Image Compression Algorithims:

A Comparsion and Review

Sherief Elsowiny
TSYS School of Computer Science
Columbus State University
Columbus, GA, USA
elsowiny_sherief@columbusstate.edu

Abstract—Image compression remains prevalent today just as much as it was at the onset of image processing and computing. Compression algorithms exist to reduce the redundancies in the representation of data. With ever increasing images being taken and shared to this day, with quality also improving, it makes it worth revisiting image compression and exploring the fundamentals and ongoing research to further help save on optimizing our image data efficiently.

Keywords—Image Compression, Lossless, lossy, image compression algorithms, image formatting compression

I. Introduction

Image compression serves as a methodology for achieving reductions in the data used in representing an image, while maintaining the quality and details of the original image. Many methods or algorithms exist to accomplish this task where each have their own benefits and costs associated with them. At a glance, these methods fall under lossy and lossless methods where the former method aims to reduce more data in the compression ultimately sacrificing some quality in doing so.

Conversely the latter method aims to preserve the quality of the image and makes usage of more advanced algorithms to achieve the reduction in data but preserving image quality. Before one even considers the reduction in data used to represent an image, one must wonder how an image can be represented in the first place.

Various methods exist in compression with importance on data redundancy reductions being as important today as ever before.

II. Background

A. Motivation

The desire for image compression is a result of a multitude of reasons. Namely we can imagine how storage would be a big factor as data is needed to represent an image. If an image is using large amounts of data to represent itself when a simpler method is sufficient if not better, why not save that extra money and reduce the image size?

When transmitting data, it is common to exchange images whether it is displaying it on a webserver or in sending to a friend. That transmission of data relies on downloading that information and also uploading it. When large amounts of data are used, all processes involved in transmitting can be slow. A lag can be seen in some cases and can be observed if too much data is needed to represent an image and a slow connection is resulting in the downloading of that data to be too long.

When using image compression techniques, we are able to reduce some of this data whether its redundancies or particular data points, we aim to reduce the size and have a similar if not exact replica of the image afterwards.

B. Reducing Redendancies

Given that numbers and patterns are used in the representation of images, there exists useless data in some of these files where we can make use of removing such pixels or values and remain with more or less the same image albeit less storage needed to represent it.

a) Image Encoding

It would make sense for some images to have repeat values given that numbers are used to represent it, and, in those cases, some values will be repeated more than others.

b) Pixel Relation

When viewing an image, it is obvious some colors are the same across certain segments of the image. For example, the color blue can span across the entire top half of an image and in such, the same value is mapped numerically to that color and can be compressed.

c) Human Perception(Psycho-Visual Redundancy):

The way in which humans perceive color and image is a result of light and wave and color spectrums. We can manipulate the brightness and coloring of certain images and remove unnecessary data that would otherwise go unnoticed by a human observer.

C. Lossy vs Lossless

When discussing any form of compression, lossy and lossless is brought up in comparing the two types of compression.

- Lossy image compression gives a better compression ratio sacrificing some quality in the process, but still allows for a very similar resembling image.
- Lossless on the other hand aims to preserve that quality and uses various methods to either transform or encode the image.

III. Image Represetation

An image like most data represented on the computer, can be represented using numbers. There exist various ways in which to depict in image, ranging from just black and white to higher depth-colored images, but at the crux of it all, is a matrix or grid of different pixel values representing either a brightness in the case of black and white images, or a color value in the case of colored images.

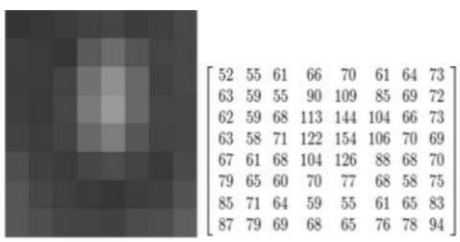

Fig. 1. Greyscale image represented as a matrix[1]

When viewing images on the computer in modern times, we are viewing a collection of pixels where each pixel is responsible for depicting a color.

We can look at the RGB representation of images which uses the red, green blue spectrum to depict images where each value has 256 depictions or shades of that color. A combination of these values allows for 16 million colors to be seen.

Fig. 2. RGB/HEX/HSL representation of red, green, and blue respectively

It is worth mentioning that images can also be represented by the HSL values corresponding to hue saturation and lightness. When representing an image in this way it helps to think of how each pixel will have a lightness associated with it.

The human eye cannot distinguish between certain levels of brightness and thus image compression can make usage of this fact.

IV. Image Formats

A. JPEG

Joint Photographic Experts Group or more commonly known as JPEG is one of many known file formats used to represent an image. Although it is known as a format, it is a compression algorithm as part of the JIFF format JPEG is known for its lossy image representation.

B. PNG

Portable Network Graphics uses a bitmap format and allows for lossless image compression. PNG was built to improve upon the GIF format. It can use a variety of color modes such as RGB and greyscale as well as bitmap. PNG is notable for its usage across the web and can display millions of different colors.

C. TIFF

Tagged Image File Format. This is another common image format supporting the similar variety of color modes present in PNG such as bitmap and RGB.

This file format is notable for its quality and most favorited among graphic designers and photographers for its preservation.

D. RAW

Image format containing data from the sensors of a digital camera. Contains unprocessed data.

E. Others

a) Graphic Interfchange Format (GIF) supports 8 bits per pixel and can contain 256 indexed colors. Also allows for frames to be combined creating animations.

b) WebP created by Google and supports both lossy and lossless compression

c) BMP(Bitmap) uncompressed format used by Microsoft internally.

V. Lossless

Lossless image compression ensures the original can be brought back and that no quality is lost when compressing the data as opposed to lossy image compression where lossy permits the deletion of certain data.

Lossless compression focuses on statistical redundancies to reduce the need to represent the image. Methods such as encoding, and transformation are used and allow a better more efficient representation of the image.

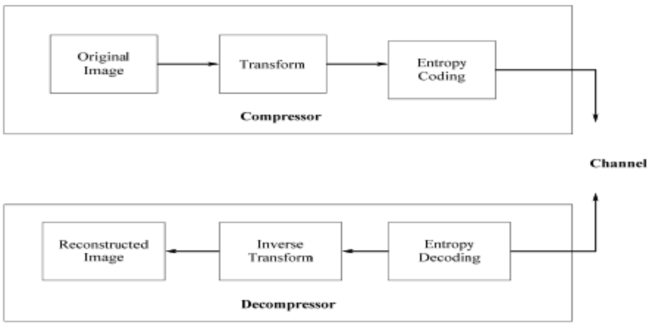

Fig. 3. Lossless Image Compressor negating the quantization step that is commonly used in lossy compression [6]

A. Huffman Coding

Huffman Coding remains a popular lossless compression technique used not only in images but also in text files and audio compression as well.

The objective of Huffman Coding is to assign in this case pixel values that are occurring more frequently, less bits for representation while giving those pixel values that are not occurring as much, more bits for their representation.

A tree data structure is used in this algorithm and constructed from the bottom up.

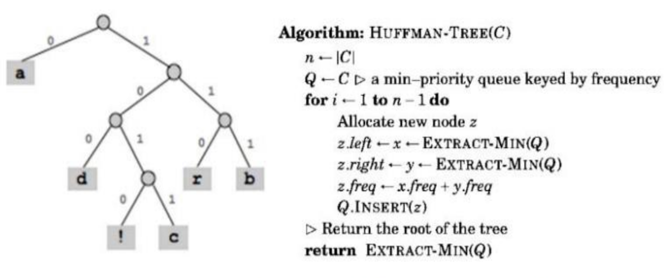

Fig. 2. Huffman Tree Algorithm [5]

B. Run Length Encoding

Run length encoding or RLE is a very simple but still effective algorithm in compressing data. In this algorithm the objective is to find runs of data i.e., values that are consecutive and repeating and replace them with the value and the frequency of which it occurs.

This algorithm is very effective for when there are distinct and minor colors shown on the image. A good example to think of would be a figure on a white background. The white background would be represented by the same value and be shown repeatedly throughout the image.

A bitmap can also be used to map the color values to different bits and find a run that way and reduce the image data in that way as well.

142 142 142 142 230 230 230 230 230 230 121 121 121 121
{142, 4} {230, 6} {121, 4}

Fig. 3. Example of run length encoding[4]

C. Dictionary based methods

Dictionary based methods involve encoding sequences of strings and patterns of data into simplified symbols, keeping track of what maps to what in a dictionary table.

a) LZW(Lempel-Ziv-Welch), LZ77, LZ78

This dictionary-based algorithm is quite popular as it aims to replace sequences of data with different codes thus creating a dynamic dictionary.
Other variations exist for the LZW algorithm each with their own minute differences.
Some notable ones are the LZ77 and LZ78 algorithms both created by Lempel-Ziv.
LZ77 works on past data and is slower in comparison to LZ78 which is faster and attempts to works on future data.

D. Lossless Predictive Coding

An interesting approach to lossless image compression is the use of predicting the values of each pixel by its neighboring and previous pixels.

In this method, in predicting the current pixel, a statistical estimation is made based on the previous and nearby pixels and the new information to be encoded is the error from the difference obtained by the actual and predicted value of the pixel.

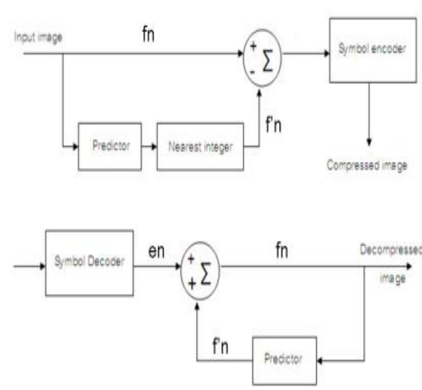

Fig. 1. Encoder/Decoder of predictive model [2]

VI. LOSSY

When dealing with lossy image compression, multiple methods are used in a sequence to achieve maximum optimal compression or in some cases allows the user to request how much compression they want. It is most ideal for images where the user is indifferent to the quality being exact and can permit the drop in quality for a savings in file size.

The compressor works on previous methods discussed, where it begins by removing pixel redundancies, quantizing the data to represent it with as few bits as possible, finally applying an encoding to the quantized data for further compression.

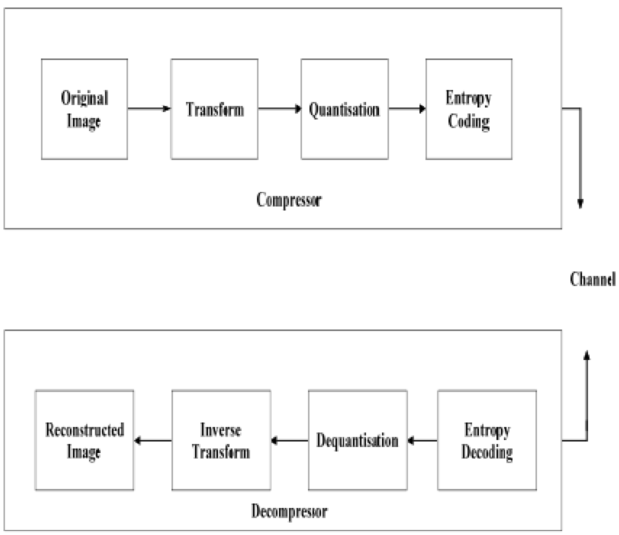

Fig. 3. Lossy Compressor and Decompressor [6]

A. Block Truncation Coding

This effective lossy algorithm involves dividing the image into N-by-N blocks of pixels. This algorithm is used mostly in greyscale images and involves calculating the mean and standard deviation of that block of pixels. A quantizer is used to reduce the grey levels, while maintaining the mean and standard deviation.

B. Discrete Cosine Transform (DCT)

The Discrete Cosine Transform algorithm is used in JPEG image compression and is also what JPEG was founded on. The algorithm makes use of representing the image data with sums of cosine functions with varying frequencies and magnitudes. This method combined with quantization is widely used and very effective in lossy compression.

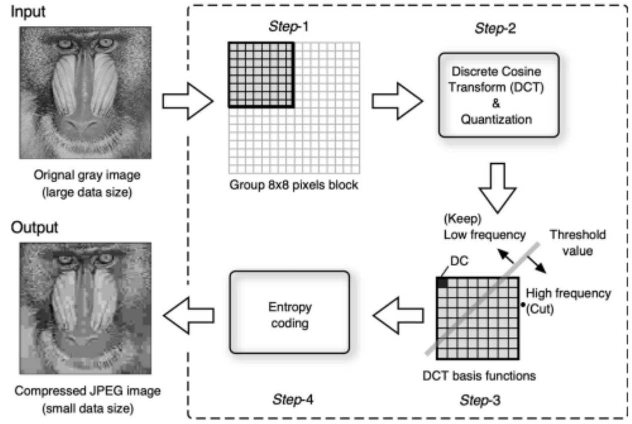

Figure 2 Process of DCT compression with threshold value(compression ratio) dictating the cutting of the high - frequency components [8]

C. Discrete Wavelet Transform (DWT)

Image data is transformed to a collection of wavelets. The image is divided into 4 sub images, each ¼ the original size of the image. The top right, bottom right and bottom left will contain the high frequency components of the original image.

The top left will contain the lower frequency components of the original image. This makes it such that it appears smoother. This upper left sub image can be used to approximate the original image.

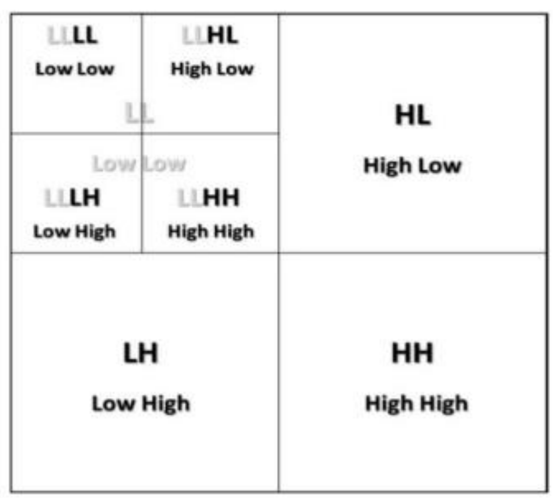

Figure 3 Decomposition into sub-bands (2 -level decomposition) [3]

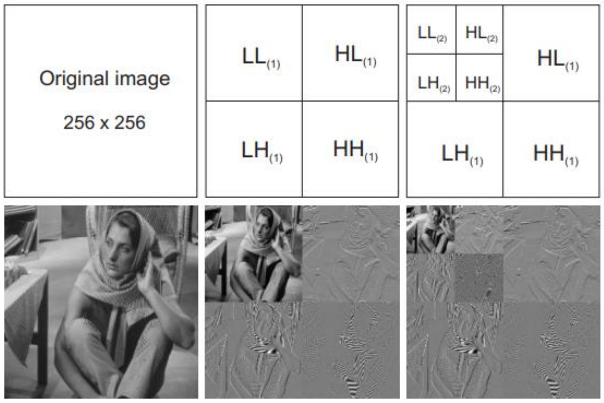

Figure 6 DWT processing) [7]

VII. Measurement

We can calculate the quality of the image, and the quality of the compression by using PSNR and MSE.

A. Peak Signal to Noise Ratio

PSNR measures the peak error between the actual and compressed image. The formula can be given by

$$PSNR(dB) = 10log_{10}(\frac{255^2}{MSE})$$

Where MSE is the Mean Squared Error and dB is decibels.

B. MSE

Mean Squared Error calculated by the cumulative squared error among the actual and compressed image. The formula is given below. M and N are the dimensions of the image, i(x,y) is the actual image and z(x,y) is the approximated image.

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x(i,j) - y(i,j)\right)^2$$

VIII. Conclusions

Image compression remains a much-researched topic with even more methods being introduced today building on combinations of new technologies and old compression techniques. As computing grows it is always worth revisiting compression algorithms especially for multi-media files such as images.

Various algorithms are given both with their benefits and cost as well as their associated compression tactic, i.e., lossy vs lossless. There are also formats that build upon these compression tactics with new formats being explored and researched all the time.

For future work I would like to explore more of these algorithms in a programmatic manner, while considering the measuring and comparison of the different methods. It is also worth exploring the combination of these algorithms and in conjunction with the fancier algorithms that are coming about in relation to machine learning and more advanced computing.

IX. References

[1] Rojatkar, Dinesh V., et al. "Image compression techniques: lossy and lossless." International Journal of Engineering Research and General Science 3.2 (2015): 912-917.

[2] Pensiri, Fuangfar, and Surapong Auwatanamongkol. "A lossless image compression algorithm using predictive coding based on quantized colors." WSEAS Transactions on signal processing 2 (2011).

[3] Kurniawan, Andri, Tito Waluyo Purboyo, and Anggunmeka Luhur Prasasti. "Implementation of image compression using discrete cosine transform (dct) and discrete wavelet transform (dwt)." International Journal of Applied Engineering Research 12.23 (2017): 13951-13958.

[4] Shantagiri, Pralhadrao V., and K. N. Saravanan. "Pixel size reduction loss-less image compression algorithm." International Journal of Computer Science & Information Technology 5.2 (2013): 87.

[5] Chiou, Paul T., Yu Sun, and G. S. Young. "A complexity analysis of the JPEG image compression algorithm." 2017 9th Computer Science and Electronic Engineering (CEEC). IEEE, 2017.

[6] Chawla, Sonal, Meenakshi Beri, and Ritu Mudgil. "Image compression techniques: a review." International Journal of Computer Science and Mobile Computing 3.8 (2014): 291-296.

[7] Costa, Daniel G., and Luiz Affonso Guedes. "A discrete wavelet transform (DWT)-based energy-efficient selective retransmission mechanism for wireless image sensor networks." *Journal of Sensor and Actuator Networks* 1.1 (2012): 3-35.

[8] Ri, S., et al. "Dynamic deformation measurement by the sampling Moiré method from video recording and its application to bridge engineering." *Experimental Techniques* 44.3 (2020): 313-327.

[9] Choudhary, Suman M., Anjali S. Patel, and Sonal J. Parmar. "Study of LZ77 and LZ78 Data Compression Techniques." *International Journal of Engineering Science and Innovative Technology (IJESIT)* 4.3 (2015).

[10] Kaur, Rajandeep, and Pooja Choudhary. "A review of image compression techniques." *Int. J. Comput. Appl* 142.1 (2016): 8-11.

[11] C. Taskin and S. K. Sarikoz, "An Overview of Image Compression Approaches," 2008 The Third International Conference on Digital Telecommunications (icdt 2008), 2008, pp. 174-179, doi: 10.1109/ICDT.2008.38.

[12] Vijayvargiya, Gaurav, Sanjay Silakari, and Rajeev Pandey. "A survey: various techniques of image compression." arXiv preprint arXiv:1311.6877 (2013).

[13] Singh, Manjari, et al. "Various image compression techniques: lossy and lossless." International Journal of Computer Applications 142.6 (2016): 23-26.

[14] Hussain, Abir Jaafar, Ali Al-Fayadh, and Naeem Radi. "Image compression techniques: A survey in lossless and lossy algorithms." Neurocomputing 300 (2018): 44-69.

[15] Kaushik, Abhishek, and Maneesha Gupta. "Analysis of image compression algorithms." International journal of engineering research and application (2012).

[16] Aguilera, Paula. "Comparison of different image compression formats." Wisconsin College of Engineering, ECE 533 (2006).

[17] Raid, A. M., et al. "Jpeg image compression using discrete cosine transform-A survey." arXiv preprint arXiv:1405.6147 (2014).

[18] Dudek, Gabriela, Przemysław Borys, and Zbigniew J. Grzywna. "Lossy dictionary-based image compression method." Image and Vision Computing 25.6 (2007): 883-889.

[19] Garg, Garima, and Raman Kumar. "Various Image Compression Techniques: A Review." IUP Journal of Telecommunications 14.1 (2022).

[20] Singh, Akhand Pratap, Anjali Potnis, and Abhineet Kumar. "A Review on Latest Techniques of Image Compression." International Research Journal of Engineering and Technology (IRJET) 3.7 (2016): 2395-0056.

[21] Dhawan, Sachin. "A review of image compression and comparison of its algorithms." International Journal of Electronics & Communication Technology, IJECT 2.1 (2011): 22-26.

[22] Al-Ani, Muzhir Shaban, and Fouad Hammadi Awad. "The JPEG image compression algorithm." International Journal of Advances in Engineering & Technology 6.3 (2013): 1055.

[23] Lin, Pao-Yen. "Basic image compression algorithm and introduction to JPEG standard." National Taiwan University, Taipei, Taiwan, ROC (2009).

[24] Shantagiri, Pralhadrao V., and K. N. Saravanan. "Pixel size reduction loss-less image compression algorithm." International Journal of Computer Science & Information Technology 5.2 (2013): 87.

[25] Jamil, Sonain, and Md Piran. "Learning-Driven Lossy Image Compression; A Comprehensive Survey." arXiv preprint arXiv:2201.09240 (2022).

[26] Theis, Lucas, et al. "Lossy image compression with compressive autoencoders." arXiv preprint arXiv:1703.00395 (2017).

[27] Alarabeyyat, A., et al. "Lossless image compression technique using combination methods." Journal of Software Engineering and Applications 5.10 (2012): 752.