Review on LC Method for JPEG Coded Photo Collection

1.SRIBACCHA NARSIMULU BE,MTech, UGC-NET,(PhD),
2.BANOTHU.SATYANARAYANA,B.Tech,M.Tech
1,2.Assistant Professor,Sriindu College of Engineering and Technology(Autonomous) ,Sheriguda Ibrahimpatnam.
narsiphdou@gmail.com,satyabanothu@gmail.com

***Abstract :**The explosion of digital photos has posed a significant challenge to photo storage and transmission for both personal devices and cloud platforms. In this project, we propose a novel lossless compression method to further reduce the size of a set of JPEG coded correlated images without any loss of information. The proposed method jointly removes inter/intra image redundancy in the feature, spatial, and frequency domains. For each collection, we first organize the images into a pseudo video by minimizing the global prediction cost in the feature domain. We then present a hybrid disparity compensation method to better exploit both the global and local correlations among the images in the spatial domain. Furthermore, the redundancy between each compensated signal and the corresponding target image is adaptively reduced in the frequency domain. Experimental results demonstrate the effectiveness of the proposed lossless compression method.*

I.INTRODUCTION

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. 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 colorless image is a grayscale 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. A bit plane of n bits can have 2n colors. The human eye can distinguish about 224 colors, although some claim that the number of colors 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 color 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 comprise 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. In our project, we have to propose a new coding method for lossless compression of JPEG coded image collections. Specifically, we compress a JPEG coded image collection by making use of both the inter correlation among images and the intra correlation within each image in the feature, spatial, and frequency domains jointly. Illustrates the architecture of our lossless encoder. For each input JPEG coded image collection (the JPEG icon we decode all JPEG files before further compression, resulting in the corresponding YUV image set. Then the prediction structure of the image set is determined based on the similarity between each pair of images in the feature domain. The prediction structure is formed in a tree structure generated from a directed graph via the minimum spanning tree (MST)algorithm in which parent nodes (i.e. images) can be used as references to predict their children. Section IV gives a detailed description of our feature domain determination of the prediction structure. Based on the prediction structure, we then exploit both the inter and intra redundancies in the spatial domain. For inter coded images, the disparity between each pair of target and reference images is reduced by joint global and local compensations in the pixel space. Specifically, larger geometric deformations and illumination differences are compensated by the global homographic and photometric transforms (illustrated by global compensation in Fig. 2), respectively, while smaller disparities are further compensated by the HEVC-like block based intra/inter prediction (local compensation. For the root image in each MST, the global compensation is bypassed and only intra prediction is performed. Transparency refers to the technique where certain pixels are layered on top of other pixels so that the bottom pixels will show through the top pixels. This is sometime useful in combining two images on top of each other. It is possible to use varying degrees of transparency, where the degree of transparency is known as an alpha value. In the context of the Web, this technique is often used to get an image to blend in well with the browser's background. Adding transparency can be as simple as choosing an unused color in the image to be the “special transparent” color, and wherever that color occurs, the program displaying the image knows to let the background show through.

The chapter is organized as follows. The prior encoding scheme and the Modified Viterbi Decoding Algorithm with some experimental results and performance analysis are presented in chapter 2. The conclusion and future work is presented in chapter 3.

II. JFEG COMPRESSION METHOD

JFEG is an image compression standard that was developed by the "Joint Photographic Experts Group". JPEG was formally accepted as an international standard in 1992. • JPEG is a lossy image compression method. It employs a transform coding method using the DCT (Discrete Cosine Transform). An image is a function of i and j (or conventionally x and y) in the spatial domain. The 2D DCT is used as one step in JPEG in order to yield a frequency response which is a function F(u, v) in the spatial frequency domain, indexed by two integers u and v. File Formats There are a large number of file formats (hundreds) used to represent an image, some more common then others. 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. The standard image format found in most paint, imaging, and desktop publishing programs. Supports 1- to 24- bit images and several different compression schemes. Developed by the Joint Photographic Experts Group, sometimes simply called the JPEG file format. It can store up to 24-bits of color. Some Web browsers can display JPEG images inline (in particular, Netscape can), but this feature is not a part of the HTML standard. The 2D DCT is used as one step in JPEG in order to yield a frequency response which is a function F(u, v) in the spatial frequency domain, indexed by two integers u and v.A lossy image compression method. We propose a novel compression scheme to further compress a set of JPEG coded correlated images without loss. Given a JPEG coded image set, we propose to remove both inter and intra redundancies by a hybrid prediction in the feature, spatial, and frequency domains. We first evaluate the pair-wise correlation between images by introducing the feature-based measurement so as to determine the prediction structure which is robust to scale, rotation, and illumination.

Compared with our preliminary work reported we not only provide more details and discussions of our scheme here, but more importantly we further improve the coding performance by introducing both the intra frame lossless compression algorithm and advanced entropy coding methods. Experimental results demonstrate the advantage of our scheme in terms of achieving much higher coding efficiency and lossless representation of JPEG coded files. Our scheme is able to greatly reduce the cost of storage and transmission of JPEG-coded image collections (e.g. geotagged images and personal albums) transparently for personal and cloud applications. In general, there are two main categories of compression. 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 then perfect reproductions of the original image. . 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.

JPEG-Compression

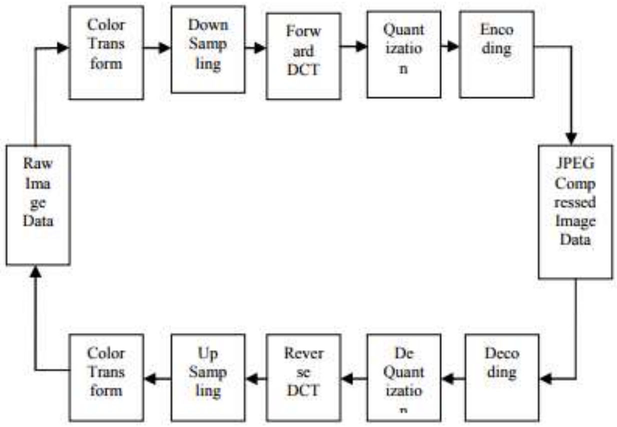

Fig.1. Block diagram of JFEG compression and decompression.

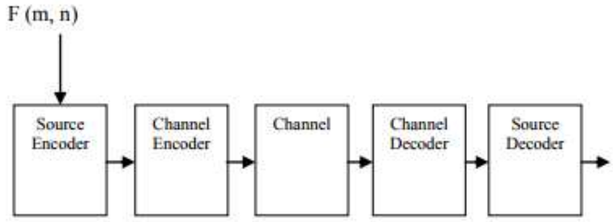

Fig .2. block diagram of image compression model

An 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 then 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. Image compression is thus essential for the efficient storage, retrieval and transmission of images. In general, there are two main categories of compression. 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 then 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. 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 n1 and n2 are the number of bits in two datasets that represent the same image, the relative redundancy of the first dataset is defined as: Rd=1/CR, where CR (the compression ratio) = n1/n2 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).

The Image Compression Model

Although 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. 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.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.

The JPEG Algorithm The Joint Photographic Experts Group developed the JPEG algorithm in the late 1980's and early 1990's. They developed this new algorithm to address the problems of that era, specifically the fact that consumer-level computers had enough processing power to manipulate and display full color photographs. However, full color photographs required a tremendous amount of bandwidth when transferred over a network connection, and required just as much space to store a local copy of the image. Other compression techniques had major tradeoffs. They had either very low amounts of compression, or major data loss in the image. Thus, the JPEG algorithm was created to compress photographs with minimal data loss and high compression ratios. Due to the nature of the compression algorithm, JPEG is excellent at compressing full-color (24-bit) photographs, or compressing grayscale photos that include many different shades of gray. The JPEG algorithm does not work well with web graphics, line art, scanned text, or other images with sharp transitions at the edges of objects. The reason this is so will become clear in the following sections. JPEG also features an adjustable compression ratio that lets a user determine the quality and size of the final image. Images may be highly compressed with lesser quality, or they may forego high compression, and instead be almost indistinguishable from the original. JPEG compression and decompression consist of 4 distinct and independent phases. First, the image is divided into 8 x 8 pixel blocks. Next, a discrete cosine transform is applied to each block to convert the information from the spatial domain to the frequency domain. After that, the frequency information is quantized to remove unnecessary information. Finally, standard compression techniques compress the final bit stream. This report will analyze the compression of a grayscale image, and will then extend the analysis to decompression and to color images. Phase One: Divide the Image Attempting to compress an entire image would not yield optimal results. Therefore, JPEG divides the image into matrices of 8 x 8 pixel blocks. This allows the algorithm to take advantage of the fact that similar colors tend to appear together in small parts of an image. Blocks begin at the upper left part of the image, and are created going towards the lower right. If the image dimensions are not multiples of 8, extra pixels are added to the bottom and right part of the image to pad it to the next multiple of 8 so that we create only full blocks. The dummy values are easily removed during decompression. From this point on, each block of 64 pixels is processed separately from the others, except during a small part of the final compression step. Phase one may optionally include a change in colorspace. Normally, 8 bits are used to represent one pixel. Each byte in a grayscale image may have the value of 0 (fully black) through 255 (fully white). Color images have 3 bytes per pixel, one for each component of red, green, and blue (RGB color). However, some operations are less complex if you convert these RGB values to a different color representation. Normally, JPEG will convert RGB colorspace to YCbCr colorspace. In YCbCr, Y is the luminance, which represents the intensity of the color. Cb and Cr are chrominance values, and they actually describe the color itself. YCbCr tends to compress more tightly than RGB, and any colorspace conversion can be done in linear time. The colorspace conversion may be done before we break the image into blocks; it is up to the implementation of the algorithm. Finally, the algorithm subtracts 128 from each byte in the 64- byte block. This changes the scale of the byte values from 0…255 to –128…127. Thus, the average value over a large set of pixels will tend towards zero. The following images show an example image, and that image divided into an 8 x 8 matrix of pixel blocks. The images are shown at double their original sizes, since blocks are only 8 pixels wide, which is extremely difficult to see. The image is 200 pixels by 220 pixels, which means that the image will be separated into 700 blocks, with some padding added to the bottom of the image. Also, remember that the division of an image is only a logical division, but in figure 1 lines are used to add clarity.

There are many algorithms that convert spatial information to the frequency domain. The most obvious of which is the Fast Fourier Transform (FFT). However, due to the fact that image information does not contain any imaginary components, there is an algorithm that is even faster than an FFT. The Discrete Cosine Transform (DCT) is derived from the FFT, however it requires fewer multiplications than the FFT since it works only with real numbers. Also, the DCT produces fewer significant coefficients in its result, which leads to greater compression. Finally, the DCT is made to work on one-dimensional data. Image data is given in blocks of two-dimensions, but we may add another summing term to the DCT to make the equation two-dimensional. In other words, applying the one-dimensional DCT once in the x direction and once in the y direction will effectively give a twodimensional discrete cosine transform. We begin examining this formula by realizing that only constants come before the brackets. Next, we realize that only 16 different cosine terms will be needed for each different pair of (u, v) values, so we may compute these ahead of time and then multiply the correct pair of cosine terms to the spatial-domain value for that pixel. There will be 64 additions in the two summations, one per pixel. Finally, we multiply the sum by the 3 constants to get the final value in the frequency matrix. This continues for all (u, v) pairs in the frequency matrix. Since u and v may be any value from 0…7, the frequency domain matrix is just as large as the spatial domain matrix. The frequency domain matrix contains values from -1024…1023. The upper-left entry, also known as the DC value, is the average of the entire block, and is the lowest frequency cosine coefficient. As you move right the coefficients represent cosine functions in the vertical direction that increase in frequency. Likewise, as you move down, the coefficients belong to increasing frequency cosine functions in the horizontal direction. The highest frequency values occur at the lower-right part of the matrix. The higher frequency values also have a natural tendency to be significantly smaller than the low frequency coefficients since they contribute much less to the image. Typically the entire lower-right half of the matrix is factored out after quantization. This essentially removes half of the data per block, which is one reason why JPEG is so efficient at compression. Computing the DCT is the most time-consuming part of JPEG compression. Thus, it determines the worst-case running time of the algorithm. The running time of the algorithm is discussed in detail later. However, there are many different implementations of the discrete cosine transform. Finding the most efficient one for the programmer's situation is key. There are implementations that can replace all multiplications with shift instructions and additions. Doing so can give dramatic speedups, however it often approximates values, and thus leads to a lower quality output image. There are also debates on how accurately certain DCT algorithms compute the cosine coefficients, and whether or not the resulting values have adequate precision for their situations. So any programmer should use caution when choosing an algorithm for computing a DCT, and should be aware of every trade-off that the algorithm has.There are many algorithms that convert spatial information to the frequency domain. The most obvious of which is the Fast Fourier Transform (FFT). However, due to the fact that image information does not contain any imaginary components, there is an algorithm that is even faster than an FFT.The Discrete Cosine Transform (DCT) is derived from the FFT.

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 the 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 bitrate 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 decorrelates 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.

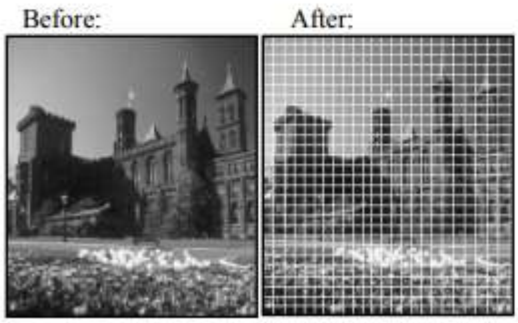

Fig .3. Example of Image Division

A flag in the coded bitstream signals whether the following bits are to be decoded with DCT algorithm or with a prediction algorithm. The semantics defined by MPEG can be applied to common video characteristics such as spatial redundancy, temporal redundancy, uniform motion, and spatial masking. In this compression schema, macroblock 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 macroblock 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. Picture coding macroblock types are (I, P, B). All (nonscalable) macroblocks within an I picture must be coded Intra (which MPEG encodes just like a baseline JPEG picture). However, macroblocks within a P picture may either be coded as Intra or Non-intra (temporally predicted from a previously reconstructed picture). Finally, macroblocks within the B picture can be independently selected as either Intra, Forward predicted, Backward predicted, or both forward and backward (Interpolated) predicted. The macroblock header contains an element, called macroblock_type, which can flip these modes on and off like switches.

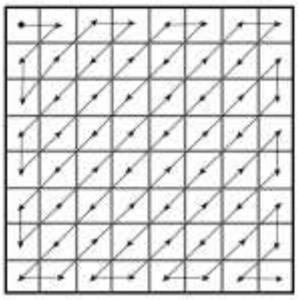

Fig.4. Zigzag Ordered Encoding

Fig .5. INPUT IMAGE, GRAYSCALE IMAGE ,AFTER APPLING DCT

In this compression schema, macroblock 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 macroblock 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.

III. 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 (n log(n)) 2 O 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. Here we develop image compression without lossless technique. Frequency based domain for image handling. In future we also involved hide some data in image it should be used to data conversion with secure system. In this paper, we focus on the efficient compression method for a set of clustered JPEG images. We notice that the clustering can be time consuming if one collection is too large. Possible solutions may involve advanced fast clustering methods and introducing assistant information. For example, we can make use of the time stamps or GPS information in the meta data of images to separate a large collection to smaller ones. We would like to pay attention to reduce the complexity of the clustering module for large scale image sets in our future work. Besides, the performance of our proposed scheme could be further improved in several ways. First, we could speed up the encoding and decoding process by introducing parallel techniques. Second, we can further reduce the complexity of the local compensation in our scheme by not only leveraging some fast algorithms proposed for HEVC but also reducing complexity by direct operating on the JPEG coded DCT coefficients. Finally, we notice that the featurebased distance approximation may not be always efficient. In the future, we would like to investigate advanced distance metrics in which the number as well as the overlapped area of matched features are taken into account. We may also introduce a light weight version of the distance metrics so that the pixel-domain distance between two images can be measured much more accurately at low computational cost.

REFERENCE

[1] Arivvkannama.M and Kurceren.R (2003) “The SP- and SI-frames designfor H.264/AVC,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 13, no. 7, pp. 637–644.

[2] Christopoulos.CSkodras.A, and Ebrahimi.T (2000) “The JPEG2000 still image coding system: an overview,” Consumer Electronics, IEEE Transactionson, vol. 46, no. 4, pp. 1103–1127, 2000.

[3] Dufaux.F, Sullivan.G.J, and Ebrahimi.T (2009) “The JPEG XR image codingstandard,” Signal Processing Magazine, IEEE, vol. 26, no. 6, pp. 195–199.

[4] MuzhirShaban.S and Zanuttigh.P (June 2015) “Compression of photo collections using geometrical information,” International Conference on, pp. 1–6.

[5] SachineDhanan.M.A and Bolles.R.C (1981) “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated Cartography,” Communications of the ACM, vol. 24, no. 6, pp. 381– 395.

[6] Sullivan G. J., J.-r. Ohm, W.-j. Han, and WiegandT. (2012) “Overview of theHigh Efficiency Video Coding (HEVC) Standard,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp. 1649–1668.

[7] WallaceG. K.(1992) “The JPEG still picture compression standard,” ConsumerElectronics, IEEE Transactions on, vol. 38, no. 1, pp. xviii– xxxiv.

[8] Weinberger M. J., Seroussi G., and Sapiro G. (2000) “The LOCO-I losslessimage compression algorithm: principles and standardization into JPEGLS,” Image Processing, IEEE Transactions on, vol. 9, no. 8, pp. 1309–1324.

[9] Wiegand, T. Sullivan G. J., Bjontegaard G., and LuthraA. (2003) “Overvie of the H. 264/AVC video coding standard,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 13, no. 7, pp. 560–576.

[10] Wu X. and MemonN. (1997) “Context-based, adaptive, lossless image coding,” Communications, IEEE Transactions on, vol. 45, no. 4, pp. 437–444.

[11] I. Matsuda, Y. Nomoto, K. Wakabayashi, and S. Itoh, “Lossless reencoding of JPEG images using block-adaptive intra prediction,” in Signal Processing Conference, 2008 16th European, Aug 2008, pp. 1–5.

[12] N. Ponomarenko, K. Egiazarian, V. Lukin, and J. Astola, “Additional lossless compression of JPEG images,” in Image

and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on, 2005, pp. 117–120.
[13] I. Bauermann and E. Steinbach, "Further lossless compression of JPEG Images," in Proc. Picture Coding Symposium (PCS 2004), 2004.
[14] M. Stirner and G. Seelmann, "Improved redundancy reduction for JPEG files," in Proc. Picture Coding Symposium (PCS 2007), 2007.
[15] G. Lakhani, "Modifying JPEG binary arithmetic codec for exploiting inter/intra-block and DCT coefficient sign redundancies," Image Processing, IEEE Transactions on, vol. 22, no. 4, pp. 1326–1339, 2013.
[16] D. Salomon and G. Motta, Handbook of data compression, 5th ed. Springer, 2009, ch. 5.15 PAQ, pp. 314–319.
[17] [Online]. Available: http://kb.winzip.com/kb/entry/72/
[18] [Online]. Available: http://www.elektronik.htw-aalen.de/packjpg/
[19] [Online]. Available: http://my.smithmicro.com/win/index.htm
[20] A. Schmieder, H. Cheng, and X. Li, "A study of clustering algorithms and validity for lossy image set compression." in Proc. The 2009 International conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'09), 2009, pp. 501–506.
[21] C.-P. Chen, C.-S. Chen, K.-L. Chung, H.-I. Lu, and G. Y. Tang, "Image set compression through minimal-cost prediction structures." in Image Processing, 2004. ICIP '04. 2004 International Conference on, 2004, pp. 1289–1292 Vol.2.
[22] Y. Lu, T.-T. Wong, and P.-A. Heng, "Digital photo similarity analysis in frequency domain and photo album compression," in Proc. The 3rd International Conference on Mobile and Ubiquitous Multimedia. ACM, 2004, pp. 237–244.
[23] O. Au, S. Li, R. Zou, W. Dai, and L. Sun, "Digital photo album compression based on global motion compensation and intra/inter prediction," in Audio, Language and Image Processing (ICALIP), 2012 International Conference on, 2012, pp. 84–90. [24] R. Zou, O. C. Au, G. Zhou, W. Dai, W. Hu, and P. Wan, "Personal photo album compression and management," in Circuits and Systems (ISCAS), 2013 IEEE International Symposium on, 2013, pp. 1428–1431.
[25] Z. Shi, X. Sun, and F. Wu, "Feature-based image set compression," in Multimedia and Expo (ICME), 2013 IEEE International Conference on, 2013, pp. 1–6.
[26] "Photo album compression for cloud storage using local features," Emerging and Selected Topics in Circuits and Systems, IEEE Journal on, vol. 4, no. 1, pp. 17–28, 2014.
[27] S. Milani and P. Zanuttigh, "Compression of photo collections using geometrical information," in Multimedia and Expo (ICME), 2015 IEEE International Conference on, June 2015, pp. 1–6.
[28] S. Milani, "Compression of multiple user photo galleries," Image and Vision Computing, 2016.
[29] Y. S. Musatenko and V. N. Kurashov, "Correlated image set compression system based on new fast efficient algorithm of Karhunen-Loeve transform," in Proc. SPIE, vol. 3527, 1998, pp. 518–529.
[30] K. Karadimitriou and J. M. Tyler, "The centroid method for compressing sets of similar images," Pattern Recognition Letters, vol. 19, no. 7, pp. 585–593, 1998.