Disclaimer:

As a condition to the use of this document and the information contained herein, the SWGIT requests notification by e-mail before or contemporaneously to the introduction of this document, or any portion thereof, as a marked exhibit offered for or moved into evidence in any judicial, administrative, legislative, or adjudicatory hearing or other proceeding (including discovery proceedings) in the United States or any foreign country. Such notification shall include: 1) the formal name of the proceeding, including docket number or similar identifier; 2) the name and location of the body conducting the hearing or proceeding; 3) the name, mailing address (if available) and contact information of the party offering or moving the document into evidence. Subsequent to the use of this document in a formal proceeding, it is requested that SWGIT be notified as to its use and the outcome of the proceeding. Notifications should be sent to: Chair@swgit.org

Redistribution Policy:

SWGIT grants permission for redistribution and use of all publicly posted documents created by SWGIT, provided that the following conditions are met:

1. Redistributions of documents, or parts of documents, must retain the SWGIT cover page containing the disclaimer.

2. Neither the name of SWGIT, nor the names of its contributors, may be used to endorse or promote products derived from its documents.

Any reference or quote from a SWGIT document must include the version number (or create date) of the document and mention if the document is in a draft status.

Section 19

Issues Relating to Digital Image Compression and File Formats

Introduction

This document provides a foundation of knowledge of compression algorithms and file formats utilized in digital imaging, including photography and scanning. It does not cover video compression algorithms or file formats. Understanding these processes and their advantages and disadvantages will allow agencies to make informed decisions for the appropriate application of file formats and compression algorithms. For a comprehensive understanding, the reader is encouraged to seek out other sources.

Compression

Compression is the process of reducing the size of a data file utilizing algorithms to rearrange the way data is organized within the file. Compression can be used to facilitate the storage and transfer of large files. The resulting file may retain all of the data or there may be data, including visual information, that is lost. Compression algorithms that retain all of the original data are "lossless," and those in which data is lost are "lossy." By setting the camera or software to the least amount of compression (or the fewest amount of pictures you can store), you will significantly decrease the amount of data lost. The decision to use lossy or lossless compression will be dictated by the intended use of the image.

Lossless Compression

When using lossless compression, no information is lost, but the compressed file uses fewer bits to represent the information. When the file is re-opened, the original data is reconstructed. Generally, lossless compression can achieve compression at a ratio of about 2:1 (thus reducing the file size by half). LZW (Lempel-Ziv-Welch algorithm) is an example of lossless compression.

Lossy Compression

When using lossy compression, information is lost and cannot be retrieved in its original form. Lossy compression can achieve compression ratios of greater than 2:1. JPEG (Joint Photographic Experts Group algorithm) is commonly used to accomplish this.

How it works

Image files can contain redundant or irrelevant data. During compression, this data is reorganized or removed. This makes the file smaller while keeping a pathway so that the data can be reproduced. Depending on the method selected, the user may or may not have control over the result. The average user of commercially available software will have limited control on how of the algorithms are deployed. The following tools are used alone or in concert with one another to achieve the desired compression for a file.

Run-length encoding is a variable length code. It is a lossless method designed to remove redundant data. No information is lost, it is just represented in a more concise way. The coded version depends on how frequently characters are repeated in the original data set. If there is much repetition, you will get a shorter coded file.

Example: 111111112223 → 182331 (2:1 compression)

In this example, a string of 12 values takes the space of only 6. There are eight occurrences of the number "1" represented by the number 18 in the string, three occurrences of the number "2" which is represented by 23 and one occurrence of the number "3" represented by 31.

Lexicographic encoding is also a variable length code. It is a lossless method designed to remove irrelevant data. The most repeated character is given the shortest code value. Code values can be stacked into packages that are more concise. No information is lost.

Example: 201121001

In this example, the number one is given the binary code value "0" because it is the most frequent value. Zero has the second highest occurrence and is given the binary code value "1". Finally, two is given the binary code value "10" because it is the least occurring. The original string contains nine numbers of 8 bits each for 72 bits or 9 bytes (9 x 8 = 72 bits or 9 bytes). In the coded version, no number needs more than two bits. Four two-bit numbers can comprise one eight-bit byte. The compressed version would only require 11 bits or less than two bytes.

Quantization encoding maps multiple values to a single replacement value. It is a lossy method designed to reduce the number of values used.

Example:	Original Value (3bits)	Encoded Value (2 bits)
7		
	}	3
6		
5		
	}	2
4		
3		
	}	1
2		
1		
	}	0
0		

In this simple example, an original value requiring 3 bits of data is transformed through quantization and now only requires 2 bits of data. For the purposes of this example, the original value was limited to 8 numbers. As the range of the original values increases there are more levels of compression available

JPEG Compression

JPEG uses some lossless algorithms, but also uses quantization. The quantization of the file can result in lost data. The amount of quantization is variable. JPEG can reduce file sizes 5:1 with minimal degradation and upwards to 20:1 with significant degradation. Many programs and cameras allow the user to choose the JPEG quality setting. Care should be taken to choose the level that is appropriate for the situation.

The JPEG algorithm begins by splitting the image into three separate channels creating three separate images. Each color channel image is broken into segments that are 8 pixels by 8 pixels in size (8x8 blocks). Each 8x8 block is represented by a mathematical function creating a new 8x8 block. Quantization is applied based on the "quality" level the user selects. The more quantization applied the smaller the file size resulting in greater loss. JPEG can be lossless if the quantization level is set to zero. After quantization, the 8x8 blocks are reassembled and the compressed color channels are combined back into one image.

As Figure 1. Demonstrates, excessive compression can have a dramatic visual effect. The image on the left has been compressed substantially more than the image on the right. Artifacts become more obvious and there is a substantial difference in image quality.

Figure 1. Demonstrates the difference between two image one using minimal compression (right) and one using more compression (left).

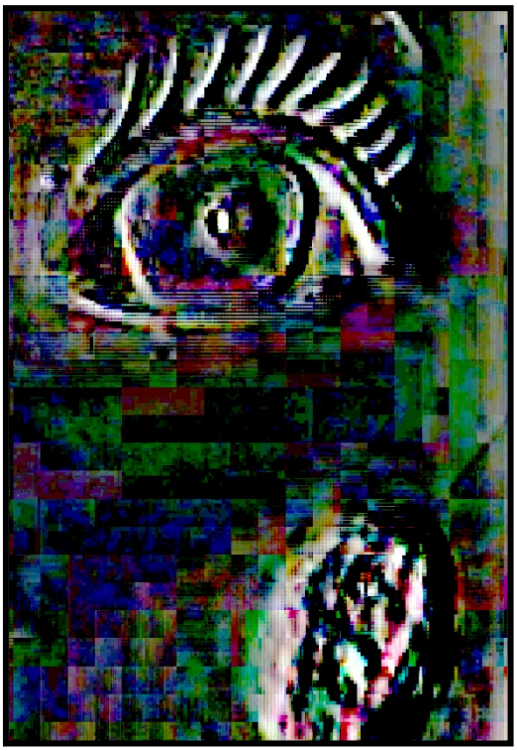
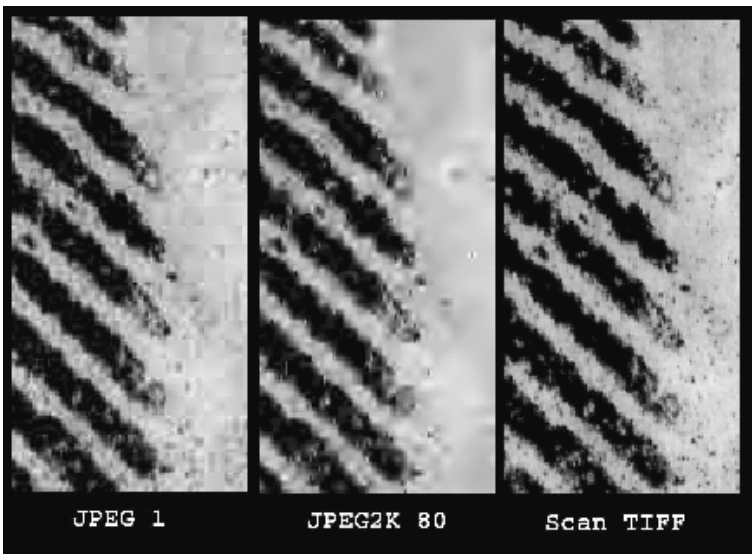


Figure 2. Represents the differences between the two images. White areas represent image data lost, dark areas represent image data preserved and colored areas represent changes in color values.

JPEG2000 is similar to JPEG but uses a different mathematical function. It does not segment the image using 8x8 blocks as JPEG does. It uses downward interpolation to create smaller versions of the image and applies a mathematical function followed by quantization to achieve compression. Compared to standard JPEG, JPEG2000 can achieve a greater compression of image files while maintaining the same image quality. JPEG2000 can reduce file sizes up to 20:1 with minimal degradation. It can compress up to 80:1; however, significant degradation occurs at this level.



In this example, the image on the right represents a portion of a fingerprint scanned using TIFF format and uncompressed. The image on the left is compressed with JPEG at 20:1 and the image in the center is compressed using JPEG2000 at 80:1. Note there is little or no difference in quality between JPEG and JPEG2000 even though there are substantial differences in the amount of compression.

Compression Artifacts

Compression artifacts are features created in the image that are not part of the original scene. Listed below, are some of the more common artifacts found when using excessive amounts of lossy compression.

Blocking - The JPEG algorithm breaks the image up into 8x8 blocks in each of the three-color channels. It processes each block separately, and then puts them all together again. In some cases, the blocks are very visible, and the colors appear altered.

Contouring – Exaggerated differences at edges and banding in a gradient.

Local color distortion – Appears as strange color patches in small locations on the image.

High frequency losses – Edges may appear fuzzy and fine detail patterns may be blurred.

Application of Compression

Compression can be applied at the time of capture or during processing and saving. This compression can be through hardware or software and may not be readily apparent to the user. The use of lossy compression and the degree to which it is applied is dependent on the end use. It may be acceptable to compress a Category I image that is used for documentation purposes. Lossless compression should be used on Category II images that are used for analysis; however, the use of lossy compression on these images does not preclude them from being analyzed if the pertinent features are retained. For more information on Category I and II images see SWGIT document "*Best Practices for Documenting Image Enhancement*".

Other Considerations

When considering compression, agencies have to balance cost, workflow, time, and image quality. Compression can make analysis more difficult even though the image is still usable. See SWGIT document "*Digital Imaging Technology Issues for the Courts*" for more information.

When considering an overall workflow, agencies should test the system from beginning to end to make sure it meets their quality needs first. Concessions based on cost and timesavings can be considered afterward. Employees should understand the philosophy behind these decisions. Specific references to archiving can be found in SWGIT Document "*Best Practices for Archiving Digital and Multi-Media Evidence (DME) in the Criminal Justice System*".

Be aware that some images are compressed for transmission or storage. It may be necessary to inquire if a received file was compressed because a higher resolution image may be available. When received images are compressed, care should be taken not to compress them further. If further processing is required, it is preferable to save a copy of the file in an uncompressed format. Processing can continue as needed then save with no compression or a lossless method. **Note:** It is recommended that the submitting agency notify the receiving agency when compression is used.

Saving Compressed Files

When saving a lossy-compressed file, any changes made are permanent. Resaving the image in an uncompressed format does not recover the data lost. Multiple resaves of a compressed file may magnify changes due to compression. Simply opening, viewing, and closing a file without saving does not result in further compression or degradation.

Users should have a good understanding of the camera settings required to accomplish the specific task. The default camera settings may not always be the best. This is also true for image processing software. When multiple users are using the same equipment, the settings are usually based on the last user's settings.

File Formats

A file format is the structure by which data is organized into a file. A file format is the common language that allows data to be shared. File formats often allow the use of compression to reduce the size of the file. The selection of file format is dependent on equipment available, workflow, and end use.

Data in an image file commonly contains a header, data block and footer. The header contains information about the image file including the type of file format, compression algorithm and possibly other metadata. The data block is the image content data. The footer may contain information about where the file ends and possibly other metadata.

Information in the header instructs the computer on how to open the image content information contained in the data block. If the header information is lost, corrupted or inconsistent with the data block the image may not open.

Some operating systems use file extensions as a convenient way for the computer to anticipate what the file format will be. However, it should be noted that file extensions can be changed and may not represent the actual file format. When this occurs, it can create problems using the file.

Common File Formats

Many image file formats exist for different applications and vendors. This is not an all-inclusive list.

JPEG File Interchange Format (JFIF) and Exchangeable Image File (EXIF) are common file formats that store JPEG-compressed information. These file formats often use the file extensions .JPG or .JPEG. This leads to confusion between JPEG, which is a compression algorithm and JFIF/EXIF that are file formats.

The EXIF format is capable of storing a large amount of metadata. Typically, when a camera is set on JPEG, an EXIF file is the result. The advantage to using EXIF is that metadata is stored in the file and can be used to document changes.

.JP2 file format is the file format for the JPG 2000 compression algorithm.

Tagged Image File Format (TIFF) is a flexible format that can be compressed or uncompressed. TIFF images from digital cameras tend to be large because they are limited on amount of compression and has all of the color values for all of the pixels. Although not common, it is possible to add a tag to a TIFF image essentially making it proprietary. The TIFF specification allows the incorporation of diverse compression algorithms, including some that are lossy. While the most common algorithms associated with the TIFF format are lossless, one can not assume this with every image.

Photoshop Document (PSD) is a format specific to Adobe software. In addition to the image information, all layer information is retained. It is useful for working within Photoshop but images cannot be used in most other applications. They are not suitable for archiving due their large size and proprietary nature.

RAW file format is not a specific file format but a class of formats. Each camera model essentially has its own version of a RAW file format. The data block of a RAW file contains the unprocessed pixel readings from the sensor chip and camera metadata.

Most RAW files are proprietary and specific to each camera model. Typically, cameras come with viewing software that requires conversion to a standard viewable format. Certain software packages also have utilities or plug-ins to handle these files but they are not necessarily compatible with all cameras.

Long-term storage of RAW files requires special considerations. There are many variables involved and it is dependent on camera model, sensor chip and processing. Each sensor has a specific way it captures data that will not be compatible with any other camera utility. Manufacturers are very hesitant about sharing this information. Provisions have to be made so that software and hardware will be available for opening the files in the future. Utilities provided by camera manufacturers are rarely supported beyond five years and may have compatibility issues with changes in operating system, file extension, etc. Open source RAW formats, such as Adobe Photoshop's Digital Negative (DNG) format, may simplify some of these cross platform concerns by converting a proprietary RAW format to an open source RAW format for archiving purposes.

There are resource considerations when capturing and storing in a RAW format. At some point, the original RAW file must be converted to a viewable format. The resulting image file after the conversion is considered a processed file and both files should be retained. This will have an impact on staff, storage facilities and equipment. It should be noted that once the conversion process has taken place the processed file cannot be converted back to its original RAW format.

Adobe Photoshop's Digital Negative (DNG) format is a royalty free RAW image format designed by Adobe systems. DNG is based on a TIFF format and mandates use of metadata. DNG was a response to demand for unifying camera RAW file formats.

Portable Network Graphics (PNG) format is used for internet applications. It does not support metadata.

Graphics Interchange Format (GIF) was originally developed by CompuServe for internet applications. It is an 8-bit format that has reduced color set, supports animation and LZW compression. It supports a non-rectangular image.

Bitmap (BMP) is a very basic format that allows most applications to open the image and store it using a different format.

Picture File (PICT) was primarily used in a Macintosh environment. It is rarely used today.

Other proprietary formats can exist that are formulated by vendors of turnkey systems. The vendor retains total control of the image using a key and third party software cannot open the file. The images may or may not be stored on site. These systems should be avoided.

Cautions

Knowing the characteristics and limitations of the compression and file format are essential to allow you to respond when an image is challenged.

Compression and changing file formats can strip metadata, and may or may not make the image unrecognizable or unusable.

Imaging management programs may alter metadata from the original file.

Incompatible file formats can create problems with interoperability between systems.

New algorithms are developed constantly that may not be valid. When implementing a new algorithm, be sure to validate it.