

1-1-2004

Image Compression and its Effect on Data

Khaled S. Alkharabsheh

Follow this and additional works at: <http://mds.marshall.edu/etd>

 Part of the [Geophysics and Seismology Commons](#), [Graphics and Human Computer Interfaces Commons](#), and the [Physical and Environmental Geography Commons](#)

Recommended Citation

Alkharabsheh, Khaled S., "Image Compression and its Effect on Data" (2004). *Theses, Dissertations and Capstones*. Paper 441.

This Thesis is brought to you for free and open access by Marshall Digital Scholar. It has been accepted for inclusion in Theses, Dissertations and Capstones by an authorized administrator of Marshall Digital Scholar. For more information, please contact zhangj@marshall.edu.

IMAGE COMPRESSION AND ITS EFFECT ON DATA

**Thesis submitted to
The Graduate College of
Marshall University**

**In partial fulfillment of the
Requirements for the degree of
Master of Science
Physical Science**

GeoBioPhysical Modeling

by

Khaled S. Alkharabsheh

**Prof. Ralph Oberly, Committee Chairperson, Ph.D
Prof. James Brumfield, Ph.D**

Marshall University

Aug. 02, 2004

ABSTRACT
IMAGE COMPRESSION AND ITS EFFECT
ON IMAGES

By Khaled S. Alkharabsheh

This thesis is intended to define and study different image compression techniques, software programs, image formats (from early ones such as “GIF” to most recent ones such as “JPEG 2000”), compression effect on compressed data (compressed images), and its effectiveness and usefulness in reducing the file size and its transmission time, as a result.

In many GeoBioPhysical applications, some information inside any image may be the keys to solve different kinds of problems and classify features. This kind of data and information has to be handled with care; i.e. it’s not allowed to be lost during the compression process. On the other hand, dealing with images is more flexible in regular applications such as images used as pictures for simple purposes such as e-mails.

An un-compressed aerial image (DOQQ) of Huntington, WV. (with “.Tiff” extension) was taken as the original file to be compressed using different techniques and software programs. The results were studied and attached to each image. The resulting file size of each image was used to perform some comparisons between different software programs that were also used, trying to find the effectiveness of each technique and software from the quality to file size ratio point of view. Some previous work and research from

different references was also studied and discussed to show the differences and the similarities between this work and previous ones.

One of the goals of this study is to find the software program(s) and the compression types those give the best quality to file size ratio, and the ones that work best for GeoBioPhysical studies.

The results show that dealing with different types of imagery is sensitive and depends strongly on the application; the user has to know what he is doing. The user has to use the proper input imagery and compress them to the proper limits to get best results.

The results of this study show that JPEG2000 software programs (such as LuraWave) are very good and effective choices. JPEG2000 and ECW are likely to be extensively used in the near future for imagery and internet usage.

DEDICATION

To my parents and my teachers in Jordan and the US.

ACKNOWLEDGMENTS

I wish to thank Professors Ralph Oberly and James Brumfield for their help and support.

TABLE OF CONTENTS

ABSTRACT	ii
DEDICATION	iv
ACKNOWLEDGMENTS	v
TABLE OF CONTENTS	vi
List of Figures	1
1. CHAPTER I	1
INTRODUCTION	1
<i>Why compression?</i>	<i>1</i>
<i>Image Transmission and Compression Basics</i>	<i>2</i>
2. CHAPTER II	9
REVIEW OF LITERATURE	9
<i>Images, Computers, and the Internet</i>	<i>9</i>
<i>The Digital Image</i>	<i>10</i>
<i>Images stored in Computer memories</i>	<i>12</i>
<i>Image formats and Previous work performed in this field</i>	<i>15</i>
3. CHAPTER III	37
METHODS	37
<i>The un-compressed “TIF” image</i>	<i>37</i>
<i>JPEG (software used is “PaperPort LE7.0 by ScanSoft Inc”)</i>	<i>39</i>
<i>Comparisons</i>	<i>40</i>
<i>ECW using ER-Mapper software in MU Digital Image Processing Lab</i>	<i>62</i>
<i>Adobe Acrobat Reader</i>	<i>68</i>
<i>Adobe Photoshop</i>	<i>70</i>
<i>ECW Compressor</i>	<i>72</i>
<i>LuraWave Compressor</i>	<i>79</i>
4. CHAPTER IV	114
RESULTS	114
<i>Compression and data</i>	<i>114</i>
<i>What does compression do to the data?</i>	<i>115</i>
CHAPTER V	170
SUMMARY AND CONCLUSIONS	170
Bibliography	177
ATTACHMENTS	184

JPEG source code.....184

List of Figures

FIGURE 1-1 IMAGES FILES ARE RELATIVELY LARGE.....	3
FIGURE 1-2 REDUCTION BY ELIMINATING SOME PIXELS ACCORDING TO A SPECIFIC ORDER.....	6
FIGURE 2-1 ANALOG TO DIGITAL CONVERSION PROCESS	11
FIGURE 2-2 BIT PLANES	12
FIGURE 2-3 BIT LAYERS.....	13
FIGURE 2-4 EVERY PIXEL HAS A VALUE THAT CAN BE REPRESENTED BY 8 BITS; EACH BIT OF EVERY BYTE IS STORED IN A MEMORY CELL	14
FIGURE 2-5 A (65,536) COLOR PNG IMAGE	17
FIGURE 2-6 FIF IMAGERY	18
FIGURE 2-7 HOW JPEG ENCODING WORKS	20
FIGURE 2-8 A WEBPAGE WHERE THE USER CAN CHOOSE THE COMPRESSION RATIO OF THE RESULT	22
FIGURE 2-9 BMP CAN BE SAVED AND SHOWN WITH DIFFERENT NUMBER OF COLORS.....	25
FIGURE 2-10 THE LESS THE COLORS ARE, THE LESS THE SIZE AND THE INFORMATION THE IMAGE HOLDS WILL BE.....	26
FIGURE 2-11 MONOCHROME	27
FIGURE 2-12 (16) COLOR BITMAP (4BIT)FIGURE	28
FIGURE 2-13 (256) COLOR (8BIT).....	29
FIGURE 2-14 (24-BIT) BITMAP (16.777216*106 COLOR)	30
FIGURE 2-15 GIF VS JPEG	34
FIGURE 2-16 GIF WORKS BETTER FOR REGULAR NON-PHOTOGRAPHIC IMAGES	35
FIGURE 2-17 THE NUMBERS INSIDE THE ARRAY CELLS ARE NO MORE DISTINGUISHABLE	36
FIGURE 3-1.....	38
FIGURE 3-2.....	39
FIGURE 3-3 HUNTINGTON. TIFF. A CLOSER LOOK AT THE BRIDGE AND THE PIXILATION EFFECT	40
FIGURE 3-4 JPEG 10%	41
FIGURE 3-5 THE ABILITY TO OPEN A JPEG IMAGE USING LURAWAVE AND DETERMINE THE COMPRESSION RATIO	42
FIGURE 3-6 JPEG 90%	43
FIGURE 3-7 JPEG 85%	44
FIGURE 3-8 JPEG 80%	45
FIGURE 3-9 JPEG 75%	46
FIGURE 3-10 JPEG 70%	47
FIGURE 3-11 JPEG 65%	48
FIGURE 3-12 JPEG 65% (ZOOMED IN)	49
FIGURE 3-13 JPEG 60%	49
FIGURE 3-14 JPEG 55%	50
FIGURE 3-15 JPEG 50%	51
FIGURE 3-16 JPEG 45%	52

	2
FIGURE 3-17 JPEG 40%	53
FIGURE 3-18 JPEG 35%	54
FIGURE 3-19 JPEG 30%	55
FIGURE 3-20 JPEG 25%	56
FIGURE 3-21 JPEG 20%	57
FIGURE 3-22 JPEG 15%	58
FIGURE 3-23 JPEG 10%	59
FIGURE 3-24 THIS SOFTWARE DOESN'T LET THE USER ENTER VALUES LESS THAN 10%	60
FIGURE 3-25 ECW; THEORETICAL COMPRESSION RATIO= 30 TIMES	63
FIGURE 3-26 PROCESS SUMMARY FOR THE 30-TIMES COMPRESSED ECW OUTPUT	63
FIGURE 3-27 ECW; THEORETICAL COMPRESSION RATIO= 99 TIMES	65
FIGURE 3-28 PROCESS SUMMARY FOR THE 99-TIMES COMPRESSED ECW	66
FIGURE 3-29 PDF IMAGES	68
FIGURE 3-30 ADOBE ACROBAT READER DEALING WITH IMAGERY	69
FIGURE 3-31 ADOBE PHOTOSHOP.....	70
FIGURE 3-32 PHOTOSHOP USER CAN REDUCE HIS PICTURES USING MANY OPTIONS.....	71
FIGURE 3-33 ECW WITH THEORETICAL COMPRESSION RATIO= 30 TIMES	72
FIGURE 3-34 ECW WITH THEORETICAL COMPRESSION RATIO= 35 TIMES	74
FIGURE 3-35 ECW COMPRESSED WITH THEORETICAL COMPRESSION RATIO= 90 TIMES	75
FIGURE 3-36 THIS SOFTWARE CAN ESTIMATE THE OUTPUT FILE SIZE BEFORE EVEN THE PROCESS STARTS.....	76
FIGURE 3-37 THE COMPRESSION PROCESS.....	76
FIGURE 3-38 COMPRESSION RESULTS SUMMARY	77
FIGURE 3-39 ECW PRE-COMPRESSION ESTIMATIONS	79
FIGURE 3-40 WITH LURAWAVE, THE USER HAS MANY OPTIONS TO SAVE HIS IMAGE	80
FIGURE 3-41 PACKBITS COMPRESSION.....	81
FIGURE 3-42 PACKBITS COMPRESSION.....	82
FIGURE 3-43 HUNTINGTON10.....	83
FIGURE 3-44 50% JPEG OBTAINED BY LURAWAVE.....	84
FIGURE 3-45 20% JPEG.....	85
FIGURE 3-46 10% JPEG.....	86
FIGURE 3-47 HUNTINGTON10.....	87
FIGURE 3-48 50% OF HUNTINGTON10.....	88
FIGURE 3-49 THE ABILITY OF LURAWAVE TO VIEW IMAGES WITH DIFFERENT QUALITY SELECTIONS	89
FIGURE 3-50 THE ABILITY OF LURAWAVE TO SAVE IMAGES WITH DIFFERENT QUALITY AND FILE SIZE SELECTIONS	90
FIGURE 3-51 LOSSLESS COMPRESSION FROM TIF TO JP2	91
FIGURE 3-52 10% QUALITY	92
FIGURE 3-53 20% QUALITY	93
FIGURE 3-54 10K. COMPRESSION RATIO= 1560.....	94
FIGURE 3-55 6KB. COMPRESSION RATIO= 2601	95
FIGURE 3-56 4K. COMPRESSION RATIO= 3900.....	96
FIGURE 3-57 2K. COMPRESSION RATIO= 7799.....	97
FIGURE 3-58 A 10KB IMAGE. COMPRESSION RATIO= 1560. INPUT IS HUNTINGTON10	98

FIGURE 3-59 6K IMAGE. COMPRESSION RATIO= 2600. INPUT IS HUNTINGTON10	99
FIGURE 3-60 4K IMAGE. COMPRESSION RATIO= 3897. INPUT IS HUNTINGTON10	100
FIGURE 3-61 2K IMAGE. COMPRESSION RATIO= 7807. INPUT IS HUNTINGTON10	100
FIGURE 3-62 LURAWAVE CAN THE COMPRESSION RATIO OF ANY JPEG IMAGE.....	101
FIGURE 3-63 THE MORE THE IMAGE IS COMPRESSED, THE LONGER IT WILL TAKE TO OPEN IT	102
FIGURE 3-64 JP2 IS DESIGNED FOR INTERNET USAGE.....	103
FIGURE 3-65 THE 10KB JP2 IMAGE	104
FIGURE 3-66 THE JP2 ICON	105
FIGURE 3-67 THE IMAGE IS SAVED AGAIN AS TIFF. FILE SIZE= 15.2MB AGAIN.	105
FIGURE 3-68 THE ABILITY TO SAVE THE IMAGE AGAIN USING INTERNET EXPLORER	106
FIGURE 3-69 THE TIFF IMAGE	107
FIGURE 3-70 TIFF FILE COMPRESSED 200 TIMES TO JP2. FILE SIZE= 77.9KB	108
FIGURE 3-71 TIFF FILE COMPRESSED 900 TIMES TO JP2.....	109
FIGURE 3-72 100 TIMES.	110
FIGURE 3-73 RETRIEVING THE FILE SIZE OF THE ORIGINAL IMAGE BY VIEWING THE PROPERTIES OF ANY JP2 IMAGE	111
FIGURE 3-74 200 TIMES	112
FIGURE 3-75 900 TIMES.	113
FIGURE 4-1 IMAGE STATISTICS.....	116
FIGURE 4-2 CLASSIFICATION	117
FIGURE 4-3 CHANGING PIXEL VALUES CAN CHANGE THE RESULTS OF THE CLASSIFICATION PROCESS AND DECISIONS AS A RESULT	118
FIGURE 4-4 BUILDINGS IN TIFF IMAGE.....	120
FIGURE 4-5 HISTOGRAM FOR TIFF.....	121
FIGURE 4-6 SCATTERGRAM FOR TIFF.....	122
FIGURE 4-7 LOCATION OF THE RIVER AREA ON THE CORRESPONDING SCATTERGRAM	123
FIGURE 4-8 THE X-AXIS REPRESENTS THE RED BAND WHILE THE Y-AXIS REPRESENTS THE GREEN ONE.....	124
FIGURE 4-9 STARTING THE PROCESS OF STATISTICS CALCULATIONS	126
FIGURE 4-10 STATISTICS CALCULATION PROCESS.....	126
FIGURE 4-11 HISTOGRAM FOR 95% JPEG	129
FIGURE 4-12 SCATTERGRAM FOR 95% JPEG.....	130
FIGURE 4-13 HISTOGRAM FOR 85% JPEG	132
FIGURE 4-14 SCATTERGRAM FOR 85% JPEG.....	133
FIGURE 4-15 HISTOGRAM FOR 75% JPEG	135
FIGURE 4-16 SCATTERGRAM FOR 75% JPEG.....	136
FIGURE 4-17 HISTOGRAM FOR 65% JPEG	138
FIGURE 4-18 SCATTERGRAM FOR 65% JPEG.....	139
FIGURE 4-19 HISTOGRAM FOR 50% JPEG	141
FIGURE 4-20 SCATTERGRAM FOR 50% JPEG.....	142
FIGURE 4-21 HISTOGRAM FOR 30% JPEG	144
FIGURE 4-22 SCATTERGRAM FOR 30% JPEG.....	145
FIGURE 4-23 HISTOGRAM FOR 20% JPEG	147
FIGURE 4-24 SCATTERGRAM FOR 20% JPEG.....	148

FIGURE 4-25 10% JPEG	151
FIGURE 4-26 HISTOGRAM FOR 10% JPEG	152
FIGURE 4-27 SCATTERGRAM FOR % JPEG.....	153
FIGURE 4-28 STD. DEV. VARIATIONS FOR JPEG IMAGERY.....	156
FIGURE 4-29 HISTOGRAM FOR 30 TIMES COMPRESSED ECW	159
FIGURE 4-30 SCATTERGRAM FOR 30 TIMES COMPRESSED ECW.....	160
FIGURE 4-31 HISTOGRAM FOR 99 TIMES COMPRESSED ECW	162
FIGURE 4-32 SCATTERGRAM FOR 99 TIMES COMPRESSED ECW.....	163
FIGURE 4-33 JPEG 10%	165
FIGURE 4-34 ECW99	166
FIGURE 4-35 SCATTERGRAM FOR HUTNOGTON10.....	167
FIGURE 4-36 SCATTERGRAM FOR 99 TIMES COMPRESSED ECW.....	168

1. CHAPTER I

Introduction

Why compression?

The most important task of image compression is to easily use, manage and transmit the continuously growing high quality image databases we have with limited equipment & funding.

Data and information –in all their types and kinds- are much more valuable and useful when they can be transmitted and stored easier and faster in smaller spaces and with less cost. The need to fit more and more data into smaller spaces where massive amounts of data are being accessed all the time and from all around the world and where bandwidth costs money is an active field of study. Competition in data distribution and sales also boost studies in this field.

Images are considered to be one form of data. The age of data compression is almost the age of the computer itself. Data compression in all its different forms is well known to computer users. Some well-known techniques and standards are ZIP and RAR. Some are more effective than others, while others are more useful on certain kinds of data and in certain applications.

Developments in computers, communications, communication electronics, data acquisition, and imaging systems have progressed together. Images are now recorded,

transmitted, and displayed in larger sizes (dimensions), better quality, and with more colors. The ability to display high quality imagery with millions of colors (24-bits) has resulted in images and video files becoming larger in size. For this reason, we started looking at compression to lessen the file size and transmission time of good quality mages.

Image Transmission and Compression Basics

It became feasible to work with large un-compressed images and Orthophoto mosaics for entire cities.

As soon as computers (and their monitors and video cards) became capable of displaying more than 16 colors, pictures and images entered the field of communication strongly carrying more and more information than text can carry; one picture can explain more than thousands of words sometimes. Further developments introduced video storage and transmission before such kinds of information started flowing everywhere in the “**Internet**”. Unfortunately, images and video files (videos are successive images shown at specific speeds) are relatively larger than other types of information and data files; they need more storage space and larger transmission bandwidths.

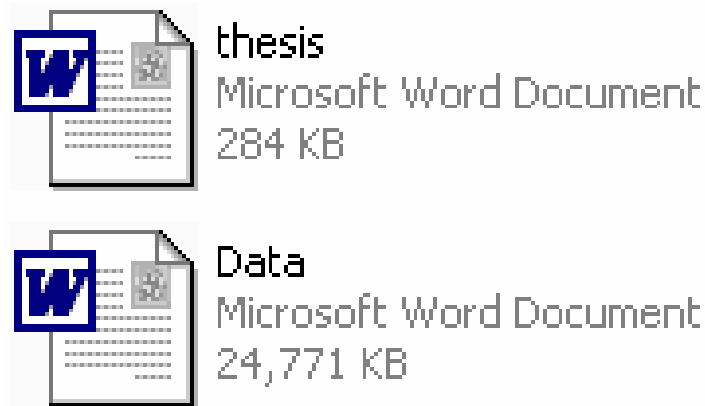


Figure 1-1 Images files are relatively large compared to text ones

Fig. 1.1 shows how large images can be. While working on this study and after adding just some images to that text, the file became very large and, the system working on it became slow. This required saving the images and data separately. I wrote the text that was much smaller and faster before I added the data and the images using a much faster computer. The size shown for the data file was at a very early stage; the final saved one was much larger! This indicates how large imaging files are and how fast the computers processing them must be.

Szuprowics (1992) states that "TV-quality video requires 720 kilobytes per frame (kbpf) displayed at 30 frames per second (fps) to obtain a full-motion effect, which means that one second of digitized video consumes approximately 22 MB (megabytes) of storage. A standard CD-ROM disk with 648 MB capacity and data transfer rate of 150 KBps could

only provide a total of 30 seconds of video and would take 5 seconds to display a single frame." (From the research performed by "Chua Lye Heng" in the 19th of May 1997).

Source: <http://pascalzone.amirmelamed.co.il/Graphics/JPEG/JPEG.htm> retrieved on June 26, 2004.

Images are useful kinds of data and the information often has to be shared and transmitted electronically; unfortunately, images are large in size and consume much memory and transmission bandwidths. The developments in the internet and data transmission all over the globe that becomes a small village boosted the need to transmit and store smaller and clearer (but still high quality and useful) images since internet surfers always prefer not to wait long for an image to load.

Trying to get over this problem and get better, less cost, and smaller size images transmitted and stored more efficiently, engineers and technicians developed several techniques to "compress" images, especially digital ones, during the last few decades.

Different software programs were compiled using several programming languages and different techniques to compress images according to some physical criteria such as colors and the ability to mix or confuse more than one color (see GIF, chapter 2).

Mathematical criteria such as **Fourier Transform**, or human criteria such as the characteristics of the human eye and brain (see JPEG, chapter 2)., are also factors.

The first format to gain widespread acceptance was "**CompuServe's**" GIF (Graphic Interchange Format) standard. Unfortunately, GIF was designed to display 8-bit graphics (256 colors) while modern computers are capable of displaying up to 16 million colors.

The next step was JPEG (or JPG). JPG format was designed by the “**Joint Picture Experts Group**”. It is capable of displaying 24-bit (more than 16 million colors).

Many developments have occurred in the last few decades in the field of GeoBioPhysical studies. Images of special formats and characteristics started to be widely used. ER- Mapper became a leader in this field; GeoBioPhysical images and data are relatively larger and need more compression with less data loss; as a result, ECW compression was found to serve this field specifically; ECW was designed to lessen the artifacts in GeoBioPhysical images in a special way that serves in this field (Fuzziness in an image or video caused by compression is called an artifact).

Image compression techniques are either lossy or lossless. *Lossy compression is any compression process that causes some of the data in the image to be lost; such lost data can not be recovered by any later de-compression process. Lossless compression is any compression process that doesn't cause any loss in the data; it just lessens the size of the image or the file.*

Being lossy or lossless, depends on the relations between the un-compressed (input) data (image) and the output data, and the ability to recover the original one from the output (when un-compressing the output). This is discussed in chapter 2.

As stated above, many software programs have been used for the purposes of compression and recovery before display; all of which use either lossy or lossless.

A combination of both is being used by some programs (such as JPEG) as well. This is also discussed in chapter2.

A digital image is an array of pixels where each pixel is represented by a byte of 8, 16, 24, 32, and maybe more bits. The number of bits per byte represents the maximum number of colors the systems can display by the relation 2^n where “n” is the number of bits. The less the number of pixels or colors per pixel to display are; the smaller the file size and the faster transmission will be.

Figure 1.2 shows how some earlier compression techniques lessen the number of pixels to display by eliminating some rows or columns from the array.

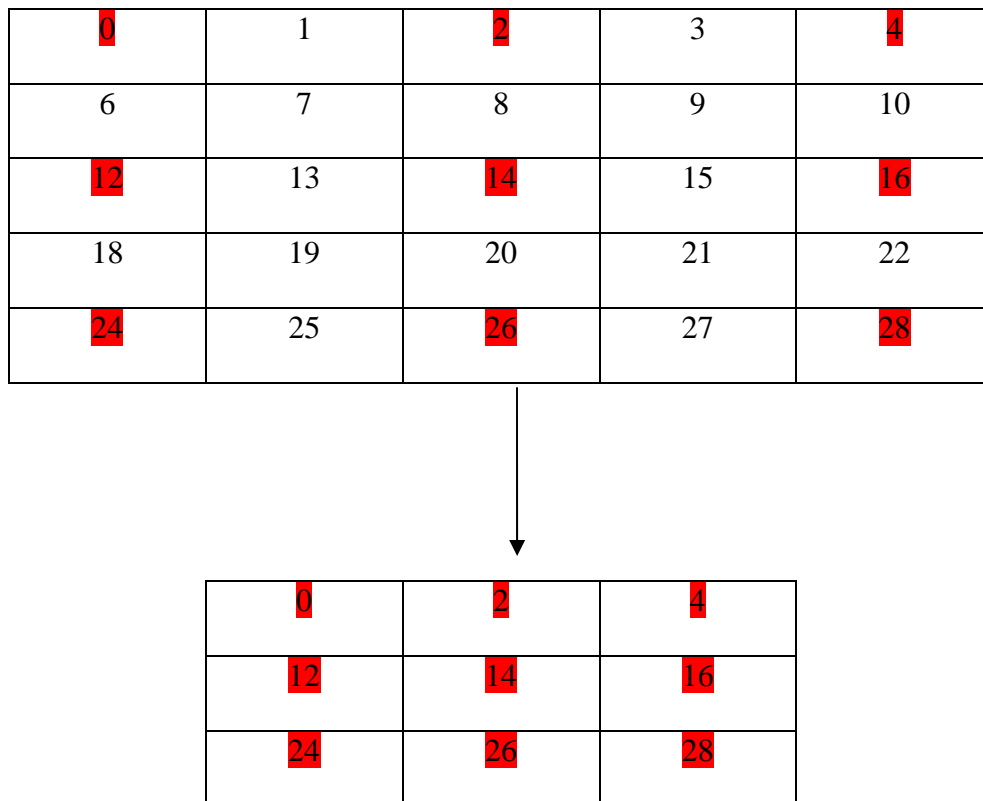


Figure 1-2 Reduction by eliminating some pixels according to a specific order

Jensen, R. J. (1996). Introductory Digital Image Processing. New Jersey. Prentice Hall, Inc.

Others (such as GIF) lessen the number of colors by **lessening the length** of the word that represents the brightness value (trying to keep all other kinds of data as is in a lossless way); and others (such as JPEG, JPEG2000, and ECW) use special sophisticated techniques that mix many possible kinds of lossy and lossless techniques according to some specific standards; however, in this case images have to be decoded at the destination according to some related techniques that “tell the decoder” how the image was compressed.

The brightness value of every pixel is stored in a memory row; the number of rows determines the number of possible pixel values that can be stored while the number of columns determines the length of the stored word. In the next chapter, we will find how “GIF” compression tries to lessen the image size by lessening the number of possible colors.

Recently, ER-Mapper has introduced a relation between its ECW file extension and JPEG especially JPEG2000, in several software programs. This is very beneficial for those who work in fields of image processing, mapping, GeoBioPhysical modeling, satellite imaging, or any other fields related to images. The benefits are greater taking into consideration that some related software programs are going to be offered free of charge and even with much more flexibility; the amazing thing is that technicians can even get the source code for some programs; change and develop them according to their needs and for the purpose of developing the program!

Developers of JPEG and JPEG2000 offer a wide range of standardized programs that can compress the image according to some international standards (Note that there are many companies that develop JPEG and JPEG200 separately). The good thing is that such programs have been working well since JPEG was introduced; they have been developed continuously since the standards were written by the “**Joint Picture Experts Group**” meanwhile those standards are becoming common and popular worldwide especially for internet purposes.

Chapter 2 discusses the basics of image compression from a technical point of view. TIF, GIF, PNG, FIF, JPEG (JPG), JPEG2000, and ECW are discussed. Previous work and results in this field cannot be ignored since they show useful results that support this study. They are discussed and studied in chapter 2 in a separate section.

2. CHAPTER II

Review of Literature

Images, Computers, and the Internet

“You point, you click; you drag and you drop. Files open and close in separate windows. Movies play, pop-ups pop, and video games fill the screen, immersing you in a world of 3-D graphics. This is the stuff we're used to seeing on our computers. It all started in 1973, when Xerox completed the “Alto”, the first computer to use a graphical user interface. This innovation forever changed the way the people work with their computers”. (Robert Valdes. 2001).

He also wrote:

“In 1996, Intel introduced AGP as a more efficient way to deliver the streaming video and real-time-rendered 3-D graphics that were becoming more prevalent in all aspects of computing. Previously, the standard method of delivery was the Peripheral Component Interconnect (PCI) bus.....This system worked well for many years, but eventually the PCI bus became a little long in the tooth. The Internet and most software were more and more graphically oriented, and the demands of the graphics card needed priority over all other PCI devices”.

This shows that designers and engineers always try to design new systems to make life easier. Robert Valdes states that the need to “deliver streaming video and real-time-rendered 3-D graphics on which the Internet and most software programs became more and more dependable” triggered the production of the AGP (Accelerated Graphics Port) which loads and communicates graphics faster and more efficiently with the CPU and the surroundings. But, what about getting further benefits from “Squeezing” and lessening the size of every image that this card has to deal with? This is the basic idea of image compression that will be studied later.

The Digital Image

Information and signals in nature (including images) are analog by default. Trying to save time and space while getting faster transmission of better quality images which can be understood and treated by computers, man could design new techniques and devices to sample images and video at specific **sampling rates** and convert them to the equivalent digital form that can be treated and altered using **Boolean Algebra** techniques. This conversion can be performed using the Analogue to Digital Converter “ADC”. This results in an image composed of digital data in a binary form.

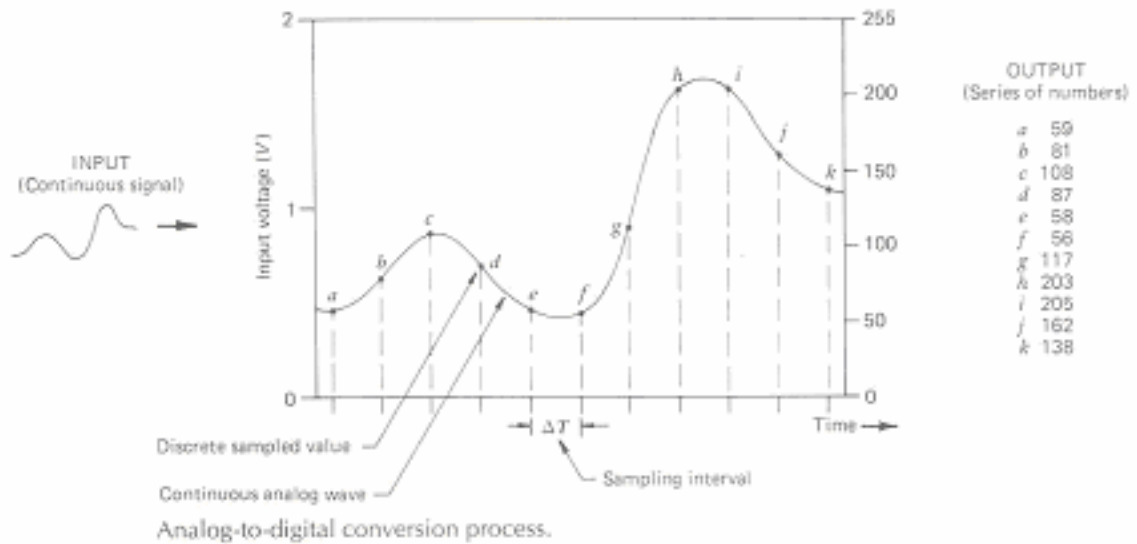


Figure 2-1 Analog to Digital conversion process

Spasov, P. (1999). Microcontroller Technology. New Jersey. Prentice Hall, Inc.

Fig. 2.1 indicates how a sampling signal of a specific frequency samples the analog one and changes the continuous pattern to another discrete one. This binary signal; however, has to be converted back to its normal analog form (using a Digital to Analog Converter “DAC”) before it can be displayed and understood by humans (this is not the case in some new display systems that can deal directly with digital information such as the Liquid Crystal Displays “LCD’s” and Diode Displays).

Digital image forms are created and preferred because digital computers can only understand, deal with, and process digital information and data. Digital images, besides being processed easier and faster by computers, can be stored much easier in smaller spaces; they also can be stored and transmitted much cheaper than analog images.

Images stored in Computer memories

Taking the picture, image acquisition systems convert the intensity of each pixel in the scene to an equivalent **Brightness Value (BV)** that can be “**Digitized**” to produce a “**Bit-Plane**”. A bit-Plane is a two dimensional array with BV’s for the elements at each position.

In a CCD (charge coupled device), the area array records the intensity of every pixel in the bands of interest (Red, Green, and Blue for a RGB picture for example), each with a specific number of bits (possible colors) forming the “Byte” that represents that intensity and then transfers it to the electronics of the system to be processed and/or stored.

An image is considered to be a 2-dimensional array of pixels and is called a “Bit Plane”.

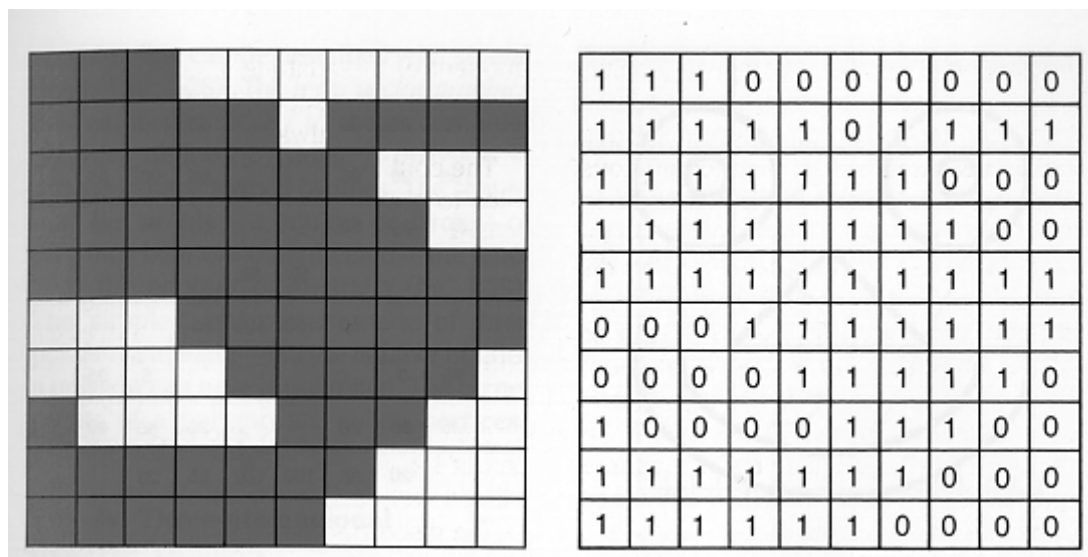


Figure 2-2 Bit planes

Source: <http://caad.arch.ethz.ch/teaching/nds/ws98/computing/image/>

Fig. 2.2 shows an image and its bit plan. Bit planes align in the third dimension to form **Bit Layers** for each color in the image; this gives the ability to store the **value** of **every bit** of every **Byte** (that represents the binary value of the corresponding **Pixel**) for **every color** (or **Band**). Images are represented and stored in a three-dimensional pattern for each color. Those dimensions are related to the terms stated above; they are the horizontal axis, the vertical axis, and the value of the each pixel (the value of the byte). In the case where successive pictures or images are used to form video or interlaced presentations; there will be one more dimension; that is time. Compression can be applied to any of those dimensions.

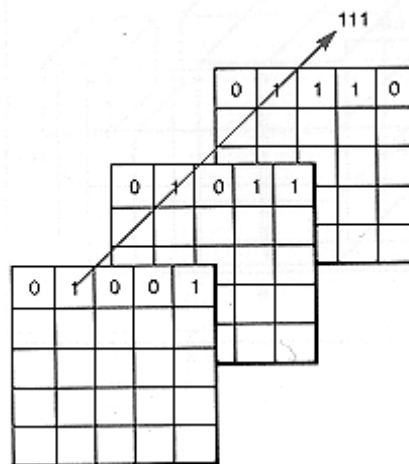


Figure 2-3 Bit layers

Fig. 2.3 shows how images are stored in computer memories as **bit layers**. After an image is acquired, the signal gets filtered by a photo or an electronic spectrometer to

separate the signals to its different band signals; then each band gets stored separately as bit layers and planes.

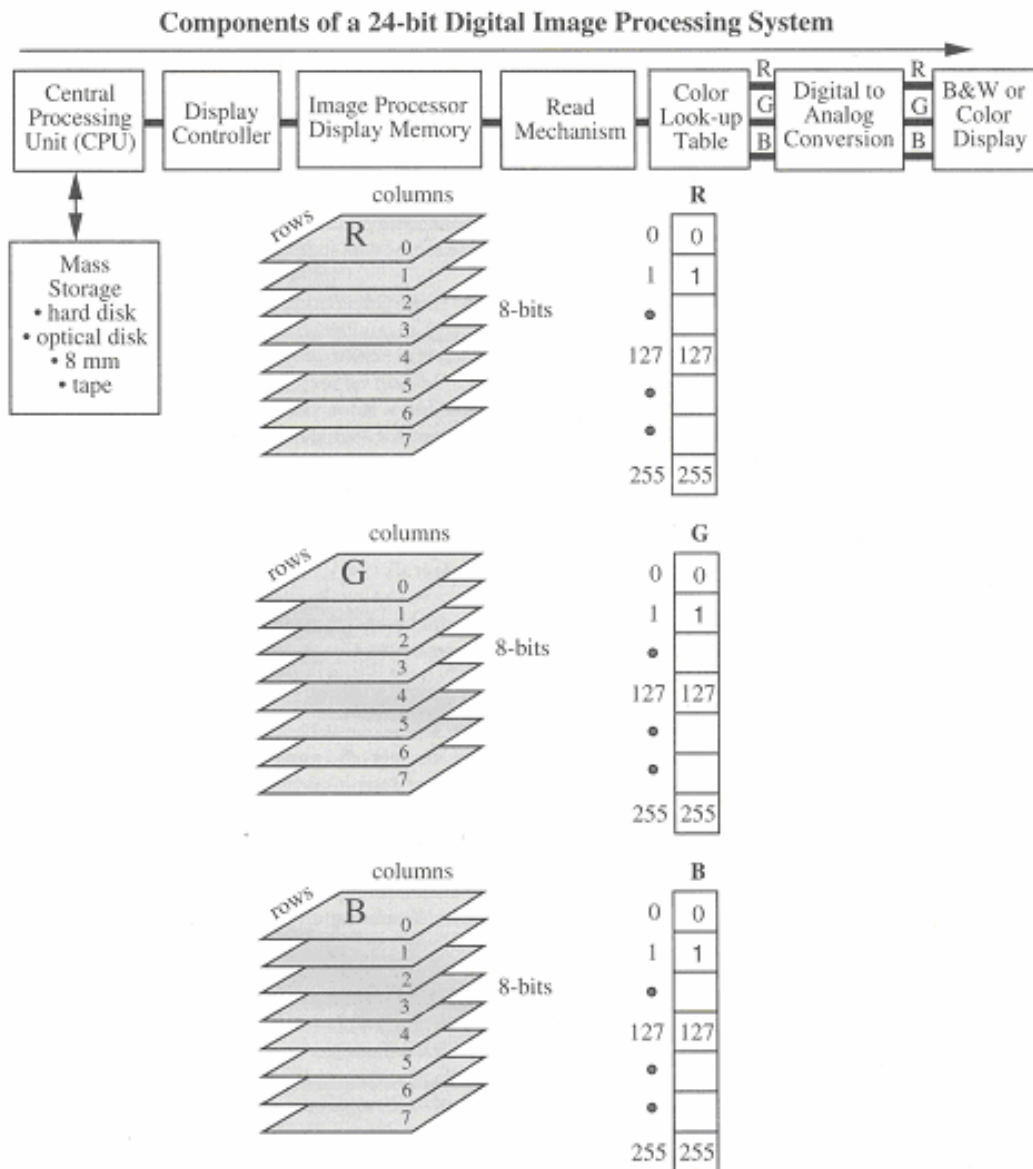


Figure 2-4 Every pixel has a value that can be represented by 8 bits; each bit of every byte is stored in a memory cell

Jensen, R. J. (1996). Introductory Digital Image Processing. New Jersey. Prentice Hall, Inc.

Fig. 2.4 shows the idea according to which 3-band (RGB) images are stored digitally in digital memories and the hardware needed to retrieve and display it.

Image formats and Previous work performed in this field

There are many image formats and standards used for imaging purposes. The most common are **TIF, GIF, PNG, FIF, BMB, JPEG (JPG), JPEG2000, and ECW**. The following is a discussion of each one.

GIF (Graphics Interchange Format)

This was invented by CompuServe Inc. GIF Stores 8-bit images and uses LZW compression. GIF supports interlacing and transparency of images. GIF has some drawbacks; it uses 8-bit images.... This means that it can only show up to 255 colors. In 1994 developers of GIF-supporting software programs faced another problem, an LZW patent. This was the reason why a new more advanced format (PNG) was created by an internet working group. PNG even supports more colors.

References for information about this section:

http://www.wfu.edu/~matthews/misc/jpg_vs_gif/JpgCompTest/JpgChromaSub.html.

Retrieved on July, 08 2004.

http://www.webopedia.com/DidYouKnow/Internet/2002/JPG_GIF_PNG.asp. Retrieved on July, 11 2004.

PNG (Portable Network Graphics)

PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Sample depths range from 1 to 16 bits per component (up to 48 or 64 bit).

“In the late 1980's CompuServe introduced a new image format which became popular with internet users - Graphic Interchange Format, or GIF. It used a long-established compression algorithm, LZW, without realizing that LZW was subject to a patent filed by the Sperry corporation, which was later taken over by Unisys.

In late 1994 Unisys dropped a bombshell by announcing that the company would start demanding royalties from all developers of GIF-supporting software. The announcement caused consternation among the internet community and ideas were passed around for a non-proprietary replacement. An informal internet working group was established by Thomas Boutell to devise an image format which would be even better than GIF. In October 1996, the World Wide Web Consortium adopted the new specification, Portable Network Graphics, as a recommended image format”. Retrieved from <http://www.webcolors.freeseerve.co.uk/png/> on July, 08 2004.

Fig. 2.5 shows a PNG image contains 65,536 different colors. The file size is 678 bytes!

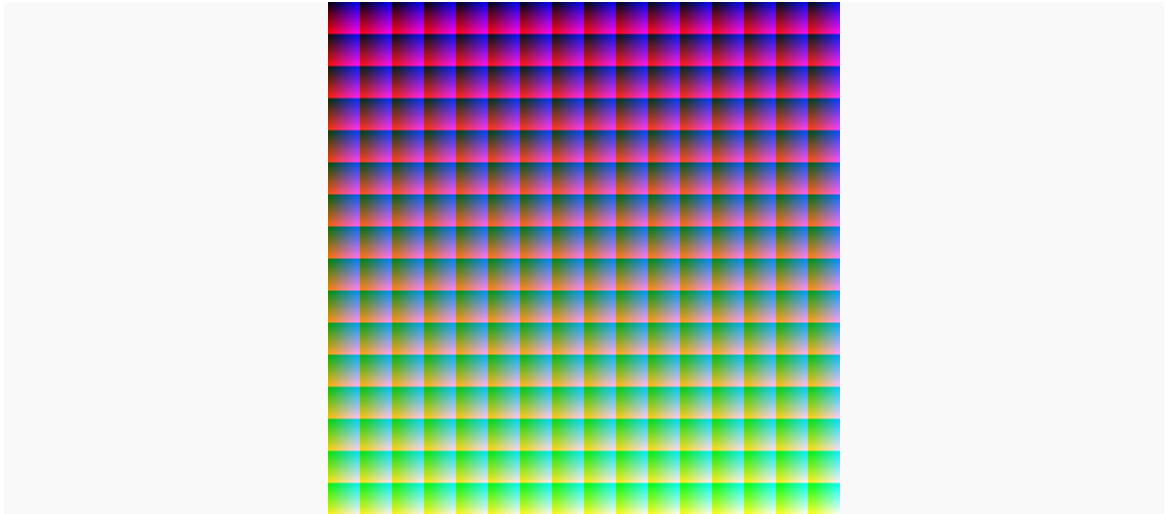


Figure 2-5 A (65,536) color PNG image

<http://www.webcolors.freeseerve.co.uk/png/png24.htm>. Retrieved on July, 08 2004.

Reference for information about this section: <http://www.webcolors.freeseerve.co.uk/png> (July, 08 2004),

http://www.webopedia.com/DidYouKnow/Internet/2002/JPG_GIF_PNG.asp (July, 11 2004), and <http://www.webcolors.freeseerve.co.uk/png/png24.htm> (July, 08 2004).

FIF (Fractal Image Format)

FIF is a graphics file format from Iterated Systems, Inc. that uses fractal geometry to compress images. (Fractals are structures possessing similar-looking forms of many different sizes. They can be used to create any real-world object, like a mountain or a

cloud, provided it doesn't correspond to a simple geometric shape.) These fractals can be expressed in mathematical terms, allowing an entire image to be recorded as repeated

patterns. This can reduce the file size to 1/100th of a pixilated version. Fig. 2.6 shows the zooming power of the FIF.

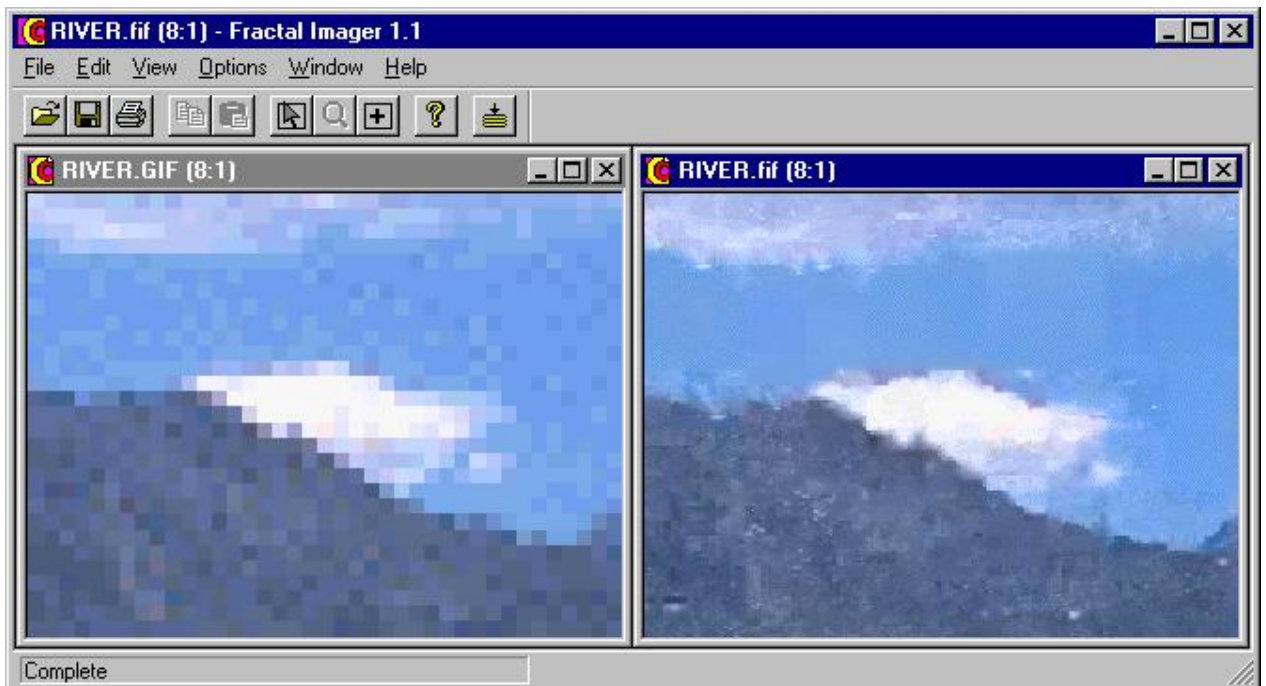


Figure 2-6 FIF imagery

<http://users.senet.com.au/~mjbone/Fractals.html>

Reference for information about this section:

<http://users.senet.com.au/~mjbone/Fractals.html>. Retrieved on July, 08 2004.

JPEG

JPEG benefits from the characteristics of the human eye and brain and the way they see a picture; human eyes are more sensitive to brightness variations than to hue variations.

JPEG itself refers to a standard for data compression and the organization that developed the standard not to a file format. There are two different kinds of JPEG; lossy and lossless; the lossy one is better known and more common. Lossless JPEG is being used but lossy is used worldwide. It is the one used with digital cameras and the internet. It uses encoding to remove any data it deemed irrelevant or unseen. JPEG compression programs allow varying degrees of compression and file size to quality ratio.

While reading about JPEG, I faced many different techniques used to lessen the data in the image such as:

1. Averaging arrays of 8*8 or 2*2 pixels and assigning the brightness value to a specific one that replaces the whole array.
2. Using encoding; using such techniques, the compressing software tries “to say” the existing information in other “more summarized” ways.

Looking at fig 2.7 (taken from a previous research. Webpage:

<http://www.geocities.com/thesciencefiles/compression/info.html>), we can understand the theory of JPEG. Look at a narrow strip near the center, containing white and red pixels:



Magnifying the strip, we get:

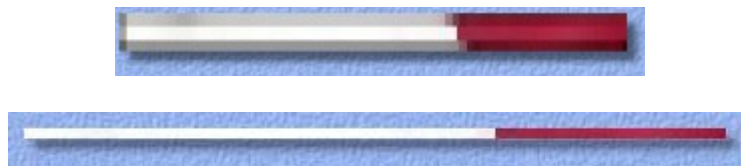


Figure 2-7 How JPEG encoding works

Suppose this row contains n number of pixels. A bitmap image would need to store $3*n$ pieces of information about this strip. “A JPEG image compressing software looks at the strip and discovers that there are 50 pixels in a row that are approximately the same shade of white, and another 20 pixels that are approximately the same color of red. It writes an instruction for the computer to store that says '**make 50 whites and 20 reds, in this location**'. This contains 6 pieces of information; the two colors, the starting positions, and the length of each color. Doing this for the whole image will greatly reduce the amount of space needed to store the file.

Compressing an image this way causes it to deteriorate. Higher compression rates (which you can set using the program you are editing the images with) will cause the computer to be less precise about how it matches up the colors, and a string of many different shades of one color might be replaced by just three strings of three different shades. This will make the colors in the image less precisely matched than they were

originally". Retrieved on July 09, 2004 from

<http://www.geocities.com/thesciencefiles/compression/info.html>.

3. Chroma Subsampling

Chroma subsampling is a method that stores color information at lower resolution than intensity information. Many compression programs perform 2x2 chroma subsampling, which breaks the image into 2x2 pixel blocks and only stores the average color information for each 2x2 pixel group.

Reference for information about Chroma Subsampling:

http://www.wfu.edu/~matthews/misc/jpg_vs_gif/JpgCompTest/JpgChromaSub.html.

Retrieved on July, 08 2004.

The following example (fig. 2.8) shows the effects of JPEG compression at various levels of quality. The shown images use "quality" factors of 100 (best) down to 5 (nearly the worst.) and therefore require space in the range 326321 down to 9438 bytes. Note here that those numbers do **NOT** represent the compression ratio; in this case, quality becomes higher with a higher number (opposite). Referring to that webpage (<http://www.cs.sfu.ca/CourseCentral/365/li/material/cgi-bin/whichjpeg.cgi>), you can select the image with the quality factor you want to view. Some other websites have compression wizards where the user can compress images of his own.

	
Factor: 10 Size=15325 bytes 5 10 25 50 75 100	Factor: 5 Size=9438 bytes 5 10 25 50 75 100
	
Factor: 50 Size=46295 bytes 5 10 25 50 75 100	Factor: 100 Size=326321 bytes 5 10 25 50 75 100
Number of Images Displayed: 1 2 4	

Figure 2-8 A webpage where the user can choose the compression ratio of the result

<http://www.cs.sfu.ca/CourseCentral/365/li/material/cgi-bin/whichjpeg.cgi>. Retrieved on June, 08 2004.

Why convert GIF to JPEG?

There are two good reasons to convert GIF to JPEG which are:

- a) To make your image files smaller.
- b) To store 24-bit-per-pixel color data instead of 8-bit-per-pixel data.

JPEG 2000

JPEG2000 was recently released. It is based on JPEG technology but with better file size to quality ratios and better compression for the same quality (see chapter 3 for results and comparisons). JPEG2000 also supports many things which were not supported (or less supported) by JPEG; for example, JPEG2000 is being developed to support data files such as what is called “MetaData”. Actually, JPEG2000 gave JPEG some ability to work on GeoBioPhysical and satellite imagery, and started to interact with ECW through ER- Mapper. JPEG and JPEG2000 are standards. Their supporting software programs are being written and sold by many different companies and firms worldwide; unfortunately, **those programs are not all the same.**

ECW (ER Mapper Compressed Wavelet)

ECW is an imagery compressed format created using the ER Mapper Compression Wizard. There are various compression techniques. The most effective technique is the **wavelet based**, which is a method of processing, quantizing and then encoding the image in a way that results in very high levels of compression, with an almost perfect reconstruction of the original image.

ER Mapper 6.1 offers wavelet based compression technology to compress images, resulting in very high quality results at very high compression rates. Typically a color image such as an airphoto can be compressed to less than 2% to 5% of its original size (50:1 to 20:1 compression ratio).

Image Compression

Khaled S. Alkharabsheh

At 50:1 compression, 100GB (100*1024 MB) of color imagery will compress down to

2G (2*1024MB).

Reference for this section:

<http://www.gisdevelopment.net/technology/ic/techip0003pf.htm>

BMP (Microsoft Windows Bitmap)

It is clear that this term is close to what we called previously “Bit Planes”. Bitmap is based on the same theory where every single band in the image is represented by uncompressed “Bit Layers” of “Bit Planes”. Bitmap is the main format supported by Microsoft Windows. It stores 1-, 4-, 8-, and 24-bit images. The following sequence of figures was noticed while working on a figure for this study; trying to save it, I found many different ways to save it with different qualities and file sizes.

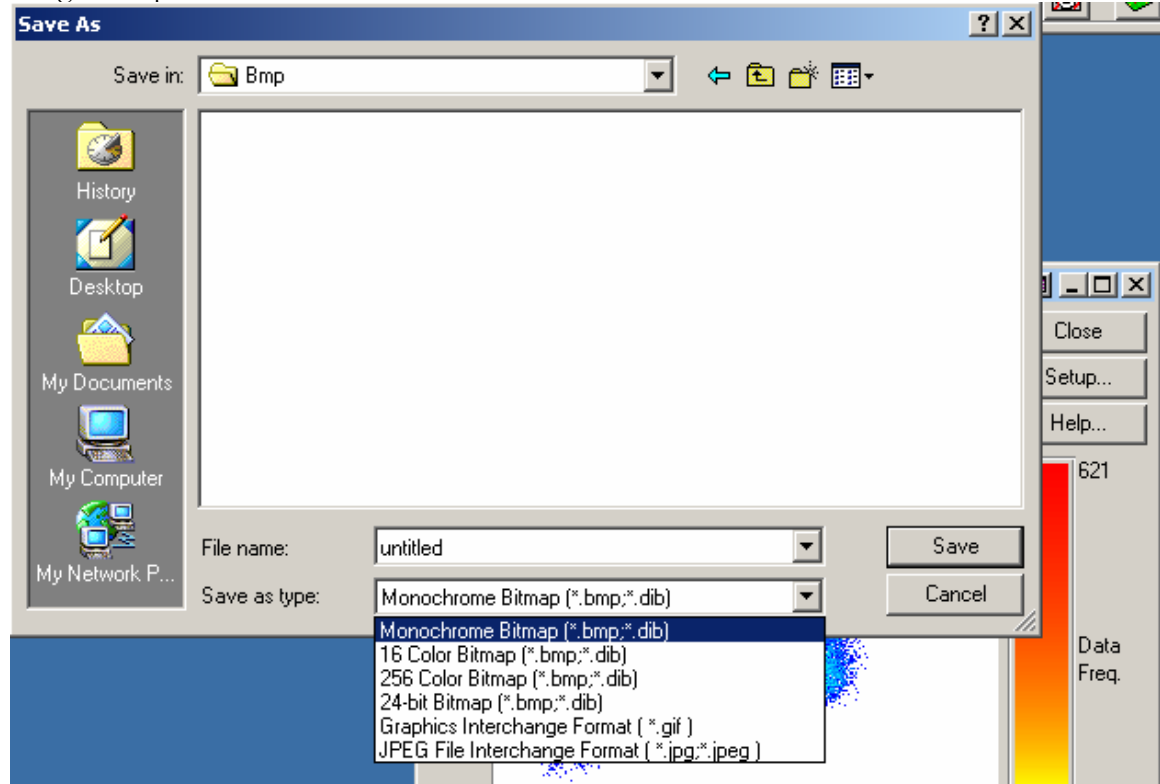


Figure 2-9 BMP can be saved and shown with different number of colors

Fig. 2.9 shows how an image can be saved in different extensions, and so, different file sizes using windows. If you choose to save your image as a “.bmp”, you still have choices in terms of the possible displayed colors; Monochrome (fig. 2.11), 16 color (fig. 2.12), 256 colors (fig. 2.13), and 24-bit (fig 2.14).

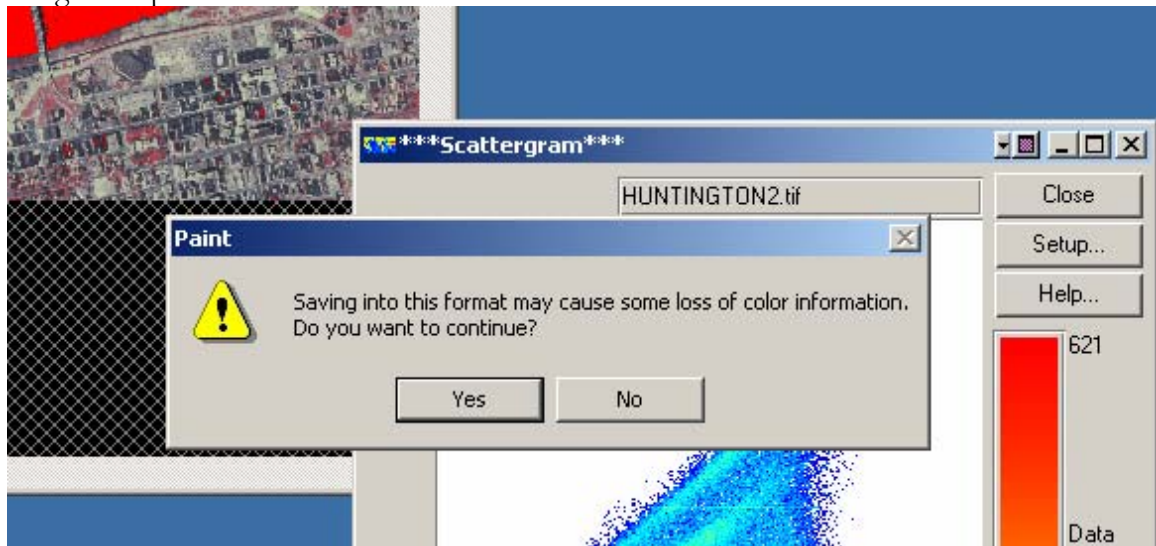


Figure 2-10 The less the colors are, the less the size and the information the image holds will be

Fig. 2.10 shows the response of the system when I tried to save the image with less color information; the system piped up a warning that some information was going to be lost, and this becomes clear when we compare the original image with the one in fig. 2.11.

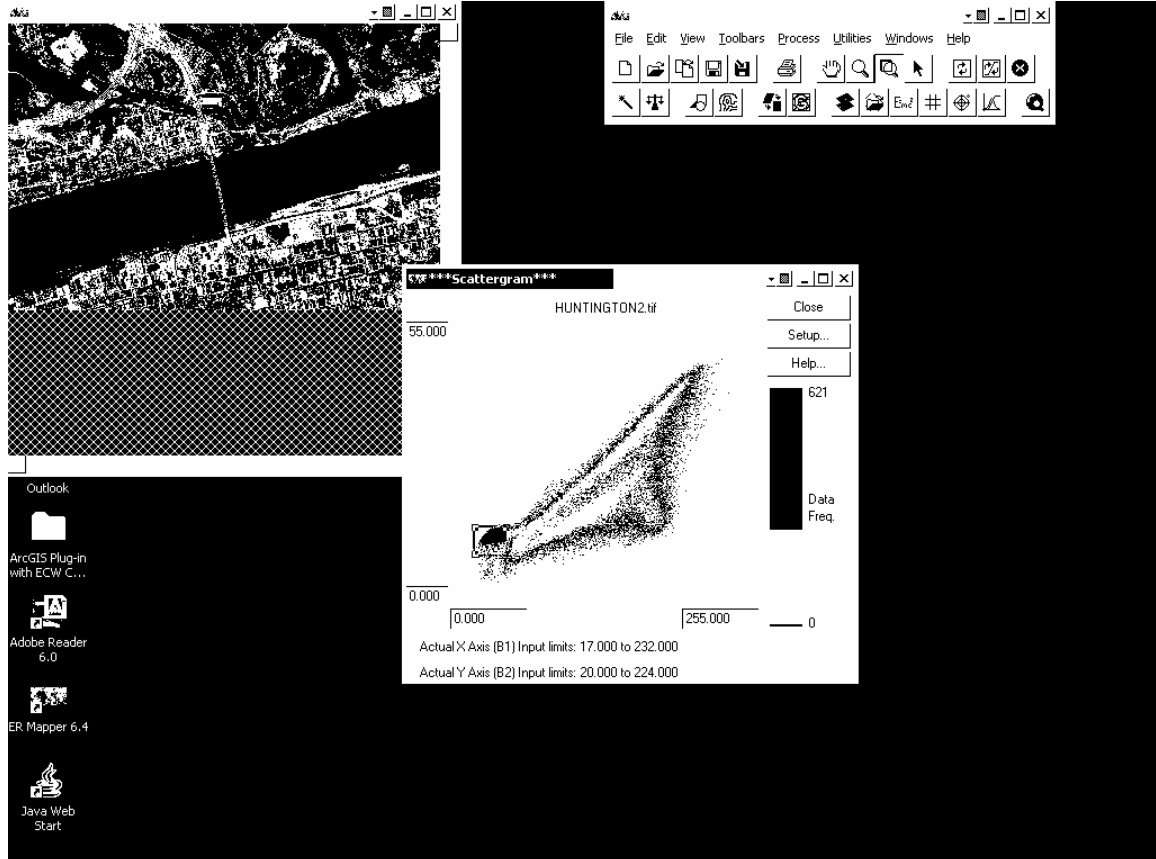


Figure 2-11 Monochrome

Fig 2.11 shows a “Monochrome image” Monochrome means that each pixel in the image will be shown by only one bit (this is the reason we use the term “Mono”); this bit represents only two colors; black and white. Any pixel here is either black 0 or white 1.

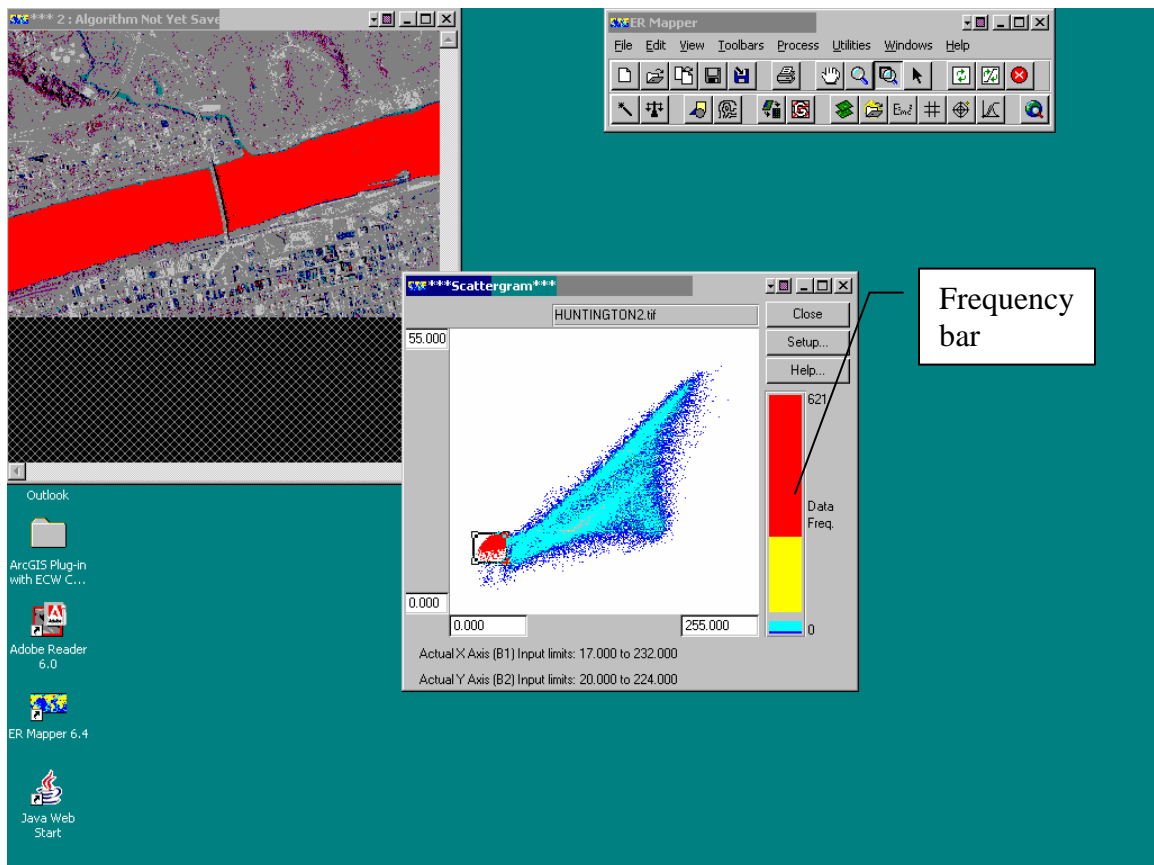


Figure 2-12 (16) Color Bitmap (4bit)Figure

Fig 2.12 above shows the same image but viewed and saved with 16 colors (Black, white, and 14 more). The difference between figures 2.11 and 2.12 in terms of the displayed colors can be clearly seen if we look at the data frequency bar on the right side of the image. In fig 2.11, we can see that the bar has only two colors which are black and

white while fig. 2.12 shows more ones those vary from red on the top to dark blue on the bottom.

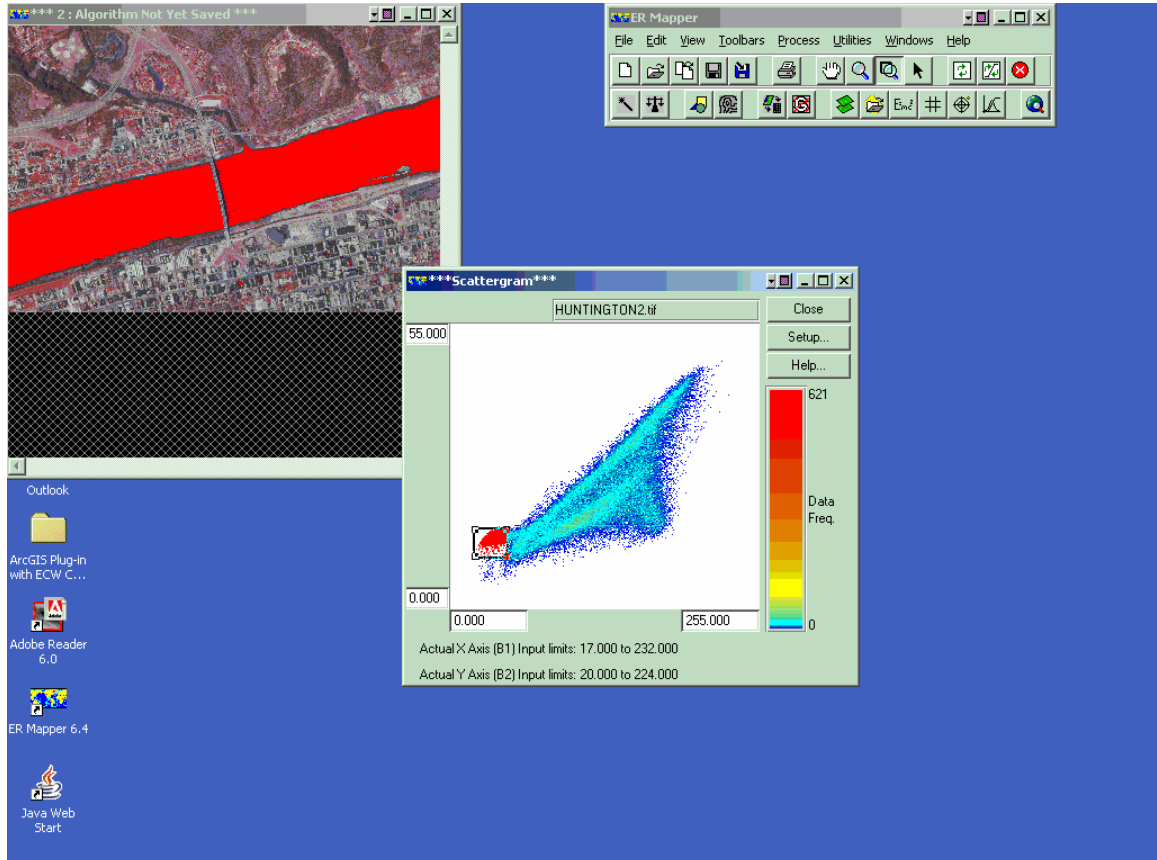


Figure 2-13 (256) Color (8bit)

Fig 2.13 shows more colors; up to 256. The frequency bar has more and more colors but they change with less smoothness than in the next figure.

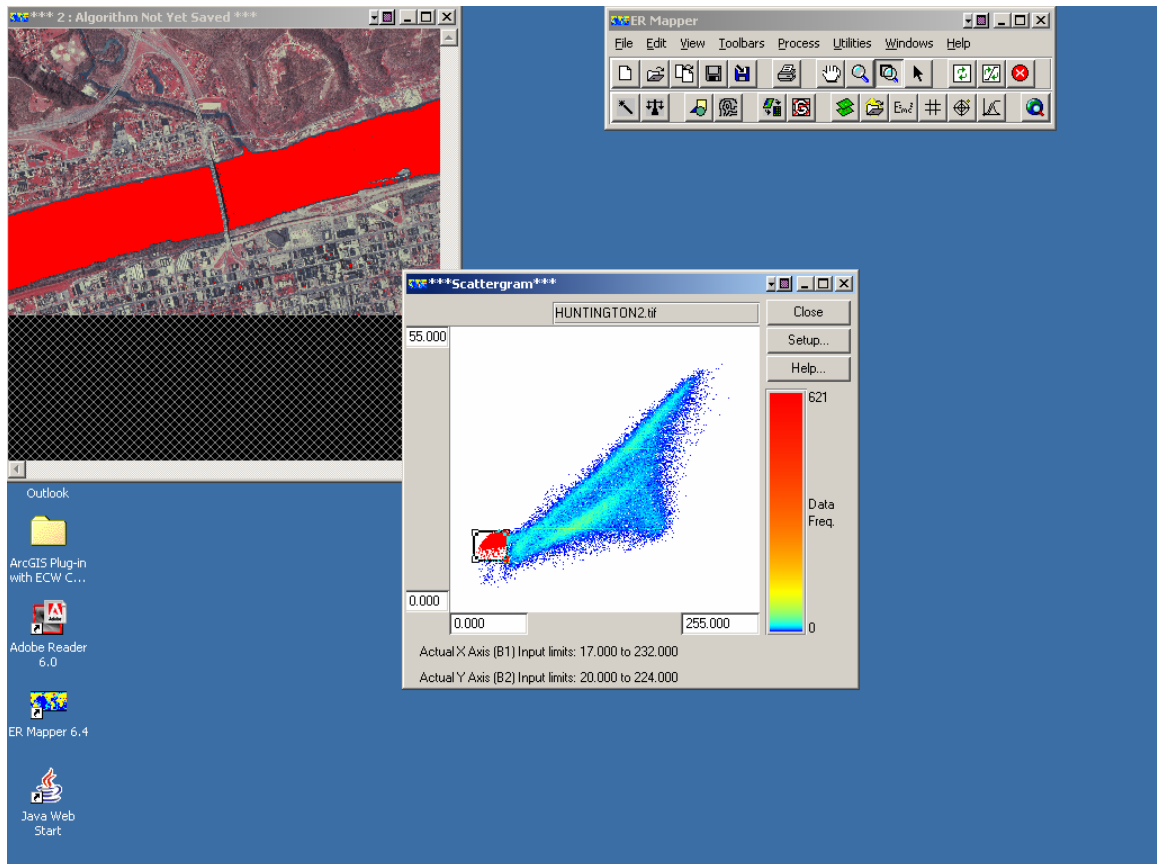


Figure 2-14 (24-bit) Bitmap (16.777216*106 Color)

Fig. 2.14 shows millions of colors. Colors in the frequency bar change smoothly; because of the very high number of displayed colors, the colors in the frequency bar seem to be of continuous change. This image is much closer to real life. 24-bit imagery was the real revolution which gave computers the ability to display real life scenes.

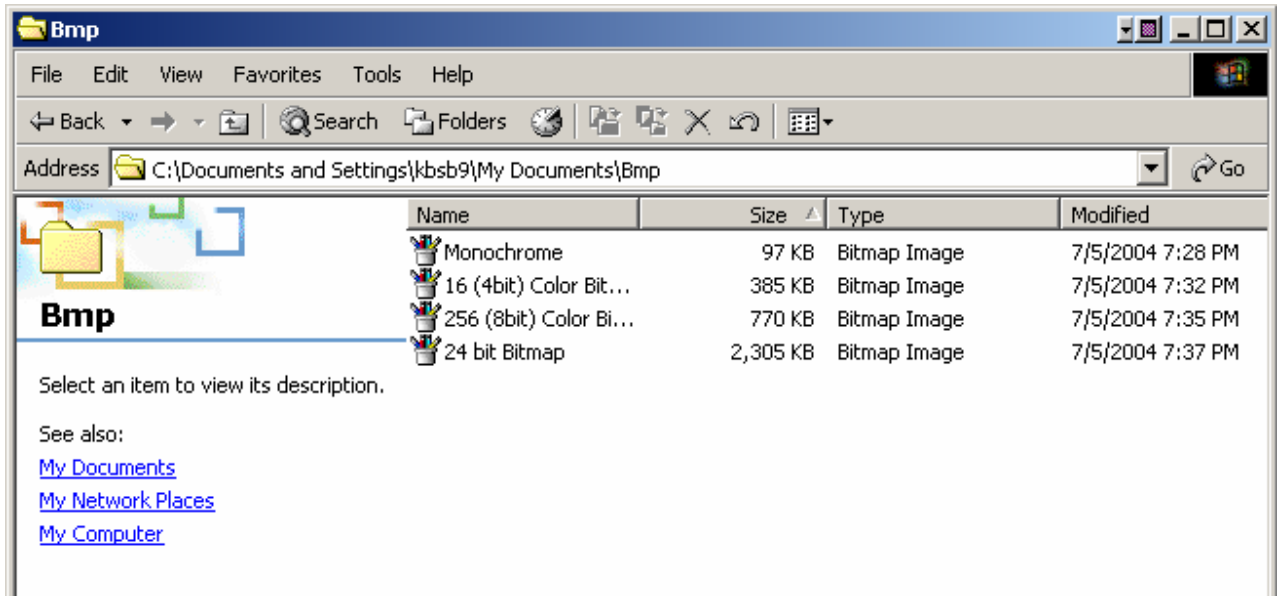


Table 2.1 The more the colors are, the larger the file is

Table 2.1 compares the different bitmap images discussed in figs 2.11- 2.14 and shows how the file size of the saved image increases with the more colors it has.

Image Formats Conclusions

The following table (Table 2.2) discusses most common Image File Formats. Source of this table is: <http://www.library.cornell.edu/preservation/tutorial/presentation/table7-1.html>

Name and Current Version	TIFF 6.0 (Tagged Image File Format)	GIF 89a (Graphics Interchange Format)	JPEG (Joint Photographic Expert Group)/JFIF (JPEG File Interchange Format)	JP2-JPX/ JPEG 2000	ImagePac, Photo CD	PNG 1.2 (Portable Network Graphics)	PDF 1.4 (Portable Document Format)
Extension(s)	.tif, .tiff	.gif	.jpeg, .jpg, .jif, .jfif	.jp2, .jpx, .j2k, .j2c	.pcd	.png	.pdf
Bit-depth(s)	1-bit bitonal; 4- or 8-bit grayscale or	1-8 bit bitonal, grayscale, or	8-bit grayscale;	supports up to 2^{14} channels, each	24-bit color	1-48-bit; 1/2/4/8-bit	4-bit grayscale; 8-bit color; up to 64-bit color

Image Compression

Khaled S. Alkharabsheh

	palette color; up to 64-bit color[1]	color	24-bit color	with 1-38 bits; gray or color		palette color or grayscale, 16-bit grayscale, 24/48-bit truecolor	support
Compression	Uncompressed Lossless: ITU-T.6, LZW, etc. Lossy: JPEG	Lossless: LZW[2]	Lossy: JPEG Lossless:[3]	Uncompressed Lossless/Lossy: Wavelet	Lossy: "Visually lossless" Kodak proprietary format[4]	Lossless: Deflate, an LZ77 derivative	Uncompressed Lossless: ITU-T.6, LZW, JBIG Lossy: JPEG
Standard/ Proprietary	De facto standard	De facto standard	JPEG: ISO 10918-1/2 JFIF: de facto standard[5]	ISO/IEC 15444 parts 1-6, 8-11	Proprietary	ISO 15948 (anticipated)[6]	De facto standard[7]
Color Mgmt.	RGB, Palette, YC _b C _r , [8] CMYK, CIE L*a*b*	Palette	YC _b C _r	Palette, YC _b C _r , RGB, sRGB, some ICC[9]	PhotoYCC	Palette, sRGB, ICC	RGB, YC _b C _r , CMYK
Web Support	Plug-in or external application	Native since Microsoft® Internet Explorer 3, Netscape Navigator® 2	Native since Microsoft® Internet Explorer 2, Netscape Navigator® 2	Plug-in	Java™ applet or external application	Native since Microsoft® Internet Explorer 4, Netscape® Navigator 4.04, (but still incomplete)	Plug-in or external application
Metadata Support	Basic set of labeled tags	Free-text comment field	Free-text comment field	Basic set of labeled tags[11]	Through external databases; no inherent metadata	Basic set of labeled tags plus user-defined tags.	Basic set of labeled tags
Comments	Supports multiple images/file[12]	May be replaced by PNG; interlacing and transparency support by most Web browsers	Progressive JPEG widely supported by Web browsers[13]	Multiple resolutions, progressive display, tiling, region of interest coding and many other advanced features	Provides 5 or 6 different resolutions of each image; unclear future	May replace GIF, though market penetration has been spotty	Preferred for printing and viewing multipage documents; strong government use
Home Page	Unofficial TIFF home page	GIF specification	JPEG home page	JPEG 2000 home page	Photo CD home page	PNG home page	PDF home page specs

Table 2.2 Common Image File Formats

GIF vs JPEG (performed using Advanced JPEG Compressor v. 4.1(1999-2002) by WinSoftmagic Development”)

1. GIF is better than JPEG for images with only a few distinct colors, such as line drawings, black and white images and small text that is only a few pixels high **because GIF was designated to show less colors with greater detail especially in patterns where lines and borders between objects take place. JPEG; on the other hand, is better in photographic and “real scene” images because JPEG compression was built on the idea that the human eye and brain are more sensitive to brightness variations than to hue variations. JPEG tries to “smoothen out” edges and lines.** This matches with the fact that some JPEG compression software programs compress the image by taking arrays of pixels and assigns the average to a new one that replaces the old ones; **in our example below, lines are represented by black pixels surrounded by white ones; as the image is more compressed, the brightness values of the black ones that represent the lines are more and more altered by the values of those surrounding values.** See figures 2.15-2.17. the results
2. GIF supports interlacing (animated scenes) while earlier version of JPEG didn't support it. Anyway, animated JPEG can now be made and displayed using “JavaScript”.
3. GIF supports transparency while JPEG doesn't.

4. JPEG images can be loaded and shown successively; this means that the viewer doesn't have to wait much to see the image if it needs time to fully load; he can see early forms with less information (note here that this is different than interlacing).

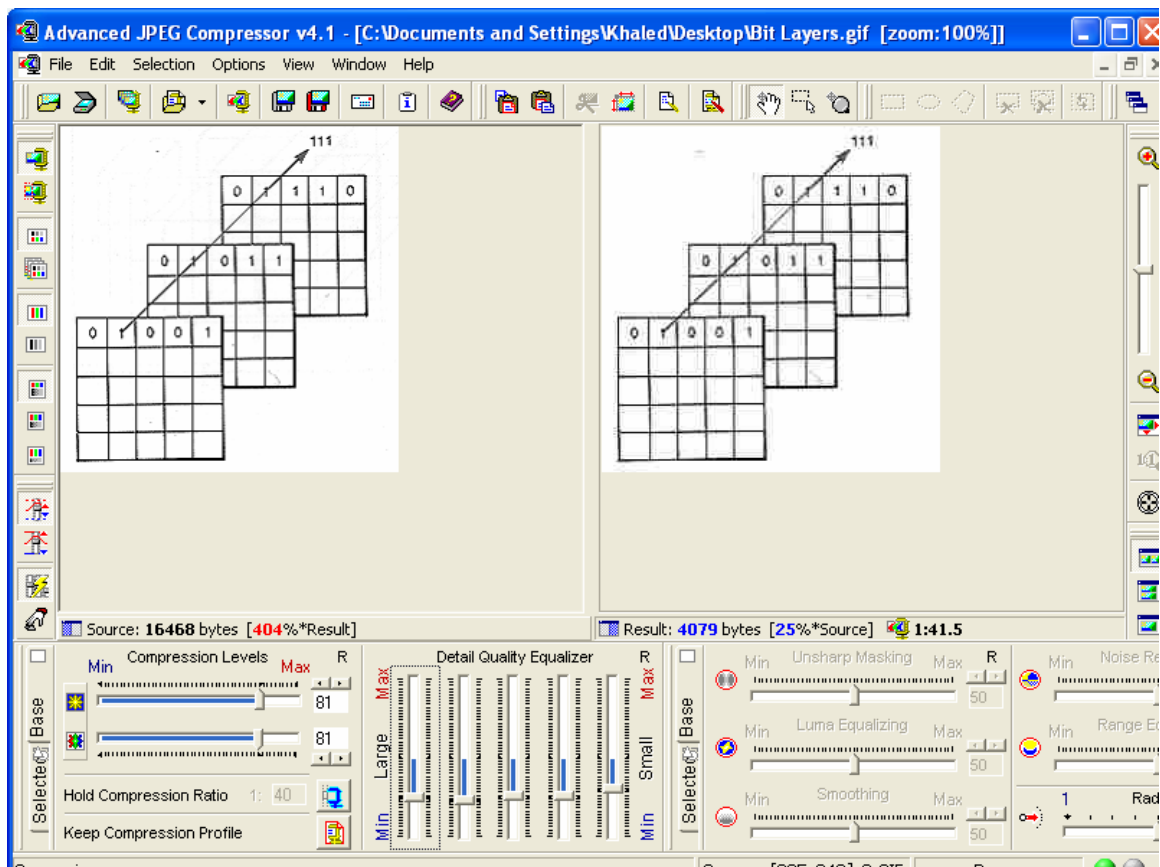


Figure 2-15 GIF vs JPEG

In Fig 2.15, a GIF image is shown as the source at the left hand side while the JPEG one is shown at the right hand side. Lines are still clear in GIF; on the other hand they started to be fuzzy in JPEG as JPEG gets less than 25% of the GIF source in terms of size.

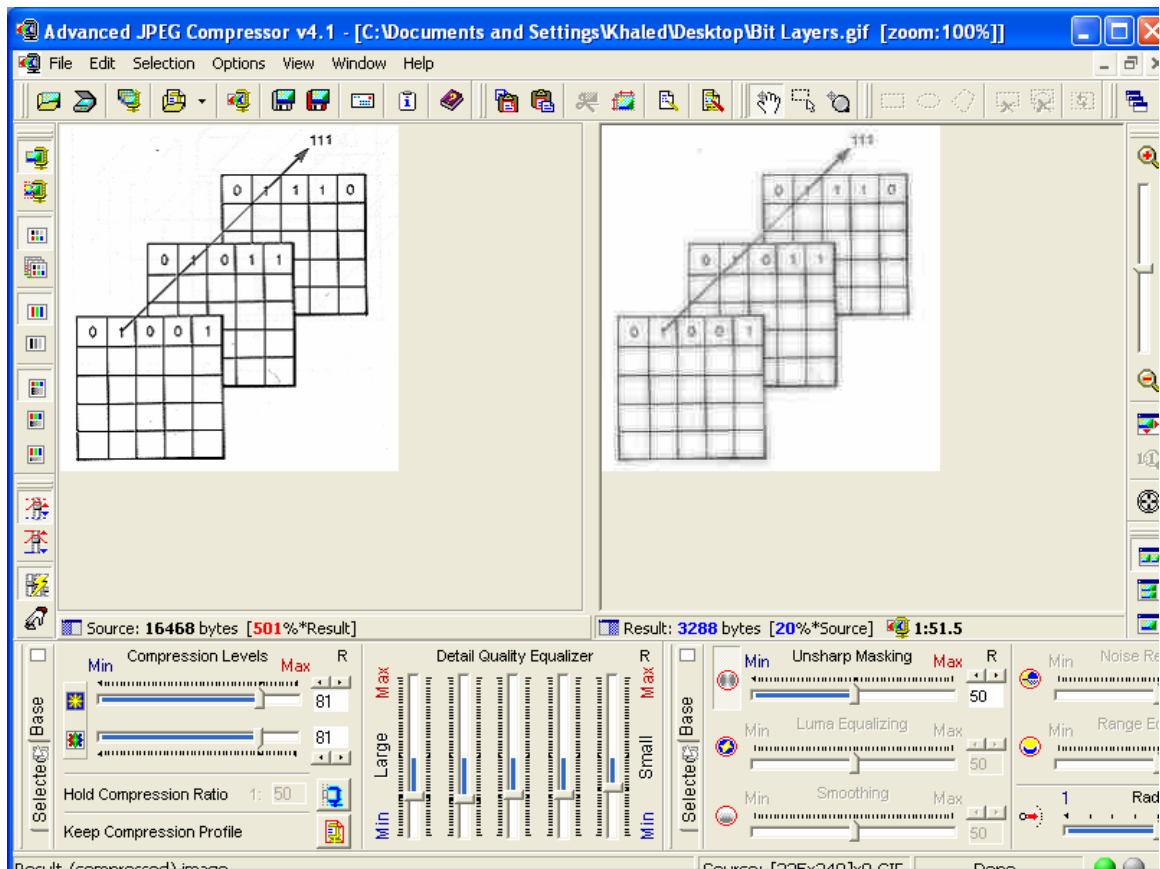


Figure 2-16 GIF works better for regular non-photographic images

JPEG is 20% of the source; lines are getting fuzzier with compression. i.e. JPEG compressing software is taking more and more information from the pixels where lines present. If we compare fig 2.15 with 2.16, we are going to notice a fuzzier image in fig 2.16 with less contrast between the white background and the black lines.

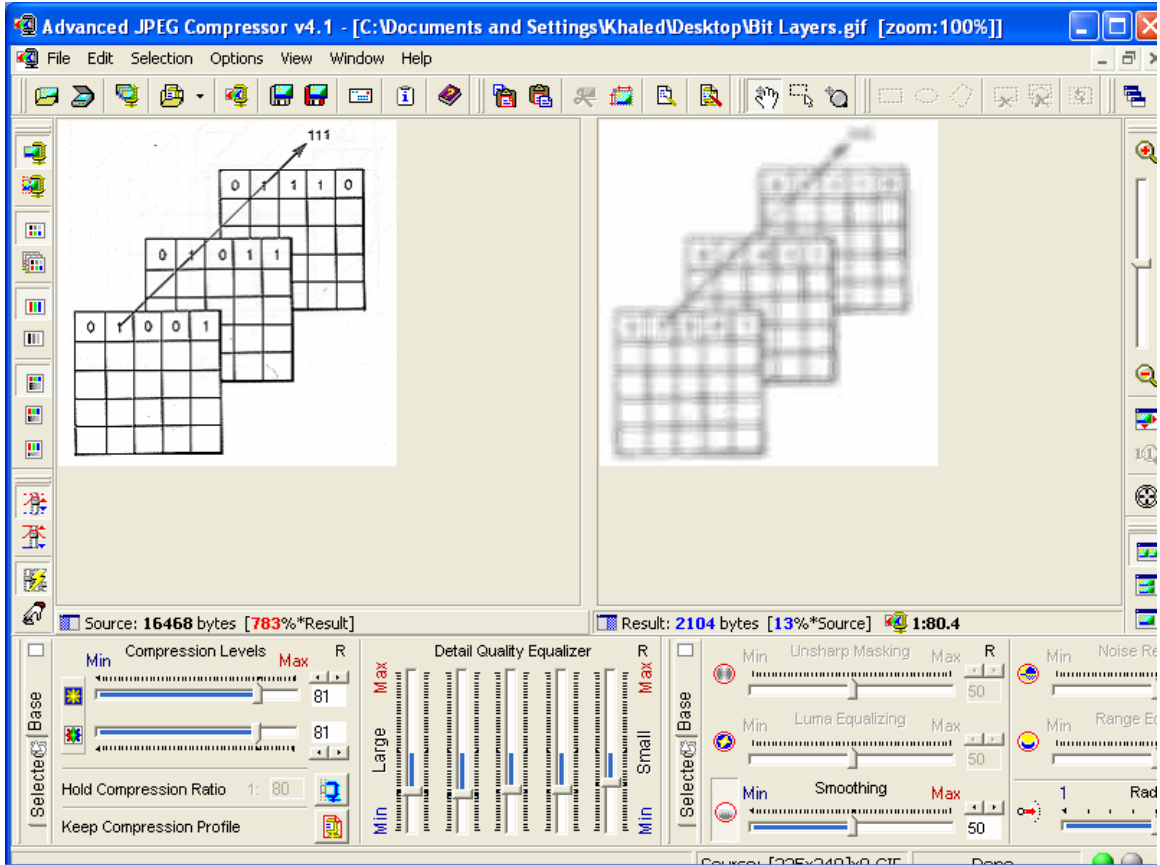


Figure 2-17 The numbers inside the array cells are no more distinguishable

Looking at the right hand side of fig. 2.17, we can see that the brightness values of the pixels representing the black lines are severely altered by compression (due to the surrounding white pixels). Pixels are becoming coarser; this supports the statement that JPEG compression replaces an array of pixels by only one pixel of a brightness value that has to do with the brightness values of all the ones in the array or their average. The image shown in fig 2.17 has much less contrast than the ones shown in figs 2.15 and 2.16.

3. CHAPTER III

Methods

In chapter 2, we discussed the difference between JPEG and GIF and the effect of JPEG compression on line shaped objects (see figs 2.15-2.17). In this chapter, different images compressed using different software programs and compressing ratios are going to be discussed. As the image is further compressed using JPEG, lines are supposed to be fuzzier and less clear according to the result in chapter 2 (figs 2.15-2.17). This effect is to be discussed in this chapter; a specific area where a road will be picked as an example on a linear object. This area will be included inside a box of a specific color in all JPEG images and the differences are to be noticed by the eye.

The un-compressed “TIF” image

Before we start with the un-compressed image of the type “TIF” (called TIFF sometimes), it’s useful to note that this image (fig 3.1) will be called “un-compressed” just to explain its relation to its compressed versions which will be used later. This image was obtained through the internet from the DOQQ database that belongs to West Virginia University.

**Figure 3-1**

Dimensions: 2832 x 1880
Type: TIF Image
Size: 15.2 MB

Sample calculation:

Each pixel is represented by 3bytes (one for each of the three bands), and the image contains 2832×1880 pixels; this means that the size of the image in MB is $(2832 \times 1880 \times 3) / (1024 \times 1024) = 15.233\text{MB}$

Looking inside the blue box in the image shown in fig 3.1, we can notice the road edges. The line that separates the two sides of the highway can also be clearly seen. According to the results gotten from the comparison performed between JPEG and GIF in chapter 2, this line is supposed to become fuzzier and less noticeable with JPEG compression.

JPEG (software used is “PaperPort LE7.0 by ScanSoft Inc”)

Fig. 3.2 bellow shows a JPEG form of the original TIFF after compressing to 95% using the **PaperPort** software.



Figure 3-2

Dimensions: 2832 x 1880
Type: JPEG Image
Size: 1.13 MB

Compression ratio (by “LuraWave”)= 13.4.

Note here that “LuraWave” software that will be used later is capable of giving the compression ratio of any previously compressed image and the size of the original one from which the compressed file was gotten. This will be used to get the compression ratio for all the JPEG images in this section.

No major differences can be noticed in the area included inside the blue box in fig 3.2

Comparisons

Since studying the effects of compression on the image, the information it has, and its file size are the key objects of this thesis. The “pixilation” of the image and the less identifiable an object becomes with more compression are to be discussed in the next two figures. Let’s concentrate on the bridge and see how less clear its edges become with compression.



Figure 3-3 Huntington. Tiff. A closer look at the bridge and the pixilation effect



Figure 3-4 JPEG 10%

We can see from the figures above that the un-compressed image represents more information and a clearer color for the river. The pixel sizes are smaller; this means that the image can be zoomed more with better display quality. In fig. 3.3, bridge details and edges are clearer and more noticeable.

Let's try to explain how JPEG approaches images from our results in figs. 3.3 and 3.4. In fig. 3.4, there are fewer details and larger pixels with a smaller range of BV's (brightness values), and this supports what was discussed in chapter 2 that JPEG was built on the fact that the human eye and brain detect changes in brightness less than

those in hue. If we look at fig. 3.4, we can see that the BV of many pixels are more similar.

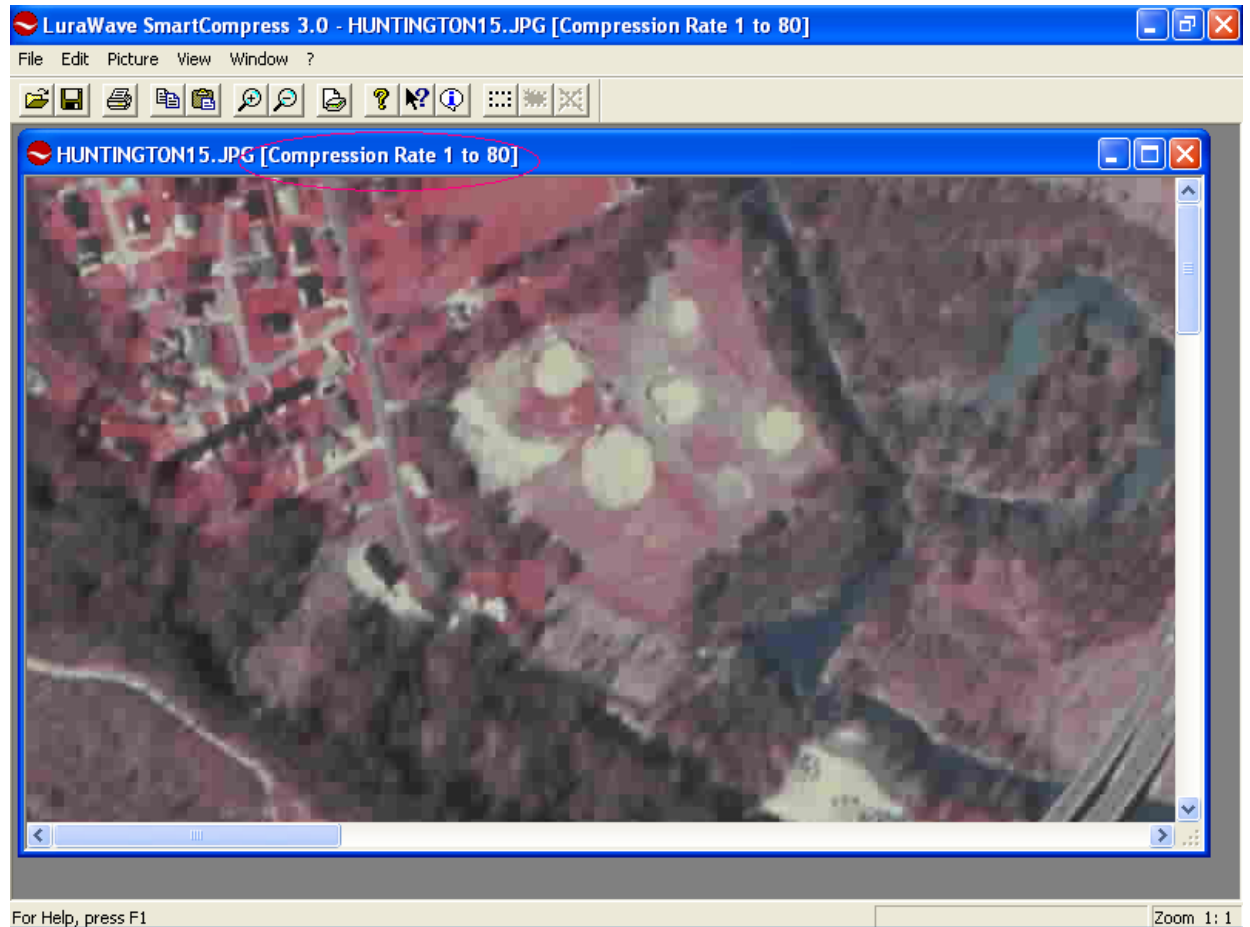


Figure 3-5 The ability to open a JPEG image using LuraWave and determine the compression ratio
Note here that calculations similar to those performed for the last image can't be applied in this case since the percentage used in this software (and many others) doesn't refer to any relations between the sizes of the images or to the quality ratio. However, compression ratio can be determined when LuraWave is used to open the image. This

software can determine the ratio and show it at the top of the window (see sample calculation for Fig. 3.1).

Some software programs (such as “LuraWave SmartCompress”, “Adobe Photoshop”, or ECW compressing programs) use such a ratio to show size or quality ratios. In this case, the relation is clearly stated during the compression process in order to help the user determine what ratio to use.

Note here also that some software programs (such as “LuraWave”) give the ability to control noise, resolution, size, and color with which the output compressed image is going to be saved. LuraWave also has the option of controlling the way the image can be displayed by controlling many factors such as quality, size, resolution, and color.



Figure 3-6 JPEG 90%

Dimensions: 2832 x 1880
Type: JPEG Image
Size: 836 KB

Compression ratio (by “LuraWave”)= 18.6.

In fig 3.6, the area inside the blue box is still clear and the road still has clear edges.



Figure 3-7 JPEG 85%

Dimensions: 2832 x 1880
Type: JPEG Image
Size: 539 KB

Compression ratio (by "LuraWave")= 29.



Figure 3-8 JPEG 80%

Dimensions: 2832 x 1880
Type: JPEG Image
Size: 479 KB

Compression ratio (by “LuraWave”)= 33

Figs. 3.7 and 3.8 do not show any difference at this scale, and this means that we are still getting high quality images with smaller and smaller file sizes. The road edges inside the blue box are still clear.



Figure 3-9 JPEG 75%

Dimensions: 2832 x 1880
Type: JPEG Image
Size: 421 KB

Compression ratio (by "LuraWave")= 37.



Figure 3-10 JPEG 70%

Dimensions: 2832 x 1880
Type: JPEG Image
Size: 363 KB

Compression ratio (by "LuraWave")= 43.

Fig. 3.10; 70% JPEG, no noticeable data loss at this scale.



Figure 3-11 JPEG 65%

Compression ratio (by “LuraWave”)= 46.

Fig. 3.11 shows a 65% compressed JPEG, very little data loss noticeable only when zooming in to large scales. See Fig 3.12 and note the changes around the edges.

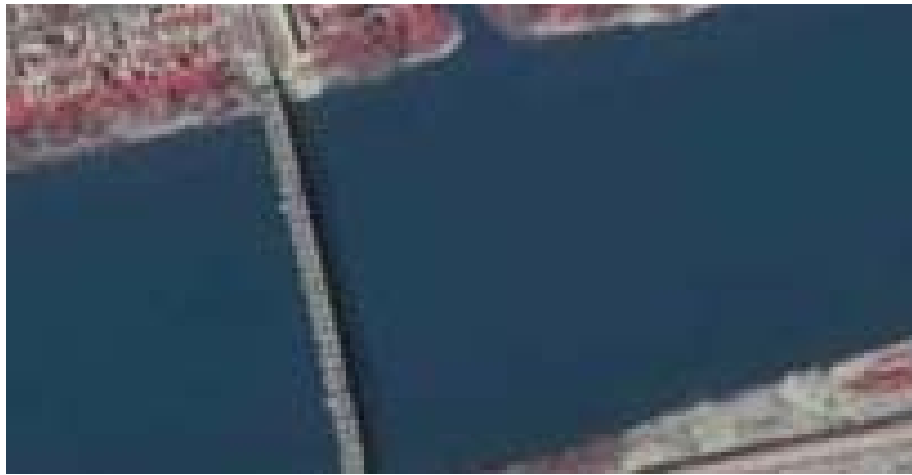


Figure 3-12 JPEG 65% (zoomed in)

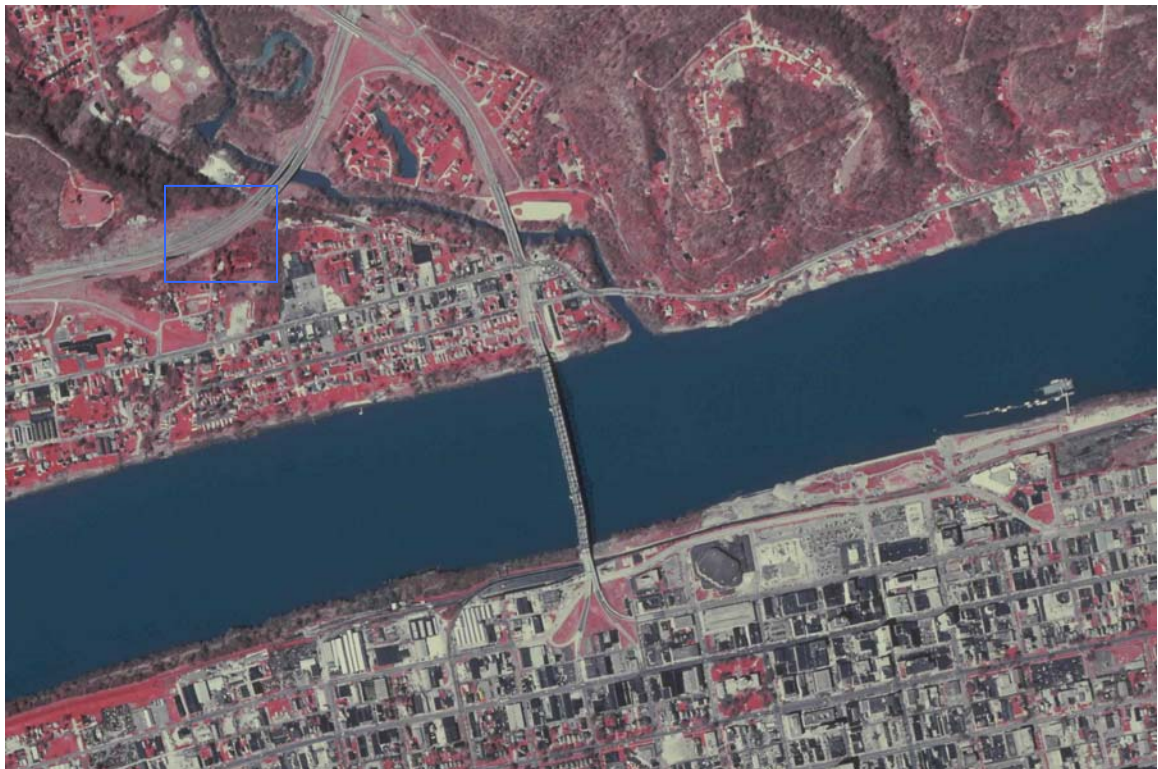


Figure 3-13 JPEG 60%

Compression ratio (by "LuraWave")= 52



Figure 3-14 JPEG 55%

Compression ratio (by “LuraWave”)= 55.

Fig 3.12 JPEG 55%

Fig. 3.14 shows that 55% was the “edge” between having a good compressed image with few noticeable changes (at this scale) and other more compressed ones with good quality **but noticeable differences and pixilation.**



Figure 3-15 JPEG 50%

Compression ratio (by “LuraWave”)= 58.

Fig. 3.15 shows noticeable pixilation. The image started to show some lighter shades of blue, especially in the river area.



Figure 3-16 JPEG 45%

Compression ratio (by “LuraWave”)= 63

Fig. 3.16 shows that the lighter shades of blue are “spreading out” in the river area.



Figure 3-17 JPEG 40%

Compression ratio (by “LuraWave”)= 65.

Fig. 3.17, more and more “lighter” blue shades.



Figure 3-18 JPEG 35%

Compression ratio (by “LuraWave”)= 70.

In Fig. 3.18 the river is brighter and the color is not similar to that blue in the fig. 3.1. At this compression ratio, the line that separates the two sides of the highway is not clear as before; this ratio can be considered the critical limit between the clearness and the fuzziness of the area.



Figure 3-19 JPEG 30%

Compression ratio (by “LuraWave”)= 72.

In Fig. 3.19, the bridge started to be pixelated at higher scales. The area inside the blue box is fuzzier than in fig 3.18.



Figure 3-20 JPEG 25%

In Fig. 3.20 that shows an image with 75 times compression ratio (by “LuraWave”), the overall color of most of the objects started to look clearly “pale”. In the blue box, the two sides of the road are not distinguishable anymore at this scale.



Figure 3-21 JPEG 20%

Compression ratio (by “LuraWave”)= 78.

Figs. 3.21- 3.23 show how the information in the image is severely altered by compression; BV’s are altered and are different than in the un-compressed image. In figs. 3.22 and 3.23, the road area inside the blue box became extremely fuzzy.



Figure 3-22 JPEG 15%

Compression ratio for fig. 3.20 (by “LuraWave”)= 80.



Figure 3-23 JPEG 10%

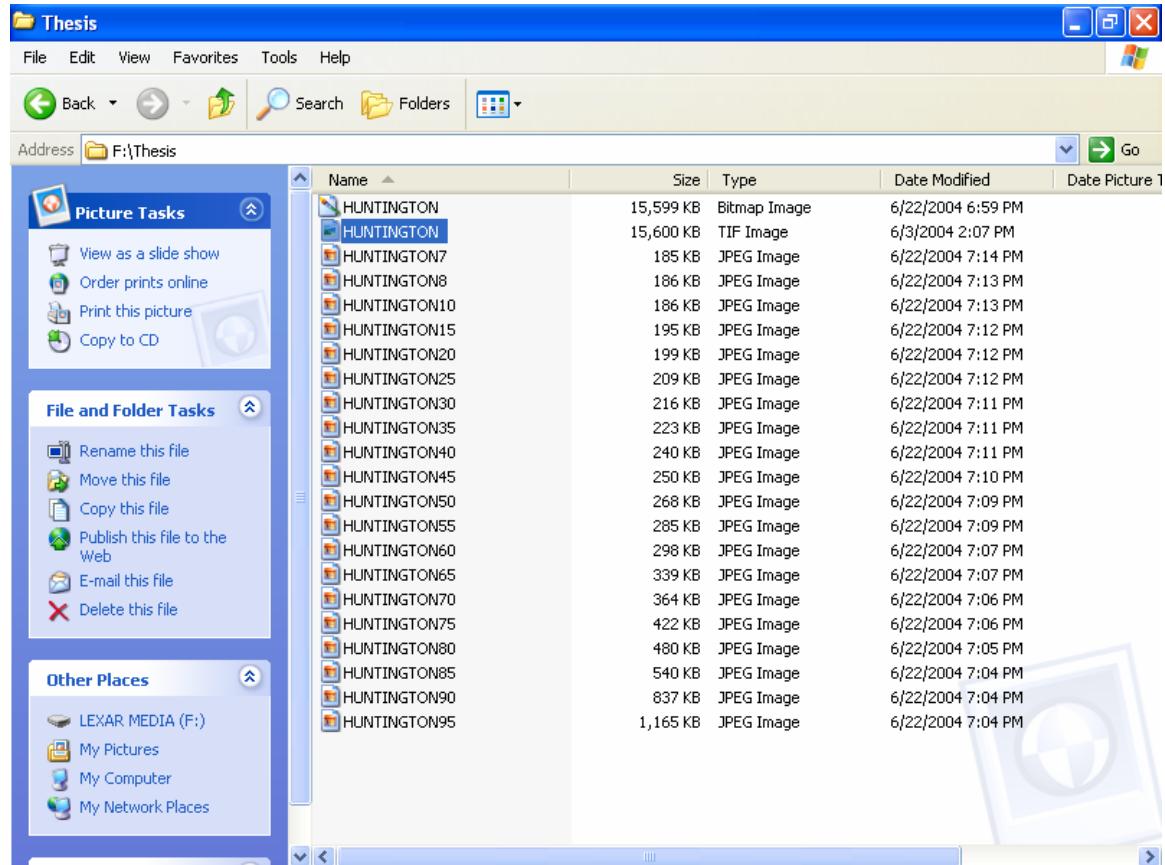
Compression ratio (by “LuraWave”)= 84.

Fig. 3.23 shows the result of the compression of TIFF to 10% using “PaperPort”. The file size was highly reduced but the quality became clearly low.



Figure 3-24 Shows that his software doesn't let the user enter values less than 10%.

Fig. 3.24, this specific software doesn't accept compression percentages less than 10%.



Name	Size	Type	Date Modified	Date Picture 1
HUNTINGTON	15,599 KB	Bitmap Image	6/22/2004 6:59 PM	
HUNTINGTON	15,600 KB	TIF Image	6/3/2004 2:07 PM	
HUNTINGTON7	185 KB	JPEG Image	6/22/2004 7:14 PM	
HUNTINGTON8	186 KB	JPEG Image	6/22/2004 7:13 PM	
HUNTINGTON10	186 KB	JPEG Image	6/22/2004 7:13 PM	
HUNTINGTON15	195 KB	JPEG Image	6/22/2004 7:12 PM	
HUNTINGTON20	199 KB	JPEG Image	6/22/2004 7:12 PM	
HUNTINGTON25	209 KB	JPEG Image	6/22/2004 7:12 PM	
HUNTINGTON30	216 KB	JPEG Image	6/22/2004 7:11 PM	
HUNTINGTON35	223 KB	JPEG Image	6/22/2004 7:11 PM	
HUNTINGTON40	240 KB	JPEG Image	6/22/2004 7:11 PM	
HUNTINGTON45	250 KB	JPEG Image	6/22/2004 7:10 PM	
HUNTINGTON50	268 KB	JPEG Image	6/22/2004 7:09 PM	
HUNTINGTON55	285 KB	JPEG Image	6/22/2004 7:09 PM	
HUNTINGTON60	298 KB	JPEG Image	6/22/2004 7:07 PM	
HUNTINGTON65	339 KB	JPEG Image	6/22/2004 7:07 PM	
HUNTINGTON70	364 KB	JPEG Image	6/22/2004 7:06 PM	
HUNTINGTON75	422 KB	JPEG Image	6/22/2004 7:06 PM	
HUNTINGTON80	480 KB	JPEG Image	6/22/2004 7:05 PM	
HUNTINGTON85	540 KB	JPEG Image	6/22/2004 7:04 PM	
HUNTINGTON90	837 KB	JPEG Image	6/22/2004 7:04 PM	
HUNTINGTON95	1,165 KB	JPEG Image	6/22/2004 7:04 PM	

Table 3.1 File size summary

Table 3.1 summarizes the results of compression and their effect on the file size; it starts by indicating the size of the un-compressed image, and then goes through the compressed ones starting from the smaller to the larger ones in size. It shows that the file size decreases with more compression (less percentage). Note here that there are two values less than 10% those are 8 and 7%; after clicking “OK” on the warning window (fig. 3.24), the software compresses the image as if a value of 10% was entered.

Results

The good characteristics of JPEG compression software programs that make them widely used worldwide are the ability to preset the compression or quality ratio before the compression process starts and the fact that JPEG has been extensively used in internet applications since found.

ECW using ER-Mapper software (By Earth Resource Mapping) in MU Digital Image Processing Lab

ECW (ER Mapper Compressed Wavelet) was developed by ER-Mapper and was dedicated to the fields where large imagery files are used. It was designed to compress the image very efficiently and with less data and information loss and to give the ability to work on the data with as less effects on the results. ECW is different than JPEG by the way it deals with and compresses the image.

The developments in ECW and the efficiency of JPEG have prompted new versions of ECW that are related to JPEG2000.

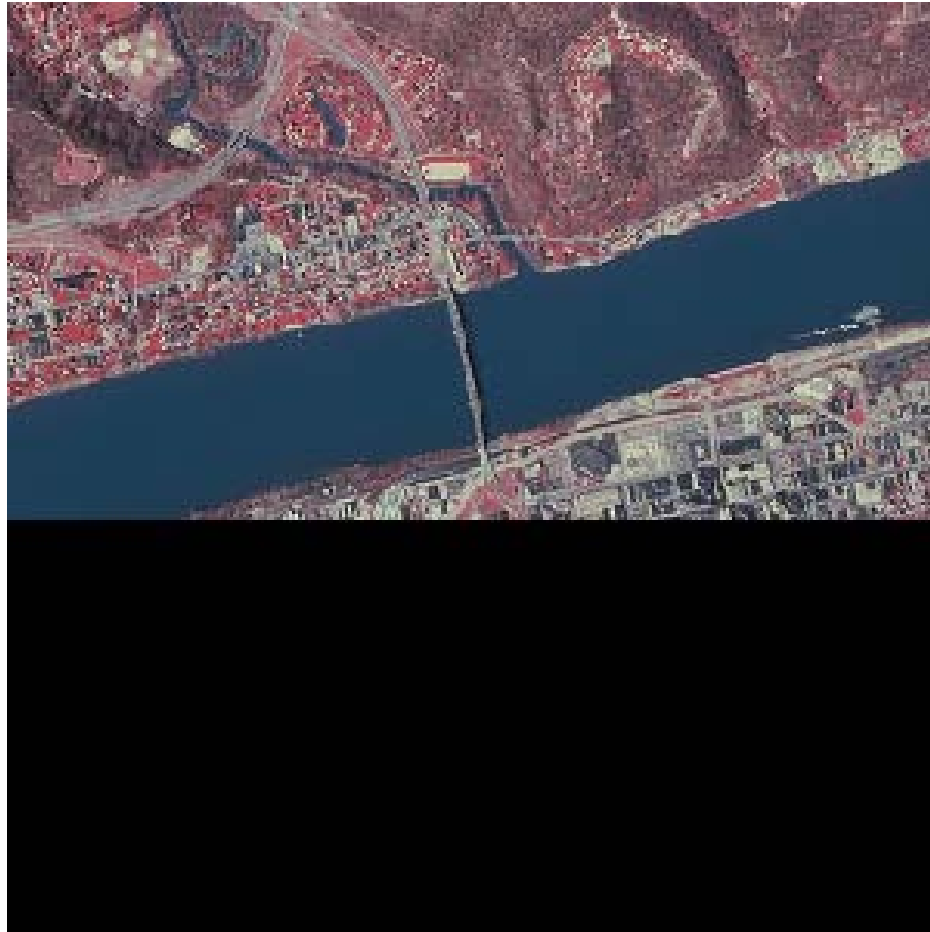


Figure 3-25 ECW; theoretical compression ratio= 30 times

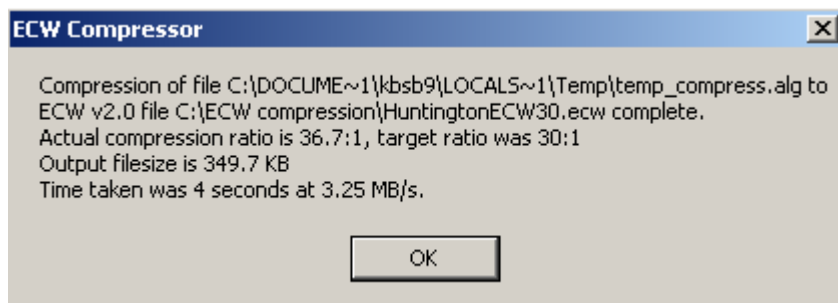


Figure 3-26 process summary for the 30-times compressed ECW output

$$15.2 * 1024 / 30 = 518.83. \text{ (Estimated)}^{(1)}$$

$$15.2 * 1024 / 36.7 = 424.11. \text{ (Actual)}^{(2)}$$

Fig. 3.25 shows an image that is 1/30 times the size of the original un-compressed image. This figure gives an indication about the high quality outputs ECW gives. The calculations above show the file size relation between the un-compressed image and the outputs of the ER-Mapper ECW (in MU Digital Image processing lab). The original image is 15.2Mbyte which is equal to 15.2×1024 Kbytes. If it was divided by 30, we will get the estimated file size of the output ⁽¹⁾. But, the system did give a summary that shows a 36.7 compression ratio; so, to get the actual output file size, the size of the original image has to be divided by 36.7 ⁽²⁾.

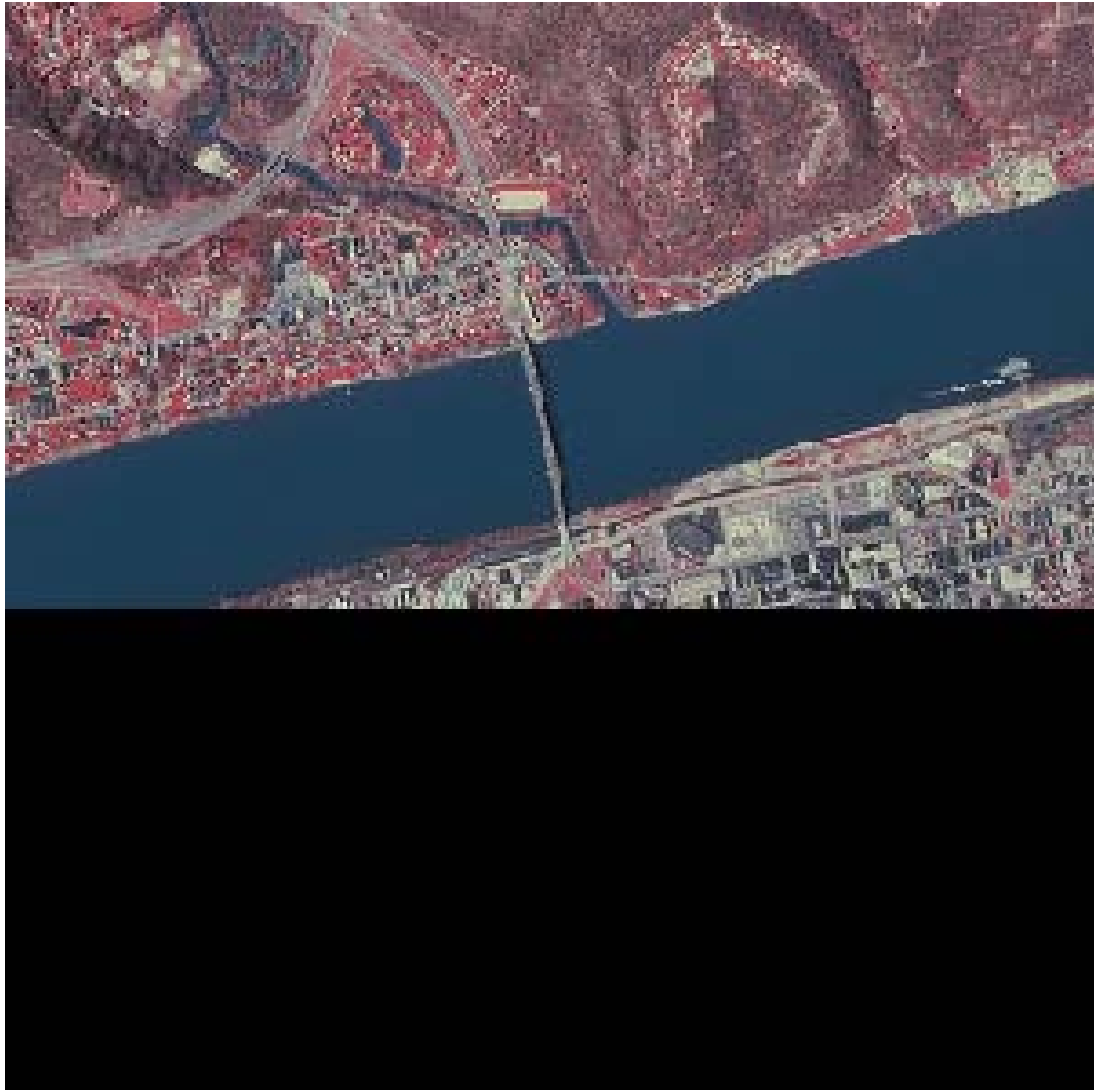


Figure 3-27 ECW; theoretical compression ratio= 99 times

Type: Ecw File
Date Modified: 6/22/2004 4:17 PM
Size: 201 KB

Fig. 3.27 above shows a 99 times compressed ECW image; it also shows a high quality outputs. Note here that in most cases this software gives compression ratios up to 85 times or a little more; any larger values entered by the user will be considered as that maximum that the software gives; which is around 85 as stated.

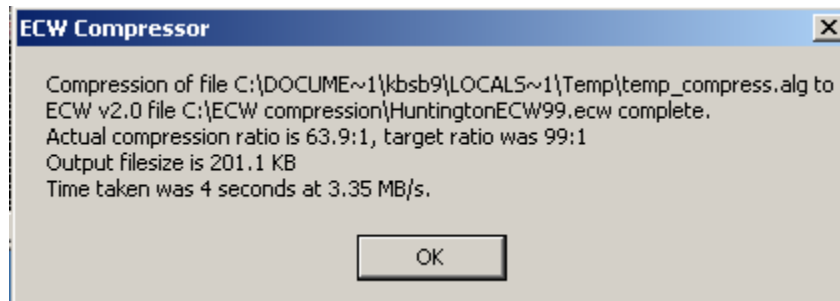


Figure 3-28 process summary for the 99-times compressed ECW

$$15.2 * 1024 / 99 = 157.22.$$

$$15.2 * 1024 / 63.9 = 243.58; \text{ close to file size but not accurate.}$$

This compression technique worked very well; many images were compressed using many different compression ratios while not much quality difference was noticed.

ECW shares the characteristic of presetting and controlling the compression and quality ratios of the output before the process starts with JPEG; it even gives some statistical summary windows after the process finishes (see fig. 3.28). Anyway, even though ECW was designed to work mainly with the internet, it is still not widely known or used in this field.

Name	Size	Type	Date Modified
HuntingtonECW30	350 KB	Ecw File	6/22/2004 8:16 PM
HuntingtonECW35	323 KB	Ecw File	6/22/2004 8:15 PM
HuntingtonECW40	296 KB	Ecw File	6/22/2004 8:14 PM
HuntingtonECW45	281 KB	Ecw File	6/22/2004 8:13 PM
HuntingtonECW50	266 KB	Ecw File	6/22/2004 8:12 PM
HuntingtonECW55	256 KB	Ecw File	6/22/2004 8:11 PM
HuntingtonECW60	244 KB	Ecw File	6/22/2004 8:10 PM
HuntingtonECW65	237 KB	Ecw File	6/22/2004 8:09 PM
HuntingtonECW70	229 KB	Ecw File	6/22/2004 8:08 PM
HuntingtonECW75	224 KB	Ecw File	6/22/2004 8:06 PM
HuntingtonECW80	216 KB	Ecw File	6/22/2004 8:05 PM
HuntingtonECW85	212 KB	Ecw File	6/22/2004 8:01 PM
HuntingtonECW90	208 KB	Ecw File	6/22/2004 7:59 PM
Huntingtonecw95	262 KB	Ecw File	6/22/2004 7:54 PM
HuntingtonECW95second trial	205 KB	Ecw File	6/22/2004 8:19 PM
HuntingtonECW96	203 KB	Ecw File	6/22/2004 8:22 PM
HuntingtonECW97	203 KB	Ecw File	6/22/2004 8:21 PM
HuntingtonECW98	202 KB	Ecw File	6/22/2004 8:20 PM
HuntingtonECW99	202 KB	Ecw File	6/22/2004 8:17 PM
HuntingtonECW99	1 KB	ER Mapper Raster ...	6/22/2004 8:17 PM
HuntingtonECW301	1 KB	ER Mapper Raster ...	6/22/2004 8:16 PM
HuntingtonECW359	1 KB	ER Mapper Raster ...	6/22/2004 8:15 PM
HuntingtonECW409	1 KB	ER Mapper Raster ...	6/22/2004 8:14 PM
HuntingtonECW459	1 KB	ER Mapper Raster ...	6/22/2004 8:13 PM
HuntingtonECW508	1 KB	ER Mapper Raster ...	6/22/2004 8:12 PM
HuntingtonECW558	1 KB	ER Mapper Raster ...	6/22/2004 8:11 PM
HuntingtonECW609	1 KB	ER Mapper Raster ...	6/22/2004 8:10 PM

Table 3.2 ECW compressed images and file size summary

Refer to Table 3.2 for the results of compressing the original image with many different compressions ratios; it shows the file size of the outputs and the descending file size as the image is more compressed. Note here that for each image, the size of the output compressed image is close to the size of the original compressed image divided by the actual compression ratio.

Adobe Acrobat Reader v6.0 (1987-2003) by Adobe Systems Incorporated

Adobe Acrobat in general is an effective way of saving and retrieving documentation. It has the ability to save and view high quality compressed images of small file sizes.

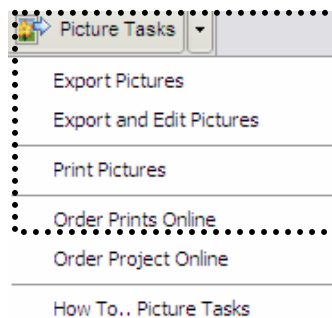
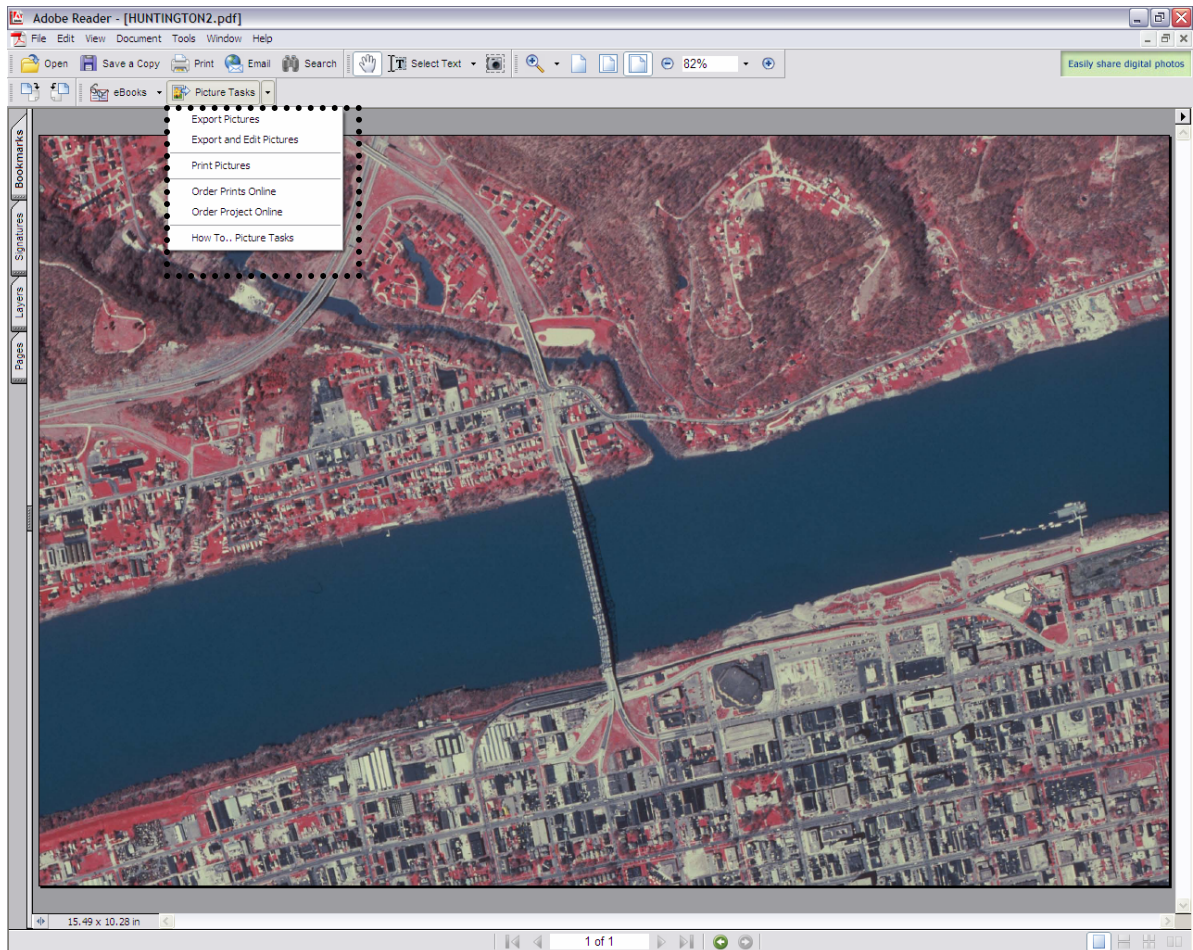


Figure 3-29 PDF images

Fig 3.29 above shows how Adobe Acrobat deals with compressed imagery. .pdf images are high quality images which can be easily edited, de-compressed and exported to other folders as JPEG, or printed out. The good thing about .pdf images is the ability to relate them to text and put them together in small files (smaller than word documents).

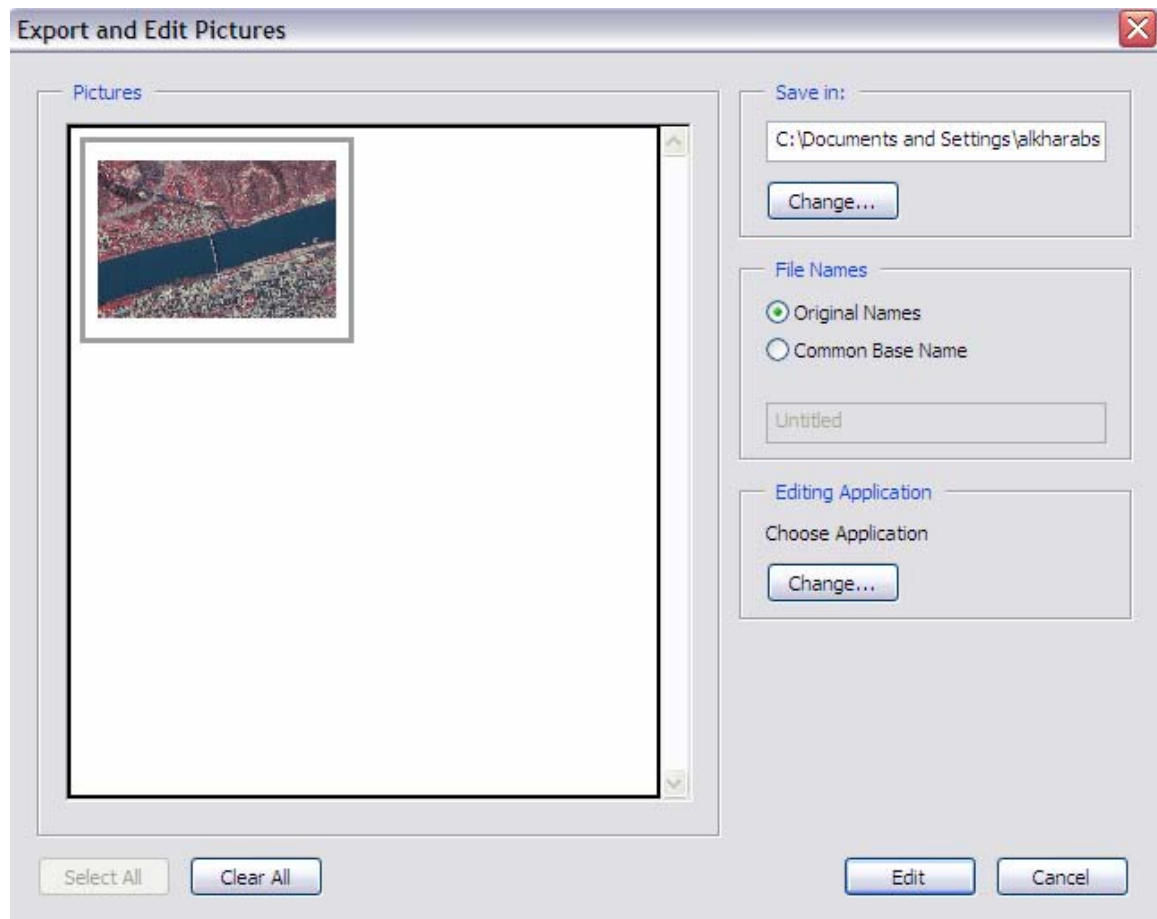


Figure 3-30 Adobe Acrobat Reader dealing with imagery

Fig. 3.30 above shows the “Export and Edit” process.

Adobe Photoshop Album 2.0 Starter Edition (2003) by Adobe Systems Incorporated

Adobe Photoshop is used widely in processing images for internet use, managing picture albums and folders, and in emailing high quality small file size imagery with the ability to control quality, file size, and the physical size of the image (in pixels or inches). When emailing a picture using Adobe Photoshop, the user can also chose to email the images as a PDF slideshow, and individual attachment, or to convert it to JPEG prior to emailing it.

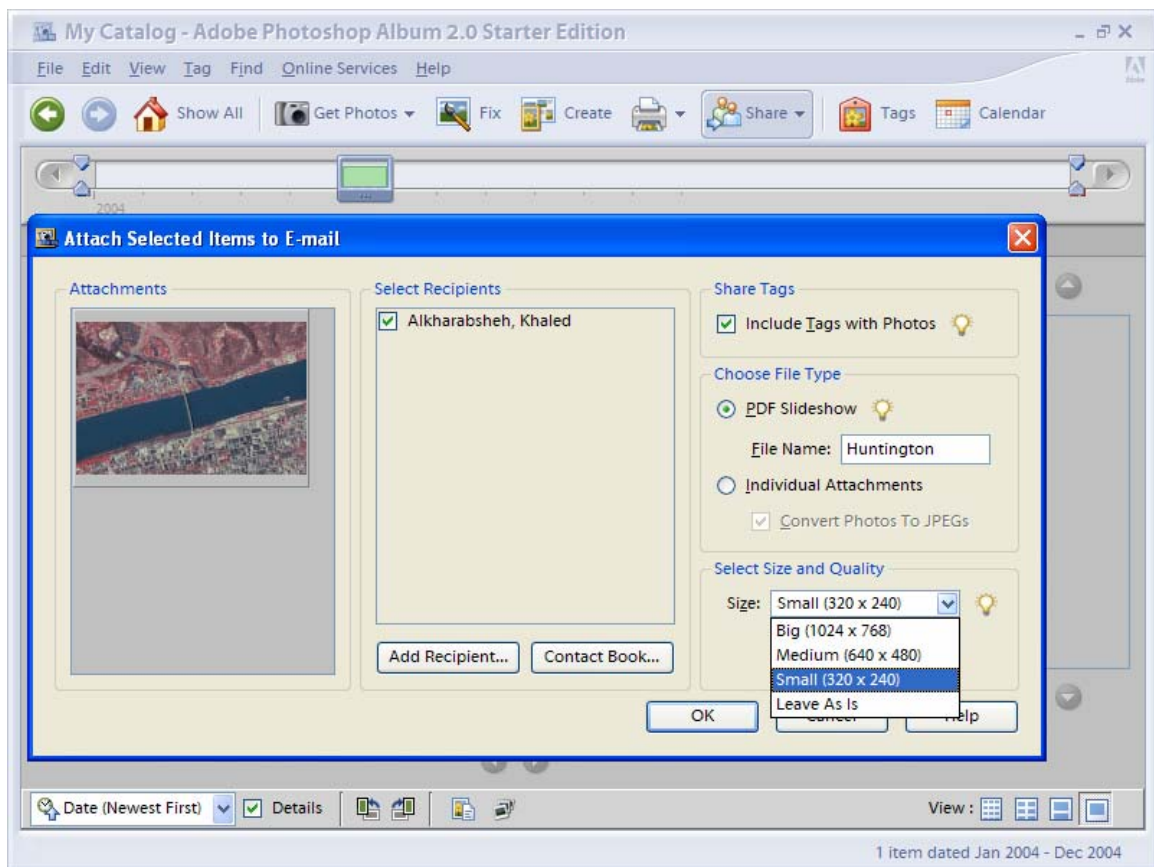


Figure 3-31 Adobe Photoshop

Fig. 3.31 shows how the user can pre-determine the type of the attached image and its physical size.

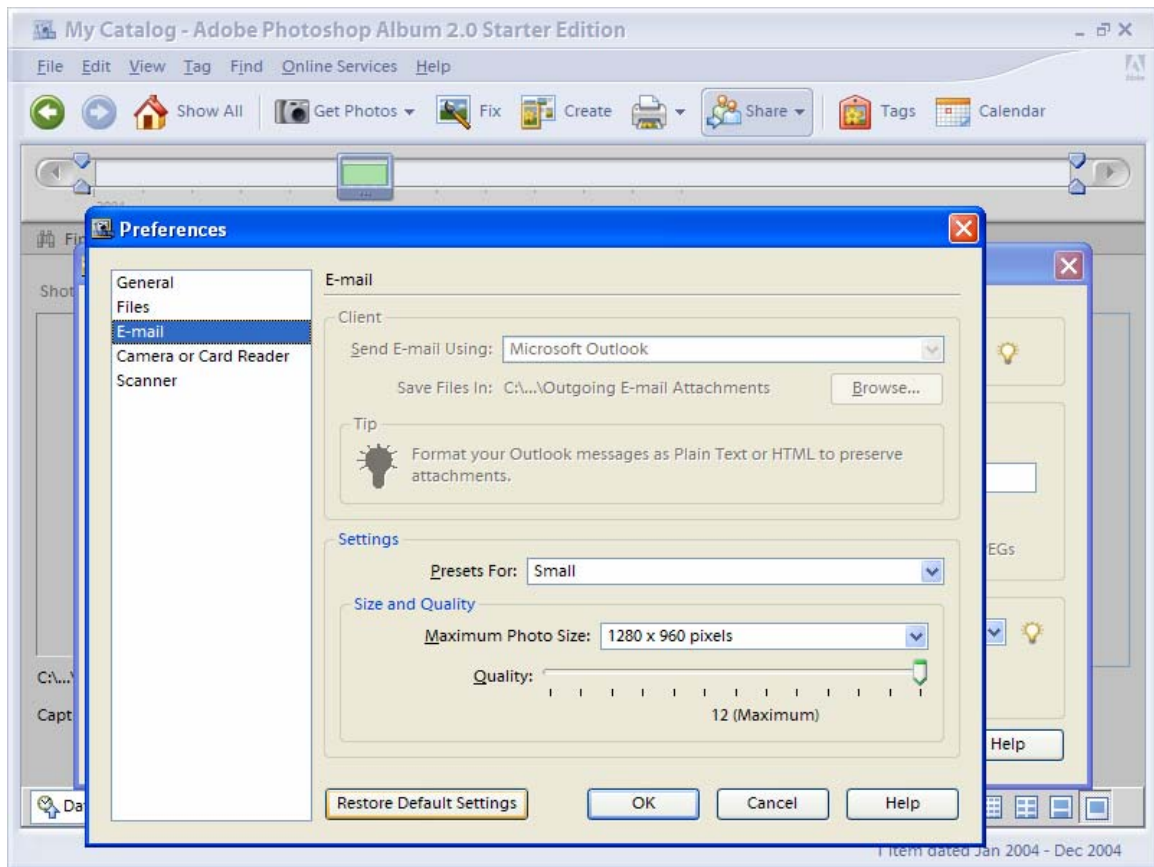


Figure 3-32 Photoshop user can reduce his pictures using many options

Fig. 3.32 shows how the user can pre-determine the quality and again the physical size of the image.

ECW Compressor v2.6 by Earth Resource Mapping

This software is different than that used in a previous section of this chapter; in that section, ER-Mapper software itself was used in the lab where it was equipped with the ECW compression wizard. The one used in this section was downloaded online from http://www.ermapper.com/downloads/download_view.aspx?PRODUCT_VERSION_ID=152 on June 26, 2004.

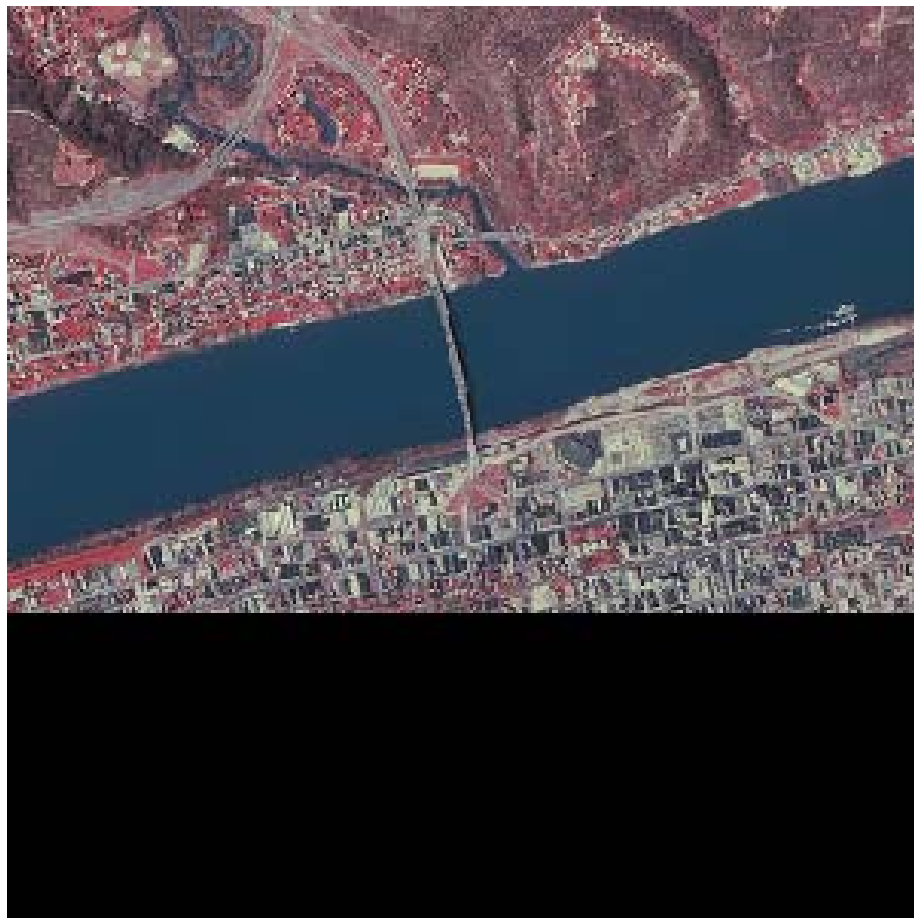


Figure 3-33 ECW with theoretical compression ratio= 30 times

Fig. 3.33 shows the output image of compressing the original TIFF image 30 times using this software; it is different than that shown in fig 3.25 (same **theoretical** compression ratio) but it shows very similar quality and a file size of 464KB which is greater than that for the image shown in fig. 3.25 (350 KB).

What was the reason?

Apparently, the reason is the **higher actual compression ratio** for fig 3.25 where it was (37.6) while it was (33.7) for fig. 3.33.

What does this mean?

This means that any image compression process is **not** consistent. From the result discussed above, we can conclude that compression ratios will not be the same if the same image was compressed twice using the same software and the input compression ratio.

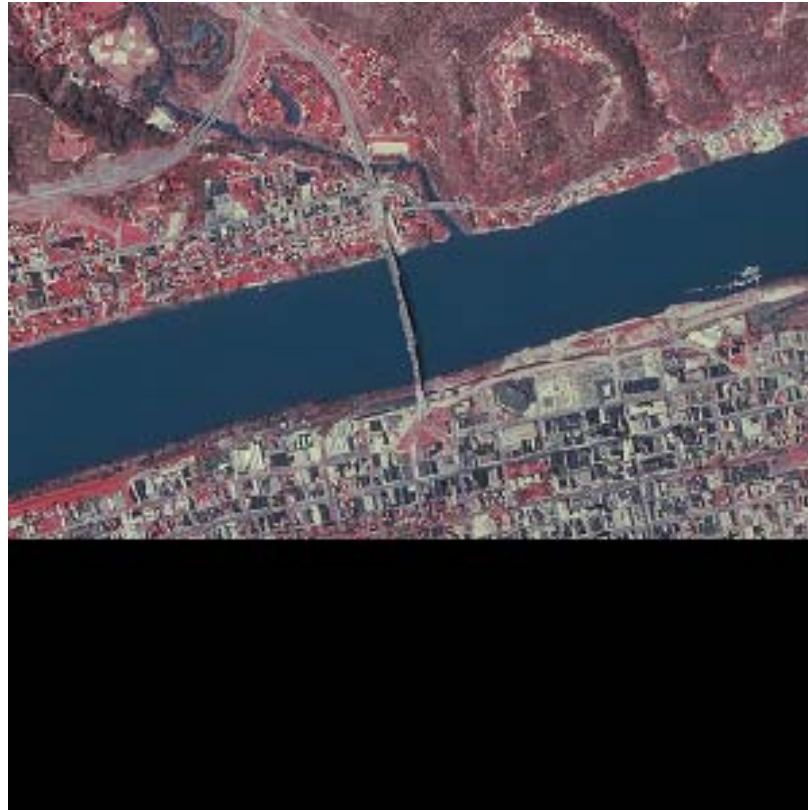


Figure 3-34 ECW with theoretical compression ratio= 35 times

Fig. 3.34 above shows the same input image (TIFF) compressed 35 times; the output is still a high quality DOQQ that can be used for mosaics, GeoBioPhysical imaging and internet applications.



Figure 3-35 ECW compressed with theoretical compression ratio= 90 times

In fig, 3.35, a 90 times (theoretical) compressed version; the actual compression ratio was only 58.8

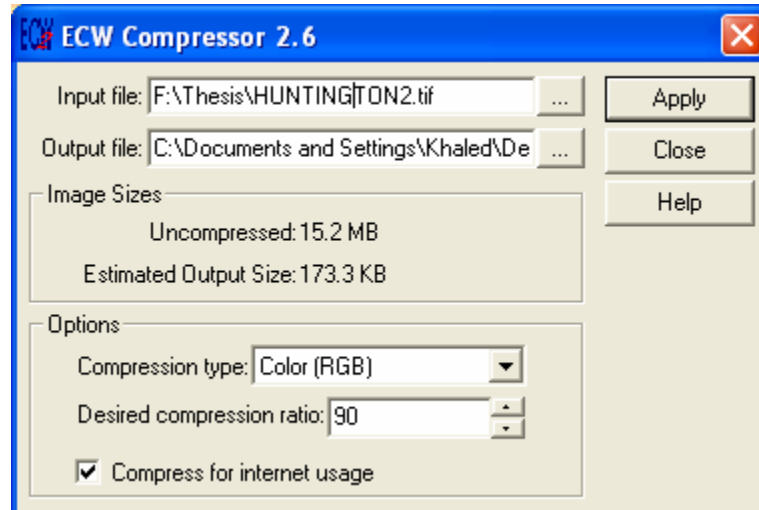


Figure 3-36 This software can estimate the output file size before even the process starts

Fig. 3.36 shows one of the advantages of this software; it can give some estimations about the possible outputs. This is useful since it helps the user manage his work and have some idea about what the output will look like and how big the output file will be. ER-Mapper compressing wizard does not have this characteristic.

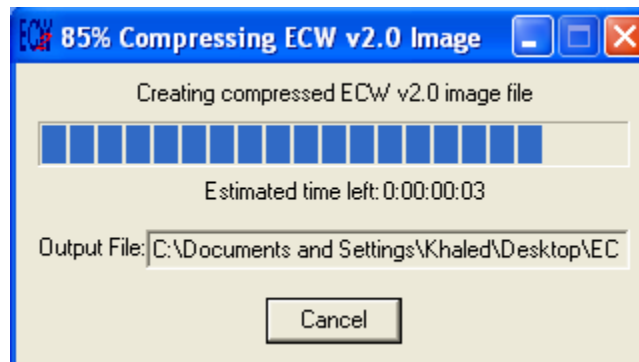


Figure 3-37 The compression process

Fig. 3.37 shows the progress of the compression process and the estimated time left that the system shows.

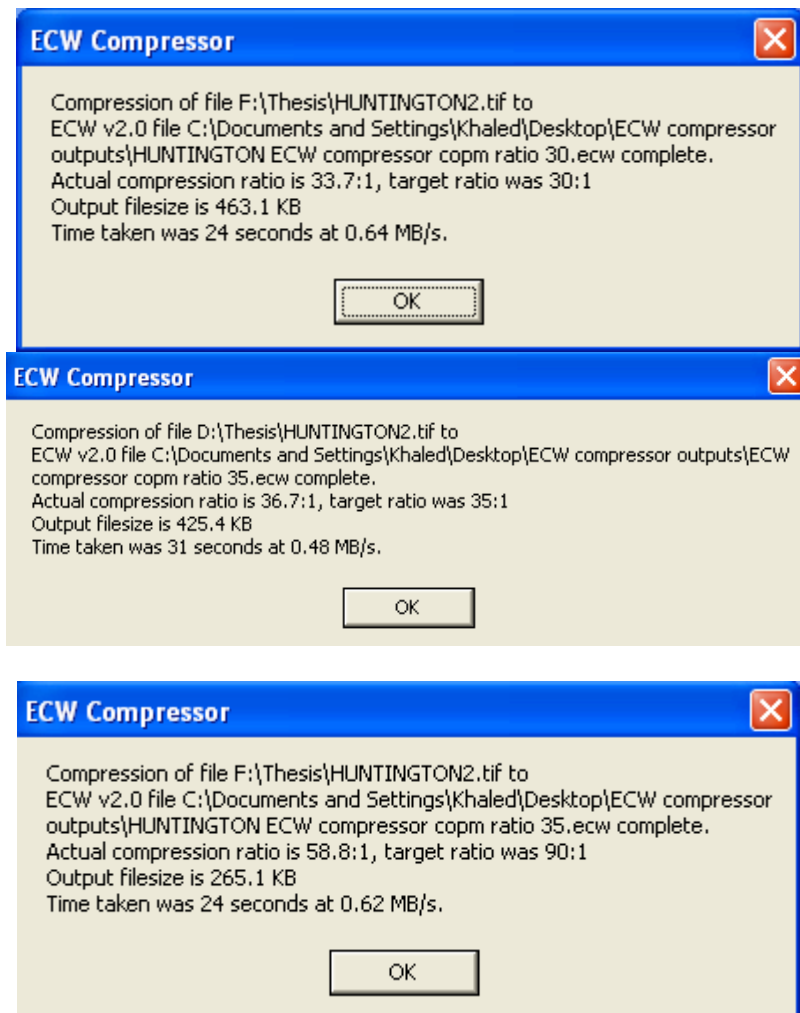


Figure 3-38 Compression results summary

Fig. 3.38 shows the compression process summary that the system gives right after the process is finished; this is similar to what the ER-Mapper compressing wizard gave in a previous section. (see figs. 3.26 and 3.28).

Actual results summary

Note here that ratios match with file sizes in case of ECW and Lura jp2

Name	Size	Type
ECW compressor copm ratio 30	464 KB	Ecw File
ECW compressor copm ratio 35	426 KB	Ecw File
ECW compressor copm ratio 90	266 KB	Ecw File

Table 3.3 Actual results summary

Calculations for this case;

$15.2 * 1024 / 90 = 172.9\text{K}$ (supposed to be; see fig. 3.34 above). But we have to divide by the **actual** compression ratio which is 58.8; so;

$15.2 * 1024 / 58.8 = 264.71\text{K}$ which is very close to 266.

$15.2 * 1024 / 35 = 444.71\text{K}$

Table 3.3 above shows the file size summary of the images in this section. The calculations show how misleading and mistaken the user can be if he uses the “theoretical” compression ratio used to start the compression process to calculate the

possible output file size. He/she should use the actual ratio that the summary results window gives.

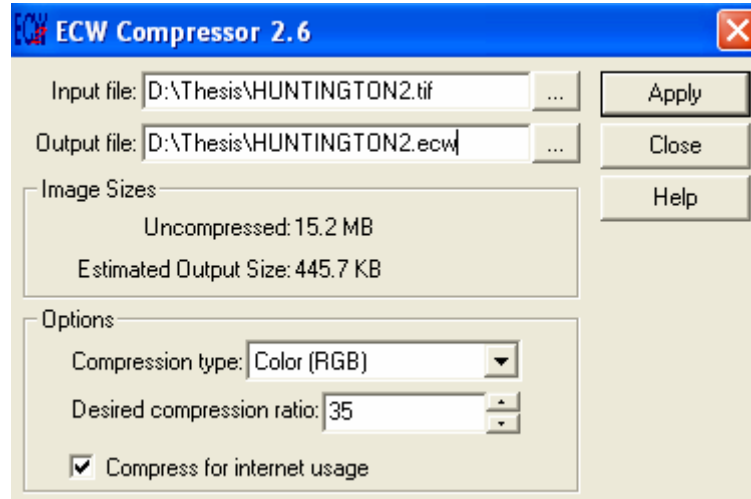


Figure 3-39 ECW pre-compression estimations

The actual compression ratio is 33.7; so;
 $15.2 * 1024 / 36.7 = 424.11\text{K}$ which is very close to 466.
 $15.2 * 1024 / 30 = 518.83\text{K}$
 The actual compression ratio is 33.7; so;
 $15.2 * 1024 / 33.7 = 461.861\text{K}$ which is very close to 464.

LuraWave SmartCompress 3.0 (1996-2002) by Algo Vision LuraTech GmbH.

For this part, the result of compressing the original image to 10% using "**PaperPort**" software (I decided to call it "HUNTINGTON10" to distinguish it) was picked to be further compressed using "LuraWave". LuraWave allows a wide range of compression options; all those options were used with different compression ratios.

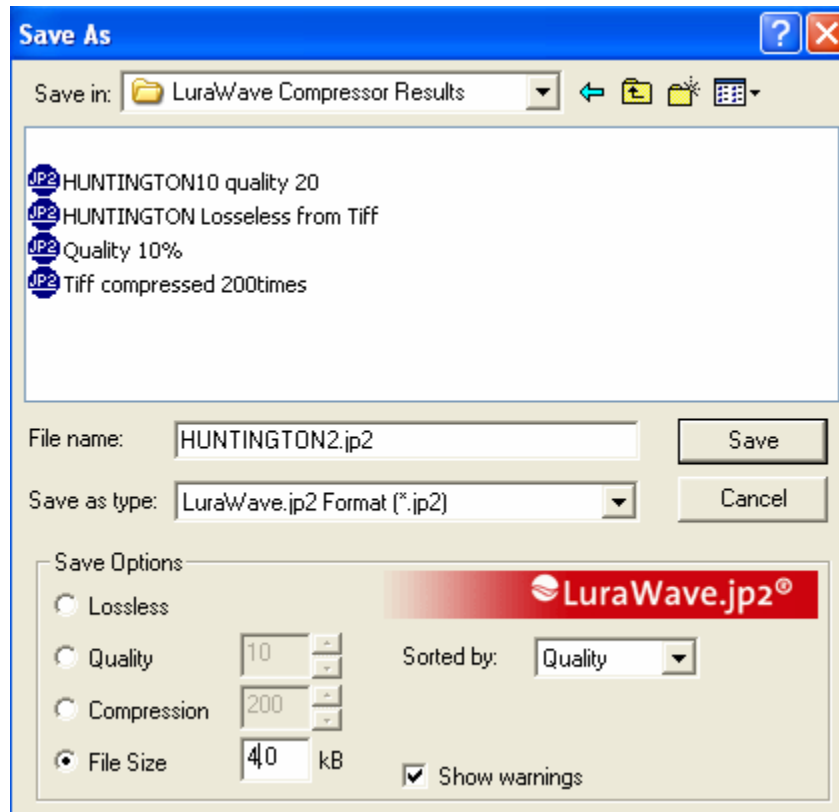


Figure 3-40 With LuraWave, the user has many options to save his image

Fig. 3.40 shows the several options and choices to save the image depending on the application. He/She can choose the lossless or PackBits compressions. He/She can also compress the image by pre-determining the output quality, compression ratio, or even the output file size.

“PackBits” Compression:

Figure 3-41 PackBits Compression

Fig. 3.41 shows the “PackBits compression technique”. The way this technique approaches the image can be related to its name. Using this technique, the compressing software compresses the image by encoding and lessening its bit as an electronic array of bytes (pixel values); bytes consist of bits.



Figure 3-42 PackBits Compression

Image size= 38.8KB

Fig. 3.42 shows an image compressed using the “PackBits compression technique”. 186k If we try to compare this image (Fig 3.42) with Huntington 10 (see fig. 3.43) we will find that it still retains more data and information with a smaller file size (38.8 KB) (186 KB for Huntington10). Even though it is hard to try to distinguish the differences between those two images in terms of the carried information and data at this scale, some color differences can be noticed using the eye; the observer can also compare the color information they both carry. Looking inside the yellow boxes where the red color

represents green (trees and grass), the observer can notice that color sharpness is higher in the “PackBits” case. Pixel colors are sharper and the pixels themselves are more distinguishable; this means that the image is clearer and can be zoomed to higher scales.



Figure 3-43 Huntington10

LuraWave’s JPEG

In any compression process, an input image is used. In this study the original uncompressed TIF image has been the input for all the experiments until this section. For the purpose of finding the effect of changing the input image on the output in addition to

the original objectives of this study, I decided to use different input images in this section and find the effect on some output images compressed exactly with the same technique and the same ratios; the only difference was the input image. The input image is indicated under each output.

Fig. 3.44 below shows a 50% of quality image compressed using LuraWave using the TIF (the original) image as the input.

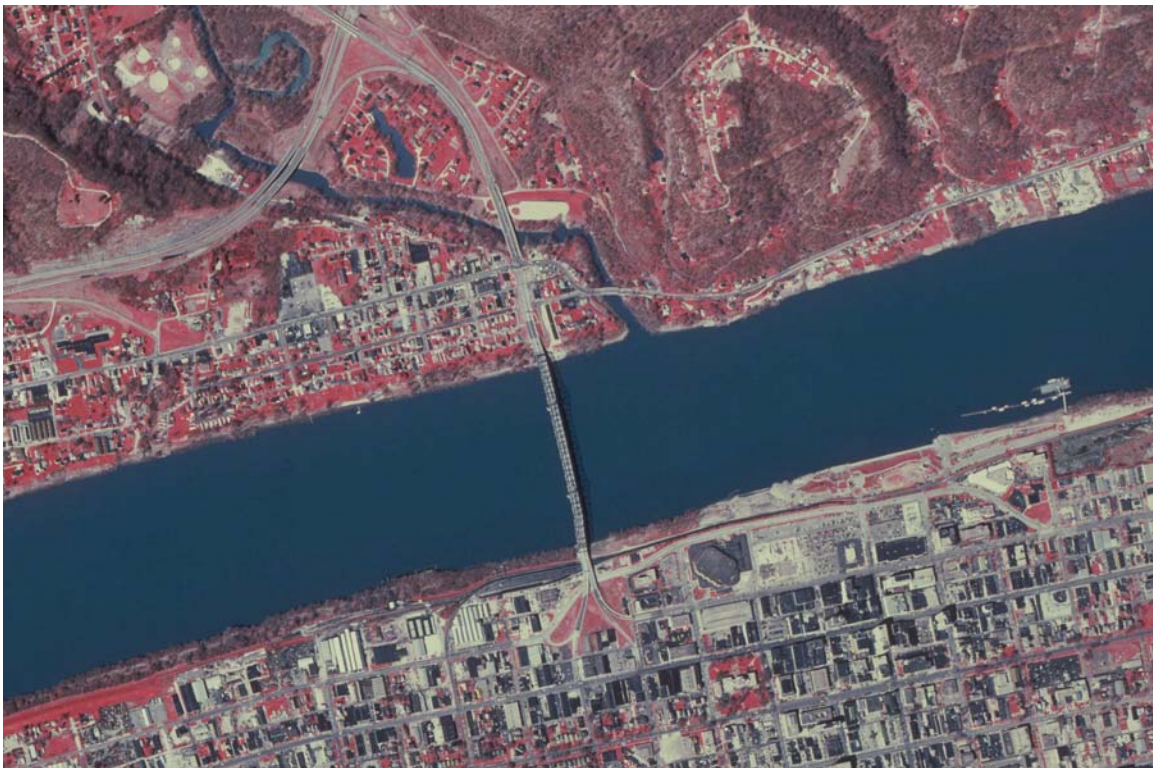


Figure 3-44 50% JPEG obtained by LuraWave
Compression ratio= 31 times. File size= 507K.

Fig. 3.45 below shows a 20% of quality image compressed using LuraWave using the original TIFF image as the input.



Figure 3-45 20% JPEG

Compression ratio= 54. File size= 290KB.

Fig. 3.46 below shows a 10% of quality image compressed using LuraWave using the original TIFF image as the input.



Figure 3-46 10% JPEG

Compression ratio= 82. File size= 190KB.

Observation:

If we compare the JPEG images obtained using LuraWave with previous ones obtained using PaperPort, we can generally say that LuraWave sustains much more color details for the same file size and/or compression ratio. (Compare Huntington10 (186KB) and the 10% JPEG shown above in fig. 3.46). Fig. 3.46 carries more color information for almost the same file size.

Input image is Huntington10



Figure 3-47 Huntington10

Huntington10; Compression rate= 84. File size= 185KB.

Fig. 3.47 above shows the 10% JPEG (the output of the PaperPort) which will be used as the input for the following output image (fig. 3.48).

Fig. 3.48 below shows a 50% of quality image compressed using LuraWave using Huntington 10 (10% JPEG output of the PaperPort) as the input.

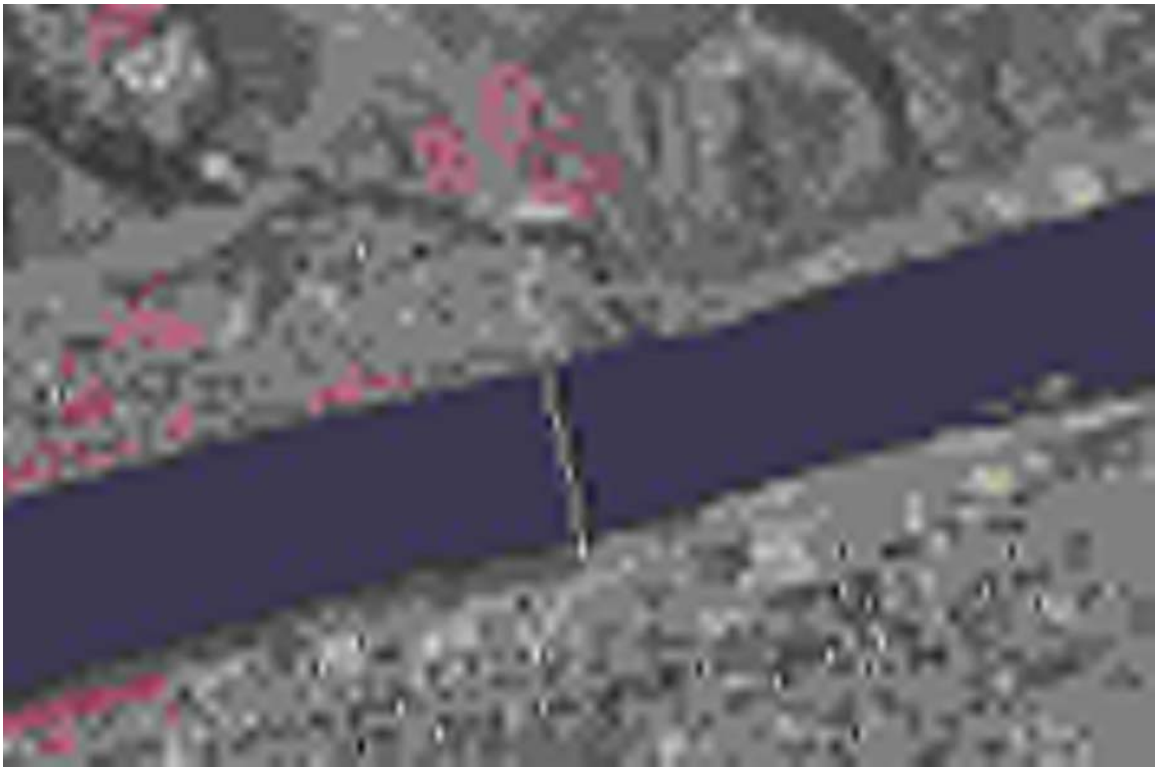


Figure 3-48 50% of Huntington10
Compression ratio 7807. File size= 1.99KB.

The image has very low quality and very few number of classes.

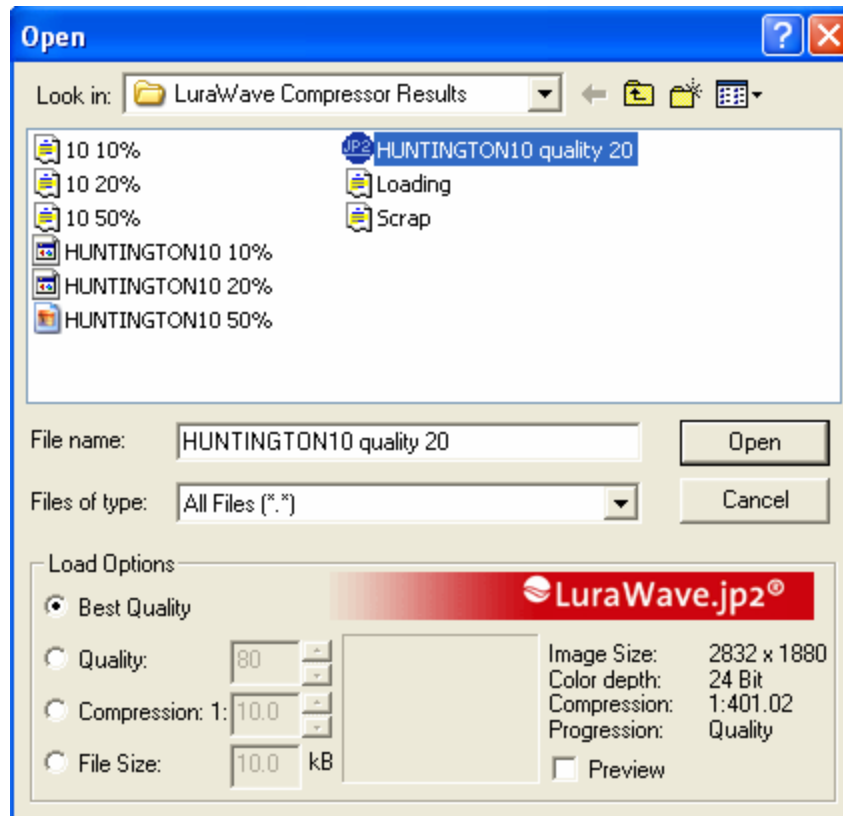


Figure 3-49 The ability of LuraWave to view images with different quality selections

Another useful characteristic of LuraWave in addition to the ability to save an image with different quality and compression ratios, the user can even **open** a saved image with the same options depending on the application. This helps in saving time, work, and space while doing the job efficiently (see figure 3.49 above).

JP2

Now, we go back to using original TIFF only as the input.

With JP2, the image can be compressed and saved in many different ways and options where quality, resolution, color, compression ratio, and file size can be controlled by the user before the compression process starts (see fig. 3.50 bellow).

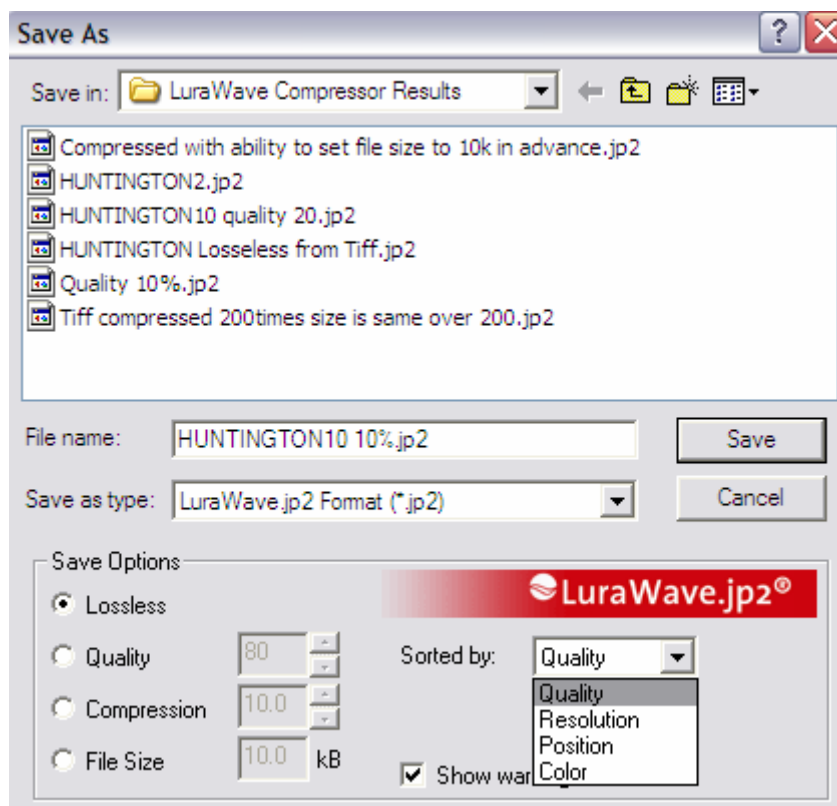


Figure 3-50 The ability of LuraWave to save images with different quality and file size selections

Lossless compression from TIF to JP2



Figure 3-51 Lossless compression from TIF to JP2

File size= 4.77MB

Figure 3.51 shows an effective way to compress TIFF image in the lossless way using LuraWave; the result was interesting since the file size was reduced from 15.2MB to 4.77MB only in a lossless way!

Quality control (the output is still JP2)**Figure 3-52 10% quality**

File size= 34.5KB

Fig. 3.52 above shows the new useful JP2 (the JPEG 2000 file extension that belongs to LuraWave). Compressing TIFF to 10% quality gave a 34.5KB file size image with a much higher quality than if the same file size image was gotten using many other regular JPEG compressing software programs.



Figure 3-53 20% Quality

File size= 38.8KB

Fig. 3.53 shows a **20%** compressed JP2 image with relatively good results. The file size and the quality are good. The noticeable thing here is that an image of twice the quality (20%) was obtained for only 4.3KB (note that quality here does not mean my personal preference; it is the quality percentage that the user defines before compressing the image; i.e. it is a relative percentage used to compress the image just like compression ratio and file size).

Size controllability (output is still JP2)

Note here that in this section different inputs might be used.

Tiff (as the input)

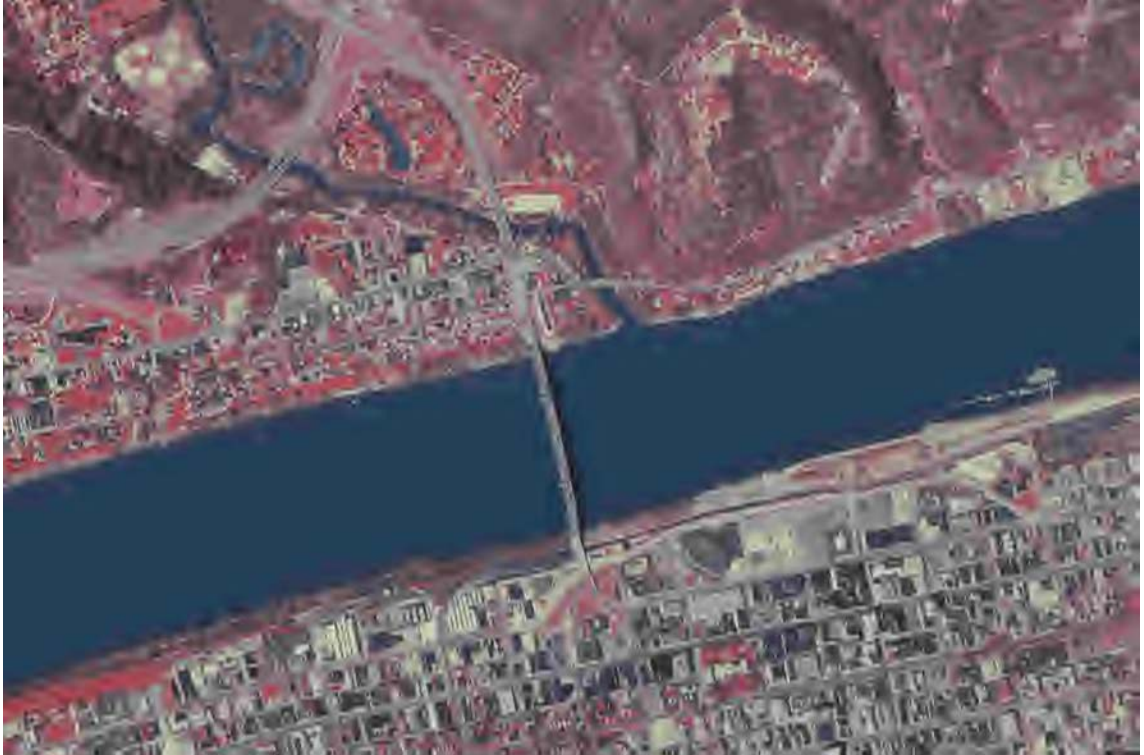


Figure 3-54 10K. Compression ratio= 1560

Fig. 3.54 above shows how the user can control the output compressed image by pre-determining the output file size; it shows a 10KB image compressed using LuraWave's JP2. If we calculate the compression ratio, it is 1560 times.

The figure above shows a small file size image that saves much space; but, the quality is not good and the image will definitely give wrong results if used in any classification process or other kinds of research.



Figure 3-55 6KB. Compression ratio= 2601

In fig. 3.55he image is very fuzzy and the river started showing two different colors.

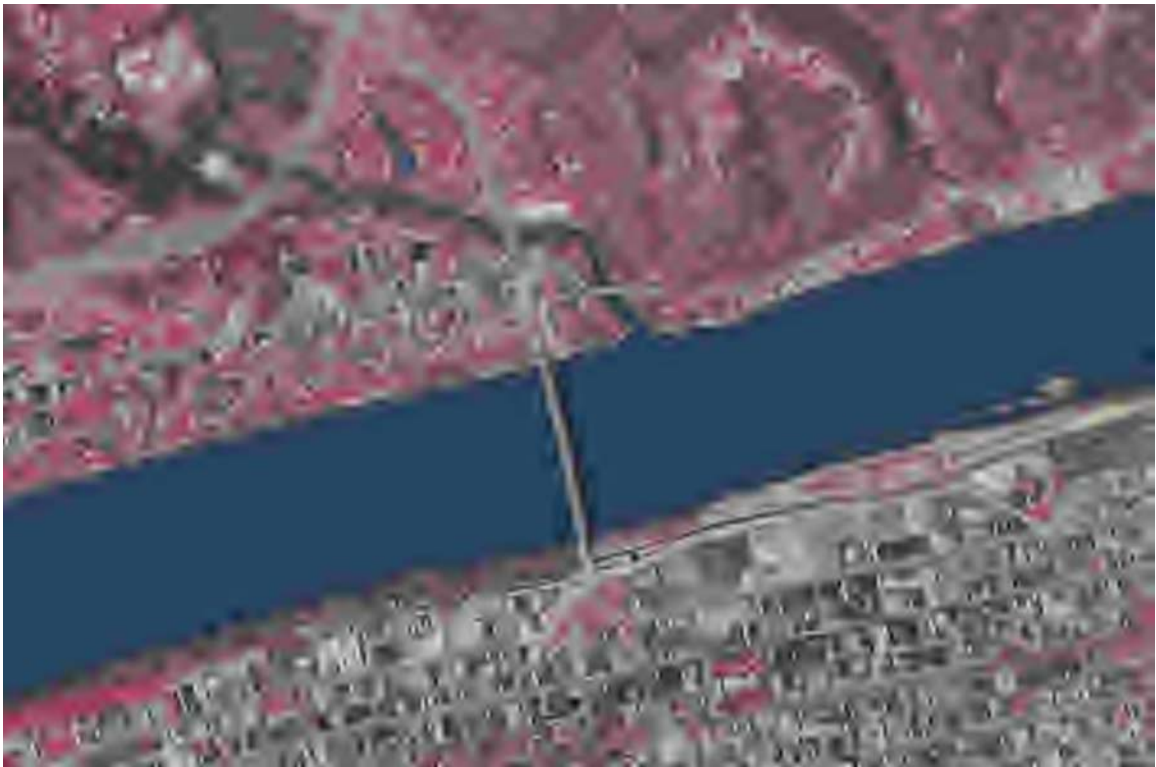


Figure 3-56 4K. Compression ratio= 3900

Fig. 3.56 shows a fuzzier image with smaller file size; the lines and the edges are hardly distinguished.

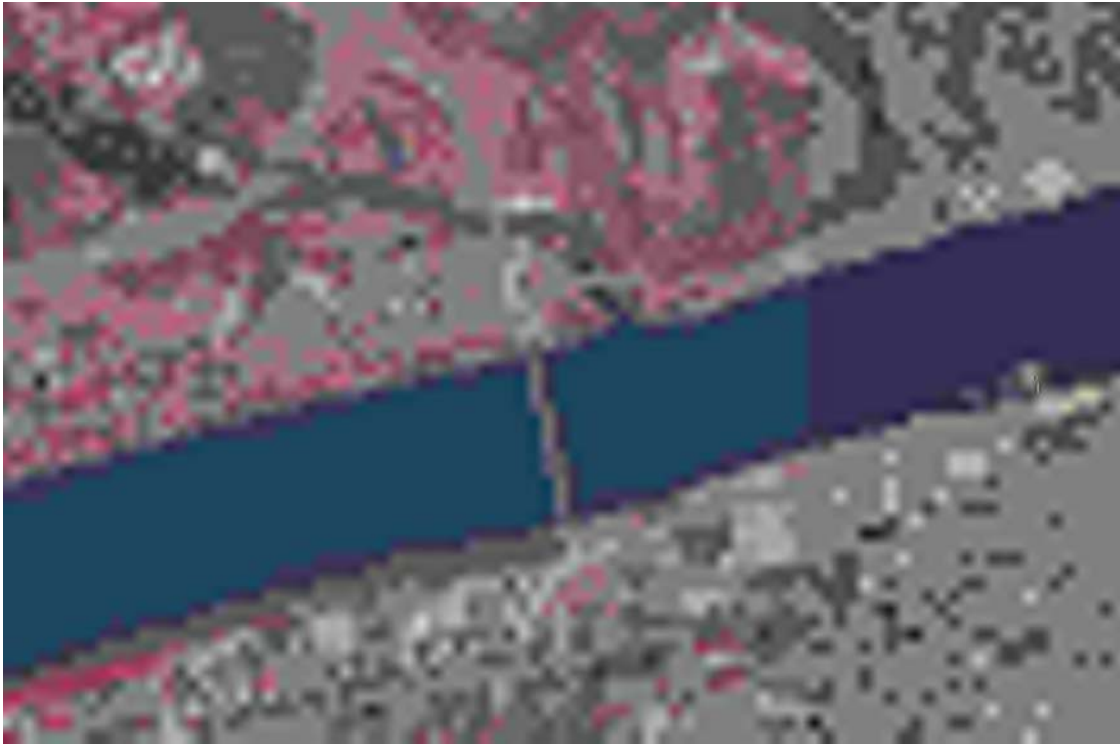


Figure 3-57 2K. Compression ratio= 7799

Figs. 3.54-3.57 show the outputs of pre-determining the output file size to 6,4, and 2KB respectively. Compression ratios are calculated (see below figures). Image quality is low, the numbers of classes are not more than 5 or 6, and the river clearly shows two different colors.

Huntington10 as the input image

Figure 3-58 A 10KB image. Compression ratio= 1560. Input is Huntington10

10K image. Compression ratio= 1560.

Comparing fig. 3.58 with fig. 3.54, we find that we get less quality for the same file size if we use a less quality input. This is because of the input image used. Huntington10 had been compressed using PaperPort and some of the data it carries was lost due to the compression process. This means that when it was compressed again to a 10KB file size image (fig. 3.58) it gave less quality than that gotten by compressing TIFF to the same file size (see fig. 3.54).



Figure 3-59 6K image. Compression ratio= 2600. Input is Huntington10



Figure 3-60 4K image. Compression ratio= 3897. Input is Huntington10

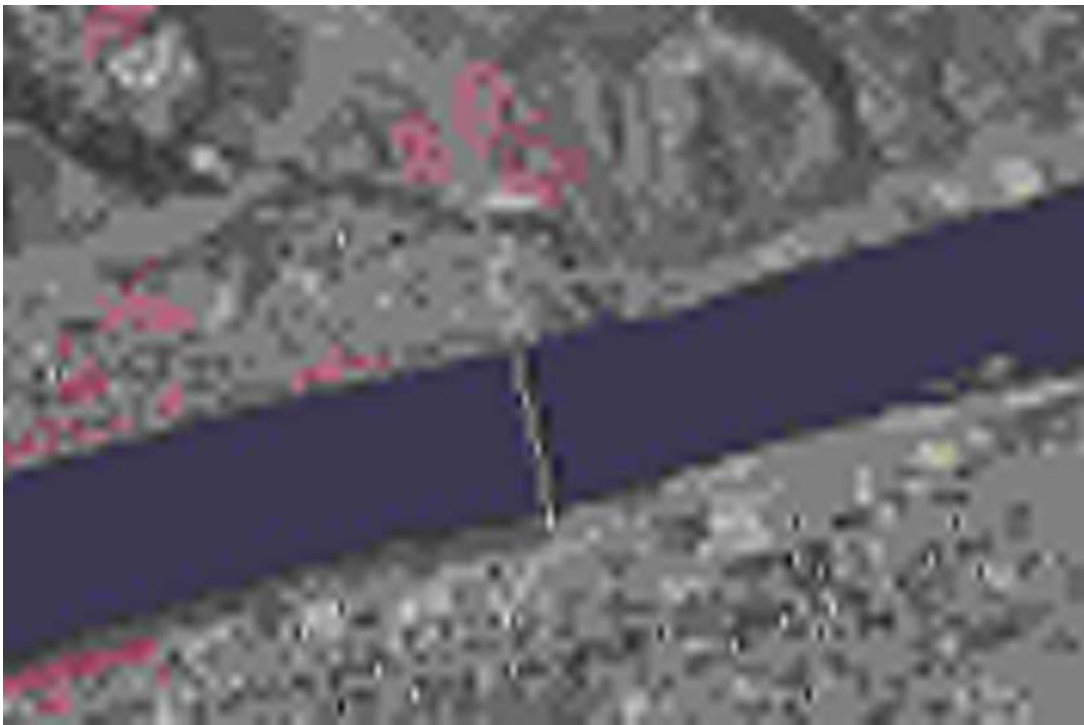


Figure 3-61 2K image. Compression ratio= 7807. Input is Huntington10

Actual size was 1.99KB and 4K on disk.

In fig. 3.61, the number of classes is not more than 3-4. Pixel values do not vary much and so many objects are going to be confused if a classification process is to be performed using this image. In the original TIFF image, there are many different objects; in the image shown in fig 3.61 all buildings, roads, and green will seem to be one kind of objects.

Compare figs. 3.54-3.57 with figs. 3.58-3.61 to find that TIF is a **better choice to get pre-determined file size images**. This means that since the **output file size is the same**, and the same disk space will be taken, the user **should use better quality images as inputs**.



Figure 3-62 LuraWave can the compression ratio of any JPEG image

Fig. 3.62 shows the BackBits compressed image with its compression ratio; as stated before, BackBits compression gives very good results. LuraWave can give the compression ratio of any JPEG image if it was opened using this software.

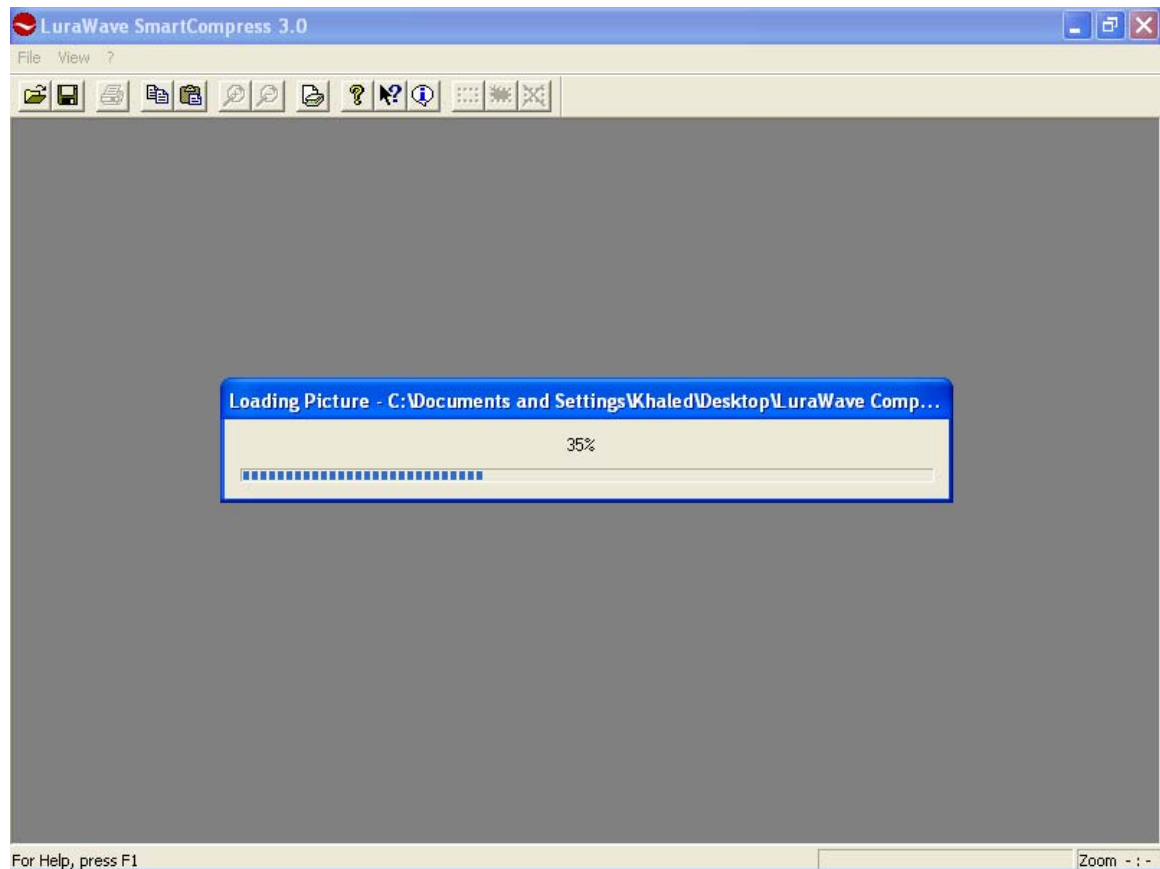


Figure 3-63 the more the image is compressed, the longer it will take to open it

Fig. 3.63 **shows the effect of compression on the access time**; the more the image is compressed, the more complex the process will be; this means that the decoding process will be more complicated and the image will take longer time to open relative to another uncompressed image (or less compressed) of the same size or a little larger. This means that the small size image will be transmitted faster (through the internet for example) while it will take **relatively** longer to be decoded at the destination. Anyway, the

complicated decoding process sometimes takes less time than opening the original large un-compressed file because it needs less CPU execution cycles due to the much smaller size. For example, while compressing and opening the last image (which was compressed to 2K using LuraWave), I noticed some delay; i.e. it took a little bit longer time than it took compressing and opening the other less compressed images. I concluded that the process was more complex for the 2K image. Anyway, the opening and loading times for that complexly compressed 2K image was less than that for the un-compressed 15.2MB Tiff one.

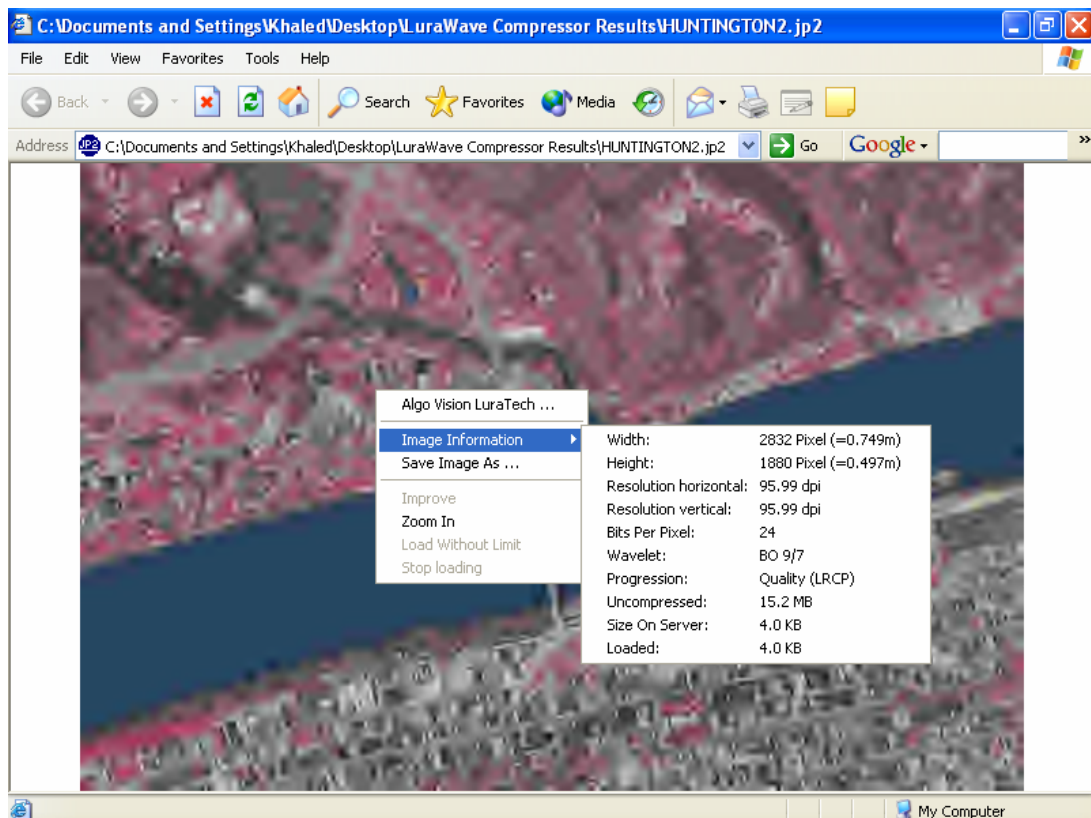


Figure 3-64 JP2 is designed for internet usage.

Fig. 3.64 shows that JPEG 2000 formats can be used for and opened easily with Internet Explorer; many image properties and information can be gotten just by right clicking the mouse and pointing it to “Image Information”.

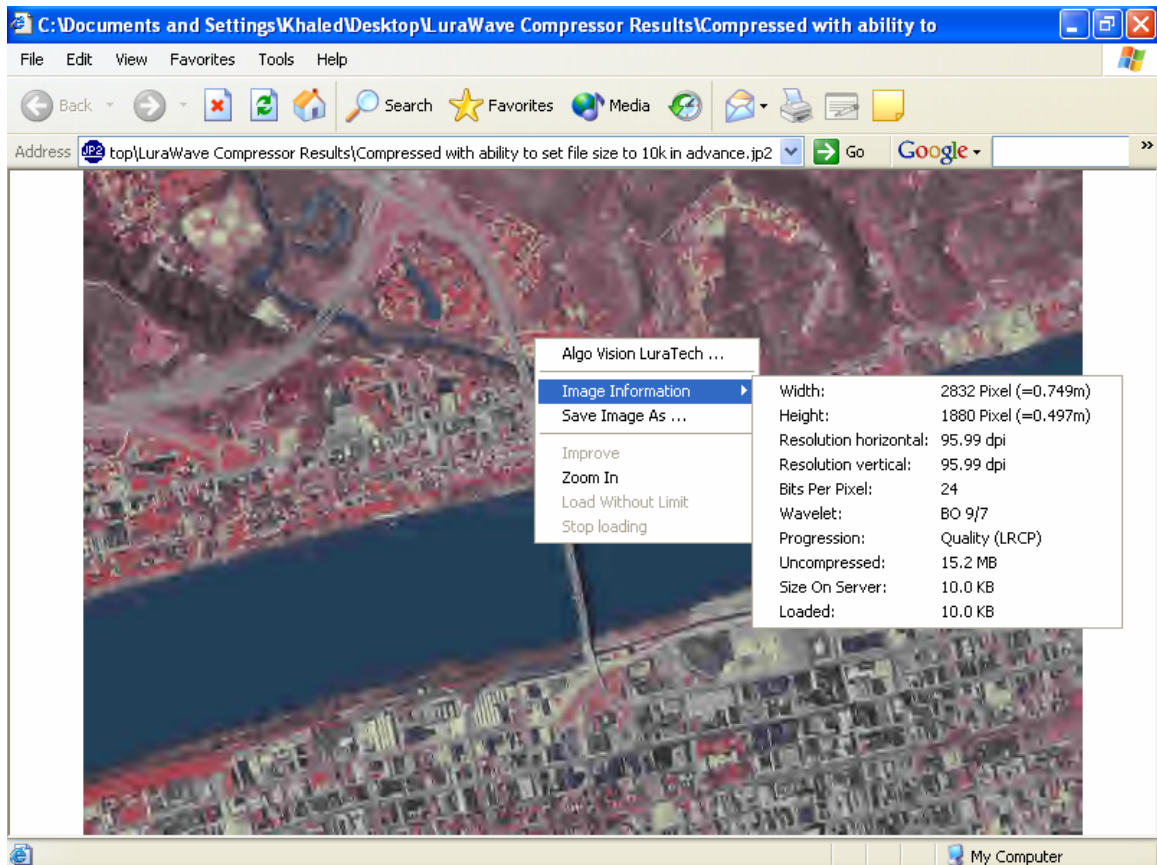


Figure 3-65 The 10KB JP2 image

Fig. 3.65 shows how we can get information about JP2 images using Internet Explorer. Even the file size of the un-compressed source can be determined this way.



Figure 3-66 The JP2 icon

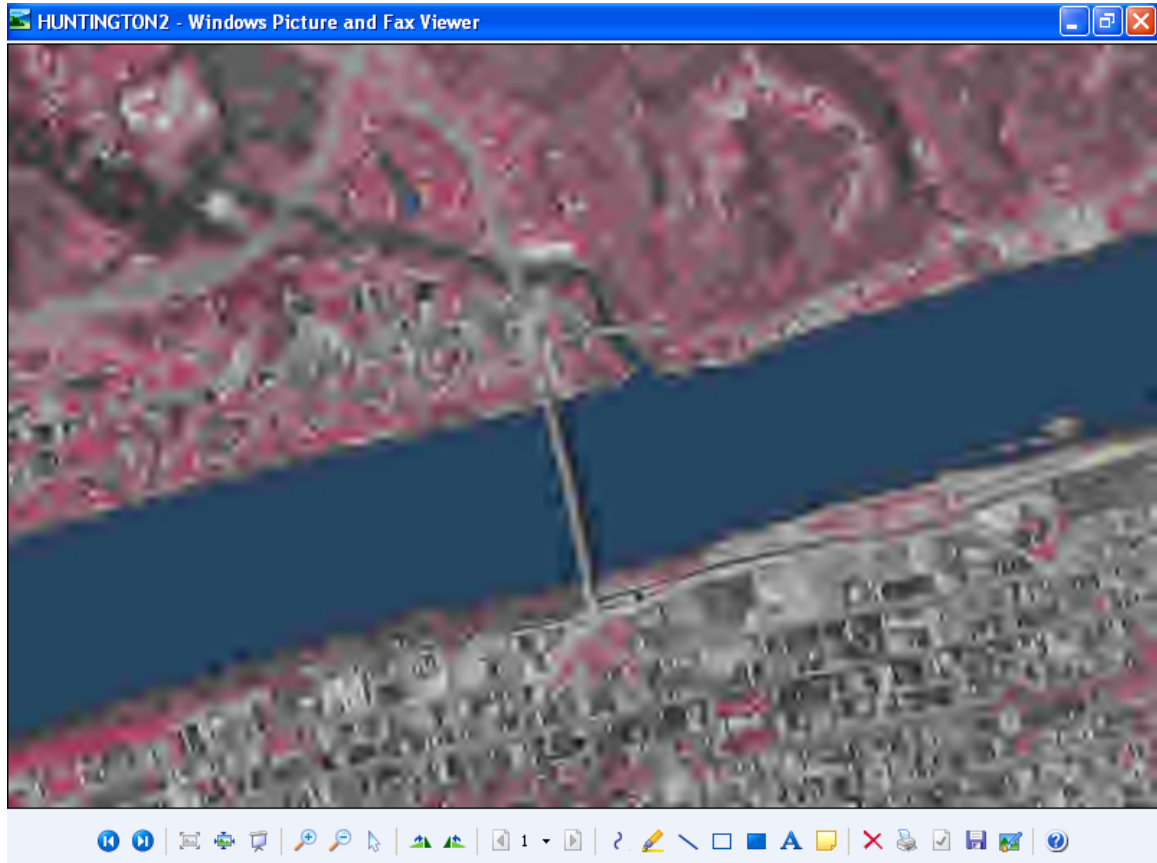


Figure 3-67 The image is saved again as TIFF. File size= 15.2MB again.

JP2 images can be decompressed using Internet Explorer and the original file size can be recovered. If lossy compression techniques were used, the lost data cannot be recovered; this becomes clear if we compare fig. 3.67 with TIFF.

Using this technique of de-compressing imagery, the user can save the decompressed image using many different formats.

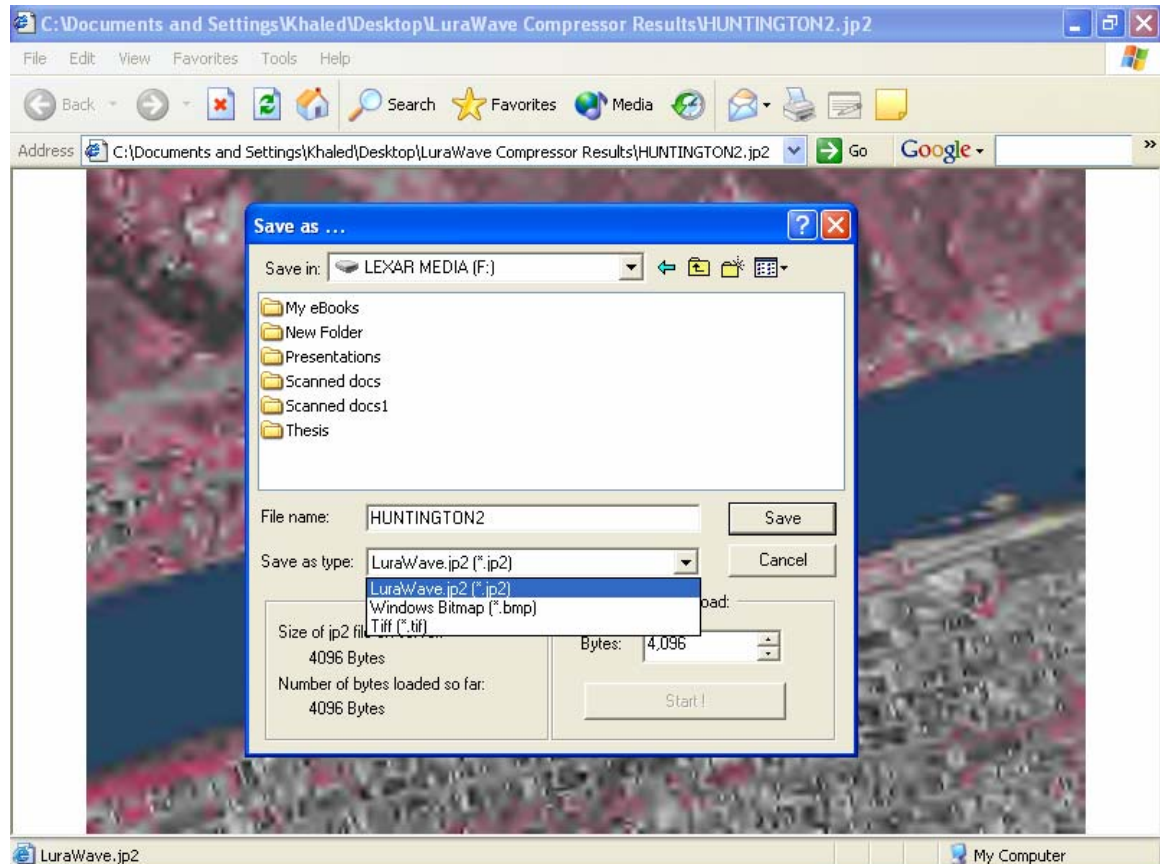


Figure 3-68 The ability to save the image again using Internet Explorer

Fig. 3.68 shows the ability of Internet Explorer to save the image again and retrieve the file size of the original source. Anyway, this is not recommended for images compressed using lossy techniques since the lost information can not be recovered. This technique recovers the original image if the original was compressed in a lossless way.

The ability to control compression ratio

1. TIFF as the input



Figure 3-69 The TIFF image

Tiff file compressed 100 times to JP2. File size= 155KB. Almost twice the one in fig.3.66 which was compressed 200 times

LuraWave gives the ability to compress the image by determining the input to output compression ratio. Figs. 3.69-3.71.



Figure 3-70 Tiff file compressed 200 times to JP2. File size= 77.9KB

Going back to the road area inside the blue box, this image is 77.9KB but the two sides of the road are still distinguishable and the line that separates them is still clear.



Figure 3-71 Tiff file compressed 900 times to JP2.

File size= 17.3KB. Exactly= $15.2 \times 1024 / 900$.

Note that 1KB= 1024Bytes.

2. Now, using (Huntington10) as the input



Figure 3-72 100 times.

. Compression ratio= 100. File size= 156KB

Calculations: $15.2 * 1024 / 100 = 155.648 \text{KB}$; almost= 156K.

There are two proofs that Huntington10 was compressed according to the size of the original TIF image. The compression ratio for the last image and the following two match the ratio between 15.2MB and the size of the result.

LuraWave can find the size of the source un-compressed image from which the compressed image can be obtained (especially JP2) even if the compression happened

using a different computer and software. LuraWave also can resave the compressed image with the size of the original and with the same extension (TIF, for example). Refer to the re-saved JP2 image in the last section.

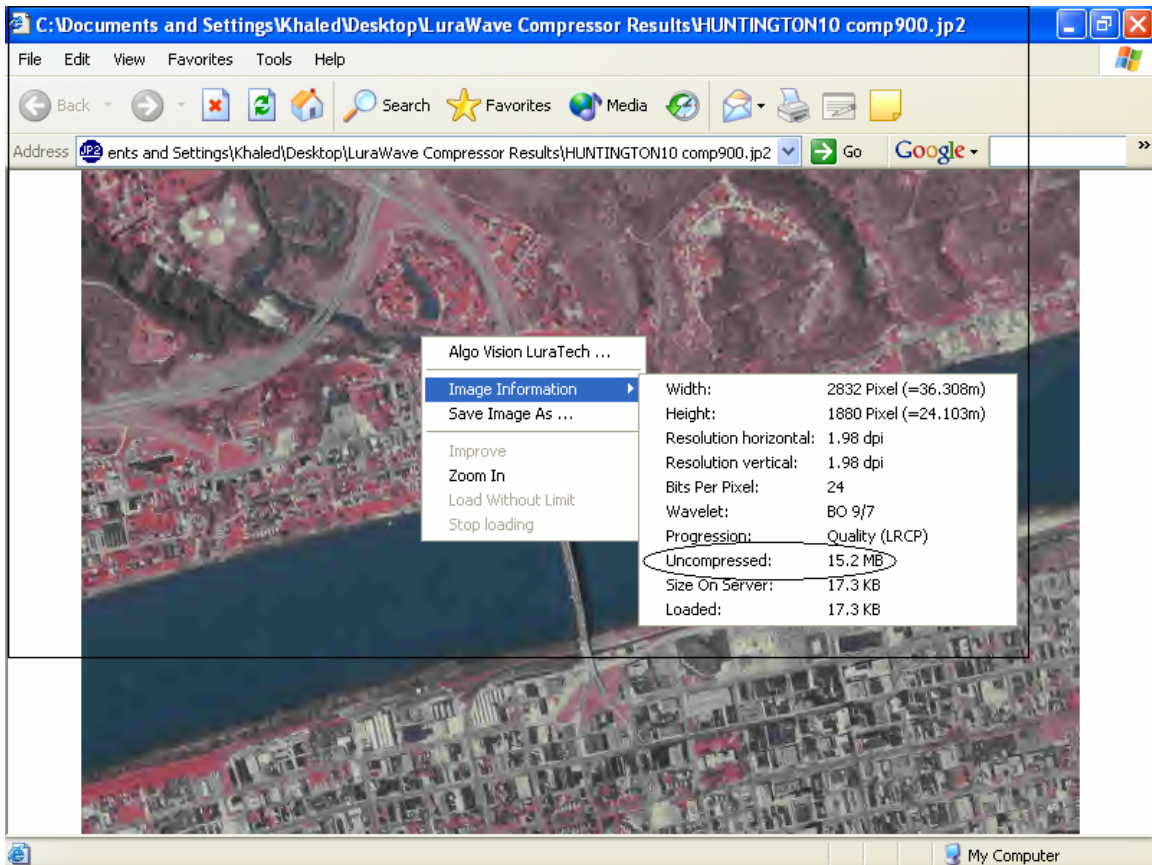


Figure 3-73 Retrieving the file size of the original image by viewing the properties of any Jp2 image

Fig. 3.73 above proves that some software programs such as LuraWave can determine the file size of the original uncompressed image.



Figure 3-74 200 times

Fig. 3.74 shows a Compression ratio of 200 times. File size= 78KB.
 $15.2M * 1024 / 200 = 77.824K$ which is pretty close to the file size (78K).

This image shows less quality than that corresponds to TIF image (when TIFF was the input).



Figure 3-75 900 times.

Fig. 3.75 shows a Compression ratio of 900 times. File size= 18KB.
 $15.2M * 1024 / 900 = 17.29K$ which is pretty close to the file size (18K). This means that this software gives accurate results in terms of compression ratios as well as it gives in terms of “predefined” file size.

LuraWave gives high quality small size images. But looking to the road area inside the blue box and comparing it to other edges in the image, we can conclude that LuraWave software blends the edges more than the previously discussed software programs. It also blends different edges differently and to different extents trying to keep the main objects of the image clearly distinguishable. (Look at the bridge in fig. 3.71; it is not very fuzzy while the road area is too fuzzy).

4. CHAPTER IV

Results

Compression and data

Digital images are binary numbers expressing numerical values (Brightness value).

Working on the image and compressing it using compression techniques by altering those values to get smaller file sizes and transmission time has its effects on the data and the image. Although compression has its own benefits, it also has its drawbacks due to the changes it causes in the data.

Despite the fact that the data and the image that one has after compression is related by some means to the original, they are different and new. In many applications where images and pictures are needed for commercial, internet, TV, or any normal usage, the difference has little significance and the most important factors are speed of transmission and display in addition to cost. As soon as the image and the video are of acceptable and competing quality, compression limits are a little flexible. But, when images and pictures are to be used for different studies or researches, defense, military, transportation, development, projects.....etc. where algorithms and operations have to be performed on images and data for the benefits of the study, and where images and data are accompanied by related documentations (or what is called “Meta Data”), there are limits and restrictions to any changes performed on any image to insure reliability, accuracy, and sustained compatibility of the data and the images.

In Marshall University's Optical and Digital Image Processing Laboratory aerial and satellite digital images are used for transportation and mapping study purposes, related data, and metadata. Fast transmission of small file size images and data is very important.

What does this mean? Does it mean that we can compress our images and data without limits just to save time and money?

Accuracy, reliability, and compatibility of work must be balanced with the limited compatibility between available software programs and their ability to work on images with specific formats and extensions. A balance is needed between getting further compressed images and getting new computers and networks of high speeds and high storage capacities. Not all the work can be performed using low quality images and slow networks and computers of small storage capacities.

What does compression do to the data?

To show the effect of compression on the data, the original TIF image was picked to be studied. Several other compressed forms of that image were also picked to show the difference between them and the original one from a statistical point of view.

Images and statistics

As a two-dimensional array of numerical values, a digital image can be studied from numerical and statistical points of view. Scatter diagrams and histograms of the image

can be used to show the relations between the numerical values and the other and the relations between the values and their frequency of occurrence.

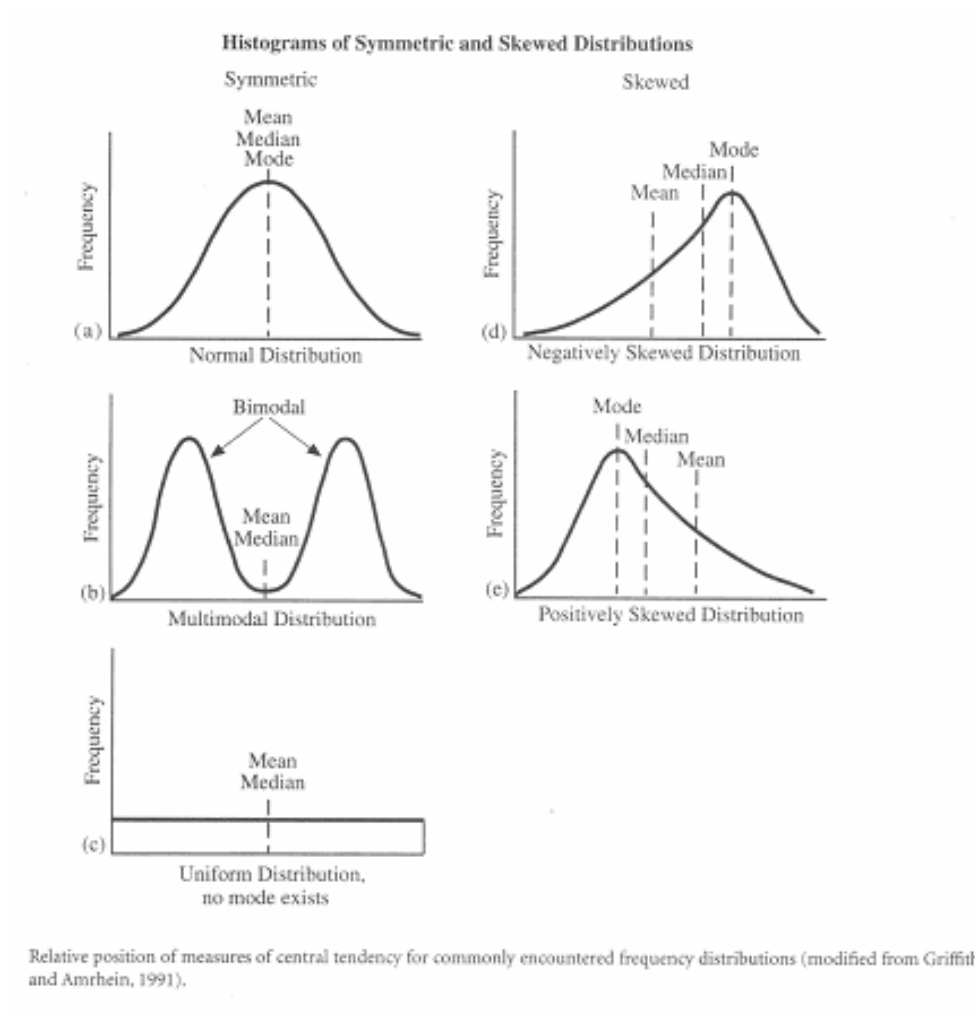


Figure 4-1 Image Statistics

In fig. 4.1, histograms of digital images show the relation between the brightness values and their frequency of occurrence.

Many useful operations such as different classification processes can be performed on the image according to the available statistical information, data, and the distribution of that data.

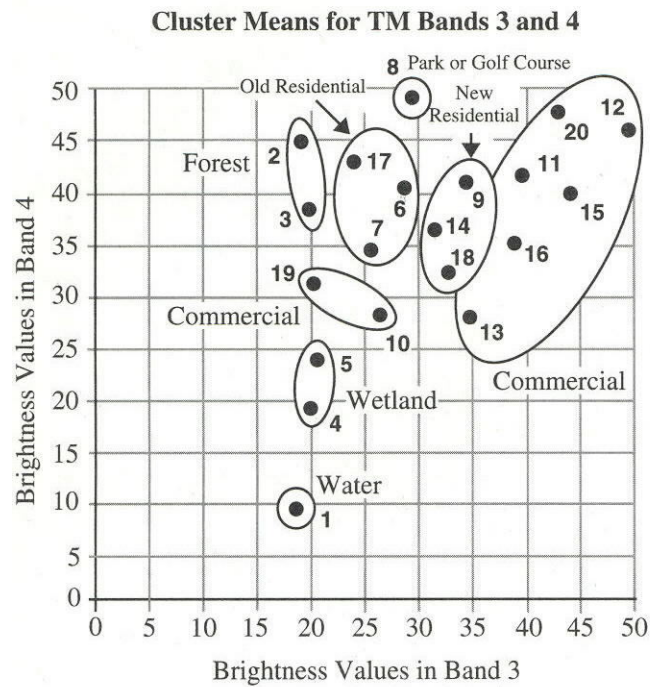


Figure 4-2 Classification

Source: Jensen, R. J. (1996). Introductory Digital Image Processing. New Jersey. Prentice Hall, Inc.

Fig. 4.2 shows the how the classification of data can be applied on data using scattergrams.

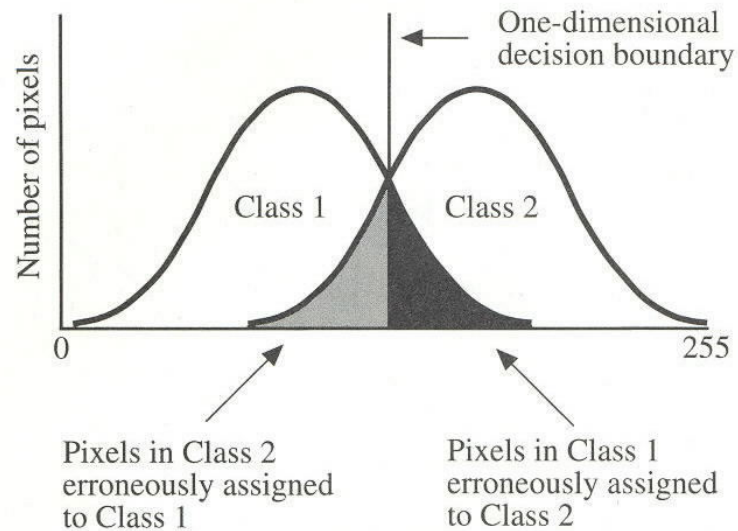


Figure 4-3 Changing pixel values can change the results of the classification process and decisions as a result

Source: Jensen, R. J. (1996). *Introductory Digital Image Processing*. New Jersey: Prentice Hall, Inc.

Pixel values can be altered and classified according to their numerical values (see fig. 4.3 above). Those values are altered by compression. Does this mean that severe compression processes can change classification results and classification decisions? This is what will be discussed in the next few pages.

To show how compression affects any classification process, let's consider the area enclosed by the yellow box in fig. 4.4. Assume that the brightness value before compression for a pixel inside that box was 35 in band 3 and 37 in band 4, it will be assigned to "new residential area". But, if after compression the brightness value of the

pixel was altered in band 3 and became 25, the pixel will be assigned to “New residential”. If a researcher was studying such an image or a mosaic for a specific city trying to figure out how fast the city is being developed by finding out the ratio between new and old buildings, his work might become inaccurate if he uses an image that was severely compressed such as Huntington10.

Studied images and results

The following are the images picked as samples to show the effect of compression on data and decision-making during the classification process:

1. **The original decompressed TIF** (obtained online from <http://www.wvgis.wvu.edu/data/dataset.php?action=search&ID=74> and decompressed using zip and MrSID Viewer version 2.0 (1997-1998) by Lizard Tech, Inc. & International Land Systems).



Figure 4-4 Buildings in TIFF image

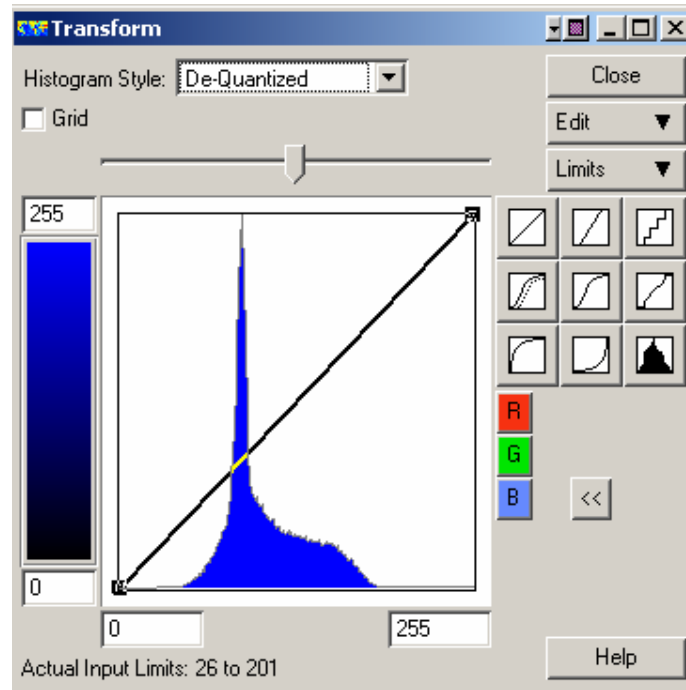


Figure 4-5 Histogram for TIFF

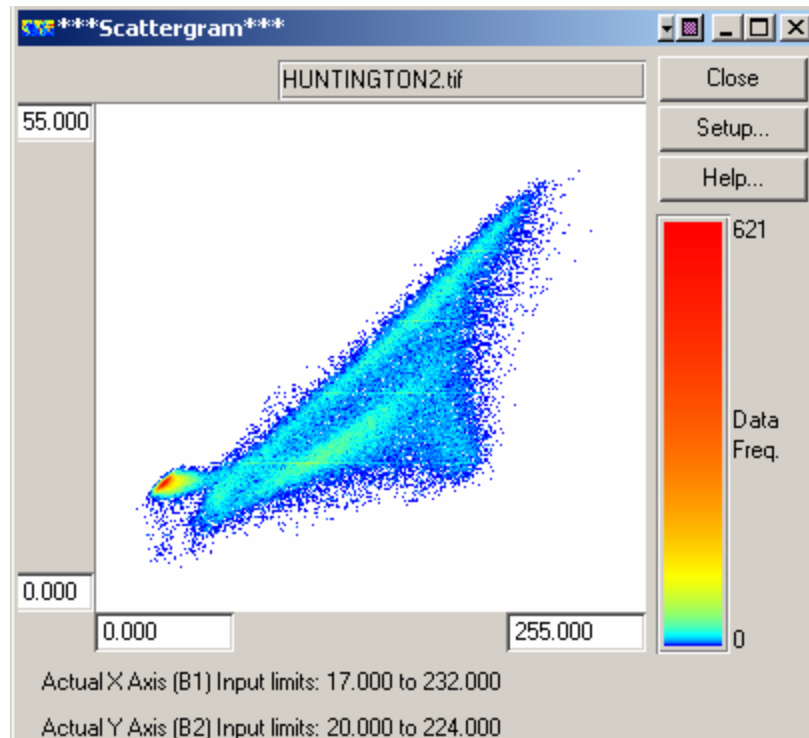


Figure 4-6 Scattergram for TIFF

Figs 4.4-4.6 show TIF image, its histogram, and its scattergram. The histogram shows that the BV's vary between 26 and 201; BV variation shows different points where samples show distinguishable objects. Samples with high frequencies show very similar BV's of high occurrence in the image, and this is clearly the river (water).

To notice the differences between this image (TIF) and the next compressed forms, we can notice the changes that will occur **in scattergrams** and the histograms as the image is further compressed; the best area to notice such changes is the one that represents water

(red color in Fig. 4.7), and peaks (high frequency values with small difference in brightness values) **in histograms**.

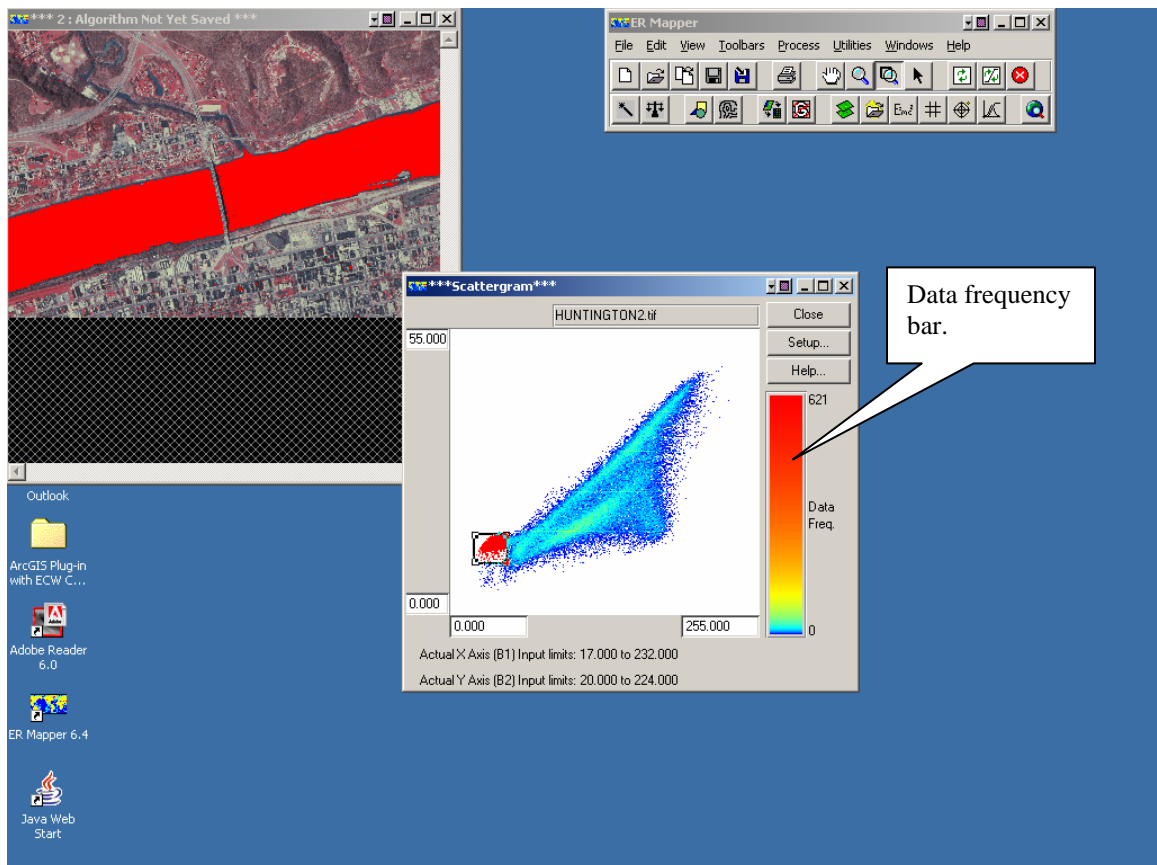


Figure 4-7 location of the river area on the corresponding Scattergram

Fig. 4.7 above show how the user can view and control an algorithm using ER-Mapper. He/She can pick some areas in the scattergram and view them in the image window and vise versa.

Showing a digital image in different ways such as histograms and scattergrams which are different than the regular “picture or “binary map”” displays is useful for different applications. The usefulness of scattergrams is further boosted by the “Data frequency” bar (in the right side of fig. 4.7). Scattergrams help to distinguish and classify objects as discussed above.

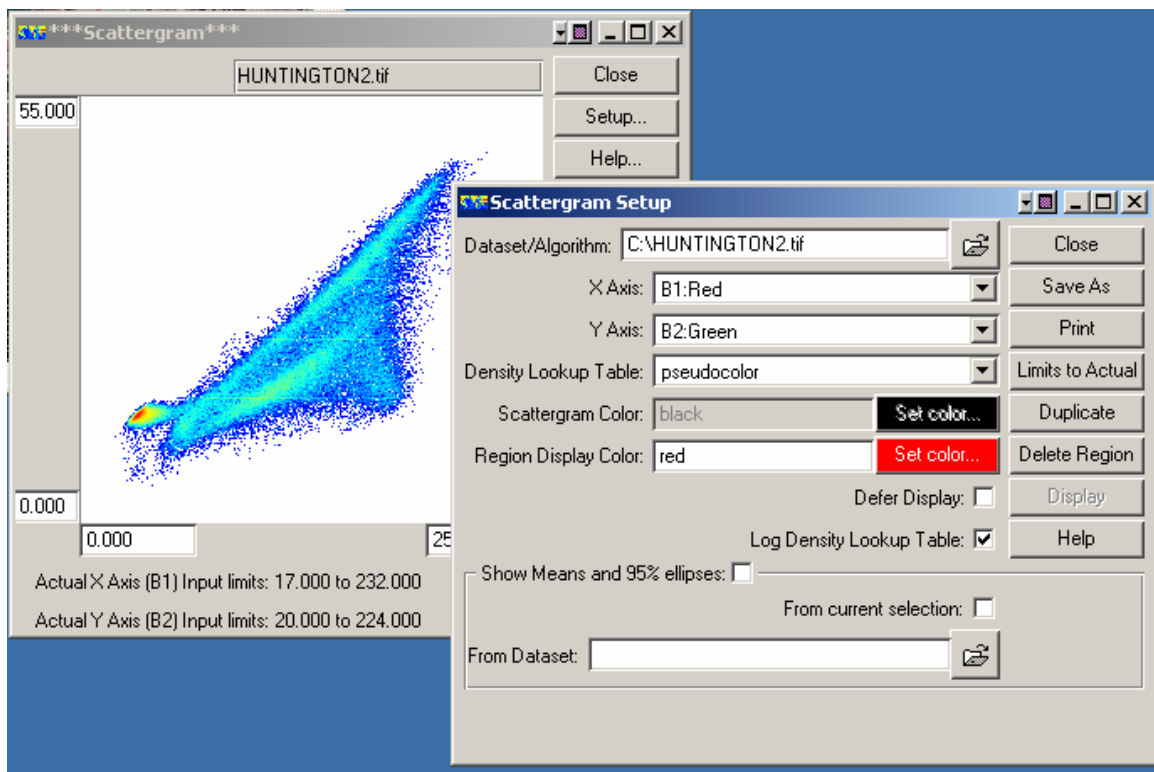


Figure 4-8 The x-axis represents the Red band while the y-axis represents the Green one

Fig. 4.8 shows the color bands used as the axes for the scattergram used in this study.

See “a” and “b” below.

Note the following:

- a. In all scattergrams, the “x” axis represents the red band while the “y” axis represents the green band.
- b. In all histograms, the drawn frequency represents the blue band; this combination was set in an attempt to discuss all three bands.
- c. In all images used, pixel values in all bands range between 0-255; this means that our original TIF image is an 8-bit image. This is consistent with the fact that the image is a “DOQQ” aerial photograph for the Huntington area; DOQQ’s are 8-bit images.

Statistics

How to calculate them:

As stated before, digital images are electronic bitmaps and bit layers where numerical values represent the information. Studying such numerical data from a statistical point of view where the max, min, mean, median, standard deviation,etc. are considered is useful and is one of the basis of many studies and applications.

As stated earlier in this chapter, since such numerical values are the image itself, altering those values changes the image. Sometimes, the differences between images can not be easily noticed by the human eye or many other regular techniques. Statistical calculations

give an indication about the image and its numerical values; they give an overall idea

about all the available information and the values without the need to study every single

pixel separately. They give an idea about the average pixel value (the mean), how the

data is distributed in the histogram and the median, and how far the data points are from

the mean (standard deviation).

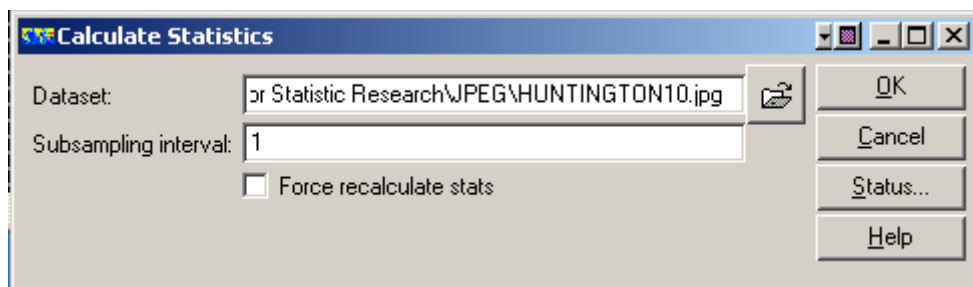


Figure 4-9 Starting the process of statistics calculations

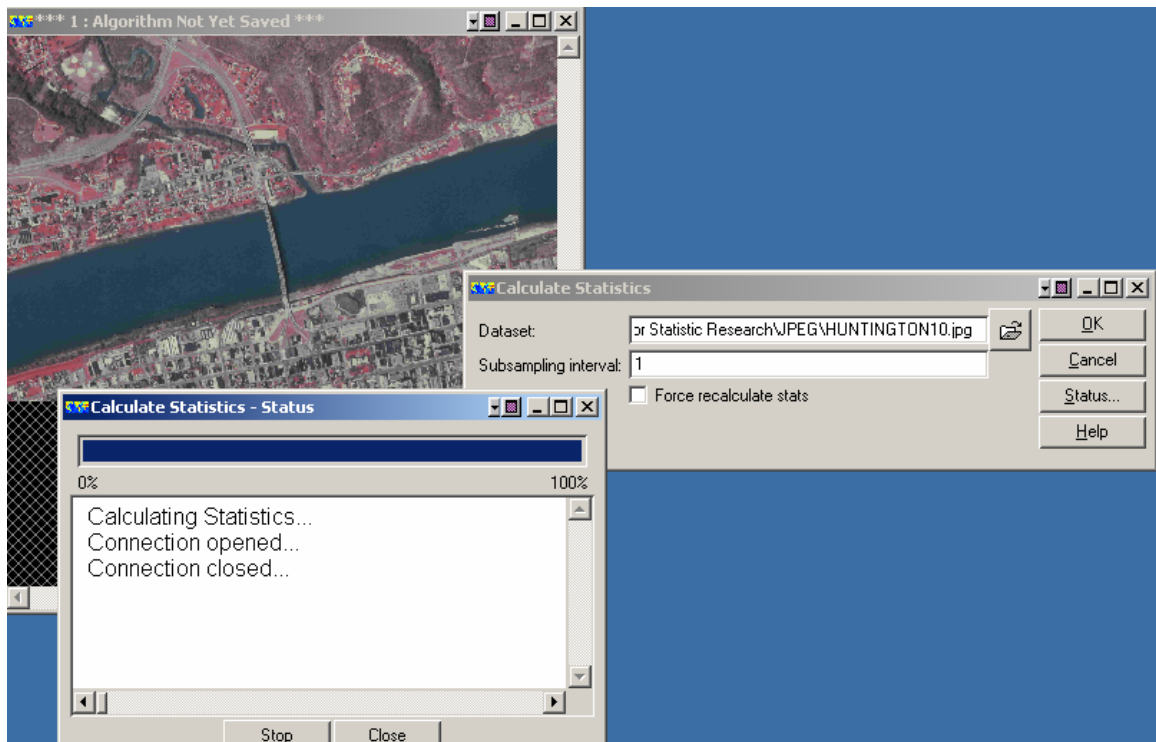


Figure 4-10 Statistics calculation process

Fig. 4.10 shows the progress of the statistics calculation process. As stated above, **if the results were different between images** (the mean, median, or the std. deviation for ex.), **the pixel values show a different pattern**. Because the digital image is nothing more than numerical pixel values in a 2-D plan, **getting different statistic results** between any two studied images **means that they are different** even if the eye sees them identical; numbers are more accurate.

See the following results for TIFF.

```

STATISTICS FOR DATASET: HUNTINGTON2.tif (using HUNTINGTON2.tif
classes/regions)
REGION: All
          Band1      Band2      Band3
          -----      -----      -----
Non-Null Cells      5324160      5324160      5324160
Area In Hectares    532.416      532.416      532.416
Area In Acres       1315.629      1315.629      1315.629

Minimum             13.000        15.000        24.000
Maximum             230.000       233.000       203.000
Mean                104.597       95.415        104.516
Median              110.000       82.000        93.000
Std. Dev.           52.012        38.195        28.382
Std. Dev. (n-1)    52.012        38.195        28.382
Corr. Eigenval.    2.672         0.306         0.022
Cov. Eigenval.    4403.295      542.328       24.043

Correlation Matrix
-----
Band1              1.000         0.799         0.731
Band2              0.799         1.000         0.972
Band3              0.731         0.972         1.000
Determinant                          0.018

Corr. Eigenvectors
-----
Band1              0.543         -0.830        0.127

```

Image Compression

Khaled S. Alkharabsheh

Band2	0.601	0.279	-0.749
Band3	0.587	0.483	0.650
Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.543	0.601	0.587
Band2	-0.830	0.279	0.483
Band3	0.127	-0.749	0.650
Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2705.220	1587.705	1078.732
Band2	1587.705	1458.890	1053.391
Band3	1078.732	1053.391	805.556
Determinant	57415576.243		
Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.749	-0.659	0.072
Band2	0.541	0.545	-0.641
Band3	0.383	0.519	0.764
Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.749	0.541	0.383
Band2	-0.659	0.545	0.519
Band3	0.072	-0.641	0.764

Table 4.1 statistical calculations for TIFF

2. 95% JPEG

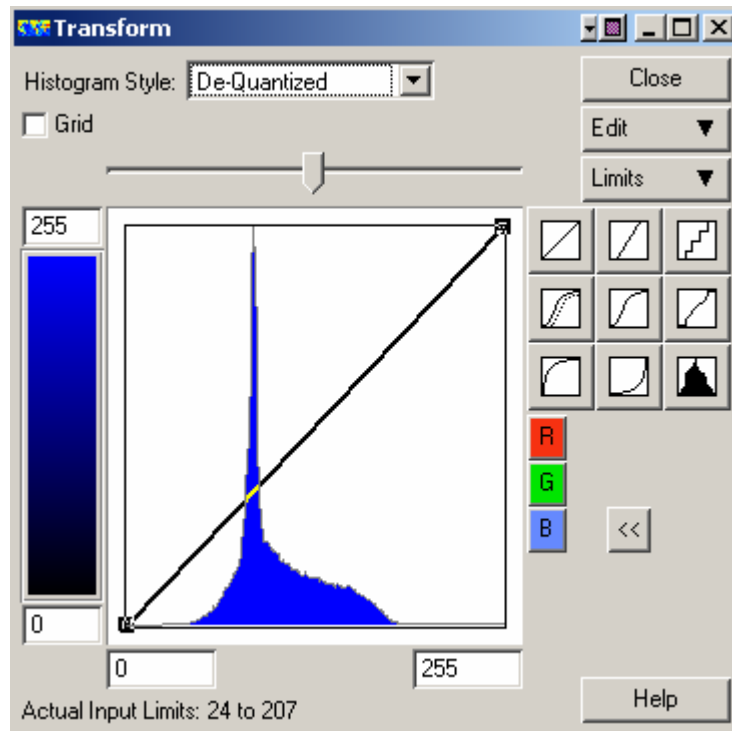


Figure 4-11 Histogram for 95% JPEG

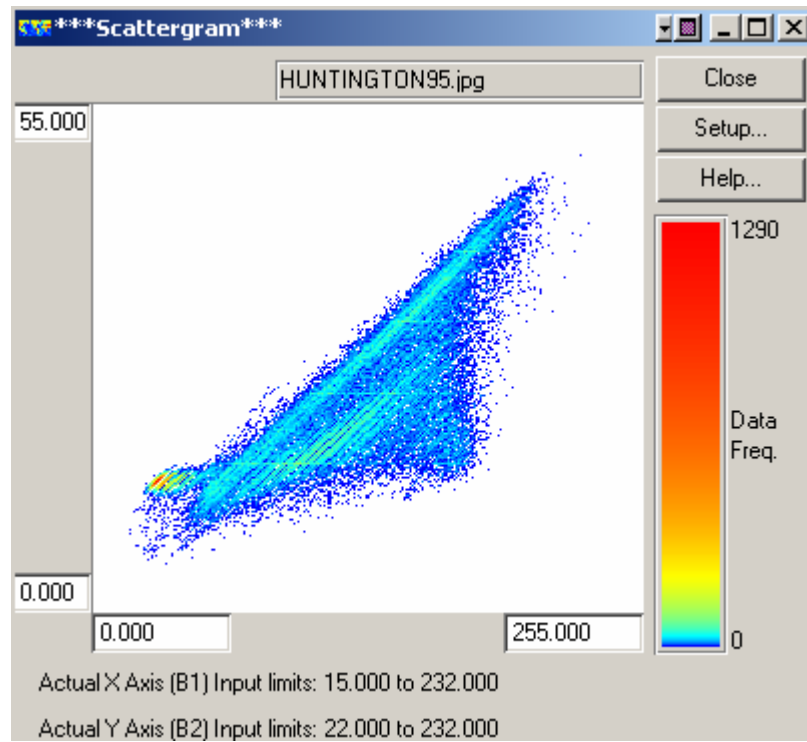


Figure 4-12 Scattergram for 95% JPEG

Statistics

STATISTICS FOR DATASET: HUNTINGTON95.jpg (using HUNTINGTON95.jpg classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416
Area In Acres	1315.629	1315.629	1315.629
Minimum	8.000	17.000	22.000
Maximum	246.000	232.000	222.000
Mean	102.496	97.315	102.416
Median	109.000	86.000	92.000
Std. Dev.	51.552	37.643	28.222

Image Compression

Khaled S. Alkharabsheh

Std. Dev. (n-1)	51.552	37.643	28.222
Corr. Eigenval.	2.690	0.290	0.020
Cov. Eigenval.	4346.222	503.053	21.819

Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.811	0.744
Band2	0.811	1.000	0.973
Band3	0.744	0.973	1.000
Determinant		0.016	

Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.546	-0.828	0.129
Band2	0.599	0.278	-0.751
Band3	0.586	0.487	0.648

Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.546	0.599	0.586
Band2	-0.828	0.278	0.487
Band3	0.129	-0.751	0.648

Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2657.613	1573.677	1083.065
Band2	1573.677	1416.995	1034.174
Band3	1083.065	1034.174	796.485
Determinant		47704122.754	

Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.749	-0.658	0.074
Band2	0.539	0.541	-0.646
Band3	0.386	0.524	0.760

Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.749	0.539	0.386
Band2	-0.658	0.541	0.524
Band3	0.074	-0.646	0.760

Table 4.2 Statistical calculations for Huntington 95%

3. 85% JPEG

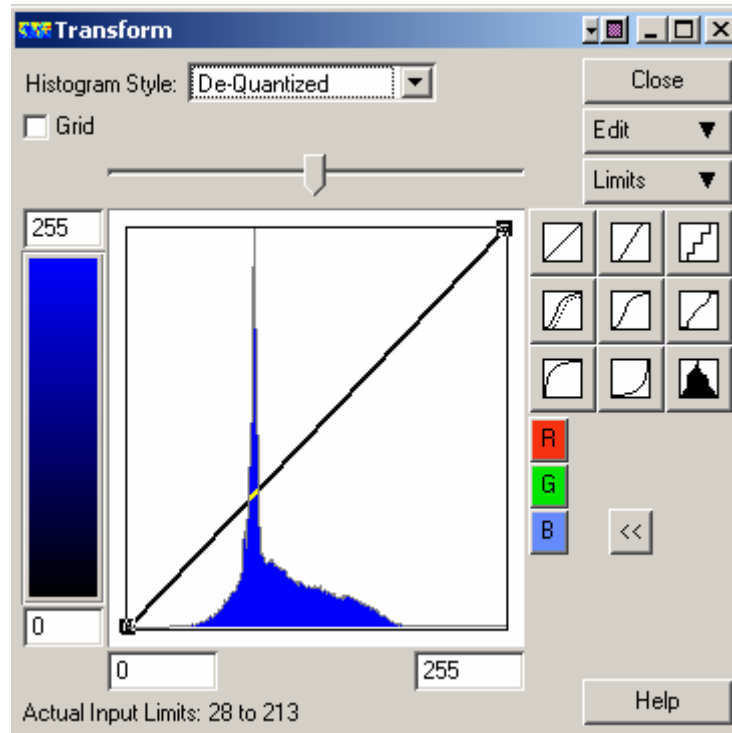


Figure 4-13 Histogram for 85% JPEG

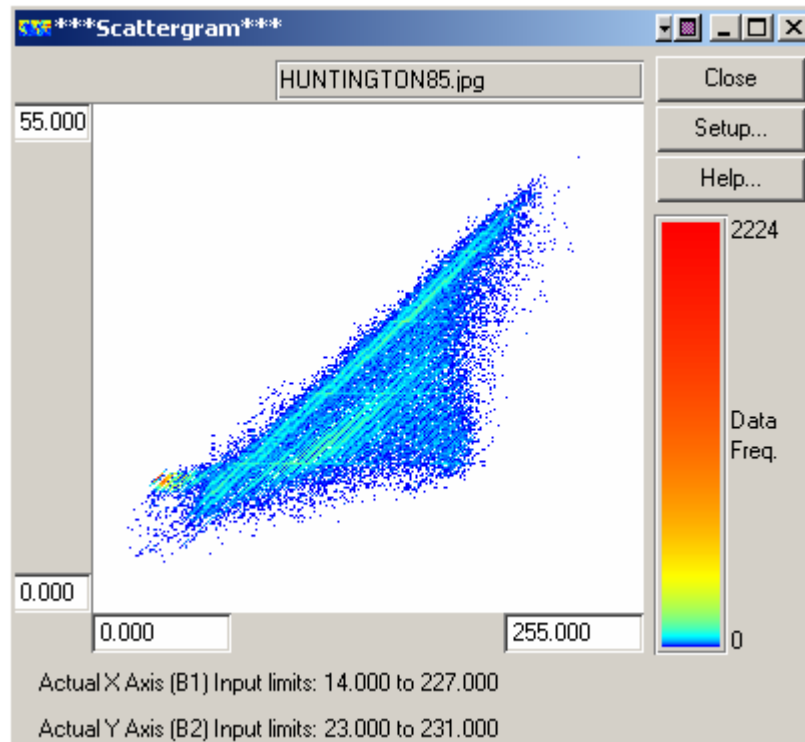


Figure 4-14 Scattergram for 85% JPEG

Statistics

STATISTICS FOR DATASET: HUNTINGTON85.jpg (using HUNTINGTON85.jpg classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416
Area In Acres	1315.629	1315.629	1315.629
Minimum	10.000	22.000	25.000
Maximum	253.000	231.000	223.000
Mean	102.564	97.683	102.556

Image Compression

Khaled S. Alkharabsheh

Median	110.000	86.000	92.000
Std. Dev.	50.645	36.876	28.193
Std. Dev. (n-1)	50.645	36.876	28.193
Corr. Eigenval.	2.707	0.273	0.020
Cov. Eigenval.	4233.971	464.898	20.696

Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.817	0.762
Band2	0.817	1.000	0.976
Band3	0.762	0.976	1.000
Determinant		0.015	

Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.548	-0.829	0.112
Band2	0.597	0.294	-0.746
Band3	0.586	0.476	0.656

Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.548	0.597	0.586
Band2	-0.829	0.294	0.476
Band3	0.112	-0.746	0.656

Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2564.883	1525.730	1088.464
Band2	1525.730	1359.815	1014.476
Band3	1088.464	1014.476	794.867
Determinant		40737651.513	

Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.747	-0.662	0.063
Band2	0.536	0.542	-0.648
Band3	0.394	0.518	0.759

Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.747	0.536	0.394
Band2	-0.662	0.542	0.518
Band3	0.063	-0.648	0.759

Table 4.3 Statistical calculations for Huntington 85%

4. 75% JPEG

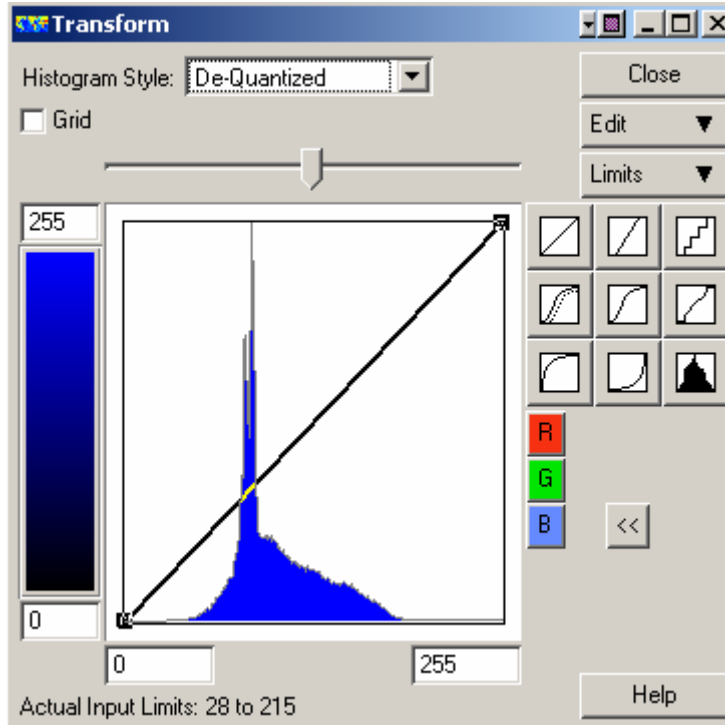


Figure 4-15 Histogram for 75% JPEG

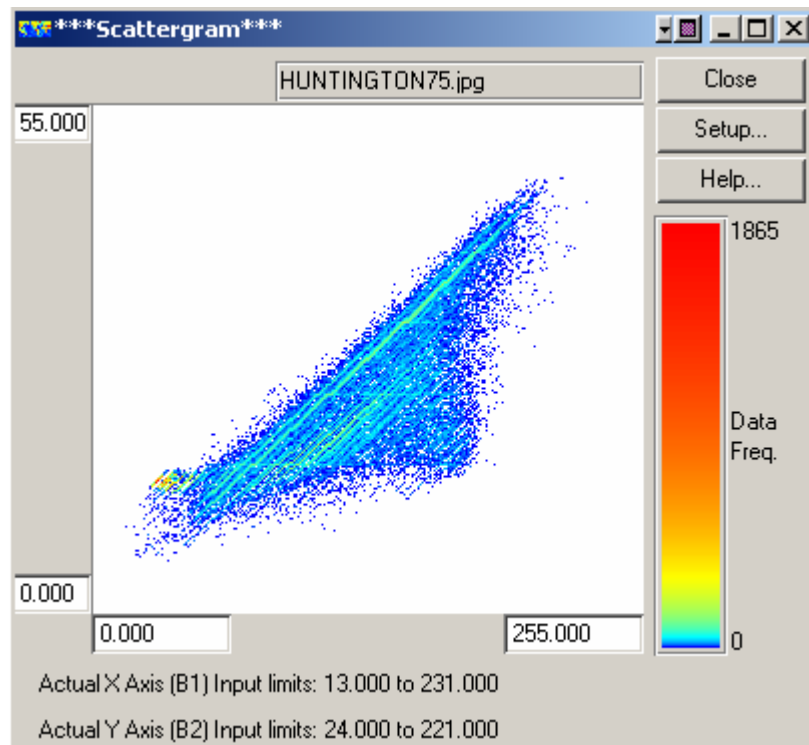


Figure 4-16 Scattergram for 75% JPEG

Statistics

STATISTICS FOR DATASET: HUNTINGTON75.jpg (using HUNTINGTON75.jpg classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416
Area In Acres	1315.629	1315.629	1315.629
Minimum	9.000	19.000	19.000
Maximum	249.000	234.000	226.000
Mean	102.672	97.817	102.668

Image Compression

Khaled S. Alkharabsheh

Median	109.000	86.000	93.000
Std. Dev.	50.278	36.566	28.418
Std. Dev. (n-1)	50.278	36.566	28.418
Corr. Eigenval.	2.723	0.257	0.020
Cov. Eigenval.	4216.862	434.943	20.746
Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.826	0.778
Band2	0.826	1.000	0.977
Band3	0.778	0.977	1.000
Determinant		0.014	
Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.550	-0.829	0.104
Band2	0.595	0.302	-0.745
Band3	0.586	0.471	0.659
Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.550	0.595	0.586
Band2	-0.829	0.302	0.471
Band3	0.104	-0.745	0.659
Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2527.911	1518.204	1111.741
Band2	1518.204	1337.046	1014.759
Band3	1111.741	1014.759	807.594
Determinant		38050172.469	
Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.744	-0.665	0.059
Band2	0.534	0.540	-0.651
Band3	0.402	0.516	0.757
Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.744	0.534	0.402
Band2	-0.665	0.540	0.516
Band3	0.059	-0.651	0.757

Table 4.4 Statistical calculations for Huntington 75%

5. 65% JPEG

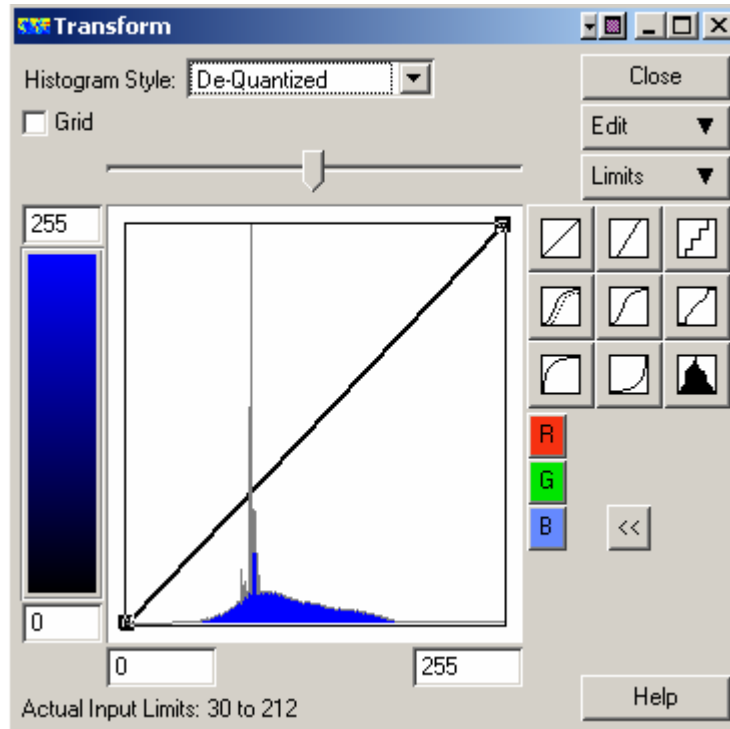


Figure 4-17 Histogram for 65% JPEG

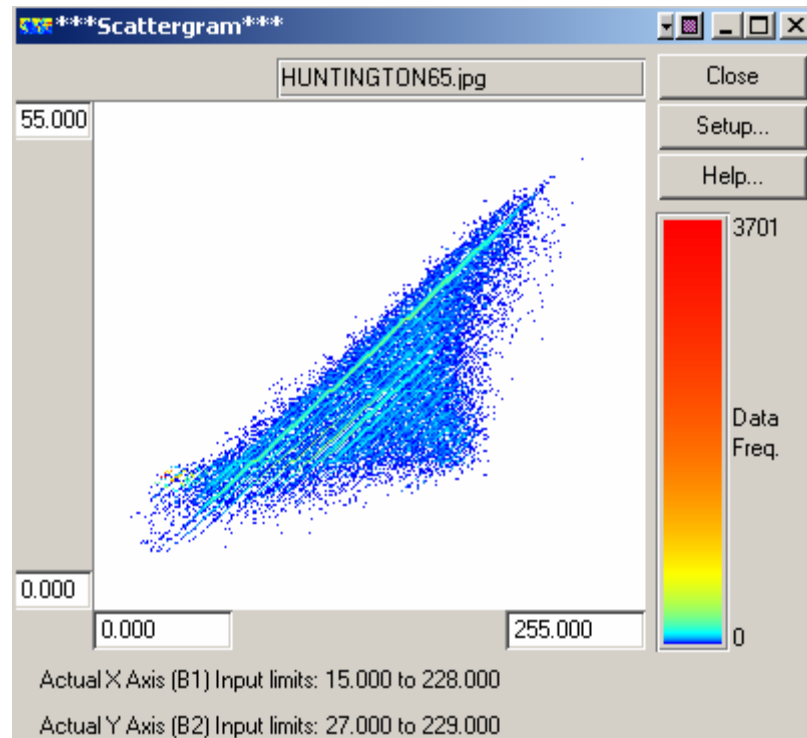


Figure 4-18 Scattergram for 65% JPEG

Statistics

STATISTICS FOR DATASET: HUNTINGTON65.jpg (using HUNTINGTON65.jpg classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416
Area In Acres	1315.629	1315.629	1315.629
Minimum	8.000	21.000	21.000
Maximum	247.000	229.000	228.000
Mean	103.159	97.883	103.571
Median	111.000	88.000	94.000
Std. Dev.	49.504	35.869	28.308
Std. Dev. (n-1)	49.504	35.869	28.308

Image Compression

Khaled S. Alkharabsheh

Corr. Eigenval.	2.729	0.249	0.021
Cov. Eigenval.	4103.622	413.582	21.390

Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.827	0.787
Band2	0.827	1.000	0.976
Band3	0.787	0.976	1.000
Determinant		0.014	

Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.551	-0.830	0.089
Band2	0.594	0.315	-0.740
Band3	0.586	0.460	0.667

Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.551	0.594	0.586
Band2	-0.830	0.315	0.460
Band3	0.089	-0.740	0.667

Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2450.633	1467.707	1103.135
Band2	1467.707	1286.602	991.453
Band3	1103.135	991.453	801.358
Determinant		36302188.627	

Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.743	-0.667	0.049
Band2	0.531	0.543	-0.651
Band3	0.408	0.510	0.758

Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.743	0.531	0.408
Band2	-0.667	0.543	0.510
Band3	0.049	-0.651	0.758

Table 4.5 Statistical calculations for Huntington 65%

6. 50% JPEG

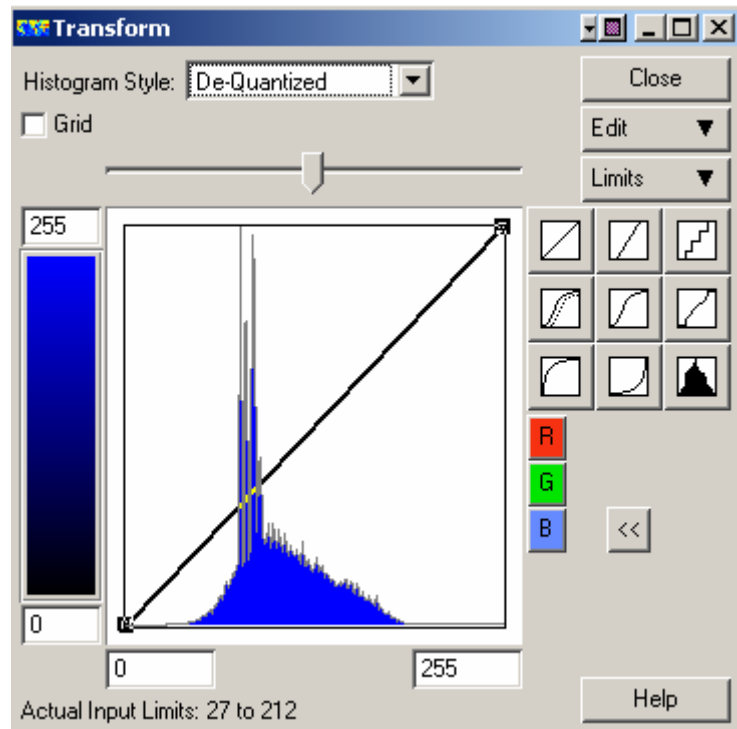


Figure 4-19 Histogram for 50% JPEG

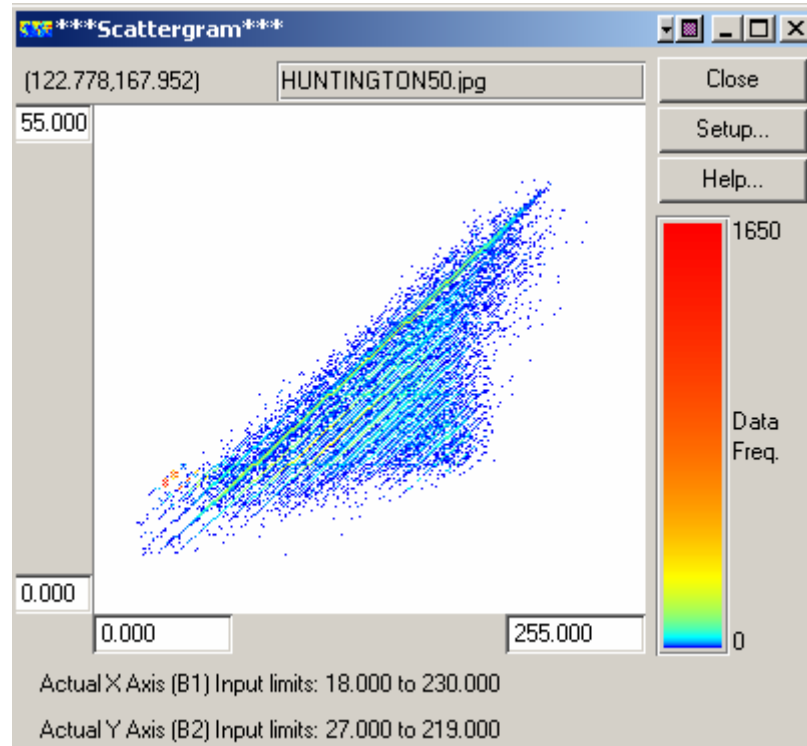


Figure 4-20 Scattergram for 50% JPEG

Statistics

STATISTICS FOR DATASET: HUNTINGTON50.jpg (using HUNTINGTON50.jpg classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416
Area In Acres	1315.629	1315.629	1315.629
Minimum	11.000	21.000	21.000
Maximum	250.000	235.000	228.000
Mean	103.073	98.307	103.332
Median	110.000	89.000	95.000
Std. Dev.	48.640	35.472	28.695

Image Compression

Khaled S. Alkharabsheh

Std. Dev. (n-1)	48.640	35.472	28.695
Corr. Eigenval.	2.752	0.225	0.024
Cov. Eigenval.	4054.787	368.800	23.875
Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.839	0.811
Band2	0.839	1.000	0.975
Band3	0.811	0.975	1.000
Determinant		0.015	
Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.554	-0.829	0.070
Band2	0.591	0.333	-0.734
Band3	0.586	0.449	0.675
Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.554	0.591	0.586
Band2	-0.829	0.333	0.449
Band3	0.070	-0.734	0.675
Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2365.827	1447.086	1131.712
Band2	1447.086	1258.237	992.456
Band3	1131.712	992.456	823.398
Determinant		35702864.578	
Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.736	-0.676	0.037
Band2	0.530	0.542	-0.652
Band3	0.421	0.500	0.757
Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.736	0.530	0.421
Band2	-0.676	0.542	0.500
Band3	0.037	-0.652	0.757

Table 4.6 Statistical calculations for Huntington 50%

7. 30% JPEG

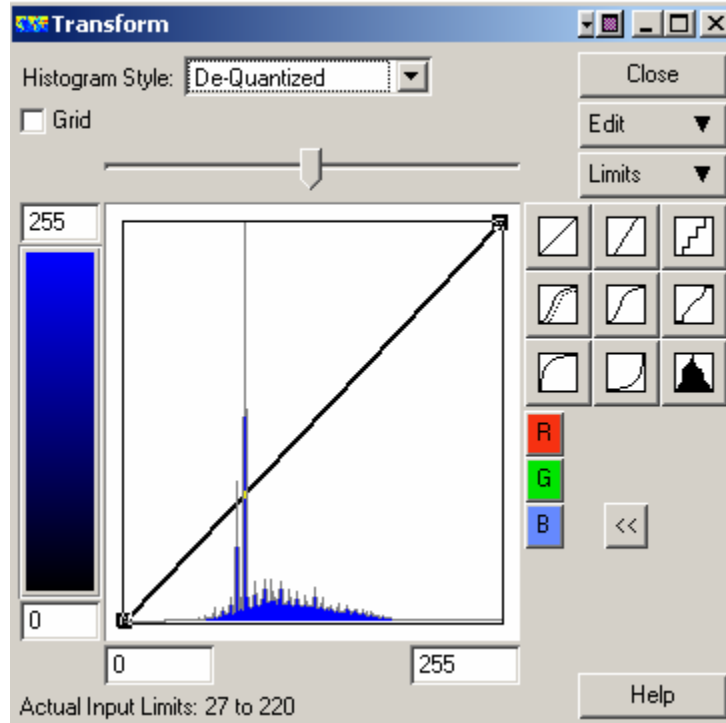


Figure 4-21 Histogram for 30% JPEG

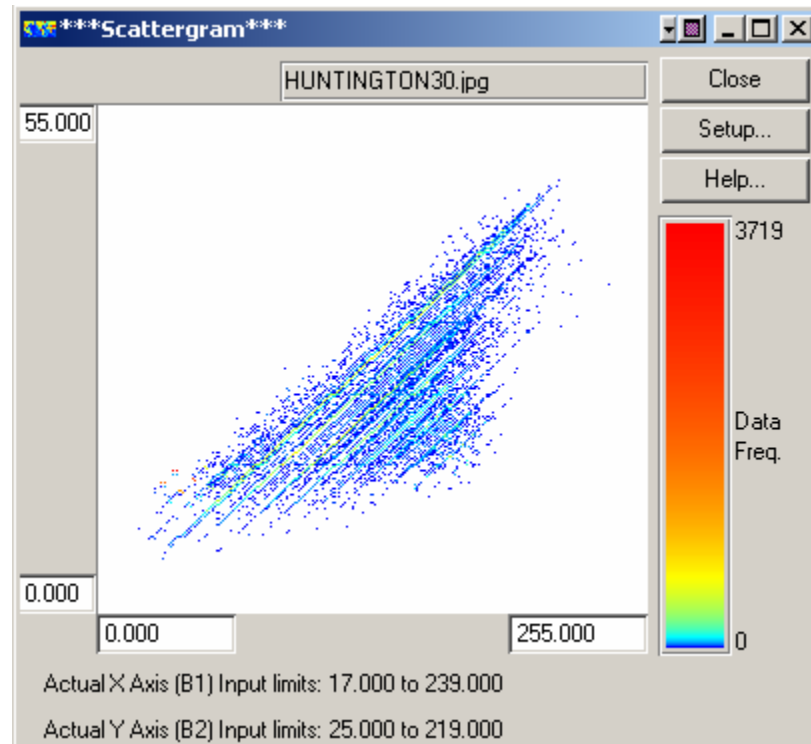


Figure 4-22 Scattergram for 30% JPEG

Statistics

STATISTICS FOR DATASET: HUNTINGTON30.jpg (using HUNTINGTON30.jpg classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416
Area In Acres	1315.629	1315.629	1315.629
Minimum	14.000	13.000	25.000
Maximum	249.000	233.000	228.000
Mean	103.389	98.947	104.046
Median	110.000	89.000	96.000
Std. Dev.	47.509	34.217	29.094
Std. Dev. (n-1)	47.509	34.217	29.094

Corr. Eigenval.	2.770	0.209	0.021
Cov. Eigenval.	3915.574	338.313	20.477
Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.833	0.840
Band2	0.833	1.000	0.979
Band3	0.840	0.979	1.000
Determinant		0.012	
Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.556	-0.831	-0.018
Band2	0.587	0.408	-0.699
Band3	0.589	0.378	0.715
Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.556	0.587	0.589
Band2	-0.831	0.408	0.378
Band3	-0.018	-0.699	0.715
Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2257.080	1354.514	1160.981
Band2	1354.514	1170.812	974.602
Band3	1160.981	974.602	846.471
Determinant		27125021.312	
Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.732	-0.681	-0.016
Band2	0.518	0.573	-0.635
Band3	0.442	0.457	0.772
Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.732	0.518	0.442
Band2	-0.681	0.573	0.457
Band3	-0.016	-0.635	0.772

Table 4.7 Statistical calculations for Huntington 30%

8. 20% JPEG

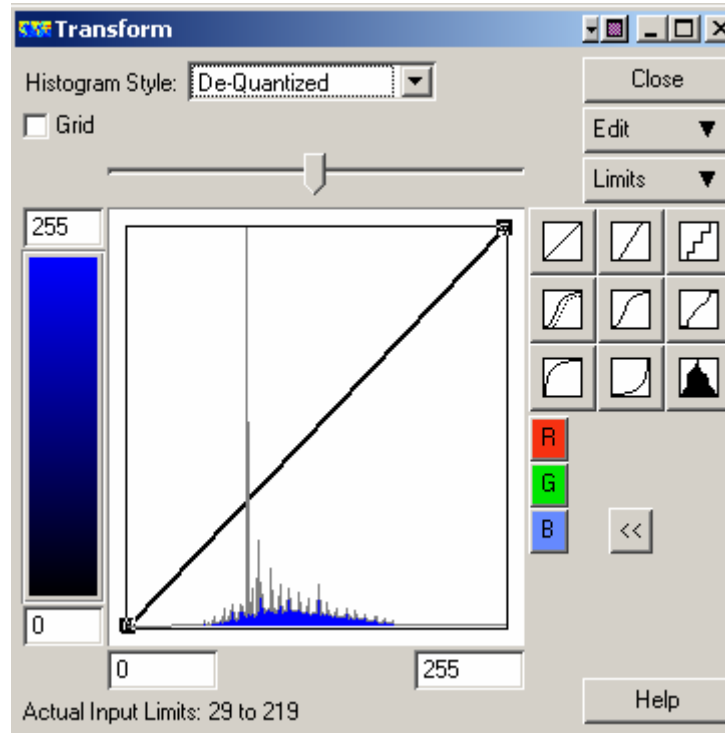


Figure 4-23 Histogram for 20% JPEG

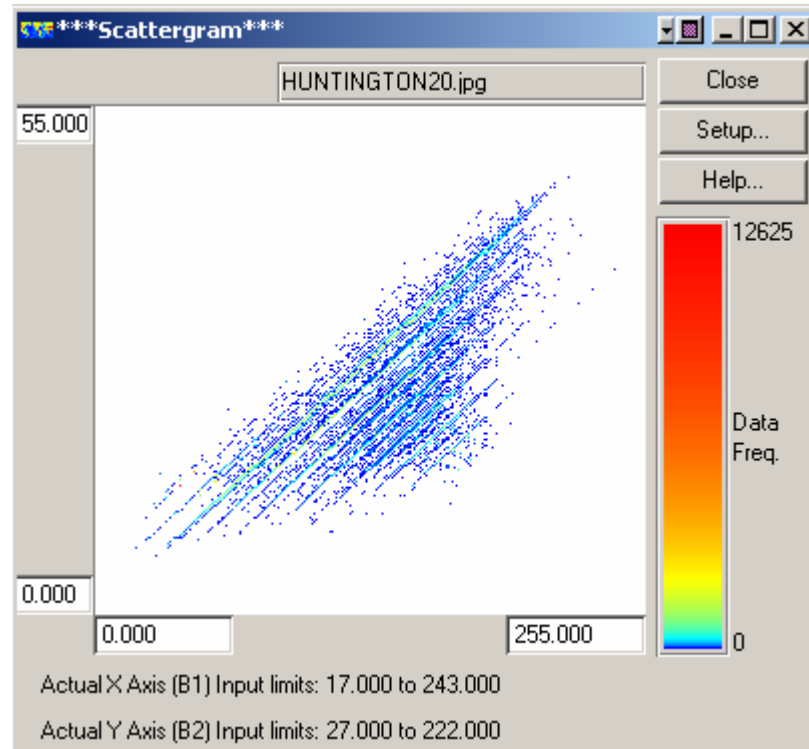


Figure 4-24 Scattergram for 20% JPEG

Statistics

Display Dataset Statistics			
Minimum	16.000	22.000	23.000
Maximum	250.000	232.000	231.000
Mean	103.322	98.826	104.588
Median	109.000	92.000	97.000
Std. Dev.	46.130	34.575	29.022
Std. Dev. (n-1)	46.130	34.575	29.022
Corr. Eigenval.	2.790	0.186	0.025
Cov. Eigenval.	3851.585	289.299	24.793
Correlation Matrix			
	Band1	Band2	Band3
Band1	1.000	0.860	0.848
Band2	0.860	1.000	0.975
Band3	0.848	0.975	1.000
Determinant	0.013		
Corr. Eigenvectors			
	PC1	PC2	PC3
Band1	0.559	-0.828	0.039
Band2	0.587	0.363	-0.723
Band3	0.585	0.428	0.689
Inv. of Corr. Ev.			
	PC1	PC2	PC3
Band1	0.559	0.587	0.585
Band2	-0.828	0.363	0.428
Band3	0.039	-0.723	0.689
Covariance Matrix			
	Band1	Band2	Band3
Band1	2127.982	1371.823	1134.712
Band2	1371.823	1195.425	978.087
Band3	1134.712	978.087	842.269
Determinant	27625451.707		
Cov. Eigenvectors			
	PC1	PC2	PC3
Band1	0.718	-0.695	0.018
Band2	0.535	0.536	-0.653
Band3	0.445	0.479	0.757
Inv. of Cov. Ev.			
	PC1	PC2	PC3

Table 4.8

STATISTICS FOR DATASET: HUNTINGTON20.jpg (using HUNTINGTON20.jpg classes/regions)

REGION: All

	Band1	Band2	Band3
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416

Image Compression

Khaled S. Alkharabsheh

Area In Acres	1315.629	1315.629	1315.629
Minimum	16.000	22.000	23.000
Maximum	250.000	232.000	231.000
Mean	103.322	98.826	104.588
Median	109.000	92.000	97.000
Std. Dev.	46.130	34.575	29.022
Std. Dev. (n-1)	46.130	34.575	29.022
Corr. Eigenval.	2.790	0.186	0.025
Cov. Eigenval.	3851.585	289.299	24.793

Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.860	0.848
Band2	0.860	1.000	0.975
Band3	0.848	0.975	1.000

Determinant 0.013

Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.559	-0.828	0.039
Band2	0.587	0.363	-0.723
Band3	0.585	0.428	0.689

Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.559	0.587	0.585
Band2	-0.828	0.363	0.428
Band3	0.039	-0.723	0.689

Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2127.982	1371.823	1134.712
Band2	1371.823	1195.425	978.087
Band3	1134.712	978.087	842.269

Determinant 27625451.707

Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.718	-0.695	0.018
Band2	0.535	0.536	-0.653
Band3	0.445	0.479	0.757

Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.718	0.535	0.445
Band2	-0.695	0.536	0.479
Band3	0.018	-0.653	0.757

Table 4.9 Statistical calculations for Huntington 20%

9. 10% JPEG



Figure 4-25 10% JPEG

File size= 185KB

See efficiency comparison with ECW99 (page 148).

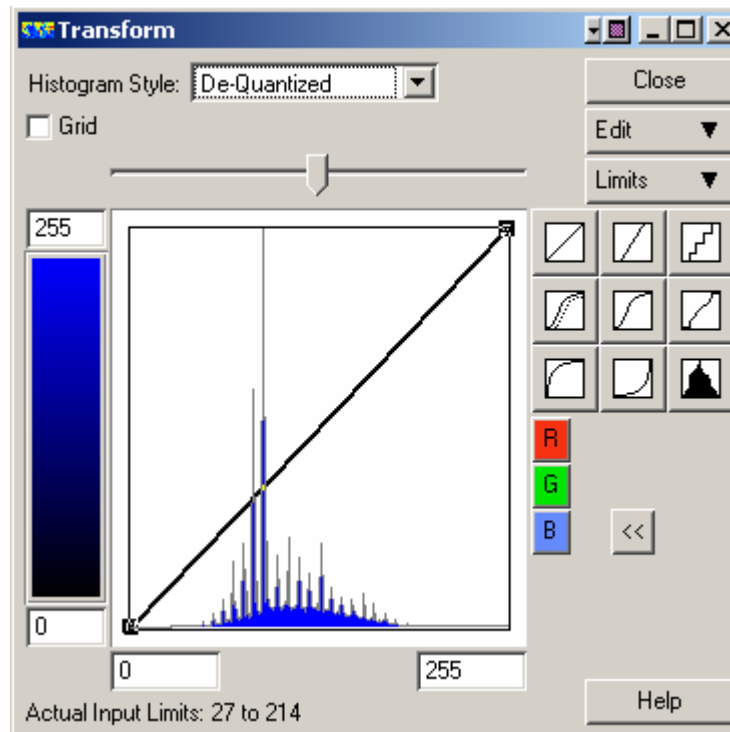


Figure 4-26 Histogram for 10% JPEG

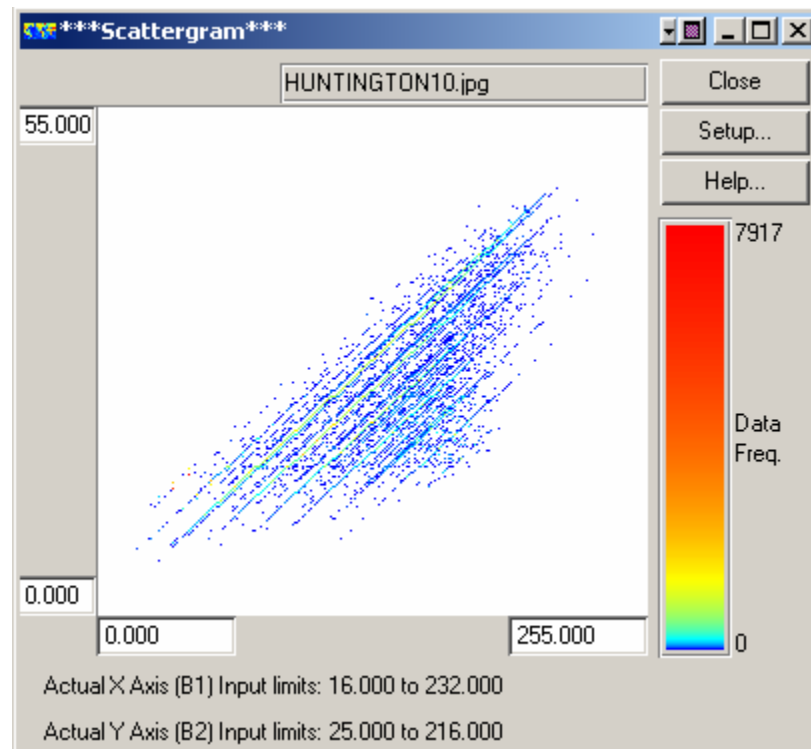


Figure 4-27 Scattergram for % JPEG

Statistics

STATISTICS FOR DATASET: HUNTINGTON10.jpg (using HUNTINGTON10.jpg classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416
Area In Acres	1315.629	1315.629	1315.629
Minimum	15.000	21.000	17.000
Maximum	249.000	233.000	232.000
Mean	103.430	99.437	105.603
Median	109.000	91.000	98.000
Std. Dev.	45.965	33.819	28.435
Std. Dev. (n-1)	45.965	33.819	28.435

Image Compression

Khaled S. Alkharabsheh

Corr. Eigenval.	2.764	0.204	0.032
Cov. Eigenval.	3726.901	307.273	30.877

Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.852	0.825
Band2	0.852	1.000	0.966
Band3	0.825	0.966	1.000
Determinant		0.018	

Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.558	-0.826	0.077
Band2	0.589	0.329	-0.738
Band3	0.584	0.457	0.671

Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.558	0.589	0.584
Band2	-0.826	0.329	0.457
Band3	0.077	-0.738	0.671

Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2112.755	1323.798	1078.641
Band2	1323.798	1143.723	929.440
Band3	1078.641	929.440	808.573
Determinant		35359738.414	

Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.727	-0.686	0.040
Band2	0.530	0.523	-0.668
Band3	0.437	0.506	0.743

Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.727	0.530	0.437
Band2	-0.686	0.523	0.506
Band3	0.040	-0.668	0.743

Table 4.10 Statistical calculations for Huntington 10%

Display Dataset Statistics

STATISTICS FOR DATASET: HUNTINGTON10.jpg (using HUNTINGTON10.jpg classes/regions)
REGION: All

	Band1	Band2	Band3
Non-Null Cells	5324160	5324160	5324160
Area In Hectares	532.416	532.416	532.416
Area In Acres	1315.629	1315.629	1315.629
Minimum	15.000	21.000	17.000
Maximum	249.000	233.000	232.000
Mean	103.430	99.437	105.603
Median	109.000	91.000	98.000
Std. Dev.	45.965	33.819	28.435
Std. Dev. (n-1)	45.965	33.819	28.435
Corr. Eigenval.	2.764	0.204	0.032
Cov. Eigenval.	3726.901	307.273	30.877
Correlation Matrix	Band1	Band2	Band3
Band1	1.000	0.852	0.825
Band2	0.852	1.000	0.966
Band3	0.825	0.966	1.000
Determinant	0.018		
Corr. Eigenvectors	PC1	PC2	PC3
Band1	0.558	-0.826	0.077
Band2	0.589	0.329	-0.738
Band3	0.584	0.457	0.671
Inv. of Corr. Ev.	PC1	PC2	PC3
Band1	0.558	0.589	0.584
Band2	-0.826	0.329	0.457
Band3	0.077	-0.738	0.671
Covariance Matrix	Band1	Band2	Band3
Band1	2112.755	1323.798	1078.641
Band2	1323.798	1143.723	929.440
Band3	1078.641	929.440	808.573
Determinant	35359738.414		

Table 4.11 Statistics window for Huntington10

How was the statistics changing with JPEG compression and how was it fluctuating? This is what will be answered in the following comparisons.

When we compare images, we should compare them from one common point of view or statistical output that belongs to one common band. In this part of the study, I did choose to study the fluctuation in the standard deviation between some JPEG images compressed using PaperPort.

Image	Std. Div. (Band 2)
Tiff	38.195
JPEG 95%	37.643
JPEG 85%	36.876
JPEG 75%	36.566
JPEG 65%	35.869
JPEG 50%	35.472
JPEG 30%	34.217
JPEG 20%	34.575
JPEG 10%	33.819

Table 4.12 Standard Deviation for JPEG images

Std. Dev. Variation for JPEG

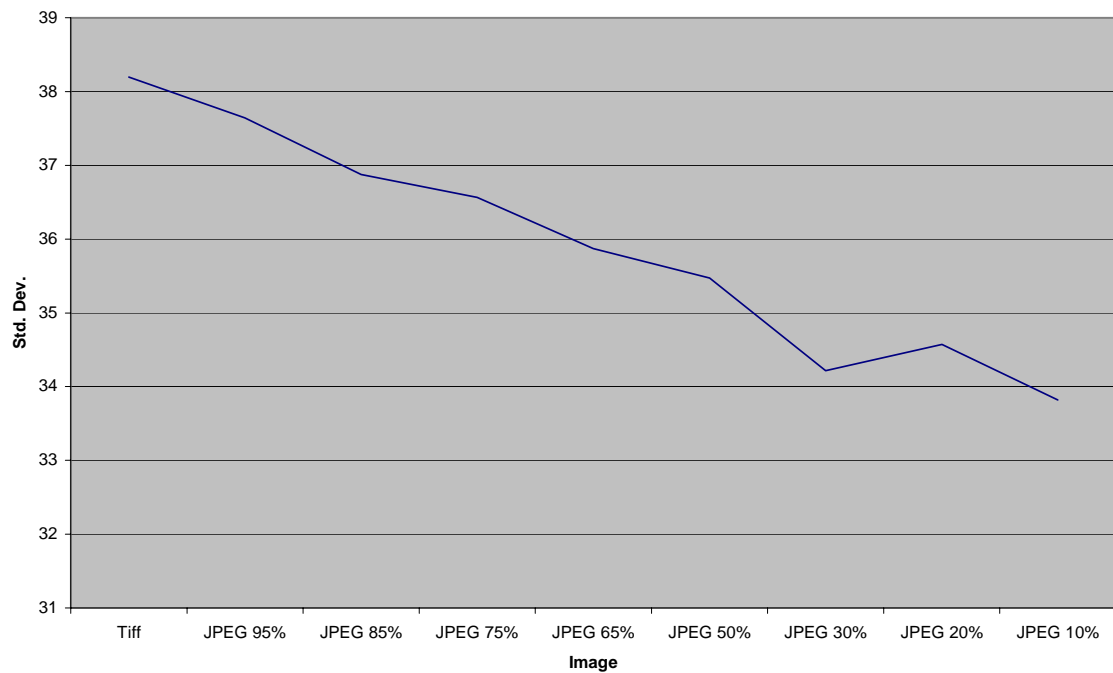


Figure 4-28 Std. Dev. Variations for JPEG imagery

It can be clearly noticed (due to **more lost** data samples as the image is further compressed, and to more spread that will be in the scattergram) that the more the image is

compressed, the less the number of data samples, less noticeable are the differences between the existing objects, less distinguishable the objects, and the less clear a noticeable area will be. (see scattergrams for TIF file (fig. 4.6) and 10% JPEG (fig. 4.27)). What does this mean?

The results noticed above are strongly related to the definition of JPEG compression where the compressing software uses many different techniques and standards to compress the image among which are: (obtained JPEG images do agree with what is going to be stated, see discussion below)

- a) Lessening the number of pixels in the image by eliminating some rows and columns; this matches with the decreasing number of data samples in scattergrams.
- b) Averaging specific numbers of pixels (8 in most cases) and assigning the result to only one pixel; this means that an area becomes less noticeable as the image is compressed further.

Discussion and proofs

Looking at the JPEG compressed images, we can see strong relations between them from one side and “a” and “b” in the discussion above (last paragraph) from the other; depending on the compression ratio. As the image is more compressed, the number of pixels decreases (see Huntington10 and its corresponding scattergram). If the physical dimension of the images were sustained (image was displayed at the same zoom), we can

see that the pixels in Huntington10 are larger than those in TIF; if larger pixels cover the same area on the screen, we can conclude that the number of pixels did decrease. This matches what was concluded and seen from Huntington10's scattergram (fig. 4.27).

If we ask ourselves how did the larger pixels in Huntington10 replace the smaller ones in TIF, we find that the compressing software did eliminate some pixels ("a" above) and assign the averages (the mean) of every pixel array of a specific size to a single pixel that replaces the array ("b" above).

Note here that as the objects in the image become less distinguishable in the scattergrams, the harder and more complicated any classification process of any kind will be.

10. 30 times compressed ECW

Note here the difference between the numbers used **to name** JPEG and ECW compressed images (ex. **30 or 99** times compressed ECW, Huntington**10**). **In the first case**, the image is **more compressed as the number goes down** since it represents some percentage while **in the second case** the image is **more compressed as the number goes up** since it represents the file size ratio between the original image and the compressed one (the theoretical compression ratio).

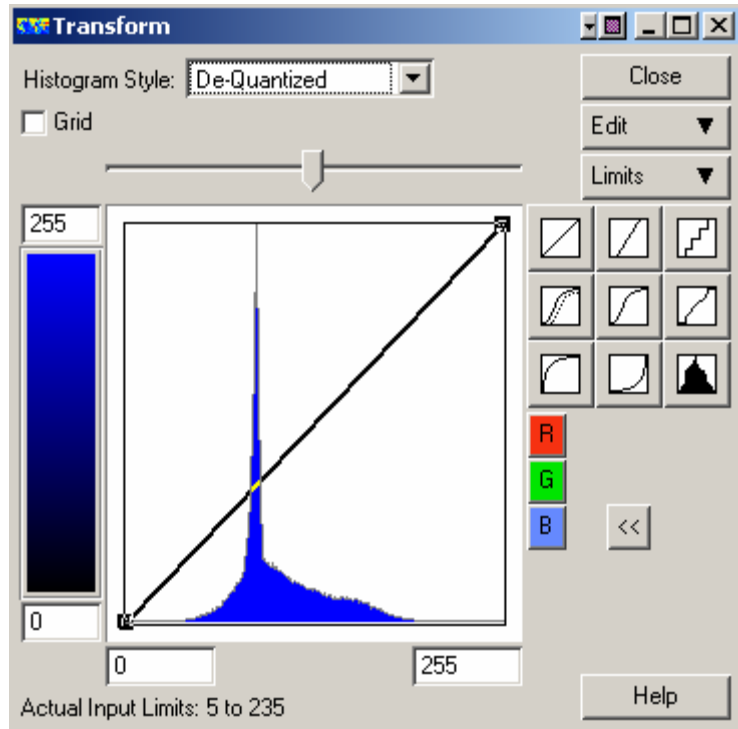


Figure 4-29 Histogram for 30 times compressed ECW

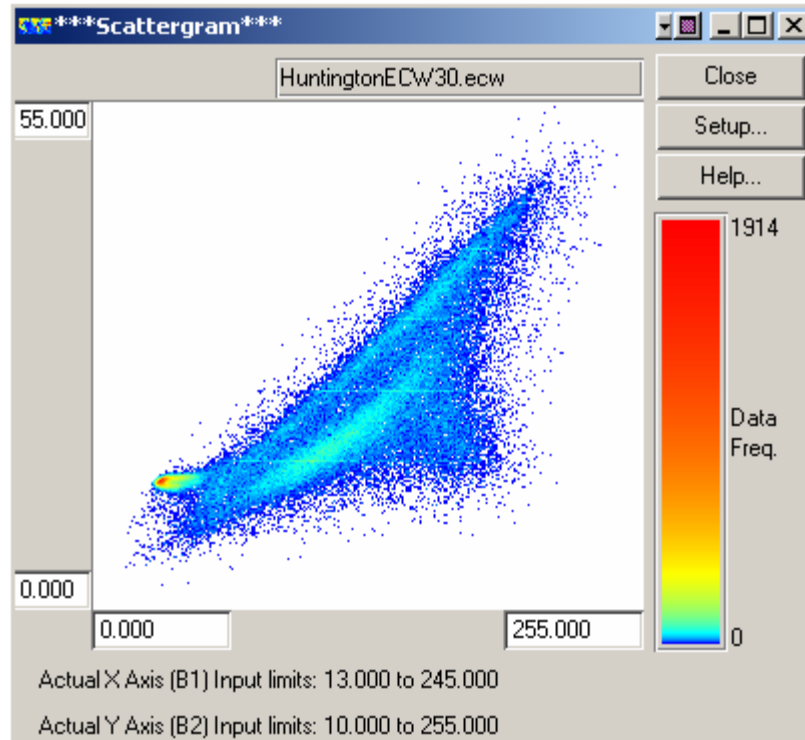


Figure 4-30 Scattergram for 30 times compressed ECW

Statistics

STATISTICS FOR DATASET: HuntingtonECW30.ecw (using HuntingtonECW30.ecw classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	4383892	4383892	4383892
Area In Hectares	438.389	438.389	438.389
Area In Acres	1083.283	1083.283	1083.283
Minimum	0.000	5.000	18.000
Maximum	255.000	246.000	243.000
Mean	101.823	92.681	102.859
Median	110.000	80.000	93.000
Std. Dev.	52.968	36.232	27.282
Std. Dev. (n-1)	52.968	36.232	27.282
Corr. Eigenval.	2.674	0.305	0.021
Cov. Eigenval.	4324.195	517.312	21.170

Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.803	0.730
Band2	0.803	1.000	0.972
Band3	0.730	0.972	1.000
Determinant		0.017	
Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.544	-0.828	0.136
Band2	0.601	0.271	-0.752
Band3	0.586	0.491	0.645
Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.544	0.601	0.586
Band2	-0.828	0.271	0.491
Band3	0.136	-0.752	0.645
Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2805.609	1541.020	1054.366
Band2	1541.020	1312.786	960.484
Band3	1054.366	960.484	744.282
Determinant		47356895.160	
Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.776	-0.627	0.073
Band2	0.514	0.560	-0.650
Band3	0.366	0.542	0.756
Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.776	0.514	0.366
Band2	-0.627	0.560	0.542
Band3	0.073	-0.650	0.756

Table 4.13 Statistical calculations for Huntington ECW30

11. ECW99

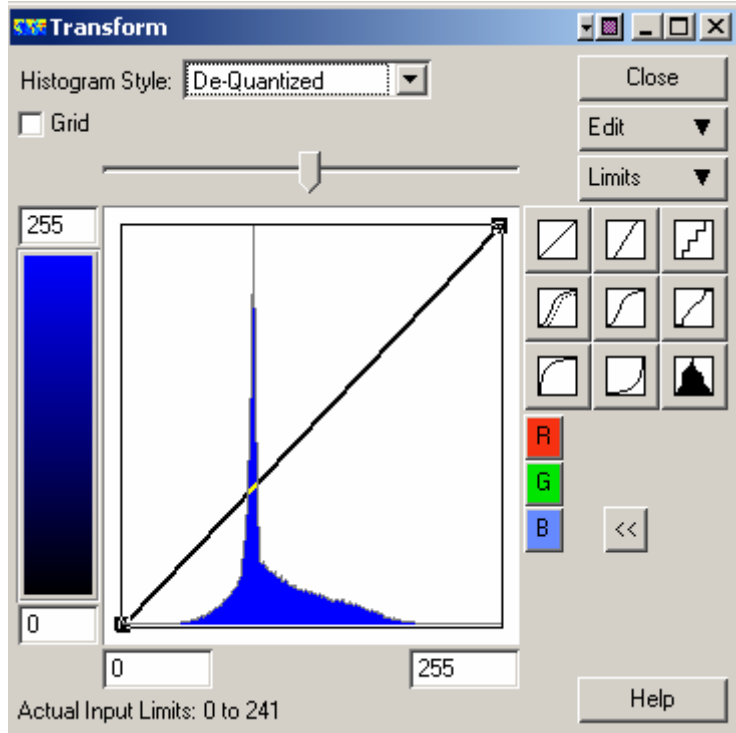


Figure 4-31 Histogram for 99 times compressed ECW

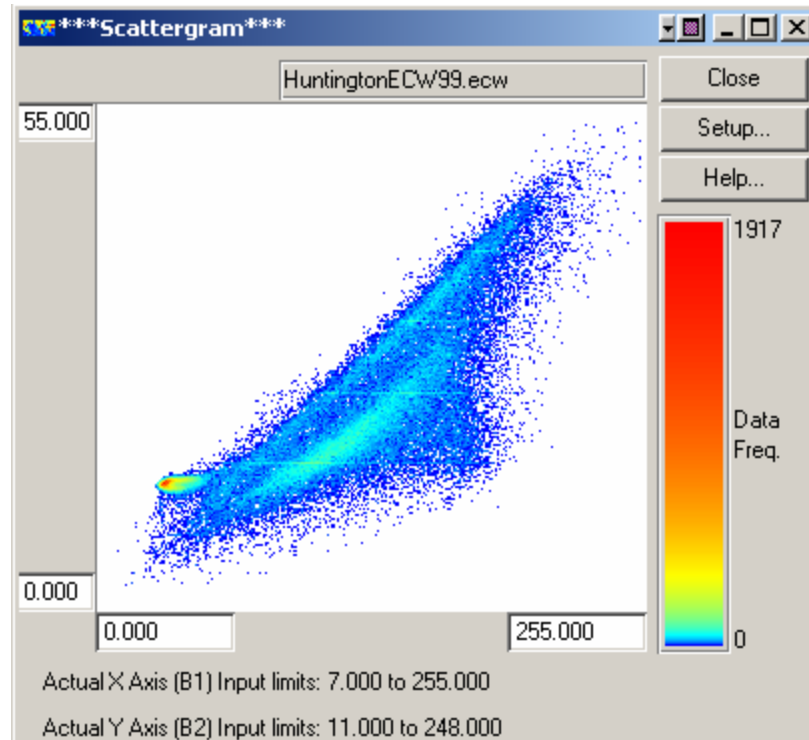


Figure 4-32 Scattergram for 99 times compressed ECW

Statistics

STATISTICS FOR DATASET: HuntingtonECW99.ecw (using HuntingtonECW99.ecw classes/regions)

REGION: All

	Band1	Band2	Band3
	-----	-----	-----
Non-Null Cells	4383892	4383892	4383892
Area In Hectares	438.389	438.389	438.389
Area In Acres	1083.283	1083.283	1083.283
Minimum	0.000	0.000	0.000
Maximum	255.000	255.000	255.000
Mean	101.821	92.684	102.860
Median	110.000	81.000	93.000
Std. Dev.	52.838	35.923	27.157
Std. Dev. (n-1)	52.838	35.923	27.157
Corr. Eigenval.	2.684	0.297	0.019
Cov. Eigenval.	4306.422	494.792	18.531

Correlation Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	1.000	0.813	0.735
Band2	0.813	1.000	0.973
Band3	0.735	0.973	1.000
Determinant		0.015	
Corr. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.545	-0.825	0.148
Band2	0.601	0.261	-0.756
Band3	0.585	0.501	0.638
Inv. of Corr. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.545	0.601	0.585
Band2	-0.825	0.261	0.501
Band3	0.148	-0.756	0.638
Covariance Matrix	Band1	Band2	Band3
-----	-----	-----	-----
Band1	2791.818	1543.204	1054.067
Band2	1543.204	1290.442	949.208
Band3	1054.067	949.208	737.486
Determinant		39486452.098	
Cov. Eigenvectors	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.777	-0.624	0.081
Band2	0.513	0.552	-0.657
Band3	0.366	0.552	0.749
Inv. of Cov. Ev.	PC1	PC2	PC3
-----	-----	-----	-----
Band1	0.777	0.513	0.366
Band2	-0.624	0.552	0.552
Band3	0.081	-0.657	0.749

Table 4.14 Statistical calculations for ECW99

Let's compare ECW99 and 10%JPEG

I. Quality



Figure 4-33 JPEG 10%

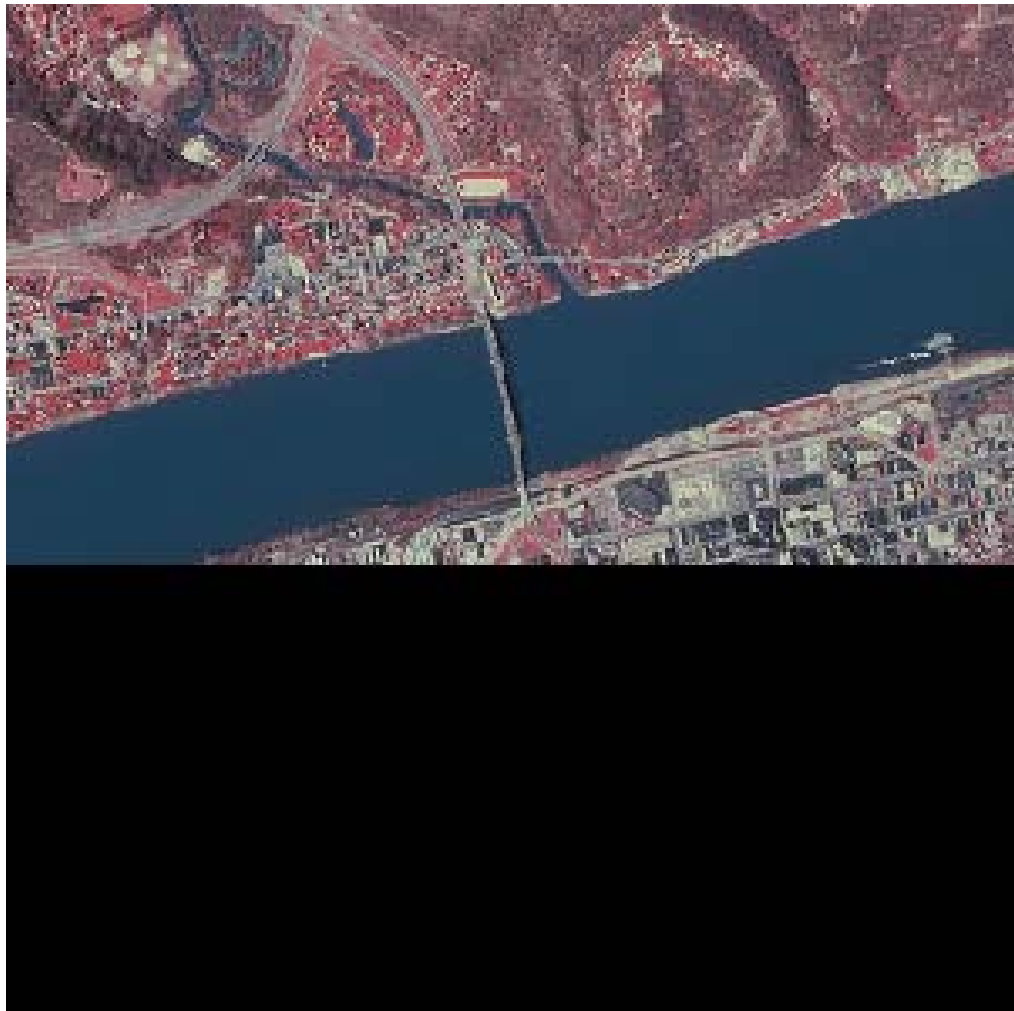


Figure 4-34 ECW99

Figs. 4.43 and 4.44 show the difference between the outputs of the two compressing programs. Compression is applied on images to reduce the file size while keeping the desired quality (according to the application and as needed). The more quality the compressing software gives for a specific file size, the better it is considered. Using the eye we can say that ECW gave much better compressed images; but, what about numbers and statistics? This question is answered in part II on the next page.

II. File size

Type: JPEG Image
Size: 185 KB

Type: ECW File
Size: 201 KB

JPEG saved a little more space (16KB). Now, we are going to discuss the quality each program gives and how much data it preserves during the process.

III. Less data disturbance and changes (quality)

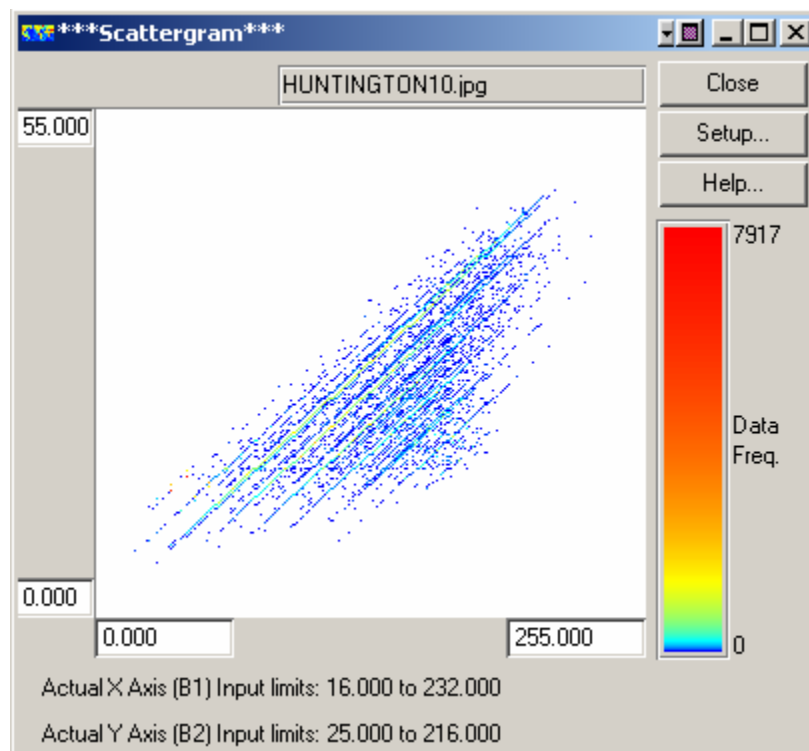


Figure 4-35 Scattergram for Hutnogton10

Fig. 4.35 that represents the scattergram for Huntington10 shows a smaller number of “more scattered or spread” sample points. The distance between pixels

is larger in both directions of the scattergram and the area that represents the water does not have distinguishable colors anymore. Compare fig. 4.35 with fig. 4.36 to see how “more dense” the scattergram in fig. 4.45 is and the more information it has.

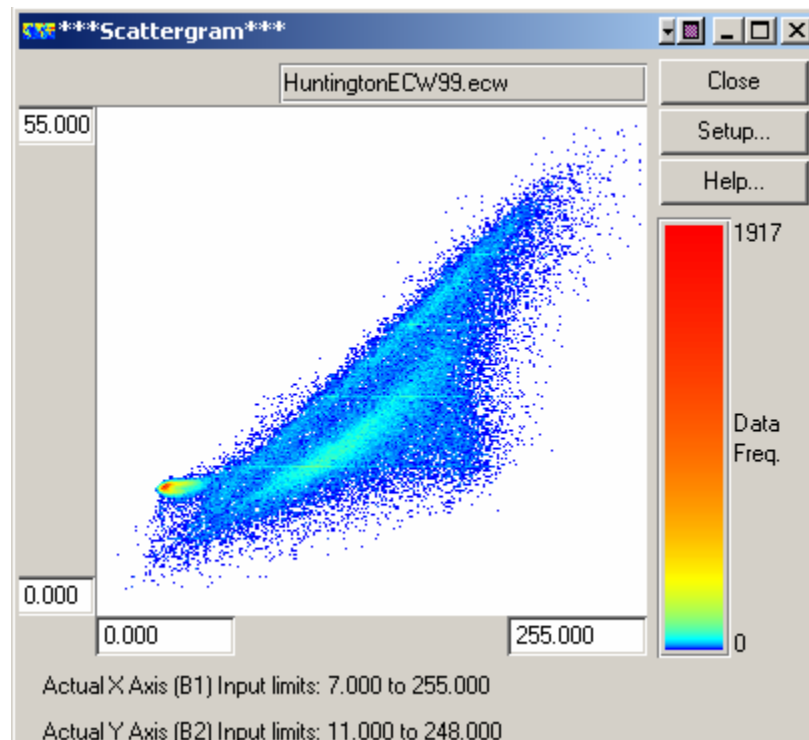


Figure 4-36 Scattergram for 99 times compressed ECW

According to the discussion above, we can conclude that ECW retains more data and data samples and is better in terms of less data destruction. See how many more samples are still there, and how clearly different objects are (especially water) **just for 16KB more storage space! This is certainly going to affect any classification attempts.**

Results for this part

Images, photographs, and pictures for personal use are typically less critical than those used in fields of studies such as mapping, GeoBioPhysical modeling, and transportation. They can be less clear and accurate; so, they can be compressed further using cheap compression software programs that allow storage in smaller spaces and be transmitted for less cost. ER-Mapper created ECW compressor to serve the fields of GeoBioPhysical studies and to retain the data that is necessarily to be kept during compression.

Personally, I would use 10% JPEG for some personal purposes (such as to show a rough image of Huntington on my website or to send it to a friend as an attachment to an e-mail), but I'd rather sacrifice 16more kilobytes and use ECW99 for my studies and research.

CHAPTER V

Summary and Conclusions

There are many different image formats out there; some of which are compressed while others are not. Compressed ones use either lossy or lossless techniques to give different qualities, file sizes, and number of colors. Formats might differ in many things such as the number of colors they show, the number of pixels, and the geometrical size of the image (the area it takes on the screen).

An image can be accessed by the compressing software in different ways where the number of colors or the number of pixels can be reduced. The image and the pixel values can also be encoded as data.

The creation of JPEG boosted the usage of photographic imagery on the internet, but it didn't replace GIF entirely since GIF continued working better with images of few colors and geometrical shapes and simple lines. PNG overcame GIF by the number of displayed colors; but, GIF is still better in animated scenes.

Moving through this research, I find that everybody is moving towards compressed imagery, even those working in fields related to the study of earth and earth resources. Researchers in such fields used to deal with very large bulky images that used to take

much time and storage space to process and save. ECW and its new relations with JPEG2000 show much promise for satellite and earth studies imagery fields.

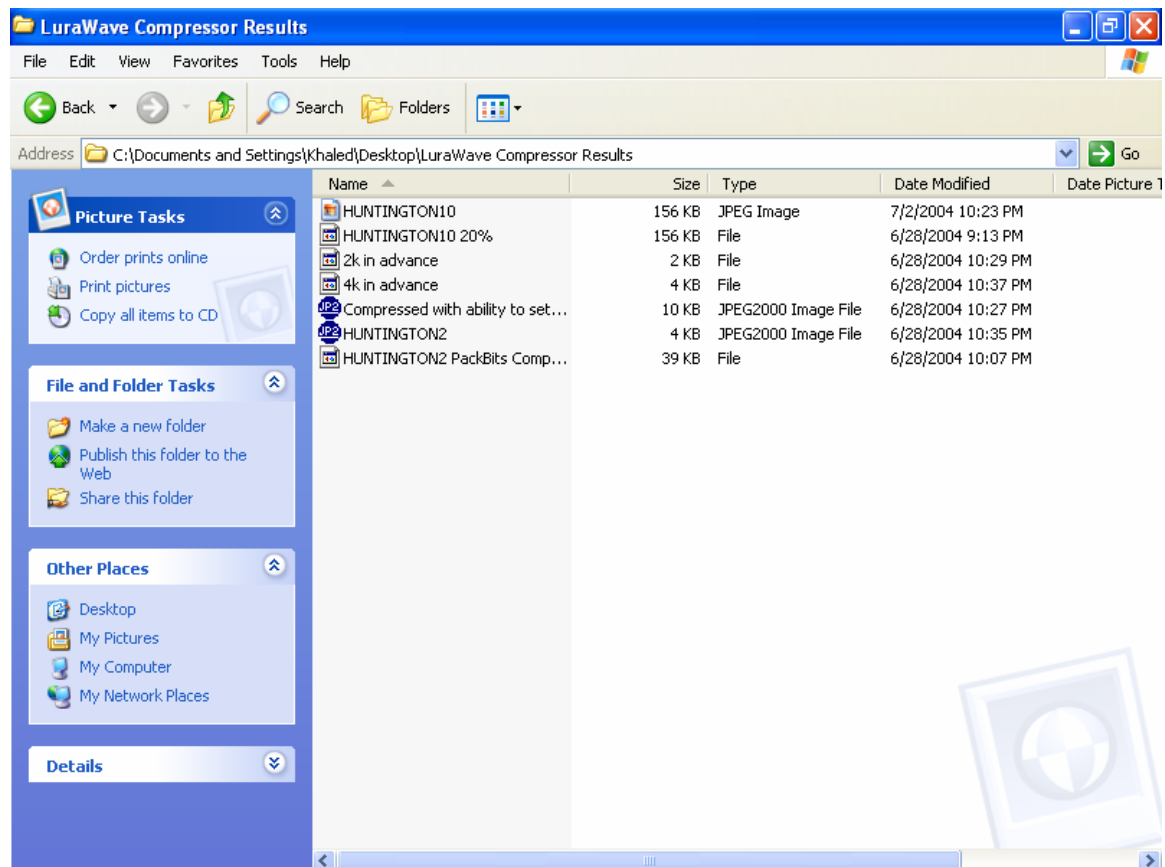
Un-compressed images are not being used that much anymore; they're just being used to get compressed ones. Nevertheless, they have to be displayed and studied at least once at the beginning of the study. Otherwise, the effect of compression will not be known. Lost data cannot be recovered once the image is compressed using lossy compression (which is used most of the time due to the high compression rates it gives). Original images have to be saved for future need and use.

The most important factors that restrict and control the compression ratio, the software used, and the output extension (.jpg, .ecw, .jp2, ...etc.) are the purpose for the image use, the available compression software, the compatibility with the used hardware and the software, and its future needs.

It's obvious that an image that will be sent as an e-mail can be compressed (or reduced in geometrical size) more freely than another one that will be used to create mosaics for a specific city or for some studies such as soil properties for example.

Table 3.2 compares the file size of different outputs of the original image compressed using different software programs. Refer to chapter 3 to compare quality.

This table shows that traditional JPEG compression, that was considered to be an excellent way to get high quality compressed imagery, does not compete with the results obtained using developed forms of JPEG such as JPEG2000 and many others such as ECW, and PackBits compression.



Name	Size	Type	Date Modified	Date Picture 1
HUNTINGTON10	156 KB	JPEG Image	7/2/2004 10:23 PM	
HUNTINGTON10 20%	156 KB	File	6/28/2004 9:13 PM	
2k in advance	2 KB	File	6/28/2004 10:29 PM	
4k in advance	4 KB	File	6/28/2004 10:37 PM	
Compressed with ability to set...	10 KB	JPEG2000 Image File	6/28/2004 10:27 PM	
HUNTINGTON2	4 KB	JPEG2000 Image File	6/28/2004 10:35 PM	
HUNTINGTON2 PackBits Comp...	39 KB	File	6/28/2004 10:07 PM	

Table 3.2 file size of different outputs of the original image compressed using different software programs

The last table (3.2) displays the differences between selective outputs for the software programs used in this study as an overall conclusion. It shows the file sizes to allow the reader compare. Taking this table, the previous results in this chapter and the last one, and the output images, we can conclude that there is no perfect technique or software

program that is capable of compressing imagery without data loss. The user has to sacrifice some quality for disk space and fast transmission of data. On the other hand, we can test and compare the different techniques and programs to get the best and the one that is closer to the ideal..... the cost and the price of the software has to be taken into consideration (fortunately many effective programs and tools are being offered for free online).

Dealing with different types of imagery is sensitive and depends strongly on the application; the user has to know what he is doing, what images he/she is compressing, to which extent he has to compress them, and the limits after which significant and important data and information start to be altered. He/she has also to use the proper input imagery from which he can get best results; in this study for example, we found that using TIFF as the input image was much better to be used as the input for a 10K output compressed JPEG.

Taking into consideration the results of this study and the ability to get them for free, we can conclude that JPEG2000 software programs (such as LuraWave) are very good and effective choices; compressed images obtained by LuraWave in chapter 3 were very good small size high quality images.

We can say that JPEG2000 and ECW are likely to be extensively used in the near future for imagery and internet usage. The results of this study show relatively excellent quality

to size ratios and great ability to sustain data and details even at high compression ratios. ER-Mapper is trying to mix both of them and get the benefits of both; this will also boost their effectiveness and abilities.

Finally, if we compare this study to other previous ones, we will find that the results were reasonable. Another related article about the work performed by M. Siragusa & D. J. McDonnell (Department of Endodontics, School of Dentistry, Universidad Nacional de Rosario, Santa Fe, Rosario, Argentina) discusses the “Limit of Image Compression for Diagnosis in Endodontics”; similar experiment were performed on some images in that study showing similar results. Refer to EBSCO Host Research Databases

[http://web8.epnet.com/resultlist.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+db+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2Dimage++compression+db%5B0+%2Daph+op%5B2+%2Dand+op%5B1+%2Dand+op%5B0+%2D+AF78&fn=251&\)](http://web8.epnet.com/resultlist.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+db+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2Dimage++compression+db%5B0+%2Daph+op%5B2+%2Dand+op%5B1+%2Dand+op%5B0+%2D+AF78&fn=251&)) to

access the following webpage that shows Siragusa’s article:

http://web8.epnet.com/externalframe.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+db+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2Dimage++compression+db%5B0

http://web8.epnet.com/externalframe.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+db+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2D+image++compression+db%5B0+%2D+aph+op%5B2+%2D+and+op%5B1+%2D+and+op%5B0+%2D+AF78&fi=aph_8594907_AN&lpdf=true&pdfs=91K&bk=R&tn=817&tp=CP&es=cs%5Fclient%2Easp%3FT%3DP%26P%3DAN%26K%3D8594907%26rn%3D252%26db%3Daph%26is%3D01432885%26sc%3DR%26S%3DR%26D%3Daph%26title%3DInternational%2BEndodontic%2BJournal%26year%3D2002%26bk%3D&fn=251&rn=252

Another useful article the discusses the ability to a way to compress and transmit different parts of a digital image separately; Xerox researcher Robert R. Buckley has come up with a way that performs such operations. Refer to

http://web8.epnet.com/externalframe.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+db+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2D+image++compression+db%5B0+%2D+aph+op%5B2+%2D+and+op%5B1+%2D+and+op%5B0+%2D+AF78&fi=aph_12579130_AN&lpdf=true&pdfs=302K&bk=R&tn=817&tp=CP&es=cs%5Fclient%2Easp%3FT%3DP%26P%3DAN%26K%3D12579130%26rn%3D47%26db%3Daph%26is%3D1099274X%26sc%3DR%26S%3DR%26D%3Daph%26title%3DTechnology%2BReview%26year%3D2004%26bk%3D&fn=1&rn=47 to access this article or use EBSCO Host

Research Databases

http://web8.epnet.com/resultlist.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+db+aph+cp+1+9D78&_us=hd+False+hs+True+or+Date+fh+False+ss+SO+sm+ES+dstb+ES+ri+KAAACB3A00039792+942D&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2D+image++compression+db%5B0+%2D+aph+op%5B2+%2D+and+op%5B1+%2D+and+op%5B0+%2D+AF78&fi=aph_942D

Image Compression

Khaled S. Alkharabsheh

[uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2Dimage++compression+db%5B0+%2Daph+op%5B2+%2DAnd+op%5B1+%2DAnd+op%5B0+%2D+AF78&uh=1&sci=S2&sbt=1&lfr=ES](#) to find it under

the title “Unpacking Pixels”.

Bibliography

Books

- Jensen, R. J. (1996). Introductory Digital Image Processing. New Jersey. Prentice Hall, Inc.
- Mitchel, F. H. Jr. & Mitchel, F.H. Sr. (1988). Introduction to Electronics Design. New Jersey. Prentice Hall, Inc.
- Savant, C. J. Jr. et. al. (1991). Electronics Design. Redwood city, California. The Benjamin/Cummings Publishing Company, Inc.
- Spasov, P. (1999). Microcontroller Technology. New Jersey. Prentice Hall, Inc.
- Wakerly, J. F. (2000). Digital Design. New Jersey. Prentice Hall, Inc.
- Watkinson, J. (1999, 2000). MPEG-2. Oxford. Focal Press.

Articles and Databases

http://web8.epnet.com/resultlist.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+db+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2Dimage++compression+db%5B0+%2Daph+op%5B2+%2Dand+op%5B1+%2Dand+op%5B0+%2D+AF78&fn=251&.

http://web8.epnet.com/externalframe.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+db+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2Dimage++compression+db%5B0

Image Compression

Khaled S. Alkharabsheh

[http://web8.epnet.com/externalframe.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+dbs+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2D+image++compression+db%5B0+%2D+aph+op%5B2+%2D+and+op%5B1+%2D+and+op%5B0+%2D+AF78&fi=aph_12579130_AN&lpdf=true&pdfs=302K&bk=R&tn=817&tp=CP&es=cs%5Fclient%2Easp%3FT%3DP%26P%3DAN%26K%3D12579130%26rn%3D47%26db%3Daph%26is%3D1099274X%26sc%3DR%26S%3DR%26D%3Daph%26title%3DTechnology%2BReview%26year%3D2004%26bk%3D&fn=1&rn=47](http://web8.epnet.com/externalframe.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+dbs+aph+cp+1+9D78&_us=hd+False+hs+True+cst+0%3B1%3B2%3B3+or+Date+fh+False+ss+SO+sm+ES+sl+0+dstb+ES+ri+KAAACB3A00039792+C486&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2D+image++compression+db%5B0+%2D+aph+op%5B2+%2D+and+op%5B1+%2D+and+op%5B0+%2D+AF78&fi=aph_8594907_AN&lpdf=true&pdfs=91K&bk=R&tn=817&tp=CP&es=cs%5Fclient%2Easp%3FT%3DP%26P%3DAN%26K%3D8594907%26rn%3D252%26db%3Daph%26is%3D01432885%26sc%3DR%26S%3DR%26D%3Daph%26title%3DInternational%2BEndodontic%2BJournal%26year%3D2002%26bk%3D&fn=251&rn=252)

http://web8.epnet.com/resultlist.asp?tb=1&_ug=sid+98B67B46%2D8C8C%2D4FBD%2D8429%2D3273D3A555FF%40sessionmgr3+dbs+aph+cp+1+9D78&_us=hd+False+hs+True+or+Date+fh+False+ss+SO+sm+ES+dstb+ES+ri+KAAACB3A00039792+942D&_uso=hd+False+tg%5B2+%2D+tg%5B1+%2D+tg%5B0+%2D+st%5B2+%2D+st%5B1+%2D+st%5B0+%2D+image++compression+db%5B0+%2D+aph+op%5B

[ES](#)

World Wide Web

<http://computer.howstuffworks.com/question289.htm>. Retrieved on July, 05 2004.

<http://computer.howstuffworks.com/question408.htm>. Retrieved on July, 05 2004.

<http://computer.howstuffworks.com/search.php>. Retrieved on July, 05 2004.

<http://www.cs.sfu.ca/CourseCentral/365/li/material/cgi-bin/whichjpeg.cgi>. Retrieved on June, 08 2004.

<http://www.gamesx.com/misctech/compress/caindex.htm>. Retrieved on July, 02 2004.

<http://www.ibiblio.org/otis/notes/otis-compression.html#JPEGis>. Retrieved on July, 04 2004.

http://www.wfu.edu/~matthews/misc/jpg_vs_gif/JpgCompTest/JpgChromaSub.html.

Retrieved on July, 08 2004.

<http://www.snapfiles.com/freeware/gmm/fwgcomp.html>. Retrieved on July, 08 2004.

<http://www.snapfiles.com/shareware/gmm/swgcomp.html>. Retrieved on July, 08 2004.

<http://www.softlandmark.com/imageComp1.htm>. Retrieved on July, 08 2004.

<http://www.geocities.com/thesciencefiles/compression/info.html>. Retrieved on July, 09 2004.

<http://www.ggcomputer.com/jpg-converter.htm>. Retrieved on July, 09 2004.

<http://users.senet.com.au/~mjbone/Fractals.html>. Retrieved on July, 08 2004.

<http://www.altamira-group.com>. Retrieved on July, 11 2004.

<http://www.webopedia.com/TERM/J/JPEG.html>. Retrieved on June, 16 2004.

http://www.webopedia.com/DidYouKnow/Internet/2002/JPG_GIF_PNG.asp. Retrieved on July, 11 2004.

<http://www.colosseumbuilders.com/imageformats/jpegenco.cpp>. Retrieved on July, 21 2004.

<http://www.colosseumbuilders.com/imageformats/compressedimageformats.htm>. Retrieved on July, 05 2004.

<http://www.colosseumbuilders.com/imageformats/>. Retrieved on July, 14 2004.

<http://www.webdevelopersnotes.com/graphics/jpgs.php3>. Retrieved on July, 08 2004.

<http://www.gamesx.com/misctech/compress/caindex.htm>. Retrieved on July, 08 2004.

<http://www.chami.com/tips/internet/062298I.html>. Retrieved on June, 25 2004.

<http://www.jpegwizard.com/>. Retrieved on July, 08 2004.

<http://www.jpeg.org/jpeg2000/>. Retrieved on July, 16 2004.

<http://www.cs.sfu.ca/CourseCentral/365/li/material/notes/Chap4/Chap4.2/Chap4.2.html>.

Retrieved on July, 19 2004.

<http://dmoz.org/Computers/Software/Graphics/Optimization/>. Retrieved on July, 08 2004.

<http://pascalzone.amirmelamed.co.il/Graphics/JPEG/JPEG.htm> retrieved on June 26, 2004

<http://www.chami.com/tips/internet/062298I.html>. Retrieved on June, 11 2004.

<http://www.ermapper.com/ecw/>. Retrieved on May, 27 2004.

<http://www.ermapper.com/downloads/default.aspx#1>. Retrieved on May, 27 2004.

<http://www.gisdevelopment.net/technology/ic/techip0003pf.htm>. Retrieved on July, 08 2004.

<http://www.webopedia.com/TERM/F/FIF.html>. July 8, 2004.

<http://caad.arch.ethz.ch/teaching/nds/ws98/computing/image/>. Retrieved on July, 01 2004.

<http://www.algovisionluratech.com/>. Retrieved on July, 13 2004.

http://www.webopedia.com/TERM/L/lossy_compression.html. Retrieved on July, 08 2004.

<http://www.ermapper.com/ecw/>. Retrieved on May, 28 2004.

Image Compression

Khaled S. Alkharabsheh

<http://www.gamesx.com/miscitech/compress/caindex.htm>. Retrieved on July, 08 2004.

<http://www.w3.org/Graphics/PNG/>. Retrieved on July, 08 2004.

<http://www.webcolors.freemove.co.uk/png/>. Retrieved on July, 08 2004.

<http://www.webcolors.freemove.co.uk/png/png24.htm>. Retrieved on July, 08 2004.

<http://www.winsoftmagic.com/?ref=g30> Retrieved on June, 10 2004.

<http://www.wvgis.wvu.edu/data/dataset.php?action=search&ID=74>. Retrieved on May, 13 2004.

Software Programs Used

1. Er-Mapper 6.4 by Earth Resource Mapping.
2. ECW compressor v2.6 by Earth Resource Mapping. downloaded on June 2, 2004 from
http://www.ermapper.com/downloads/download_view.aspx?PRODUCT_VERSION_ID=152.
3. Er-viewer 2.0e by Earth Resource Mapping. Downloaded on June 2, 2004 from
http://www.ermapper.com/downloads/download_view.aspx?PRODUCT_VERSION_ID=48.
4. Advanced JPEG Compressor v. 4.1(1999-2002) by WinSoftmagic Development; downloaded on June 2, 2004 from <http://www.winsoftmagic.com/?ref=g53>.
5. Adobe Acrobat Reader v6.0 (1987-2003) by Adobe Systems Incorporated.
6. Adobe Photoshop Album 2.0 Starter Edition (2003) by Adobe Systems Incorporated.
Downloaded on June 2, 2004 from:
<http://www.adobe.com/support/downloads/detail.jsp?ftpID=2176>.
7. LuraWave SmartCompress 3.0 (1996-2002) by Algo Vision LuraTech GmbH. Berlin, Germany. Downloaded on June 2, 2004 from <http://www.algovision-luratech.com>.
8. PaperPort (PaperPort LE7.0 by ScanSoft Inc).
9. MrSID Viewer version 2.0 (1997-1998) by Lizard Tech, Inc. & International Land Systems.

Attachments
JPEG source code

JPEG software programs are being compiled, distributed, and sold by many companies and firms; they have been written using many different programming languages and codes where the software tells the CPU what to do with the image and data. Different codes those perform the same task can be written using the same or different languages. The differences between the different software programs in performing a specific task are due to the way the code was written, the way the code makes the CPU work on the image, and the creativity of the compiler(s).

Refer to <http://www.colosseumbuilders.com/imageformats/jpegenco.cpp>

to get an example of a JPEG source code.