STDI-0003
September 1998

**National Imagery and Mapping Agency**

# National Imagery Transmission Format Standard Imagery Compression Users Handbook

22 September 1998

# FOREWORD

The National Imagery Transmission Format Standard (NITFS) is the standard for formatting digital imagery and imagery-related products and exchanging them among the Department of Defense (DOD), other Intelligence Community (IC) members, and other United States Government departments and agencies. This handbook is provided to assist users in its application and use.

The National Imagery and Mapping Agency (NIMA), Standards and Interpretability Division developed this handbook based on currently available information.

The DOD and IC are committed to interoperability of systems used for formatting, transmitting, receiving, and processing imagery and imagery-related information. This handbook describes the various compression algorithms resident within the NITFS and provides information regarding their use.

Beneficial comments (recommendations, additions, deletions) and other pertinent data which may be of use in improving this document should be addressed to the S. Danny Rajan, Chair, National Imagery Transmission Format Standard (NITFS) Technical Board (NTB), National Imagery and Mapping Agency (NIMA), Standards and Interoperability, MS P-24, 12310 Sunrise Valley Drive, Reston, VA 20191-3449.

# TABLE OF CONTENTS



# LIST OF FIGURES



# LIST OF TABLES

# LIST OF IMAGES

# 1.0 SCOPE

## 1.1 PURPOSE

The purpose of this handbook is to help readers understand the various National Imagery Transmission Format Standard (NITFS) compression algorithms. It provides technical information in lay mans terms so that readers without scientific or technical backgrounds can understand how compression algorithms are used in NITFS.

The handbook is designed for tactical users who work primarily with tactical images that are about 4 megabytes or smaller in size. Some of the compression algorithms and procedures practiced at National imagery exploitation centers are identified, but they are not addressed in detail. However, National users should find this handbook informative.

## 1.2 INTRODUCTION

The NITF family of documents, comprised of Military Standards (MIL-STDs) and other documents, contains words and phrases (e.g., pixel, bits per pixel, Huffman tables, lossy and lossless) that create apprehension and uncertainty if the user is not an engineer or scientist. Additionally, complex mathematical formulas are presented with little or no explanation. It seems to the non-technical type that these documents are written primarily for those who possess a Doctorate in physics. Many have probably wondered what is a pixel? Is it a special type of pixie or is it a leprechaun? If the user does not know what a pixel is, he cannot understand compression presented as some number of bits per pixel. This handbook addresses these issues and presents them so they can be understood and used by those who are not learned and expert in the mysteries of physics, electrical engineering, computer science, and similarly subjects.

Various compression algorithm examples are presented so that the user can readily understand their effects on images. Further discussion of why, when, and, most importantly, how to use the compression routines within NITFS is presented. Instances when compression should not be used are also covered. The example imagery used and discussed is the same type used to satisfy imagery intelligence and imagery related intelligence requirements in the Department of Defense (DOD) and other Intelligence Community (IC) organizations.

Remember, it is not necessary for the user to understand every technical detail or to be able to solve the mathematical formulas. A computer system designed specifically to perform NITFS operations should be used to apply NITF and its compression algorithms. Hopefully, the user is familiar with NITF software applications. These applications will perform the routines automatically; the user needs to know when to use or not use compression, what kind of compression to use, and the reasons why. This handbook provides enough information for users to impress others with their compression knowledge. If nothing else, users should at least check the summaries for each of the compression algorithms.

### 1.3 ORGANIZATION

The handbook is organized to assist readers in understanding and using the various NITF compression algorithms. Information is presented in logical order. Since the handbook's primary purpose is to address compression, NITF information is presented in appendix A.

- Compression – description
  - Lossless compression
  - Lossy compression
  - Proprietary and commercial compression
- Using Compression
  - Why and when
- Image Types
  - Visual (Electro-optical), Infrared (IR), Synthetic Aperture Radar (SAR), Multi-spectral, Color, Video (single frame)
- NITF Compression Algorithms
  - JPEG
  - Bi-Level Compression
  - Video Compression
  - National Image Compression
- Other Compression Technologies
  - Wavelets
  - Fractals
  - Place Holder (Other)
- JPEG 2000
- Summary
- Appendices
- NITF (Background and History
- Acronyms and Abbreviations
- Points of Contact

Use this organization to go quickly to the topic needed.

# 2.0 COMPRESSION

## 2.1 GENERAL DISCUSSION

To those of us who are not experts in technical matters, compression appears to be magic. However, all of us are familiar with acronyms or abbreviations. Typically, we use them so that we use less space on the written page. For example, NITF is the abbreviation for National Imagery Transmission Format. This concept of transforming an object so that it uses less space is the core idea behind all compression techniques.


FIGURE 1: INDIVIDUAL PIXEL EXAMPLE

Since we wish to compress images or pictures, we should understand how a computer displays and stores pictures. If the user looks closely at his computer monitor, he will see that the picture or text displayed is actually composed of many small colored squares, or pixels. The two pictures of a leprechaun's hat shown here demonstrate this. One picture shows an enlarged view of only a small portion of the other, making it easy to begin to see the pixels that make up the image. It is also easy to realize that a pixel is not a magic leprechaun or pixie, but rather the individual elements that make up an image. The number of pixels in an image determines its size and/or detail level. The more pixels in an image at a given size, the greater detail that it will present. Another way to demonstrate pixels is to examine a single letter. Shown here is the letter "A". Note that the individual pixels can be observed in the right hand "A".

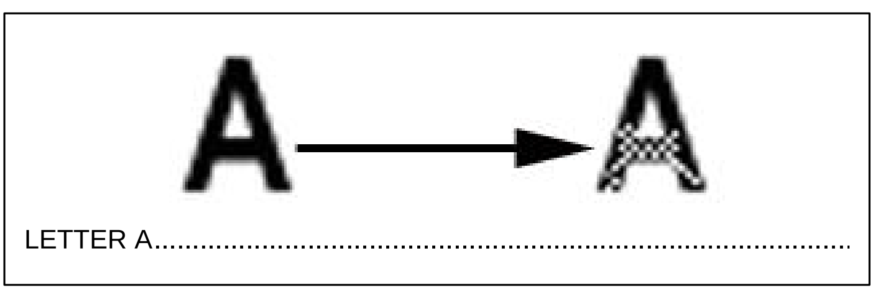



FIGURE 2: LETTER "A" PIXEL EXAMPLE

Now, let's address bits-per-pixel. A digital image is displayed on a computer's monitor in pixels. Each pixel is identified with a certain predefined color or shade of gray. The number of colors available, or color depth, and their exact value depends on the number of bits used to store each pixel. This is referred to as the number of *bits-per-pixel*. If each pixel occupies one byte (8 bits) the image can only have one of 256 ($=2^8$) colors (or shades of gray). If each pixel occupies 2 bytes (16 bits), there are 65536 ($=2^{16}$) available colors. For a three byte pixel (24 bits), there are over 16.7 million ($=2^{24}$) available colors. Remember, that the number of pixels and *bits-per-pixel* determine its size. An image composed of one byte pixels that is 4 pixels in length by 4 pixels wide (4 x 4 x 1 byte) constitutes a file of 16 bytes. Following this logic, for the same one byte pixels, if the size is 512 x 512 x 1, this creates an image file of 262,144 bytes. A 1024 X 1024 X 1 makes an image file of 1,048,576 bytes. This means that a 2048 x 2048 x 1 byte image file is 4,194, 304 bytes, which just happens to correspond to the average size of the majority of the images used by the tactical intelligence community. One does not have to be a rocket scientist to realize that if this same image is composed of 3 byte pixels (or color) then the image file is three times as large (12,582, 912 bytes). The National community also produces 11 and 12 bit imagery, which is stored in 2 bytes. (Note: digital image data is always stored in 1-byte chunks, in the case of national data, the top 4 or 5 bits are zero filled. This is to bring 11 or 12 bit imagery up to 16 bits/2bytes). This type of imagery is not addressed. This handbook primarily addresses *8 and 24* bit image data.

As we have just seen, image files can be very large, and that large file size is the reason that compression is used. We wish to reduce the size of the image files so we may transmit them faster or so that we may store more of the files on a computer hard drive. Compression then, simply reduces the number of bytes that comprise the image. For example, and 8 bit per pixel image could be compressed to 1 bit per pixel, and 8:1 compression ratio.

### 2.2 Types of Compression

Compression can be accomplished using a wide variety of software and/or hardware. There are two basic types or classifications of compression: *lossless and lossy*.

Lossless compression consists of those techniques guaranteed to generate an exact duplicate of the input data stream (or original image) after the data has been compressed and then decompressed back into its original form. Lossless compression techniques are typically used for textual files, such as database records, word-processing files and other documents because loss of data is unacceptable in these applications. This type of compression can also be used on digital images. For digital images, the number of bits that comprise the original image is reduced (compressed) without losing any data or quality of the original uncompressed image.

*Lossy* compression allows a certain loss of accuracy in exchange for greatly increased compression. Lossy compression proves most effective when applied to digital images and digitized voice. In this type of compression the size of the image is reduced by <u>actually discarding</u> some pictorial data.

Data compression consists of taking a stream of symbols and transforming them into codes (figure 3). One way to look at data compression is in terms of the redundancy of an image or textual message. The redundancy of information in a mess-age is derived from the repetitive nature of the symbols that appear in the message and not the actual content of the message. In a message, redundant information takes extra bits to encode, and if the redundant information can be removed, the size of the message can be reduced.

| **Symbol** | **Code** |
|---|---|
| **A** | **1100** |
| **E** | **100** |

FIGURE 3: SYMBOL AND CODE EXAMPLE

In lossless compression the decision to output a certain code for a certain symbol or set of symbols is based upon a model. The model is a collection of data and rules used to process input symbols and determine which code(s) to output. A program uses the model to accurately define the probabilities for each symbol. The model feeds the coder its probabilities. The coder then produces an appropriate code based upon the probabilities. This process is graphically portrayed below.

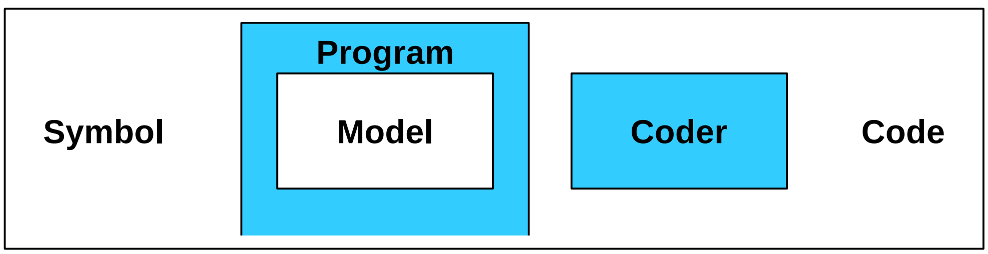



FIGURE 4: LOSSLESS COMPRESSION PROCESS

In both lossy and lossless image compression, digital data is compressed. Usually users are provided digital image data. However, in some organization's hardcopy imagery must be digitized to provide digital image data. The desired compression rate is usually associated with a quality factor, which determines just how much compression occurs. The more the data is compressed the greater the data loss, the more the redundancy in the data is reduced. Therefore, the greater the compression that takes place. The greater the compression, the greater the loss of quality compared to the original uncompressed image.

Digital images are compressed using essentially the same procedures for either lossless or lossy compression. A model is used to process the input symbols and determine which codes to output. A program then uses the model to accurately define the probabilities for each symbol. The coder then produces an appropriate code based upon the probabilities. This process is depicted in figure 4.

Speaking of digitizing operations. It is better to digitize an image at a high bit per pixel setting rather than a low one. The resulting higher quality images will be of a much larger file size. However, compression can be used to reduce the file size. The higher quality of the image provides a better quality-compressed image than one digitized at a low bit per pixel rate.

It is necessary to understand a few more details about symbols and codes. There are two basic techniques that develop the collection of rules used to process the input symbols and determine which codes should be outputted for each input symbol. In *statistical* encoding a single symbol is encoded at a time using the probability of that character's appearance. In *dictionary-based* encoding a single code is used to replace strings of input symbols. It reads in input data and looks for groups of symbols that appear in a reference table as shown in figure 5. If a match is found, a pointer or index into the reference table is outputted instead of the code for the symbol.

| Symbol | Probability | Code |
|---|---|---|
| A | 20% | 1100 |
| E | 50% | 100 |

**STATISTICAL EXAMPLE**

| String of Symbols | Pointer | Code |
|---|---|---|
| And | 550/2 | 11010 |
| Qu | 173/46 | 01101111 |

**DICTIONARY EXAMPLE**

FIGURE 5: ENCODING EXAMPLES

The tables can be built by either reading the text (or image) once to generate the statistics, compiling the table and transmit it with the file. The tables can also be continually modified as a new character or symbol is read and coded. Techniques that dynamically modify the tables as they are being processed are known as adaptive techniques. Those that generate the tables only once are known as static techniques.

Static tables can increase the overhead, which essentially lessens the compression ratio, since the table usually is sent with the file. Conversely, adaptive encoding is more computer intensive and the compressor and decompressor must possess identical models so that the decoder will be able to interpret the output of the coder. NITF (Version 2.1) is adaptive and will always sends the table with the file.

The NITF uses *Huffman* coding which is statistically based and produces a single code for each symbol. In Huffman coding a single code is produced for each symbol. Coding varies the length of the symbol in proportion to its information content. Symbols with a low probability of appearance are encoded with a code using many bits. Symbols with a high probability of appearance are represented with a code using fewer bits.

| Symbol | Huffman Code | Info Content | Bit Count | No. of Occurrences |
|---|---|---|---|---|
| E | 100 | 1.26 bits | 1bits | 20 |
| A | 1100 | 1.26 bits | 2 bits | 20 |
| X | 01101111 | 4.00 bits | 3 bits | 3 |

FIGURE 6: HUFFMAN CODING EXAMPLE

All Huffman codes have to be an integral number of bits long. In figure 6, Huffman Coding Example, the coding does not create codes with the exact information content required. In most cases it is a bit above or below the actual number of bits, leading to the deviation from the optimum.

Most compression applications are designed with predetermined quality settings. These settings are referred to as "Q" settings or levels. The lower the "Q" setting the greater the compression and loss of quality. Most NITF tactical imagery compression applications have five settings, from Q1 to Q5, so Q1 would produce the greatest compression ratio and Q5 would produce a much smaller compression ratio.

This is all we need to know to understand how the various NITF compression algorithms work. As each algorithm is addressed, additional information will be presented.

### 2.3 Commercial and Proprietary Compression

Though this handbook addresses NITF compression techniques, there is a great many commercial and proprietary compression applications. Most vendors ensure their product is capable of using the standard compression algorithms. Many commercial software suites also include a proprietary compression algorithm that only their software will perform. Standard compression algorithms included in most popular commercial suites at a minimum include the Joint Photographic Experts Group (JPEG), Lempel-Ziv-Welch (LZW), Run-Length-Encoding (RLE), Pack Bits, and Fax Group III. Many companies use a unique proprietary compression algorithm, which only their application can produce or open. Users cannot decompress or compress an image using this technique unless they have the particular application that produced the image file. It is much better to use a standard technique if possible, since most intended recipients will not have the required proprietary software application.

Users have no doubt been exposed to various imagery formats on the Internet and by opening files. Some of the more common include JPEG, Bit Mapped Image (BMP) Tagged Image Format (TIF) and CompuServe's Graphic Image Format (GIF). Most NITF (format) application suites allow users to access these types of image formats and others as well.

The NITF uses non-proprietary and commercially adopted compression algorithms. There are NITF applications that run on personal computers, workstations and even laptops. They span the majority of the operating systems, Window 3.xx, Windows 95, and most versions of UNIX. The NITF format is identified by the "*.ntf" extension, for example, Bigpic.ntf. The technical specifications concerning the format are available to anyone who requests them. If a user enjoys coding software programs, he can create his own application. The Government mandated that all DOD and other IC members use NITF for image transmittal. This mandate ensures interoperability among a wide range of government agencies and organizations. Users can receive (or send) an image from the Central Intelligence Agency (CIA), any of the four services, the State Department, and the Coast Guard, just to name a few.

### 2.4 Using Compression

We use compression to decrease imagery file sizes to transmit usable images in as short a time as possible or to store more images on hard drives. Remember that the average image size used in the tactical intelligence community is around 4 megabytes. Figure 7 shows how long it would take to transmit that file size at various rates.

| | 4 megabyte image file<br>Bits per Second | Time to Transmit |
|---|---|---|
| Tactical Radios | 2400 | 3 hr 42 min |
| Telephones | 4800 | 1 hr 51 min |
| | 9600 | 55 min |
| Satellite & Other High Speed Comms | 64,000 | 8 min |
| | T1 | 4 min |

FIGURE 7: TRANSMISSION TIME

(Transmission times are for ideal case with no overhead.) Transmission time over a T1 line is a respectable 4 minutes, but over a regular telephone line at 9600 bits per second, the transmission time goes up to 55 minutes. If we compress the image by 50% or a 2:1 compression ratio it will only take one half the time as shown in figure 8.

| Compressed File Size | Time to Transmit<br>2400 | 9600 | 64,000 |
|---|---|---|---|
| 2 megabytes (2:1) | 1 hr 51 min | 27.5 min | 4 min |
| 1 megabyte (4:1) | 55.5 min | 13 min | 2 min |
| 160,000 bytes (25:1) | 8.9 min | 2.1 min | 20 sec |
| 80,000 bytes (50:1) | 4.3 min | 1.1 min | 10 sec |
| 40,000 bytes (100:1) | 2.1 min | 33 sec | 5 sec |

FIGURE 8: TRANSMISSION TIMESAT VARIOUS COMPRESSION RATIOS

Note that the times are still excessive for our original 4-megabyte image until around the 25:1 compression point for the 2400 and 9600 rates. Users cannot afford to give up scarce communications resources if the image transmission times are excessive. So we want to make the image file as small as possible, yet still provide a useful image.

Consider the intended use of the image by the recipient. If they were going to conduct imagery analysis on the image they need as good a quality image as can be sent. Therefore, use a small compression ratio, say 2:1 or lossless compression. However, if the intended use is to just visually see if a bridge crosses a river for example, then we can use a lossy compression algorithm resulting in a much higher compression ratio. Also consider where the recipient is and their communications capability. When sending an image to a ground unit that only has tactical radios capable of 2400 bits per second transmission and receipt, sending uncompressed images is out of the question. Conversely, if the intended recipient has T1 or T3 connectivity, compression is not an issue at all.

Remember that compression does not have to be used in all cases. If the recipient requests uncompressed images they probably possess high bandwidth communications capability. In most all cases they know what they are asking for, and how long it will take to receive.

Users may think whoever wrote this manual is nuts, smaller is better. Simply decrease the size of the image, making an itty-bitty file, which can be sent super quick. This is not a good idea. If the actual size of the image is reduced, granted it is smaller but the capability to discern details and orientation from the image is lost. Making it bigger so it is usable does not work. Reduction of the image results in loss of quality and subsequent enlargement enables users to see the quality loss in greater detail. In figure 9, loss of detail is already appearing in the enlargement.

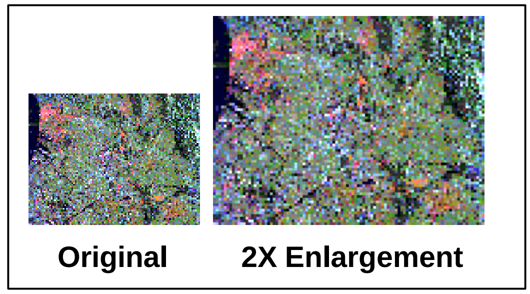



FIGURE 9: ITTY-BITTY IMAGE

### 2.5 HISTORY TAGS/CONCATENATION

It would be nice if, having compressed an image with one of the NITF algorithms, users could decompress it, manipulate it (crop off a side, for example), and recompress it without any further image degradation beyond what was lost initially. Unfortunately this is not the case. In general, recompressing an altered image loses more information. Therefore it’s important to minimize the number of generations of and type of compression between initial and final versions of a particular image. Concatenation is the name for linking multiple compression and decompression events. For example, if users decompress and recompress an NITF image at the same Q setting first used, little or no further degradation occurs. This means users can make minor local (e.g., adding an annotation) modifications to a JPEG NITF image without material degradation of other areas of the image. The areas users work in will still experience degradation. Surprisingly, this works better the lower the quality setting, but users must use exactly the same setting or higher, or the image will be significantly degraded.


**Original Compressed Once Q3** **Cropped Recompressed Q1**

FIGURE 10: CROPPING EXAMPLE

Unfortunately, cropping does not count as a minor local change! JPEG processes the image in small blocks (8x8 pixels), and cropping usually moves the block boundaries so that the image looks completely different to the JPEG process. This is apparent in the figure and can be easily seen in the text on the cropped image. Users can take advantage of the low-degradation if they are careful to crop the top and left margins only by a multiple of the block size (typically 16 pixels), so that the remaining blocks start in the same place. Cropping can also cause JPEG to reproduce the entire cropped image as a block or neighborhood area (see figure 11) when it decompresses the image.

FIGURE 11: JPEG CROPPING ERROR

To help make sure the same Q settings and algorithm are used if it becomes necessary to recompress an image file NITF Version 2.1 contains a Softcopy History Tag. The information in this tag provides a chronological listing of processing events, starting with the first event. A field for general comments for the users is also in the tag. Users can glean all types of information from the Softcopy History Tag. The type of compression and the Q setting previously applied to and NITF image file is in this tag. It will to guide users as to which algorithm and Q setting to use when recompressing an image. The information may not be present, since use of this tag is not mandatory.

## 2.6 IMAGE TYPES

The following images will be used to demonstrate NITF compression algorithms. Each time that a particular NITF compression algorithm is used the original uncompressed image will be presented at the top of the page followed by the compressed image.

These images are also provided in soft copy so that users may conduct their own compression experiments to see if they can get similar results with their particular NITF application.  The images are color, visual (visible electro-optical), infrared (IR), synthetic aperture radar (SAR), and single video frames collected by an Unmanned Aerial Vehicle (UAV).  The video frame was captured and converted from 24-bit color to 8-bit grayscale.  We are only addressing 8-bit grayscale and 24-bit color imagery.

This is a good time to address the two types of color models used in the NITF.  First, let's cover Cyan-Magenta-Yellow-Black (CMYK).  CMYK is pronounced as separate letters.  CMYK is a color model in which all colors are described as a mixture of these four process colors.  CMYK is the standard color model used in offset printing for full-color documents.  In contrast, display devices (monitors) generally use a different color model called RGB, which stands for Red-Green-Blue.  The sample color image is RGB, but when users compress color images they will be converted to the CMYK model.  They are normally converted back to the RGB model when users decompress the file.  So we have two color models one for printing and one for viewing.

We are using a variety of image types to show that compression may have very different effects on each.  In the examples, note that some image types undergo wide variations in quality after compression while others do not.

Image 1: Color Example

Image 2: Visual Example

Image 3: Infrared Example

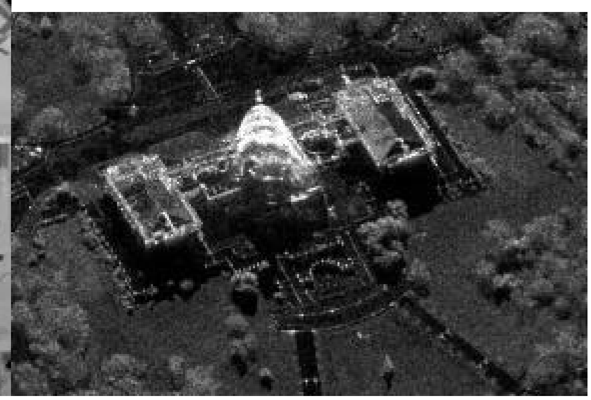


Image 4: Synthetic Aperture Radar Example

Image 5: UAV Example

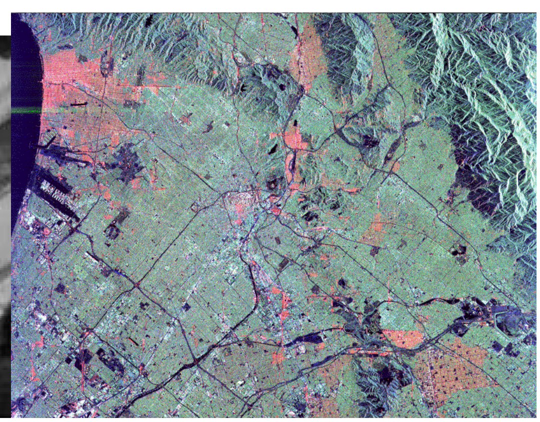


Image 6: Multi-Spectral Example

# 3.0 NITF COMPRESSION ALGORITHMS

## 3.1 JOINT PHOTOGRAPHIC EXPERTS GROUP IMAGE COMPRESSION

The first NITF algorithm that we will address is the Joint Photographic Experts Group (JPEG) image compression algorithm. The complete specifications are contained in MIL-STD-188-198A dated 15 December 1993. JPEG is one of the most popular and widely used compression algorithms. It is used on the World Wide Web (WWW) and was adapted for use in the NITF in the early 1990s. NITF JPEG also follows a model similar to those discussed previously to produce a compressed image. NITF JPEG's Forward Discrete Cosine Transform (FDCT) (shown in figure 15) is just another mathematical formula. JPEG divides the image into 8x8 minimum coding units or neighborhoods, and then calculates the FDCT of each neighborhood. The quantizer maps coefficients of similar values into the same value. This reduces the number of unique values and makes for more efficient coding. For decompression, JPEG recovers the quantized FDCT coefficients from the compressed data stream, takes the inverse transform (using embedded tables) and displays the image. This is a simple description; technical details are in the Military Standard. We need to know two things: 1) how the various Q settings impact the quality of an image, and 2) the time available to transmit the compressed image.

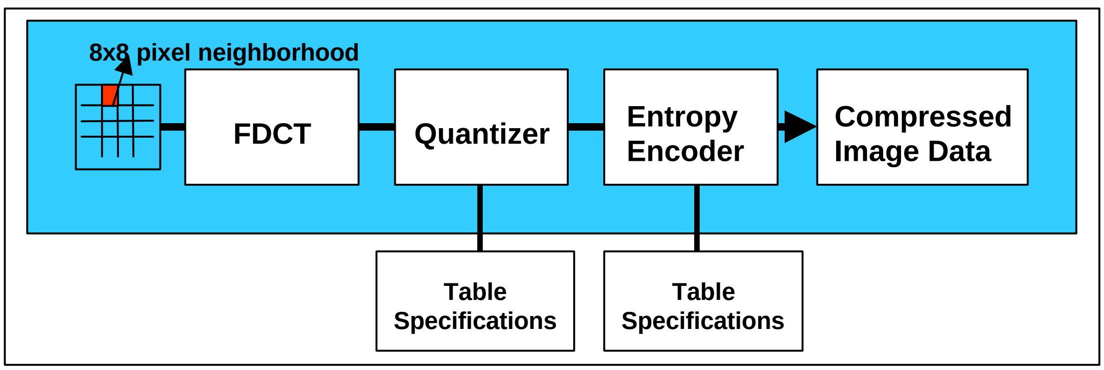



FIGURE 12: NITF JPEG DCT COMPRESSOR

NITF JPEG provides several variants, DCT Lossy (8 & 12 bit), Lossless and Downsample JPEG. Each is addressed.

## 3.2 DCT LOSSY

Remember that in lossy compression we sacrifice some quality for increased compression ratios. When users use DCT Lossy they can choose from 5 different Q levels (Q1 – Q5). Q1 provides the greatest compression with the greatest quality loss, and Q5 the least compression with the highest quality.

Now lets see how DCT Lossy performs on our test images. We will compress each of the sample images at every Q settings. As each image is presented the type of compression, Q setting, file size and time to transmit at 9600 bits per second will be shown. Note that transmission times shown are for the ideal case for a clean line with no overhead. This will help users in quickly assessing the effectiveness of compression as related to transmission time. The transmission times shown for the compressed images were compared to the original uncompressed images transmission time. Remember that whether a transmission time is acceptable or not depends on the situation, the time available, and the patience of the recipient, and other subjective factors.

The compression ratio will be shown for each Q setting. The conventional #:# format and also in bits per pixel. Bits per pixel compression is determined by the formula: Image bits (8, 24, etc) X 1 divided by the compression ratio expressed as a fraction. For example 25:1 compression equals of an 8-bit image is .32 bits per pixel compression. (8 X 1/25 = .32)

Color Original – 877, 364 Bytes (12 minutes 11 seconds)

DCT JPEG Q1 Setting –15,703 Bytes (13 seconds)

The DCT JPEG Q1 setting produced a compression ratio of 43:1 (.56 bits per pixel). Color images compress very well. Compare the two for differences. Note the different time to transmit.

Color Original – 877, 364 Bytes (12 minutes 11 seconds)

DCT JPEG Q2 Setting – 19,851 Bytes (15 seconds)

Using DCT JPEG Q2 setting we get a compression ratio of 34:1 (.70 bits per pixel). There is very little change in quality and we can still transmit the image in seconds.

Color Original – 877, 364 Bytes (12 minutes 11 seconds)

DCT JPEG Q3 Setting – 43,255 Bytes (36 seconds)

DCT JPEG Q3 we attained a compression ratio of 15.6:1 (1.53 bits per pixel). Remember that the quality of the image is increasing as the compression ratio decreases. We can still transmit the image in seconds at 9600 bits per second.

Color Original – 877, 364 Bytes (12 minutes 11 seconds)

DCT JPEG Q4 Setting – 53,660 (44 seconds)

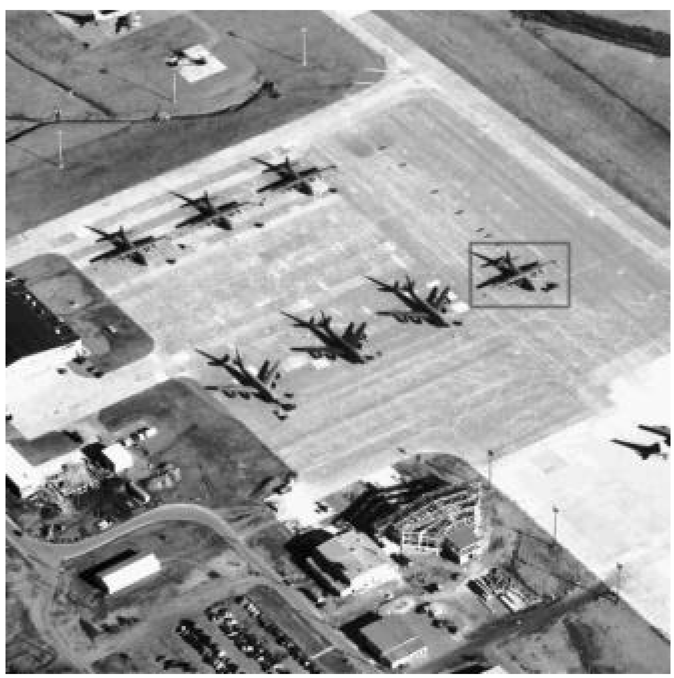
DCT JPEG Q4 setting provided a compression ratio of 12.6:1 (1.9 bits per pixel). The time to transmit is still under a minute, but there is benefit in the quality increase?

Color Original – 877, 364 Bytes (12 minutes 11 seconds)

DCT JPEG Q5 – 77,467 Bytes (1 minute, 4 seconds)

DCT JPEG Q5 setting produced a compression ratio of 8.7:1 (2.74 bits per pixel) and provides the greatest quality. It now takes over a minute to transmit the file.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)



DCT JPEG Q1 Setting – 301,070 Bytes (4 minutes, 10 seconds)

DCT JPEG Q1 setting on this image attained a compression ratio of 13.9:1 (.57 bits per pixel) and transmit time is just over 4 minutes. This grayscale image is the average size of the majority of the images users will encounter.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)

DCT JPEG Q2 Setting – 352,875 Bytes (4 minutes, 54 seconds)

DCT JPEG Q2 setting produced a compression ratio of 11.8:1 (.67 bits per pixel) and added 44 seconds to the transmission time.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)

DCT JPEG Q3 Setting – 521,903 Bytes (7 minutes 14 seconds)

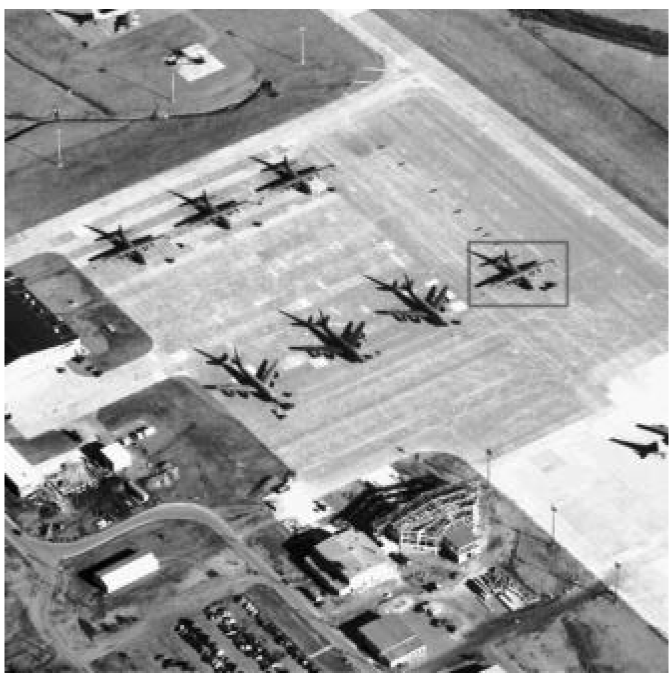
DCT JPEG Q3 setting attained a compression ratio of 8:1 (.99 bits per pixel) with an increase in quality. Notice the increase in transmission time.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)

DCT JPEG Q4 Setting – 595,837 Bytes (8 minutes 16 seconds)

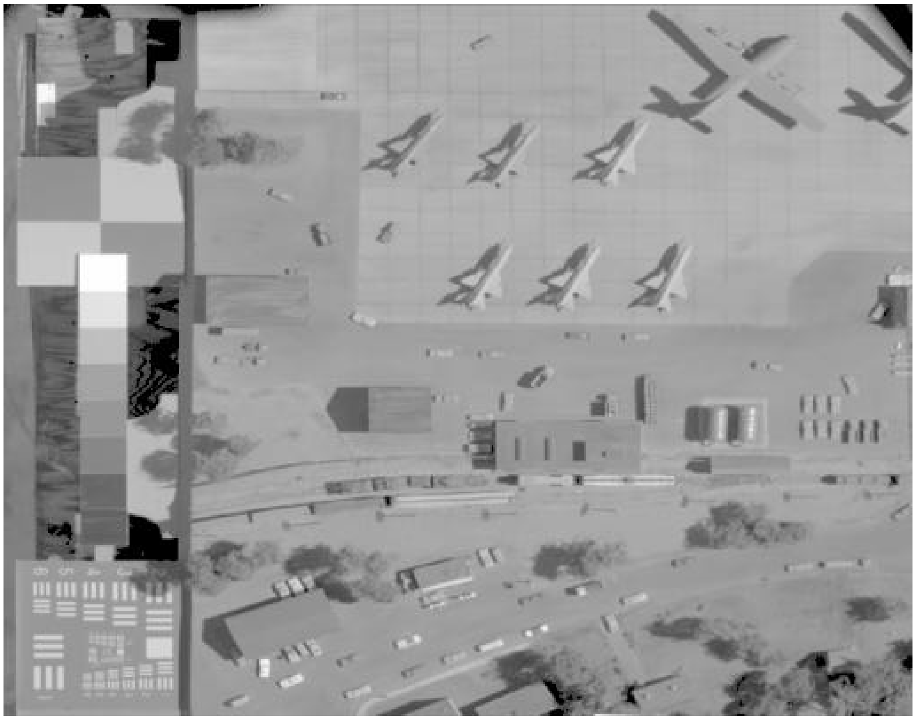
DCT JPEG Q4 setting provided a compression ratio of 7:1 (1.14 bits per pixel). This quality increase added about a minute to the transmission time.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)



DCT JPEG Q5 – 764,632 Bytes (10 minutes, 37 seconds)

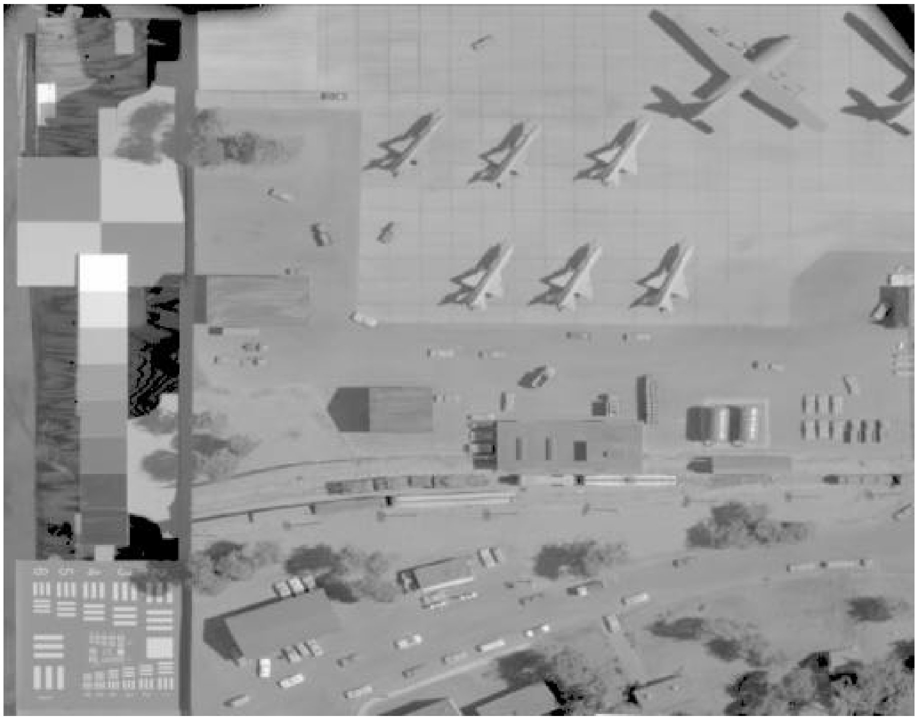
At the highest DCT JPEG quality setting, Q5, the compression ratio is 5.4:1 (1.46 bits per pixel). Note that it now takes over 10 minutes to transmit this image.



Original IR – 615,004 Bytes (8 minutes, 32 seconds)

DCT JPEG Q1 Setting – 14,083 Bytes (11 seconds)

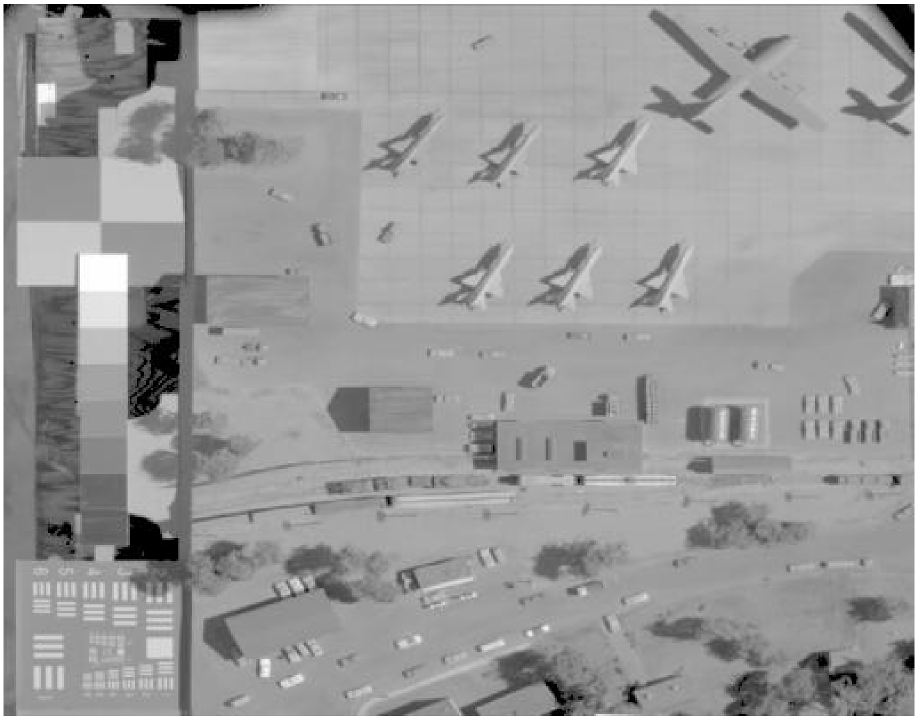
DCT JPEG Q1 setting a compression ratio of 43.6:1 (.18 bits per pixel) was produced and a very short transmission time. Infrared (IR), like color imagery compresses well.



Original IR – 615,004 Bytes (8 minutes, 32 seconds)

DCT JPEG Q2 Setting – 18,096 Bytes (15 seconds)

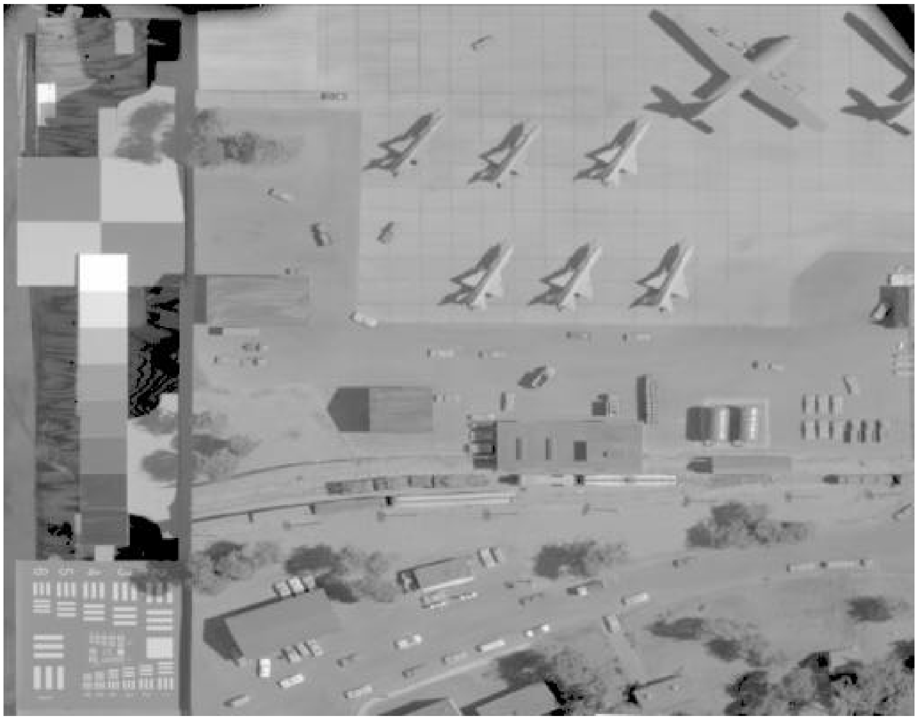
DCT JPEG Q2 setting attained a 34:1 compression ratio (.24 bits per pixel) yet still provides a more than acceptable transmission time.



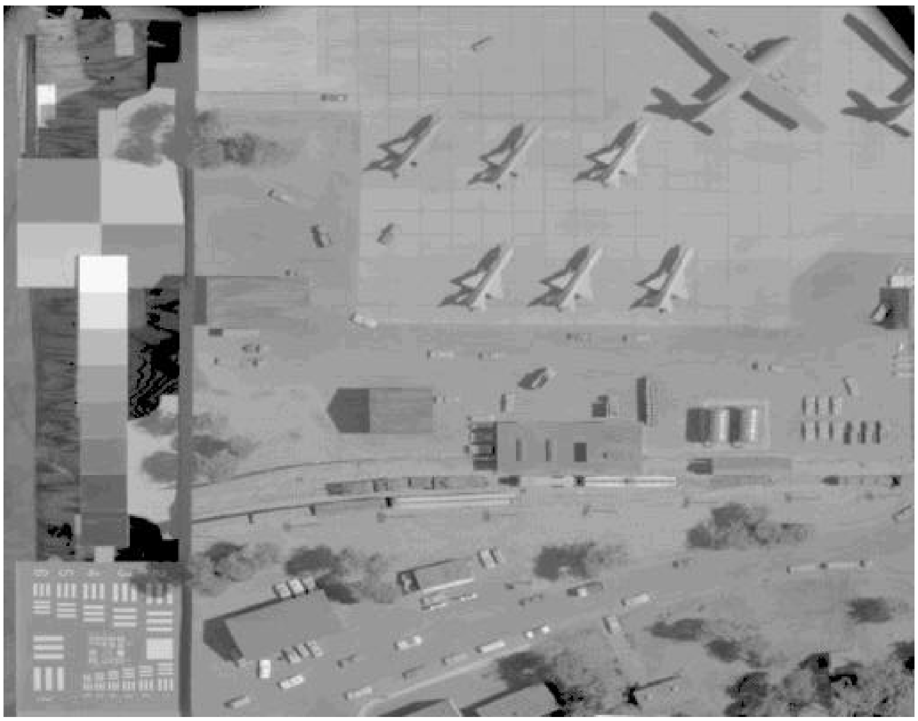
Original IR – 615,004 Bytes (8 minutes, 32 seconds)

DCT JPEG Q3 Setting – 23,734 Bytes (19 seconds)

The DCT JPEG Q3 setting attained a compression ratio of 25.9:1 (.31 bits per pixel) and the transmission time is still quite respectable.

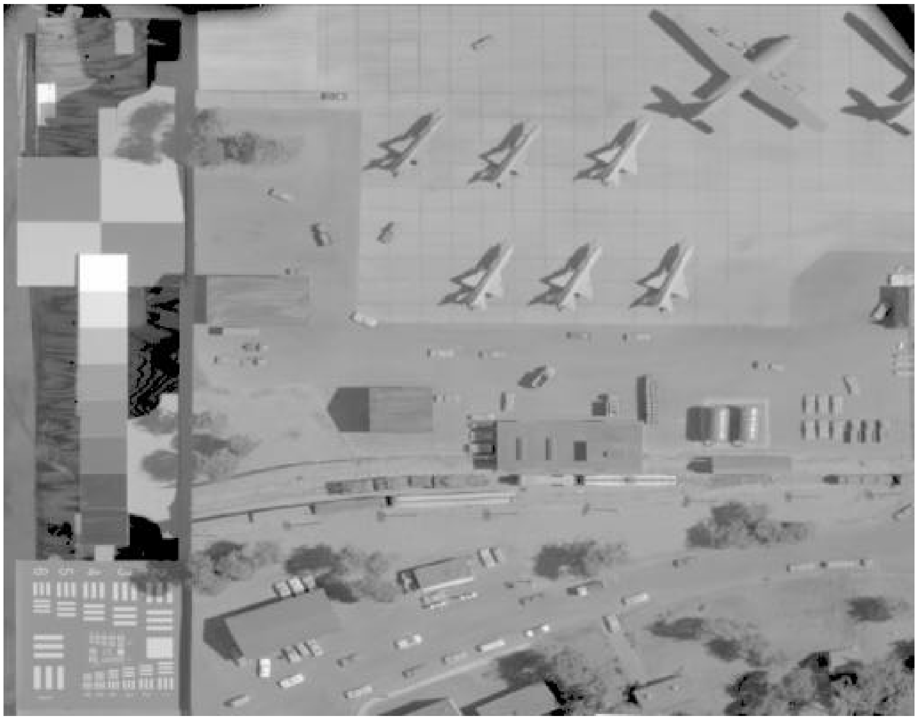


Original IR – 615,004 Bytes (8 minutes, 32 seconds)

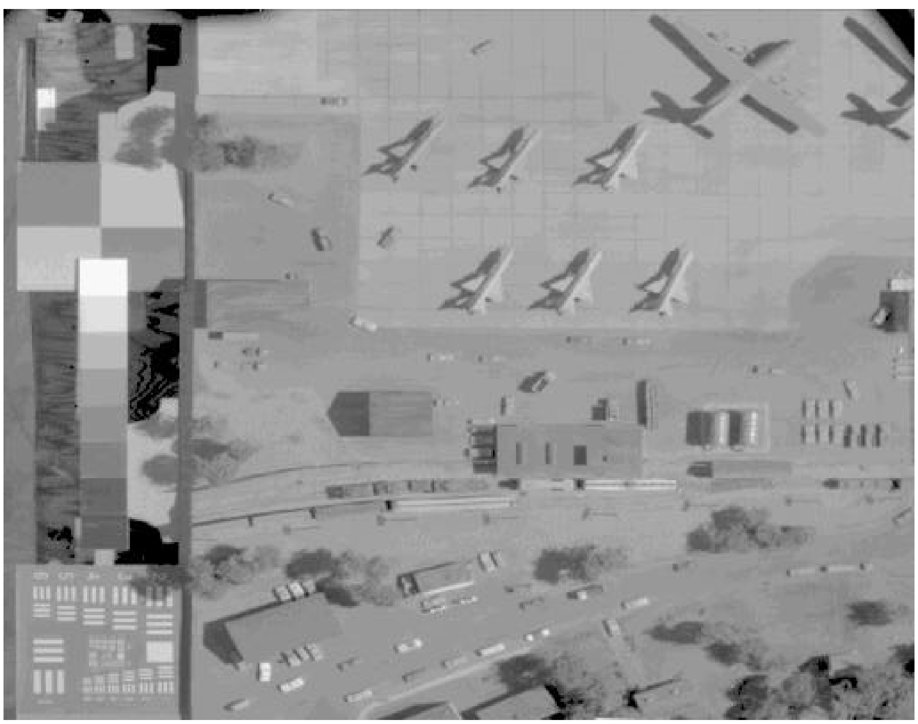


DCT JPEG Q4 Setting – 42,119 Bytes (35 seconds)

The DCT JPEG Q4 setting attained a compression ratio of 14.6:1 (.55 bits per pixel), which provided a modest increase in quality and only increased the transmission time by 16 seconds.

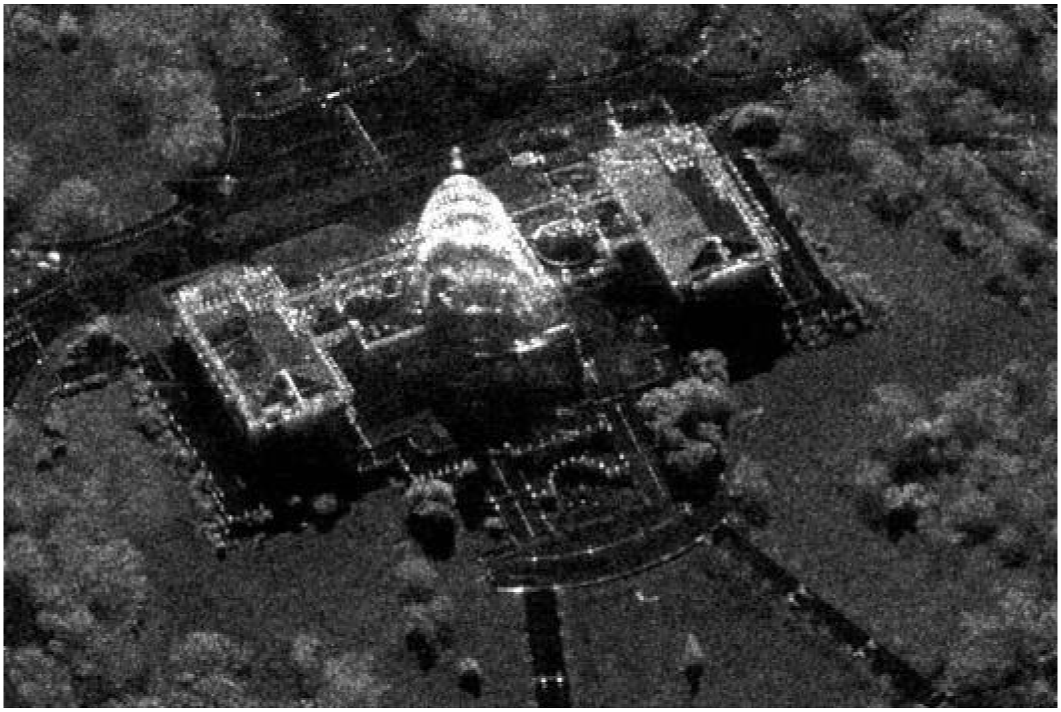


Original IR – 615,004 Bytes (8 minutes, 32 seconds)

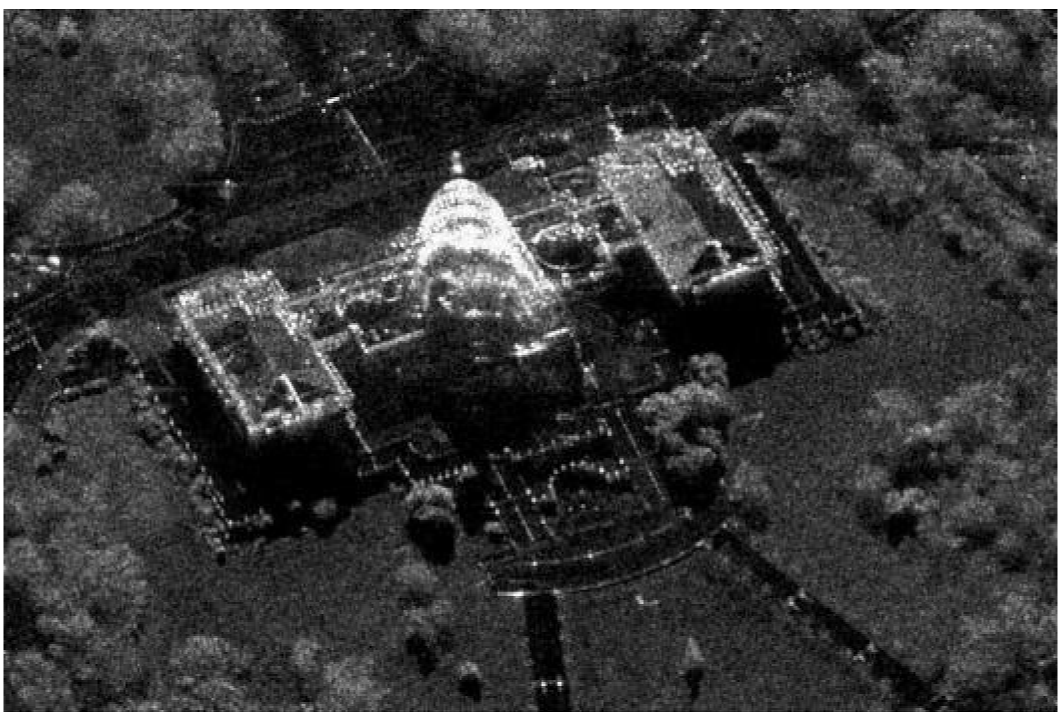


DCT JPEG Q5 Setting – 115,124 Bytes (1 minute, 35 seconds)

The highest DCT JPEG quality setting Q5, attained a compression ratio of 5.3:1 (1.5 bits per pixel). Transmission time is now just over a minute and a half.

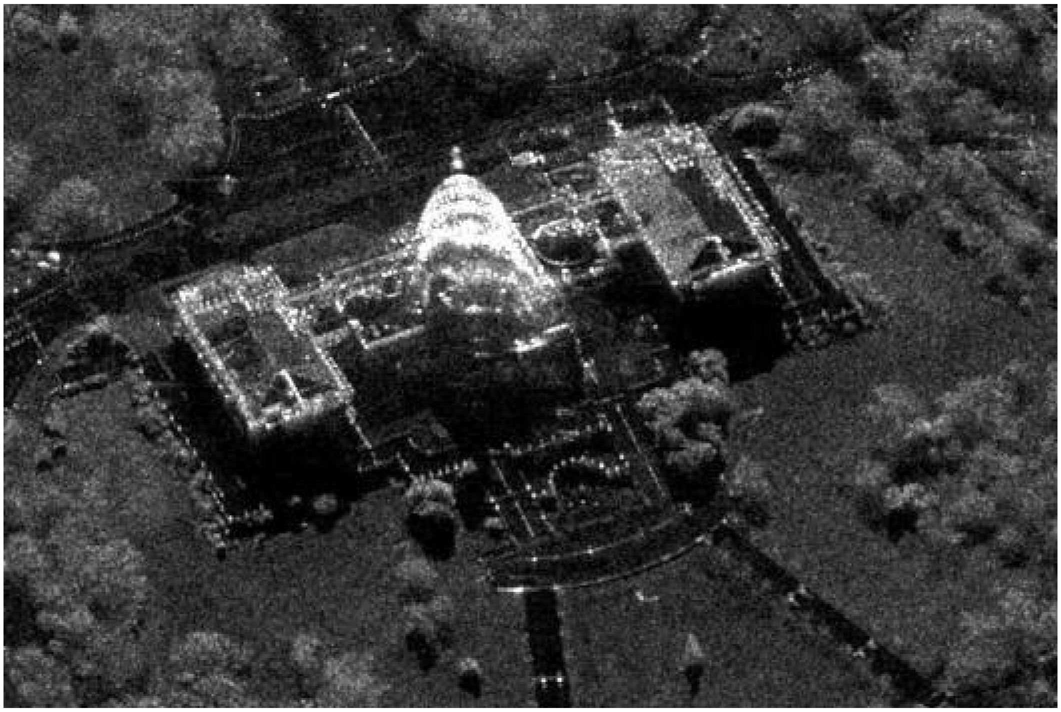


SAR Original – 961,522 Bytes (13 minutes, 21 seconds)



DCT JPEG Q1 Setting – 97,114 Bytes (1 minute, 20 seconds**)**

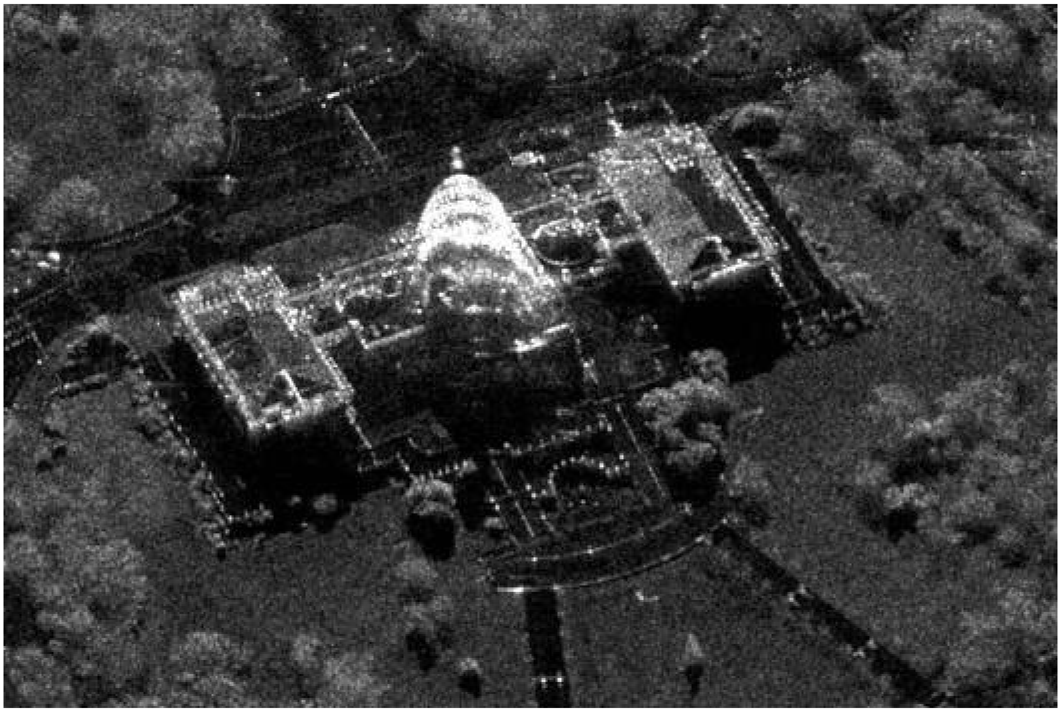
This time, on radar imagery, the Q1 setting attained a compression ratio of 9.9:1 (.81 bits per pixel and lowers the transmission time considerably. Radar imagery does not typically compress as well as other types of imagery.



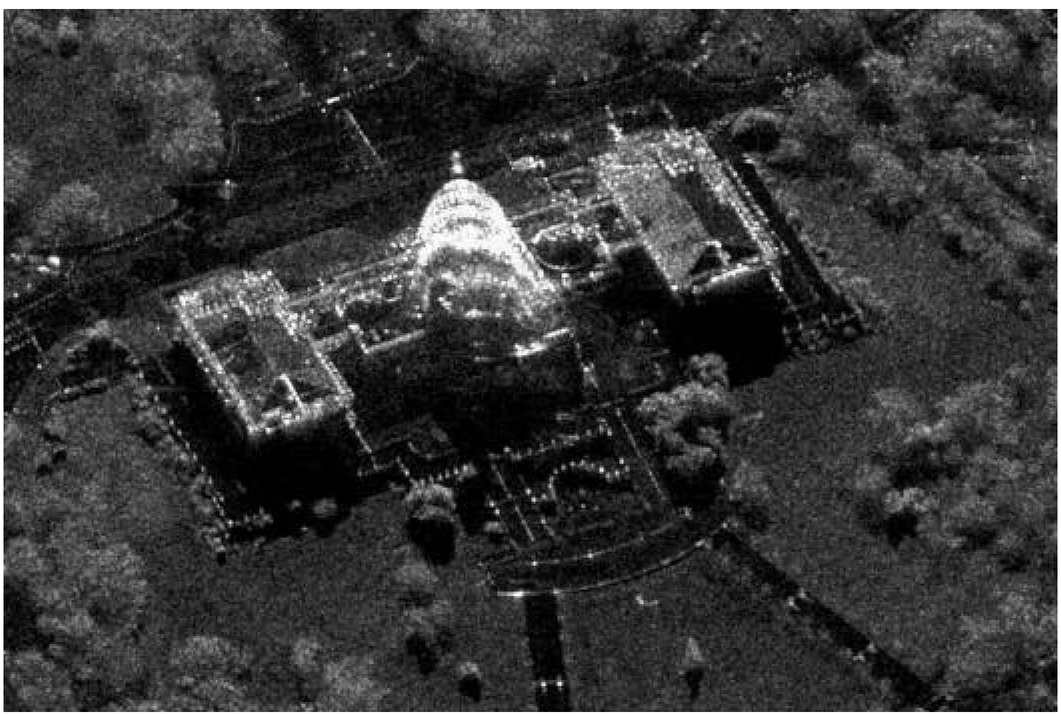
SAR Original – 961,522 Bytes (13 minutes, 21 seconds)

DCT JPEG Q2 Setting – 136, 584 Bytes (1 minute, 53 seconds)

DCT JPEG Q2 setting provided a compression ratio of 7:1 (1.14 bits per pixel) and only increased the transmission time by 33 seconds.

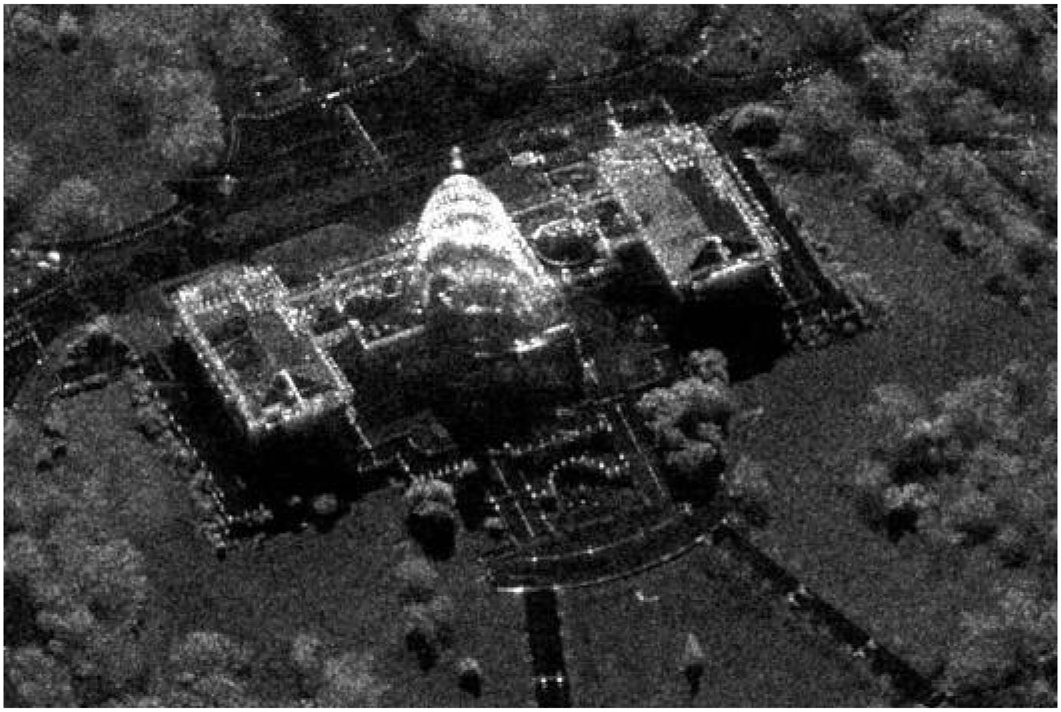


SAR Original – 961,522 Bytes (13 minutes, 21 seconds)

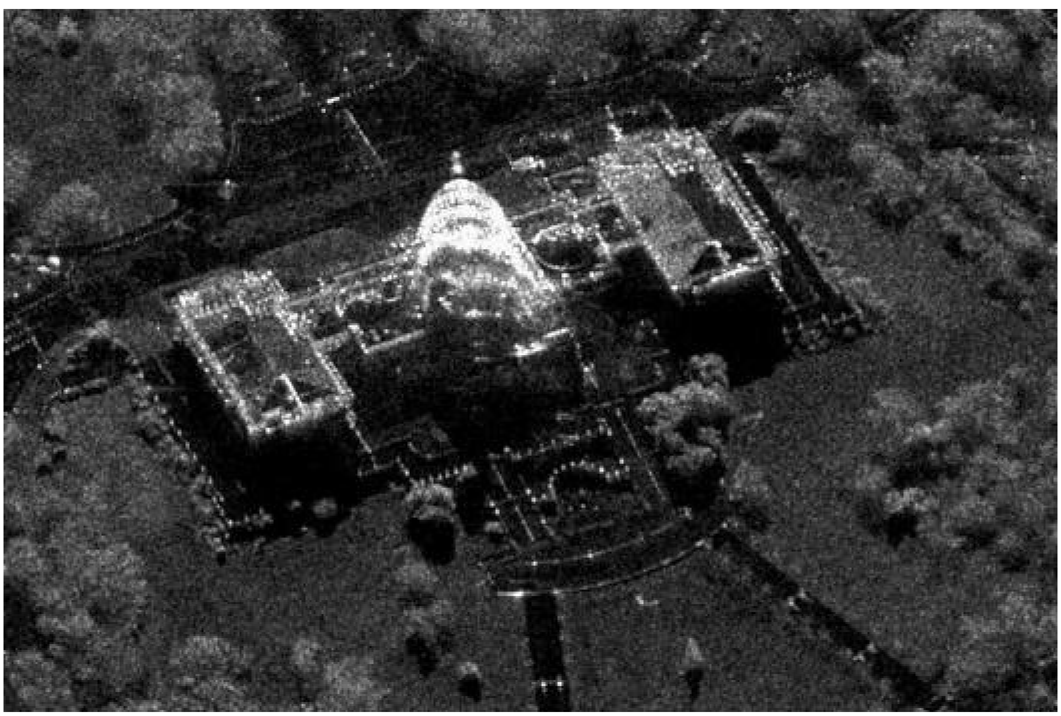


DCT JPEG Q3 Setting – 156,280 Bytes (2 minutes, 10 seconds)

DCT JPEG Q3 setting attained a compression ratio of 6:1 (1.3 bits per pixel) but increased our transmission time to more than 2 minutes at 9600 bits per second.

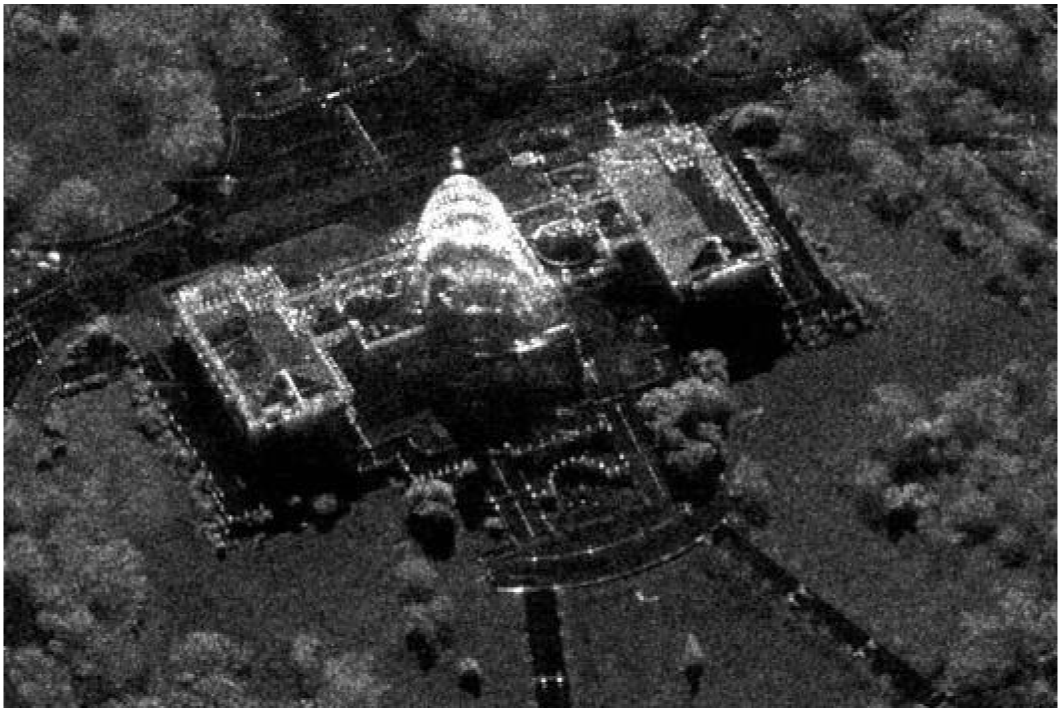


SAR Original – 961,522 Bytes (13 minutes, 21 seconds)

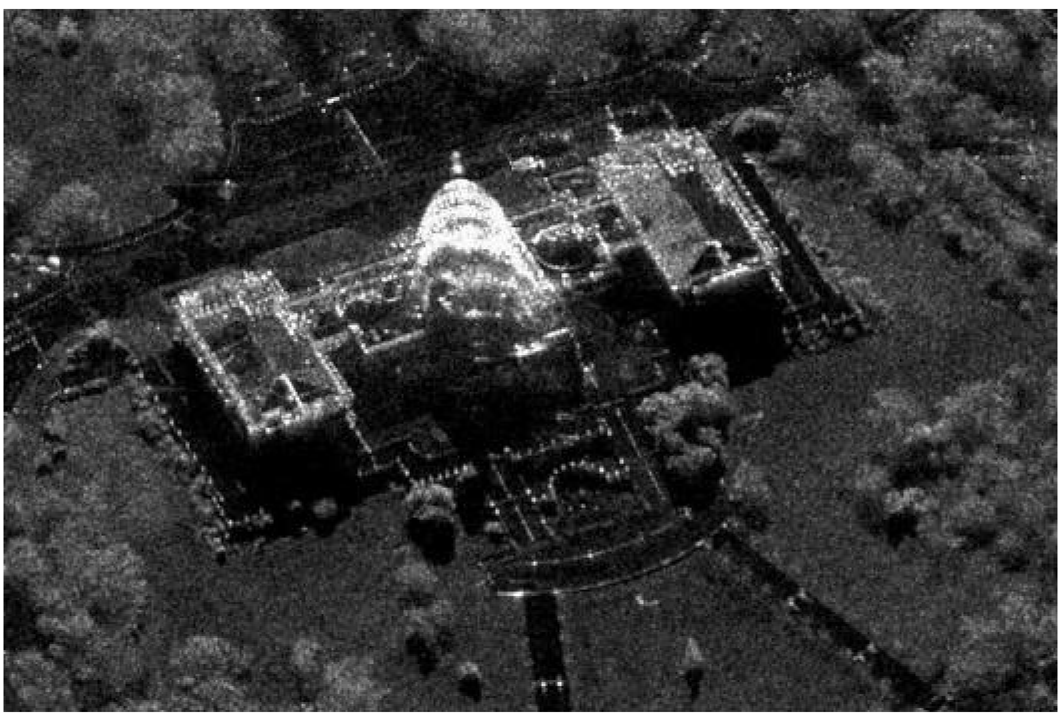


DCT JPEG Q4 Setting – 220,243 Bytes (3 minutes, 3 seconds)

DCT JPEG Q4 setting attained a compression ratio of 4.3:1 (1.83 bits per pixel) and adds almost a minute more to the transmission time.



SAR Original – 961,522 Bytes (13 minutes, 21 seconds)



DCT JPEG Q5 Setting – 445,734 Bytes (6 minutes, 11 seconds)

DCT JPEG Q5 setting attained a compression ratio of 2.1:1, (3.71 bits per pixel) but added 3 minutes to the transmission time. Users gain quality but increase transmission time.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

DCT JPEG Q1 Setting – 47,569 Bytes (39 seconds)

For the UAV video frame, the Q1 setting attained a compression ratio of 25.8:1 (.31 bits per pixel), and significantly lowers the transmission time.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

DCT JPEG Q2 Setting – 68,402 Bytes (57 seconds)

DCT JPEG Q2 setting provided a compression ratio of 17.9:1 (.44 bits per pixel) with only a modest increase in transmission time.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

**DCT JPEG Q3 Setting – 71,684 Bytes (59 seconds)**

This setting, Q3, attained a 17:1 compression ratio (.47 bits per pixel) which takes just under a minute to transmit or receive at 9600 bits per second.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

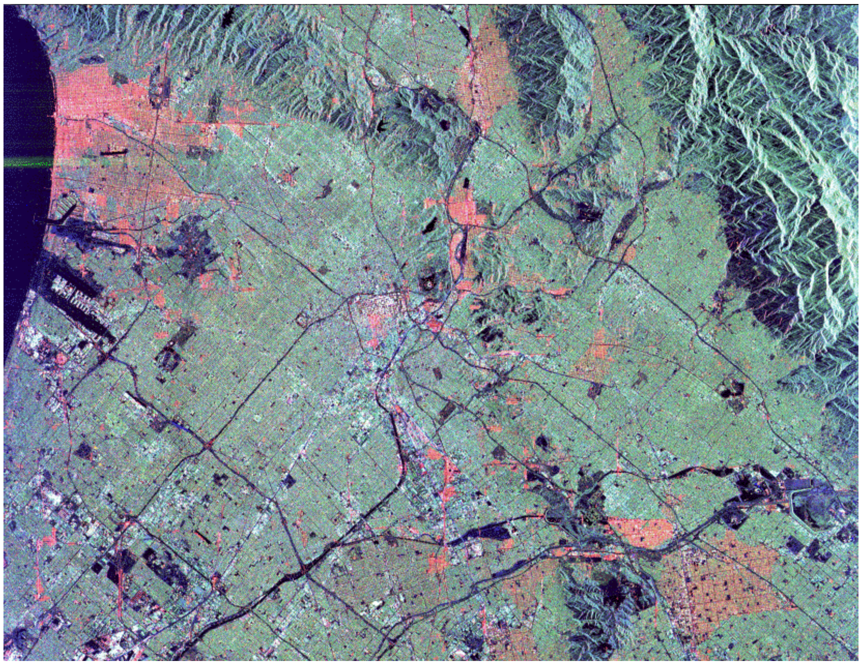
DCT JPEG Q4 Setting – 141,256 Bytes (1 minute, 57 seconds)

The compression ratio is 8.7:1 (.92 bits per pixel) and the required transmission time is now almost 2 minutes at the DCT JPEG Q4 setting.

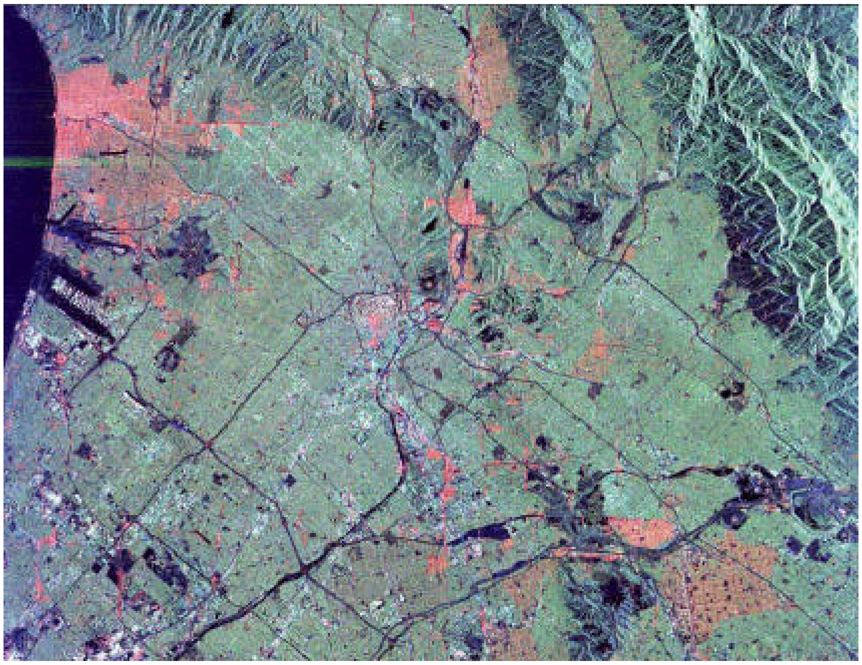
UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

DCT JPEG Q5 Setting - 481,348 Bytes (6 minutes, 41 seconds)

At the DCT JPEG Q5 setting, the compression ratio is 2.5:1 (.03 bits per pixel) but the transmission time is more than some people prefer.

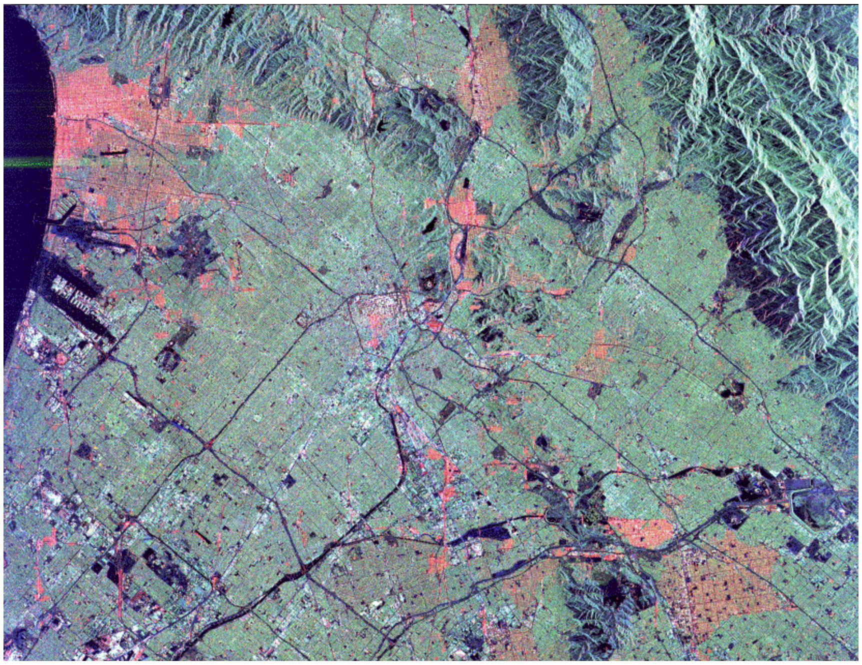


Multi-Spectral Original – 1,572,193 Bytes (21 minutes, 50 seconds)

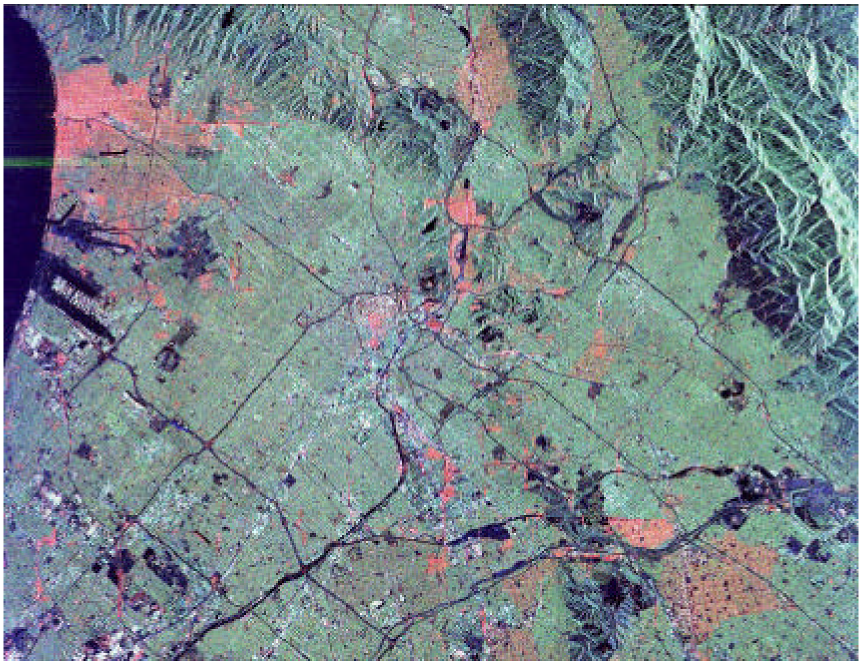


DCT JPEG Q1 Setting – 91,500 Bytes (1 minute, 16 seconds)

DCT JPEG Q1 setting produced a compression ratio of 17:1 (3.13 bits per pixel) and significantly lowered the transmission time.

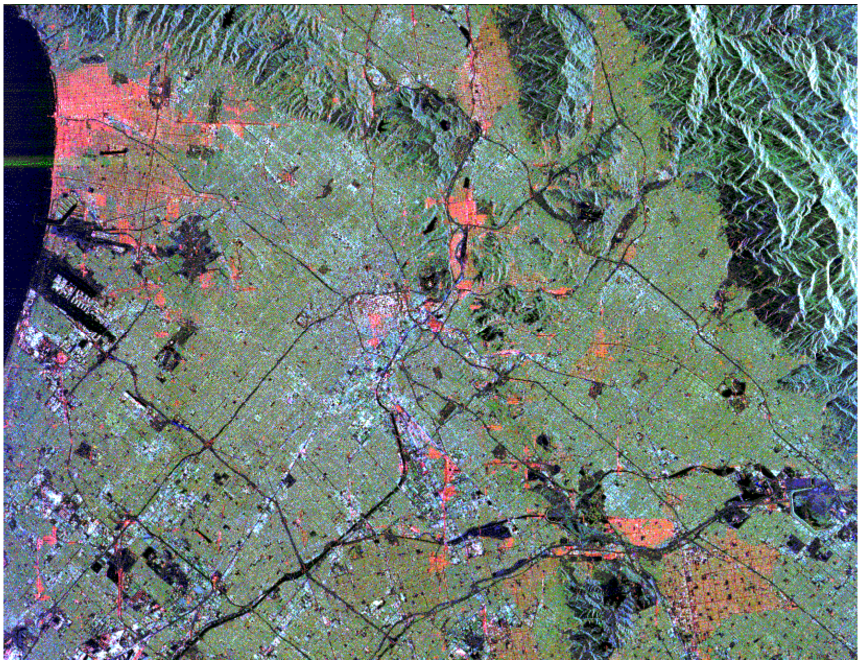


Multi-Spectral Original – 1,572,193 Bytes (21 minutes, 50 seconds)

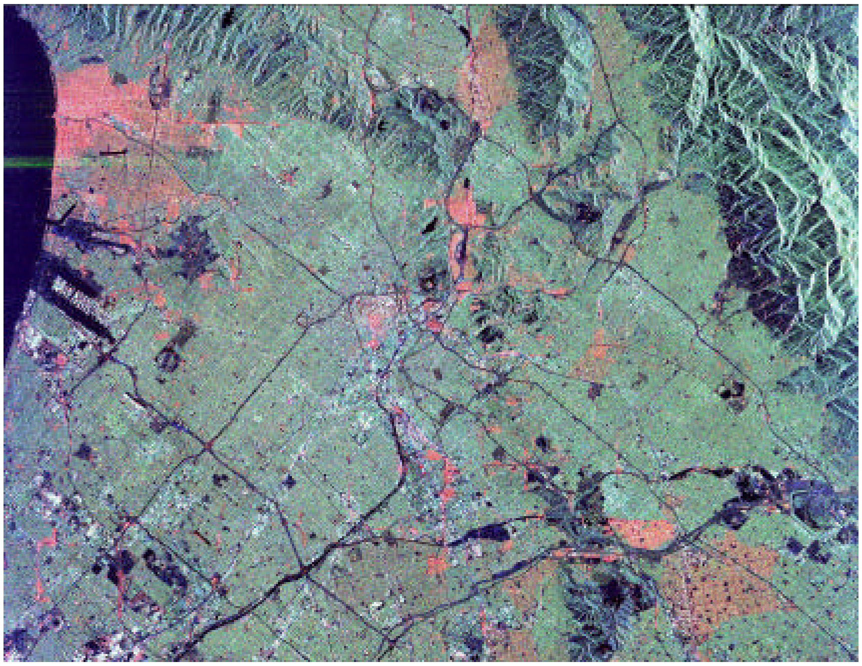


DCT JPEG Q2 Setting – 149, 507 Bytes (2 minutes, 4 seconds)

DCT JPEG Q2 setting provided a compression ratio of 10.5:1 (2.28 bits per pixel) and increases the transmission time by almost a minute.

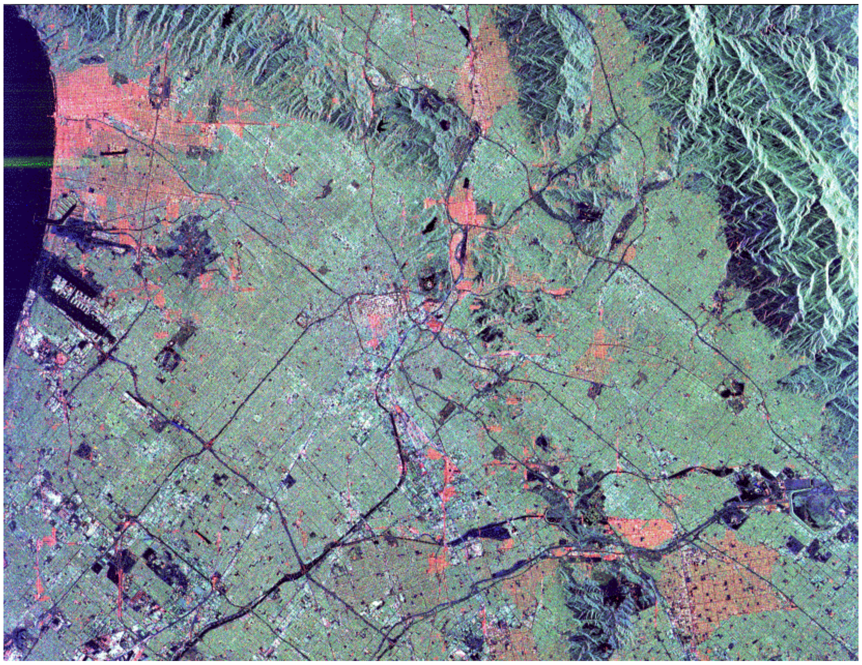


Multi-Spectral Original – 1,572,193 Bytes (21 minutes, 50 seconds)

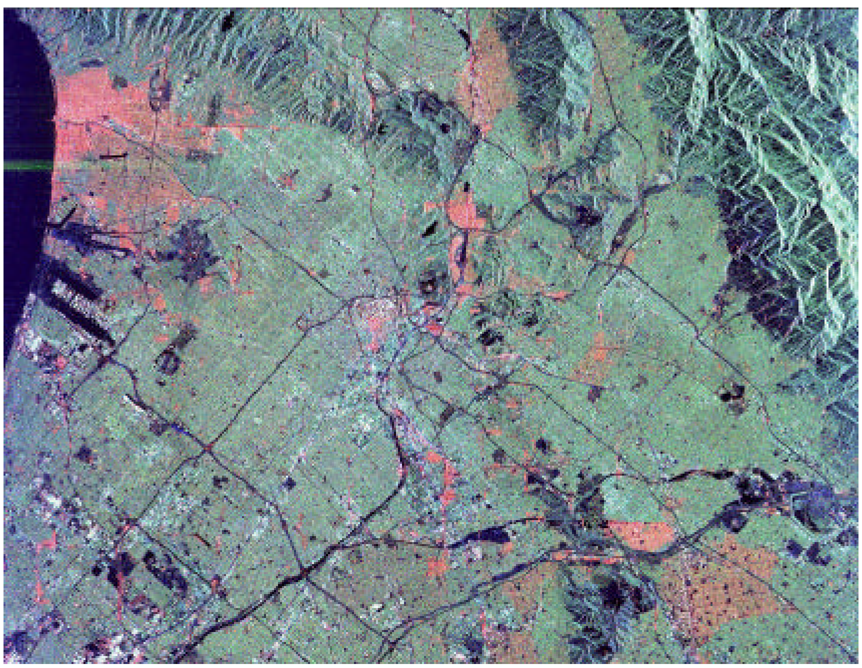


DCT JPEG Q3 Setting – 259,577 Bytes (3 minutes 36 seconds)

DCT JPEG Q3 setting provided a compression ratio of 6:1 (3.96 bits per pixel) with the corresponding increase in time.

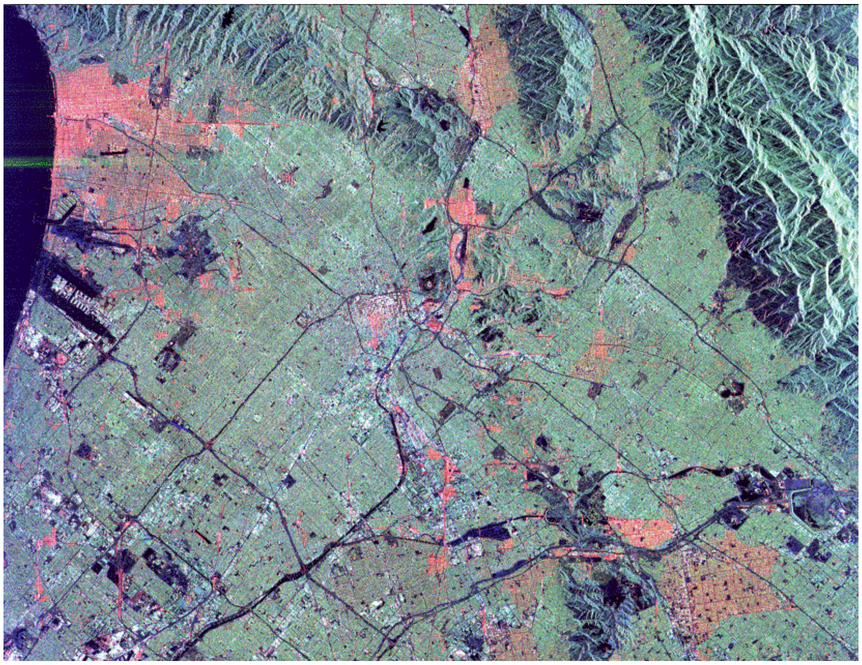


Multi-Spectral Original – 1,572,193 Bytes (21 minutes, 50 seconds)

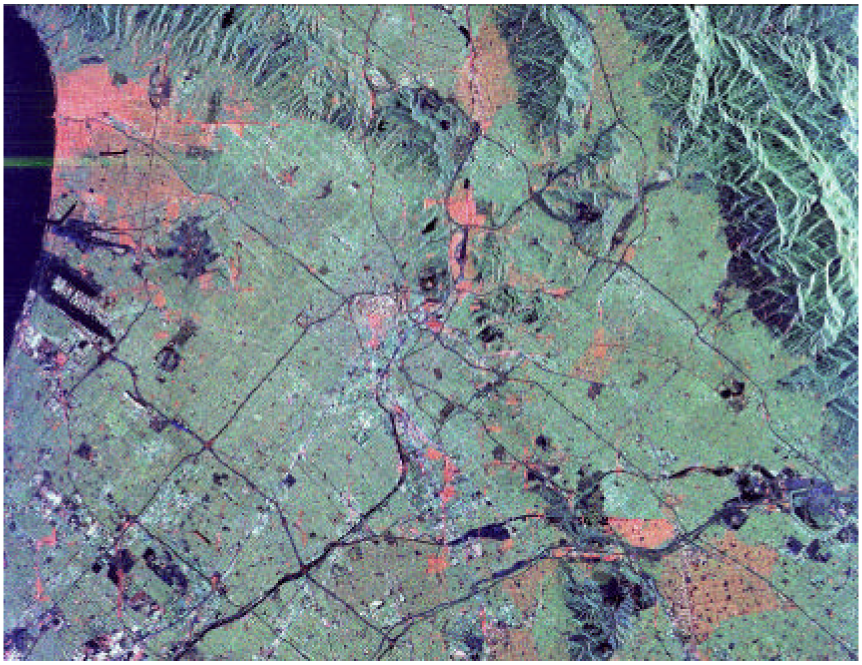


DCT JPEG Q4 Setting – 409,620 Bytes (5 minutes, 41 seconds)

The Q4 setting compression ratio is 5.4:1 (6.25 bits per pixel), a modest increase in quality results in almost two more minutes of transmission time.



Multi-Spectral Original – 1,572,193 Bytes (21 minutes, 50 seconds)



DCT JPEG Q5 Setting – 747,728 Bytes (10 minutes, 23 seconds)

At DCT JPEG Q5 setting the compression ratio is 2.1:1 (11.41 bits per pixel) and now will require over ten minutes to transmit at 9600 bits per second.

Now lets review how the NITF DCT JPEG compression algorithms performed and summarize the results. Users should have noticed that the same setting produced different compression ratios on different types of images. This clearly demonstrates the adaptive nature of DCT JPEG; it adapts to each image based upon their content, thus producing different compression ratios. Users should have noticed that color for the most part compressed well since there is a lot of redundancy in color. Users should have also noticed that though the images were of different sizes, even small images could take a long time to transmit.

TABLE 1: DCT JPEG RESULTS SUMMARY

| Image | Original | Q1 | Q2 | Q3 | Q4 | Q5 |
|---|---|---|---|---|---|---|
| Color | 1:1<br>12:11 | 43:1<br>00:13 | 34:1<br>00::15: | 16:1<br>00:36 | 12.6:1<br>00:44 | 8.7:1<br>01:04 |
| Visual | 1:1<br>58:18 | 13.9:10<br>04:10 | 11.8:1<br>04:54 | 8:1<br>07:14 | 7:1<br>08:16 | 5.4:1<br>10:37 |
| IR | 1:1<br>08:32 | 43.6:1<br>00:11 | 33.9:1<br>00:15 | 25/9:1<br>00:19 | 14.6:1<br>00:35 | 5:1<br>01:35 |
| SAR | 1:1<br>13:21 | 9.9:1<br>01:20 | 7:1<br>01:53 | 6:1<br>02:10 | 4:1<br>03:03 | 2:1<br>06:11 |
| UAV | 1:1<br>17:05 | 25:1<br>00:39 | 17.9:1<br>00:57 | 17:1<br>00:59 | 8.7:1<br>01:57 | 2.5<br>06:41 |
| Multi Spectral | 1:1<br>21:50 | 17:1<br>01:16 | 10.5:1<br>02:04 | 6:1<br>03:36 | 3.8:1<br>05:41 | 8.7:1<br>10:23 |
| Compression Ratio<br>Time to Transmit (mm:ss @ 9600 Bits per Second) | | | | | | |

If we study the table, we quickly see that the higher the quality setting, the greater the transmission time. The table shows that the compression ratio achieved varies depending on the image. Also notice that any given Q setting produced different compression ratios from image to image.

Note that the 4-megabyte visual sample, (the average tactical size image) still takes minutes to transmit even at the lowest quality setting. In order to shorten this transmission time we must decrease the file size even more. We will use some other lossy compression algorithm that provides much higher compression ratios. We pay a price though, the higher the compression the greater the quality loss. Users should consider this as they go to the next NITF JPEG compression variant.

### 3.3 DOWNSAMPLE JPEG (DS JPEG)

What is Downsample JPEG? This compression technique uses essentially the same DCT JPEG compression applied to an image that has been downsampled. What this means is that the image size has been reduced using downsampling which throws away image data. This process is illustrated in the figure, the original 2048 X 2048 size is downsampled to 1024 X 1024 and then to 512 X 512. In this process the algorithm would selectively throw away image data in both the X and Y-axis to reach the next smaller file size, until the smallest is reached. In the reverse process or upsampling, the image is upsampled to restore it to its original size. This is also accomplished in steps and the information that was discarded is replaced selectively by interpolation. The image data that was thrown away is replaced by filling in values of the discarded data. The small 512 X 512 box in the center of the figure is what is compressed by JPEG, and this combination of using downsampling and then compression achieves much greater compression ratios, yet surprisingly still provides good quality imagery. At present, this technique has not been applied to color images. This technique also has five Q settings, with Q1 providing the greatest compression and Q5 the least compression. Quality loss again is greatest at the Q1 setting and the least at Q5. The Q4 setting in Downsample JPEG is also known as NIMA Method 4 Compression.

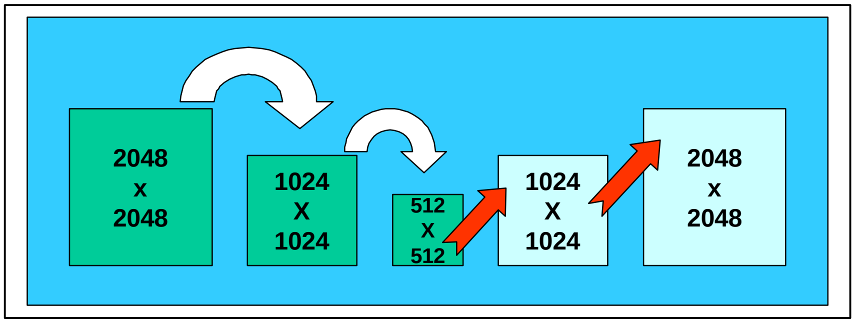



FIGURE 13: DOWNSAMPLE AND UPSAMPLE PROCESS

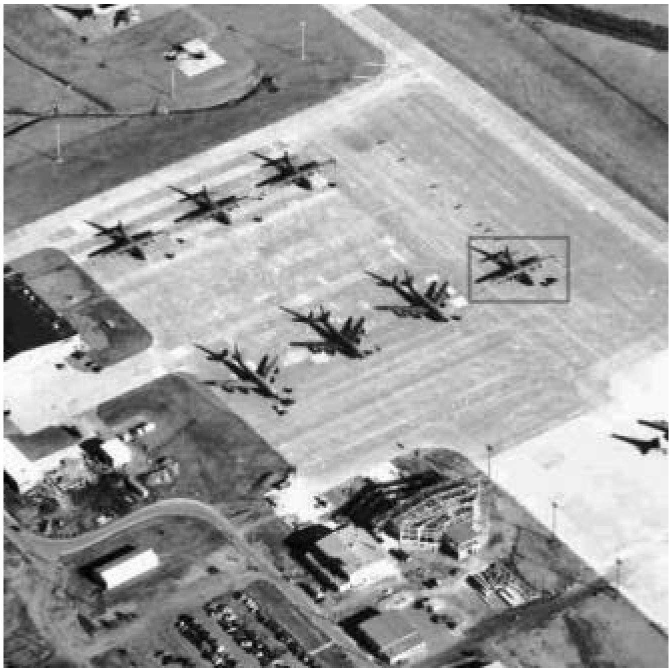
Now let's proceed and see how this technique performs. Again users should consider the time and quality factors as they proceed through Downsample JPEG. Consider a requestor who only has one minute to receive an image at a very slow receipt rate.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)

DS JPEG Q1 Setting – 15,853 Bytes (13 seconds)

This setting, DS JPEG Q1, provided 264:1 (.03 bits per pixel) and we can transmit it in less than 15 seconds. Consider the quality compared to the time saving.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)



DS JPEG Q2 Setting – 25,123 Bytes (20 seconds)

At 167:1 (.04 bits per pixel) we pick up some quality, and are still capable of transmitting the image in under 30 seconds.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)

DS JPEG Q3 Setting – 35,313 Bytes (29 seconds)

This setting, Q3, provided 118:1 compression (.07 bits per pixel). The transmission time is still short, note the quality.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)

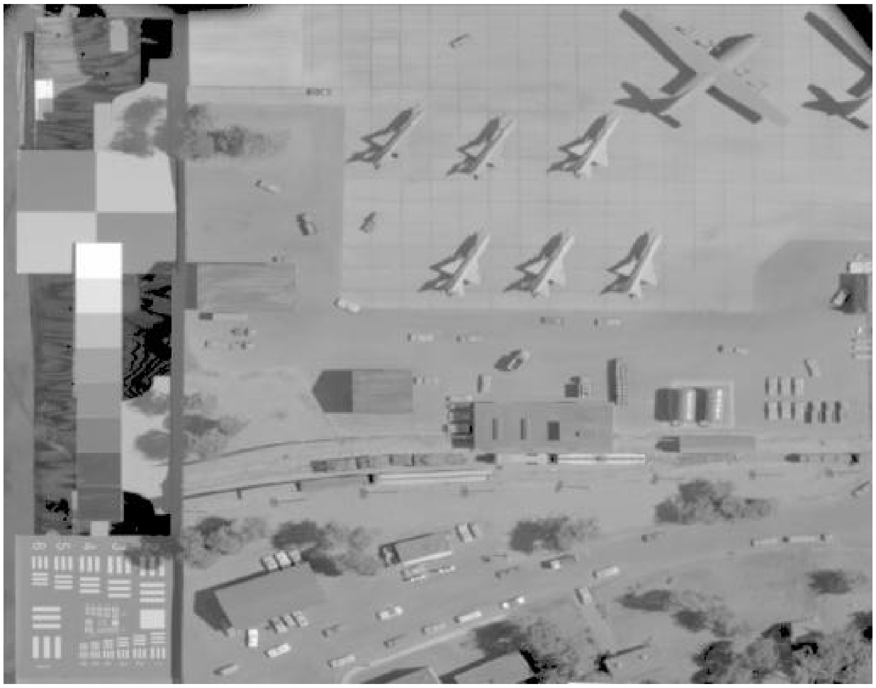
DS JPEG Q4 Setting – 51,378 Bytes (42 seconds)

This setting provided and 81:1 (.10 bits per pixel) compression ratio, but the transmit time is still under a minute. This is the setting that is known as NIMA Method 4.

Visual Original – 4,197,074 Bytes (58 minutes 18 seconds)

DS JPEG Q5 Setting – 138,144 Bytes (1 minute, 55 seconds)

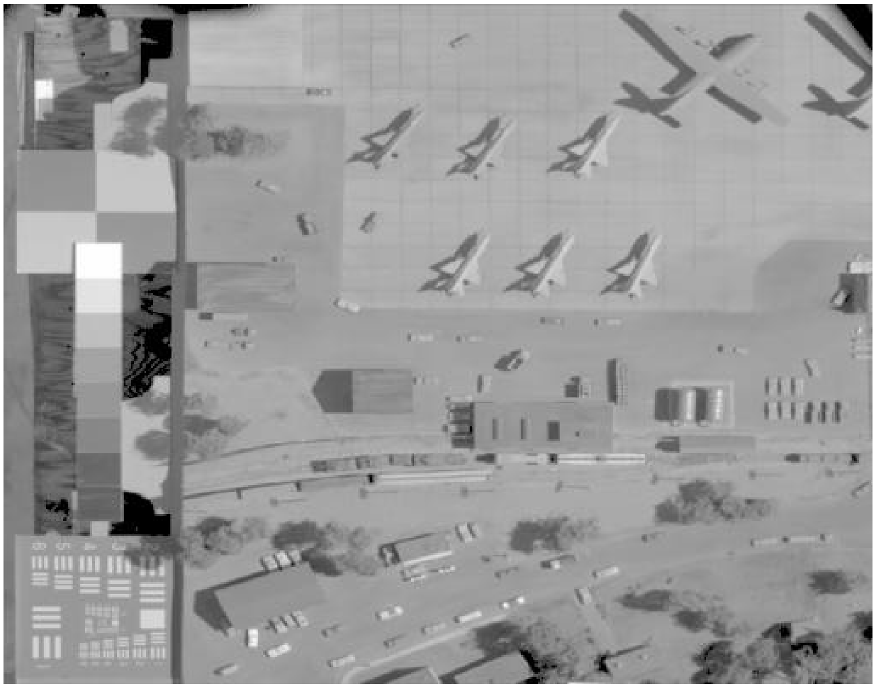
At this setting we achieved the highest quality at a 30:1 (.26 bits per pixel) compression ratio. The image now takes almost 2 minutes to transmit, which is still a vast improvement when compared to the time for the original.



Original IR – 615,004 Bytes (8 minutes, 32 seconds)

DS JPEG Q1 Setting 2356 Bytes (1 second)

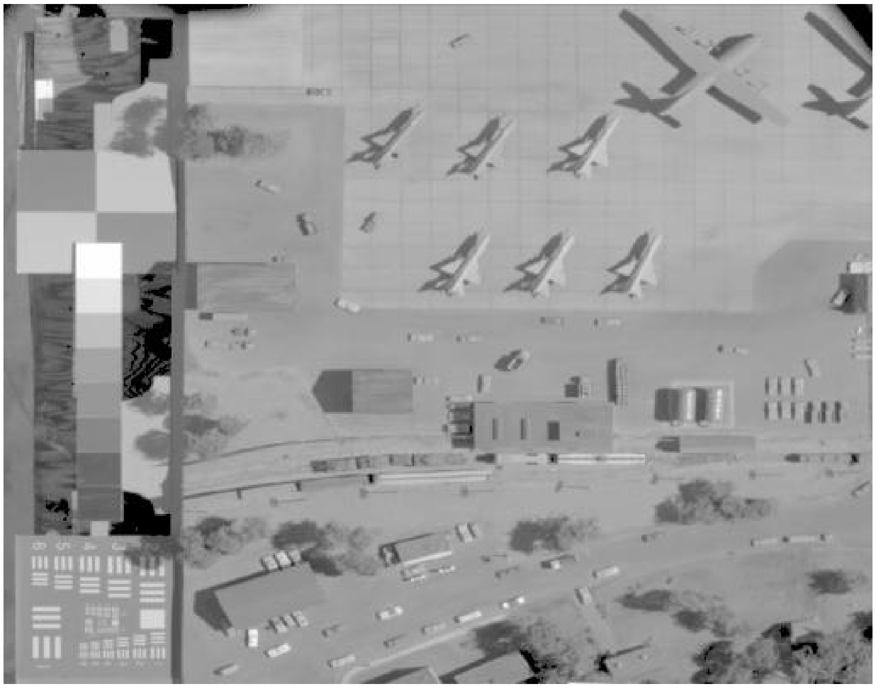
On IR, DS JPEG at this setting produced an unusable image. This setting, Q1, provided 261:1 (.03 bits per pixel) compression ratio. Users would not use this setting on this particular image.



Original IR – 615,004 Bytes (8 minutes, 32 seconds)

DS JPEG Q2 Setting – 3,099 Bytes (2 seconds)

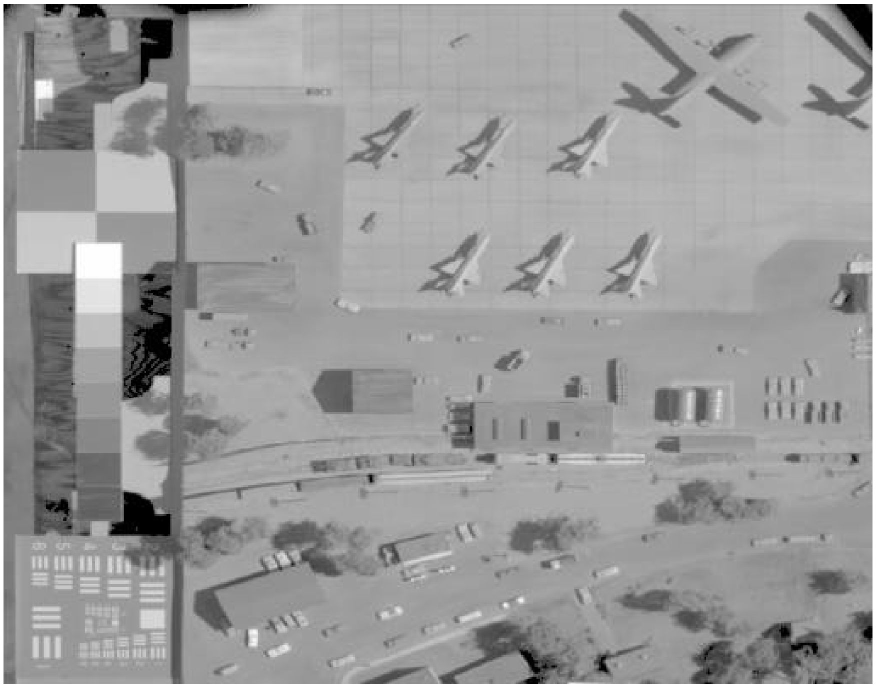
At the DS JPEG Q2 setting the image quality has improved, but still is not usable.  This setting provided a compression ratio of 198:1 (.04 bits per pixel).



Original IR – 615,004 Bytes (8 minutes, 32 seconds)

DS JPEG Q3 Setting – 3,718 Bytes (3 seconds)

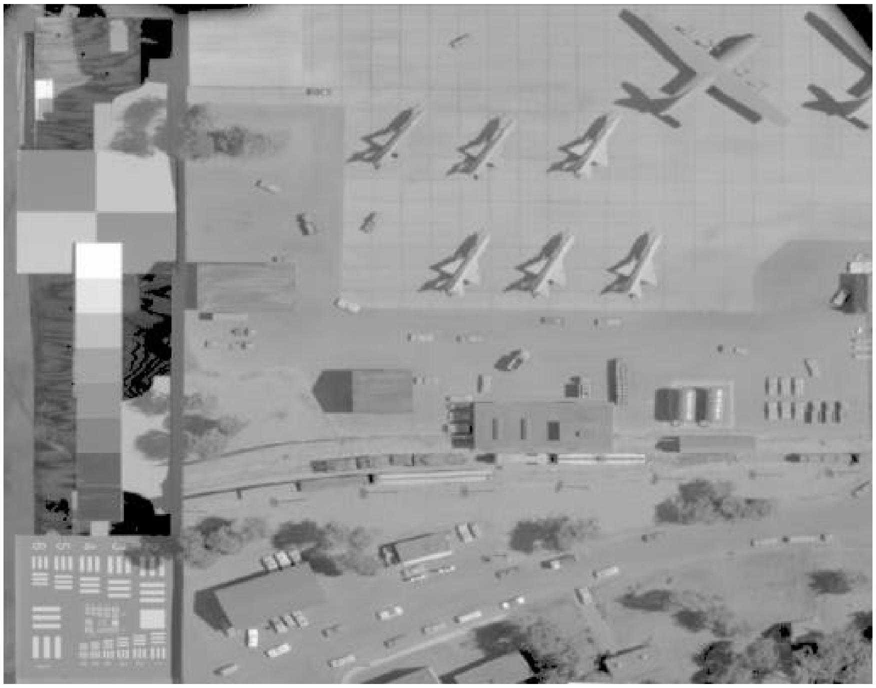
DS JPEG Q3 produced a compression ratio of 165:1 (.05 bits per pixel). This image is still not usable.



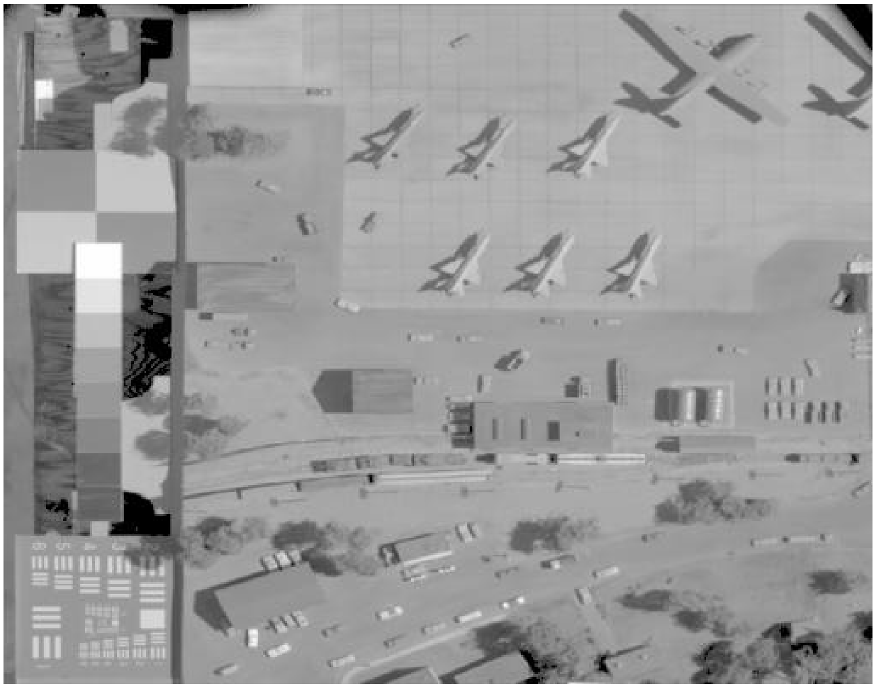
Original IR – 615,004 Bytes (8 minutes, 32 seconds)

DS JPEG Q4 Setting – 4,799 Bytes (3 seconds)

DS JPEG provided and 128:1 (.06 bits per pixel) compression ratio at the Q4 setting. Remember that this setting is known as NIMA Method 4.

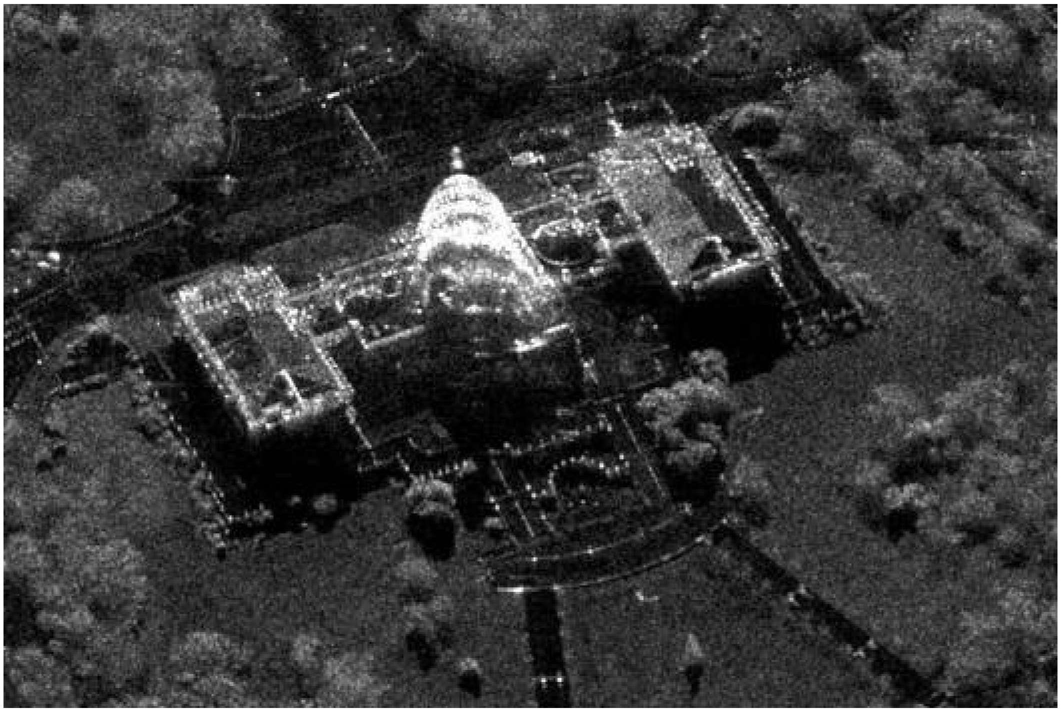


Original IR – 615,004 Bytes (8 minutes, 32 seconds)



DS JPEG Q5 Setting – 11, 918 Bytes (9 seconds)

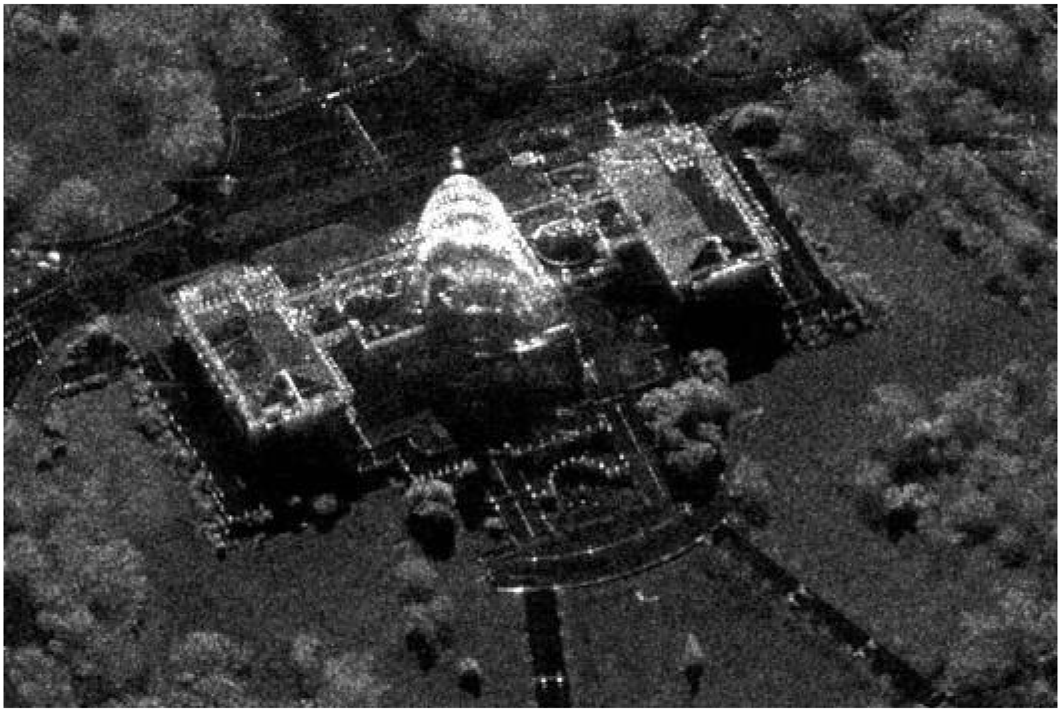
This setting provided a compression ratio of 51:1 (.16 bits per pixel). This series demonstrates the adaptive characteristics of DS JPEG, the quality improved at each successive setting. When users use this compression scheme try different settings and be sure to look at the results.



SAR Original – 961,522 Bytes (13 minutes, 21 seconds)

DS JPEG Q1 Setting – 6,463 Bytes (5 seconds)

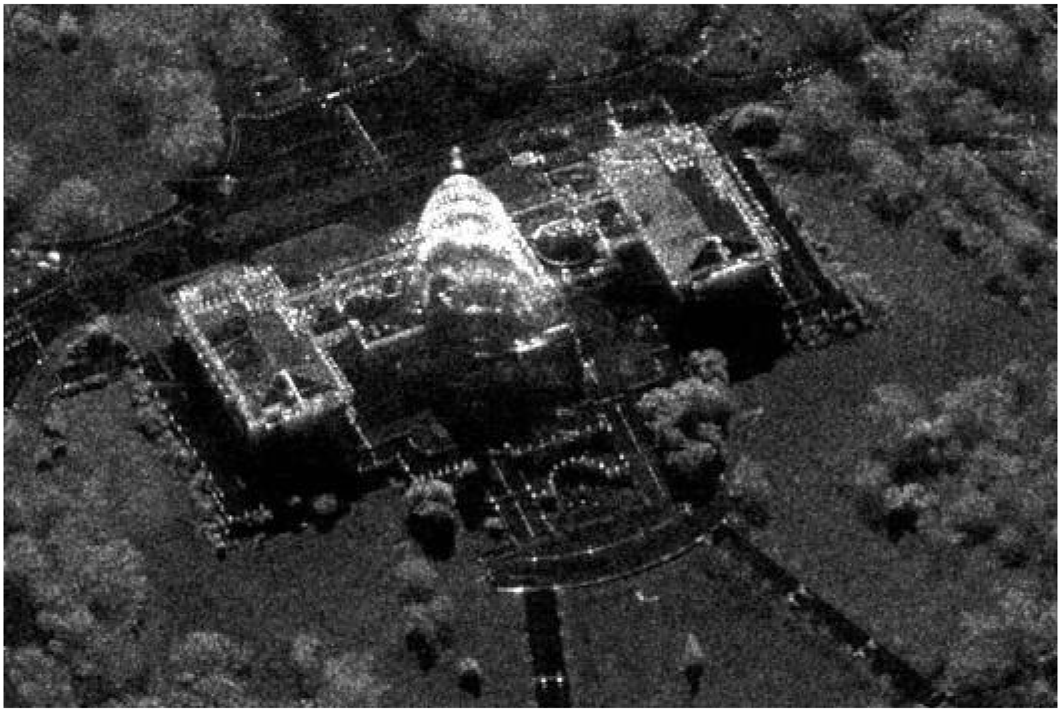
At this setting a compression ratio of 148:1 (.05 bits per pixel) was attained, though the quality is probably not acceptable.



SAR Original – 961,522 Bytes (13 minutes, 21 seconds)

DS JPEG Q2 Setting – 10,796 Bytes (8 seconds)

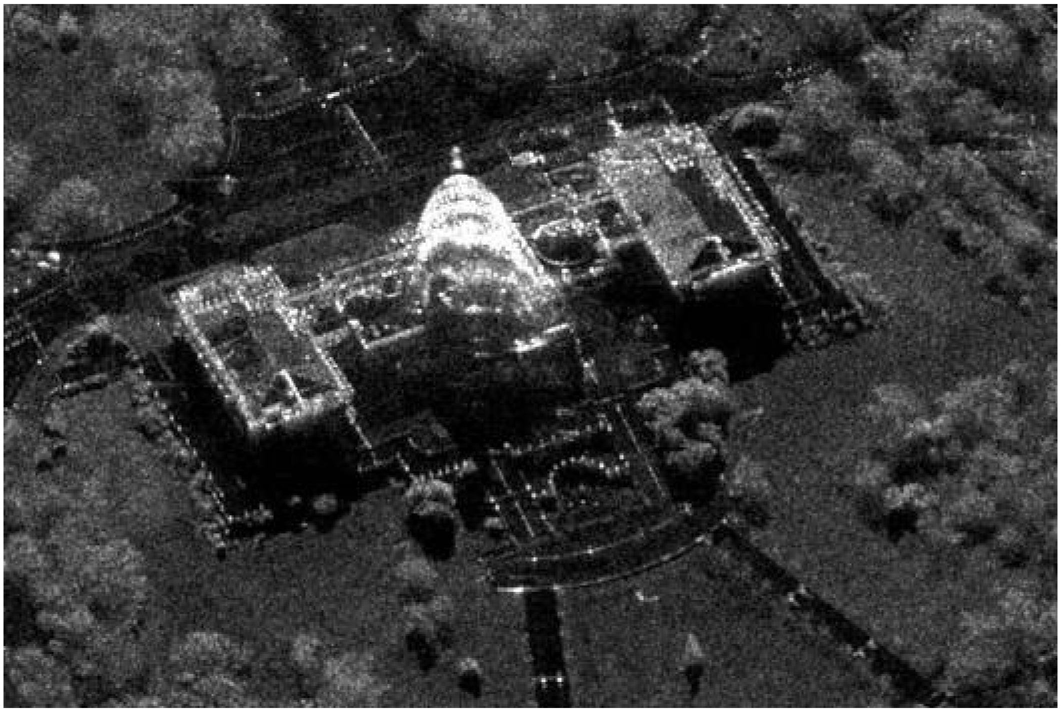
This setting, DS JPEG Q2, provided increased quality at a compression ratio of 198:1 (.09 bits per pixel). The transmission time is only 8 seconds.



SAR Original – 961,522 Bytes (13 minutes, 21 seconds)

DS JPEG Q3 Setting – 15,508 Bytes (12 seconds)

Note the increase in quality. This setting produced a 62:1 (.13 bits per pixel) compression ratio. This file can be transmitted very quickly.



SAR Original – 961,522 Bytes (13 minutes, 21 seconds)

DS JPEG Q4 Setting – 23,528 Bytes (19 seconds)

At this setting, a compression ratio of 41:1 (.20 bits per pixel) was produced. The transmission time is still very quick and the quality is probably sufficient to satisfy most users.

SAR Original – 961,522 Bytes (13 minutes, 21 seconds)

DS JPEG Q5 Setting – 61,848 Bytes (51 seconds)

A compression ratio of 15:1 (.51 bits per pixel) was produced at our highest quality setting, Q5. The transmission time is just under a minute. This series of radar images also demonstrates the adaptability of DS JPEG, though not as striking as the infrared series.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

DS JPEG Q1 Setting – 4,688 Bytes (3 seconds)

On the UAV (video frames converted to 8 bit grayscale) the DS JPEG Q1 setting produced a 262:1 (.03 bits per pixel) compression ratio. The quality is surprisingly good at this rate of compression, and the transmission time is negligible, 3 seconds.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

DS JPEG Q2 Setting – 6,461 Bytes (5 seconds)

DS JPEG Q2 setting resulted in a 190:1 (.04 bits per pixel) compression ratio the quality is even better. Again, the adaptive characteristic of this compression algorithm is being demonstrated. The IR quality at this setting was unacceptable, while the results on this video frame are quite good.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

DS JPEG Q3 Setting – 8,730 Bytes (7 seconds)

This setting, DS JPEG Q3, provided a compression ratio of 140:1 (.06 bits per pixel). The quality is excellent.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

DS JPEG Q4 Setting – 11,820 Bytes (9 seconds)

At 104:1 (.08 bits per pixel) compression, quality and time differences must be considered. Since this setting only adds 2 seconds to the transmission time, users probably would use this setting for this image.

UAV Original – 1,230,482 Bytes (17 minutes, 5 seconds)

DS JPEG Q5 Setting – 32,200 Bytes (26 seconds)

This setting, Q5, provided a compression ratio of 38:1 (.21 bits per pixel). The quality is the best we can achieve using DS JPEG, note however that the transmission time is now just under 30 seconds, 15 more seconds than the Q4 setting.

Let's review DS JPEG. First, we know that it can only be applied to grayscale images. Second, DS JPEG is highly adaptive, it worked extremely well on grayscale and the UAV video frames, not as well on the radar images, and only at higher Q settings for the infrared images. Table 2 summarizes the results for DS JPEG.

TABLE 2: DS JPEG RESULTS SUMMARY

| Image | Original | Q1 | Q2 | Q3 | Q4 | Q5 |
|---|---|---|---|---|---|---|
| Visual | 1:1<br>58:18 | 264:1<br>00:13 | 167:1<br>00:20 | 118:1<br>00:29 | 891:1<br>00:42 | 30:1<br>01:55 |
| IR | 1:1<br>08:32 | 261:1<br>00:01 | 198:1<br>00:02 | 165:1<br>00:03 | 128:1<br>00:03 | 51:1<br>00:09 |
| SAR | 1:1<br>13:21 | 148:1<br>00:05 | 89:1<br>00:08 | 62:1<br>00:13 | 41:1<br>00:19 | 15:1<br>00:51 |
| UAV | 1:1<br>17:05 | 262:1<br>00:03 | 190:1<br>00:05 | 140:1<br>00:07 | 104:1<br>00:09 | 38:1<br>00:26 |
| Compression Ratio<br>Time to Transmit (mm:ss @ 9600 Bits per Second) | | | | | | |

If users study the table they will discover that again none of the images were compressed at exactly the same rate for a given Q setting. Notice that the compression ratios range from 264:1 to 15:1, which highlights the fact that each type and individual image compresses uniquely. The column highlighted in yellow is the Q setting that correlates to NIMA Method 4 which was the catalyst for the implementation of DS JPEG. It is also readily apparent that DS JPEG dramatically reduces the time required to transmit an image. This probably makes it the compression of choice if the recipient only has a 9600 bits per second modem or radio.

### 3.4 LOSSLESS JPEG

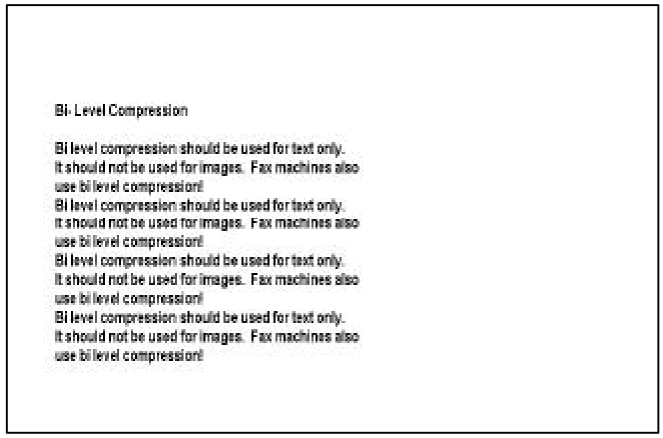
Before we leave JPEG, there is one other variant within NITF and that is Lossless JPEG. Lossless consists of those techniques guaranteed to generate an exact duplicate of the input data stream (or original image) after the data has been compressed and then decompressed back into its original form. No differences can be found. Lossless compression rates are very small and there will be no significant reduction in file sizes for images. Compression rates are on the order of 2 or 2.5:1. This handbook does not illustrate lossless compression, since users would be looking at two identical images. Lossless compression techniques are also used on textual media, and provide higher compression rates than for imagery. Users may experiment with lossless compression using either the sample images from this handbook, or some of their own choosing. Lossless image compression is normally accomplished for those users who cannot tolerate any loss of quality.

### 3.5 BI-LEVEL IMAGE COMPRESSION

So far we have addressed only images, but the NITF handles textual information also. BI-level compression is the same algorithm that is used in commercial facsimile devices. MIL-STD-188-161 (Group 3 Facsimile Apparatus for Document Transmission) and MIL-STD-188-196 DOD Interface Standard, Bi-level Image Compression for the NITFS) contains relevant technical information if users must learn more about this type of lossless compression or facsimile devices.

The NITF implementation of Bi-level compression provides three different modes of operation: mode 1, one-dimensional coding; mode 2, two-dimensional coding with standard vertical resolution, and mode 3, two-dimensional coding with higher vertical resolution. All of these modes are lossless.

It isn't necessary to know the details of Bi-level compression. Users only need to know that the fundamental concept of this coding algorithm is to detect run lengths of one of two colors (for example, black or white) in an image. These run lengths are then replaced with Huffman codes. Synchronization codes are embedded that indicate the beginning of an image, the end of a line, or like information.


Bi-Level Compression

Bi level compression should be used for text only. It should not be used for images. Fax machines also use bi level compression!
Bi level compression should be used for text only. It should not be used for images. Fax machines also use bi level compression!
Bi level compression should be used for text only. It should not be used for images. Fax machines also use bi level compression!
Bi level compression should be used for text only. It should not be used for images. Fax machines also use bi level compression!

FIGURE 14: EXAMPLE TEXT IMAGE

The text size is 472,000 bytes. Examples of compression using the three modes of Bi-level compression are shown in figure 14. Remember to use this type of compression only on textual images. Bi-level compression converts grayscale or color images to only two colors.

| Mode 1 | Mode 2 | Mode 3 |
|---|---|---|
| 52:1 | 67:1 | 59:1 |
| (.15 bits per pixel) | (.11 bits per pixel) | (.13 bits per pixel) |

FIGURE 15: BI-LEVEL IMAGE COMPRESSION RESULTS

### 3.6 VIDEO COMPRESSION

The use of video imagery is increasing. Technically, the reader can refer to it as Motion Imagery. Motion Imagery is defined as imaging sensor/systems that generate sequential or continuous streaming images at specified temporal rates (normally expressed as frames per second). This type of imagery includes electro optical (video and television), infrared, complex waveforms based on radar imaging, Motion Target Indication (MTI), and acoustic water falls. We are not going to address all of these technologies, but it is necessary to know of their existence and that NIMA has published a Video Imagery Standards Profile Version 1.3 (VISP-1.3). This profile summarizes the results of work accomplished by the Department of Defense/Intelligence Community/United States Imagery and Geospatial System (DOD/IC/USIGS Video Working Group (VWG). The profile identifies Motion Picture Expert Group–2 (MPEG-2) as the approved video compression format. At present only single frames captured or "frame grabbed" from video sources are transmitted by NITF systems. The UAV image used in the sample images was a single frame of video. The profile stipulates that the NITF be used for single frames of video. If users transmit video they must use compression.

### 3.7 NATIONAL IMAGE COMPRESSION

Users should be aware that this handbook does not address NITF compression as it applies to national imagery. Some of the items that are different in the national arena are:

- National compresses 8, 11, and 12 bit imagery data using both JPEG lossy and JPEG lossless compression (as well as other non- JPEG compression algorithms). The compressed data is transmitted in NITF format. Uncompressed national image data is stored in 2 bytes (11 and 12 bit data)

- For national, the term Quality levels apply to different bit rates, of which there are only three, plus lossless

- In addition to the compression of an image within national systems, there is a separate process, called pre-processing" that is conducted to remap the pixel values so that the compression process will be optimized. This requires the receiver to conduct a "post-processing" function after decompressing the image. Receivers of this data must understand the post processing of this algorithm.

### 3.8 OTHER COMPRESSION ALGORITHMS

There are other compression algorithms that users may come in contact with. Two, which users should be aware of, are Wavelets and Fractal compression.

Wavelets is a mathematical approach to signal representation in terms of time and frequency rather than merely frequency as would be the result of the DCT transform.

Wavelets compression breaks the signal down into many interrelated smaller components. The high-resolution image is decomposed into a hierarchy of components. Each level in the hierarchy contains more details than the next. The hierarchy is a pyramid type structure that is repetitive application of frequency scaling to Wavelet functions. The frequencies are divided into low and high frequency bands. The high frequency contains the detailed information. The low bank is then spilt again into high and low. This process is repeated until the lowest band contains only the "noise" information from the image. Lossless or lossy compression techniques are then performed on the signal's components.

Wavelets compression achieves very high compression rates. Ratios above 100:1 are common when lossy techniques are applied.

Because of the hierarchical nature of Wavelet compression, it is possible to transmit images compressed by Wavelets progressively (users may also do this with JPEG though not as easily). In progressive transmission, the user receives a sketch of the entire image at the start. As the transmission of the image continues, the image details fill in and the image becomes clearer. This means the receiver could stop the transmission when the image is clear enough.

Fractal techniques achieve compression by throwing away noise. In fractal compression the image is searched for one to one redundancy of elements within the image.

Relational information and not resolution specific data is retained in fractal coding, therefore in theory, Fractal coding produces a resolution independent representation of the image. Fractal compression achieves compression rates averaging 25 - 30:1, though higher rates can be achieved.

### 3.9 When to Not Use Compression

The key to not using compression is the intended use of the image by the recipient. Listed below are some circumstances where users should not use compression. Hopefully, the requester has included what they want to use the image for or identified the type of information or intelligence they want to extract from it in their request.

- Precision mensuration (determining exact dimensions or measurements). If a requestor is located at one the scientific and technical research centers and wants to determine the frequency of a newly discovered radar by measuring the feed horn, send it uncompressed. Any lossy technique would cause their measurements to be off.

- Bomb Damage Assessment (BDA) and Crater Analysis, again, finite measurement to deduce information. Do not use compression.

- Near original quality imagery for further analysis. Imagery analysts do not like to interpret imagery that has been compressed using lossy techniques. Send it to them uncompressed.

- Medical imagery, e.g., x-rays, Magnetic Resonance Imaging (MRI). Radiologists and doctors need the original quality images.

- If the requestor’s communication capabilities are all at the T1 and above, don’t waste time compressing it. These types of communication capabilities are usually located at major headquarters and the like. Conversely, if users know the only receipt capability is a low speed modem, they use some type of compression.

In conclusion, the primary reason for not using compression is because the **requestor specifically asked the sender not to**. Rest assured, if compressed imagery is sent to someone and that is not what they wanted, they will let the sender know.

# 4.0 JPEG 2000

## 4.1 JPEG 2000 Status

JPEG 2000 is the follow-on to the currently defined JPEG International Standard. It will most likely be based on a wavelet based compression technique. The International Standards Organization (ISO) is working a fast track to produce a working draft of the standard by November of 1998.

The JPEG 2000 standard is being designed to serve a number of markets and applications where the current JPEG standards fail such as low bandwidth dissemination of images on networks on the WWW, archive applications, medical imaging, electronic imaging, facsimile, and scanner and digital copier buffering. The standard is intended to advance imaging compression coding systems to serve applications for the next ten years.

JPEG 2000, which is based on the most current "state-of-the-art" technology, may be able to address some present shortfalls or issues within the USIGS community. Studies have shown that advanced compression algorithms can improve performance throughout the USIGS. For example, studies have already indicated that Wavelets provide an improved bit rate and image quality capability over current algorithms. This means we will be able to transmit higher quality images faster and store more of them.

JPEG 2000 potentially offers a number of benefits to the NITF community. NIMA is participating in the development of JPEG 2000 and provides representatives to the ISO meetings.

JPEG 2000 is the number one candidate to replace JPEG within the NITFS.

# 5.0 SUMMARY

## 5.1 DISCUSSION

The user has been exposed to a great deal of information, so lets summarize the major points about compression. There are two classes of compression: *lossy* and *lossless*.

*Lossless* compression means there is no change in the image data from the original.

*Lossy* compression discards information from the image. The higher the compression the greater the quality loss.

The reader should know there are several NITF compression algorithms that can be used depending on what type of imagery or textual data we are addressing.

The reader should recall that *DCT JPEG* could be used on all types of images (grayscale, color, multi-spectral, etc.); it performs well on all of them. DCT JPEG produced compression ratios ranging from a high of around 43:1 to a low of 2:1 on the sample images. It achieved the highest compression rates when applied to color images.

*DS JPEG* produces very high compression rates. Results ranged from more than 260:1 to a low of 15:1 on the samples. It provided very good quality on the UAV (video) images at the lowest Q settings. DS JPEG only can be applied to grayscale imagery; it can not be used on color.

DCT JPEG and DS JPEG are adaptive. They both compress a particular image based upon the information content or data of the image.

*Lossless JPEG* compression is used when no quality loss is acceptable but reduction is necessary.

*Bi-level compression* is a lossless compression algorithm used for textual information. It cannot be used on regular images.

The reader should remember why a user should and should not use compression. Use compression to make the files smaller so that they may be transmitted faster to recipients. The main reason not to use compression is that the requestor specifically **asked us not too**.

To assist the reader selecting a particular compression algorithm, table 3 on the following page presents summary information about each of the NITF algorithms.

TABLE 3: NITF IMAGE COMPRESSION ALGORITHM SUMMARY

| NITF Compression Algorithm | Image Type | Use & Remarks |
|---|---|---|
| *No Compression* | All | When no loss of **quality** (data) can be tolerated. Usually for recipients who possess large bandwidth capabilities (T1/T3). Medical imagery usually should not be compressed. |
| *Lossless JPEG* | All | When faster transmission times are desired with no loss of **quality**. Lossless techniques provide only modest decreases in the file size. Compression is on the order of 2:1 rate. |
| *DCT JPEG* | All | When transmission **time** is more important than quality. Provides lower compression rates than DS JPEG with less quality loss. This algorithm performs very well on all types of images. |
| *DS JPEG* | Grayscale | When recipients are bandwidth disadvantaged and are only capable of slow communication speeds or the **fastest** possible transmission **time** is required. Provides highest compression rates of all NITF JPEG variants. Extremely adaptive to image content. Q setting 4 is known as NIMA Method 4. Cannot be used on color images |
| *Bi-level* | Text/Bi-Level | Use one of three modes for textual information attached to NITF file. Mode 2 usually attains highest compression ratio. |

The user should have a better understanding of the NITF compression algorithms by now. The user should be aware of why, when and most importantly, when not to use compression techniques on different types of images.

The user should be aware that at some time in the future, new compression capabilities will be added to the NITF.

The user should use this handbook as a ready reference when they forget or cannot recall a particular topic. Experiment with the images on the CD. The best way to reinforce the information in this handbook is to actually compress the images and assess the quality impact.

A final rule of thumb – If the image is unusable after compression and decompression do not transmit it. Only transmit images that the receiver would use.

# APPENDIX A NITF BACKGROUND, HISTORY AND DESCRIPTION

The NITFS is the collaborative result of a US Government and Industry effort to provide a common facility for exchanging imagery, imagery derived information, and associated geospacial metadata. The purpose of the NTFS is to provide a common standard for the exchange and storage of files composed of images, graphics, text and associated data.

Technical review, community coordination and overall planning of the NITFS have been accomplished through the NITFS Technical Board (NTB) and its ad hoc working groups. The NTB has evolved over the years into a true consensus-based forum emphasizing cooperation and partnership between government and industry. The NTB operates under the authority of the Imagery Standards Management Committee and Geospacial Standards Management Committee (ISMC/GSMC), which is responsible for the selection and management of imagery and geospacial standards for the DOD, Intelligence Community (IC) and overall United States Imagery and Geospacial Information System (USIGS) community.

**NITF 1.0 (1984-1990).**

By 1984, the need for a standard data format became obvious to the imaging community, and a project was initiated to develop such a format. The original goal was to develop a co-standard that could be added to all of the existing systems and incorporated into new systems during the acquisition phases. The original result of this effort was Version 1.0 of the NITF, which was never implemented or fielded. The NTB was officially established at this point to continue and manage the technical development, validation, certification, and integration of the format into DOD. A Defense Support Project Office (DSPO) representative was appointed to manage and co-chair the NTB. An Intelligence Communications Project (INCA) representative was appointed to manage validation, certification, and testing as well as to co-chair the NTB.

**NITF 1.1 (1989-1994**

Version 1.1, an improved format, was developed, validated and proposed as the implementation baseline. The NITF Configuration Control Board (NCCB), chaired by a representative from the Office of the Assistant Secretary of Defense (OASD) for Command, Control, Communications and Intelligence (C3I) approved Version 1.1 for general implementation in March 1989. In 1990 a certification test facility was established in the Washington, DC area under INCA sponsorship, but was moved to the Joint Interoperability Test Command (JITC), Fort Huachuca, Arizona, in 1991 when the Defense Intelligence Agency assumed INCA’s responsibilities. By March 1992, over thirty different systems configurations had been tested and certified as compliant with NITF Version 1.1.

**NITF/NITF 2.0 (1994-2000)**

Development of an improved version of the NITF, intended to address the shortfalls with the previous versions was initiated in 1988. Initially, the new version was called NITF 2.0. A key improvement was the inclusion of communications support that would enable NITF to be transmitted over tactical circuits. This communications support was provided via the definition of the Tactical Communications Protocol 2 (TACO2). Additionally, improved image compression, forward error correction and enhanced graphics also began development. In May 1989, the Chairman of the Committee on Imagery Requirements and Exploitation (COMIREX) directed the adoption of the NITF as the IC standard for the transmission of secondary imagery. In 1991, the OSD directed that NITF be documented as a DOD Standard, and its name was changed to the National Imagery Transmission Format Standard (NITFS). The NITFS encompasses not only the format, but also the compression algorithms standards (i.e., JPEG, ARIDPCM, Bi-level, VQ), computer graphics metafile (CGM) standards, and communication protocol standard (TACO2). By 1994, the NITFS was being implemented in a variety of systems that went beyond the “secondary imagery dissemination”; in fact, currently, all or components of the suite of standards are or will be implemented by a variety of components within the USIGS Technical Architecture (UTA), including: primary dissemination systems, Unmanned Airborne Vehicles (UAV), digital imagery and geospatial archives and libraries, and commercial satellite vendors. As of March 1997, and estimated 18 commercial vendors have developed NITF compliant systems. Additionally, approximately 13,000 software licenses have been sold by industry providing commercial products supporting NITF 2.0.

**NITFS/NITF 2.1 (1998-2004)**

A number of factors have driven the changes made to NITF 2.0 during recent years. Among these are: the creation of the National Imagery and Mapping Agency, the mandate for the selection of implementation of commercial/international standards over government/military standards where possible; user requirements for improved fusion of information, whether imagery, geospatial, or other data type; and the ever increasing need to share data within and external to systems of the DOD/IC. NITF 2.1 is based on extensive coordination among NITFS users within the USIGS community, NATO and Allied Nations, as well as with commercial vendors and groups dealing with related standards and technologies.

The NITF 2.1 standard, has been reviewed by the USIGS community, as well as through the NATO and International Standards Organization. NATO has adopted a standard, identical to NITF 2.1 (WHICH Change Notice 1) called the NATO Secondary Imagery Format (NSIF), and the ISO has adopted the Basic Imagery Interchange Format (BIIF). Detailed information regarding this coordination is provided in the NITFS Program Plan, NIMA document NNPP. This large-scale review has ensured that MIL-STD-2500B is technically aligned with the other two documents. In addition, the quality of the document, from the editorial and organizational perspective has improved considerably by having a widespread review by international readers. MIL-STD-2500B was formally approved by the ISMC/GSMC on 22 August 1997.

NITFS/NITF 2.1 compliance testing will begin formally on 1 October 1988, although it is expected that there will be a number of Government systems and activities that will not be prepared to transition from NITF 2.0 to NITF 2.1 until the Mid 1999 – Mid 2000 timeframe. However, most commercial NITFS Systems and several Government systems are expected to implement and become compliant to NITF 2.1 by the end of 1998, or early 1999. Other NITF users will require significant time to assess the impact and determination for funding the necessary changes and implementation costs. Re-validation of compliance to the standard should be considered an integral part of the 2500B enhancement and implementation to all “NITF” systems and data.

**NITF FORMAT**

An NITF file typically includes one or more images with annotations such as graphic symbols or textual labels into a single computer file. It may also contain multi-page text reports and other information as well. It will always include security classification data. The security classification is attached to each item individually, and the overall classification (highest classification) is attached to the entire file. These parts of the file are organized in a logical structure.

NITF also provides for non-destructive overlaying of each of the components of this image. Think of the NITF structure like pages in a book, with each page assigned a specific order. This is known as the Display Level within the NITF structure and indicates the order in which objects are displayed on the screen. The lower display levels comprise the background and the higher ones the foreground.

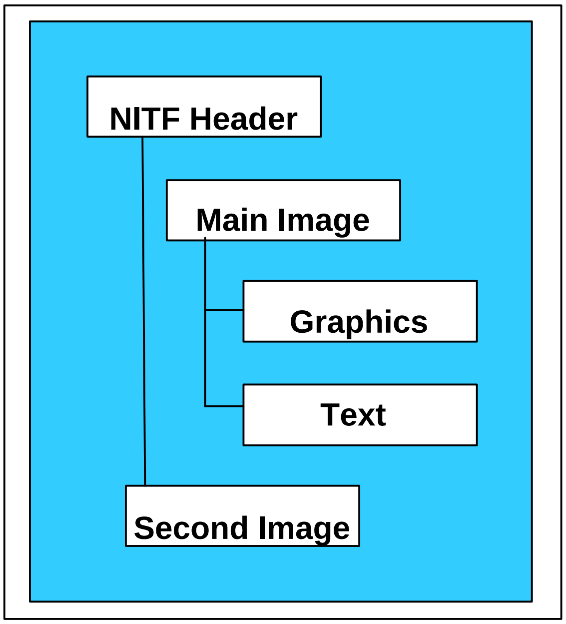



FIGURE 16: EXAMPLE NITF FILE

An example NITF file is shown in the figure and contains a main image, which has an arrow pointing to an object of interest (graphics), a text line identifying the object, and a second image that shows more detail of the object. This file can contain any number of these components. The graphics and text are non-destructive, which means they float on top of the image.

The NITF allows multiple images to be combined in the same file in a variety of ways, including overlays.

TABLE 4: PARTS OF A NITF FILE

| | |
|---|---|
| **Images** | Can be any type and any size and there can be one or multiple images of different types. |
| **Headers** | Every NITF file may include a header which includes information concerning the location of the image (coordinates), date and time it was collected, the security classification, type of imagery, etc.. |
| **Graphics** | For graphical annotation of images, NITF uses Computer Graphics Metafile (CGM) to draw lines and figures (arrows, boxes, military symbols, etc.) The user can specify the settings for line width, color, etc. |
| **Text** | NITF allows users to use text in several different ways. They may attach labels to a specific part of an image (or images), for example name a specific building. They may also include comprehensive written textual reports describing the image contents in as much detail as they wish to use. |
| **Communications** | Though not a part of the file, NITF includes the TACO 2 protocol, so that users may transfer the file via any suitable communications links they can use. They may also use other protocols to transfer the files electronically. |

The NITF includes a series of compliance levels (CLEVEL 1 to CLEVEL 6) for its files. These levels identify the specific parameters for the image file (or files) in the NITF file. The levels and maximum image files sizes are shown below:

TABLE 5: NITF COMPLEXITY LEVELS

| | |
|---|---|
| **CLEVEL 3** | 50 megabytes |
| **CLEVEL 5** | 1 gigabyte |
| **CLEVEL 6** | 2 gigabytes |
| **CLEVEL 7** | 10 gigabytes |

The NITF allows the user the flexibility to perform many operations to an image or to multiple images. Users may attach multiple images, graphics, symbols and text in one file.

Additional information on the NITF is contained in the family of Military Standards, which address NITF.

# APPENDIX B ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| ARIDPCM | Adaptive Recursive Differential Pulse Code Modulation |
| BDA | Bomb Damage Assessment |
| bmp | Bit Mapped Image |
| BWC | Bandwidth Compression |
| C3I | Command, Control, Communications, and Intelligence |
| CD | Compact Disk |
| CFS | Center for Standards |
| CGM | Computer Graphics Metafile |
| CIA | Central Intelligence Agency |
| CLEVEL | Compliance Level |
| CMP | Lead Technology's proprietary compression format |
| CMYK | Cyan, Magenta, Yellow, Black |
| COMIREX | Committee on Imagery Requirements and Exploitation |
| DCT JPEG | Discrete Cosine Transform JPEG |
| DCT | Discrete Cosine Transform |
| DISA | Defense Information System Agency |
| DOD | Department of Defense |
| DS JPEG | Down Sample JPEG |
| DSPO | Defense Support Program Office |
| FDCT | Forward Discrete Cosine Transform |
| FFT | Fast Fourier Transform |
| Gif | Graphic Interchange Format |
| IC | Intelligence Community |
| INCA | Intelligence Communications Architecture Project Office |
| IR | Infrared |
| ISMC/GSMC | Imagery Standards Management Committee/Geospacial Standards Management Committee |
| ISO | International Standards Organization |
| JIEO | Joint Interoperability and Engineering Organization |

| | |
|---|---|
| JITC | Joint Interoperability Test Command |
| JPEG | Joint Photographic Experts Group |
| LZW | Lempel-Ziv-Welsh |
| MIL-STD | Military Standard |
| mm | minute minute |
| MOA | Memoranda of Agreement |
| MPEG 2 | Motion Picture Experts Group 2 |
| MRI | Magnetic Resonance Imaging |
| MTI | Motion Target Indicator |
| NATO | North Atlantic Treaty Organization |
| NIMA | National Imagery and Mapping Agency |
| NITF | NATIONAL IMAGERY TRANSMISSION FORMAT |
| NITFS | National Imagery Transmission Format Standards |
| NTB | NITF Technical Board |
| OASD | Office of the Assistant Secretary of Defense |
| OSD | Office of the Secretary of Defense |
| RGB | Red, Green, Blue |
| RLE | Run Length Encoding |
| SAR | Synthetic Aperture Radar |
| ss | second second |
| TACO 2 | Tactical Communication Protocol 2 |
| Tif | Tagged Image Format |
| UAV | Unmanned Aerial or Airborne Vehicle |
| USIGS | United States Imagery and Geospacial System |
| UTA | USIGS Technical Architecture |
| VISP-1.3 | Video Imagery Standard Profile Version 1.3 |
| VQ | Vector Quantization |
| VWG | Video Working Group |
| WWW | World Wide Web |
| Y2K | Year 2000 |

# APPENDIX C  POINTS OF CONTACT

***NITF Matters***:

Danny Rajan
National Imagery and Mapping Agency
(703) 262-4416
E-mail:  rajans@nima.mil

***Video Matters***

Gordon Ferrari
National Imagery and Mapping Agency
(703) 262-4417
E-mail: ferrarig@nima.mil