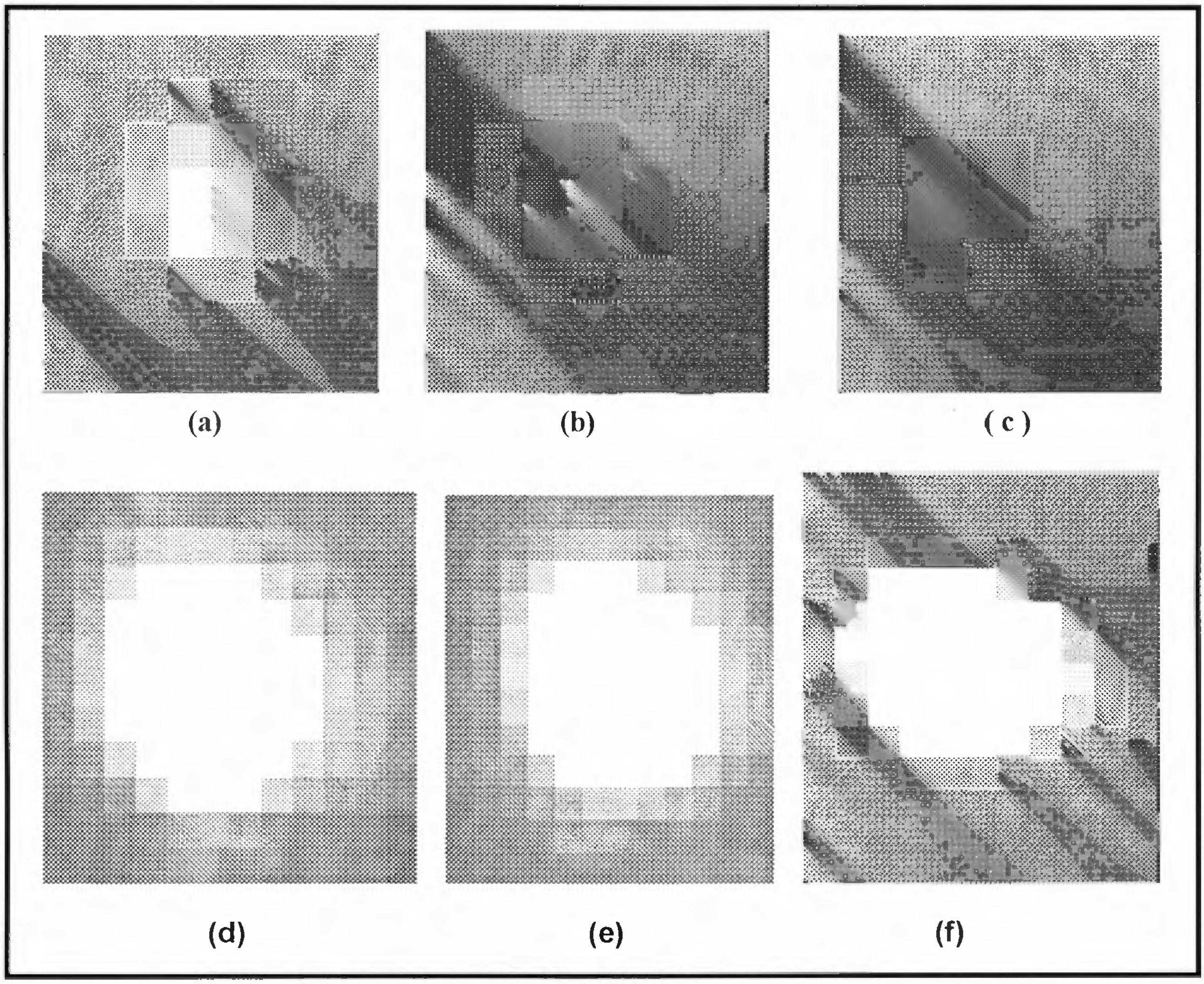


FIG 6.6: The relative dimensions of the pixel in relation to the target size

6.4 Pointing Precision of Target location due to Compression

The difference between the mean target location of the original image, and the mean target location of the reconstructed data set was used as a measure to estimate the precision. This was done under the assumption, that the original data sets were free from errors and they were "true values". The precision results of locating a target in the compressed images are summarised in tables 6.2 to 6.10.

The location of a target can be achieved with precision of 0.01 to 0.03 pixels in the images captured by both the CCD video and DCS digital cameras under the quality factors ranging between 60 and 90. In the case of quality factors above 90 and below 60, location of a target can be achieved with precision ranging between 0.03 and 0.06. However, with the scanned images, the precision ranges from 0.02 to 0.18 for quality factors between 60 and 90 whereas quality factors above 90 and below 60 achieve precision of 0.05 to 0.22. It can therefore be concluded that the precision degrades as the original image is subjected to more compression.

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.02	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03
(pixels)	Y	0.02	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0 .03	0.03

Table 6.2: Precision of JPEG(DCT-based) at various quality factors Left image

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.06	0.01	0.02	0.01	0.02	0.02	0.03	0.03	0.03	0.03	0.03
(pixels)	Y	0.05	0.01	0.02	0.02	0.03	0.03	0.03	0.02	0.05	0 .03	0.06

Table 6.3: Precision of JPEG(DCT-based) at various quality factors Prop image.

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.02	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02	0.03
(pixels)	Y	0.02	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.03	0 .03	0.03

Table 6.4: Precision of JPEG(DCT-based) at various quality factors Frame image.

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.03	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.02
(pixels)	Y	0.03	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0 .03	0.02

Table 6.5: Precision of JPEG(DCT-based) at various quality factors Rock image.

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.05	0.01	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02	0.03
(pixels)	Y	0.05	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0 .03	0.04

Table 6.6: Precision of JPEG(DCT-based) at various quality factors Door image.

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.04	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02
(pixels)	Y	0.04	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0 .02	0.02

Table 6.7: Precision of JPEG(DCT-based) at various quality factors Call image.

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.06	0.12	0.11	0.14	0.15	0.14	0.16	0.13	0.18	0.15	0.16
(pixels)	Y	0.10	0.12	0.13	0.12	0.19	0.19	0.11	0.18	0.18	0.16	0.22

Table 6.8: Precision of JPEG(DCT-based) at various quality factors New image

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.11	0.15	0.12	0.10	0.16	0.15	0.13	0.13	0.13	0.16	0.21
(pixels)	Y	0.14	0.12	0.08	0.08	0.12	0.18	0.11	0.12	0.18	0.1	0.22

Table 6.9: Precision of JPEG(DCT-based) at various quality factors Paw image

Quality Factor		95	90	85	80	75	70	65	60	55	50	40
Precision	X	0.03	0.02	0.02	0.02	0.02	0.03	0.04	0.04	0.05	0.05	0.06
(pixels)	Y	0.22	0.08	0.05	0.10	0.13	0.12	0.13	0.14	0.16	0 .13	0.20

Table 6.10: Precision of JPEG(DCT-based) at various quality factors Mbh13 image

Chapter 7

Conclusions and Recommendations

The goal of this thesis was to quantitatively evaluate the geometric effects of baseline JPEG image compression on grey scale images and to determine the optimal quality factor that can produce an image which is visible accepted and with minimum degradation in quality. Image compression techniques, primary image file formats relevant to softcopy photogrammetry, remote sensing and multimedia GIS were reviewed and discussed.

TIFF is a format for storage, interchange, display and printing of bitmap images. Its main strength is a highly flexible and platform-independent format which is supported by numerous image processing applications. TIFF is a full-featured format in the public domain, capable of supporting compression, tilling, colorimetry calibration, and the extension to include geographic metedata . The TIFF file extension, GeoTIFF, is embedded in TIFF file format as tags to handle cartographic and geographic information. Another feature of TIFF which is also useful is the ability to decompose an image by tiles rather scanlines. GIF is a bitmap format to store bitmap data with a pixel depths of 1 to 8 bits. GIF images are always stored using RGB color model and pallete data. Sunraster file formats is capable of storing black-and -white, grey-scale, and color bitmapped data of any pixel depth. JFIF is the JPEG interchange file format which handles JPEG compressed files across hardware platforms. In addition to bitmap and vector file formats, there are also animation, multimedia, meta and scene file formats. Storage of multimedia data and information in a disk is similar to image file formats.

The main aim of image compression techniques is to minimise image data volume while minimising information, and all basic image compression groups have advantages and disadvantages. The distortions introduced by the simple compression (i.e. truncation) and interpolative(i.e. subsampling) techniques for a given level of compression are generally much larger than the more sophisticated methods available for image compression.

Transform based (e.g DCT) techniques better preserve subjective image quality, and are less sensitive to statistical image property changes both inside a single image and between images. Prediction techniques (DPCM), on the other hand, can achieve larger compression ratios in a much less expensive way, tend to be much faster than transform based or Vector Quantization compression schemes, and are easily realised in hardware. Adaptive techniques improve performance substantially.

However, JPEG is becoming more widely accepted as computing industry's standard encoding method for grey scale and continuous-tone images for the computing industry. The JPEG standard is actually a collection of compression algorithms that can be used in different circumstances. The four JPEG compression modes are: baseline (sequential) encoding, progressive encoding, lossless encoding, and hierarchical encoding. The baseline mode is the basic coding and decoding method in the JPEG standard. Baseline JPEG standard achieves compression by dividing the image into 8 x 8 blocks, with each block compressed using a discrete cosine transform. The block artefacts generated in the reconstructed images by this approach can render block coding unacceptable for industrial photogrammetric applications.

In the experiment, the baseline JPEG was used to compress and decompress a set of test images (grey scale images with retroreflective targets) at quality factors between 30 and 95. The algorithm has been tested on grey scale images containing various level of information content with different image capture capabilities. Extracting targets from these images is more reliably accomplished by using the weighted centre of gravity, with grey value as weight technique. Both compression ratios and geometric distortions were calculated, and from the evaluation given in chapter 5, a large amount of compression can be achieved when using JPEG(DCT-based) lossy compression algorithm. In general terms, the compression ratio results in table 5.1 indicates that the JPEG algorithm is more capable of encoding grey scale or continuous-tone images for the computing industry.

Quantitative evaluation also shows that:

- the location of a target can be achieved with precision of 0.01 to 0.03 pixels in the images captured by both the CCD video and DCS digital cameras under the quality factors ranging between 60 and 90. In the case of quality factors above 90 and below 60, location of a target can be achieved with the precision ranging between 0.03 and 0.06; However, with the scanned images, the precision ranges from 0.02 to 0.18 for quality factors between 60 and 90 whereas quality factors above 90 and below 60 achieve precision of 0.05 to 0.22. It can therefore be concluded that the precision degrades as the original image is subjected to more compression (i.e., targets locations geometrically shifted as the quality factor is decreased). Moreover, there was no consistent reduction in accuracy with increasing level compression; and;

- target location error for quality factor between 60 and 90 (i.e., rms errors in the range of 1/100 to 3/100 pixels) are within the limits of the repeatability of the target centroiding algorithm and therefore are quite insignificant and can be neglected. In other words, industrial photogrammetric imagery wherein circular signalised targets are to be measured can be compressed using the JPEG baseline algorithm down to quality factors of 60. Thus, it is safe to use these images for precise photogrammetric analysis and point determination. Smaller quality factors may distort the images considerably and are not recommended for industrial photogrammetric measurement and analysis. Reconstructed images below 60 and above 90 (i.e., rms errors in the range of 0.05 to 0.06) can be used for tasks that do not require precise point determination, such as the extraction and interpretation of features. Quality factors around 5 to 10 might be useful in preparing an index of a large image library, because these quality factors produce more compression which saves a significant amount of space. Quality factors above 90 are mainly of interest for experimental purpose and not recommended for normal use in industrial photogrammetric measurements. In the other direction, quality factor below 60 will produce very small files of low image quality.

In future investigations the following points should be further addressed:

- the influence of radiometric distortions should be considered in accessing the effects of JPEG image compression for industrial photogrammetric purposes;

- computing time required to compress and decompress photogrammetric imagery using JPEG compression standard;

- the influence of the pixel size and pixel position in the image and the effects of noise on the precision of pointing; and;

- the effects of JPEG compression on non targeted features such as edges, roads and buildings.

9.0 References

Aldius Microsoft Technical Memorandum, 1988. TIFF Revision 5.0

Algarni D, A., 1996. Compression of Remotely Sensed Data Using JPEG. *International Archives of Photogrammetry & Remote Sensing*, Vol. XXXI, Part B3, pp 24-28, Vienna, Austra

Buford Koegel J.J., 1994. Multimedia Interchange. (In) Multimedia Systems (ed) John F. Koegel Buford, Reading, Massachusetts pp 323-340.

Cavigioli C., 1993. JPEG Compression. In Applications Reference Manual , Analog Devices, Inc, USA (ed) Steve Guinta, pp 45-48.

Cavigioli C., 1993. Image Compression: Spelling Out the Options In Applications Reference Manual, Analog Devices, Inc, USA (ed) Steve Guinta, pp 41-43.

De Natale F. G. B., 1993. Optimized Bi-linear Interpolation and DCT: A Two-source Coding for Image Compression . *In Proceeding Data Compression Conference*, editors James A Storer and Martin Cohn, pp 487, Snowbird, Utah. IEE Computer Society Press

Eskicioglu A. M., Fisher P. S., 1993. A Survey of Quality Measures for Gray Scale Image Compression. *In Proceeding Data Compression Conference*, editors James A Storer and Martin Cohn, pp 487, Snowbird, Utah. IEE Computer Society Press

Forchhammer S., 1993. Adaptive Context for JBIG Compression of Bi-level Halftone Images. *In Proceeding Data Compression Conference*, editors James A Storer and Martin Cohn, pp 431, Snowbird, Utah. IEE Computer Society Press

Franti P., Nevalainen O., Kaukoranta., 1994. Compression of Digital Images by Block Truncation Coding: A Survey. *The Computer Journal*. Vol. 37, No.4, pp 308-332

Fraser C. S., Edmundson K. L., 1996. The Metric Impact of Reduction Optics in Digital Cameras. *The Photogrammetric Record.* Vol. XV, No.87, pp 437-446

Ghafourian M. A., Huang C. M., 1993. Comparison between several Adaptive search Vector Quantization Schemes and JPEG Standard FOR Image Compression. *In Proceeding Data Compression Conference*, editors James A Storer and Martin Cohn, pp 436, Snowbird, Utah. IEE Computer Society Press

Gibbs S.J., Tsichritzis D.C., 1995. Multimedia Programming: Objects, Environments and Frameworks, Addison-Wesley Publishing, Workingham, England.

Gonzalez R. C., Woods R. E., 1992. Digital Image Processing, Addison-Wesley Publishing, Massachusetts.

Gonzalez R. C., Woods R. E., 1992. Digital Image Processing, Addison-Wesley Publishing, Massachusetts.

Isabel W., 1994. Statistical Methods and Financial Calculations, Juta & Co Ltd, Cape Town

Jain A.K., 1989. Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Clieffs, New Jersy.

Jain A.K., 1981. Image Data Compression: A Review. *Proceedings of the IEEE.* Vol. 69, No.3, pp 349-389.

Jaakola. J., Orava. E., 1994. The effects of Pixel Size and Compression on Metric Quality of Digital Aerial Images. *International Archives of Photogrammetry & Remote Sensing*, Vol. 30, Part 3/1, pp 409-415, Munich, Germany.

Jones P.W., Rabbani M. 1994. Digital Image Compression. *In Digital Image Processing Methods* (Ed.) Edward R. Douchery, New York, pp 261-326.

Kay D.C., Levine J.R., 1995. Graphic File Formats, McGraw-Hill, Inc, New York, pp. 21-234

Krzystek, P. 1991. Fully Automatic Measurement of Digital Evaluation Models with MATCH-T. *Proc. ISPRS Conference of Digital Photogrammetric Systems,* Munich, September 3-6, pp. 203-214.

Lammi J., Sarjakoski T., 1992. Compression of Digital Color Images by the JPEG. *International Archives of Photogrammetry and Remote Sensing*, vol.29, Part B2, pp. 456-460, Washington D.C., USA.

Lane T., Gladstone P., Ortiz L., Boucher J., Crocker L., Phillips G., Rossi D., Weijers G., 1994. The Independent JPEG Group's JPEG Software Manual.

Luther A. C., 1994. Digital Video and Image Compression. In Multimedia Systems (ed) John F. Koegel Buford, Reading, Massachusetts pp 143-174.

Markas T., Relif J., 1993. Multispectral Image Compression Algorithms. *In Proceeding Data Compression Conference*, editors James A Storer and Martin Cohn, pp 391-392, Snowbird, Utah. IEE Computer Society Press

Milkhail E.M., Akey M. L., Mitchell O.R., 1984. Detection and Sub-pixel Location of Photogrammetric Targets in Digital Images, *Photogrammetria,* Vol. 39, No.3, pp 63-83.

Monroe D. M., Sherlock B.G., 1993. Optimum DCT Quantization. *In Proceeding Data Compression Conference*, editors James A Storer and Martin Cohn, pp 188-194, Snowbird, Utah. IEE Computer Society Press.

Murray J.D., vanRyper., 1994. Encyclopedia of Graphics File Formats, O'Reilly & Associates, Sebastopol, CA.

Nelson M. 1991. The Data Compression Book: Featuring fast, Efficient Data Compression Techniques in C, M & T Books, Redwood City, CA, pp. 348-406.

Novak K., Shahin F.S. 1996. A Comparison of Two Image Compression Techniques for Softcopy Photogrammetry. *Photogrammetric Engineering and Remote Sensing*, pp. 695-701.

Numes P.R, Alcaim A., da Silva M. 1992. Compression of Satellite Images for Remotely Sensed Applications, *International Archives of Photogrammetry and Remote Sensing*, Volume 29, Comm.II, pp. 479-483.

Ohnesorge K. W., Stucki P., Bichsel M., 1993. Data Model for Lossless Image Compression Using Arithmetic Coding. *In Proceeding Data Compression Conference*, editors James A Storer and Martin Cohn, pp 441, Snowbird, Utah. IEE Computer Society Press.

Ozer J., 1995 Video Compression for Multimedia, AP Professional, Boston, USA.

Schalkoff R.J., 1989. Digital Image Processing and Computer Vision, John Willey & Sons, New York.

Razavi A., 1993. A High Performance JPEG Image Compression Chip Set for Multimedia Applications. *In Proceeding Data Compression Conference*, editors James A Storer and Martin Cohn, pp 454, Snowbird, Utah. IEE Computer Society Press

Robinson C., Montgomery B., Fraser C., 1995. The Effects of Image Compression on Automated DTM Generation. In Fritsch, D. (Ed.), *Photogrammetric Week '95*, Wichmann, Karlsruhe, pp 255-263.

Rubinstein M., 1990. Assessing Target Centring Agorithms For Use In Near-Real-Time-Photogrammetry. Master of Science Thesis, University of Cape Town, pp. 58-132.

Russ J.C., 1995. The Image Processing Hand Book, CRC Press, Boca Raton, Florida, USA

Sonka M., Hlavac V., Bolye R., 1993. Image Processing, Analysis and Machine Vision, Chapman & Hall Computing, London.

Storer J.A., 1992. Image and Text Compression. Kluwer Academic Publishers, New York.

Trinda J.C., 1989. Precision of Digital Target Location. *Photogrammetric Engineering and Remote Sensing*, Vol. 55, No. 6 , pp. 883-886.

Van Der Vlugt G ., 1995. Algorithm and Design Aspects of an Automated Vision Based 3-D Surface Measurement System. Doctor of Philosophy Thesis, University of Cape Town, pp. 41-60

Wallace G.K., The JPEG Still-Picture Compression Standard. Communications of the ACM, Vol. 34, No. 4, pp. 31-44.

Xue R.G., 1992. The Image Processing For The Target Centre Detection In Digital Image. Master of Science Thesis, University of Cape Town, pp. 16-43.

Appendix A

GRAPHS OF COMPRESSION RATE VERSUS DISTORTION

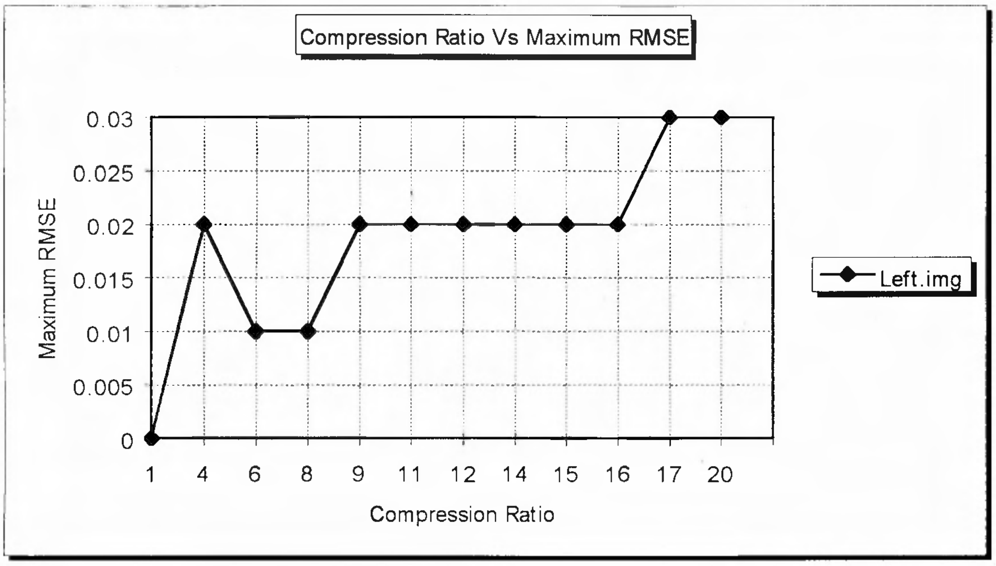


FIG A1: Compression Rate Versus Distortion for Left image using JPEG (DCT-based) Compression.

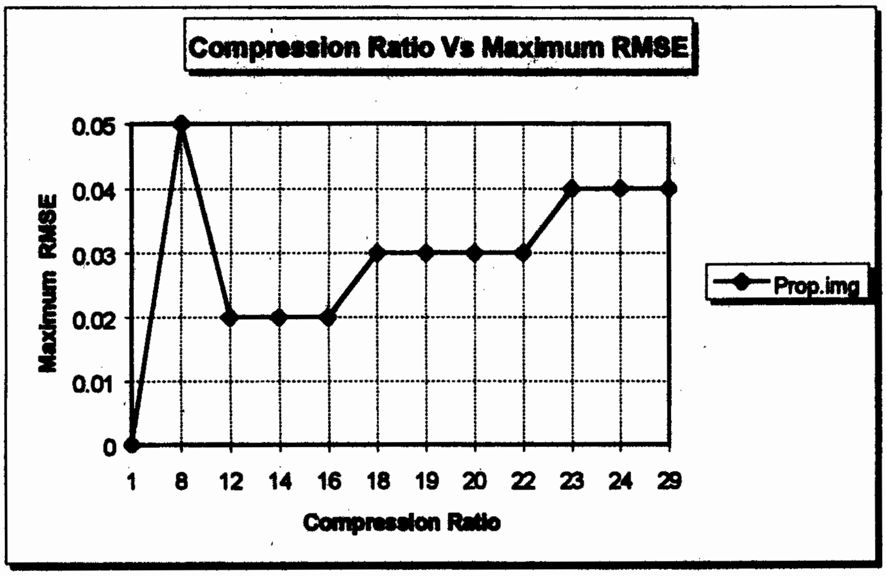


FIG A2: Compression Rate Versus Distortion for Prop image using JPEG (DCT-based) Compression

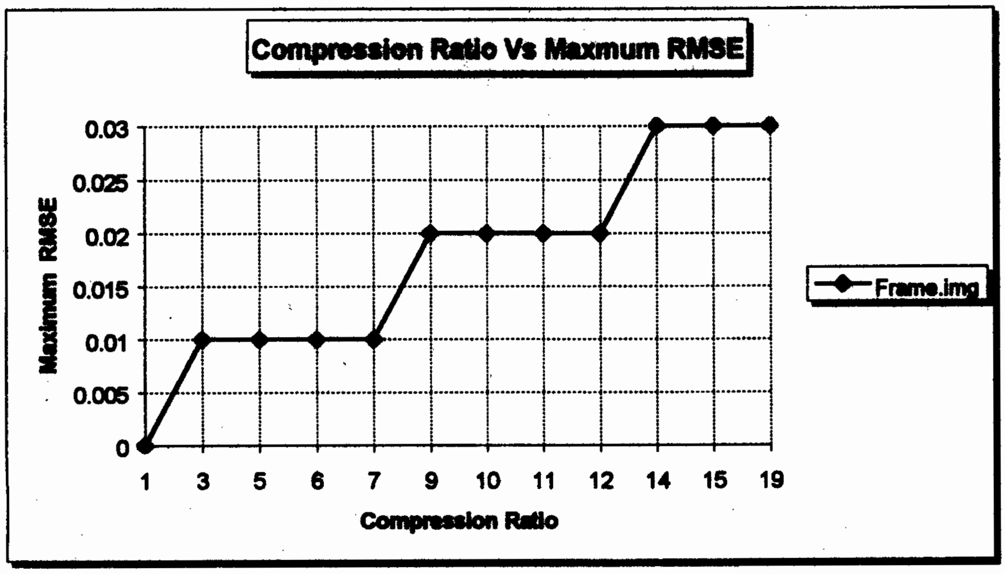


FIG A3: Compression Rate Versus Distortion for Frame image using JPEG (DCT-based) Compression

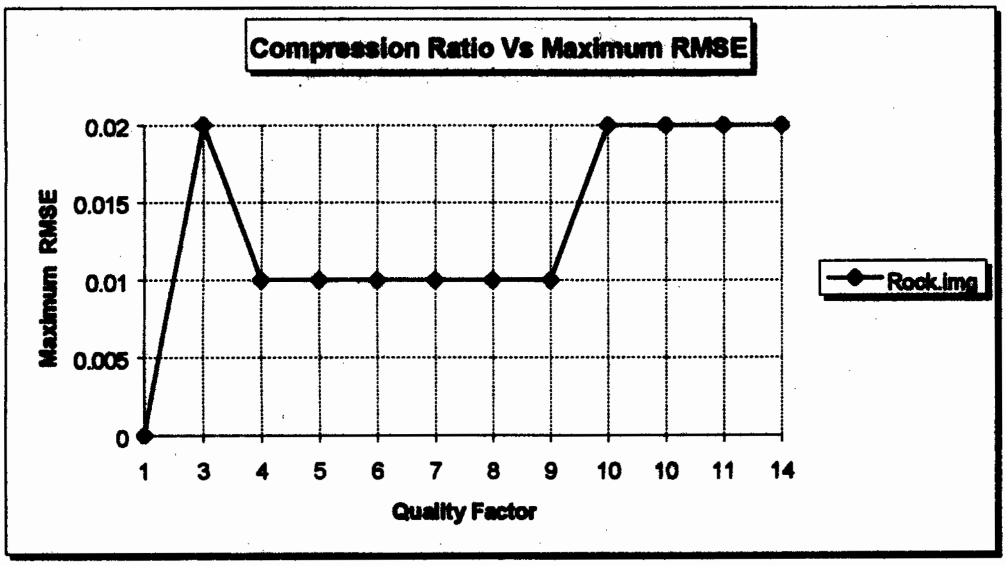


FIG A4: Compression Rate Versus Distortion for Rock image using JPEG (DCT-based) Compression

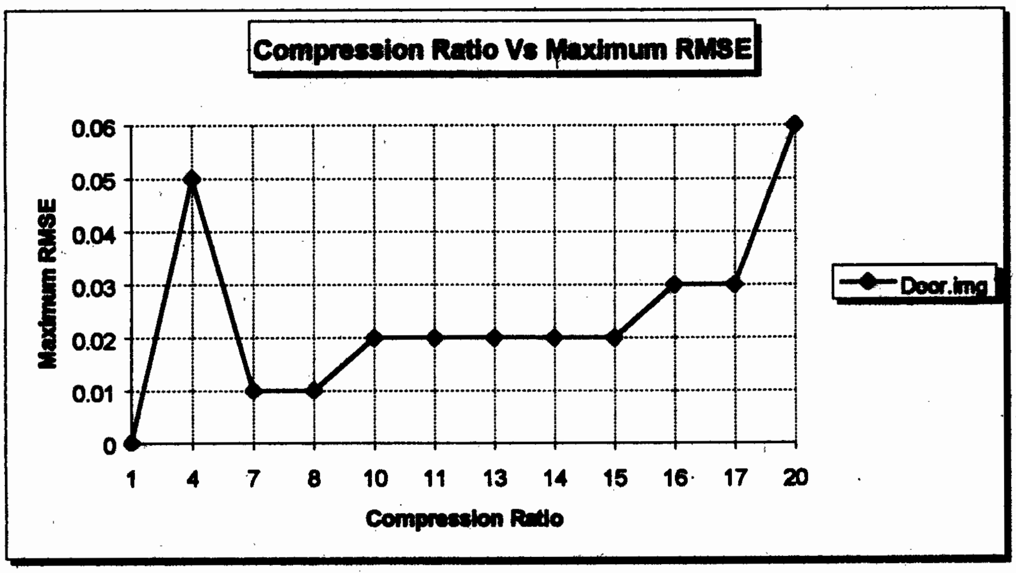


FIG A5: Compression Rate Versus Distortion for Door image using JPEG (DCT-based) Compression

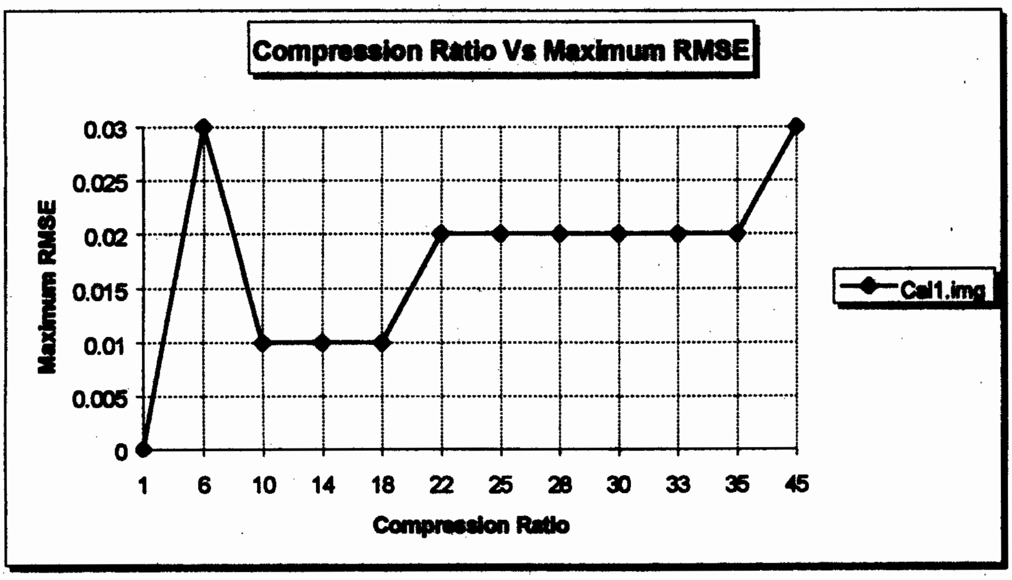


FIG A6: Compression Rate Versus Distortion for Call image using JPEG (DCT-based) Compression

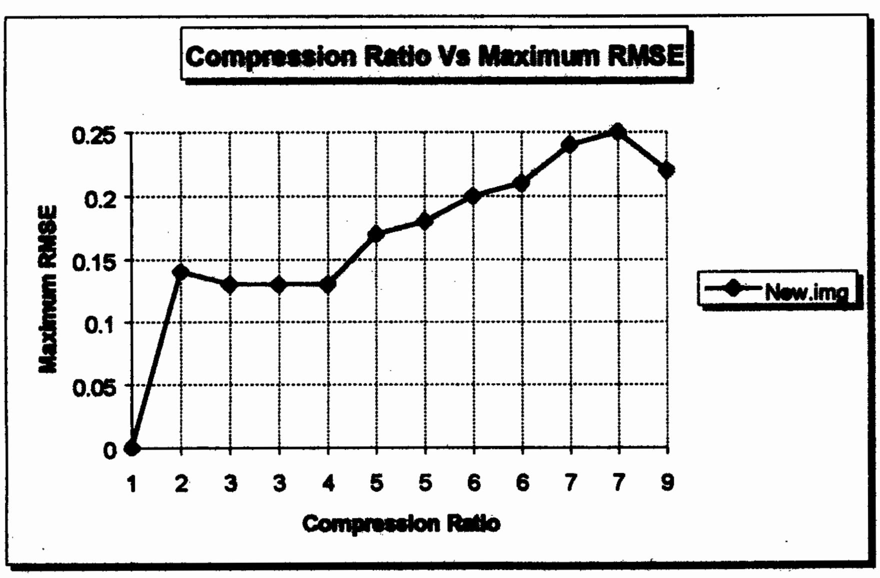


FIG A7: Compression Rate Versus Distortion for New image using JPEG (DCT-based) Compression

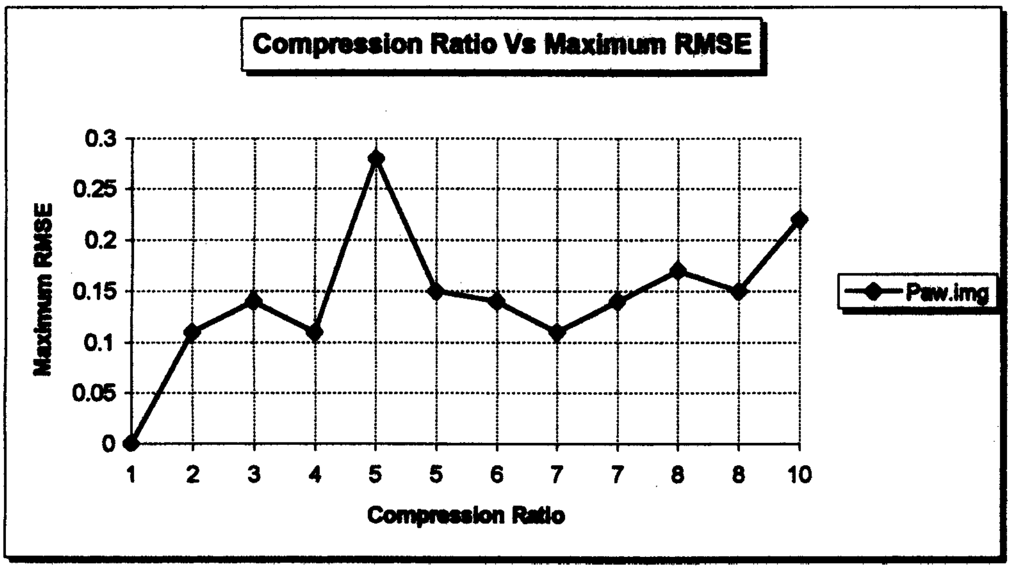


FIG A8: Compression Rate Versus Distortion for Paw image using JPEG (DCT-based) Compression

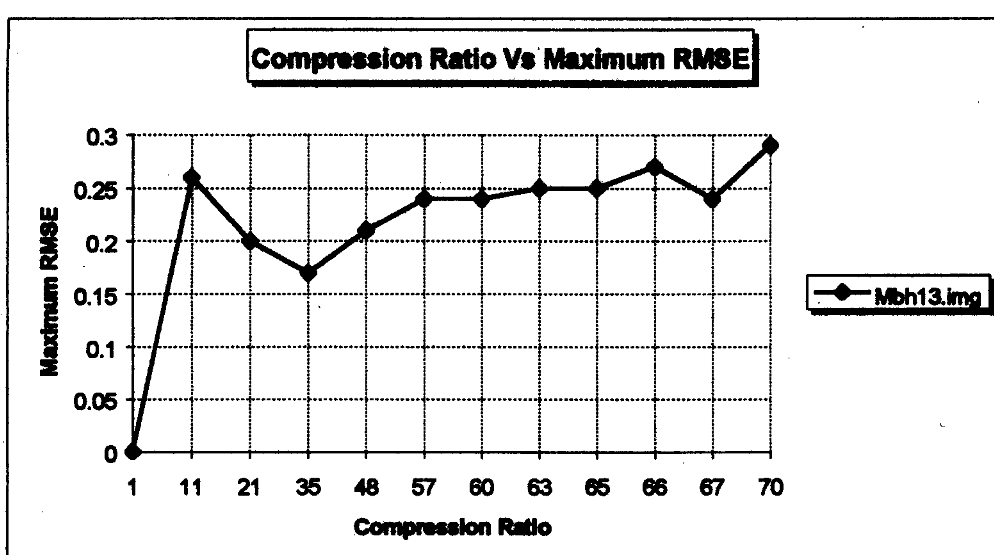


FIG A9: Compression Rate Versus Distortion for Mbh13 image using JPEG (DCT-based) Compression