Enhancing the JPEG Image Compression by Employing the Burrows-Wheeler Transformation

Yair Wiseman
Computer Science Department
Bar-Ilan University
Ramat-Gan 52900
Israel
wiseman@cs.biu.ac.il

ABSTRACT

The JPEG image compression standard employs as part of the compression algorithm Huffman codes. The developers of JPEG were aware that Huffman codes are not always the optimal option, so Huffman codes can be replaced by Arithmetic Coding. In this paper we suggest another replacement for Huffman codes – The transformation of Burrows and Wheeler. We show that using the Burrows and Wheeler algorithm can achieve an improved compression ratio for high quality images. If the image contains only a few colors, using the Burrows and Wheeler algorithm instead of Huffman codes will be even more advantageous.

Keywords: JPEG, Huffman Codes, BWT

1. INTRODUCTION

Images are actually a collection of pixels. Each pixel is defined by a numerical value, and is separately manipulated. There are many ways to store image data without degrading the original data. Some of these ways lose part of the data, but it is usually not noticeable. For example the BMP [1] standard is a well known format for image storage without data compression, whereas the GIF[2] standard is a well known format for image storage which uses the LZW [3] algorithm for compressing the image's data.

JPEG [4,5] encodes images using either Huffman coding [6] or Arithmetic coding [7]. Both of these codes are statistical codes. Usually, compressing texts by one of these techniques yields poor results. Texts of any language have certain rules of words' appearance. For example, in English the letter "q" is almost always followed by the letter "u" and the letter "z" is never followed by the letter "x". Even non-tightly connected languages like Hebrew have a few rules. Huffman coding and Arithmetic coding ignore the rules of words' appearance. Hence, their results are poorer. There are some improved versions of Huffman coding and arithmetic coding that have some ability to take the context into consideration; however, the version of Huffman coding and arithmetic coding that used by JPEG do not use these versions. Images do not have strict rules like human languages. Thus, apparently the use of a statistical coding cannot be considered as a disadvantage.

This paper claims that even though the data sequence in images does not recur precisely in the same format, there are some characteristics of data sequence which recurs. The temporal correlation which is captured by the DCT frequency coefficient in JPEG is not taken into account by current image compression methods implemented by JPEG. This correlation is left out; therefore valuable information is omitted. Hence, exploiting this information can improve the compression efficiency. Similar strategy is reinforced by the use of a dictionary compression method e.g. by the GIF standard and by the PNG standard. GIF uses LZW and PNG uses LZ77 and both get some nice results.

2. THE JPEG STANDARD

JPEG is a well known standardized image compression technique. JPEG loses information, so the decompressed picture is not the same as the original one. By adjusting the compression parameters, the degree of loss can be adjusted. The wide use of JPEG is because of two fundamental reasons: reducing the size of image files, and storing full color information.

Reducing image files will be an important procedure when we transmit files across networks or when we archive libraries. Usually, JPEG can remove the less important data before the compression, hence

JPEG will be able to compress images meaningfully, which produces a huge difference in the transmission time and the disk space.

The second advantage of JPEG is the capability of storing full color information: 24 bits/pixel or 16 million colors, while for example the GIF format, can store only 8 bits/pixel or 256 colors.

Here is a brief overview of the JPEG algorithm:

The first step transforms the image color into a suitable color space. There are several methods to transform the image into a color space [8,9]. The most common methods are the split into RGB components [10] or the split into YUV components [11]. These components are interleaved together within the compressed data. The ratio between these components is usually not one to one. When YUV components are used, usually the Y component will have a four times weight. The human eye is less sensitive to the frequency of chrominance information than to the frequency of luminance information which is represented by the Y component in the YUV format. Hence, the Y component gets a higher weight [12].

The second step groups the pixels into blocks of 8X8 pixels. Then, it transforms each block through a Forward Discrete Cosine Transform (FDCT) [13]. The DCT gives a frequency map, with 8X8 or 64 elements. The transformation keeps the low frequency information which a human eye is sensitive to. In each block the DCT coefficients are composed of:

- A single Direct Current (DC) coefficient number, which represents the average intensity level value in each block.
- The remaining 63 are named Alternating Current (AC) coefficients. They reflect the frequency information of their row and column.

The next step is the *quantization*. The 63 AC coefficients are ordered into a zig-zag sequence which arranges them into a one dimensional array. In each block, each of the 64 frequency components is divided by a separate "*quantization coefficient*". The quantization coefficients are set according to the desired image quality. The results of the division are rounded to integers. This step loses some information because of the rounding. Furthermore, it can be noted that even if the quantization coefficient is 1, some information will be lost, because typically the DCT coefficients are real numbers.

The last step encodes the reduced coefficients using either Huffman or Arithmetic coding. Usually a strong correlation appears between DC coefficients of adjacent 8X8 blocks. Therefore, JPEG encodes the difference between each pair of adjacent DC coefficient. Baseline JPEG model uses two different Huffman trees to encode the data, one for the DC coefficients' length and the other for the AC coefficients' length.

Finally, the compression parameters are written in the file header, so that the decoder module will be able to decode the compressed image. JPEG's procedure is summarized in Figure 1.

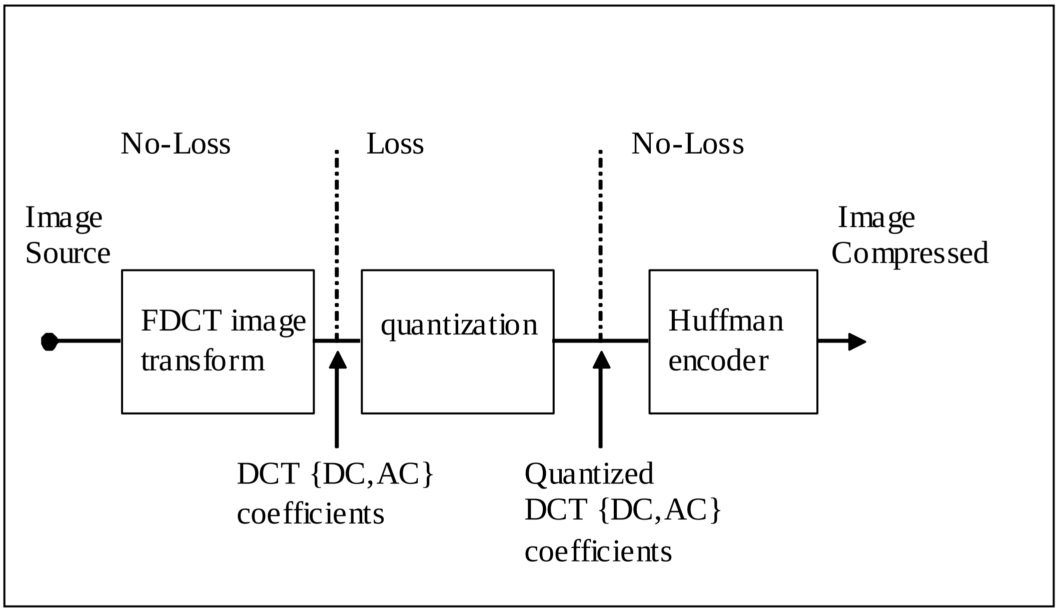

Figure 1: JPEG Model for a lossy image compression

The decompression process performs an inverse procedure. It decompresses the Huffman or the Arithmetic codes. Then, it makes the inversion of the Quantization step. In this stage, the decoder raises the small numbers by a multiplication of them by the quantization coefficients. The results are not accurate, but they are close to the original numbers of the DCT coefficients. Finally, an Inverse Discrete Cosine Transform (IDCT) is performed on the data received from the previous step.

JPEG has some disadvantages. Unfortunately, even with a JPEG viewer, it takes a longer time to decode and view a JPEG image than to view an image of a simpler format such as GIF, BMP, etc. Another disadvantage is the compression method that does not take into account the temporal correlation of the coefficients.

3. THE BURROWS-WHEELER TRANSFORMATION

The Burrows-Wheeler Transformation [14,15] is a context based method. The method utilizes repetitions of words' sequences in order to compress better similarly to dictionary methods. The Burrows-Wheeler Transformation does not lose any information in the compression procedure.

The traditional dictionary compression methods are the Lempel Ziv methods. WINZIP [16] of Windows and gzip[17] of UNIX use versions of the Lempel Ziv coding. Usually, the Burrows-Wheeler Transform gives better results than the versions of the Lempel Ziv coding. Hence, some Burrows-Wheeler compression utilities have been implemented for many environments [18] and nowadays the newly text compression utilities are based on Burrows-Wheeler algorithm e.g. Zzip [19] of Windows and bzip2 of UNIX [20]. The main deficiency of the Burrows-Wheeler Transform is the long time of execution.

The Burrows-Wheeler Transform has some basic steps:

The first step puts pointers to the file at each character. Then, the pointers are being sorted according to the characters which they are pointing to. The preceding characters of each of the pointers are sent to the next step according to the order of the sorted pointers. Actually, this sequence of characters in the output has the same characters as the original file, but the order of the characters is different.

The second step performs the *"move to front"* algorithm [21]. This algorithm keeps all 256 possible characters in a list. When a character is to be sent to the next step, its position in the list will be sent. After the sending the character will be moved from its current position in the list to the front of the list (position 0).

The next step applies a run-length coding to the output of the previous step. The output of the run-length coding is compressed by Huffman coding or Arithmetic coding.

The Burrows-Wheeler compression is quite slow because of the sorting, so usually the file is split into some blocks, in order to reduce the execution time. However, reducing the block size might cause a worse compression, because there will be fewer repetitions of strings. Burrows-Wheeler compression removes the temporal correlation and when there are fewer strings in a block, it will be harder to detect the correlation.

4. BURROWS-WHEELER BASED JPEG

Even though the repetitions of items' sequences in images do not exactly recur, the characteristics of the sequences in images repeat on themselves. Particularly, in JPEG images the AC coefficients in the beginning of a block are usually higher than the AC coefficients in the end of a block. In fact, most of the last AC coefficients in a block are zero.

In [22,23] the authors suggest to check what is the "degree of repetition". If the degree of repetition is high enough, they suggest to omit the run-length encoding and the entropy encoding of JPEG. Instead, they suggest to apply the Burrows- Wheeler Transform. However, their definition of "degree of repetition" is unclear. Moreover, the motivation for the omission of JPEG's run-length coding is unclear. JPEG's run-length coding is designed to deal with the significant number of zeros usually appearing in JPEG and their tendency to be located in the end of the block. Burrows-Wheeler's run-length coding does not have this advantage. Apparently, this replacement can damage the compression. Thus, we did not succeed to obtain good results using their method.

We would like to suggest another algorithm:

- Transform the image color into a suitable color space.
- Apply DCT.
- Perform the quantization procedure.
- Instead of assigning the Huffman codes to each DC value and each pair of zeros' length and an AC value, assign a number to each of them.
- Process the above numbers by the Burrows-Wheeler transform.
- Process the output by Move-to-front procedure and Run-length coding.
- Finally compress by Arithmetic coding.

The concept of this technique is performing the DCT and the quantization as they are done in the original JPEG algorithm. Then, we take the run-length scheme of JPEG including the zigzag order, but instead of giving variable length codes as Huffman coding does, we give fix length codes. These fixed length codes are the input of Burrow-Wheeler compression. The suggested algorithm is depicted in Figure 2.

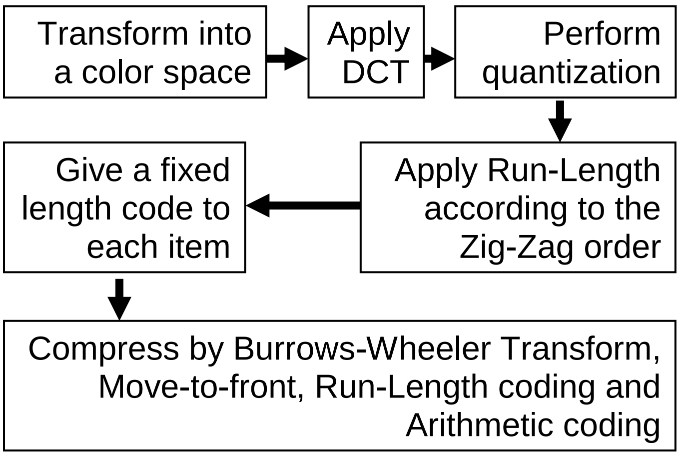

Figure 2: The suggested algorithm

5. WHY CAN BURROWS-WHEELER BE APPLIED?

In JPEG format the source image samples are grouped into 8X8 blocks, shifted from unsigned integer to signed integer, and Forward DCT is applied to them. The DCT-based compression can be viewing the FDCT as a harmonic analyzer and the IDCT as a harmonic synthesizer. In JPEG format, each 8X8 block of the source image samples is effectively a 64-point discrete signal which is a function of the 2D dimensional space *x*, and *y*. The FDCT takes such a signal as its input and decomposes it into 64 orthogonal basis signals. The output of the FDCT is the set of unique 64 basis signal amplitudes, which can be regarded as the relative amount of the 2D spatial frequency contained in the 64-point input signal.

The results of these mathematic calculations are treated as integers. Moreover, usually, these results are divided by some numbers and the quotients are accommodated as integers. This dividing and casting into integers is called "quantization".

Obviously, rounding floating-point numbers to integers causes some loss of data, so the values of JPEG's coefficients are inaccurate; however, objects in reality tend to have low frequencies values. So, long strings with low values may recur. Since the values are inaccurate, small differences will be omitted and as a result, many strings will be totally equal.

The output of the quantization goes into the Burrows-Wheeler Transform. The Burrows-Wheeler algorithm will achieve a better compression ratio because of the equal strings, so the inaccuracy of the data becomes an advantage.

In [24] the authors suggest to apply Burrows-Wheeler to a wavelet transformation similar to the DCT of JPEG, but without a quantization. As explained above this yields worse results.

It should be noted that unlike Huffman coding or arithmetic coding, Burrows-Wheeler Transform has the ability to take into its considerations the context inside the processed block. The sorting stage makes Burrows-Wheeler Transform a context-aware compression, whereas Huffman coding treats each value separately.

6. EXPERIMENTAL RESULTS

The method has been tested on some images. The images can be seen in Figure 3. The original images are very large e.g. the upper left image is 5MB in BMP format. The images were compressed by the original JPEG method and by the Burrows-Wheeler based JPEG.

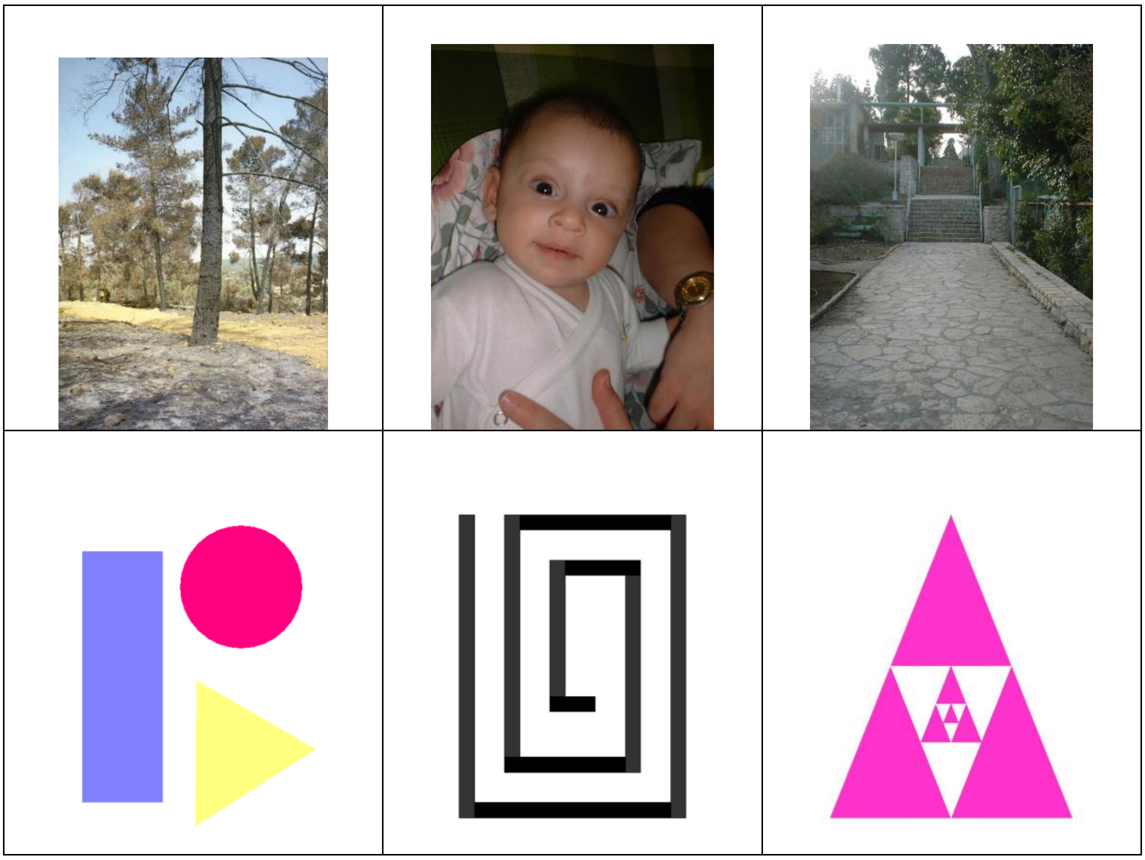

Figure 3: The tested images

The version of the Burrows-Wheeler Transform was taken from [20]. The results of the compression efficiency are given in Figure 4. The results are of the upper images in Figure 3, but almost the same results have been obtained when using other real life images. The chunks of data, which have been used by the Burrows-Wheeler Transformation, were from 100KB to 900KB. The size of chunks has no influence on Huffman coding, so the original JPEG compression is not influenced by the different sizes. However, Burrows-Wheeler is influenced by the size of chunks and the influence can be seen in the results. 90, 95 and 100 are the JPEG parameter of the image quality. Obviously, reducing the quality of an image will diminish the image size.

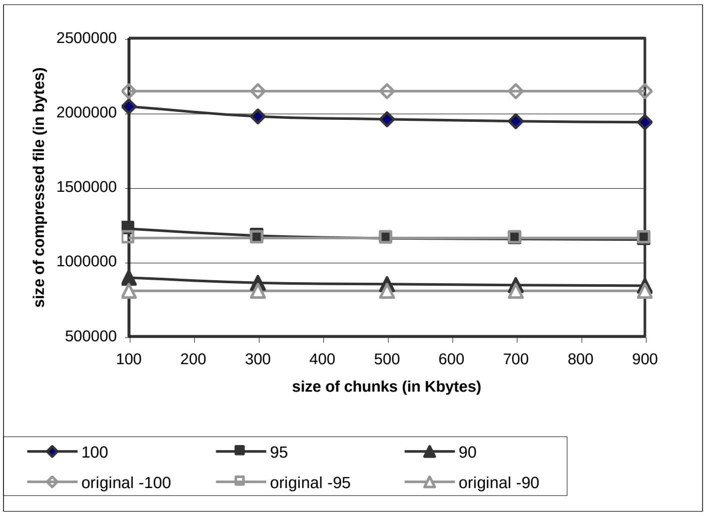

Figure 4: Compression efficiency of real life images

When using larger chunks the compression obviously will become better, whereas using smaller chunks will yield poorer compression. When the image quality is below 90, the size of the compressed blocks will become too small. Most of the data will be long lists of zeros that will be represented by a few bits. This leads to very short strings of data. Hence a fewer repetitions occur, thus the compression becomes poorer. In such cases the original Huffman should be applied in order to get better results. The Burrows-Wheeler based JPEG should be applied just for high quality real life images.

It should be noted that real life images usually yield poor compression ratio if high quality is need. Original JPEG with the highest parameter of quality – 100, reduces the image size into 41.86% of the original size, whereas Burrows-Wheeler based JPEG reduces the image size into 37.79% of the original size. Both of them are a lossy compression method. Comparing these results to lossless compression methods like PNG [25] shows that the loss of data is still worthy. PNG (Deflate) reduces the image size into only 94.20% of the original size.

JPEG-2000 [26] is a newly published standard. However, as shown in [27], high-quality images produce better compression ratio using the traditional JPEG. JPEG-2000 is used for fair quality and poor quality images. Since our tool is aimed for high quality images, we use the traditional JPEG. Moreover, the traditional JPEG is still widely used.

In [28] the author claims that arithmetic coding based JPEG creates 5 to 10 percent smaller files than Huffman coding based JPEG. These results have been obtained using the Q-Coder [29] – a specific variant of arithmetic coding employed by JPEG. Unfortunately, this variant is subject to a patent of IBM, AT&T, and Mitsubishi. In addition, the arithmetic coding can save 5 to 10 percent when the picture quality is not high, because low quality makes the variety of AC elements smaller and in such a case the arithmetic coding can almost reach the entropy, whereas Huffman coding can reach the entropy just when there exists a big variety of AC elements. Our paper aims to improve the compression of high quality pictures where there is no big difference between Huffman coding and arithmetic coding.

Unlike real life images, synthetic images have much more repetitions; hence the results are significantly better even for poor quality images. The lower images results in Figure 3 can be seen in Figure 5. The results are better, even when using small chunks for Burrows-Wheeler. It can be seen that there seems to be a near-constant improve in the compression. This suggests that there are strong temporal correlations in the zigzag scans of the synthetic data. The results are of the traditional JPEG, however, JPEG-2000 will compress almost the same way when the compressed image contains no fine details and each shape within the image is of one color [26].

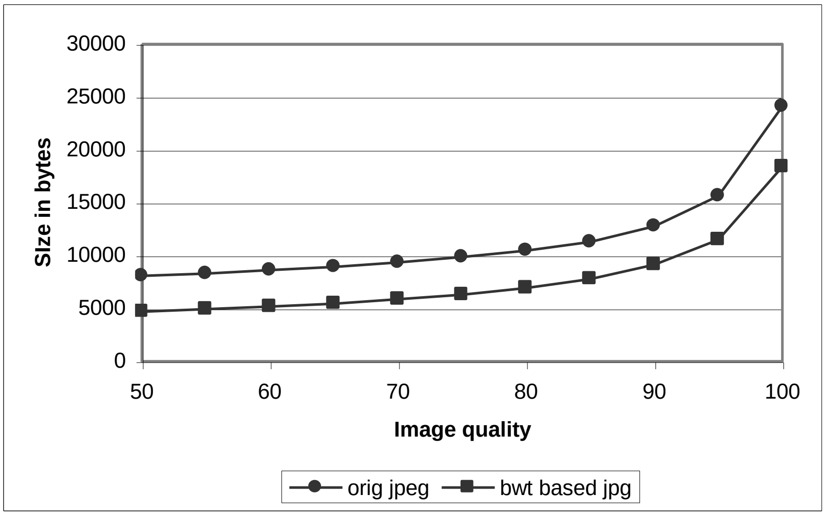

Figure 5: Compression efficiency of synthetic images

The lossless PNG compression technique [25] is well-suited for synthetic images and PNG indeed outperforms both the original JPEG and Burrows-Wheeler based JPEG, if JPEG specifies high quality images. In point of fact, original JPEG has to specify at most 78 as its parameter of quality and Burrows-Wheeler based JPEG has to specify at most 92 as its parameter of quality in order to get better results than PNG.

The main deficiency of Burrow-Wheeler based JPEG is the execution time. Burrow-Wheeler is much more time consuming than Huffman. The real life images in Figure 3 can be compressed in about 3 seconds by the original JPEG on a SUN Ultra-4 Sparc machine. Burrows-Wheeler based JPEG will compress the same data in about double time. The decompression, however, consumes roughly just one second, which is expedient, since usually an image is created just one time, but is seen several times.

7. CONCLUSIONS

The test' results show potential benefits and advantages. High quality images are very important for some implementations such as astronomic data or military data. In such cases Burrows-Wheeler based JPEG can yield better results. The synthetic images do not require high quality. Synthetic images are very common e.g. in companies' logos. The Burrow-Wheeler based JPEG could be used to reduce the size of Internet sites with synthetic images. The main deficiency of the Burrows-Wheeler based JPEG is the long compression time. However, nowadays computers have much stronger processors that can perform even complicated tasks in a reasonable time.

8. REFERENCES

[1] Luse M. The BMP File Format *Dr. Dobb's Journal, Vol 9, Issue 10, pp. 18-22* 1994.
[2] Willard L., Lempel A., Ziv J. and Cohn M. Apparatus and method for compressing data signals and restoring the compressed data signals *US patent - US4464650* 1984.
[3] Welch T. A. A Technique for High-Performance Data Compression *IEEE Computer Society* 17 pp. 8-19 1984.
[4] Wallace G. K. *The JPEG Still Picture Compression Standard* Communication of the ACM 34, pp. 3-44, 1991.
[5] Information Technology *Digital Compression and Coding of Continuous-Tone Still Images Requirements and Guidelines* International Standard ISO/IEC 10918-1, 1993.
[6] Huffman D. A method for the Construction of Minimum Redundancy Codes *Proc. of the IRE 40, pp.1098-1101* 1952.
[7] Witten I. H., Neal R. M. and Cleary J. G. Arithmetic Coding for Data Compression, *Communication of the ACM 30, pp. 520-540* 1987.
[8] Hearn D. and Baker M. P. Computer Graphic *Prentice Hall, Englewood Cliffs, NJ, pp. 295-307* 1986.

[9] Jain A. K. Fundamental of Digital Image Processing *Prentice Hall Information and Sytem Sciences Series, pp. 553-557* 1986.
[10] Hunt R. W. G. The Reproduction of Colour *Fountain Press England, pp. 507-511* 1995.
[11] Laplante P. A. and Stoyenko A. D. Real Time Imaging, Theory, Techniques and Applications *IEEE Press Inc. NY, pp. 122-124* 1996.
[12] Awcock G. J. Applied Image Processing *McGraw-Hill Book Company, pp. 260-261* 1996.
[13] Rao K. R. and Yip P. Discrete Cosine Transform Algorithms, Advantages, *Applications Academic Press Inc., London* 1990.
[14] Burrows M. and Wheeler D. Block sorting Lossless Data Compression Algorithm, *System research center, research report 124, Digital System research Center, Palo Alto, CA* 1994.
[15] Nelson M. R. Data Compression with the Burrows Wheeler Transformation, *Dr. Dobb's Journal, pp. 46-50* 1996.
[16] WinZip, *Nico Mak Computing, Inc., Mansfield, CT, USA* 1998.
[17] gzip, *Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA, USA* 1991.
[18] Manzini G., The Burrows-Wheeler Transform: Theory and Practice, *Lecture Notes in Computer Science, Springer Verlag, Volume 1672, pp. 34-47* 1999.
[19] Debin D., http://debin.net/zzip/index.php, June 2002.
[20] SGI® IRIX® Freeware distribution, *1600 Amphitheatre Pkwy. Mountain View, CA, USA*, Edition of February 2003.
[21] Elias P., Interval and recency rank source coding: two on-line adaptive variable-length schemes, IEEE Transactions on Information Theory, Vol. 33 pp. 3-10, 1987.
[22] Baik H. K., Ha D. S., Yook H. G., Shin S. C. and Park M. S., Selective Application of Burrows-Wheeler Transformation for Enhancement of JPEG Entropy Coding, *Proceedings of International Conference on Information, Communications & Signal Processing, PN.230* 1999.
[23] Baik H. K., Yook H. G., Shin S. C., Park M. S. and Ha D. S, A New Method to Improve the Performance of JPEG Entropy Coding Using Burrows-Wheeler Transformation, International Symposium on Computer and Information Sciences, pp. 502-509, Turkey, October 1999.
[24] Guo H., Burrus C. S., Waveform and Image Compression Using the Burrows Wheeler Transform and the Wavelet Transfor,. Proceedings 1997 International Conference on Image Processing (ICIP '97), pp. 65-68, Washington, DC, October 26-29, 1997.
[25] Duce D. et al, Portable Network Graphics (PNG) Specification (Second Edition), Information technology ISO/IEC 15948:2003, Oxford Brookes University, November 2003.
[26] Information Technology *JPEG 2000 Image Coding System,* International Standard ISO/IEC 15444-1:2004, 2004.
[27] Ebrahimi F., Chamik M., Winkler S., JPEG vs. JPEG2000: An objective comparison of image encoding quality, *Proceedings of SPIE Applications of Digital Image Processing, vol. 5558, pp. 300-308, Denver, CO, August 2-6*, 2004.
[28] Lane T., JPEG image compression FAQ, http://www.faqs.org/faqs/jpeg-faq/part1/, 1999.
[29] Michel J. L., and Pennebaker W. B., Software Implementation of the Q-Coder, IBM J. Research and Development, vol. 32, pp. 753-774, Nov. 1988.
[30] Wiseman Y., (2020), "Autonomous Vehicles", Encyclopedia of Information Science and Technology, Fifth Edition, Vol. 1, Chapter 1, pp. 1-11.
[31] Wiseman, Y., (2018), "In an era of autonomous vehicles, rails are obsolete", International Journal of Control and Automation, Vol. 11, No. 2, pp. 151-160.
[32] Wiseman, Y., (2021), "Intelligent Transportation Systems along with the COVID-19 Pandemic will Significantly Change the Transportation Market", The Open Transportation Journal, Vol. 15, No. 1, pp. 11-15.
[33] Wiseman Y., (2018), "Vehicle identification by OCR, RFID and Bluetooth for toll roads", International Journal of Control and Automation, Vol. 11, No. 9, pp. 67-76.
[34] Wiseman Y., (2020), "Conjoint Vehicle License Plate Identification System", The Open Transportation Journal, Vol. 14, No. 1, pp. 164-173.
[35] Wiseman, Y., (2021), "COVID-19 Along with Autonomous Vehicles will Put an End to Rail Systems in Isolated Territories", In IEEE Intelligent Transportation Systems, Vol. 13, No. 3, pp. 6-12, doi: 10.1109/MITS.2021.3049409.
[36] Wiseman, Y. (2019), "Driverless cars will make union stations obsolete", The Open Transportation Journal, Vol. 13, No. 1, pp. 109b 115.

[37] Wiseman Y., (2017, May). "Real-time monitoring of traffic congestions" In proceedings of 2017 IEEE International Conference on Electro Information Technology (EIT-2017, Lincoln, Nebraska, USA, pp. 501-505.
[38] Wiseman, Y., (2017), "Tool for online observing of traffic congestions", International Journal of Control and Automation, Vol. 10, No. 6, pp. 27-34.
[39] Wiseman Y., (2017), "Computerized traffic congestion detection system". International Journal of Transportation and Logistics Management, Vol. 1, No.1, pp. 1-8.
[40] Wiseman Y., (2018)., "Efficient Embedded Computing Component for Anti-Lock Braking System", International Journal of Control and Automation, Vol. 11, No. 12, pp. 1-10.
[41] Wiseman Y., (2018), "Ancillary ultrasonic rangefinder for autonomous vehicles", International Journal of Security and its Applications", Vol. 12, No. 5, pp. 49-58.
[42] Wiseman, Y. (2019), "Driverless cars will make passenger rail obsolete [opinion
[43] Wiseman Y., (2017), "Self-Driving Car - A Computer will Park for You", International Journal of Engineering & Technology for Automobile Security, Vol. 1, No. 1, pp. 9-16.
[44] Wiseman Y., (2017), "Remote Parking for Autonomous Vehicles", International Journal of Hybrid Information Technology, Vol. 10, No. 1, pp. 313-324.
[45] Wiseman Y., (2014), "Device for Detection of Fuselage Defective Parts", Information Journal, Tokyo, Japan, Vol. 17(9(A)), pp. 4189-4194.
[46] Wiseman Y., (2013), "Fuselage Damage Locator System", Advanced Science and Technology Letters, Vol. 37, pp. 1-4.
[47] Wiseman Y., (2010), "Take a Picture of Your Tire!", Proc. IEEE Conference on Vehicular Electronics and Safety (IEEE ICVES-2010) Qingdao, ShanDong, China, pp. 151-156.
[48] Wiseman Y., (2013), "The Effectiveness of JPEG Images Produced By a Standard Digital Camera to Detect Damaged Tyres", World Review of Intermodal Transportation Research, Vol. 4, No. 1, pp. 23-36.
[49] Wiseman Y., (2013), "Camera That Takes Pictures of Aircraft and Ground Vehicle Tires Can Save Lives", Journal of Electronic Imaging, Vol. 22, No. 4, 041104.
[50] Wiseman Y., (2017), "Safety Mechanism for SkyTran Tracks", International Journal of Control and Automation, Vol. 10, No. 7, pp. 51-60.
[51] Wiseman Y., (2017), "Automatic Persistent Inspection of SkyTran Track System", http://u.cs.biu.ac.il/~wiseman/skytran1.pdf .
[52] Grinberg I. and Wiseman Y., (2007), "Scalable Parallel Collision Detection Simulation", In Proceedings of Signal and Image Processing, Honolulu, Hawaii, pp. 380-385.
[53] Grinberg I. and Wiseman Y., (2013), "Scalable Parallel Simulator for Vehicular Collision Detection", International Journal of Vehicle Systems Modelling and Testing, Inderscience Publication, Vol. 8, No. 2, pp. 119-144.
[54] Wiseman Y., K. Schwan and P. Widener, (2004), "Efficient End to End Data Exchange Using Configurable Compression", Proceedings of The 24th IEEE Conference on Distributed Computing Systems (ICDCS 2004), Tokyo, Japan, pp. 228-235.
[55] P. Weisberg and Wiseman Y., (2009), "Using 4KB Page Size for Virtual Memory is Obsolete", Proc. IEEE Conference on Information Reuse and Integration (IEEE IRI-2009), Las Vegas, Nevada, pp. 262-265.
[56] P. Weisberg and Wiseman Y., (2015), "Virtual Memory Systems Should Use Larger Pages rather than the Traditional 4KB Pages", International Journal of Hybrid Information Technology, Vol. 8(8), pp. 57-68.
[57] Wiseman Y., (2017), "Automatic Alert System for Worn Out Pipes in Autonomous Vehicles", International Journal of Advanced Science and Technology, Vol. 107, pp. 73-84.
[58] Wiseman Y. and Grinberg I., (2016), "When an Inescapable Accident of Autonomous Vehicles is Looming", International Journal of Control and Automation, Vol. 9 No. 6, pp. 297-308.
[59] Wiseman Y. and Grinberg I., (2016), "Autonomous Vehicles Should Not Collide Carelessly", Advanced Science and Technology Letters, Vol. 133, pp. 223-228.
[60] Wiseman Y. and Grinberg I., (2016), "Circumspectly Crash of Autonomous Vehicles", Proceedings of IEEE International Conference on Electro Information Technology (EIT 2016), Grand Forks, North Dakota, USA, pp. 382-386.
[61] Y. Wiseman, "Diminution of JPEG Error Effects", The Seventh International Conference on Future Generation Information Technology, Vol. 117, pp. 6-9, (2015).
[62] Y. Wiseman, "Alleviation of JPEG Inaccuracy Appearance", International Journal of Multimedia and Ubiquitous Engineering, Vol. 11(3), pp. 133-142, (2016).

[63] Y. Wiseman, "Enhancement of JPEG compression for GPS images", International Journal of Multimedia and Ubiquitous Engineering, Vol. 10, No. 7, pp. 255-264, (2015).
[64] Y. Wiseman, "Improved JPEG Based GPS Picture Compression", Advanced Science and Technology Letters, (2015).
[65] Y. Wiseman, "The still image lossy compression standard - JPEG", Encyclopedia of Information Science and Technology, Third Edition, Vol. 1, Chapter 28, (2014).
[66] Y. Wiseman, "A Pipeline Chip for Quasi Arithmetic Coding", IEICE Journal - Trans. Fundamentals, Tokyo, Japan, Vol. E84-A No.4, pp. 1034-1041, (2001).
[67] Y. Wiseman, "Burrows-Wheeler Based JPEG", Data Science Journal, Vol. 6, pp. 19-27, (2007).
[68] Y. Wiseman, "Efficient Embedded Images in Portable Document Format (PDF)", International Journal of Advanced Science and Technology, Vol. 124, pp. 129-138, (2019).
[69] Y. Wiseman and E. Fredj, "Contour Extraction of Compressed JPEG Images", ACM - Journal of Graphic Tools, Vol. 6, No. 3, pp. 37-43, (2001).
[70] E. Fredj and Y. Wiseman, "An O(n) Algorithm for Edge Detection in Photos Compressed by JPEG Format", Proc. International Conference on Signal and Image Processing SIP-2001, Honolulu, Hawaii, pp. 304-308, (2001).
[71] Y. Wiseman, "Adjustable and Automatic Flush Toilet", International Journal of Control and Automation, Vol. 13, No. 4, pp. 1-10, (2020).
[72] D. Livshits and Y. Wiseman, "Cache Based Dynamic Memory Management for GPS", Proceedings of IEEE Conference on Industrial Electronics (IEEE ICIT-2011), Auburn, Alabama, pp. 441-446, (2011).
[73] D. Livshits and Y. Wiseman, "The Next Generation GPS Memory Management", International Journal of Vehicle Information and Communication Systems, Vol. 3(1), pp. 58-70, (2013).
[74] R. B. Yehezkael, Y. Wiseman, H. G. Mendelbaum & I. L. Gordin, "Experiments in Separating Computational Algorithm from Program Distribution and Communication", LNCS of Springer Verlag Vol. 1947, pp. 268-278, 2001.
[75] Y. Wiseman, "ARC Based SuperPaging", Operating Systems Review, Vol. 39(2), pp. 74-78, 2005.
[76] Y. Wiseman, "Advanced Non-Distributed Operating Systems Course", ACM - Computer Science Education, Vol. 37(2), pp. 65-69, 2005.
[77] M. Reuven & Y. Wiseman, "Reducing the Thrashing Effect Using Bin Packing", Proc. IASTED Modeling, Simulation, and Optimization Conference, MSO-2005, Oranjestad, Aruba, pp. 5-10, 2005.
[78] M. Reuven & Y. Wiseman, "Medium-Term Scheduler as a Solution for the Thrashing Effect", The Computer Journal, Oxford University Press, Swindon, UK, Vol. 49(3), pp. 297-309, 2006.
[79] Y. Wiseman, "The Relative Efficiency of LZW and LZSS", Data Science Journal, Vol. 6, pp. 1-6, 2007.
[80] Y. Wiseman & I. Gefner, "Conjugation Based Compression for Hebrew Texts", ACM Transactions on Asian Language Information Processing, Vol .6(1), article no. 4, 2007.
[81] I. Grinberg & Y. Wiseman, "Scalable Parallel Collision Detection Simulation", Proc. Signal and Image Processing (SIP-2007), Honolulu, Hawaii, pp. 380-385, 2007.
[82] Y. Wiseman, "ASOSI: Asymmetric Operating System Infrastructure", Proc. 21st Conference on Parallel and Distributed Computing and Communication Systems, (PDCCS 2008), New Orleans, Louisiana, pp. 193-198, 2008.
[83] Y. Wiseman, J. Isaacson & E. Lubovsky, "Eliminating the Threat of Kernel Stack Overflows", Proc. IEEE Conference on Information Reuse and Integration (IEEE IRI-2008), Las Vegas, Nevada, pp. 116-121, 2008.
[84] M. Itshak & Y. Wiseman, "AMSQM: Adaptive Multiple SuperPage Queue Management", Proc. IEEE Conference on Information Reuse and Integration (IEEE IRI-2008), Las Vegas, Nevada, pp. 52-57, 2008.
[85] R. Ben Yehuda & Y. Wiseman, "The Offline Scheduler for Embedded Transportation Systems", Proc. IEEE Conference on Industrial Electronics (IEEE ICIT-2011), Auburn, Alabama, pp. 449-454, 2011.
[86] Y. Wiseman & P. Weisberg, "Economical Memory Management for Avionics Systems", IEEE/AIAA 31st Digital Avionics Systems Conference (DASC), 2013.
[87] Y. Wiseman & Alon Barkai, "Diminishing Flight Data Recorder Size", IEEE/AIAA 31st Digital Avionics Systems Conference (DASC), 2013.
[88] R. Ben Yehuda & Y. Wiseman, "The Offline Scheduler for Embedded Vehicular Systems", International Journal of Vehicle Information and Communication Systems, Vol. 3(1), pp. 44-57, 2013.

[89] Y. Wiseman & Alon Barkai, "Smaller Flight Data Recorders", Journal of Aviation Technology and Engineering, Vol. 2(2), pp. 45-55, 2013.
[90] P. Weisberg & Y. Wiseman, "Efficient Memory Control for Avionics and Embedded Systems", International Journal of Embedded Systems, Vol. 5(4), pp. 225-238, 2013.
[91] Y. Wiseman, "Steganography Based Seaport Security Communication System", Advanced Science and Technology Letters, Vol. 46, pp. 302-306, 2014.
[92] P. Weisberg, Y. Wiseman & J. Isaacson, "Enhancing Transportation System Networks Reliability by Securer Operating System", Open Journal of Information Security and Applications, Vol. 1(1), pp. 24-33, 2014.
[93] Y. Wiseman, "Noise Abatement at Ben-Gurion International Airport", Advanced Science and Technology Letters, Vol. 67, pp. 84-87, 2014.
[94] Y. Wiseman, "Protecting Seaport Communication System by Steganography Based Procedures", International Journal of Security and Its Applications, Sandy Bay, Tasmania, Australia, Vol. 8(4), pp. 25-36, 2014.
[95] Y. Wiseman, "Noise Abatement Solutions for Ben-Gurion International Airport", International Journal of U- & E-Service, Science & Technology, Vol. 7(6), pp. 265-272, 2014.
[96] P. Weisberg & Y. Wiseman, "Virtual Memory Systems Should use Larger Pages", Advanced Science and Technology Letters, Vol. 106, pp. 1-4, 2015.
[97] Y. Wiseman & Y. Giat, "Red Sea and Mediterranean Sea Land Bridge via Eilat", World Review of Intermodal Transportation Research, Vol. 5(4), pp. 353-368, 2015.
[98] Y. Wiseman, "Can Flight Data Recorder Memory Be Stored on the Cloud?", Journal of Aviation Technology and Engineering, Vol. 6(1), 16-24, 2016.
[99] Y. Wiseman & Y. Giat, "Multi-modal passenger security in Israel", Multimodal Security in Passenger and Freight Transportation: Frameworks and Policy Applications, Edward Elgar Publishing Limited, Chapter 16, pp. 246-260, 2016.
[100] Y. Wiseman, "Traffic Light with Inductive Detector Loops and Diverse Time Periods", Contemporary Research Trend of IT Convergence Technology, Vol. 4, pp. 166-170, 2016.
[101] Y. Wiseman, "Unlimited and Protected Memory for Flight Data Recorders", Aircraft Engineering and Aerospace Technology, Vol. 88(6), pp. 866-872, 2016.
[102] Y. Wiseman, "Conceptual Design of Intelligent Traffic Light Controller", International Journal of Control and Automation, Vol. 9(7), pp. 251-262, 2016.
[103] Y. Wiseman, "Compression Scheme for RFID Equipment", Proc. IEEE International Conference on Electro Information Technology (EIT 2016), Grand Forks, North Dakota, USA, pp. 382-386, 2016.
[104] Y. Wiseman, "Efficient RFID Devices", Proc. The 42nd Annual Conference of IEEE Industrial Electronics Society (IECON 2016), Firenze (Florence), Italy, pp. 4762-4766, 2016.
[105] Y. Wiseman and I. Grinberg, "The Trolley Problem Version of Autonomous Vehicles", The Open Transportation Journal, Vol. 12, pp. 105-113, 2018.
[106] Y. Wiseman, "Compaction of RFID Devices using Data Compression", IEEE Journal of Radio Frequency Identification, Vol. 1(3), pp. 202-207, 2018.
[107] Y. Wiseman, "High Occupancy Vehicle Lanes are an Expected Failure", International Journal of Control and Automation, Vol. 12(11), pp. 21-32, 2019.
[108] Y. Wiseman, "Israel Complementary International Airport", International Journal of Control and Automation, Vol. 12(7), pp. 1-10, 2019.
[109] Y. Wiseman, "Adjusted JPEG Quantization Tables in Support of GPS Maps", Journal of Mobile Multimedia, Vol. 17(4), pp. 637-656, 2021.
[110] Y. Wiseman, "Blaumilch Canal on Ayalon Highway", Daaton, 2015, Available online at: http://www.daaton.co.il/Article.aspx?id=3290
[111] Y. Wiseman, "Revisiting the Anti-Lock Braking System", Technical Report, 2021.
[112] Y. Wiseman, "Isolated Territories and Infrastructure development: A case for land transportation investment in Madagascar", Interdisciplinary Approaches to the Future of Africa and Policy Development, IGI Global Publishing, Chapter 5, pp. 78-97, 2022.
[113] Y. Wiseman, J. Isaacson, "Safer Operating System for Vehicle Telematics", technical report, 2010.
[114] Y. Wiseman, J. Isaacson, E. Lubovsky and P. Weisberg, "Kernel Stack Overflows Elimination", Advanced Operating Systems and Kernel Applications: Techniques and Technologies, pp. 1-14, IGI Global, 2010.
[115] Y. Wiseman "Airport in Dothan Valley is Ideal", Technical Report, 2020.

[116] M. Itshak and Y. Wiseman, "Enhancing the Efficiency of Memory Management in a Super-Paging Environment by AMSQM", Advanced Operating Systems and Kernel Applications: Techniques and Technologies, pp. 276-293, IGI Global, 2010.
[117] Y. Wiseman, "Conjoint Reliable Vehicle License Plate Identification System", Technical Report, 2020.
[118] M. Reuven and Y. Wiseman, "Alleviating the Thrashing by Adding Medium-Term Scheduler", Advanced Operating Systems and Kernel Applications: Techniques and Technologies, pp. 118-136, IGI Global, 2010.
[119] Y. Wiseman, "Controlling Dynamic Traffic by Road Expansion Can be Accomplished by Double Decker Roads - Case Study", Technical Report, 2022.
[120] Y. Wiseman, "Rail in Islands is an Expected Failure", Technical Report, 2020.
[121] Y. Wiseman, "Road Planners should not Look Just Right and Left But Rather Should Also Look Up", Technical Report, 2022.
[122] Y. Wiseman "Can a Flight Data Recorder be Situated in a Cloud?", Technical Report, 2016.
[123] Y. Wiseman, "JPEG Quantization Tables for GPS Maps", Automatic Control and Computer Sciences, Vol. 55(6), 2021.
[124] Y. Wiseman, "Intelligent Transportation Systems along with the COVID-19 Guidelines will Significantly Change the Transportation Market", Techical Report, 2021.
[125] Y. Wiseman, "Cracked Pipes Alert System for Autonomous Vehicles", Technical Report, 2017.
[126] Y. Wiseman, "EPC Compression", Technical Report, 2016.
[127] Y. Wiseman, "Warning System for Cracked Pipes in Autonomous Vehicles", Advances of Machine Learning in Clean Energy and Transportation Industry, Chapter 9, pp. 261-276, 2021.
[128] Y. Wiseman, "Madagascar had better invest in a single land transportation infrastructure", Interdisciplinary Approaches to the Future of Africa and Policy Development, IGI Global Publishing, 2021.
[129] Y. Wiseman, "Teaching Research of Operating Systems", Technical Report, 2022.
[130] Y. Wiseman, "Improving the Super-Paging Swapping Algorithm", Technical Report, 2022.
[131] Y. Wiseman, "Evaluation of Different Lempel-Ziv Compression Algorithms", Technical Report, 2022.
[132] M. Dreyfuss and Y. Giat, "Optimal spares allocation to an exchangeable-item repair system with tolerable wait", European Journal of Operational Research 261 (2), pp. 584-594, 2017.
[133] Y. Giat, "The effects of output growth on preventive investment policy", American Journal of Operations Research 3 (06), pp. 474-486, 2013.