REVIEW OF IMAGE COMPRESSION AND VIDEO COMPRESSION: A SURVEY PAPER

[1]Sreelekshmi A N, [2]Anila S

Lecturer
Department of Computer Science
Sree Ayyappa College, Eramallikkara

Abstract— Multimedia images have become a vital and ubiquitous component of everyday life. The amount of information encoded in an image is quite large. Even with the advances in bandwidth and storage capabilities, if images were not compressed many applications would be too costly. The following research project attempts to answer the following questions: What are the basic principles of image compression? How do we measure how efficient a compression algorithm is? When is JPEG the best image compression algorithm? How does JPEG work? What are the alternatives to JPEG? Do they have any advantages or disadvantages? Finally, what is JPEG200?

Keywords— JPG, Pixel data, TIFF, Fidelity, MHEG, Lossless compression and Lossy compression

I. INTRODUCTION

Basically an image is a rectangular array of dots, called pixels. The size of the image is the number of pixels (width x height). Every pixel in an image is a certain color. When dealing with a black and white (where each pixel is either totally white, or totally black) image, the choices are limited since only a single bit is needed for each pixel. This type of image is good for line art, such as a cartoon in a newspaper. Another type of colourless image is a greyscale image. Grayscale images, often wrongly called "black and white" as well, use 8 bits per pixel, which is enough to represent every shade of gray that a human eye can distinguish. When dealing with color images, things get a little trickier. The number of bits per pixel is called the depth of the image (or bit plane). A bit plane of n bits can have 2^n colours. The human eye can distinguish about 2^{24} colours, although some claim that the number of colons the eye can distinguish is much higher. The most common color depths are 8, 16, and 24 (although 2-bit and 4-bit images are quite common, especially on older systems).

There are two basic ways to store colour information in an image. The most direct way is to represent each pixel's color by giving an ordered triple of numbers, which is the combination of red, green, and blue that comprises that particular color. This is referred to as an rgb image. The second way to store information about color is to use a table to store the triples, and use a reference into the table for each pixel. This can markedly improve the storage requirements of an image.

The following features are common to most bitmap files:

- Header: Found at the beginning of the file, and containing information such as the image's size, number of colors, the compression scheme used, etc.
- Color Table: If applicable, this is usually found in the header.
- Pixel Data: The actual data values in the image.
- Footer: Not all formats include a footer, which is used to signal the end of the data

1.1. BANDWIDTH AND TRANSMISSION

In our high stress, high productivity society, efficiency is key. Most people do not have the time or patience to wait for extended periods of time while an image is downloaded or retrieved. In fact, it has been shown that the average person will only wait 20 seconds for an image to appear on a web page. Given the fact that the average Internet user still has a 28k or 56k modem, it is essential to keep image sizes under control. Without some type of compression, most images would be too cumbersome and impractical for use. The following table is used to show the correlation between modem speeds and download time. Note that even high speed Internet users require over one second downloading the image.

Table 1

Modem Speed	Throughput – How Much Data Per Second	Download Time For a 40k Image
14.4k	1kB	40 seconds
28.8k	2kB	20 seconds
33.6k	3kB	13.5 seconds
56k	5kB	8 seconds
256k DSL	32kB	1.25 seconds
1.5M T1	197kB	0.2 seconds

1.2. INTRODUCTION TO IMAGE COMPRESSION

Image compression is the process of reducing the amount of data required to represent a digital image. This is done by removing all redundant or unnecessary information. An uncompressed image requires an enormous amount of data to represent it. As an example, a standard 8.5" by 11" sheet of paper scanned at 100 dpi and restricted to black and white requires more than 100k bytes to represent. Another example is the 276-pixel by 110-pixel banner that appears at the top of Google.com. Uncompressed, it requires 728k of space.

Lossless compression involves the preservation of the image as is (with no information and thus no detail lost). Lossy compression on the other hand, allows less than perfect reproductions of the original image. The advantage being that, with a lossy algorithm, one can achieve higher levels of compression because less information is needed. Various amounts of data may be used to represent the same amount of information. Some representations may be less efficient than others, depending on the amount of redundancy eliminated from the data. When talking about images there are three main sources of redundant information:

- Coding Redundancy- This refers to the binary code used to represent grey values.
- Inter pixel Redundancy- This refers to the correlation between adjacent pixels in an image.
- Psycho visual Redundancy - This refers to the unequal sensitivity of the human eye to different visual information.

In comparing how much compression one algorithm achieves verses another, many people talk about a compression ratio. A higher compression ratio indicates that one algorithm removes more redundancy then another (and thus is more efficient). If n_1 and n_2 are the number of bits in two datasets that represent the same image, the relative redundancy of the first dataset is defined as:

$R_d=1/C_R$, where C_R (the compression ratio) $=n_1/n_2$

The benefits of compression are immense. If an image is compressed at a ratio of 100:1, it may be transmitted in one hundredth of the time, or transmitted at the same speed through a channel of one-hundredth the bandwidth (ignoring the compression/decompression overhead). Since images have become so commonplace and so essential to the function of computers, it is hard to see how we would function without them.

Image Compression Model

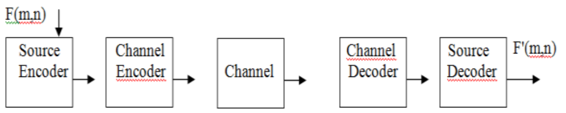

Fig 1

Image compression models differ in the way they compress data, there are many general features that can be described which represent most image compression algorithms. The source encoder is used to remove redundancy in the input image. The channel encoder is used as overhead in order to combat channel noise. A common example of this would be the introduction of a parity bit. By introducing this overhead, a certain level of immunity is gained from noise that is inherent in any storage or transmission system. The channel in this model could be either a communication link or a storage/retrieval system. The job of the channel and source decoders is to basically undo the work of the source and channel encoders in order to restore the image to the user.

1.3. FIDELITY CRITERION

A measure is needed in order to measure the amount of data lost (if any) due to a compression scheme. This measure is called a fidelity criterion. There are two main categories of fidelity criterion: subjective and objective. Objective fidelity criterion, involve a quantitative approach to error criterion. Perhaps the most common example of this is the root mean square error. A very much related measure is the mean square signal to noise ratio. Although objective field criteria may be useful in analyzing the amount of error involved in a compression scheme, our eyes do not always see things as they are. Subjective field criteria are quality evaluations based on a human observer. These ratings are often averaged to come up with an evaluation of a compression scheme. There are absolute comparison scales, which are based solely on the decompressed image, and there are relative comparison scales that involve viewing the original and decompressed images side by side in comparison. Example of absolute comparison scale is given below.

Table 2. Absolute Comparison Scale

Value	Rating	Description
1	Excellent	An image of extremely high quality. As good as desired.
2	Fine	An image of high quality, providing enjoyable viewing.
3	Passable	An image of acceptable quality.
4	Marginal	An image of poor quality; one wishes to improve it.
5	Inferior	A very poor image, but one can see it.
6	Unusable	An image so bad, one can't see it.

1.4. SOME JPEG ALTERNATIVES

The JPEG compression algorithm, the report will now proceed to examine some of the widely available alternatives. Each algorithm will be examined separately, with a comparison at the end. The best algorithms to study for our purposes are GIF, PNG, and TIFF.

1.4.1. GIF Compression

The GIF (Graphics Interchange Format) was created in 1987 by CompuServe. It was revised in 1989. GIF uses a compression algorithm called "LZW," written by Abraham Lempel, Jacob Ziv, and Terry Welch. Unisys patented the algorithm in 1985, and in 1995 the company made the controversial move of asking developers to pay for the previously free LZW license. This led to the creation of GIF alternatives such as PNG (which is discussed later). However, since GIF is one of the oldest image file formats on the Web, it is very much embedded into the landscape of the Internet, and it is here to stay for the near future. The LZW compression algorithm is an example of a lossless algorithm. The GIF format is well known to be good for graphics that contain text, computer-generated art, and/or large areas of solid color (a scenario that does not occur very often in photographs or other real life images). GIF's main limitation lies in the fact that it only supports a maximum of 256 colors. It has a running time of $O(m^2)$, where m is the number of colors between 2 and 256. The first step in GIF compression is to "index" the image's color palette. This decreases the number of colors in your image to a maximum of 256 (8-bit color). The smaller, the number of colors in the palette, the greater the efficiency of the algorithm. Many times, an image that is of high quality in 256 colors can be reproduced effectively with 128 or fewer colors.

LZW compression works best with images that have horizontal bands of solid color. So if you have eight pixels across a one-pixel row with the same color value (white, for example), the LZW compression algorithm would see that as "8W" rather than "WWWWWWWW," which saves file space.

1.4.2. PNG Compression

The PNG (Portable Network Graphic) image format was created in 1995 by the PNG Development Group as an alternative to GIF (the use of GIF was protested after the Unisys decision to start charging for use of the LZW compression algorithm).

PNG is an open (free for developers) format that has a better average compression than GIF and a number of interesting features including alpha transparency (so you may use the same image on many different-colored backgrounds). It also supports 24-bit images, so you don't have to index the colors like GIF. PNG is a lossless algorithm, which is used under many of the same constraints as GIF. It has a running time of $O(m^2 \log m)$, where m is again the number of colors in the image.

Like all compression algorithms, LZ77 compression takes advantage of repeating data, replacing repetitions with references to previous occurrences. Since some images do not compress well with the LZ77 algorithm alone, PNG offers filtering options to rearrange pixel data before compression.

As an example, of how PNG filters work, imagine an image that is 8 pixels wide with the following color values: 3, 13, 23, 33, 43, 53, 63, and 73. There is no redundant information here, since all the values are unique, so LZ77 compression won't work very well on this particular row of pixels. When the "Sub" filter is used to calculate the difference between the pixels (which is 10) then the data that is observed becomes: 3, 10, 10, 10, 10, 10, 10, 10 (or 3, 7*10). The LZ77 compression algorithm then takes advantage of the newly created redundancy as it stores the image.

Another filter is called the “Up” filter. It is similar to the Sub filter, but tries to find repetitions of data in vertical pixel rows, rather than horizontal pixel rows.

The Average filter replaces a pixel with the difference between it and the average of the pixel to the left and the pixel above it.

The Paeth (pronounced peyth) filter, created by Alan W. Paeth, works by replacing each pixel with the difference between it and a special function of the pixel to the left, the pixel above and the pixel to the upper left.

The Adaptive filter automatically applies the best filter(s) to the image. PNG allows different filters to be used for different horizontal rows of pixels in the same image. This is the safest bet, when choosing a filter in unknown circumstances.

PNG also has a no filter, or "None" option, which is useful when working with indexed color or bitmap mode images.

1.4.3. TIFF Compression

TIFF (Tagged Interchange File Format), developed in 1995, is a widely supported, highly versatile format for storing and sharing images. It is utilized in many fax applications and is widespread as a scanning output format.

The designers of the TIFF file format had three important goals in mind:

a. **Extendibility**. This is the ability to add new image types without affecting the functionality of previous types.

b. **Portability**. TIFF was designed to be independent of the hardware platform and the operating system on which it executes. TIFF makes very few demands upon its operating environment. TIFF should (and does) perform equally well in a wide variety of computing platforms such as PC, MAC, and UNIX.

c. **Revisability**. TIFF was designed not only to be an efficient medium for exchanging image information but also to be usable as a native internal data format for image editing applications.

The compression algorithms supported by TIFF are plentiful and include run length encoding, Huffman encoding and LZW. Indeed, TIFF is one of the most versatile compression formats. Depending on the compression used, this algorithm may be either lossy or lossless.

Some limitations of TIFF are that there are no provisions for storing vector graphics, text annotation, etc (although such items could be easily constructed using TIFF extensions). Perhaps TIFF’s biggest downfall is caused by its flexibility. An example of this is that TIFF format permits both MSB ("Motorola") and LSB ("Intel") byte order data to be stored, with a header item indicating which order is used. Keeping track of what is being used when can get quite entertaining, but may lead to error prone code.

Biggest advantage lies primarily in its highly flexible and platform-independent format, which is supported by numerous image-processing applications. Since it was designed by developers of printers, scanners, and monitors it has a very rich space of information elements for colorimetric calibration, gamut tables, etc.

1.5. MPEG VIDEO COMPRESSION

Most people are familiar with MPEG compression; it is used to compress video files. MPEG stands for Moving Pictures Expert Group, which is probably a friendly jab at JPEG. The founding fathers of MPEG are Leonardo Chairiglione from Italy and Hiroshi Yasuda from Japan. The basic idea is to transform a stream of discrete samples into a bit stream of tokens which takes less space, but is just as filling to the eye or ear. MPEG links the Video and Audio streams with layering. This keeps the data types synchronized and multiplexed in a common serial bit stream.

MPEG1 was developed for high bit rates in 128 Mbps range. It handles progressive non-interlaced signals. MPEG1 has parameters of (SIF) Source Input Format pictures (352 pixels x 240 lines x 30 frames/sec) and a coded bit rate less than 1.86 Mbps. As an aside, MP3 audio files are encoded using MPEG1’s audio codec.

MPEG2 was developed for lower bit rates in the 64 Mbps range that would efficiently handle interlaced broadcast video (Standard Definition Television). It decor relates multichannel discrete surround sound audio signals that have a higher redundancy factor then regular stereo sound. MPEG2 brought about the advent of levels of service. The two most common levels are the SIF Low Level 352 pixels x 240 lines x 30

frames/sec and the Main Level 720 pixels x 480 lines x 30 frames/sec.
MPEG3 was developed for High Definition Television but a few years later it was discovered that MPEG2 would simply scaled with the bit rate, which caused MPEG3 to be shelved.
MPEG4 was developed for low bit rates in the 32 Mbps range that would handle the new videophone standard (H.263). MPEG4 also has the ability to pick the subjects of a video out of the scene and compress them separately from the background.
Generically the MPEG syntax provides an efficient way to represent image sequences in the form of more compact coded data. For example, a few tokens amounting to 100 bits can represent an entire block of 64 samples to a point where you can't tell the difference. This would normally consume (64*8) or 512 bits. During the decoding process, the coded bits are mapped from the compact representation into the original format of the image sequence. A flag in the coded bit stream signals whether the following bits are to be decoded with DCT algorithm or with a prediction algorithm.
In this compression schema, macro block predictions are formed out of arbitrary 16 x 16 pixel (or 16x8 in MPEG-2) areas from previously reconstructed pictures. There are no boundaries that limit the location of a macro block prediction within the previous picture. Reference pictures (from which you form predictions) are for conceptual purposes a grid of samples with no resemblance to their coded form.

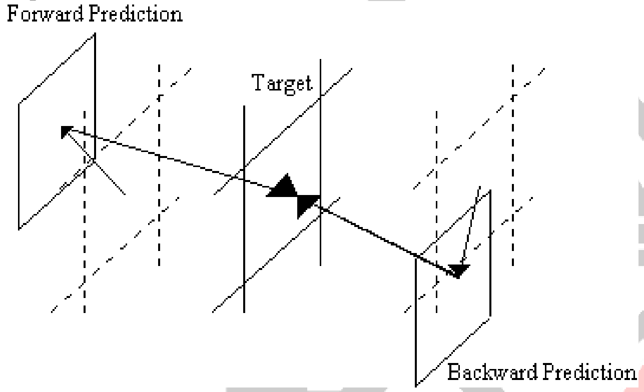

Figure 2: MPEG Predictions

Picture coding macro block types are (I, P, B). All (non-scalable) macro blocks within an I picture must be coded Intra (which MPEG encodes just like a baseline JPEG picture). However, macro blocks within a P picture may either be coded as Intra or Non-intra (temporally predicted from a previously reconstructed picture). Finally, macro blocks within the B picture can be independently selected as Intra, Forward predicted, Backward predicted, or both forward and backward (Interpolated) predicted. The macro block header contains an element, called macro block type, which can flip these modes on and off like switches.

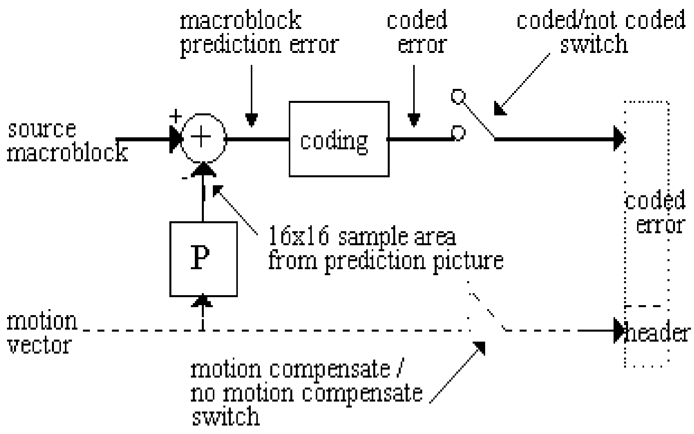

Figure 3: Macro block Coding

The component switches are:

1. Intra or Non-intra
2. Forward temporally predicted (motion forward)
3. Backward temporally predicted (motion backward)
4. Conditional replenishment (macroblock_pattern).
5. Adaptation in quantization (macroblock_quantizer_code).
6. temporally predicted without motion compensation

The first 5 switches are mostly orthogonal (the 6th is a special case in P pictures marked by the 1st and 2nd switch set to off "predicted, but not motion compensated.").

Without motion compensation:

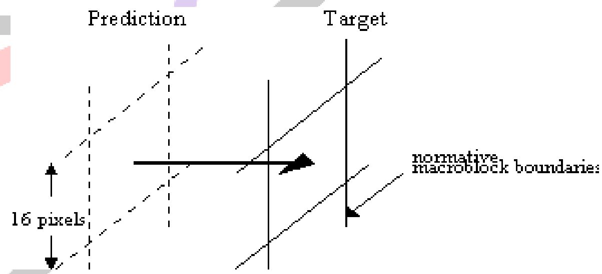

Figure 4: Encoding Without Motion Compensation

With motion compensation:

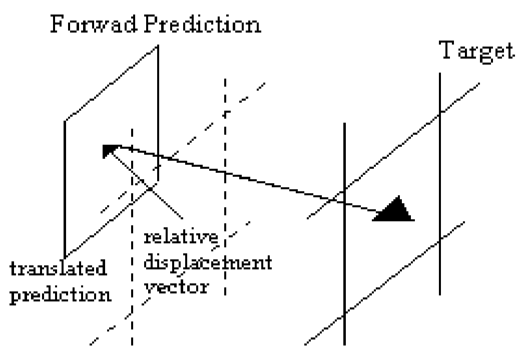

Figure 5: Encoding With Motion Compensation

Naturally, some switches are non-applicable in the presence of others. For example, in an Intra macro block, all 6 blocks by definition contain DCT data; therefore there is no need to signal either the macro block pattern or any of the temporal prediction switches. Likewise, when there is no coded prediction error information in a non-intra macro block, the macro block_quantizer signal would have no meaning.

If the image sequence changes little from frame-to-frame, it is sensible to code more B pictures than P. Since B pictures by definition are not used as prediction for future pictures, bits spent on the picture are wasted. Application requirements in temporal placement of picture coding types are random access points, mismatch/drift reduction; channel hopping, program indexing, error recovery and concealment.

Conservative compression ratios of 12:1 and 8:1 have demonstrated true transparency for sequences with complex spatial temporal characteristics such as rapid divergent motion and sharp edges, textures, etc.

MPEG is a DCT based scheme with Huffman coding and have the same definition as H.261, H.263 and JPEG.

The primary technique used by MPEG for compression is transform coding with an 8x8 DCT spatial domain blocks.

1.6. MHEG(Multimedia Hypermedia Experts Group)

MHEG is a standard that can organize and compress an entire multimedia presentation, consisting of any type and combination of multimedia files. It is a data interchange format with three primary levels of representation. MHEG classes, objects and run-time objects. MHEG has data structures which are reused in multiple classes. A module of useful definitions is used to maintain type consistency in these data structures. There are three types of identification mechanisms. The external identification does not need to reference an MHEG object to be decoded. The symbolic identification may replace any other external or internal identification. The internal identification is used to address MHEG objects.

The MHEG class hierarchy has a root class called MH-object class. It defines two data structures common to all other classes and is inherited by all lower level classes. Class identifier does just what it says, by identifying the type of each encoded class. MHEG will store its reusable objects in a database where authors can gain easy access to them.

There is a content class that describes objects to be presented to the user. Every content object is an atomic piece of information that is a particular medium type. Each object can contain either the digital data or a unique reference to the data stream.

There are virtual coordinates that are stored in the object as information on the original size and duration of the stored object. This technique avoids dependencies in the number of pixels in the target window or the audio sampling rate. This also allows the objects to be modified by changing the audio sequence or to do clipping or zooming. An example of something that cannot be altered would be changing the color of the text.

MHEG allows individual streams in interleaved audio/video sequences. There is a multiplexed content class that refers to the data with a description for each multiplexed stream. This class allows dynamic multiplexing of multiple streams. By interfacing inter-stream-synchronization mechanisms you can accomplish such tasks as lip synchronization.

There are state transitions that must be adhered to in the MHEG specifications. The preparation status defines the availability of an object. The objects then move through the following states: Not ready or not Running, Prepare, Ready, Running, Processing, and Destroy.

The link class defines a logical relationship between the action object and a content object or virtual view. This link class defines the conditions that actions are sent to the objects. This way at execution time each link instance is tied to an event. When that event occurs, the link is activated and the action is sent to the object.

The last few classes are as follows. A composite class allows composite objects to be part of other composite objects. The composite class keeps track of external references and controls the objects used to construct the presentation. Once again refer back to an object oriented design to image this structure. The container class provides a set of objects that are transferred as a whole set. The descriptor class encodes information about objects in a presentation and uses its information to determine if there are available resources for the presentation. Finally, the script class communicates with external functions or programs such as monitors, printers, or databases.

II. RELATED WORKS

In paper [1], reviewed and discussed about the image compression, need of compression, its principles and classes of compression and various algorithm of image compression. This paper attempts to give a recipe for selecting one of the popular image compression algorithms based on Wavelet and some other approaches.

In paper [5], focused on JPEG decompression techniques are very useful in 3G/4G based markets, handheld devices and infrastructures. To improve the visual quality of the JPEG document images at low bit rate and at low computational cost. In this paper, implement the decompression technique for JPEG document images.

In paper [6], initiates a new digital multimedia standard series. The purpose of the series is to make information about digital multimedia standards readily available.

In this paper [7], signal and image processing is a field which has been revolutionized by the application of computer and imaging technology. It has become very difficult to manage uncompressed multimedia (graphics, audio and video) data because it requires considerable storage capacity and transmission bandwidth. To solve this issue several techniques have been developed.

In paper [8], the development and demand of multimedia product is growing increasingly fast, contributing to insufficient bandwidth of network and storage of memory device. This justifies the use of different compression techniques to decrease the storage space and efficiency of transferring the images over the network to conserve the bandwidth.

In this paper [9], an image is the result of a visual signal of any object captured by optical devices such as human eyes, camera lenses etc. Image is the two dimensional representation of the object.

In paper [10], the increasing attractiveness and trust on digital photography has given rise to its use for visual communication. It requires storage of large quantities of data. Due to limited bandwidth and storage capacity, images must be compressed before storing and transmitting.

III. EXPERIMENTAL RESULTS

Although various algorithms have been described so far, it is difficult to get a sense of how each one compares to the other in terms of quality, efficiency, and practicality. Creating the absolute smallest image requires that the user understand the differences between images and the differences between compression methods. Knowing when to apply what algorithm is essential. The following is
a comparison of how each performs in a real world situation.

	File size in bytes	
Raw 24-bit	921600	
GIF (LZW)	118937	
TIFF (LZW)	462124	
PNG (24-bit)	248269	
PNG (8-bit)	99584	

Table 3 Summary of TIFF, GIF and PNG5

In table 3, the 8-bit PNG compression algorithm produced the file with the smallest size (and thus greater compression). Does this mean that PNG is always the best option for any screen shot? The answer is a resounding NO! Although there are no hard and fast rules for what is the best algorithm for what situation, there are some basic guidelines to follow. A summary of findings of this report may be found in the following table

Fig 2

The above screen shot was compressed and reproduced by all the three compression algorithms. The results are summarized in the following table

Image Compression Comparison
Table 4

	TIFF	GIF	PNG
Bits/pixel (max. color depth)	24-bit	8-bit	48-bit
Transparency		●	●
Interlace method		●	●
Compression of the image	●	●	●
Photographs	●		●
Line art, drawings and images with large solid color areas	●	●	

IV. CONCLUSION

The JPEG algorithm was created to compress photographic images, and it does this very well, with high compression ratios. It also allows a user to choose between high quality output images, or very small output images. The algorithm compresses images in 4 distinct phases, and does so in $O(n^2 \log(n))$ time, or better. It also inspired many other algorithms that compress images and video, and do so in a fashion very similar to JPEG. Most of the variants of JPEG take the basic concepts of the JPEG algorithm and apply them to more specific problems.
Due to the immense number of JPEG images that exist, this algorithm will probably be in use for at least 10 more years. This is despite the fact that better algorithms for compressing images exist, and even better ones than those will be ready in the near future.

REFERENCES

[1] Sachin Dhawan, Dept of ECE, UIET,Kurukshethra University,Kurukshethra,Haryana,India, Areview of image compression and Comparison of its algorithms.

[2] "Introduction" through "A Quick Comparison of Image Compression Techniques": "A Guide to Image Processing and Picture Management", A.E Cawkell, Gower Publishing Limited, 1994.

[3] http://www.oit.umass.edu/publications/at_oit/Archive/spring00/jv_compress.html

[4] http://www.dmi.unict.it/~fstanco/SCCG2001.pdf

[5] Miss. Punam N Kamble,Computer Science and Technology, Department of Technology, Shivaji University, Kolhapur, India, "Decompression of JPEG document Images: Asurvey Paper"

[6] http://www.springer.com/br/book/9780412087714

[7] Nirbay Kashyap, Assistant Professor, Dept of Computer Science and Engg., Amity Sschool of Engineering and Technology, Uttar Pradesh and Dr Shailendra Narayan Singh, Dept of Computer Science and Engg., Amity Sschool of Engineering and Technology, Uttar Pradesh ,"Review of Image Compression and Comparison of its algorithms."

[8] Afshan Mulla, Namrata Gunjikar and Radhika Naik, International Journal of Computer Applications," Comparison of different image compression techniques" ,May 2013.

[9] Gomathi K V, Lotus R, Centre for Information Technology and Engineering, Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu "Digital Image Compression Techniques"
[10] Athira B Kaimal, S Manimurugan, C S C Devadass, "Image Compression Techniques: A Survey".
[11] http://home.earthlink.net/~ritter/tiff/
[12] http://www.espipd.com/Comparison_GIF_JPEG_PNG.pdf
[13] http://netdb.chungbuk.ac.kr/CCNGIS2/VideoDBMS/@MHEG.pdf