

Diminution of JPEG Error Effects

Yair Wiseman¹

¹ Computer Science Department, Bar-Ilan University, Ramat-Gan 52900, Israel
wiseman@cs.biu.ac.il

Abstract. JPEG images are self-synchronized in cases of an error; however, after the synchronization the images are not shown as the original ones. Sometimes, the blocks are shifted to right or left and the tinge is damaged as well. This paper suggests a way how to remedy the block shift and the tinge damage.

Keywords: JPEG, Image Error, Data Synchronization

1 Introduction

In the header of each JPEG image the size of the image is encoded as X, Y whereas X and Y are integers defining the width and the height of the image respectively [1]. Unlike faxes that have a special code for end of line [2], JPEG pictures will start the next line right after the previous line comes to an end. The X value stored in the JPEG header file is the only indication for beginning of a new line. There is no codeword stands for end-of-line.

Any type of JPEG files contains several markers [3,4]. Baseline JPEG files (which is the most common JPEG format in use) have a marker indicating the beginning of the image data. This marker is the sequence of two characters 0xff and right after that 0xc0. Other JPEG formats have other markers for that.

After this marker the following information appears in accordance with this order:

1. Three bytes which are not related to this paper.
2. Two bytes containing the image height - Y (in pixels).
3. Two bytes containing the image width - X (in pixels).
4. More information unrelated to this paper.

JPEG images are split into block of 8X8 pixels [5,6]. An error in X or Y can produce a damaged image as described in Figure 1a and Figure 1b. This figure contains an image with an error in the X value. The X value was increased by eight, so each line is longer by one block of 8X8. This increase caused a surplus of one block for each line, but it is accumulated for several blocks. I.e. the second line has been pushed left by one block, the third line has been pushed left by two blocks and as a general rule line N was push by N-1 blocks.

However, most of the damages are in the rest of the file. They automatically synchronized [7,8]; however, they can cause various damages:

1. An erroneous block (or even several blocks) can be mistakenly added and this block will push the line containing this block to the left and as a result all the rest subsequence lines will be pushed to the left as well.
2. A block (or even several blocks) can be incorrectly deleted and this will pull the line containing this block to the right and as a result all the rest subsequence lines will be pulled to the right as well.
3. The average tinge of each block is stored in the beginning of each block. This value is called DC. The DC is not stored as an absolute value, but rather as a relative value [9,10]. I.e. the DC stored in each block is the difference between the current DC value and the DC value of the previous block. A DC value can be incorrect as a result of an error. In such a case, the tinge of this block can be incorrect, but since the values of all the DCs are stored as differences, one wrong DC value will trigger incorrect DC values for all the rest of the blocks in the image.



Fig 1a. Original undamaged JPEG file.



Fig 1b. JPEG file with an incorrect X value.

Figure 2a and Figure 2b demonstrate these problems. Figure 2a contains the original correct image, whereas Figure 2b contains an image with an error. There was an additional block near the jet engines in this image. This additional block causes two problems:

1. The additional incorrect block pushed the block of its line to the left and as a result all the lines in the rest of the image were pushed left as well as can be seen in the bottom of the image.
2. The additional incorrect block had a substantial difference between its DC value and the DC value of the previous block. Therefore, a large number was reduced from the DC value and this reduction made the bottom part of the picture very bright.

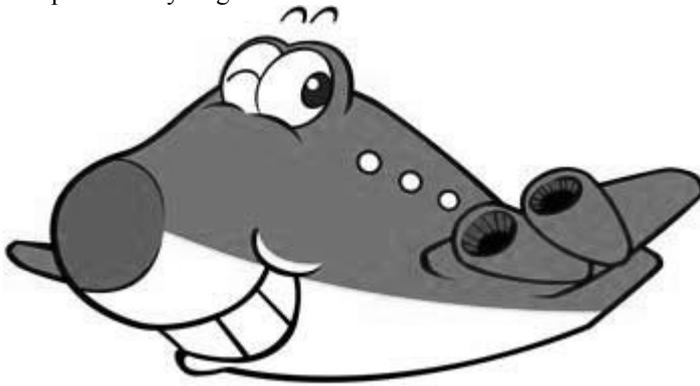


Fig 2a. Original undamaged JPEG file.

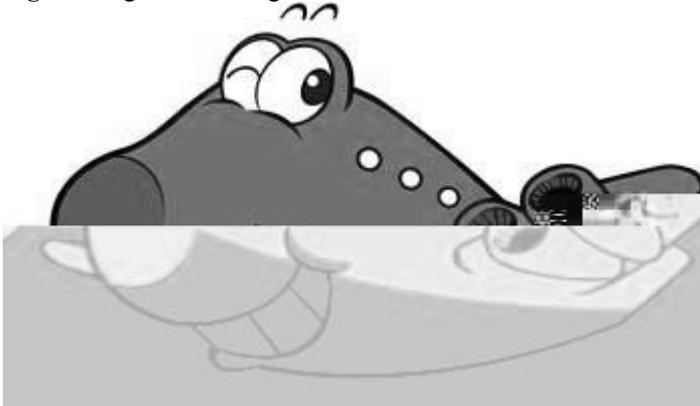


Fig. 2b. Incorrect placing of blocks and incorrect tinge because of an error.

2 How to Diminish JPEG Error Effects

Some compression formats like TIFF have an "end of line" codeword [11]. Such a codeword can help JPEG to know where the real end of line is and the lines will not

be moved as described in Figure 1a, Figure 1b, Figure 2a and Figure 2b. Indeed, an end-of-line codeword have two major disadvantages:

1. It takes some space to store end-of-line codewords for all the lines and makes the compression less efficient; however, some of the efficiency loss can be recompensed by eliminate the need for use of the X and Y components in the header.
2. JPEG usually employs Huffman as its entropy encoder although some other compression techniques are available [12]. A new codeword will make other Huffman codewords of JPEG longer. Therefore, this additional end-of-line codeword will also make the compression less efficient.

However, the less efficiency of the compression will be compensated by better images in case of errors, particularly when the communication lines are noisy and there is a realistic possibility for an error in the transfer. Also, critical safety equipment using JPEG files like [13,14,15] may necessitate more accurate and proper data. Furthermore, some of the data in JPEG files is redundant anyway and is sometimes replaced by steganography systems in order to hide data [16,17,18], so if the data does not attempt to be the best compressed, some more bits for an end-of-line codeword will not make a significant harm.

As was mention above, the DCs of JPEG are stored as differences. This is again better for compression efficiency. The differences are usually quite small and it is more efficient to store small numbers rather than longer numbers; however, storing the DC values themselves will help JPEG to show much more accurate images. If the DC values themselves were stored in the JPEG file, the bright bottom part of the image in Figure 2b would have been shown in the original tinge as it is shown in Figure 2a with no tinge change.

Yet again, storing the DC values themselves is trading off with compression efficiency; however, the less efficiency of the compression will be compensated by better images in case of errors.

3 Conclusions

Errors in JPEG images are almost always synchronized; however, the images after the synchronization are not so clear and sometimes for essential apparatus using JPEG [19,20], the data is required to be completely correct. This paper presents techniques how to resolve the problems after the synchronization and so as to make the images clearer and more comprehensible.

References

1. W. B. Pennebaker and J. L. Mitchell. "JPEG: Still image data compression standard", Springer Science & Business Media, 1993.

2. R. Buckley, D. Venable and L. McIntyre, "New developments in color facsimile and internet fax", In proceedings of Color and Imaging Conference, Vol. 1997(1), pp. 296-300, Society for Imaging Science and Technology, 1997.
3. Y. Wiseman, "The still image lossy compression standard – JPEG", Encyclopedia of Information and Science Technology, Third Edition, Vol. 1, Chapter 28, 2014.
4. G. K. Wallace, "The JPEG still picture compression standard", Communications of the ACM, Vol. 34(4), pp. 30-44, 1991.
5. Y. Wiseman and E. Fredj, "Contour Extraction of Compressed JPEG Images", Journal of Graphic Tools, Vol. 6(3), pp. 37-43, 2001.
6. E. Fredj and Y. Wiseman, "An O(n) Algorithm for Edge Detection in Photos Compressed by JPEG Format", Proceedings of International Conference on Signal and Image Processing SIP-2001, Honolulu, Hawaii, pp. 304-308, 2001.
7. S. T. Klein and Y. Wiseman, "Parallel Huffman Decoding with Applications to JPEG Files", The Computer Journal, Oxford University Press, Swindon, UK, Vol. 46(5), pp. 487-497, 2003.
8. S. T. Klein and Y. Wiseman, "Parallel Huffman Decoding", Proceedings of Data Compression Conference, DCC-2000, Snowbird, Utah, USA, pp. 383-392, 2000.
9. Y. Wiseman, "Enhancement of JPEG Compression for GPS Images", International Journal of Multimedia and Ubiquitous Engineering, Vol. 10(7), pp. 255-264, 2015.
10. Y. Wiseman, "Improved JPEG based GPS picture compression", Advanced Science and Technology Letters, Vol. 85, pp. 59-63, 2015.
11. T. Davenport and M. Vellon, "Tag Image File Format (TIFF)-Rev. 4.0", Aldus Corp. & Microsoft Corp, 1987.
12. Y. Wiseman, "Burrows-Wheeler Based JPEG", Data Science Journal, Vol. 6, pp. 19-27, 2007.
13. Y. Wiseman, "Take a Picture of Your Tire!", Proceedings of IEEE Conference on Vehicular Electronics and Safety (IEEE ICVES-2010), Qingdao, ShanDong, China, pp. 151-156, 2010.
14. Y. Wiseman, "The Effectiveness of JPEG Images Produced By a Standard Digital Camera to Detect Damaged Tyres", World Review of Intermodal Transportation Research, Vol. 4(1), pp. 23-36, 2013.
15. Y. Wiseman, "Camera That Takes Pictures of Aircraft and Ground Vehicle Tires Can Save Lives", Journal of Electronic Imaging, Vol. 22(4), paper no. 041104, 2013.
16. 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.
17. Y. Wiseman, "Steganography Based Seaport Security Communication System", Advanced Science and Technology Letters, Vol. 46, pp. 302-306, 2014.
18. J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes", In Information Hiding, Springer Berlin Heidelberg, pp. 67-81, 2005.
19. Y. Wiseman, "Device for Detection of Fuselage Defective Parts", Information Journal, Tokyo, Japan, Vol. 17(9(A)), pp. 4189-4194, 2014.
20. Y. Wiseman, "Fuselage Damage Locator System", Advanced Science and Technology Letters, Vol. 37, pp. 1-4, 2013.