Device for Detection of Fuselage Defective Parts

Yair Wiseman[1]
[1]Bar-Ilan University, Ramat-Gan, Israel, wiseman@cs.biu.ac.il

Abstract

Substandard fuselage can trigger a deadly disaster that may claim many lives. This paper proposes employing of a standard digital camera that can produce JPEG images in an attempt to pinpoint a scratch in a fuselage. The digital camera takes pictures of the fuselage. The signal processing scheme of JPEG takes for granted that the taken picture is pretty smooth. Accordingly, when there is an abrupt difference inside an individual block of a picture, the value of many of its frequency coefficients will be above what is usual; therefore it will be compressed into many more bits. For that reason, if the image is more than usually large, an embedded computer system can spot the scratch. In such cases, sometimes an action can be taken in an attempt to minimize the risky outcomes of the impairment.

Keywords: Fuselage, Discrete Cosine Transform, JPEG, Signal Processing

1. Introduction

At April 1, 2011, a Boeing 737-3H4 flew from Phoenix, Arizona to Sacramento, California. Near Yuma, Arizona, a mysterious sound was recorded on the cockpit area microphone. Few seconds later, the captain noticed that the airplane had rapidly lost cabin pressurization and called for oxygen masks on. Steady sounds with increased wind noise were heard on the cockpit voice recording. The captain announced an emergency with air traffic control. The cabin crew began to search the source of the noises and found a 2-foot hole in the fuselage. The aircraft was diverted to Yuma International Airport. The airplane landed there without further incident. The damaged part of the fuselage can be seen at Figure 1.

The cabin crew in this case eventually indeed succeeded to find the damage part; however it took some time to find the damaged part and in such cases time is very crucial.

The monitoring of the fuselage condition is extremely vital [1]. We suggest using an ordinary digital camera to locate the damaged part [2]. Nearly all digital cameras can produce JPEG pictures. JPEG is a very common method for image compression and it is also extensively used by electronic devices like scanners and digital cameras [3] as well as vehicle equipment like GPS [4].

JPEG images have many advantages like the ability of being decoded in parallel [5], the straightforwardness of adaptation for new compression methods [6] and the capability of flexible implementation for hardware from different vendors [7].

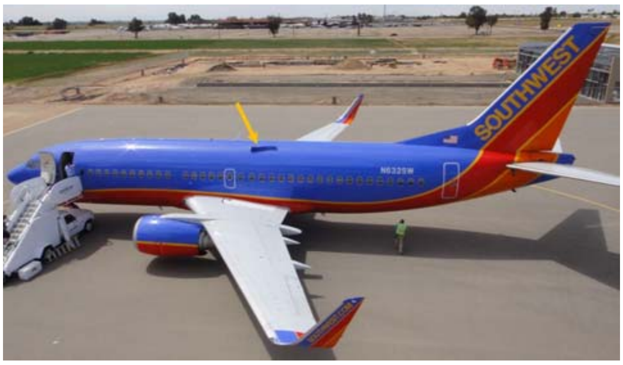

Figure. 1. Dangerous fuselage damage

Images are often stored in a compressed standard. A naive approach for image processing on compressed images would be to decompress the image and then running the image processing algorithm on the original image data. Instead, for some image operations, we can act on the compressed data directly. This gives us two benefits: first, we can use the standard digital cameras without a need to adjust the digital camera; second, the signal processing used by JPEG can let us use the frequency information embedded in the compressed data.

The rest of the paper is organized as follow: Section 2 describes the algorithm of baseline JPEG. Section 3 explains how JPEG images can be used for detecting damaged fuselage. Section 4 describes the experiments we have performed, while section 5 concludes the paper.

2. 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 YUV components [10] or the split into RGB 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].

JPEG allows samples of 8 bits or 12 bits. In the original JPEG all values contained by the same source image must have the same precision. The values are shifted from unsigned integers with range [0,2p-1] to signed integers with range [-2p-1,2p-1-1], by reducing 2p-1 from the original values, where p can be either 8 or 12. These biased values are then sent to the next step.

The second step groups the pixels into blocks of 8X8 pixels. The order of the blocks is line by line and each line is read from left to right. After the group into blocks, JPEG 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 much more 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 and the remaining 63 are named Alternating Current (AC) coefficients. These coefficients reflect the frequency information of their row and column. These coefficients indicate the amplitude of a specific frequency component of the input array. The frequency content of the sample set at each frequency is calculated by taking a weighted sum of the entire set.

Table 1 contains the values of weight for one row in a 8x8 matrix. The rows are the Index of result whereas the columns are the Index of value.

	0	1	2	3	4	5	6	7
0	+0.707	+0.707	+0.707	+0.707	+0.707	+0.707	+0.707	+0.707
1	+0.981	+0.831	+0.556	+0.195	-0.195	-0.556	-0.831	-0.981
2	+0.924	+0.383	-0.383	-0.924	-0.924	-0.383	+0.383	+0.924
3	+0.831	-0.195	-0.981	-0.556	+0.556	+0.981	+0.195	-0.831
4	+0.707	-0.707	-0.707	+0.707	+0.707	-0.707	-0.707	+0.707
5	+0.556	-0.981	+0.195	+0.831	-0.831	-0.195	+0.981	-0.556
6	+0.383	-0.924	+0.924	-0.383	-0.383	+0.924	-0.924	+0.383
7	+0.195	-0.556	+0.831	-0.981	+0.981	-0.831	+0.556	-0.195

Table 1: Values of weight for one row in a 8x8 matrix

In other words, these AC values of one row are the results of this formula:

$$F(u) = \frac{1}{2}\sum_{x=0}^{7} f(x)C(u)\cos[(2x+1)u\pi/16]$$

Where:

- x is the index of value.
- f(x) is the value itself.
- u is the index of result.
- $C(u) = 1/\sqrt{2}$ if u=0; *C(u)*=1 otherwise.
- F(u) is the result itself.

If f(x) (the intensity of each pixel) is exactly the same in the entire row, each F(u) which holds u>0, will be zero. F(0) will be the sum of the row's values divided by $2\sqrt{2}$. This explains why JPEG is better pictures which do not have sharp changes. Pictures which have sharp changes like cartoons will have larger values and hence will be compressed not as good as the pictures without the sharp changes.

As was mention above, the blocks are of 8X8, so one-dimensional DCT is enough for only one row. The DCT for an entire block is done by applying one-dimensional DCT separately for each row of eight pixels. The result will be eight rows of frequency coefficients. Then, these 64 coefficients are taken as eight columns. The first column will contain all DC coefficients, the second column will contain the first AC coefficient from each row, and so on. At that time, JPEG apply one-dimensional DCT for each of these columns.

In reality the one-dimensional DCT is not used and the entire calculation is done by one formula that reflects this calculation:

$$F(u,v) = \frac{1}{4}\sum_{y=0}^{7}\sum_{x=0}^{7} f(x,y)C(u)\cos[\frac{(2x+1)u\pi}{16}]C(v)\cos[\frac{(2y+1)v\pi}{16}]$$

Where:

- x,y are the indices of value.
- f(x,y) is the value itself.
- u,v are the indices of result.
- $C(u) = 1/\sqrt{2}$ if u=0; *C(u)*=1 otherwise.
- $C(v) = 1/\sqrt{2}$ if v=0; *C(v)*=1 otherwise.
- F(u,v) is the result itself.

It should be noticed that index 0,0 contains the DC of the DCs. This value is called the DC of the 8x8 block.

The next step is the quantization. The 63 AC coefficients are ordered into a zig-zag sequence. Firstly JPEG puts the coefficient that its row index plus column index is 0 i.e. the DC. Then JPEG puts the coefficients that their row index plus column index is 1. Next, JPEG puts the coefficients that their row index plus column index is 2 and JPEG continues until putting the last coefficient that its row index plus column index is 14. This zig-zag algorithm arranges the coefficients into one dimensional array .In each block, each of the 64 coefficients 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.

Unlike text compression method that usually use Lempel-Ziv methods [14], the last step of JPEG encodes the reduced coefficients using either Huffman or Arithmetic coding. There are also other compression algorithms that fit JPEG, but are not officially supported [15].

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 any given image, one Huffman tree for the DC coefficients' length and the other Huffman tree for the AC coefficients' length. The tree for the DC values encodes the lengths in bits (1 to 11) of the binary representations of the values in the DC fields.

The second tree encodes information about the sequence of AC coefficients. As many of them are zero, and most of the non-zero values are often concentrated in the upper left part of the 8 × 8 block, the AC coefficients are scanned in the zig-zag order specified by the quantization step i.e. processing elements on a diagonal close to the upper left corner before those on such diagonals further away from that corner, that is the order is given by (0, 1), (1, 0), (2, 0), (1, 1), (0, 2), (0, 3), (1, 2), etc.

The second Huffman tree encodes pairs of the form (n, l), where n (limited to 0 to 15) is the number of elements that are zero, preceding a non-zero element in the given order and l is the length in bits (1 to 10) of the binary representation of the non-zero quantized AC value.

The second tree also includes codewords for EOB, which is used when no non-zero elements are left in the scanning order and for a sequence of 16 consecutive zeros in the AC sequence (ZRL), necessary to encode zero-runs that are longer than 15. The Huffman trees used in baseline JPEG are static and can be found in [16].

Each 8X8 block is then encoded by an alternating sequence of Huffman codewords and binary integers (except that the codeword for ZRL is not followed by any integer), the first codeword belonging to the first tree and relating to the DC value, the other codewords encoding the (n, l) pairs for the AC values, with the last codeword in each block representing EOB.

Table 2. Example of block compression

20	**1**	0	0	0	0	0	0
0	**3**	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Table 2 brings an example of a JPEG 8X8 block after DCT and after quantization., with the DC value is in the upper left corner. The DC value is more often than not very similar to the previous DC value, so it is computed as a difference between the current value and the previous DC value. Suppose the previous DC value in this example was 15, the result of the compression will be as shown in Figure 2.

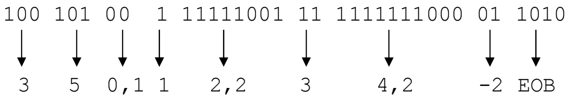

Figure 2. Ruselt of JPEG compression

The first value (110) is the Huffamn code for the DC length in bits (3). Then, the difference between the current DC value and the previous DC value is explicitly written (5). Next, the AC values are written as sequences of several zeros followed by one non-zero value. At the end a special code EOB (1010) is written.

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 3.

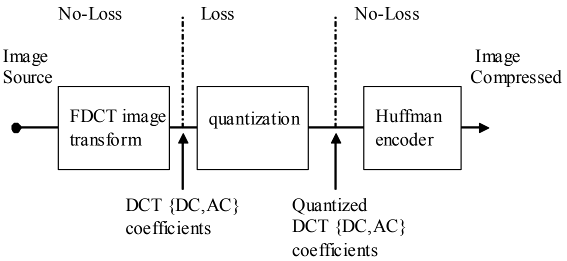

Figure 3: JPEG Model for a lossy image compression

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) [17] 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.; however it is still a very short time as can be indicated in every digital camera.

Another disadvantage is the compression method that does not take into account the temporal correlation of the coefficients.

To conclude, 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: storing full color information and reducing the size of image files, so as to avoid high traffic on the network and avoiding memory pressure [18]. JPEG is an eminent format and is described in many places e.g. [19,20].

3. Detection of Fuselage Defective Parts

As was mentioned in the previous section, the JPEG standard is based on the DCT paradigm [21,22]. The DCT changes the picture into frequency space. The frequency coefficients, which are very low magnitude, are rounded to zero. When most of the coefficients in a block are zero or very low magnitude: The compression algorithm will give a very short bits sequence for such a block. Zero sequences are treated very efficiently by JPEG compression and the results will be only few bytes.

When there is a drastic change in a block of 8X8, the value of many frequency coefficients will be high. Such a sequence will be compressed into many more bits. JPEG's standard stipulates that the block's size will be 8X8 pixels, but the algorithm will be obviously good for other small NXN pixels size too.

When there is a good algorithm that is very common in the market and does a good job, it seems to be the commonsensical choice. There might be some other good algorithms, but using those algorithms means you have to push a new standard which can be a very hard task.

When looking for the contour of an object, the goal will be to locate the object's border. The idea is to break the compressed file into its original blocks, then look in the compressed file for long bit sequences. The blocks which are compressed into long bit sequences, are presumed to be the object's border. In our implementation we took a simpler approach. We take pictures of the fuselage. We actually take a close picture of each part of the fuselage. If the entire size of a picture is above a certain threshold, we will consider this part of the fuselage as a damaged part.

If we have no idea what the threshold value should be, we can examine the probability density function (PDF) [23] of the block representation to select a suitable value. In the uncomplicated case the PDF should be mono-modal and we set the value in the inflection point.

Figure 2 demonstrates how JPEG is used for contour extraction. The original image was compressed in grayscale baseline JPEG format with 75% quality. The figure shows the original image, which is a high-resolution picture of 1000X1000 pixels. Table 1 shows the JPEG format in the upper left corner of the black square. The size of the black square is 200X200 pixels and the square is not aligned relative to JPEG's 8X8 blocks.

The JPEG file reports the difference of magnitude between the DC's coefficients of a previous block relative to the current block. In the case of a white or black area there are no changes in the coefficients' magnitudes. This type of block is encoded as six bits by the JPEG standard:

0,0,1,0,1,0.

The "00" reflects that there are no differences between the values of the previous and the current block's DC coefficients, and "1010" symbolizes the end of the block. If there is a difference between the intensity of the DC coefficients of the previous and the current blocks, the size of the encoding block will be slightly larger. For example, a block which encodes a sharp change from white to black is represented by a wide range of frequency coefficients. It is easy to select a threshold that delimits the edges of the shape from the rest of the image [24,25,26].

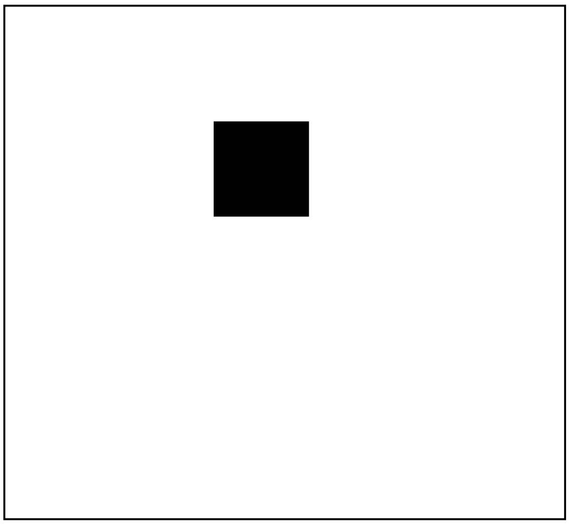

Figure. 4. Sample image and how JPEG can be used for contouring

Table 1. JPEG format of the upper left corner of the black square.

-12	18	-8	-2	3	-1	-1	1
22	-25	8	2	-3	0	1	-1
-18	20	-7	-2	2	0	-1	1
16	14	4	1	-1	0	1	1
-11	10	-2	-1	0	-1	-1	0
6	-5	1	-3	0	0	0	0
-2	2	-1	0	0	0	0	0
1	-1	0	0	0	0	0	0

Figure 4 and Table 1 show a sample of the block, which contains the upper left corner of the black square. In order to compress these values in JPEG standard, 243 bits are needed. The difference between 6 to 243 is obviously significant. By using three parameters the length of the block, its magnitude and the number of consecutive blocks the threshold can extract the contour with a range of scalar values [27,28]. The extra parameters allow more control over the resulting mechanism.

The algorithm is very simple and can be described as follows:

```
• Take a picture of the fuselage part and create an image - I
    o Set L= size_of_image (I)
    o If (L > T)
        ▪ then the fuselage is deemed to be damaged and should be checked.
```

Where T is a threshold and its value will be discussed below.

4. Experiments

We examined our technique on some damaged fuselages and tried to check whether we succeed to locate the damaged part. Clearly, obvious cases like in Figure 4 do not exist in real fuselages, but we still succeed to find most of the damaged parts by taking many picture of it in all of its parts.

We used an Olympus FE-170 digital camera with resolution of 2112X2816. The Images with no damage in the fuselage produced images in size of less than 1.3MB; whereas images in size of more than 1.5MB usually had damage in the fuselage.

Many of the fuselage damages are clear like the damage in Figure 5. It appears that many wildlife strike aircrafts [29] and in such cases the damage is usually quite large.

Since the establishment of the Federal Aviation Administration's (FAA) National Wildlife Strike Database at 1990, 99,411 reported wildlife strikes to airplanes have been reported [30]. Also strikes of bats are reported as a noteworthy difficulty for the air traffic [31]. Geese are also can cause crash of aircrafts as was happened to US Airways Flight 1549, which crash landed in the Hudson River in New York City on 15 January 2009 after a collision with a flock of geese [32].

The result of these strikes causes more than 200 human lives lost worldwide as well as financial losses (direct and indirect) of at least $1.2 billion annually to civil aviation worldwide. Particularly, more than $625 million of financial losses annually just in the United States [33].

This is more than the loss for the aviation industry caused by various terror groups [34].

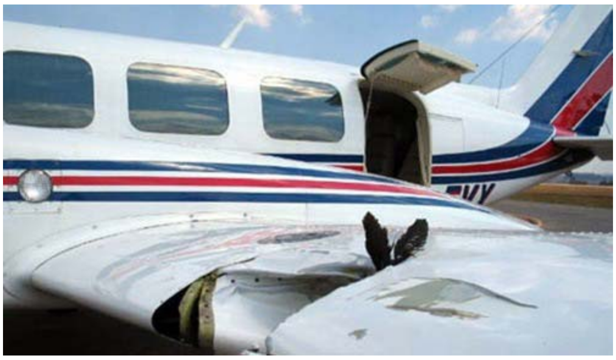

Figure 5. Large and clear damage

However, sometimes there are smaller damages. One major reason for smaller damages is hail [35]. At July 14, 2011, More than 100 flights have been called off at Denver International Airport after hail damaged about 40 aircrafts and stranded about 1,000 passengers overnight. The damages were usually not large as can been seen in Figure 6.

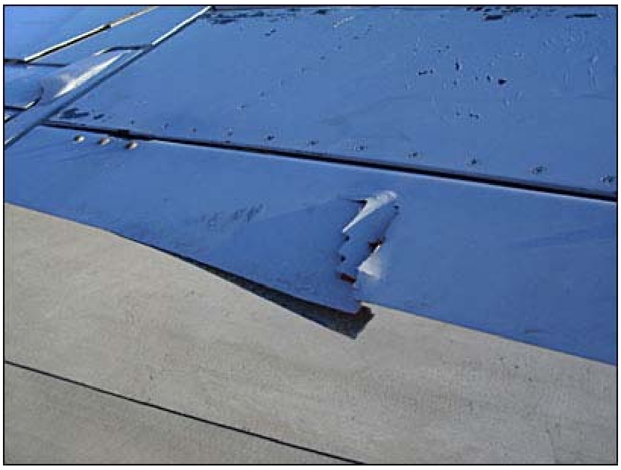

Figure 6. Hail Damage in Denver International Airport

There also some other small objects that sometime soar over airports and can cause tiny damages like the damage in Figure 7.

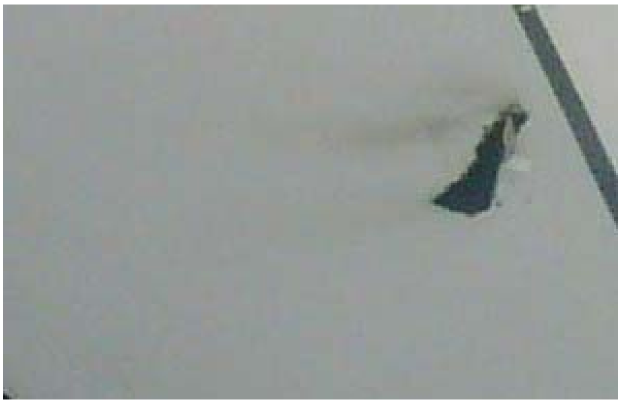

Figure 7. Small damage

Actually, there are seasons that there more such accidents [36]. Usually the months of Aug-Oct have the highest priority of having such damages as can been seen in Figure 8.

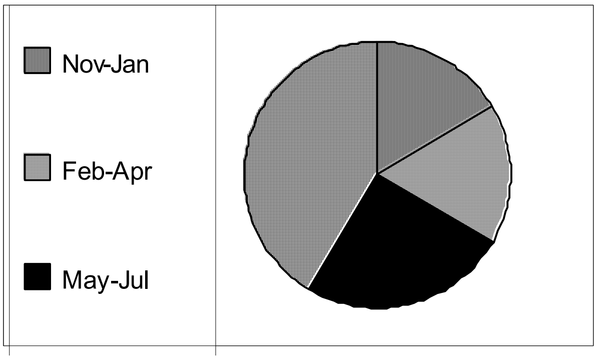

Figure 8. Strike's distribution during a year

As a matter of fact, the use of our system for large and clear damages is less important, although sometimes a cabin crew member under stress can need a help even such cases; however when a small and indistinguishable damage can make lower the pressure in the cabin and locating it without delay can be an imperative.

There are also cases where our system has produced false alarms. Figures 9 and 10 show examples for such a false alarm. The dirt on the fuselage in Figures 9 and 10 can be easily noticed. This dirt made the system assuming that the fuselage is damaged; whereas this fuselage was actually undamaged.

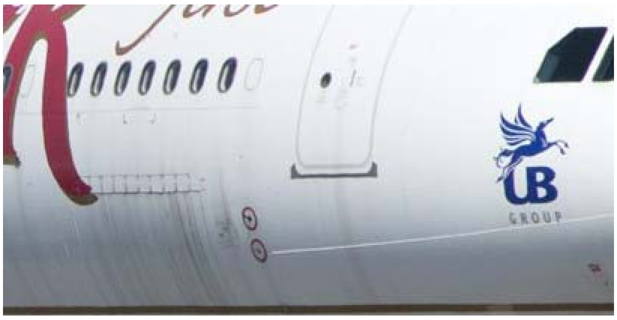

Figure 9. Dirt on the Fuselage

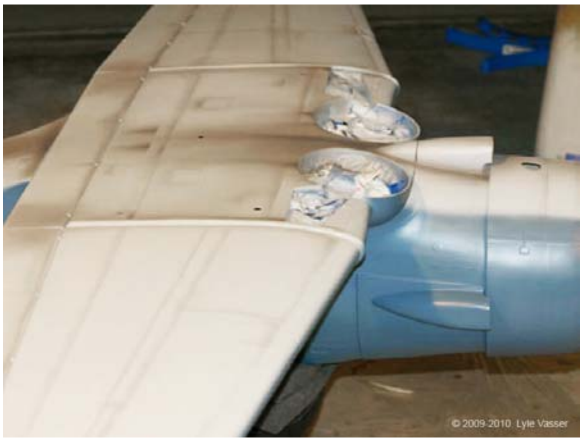

Figure 10. More Dirt on the Fuselage

The quality factor of JPEG is used to produce two quantization tables - one for the luminance (brightness) information and the other for the chrominance (color) information. Applications that support IJG (Independent JPEG Group) produce quantization tables according to the IJG specification; however, many other applications employ different quantization tables.

At this paper we have used IJG quantization tables. Actually, in our system, we obviously preferred the highest quality factor (100%), so as to get the best distinguishableness, which means in IJG that the entire quantization table is filled with 1s i.e. the frequency coefficients are not divided.

5. Conclusions and Future Work

Out of order fuselage is very risky and can cause too many loss of life. Automatic inspection of aircraft fuselage is critical and vital as was noted in [37].

Damage detection of vulnerable structures has been explored by many researchers in different engineering disciplines. Various detection methods have been developed and successfully applied to different types of structures [38,39,40,41]. In addition, various damage scenarios and circumstances have been researched [42,43].

This method can be also effective for data collection for the flight data recorder. Using this device the flight data recorder will be able to know about the size and the location of the damage during the flight time [44,45].

The suggested system in this paper can be mounted in an aircraft and it may save lives. In this paper, we explained how a system using simple equipment – digital camera can detect damages in a fuselage. In the future we would like to adapt our system to a ground camera. Conventional digital cameras cannot take a picture of so fast moving object and so far objects, so we should find a practical way how the pictures can be taken.

We also consider developing more sophisticated algorithm that can not only identify damaged fuselage, but also identify various kinds of damages. Looking at existing machine-vision [46,47,48] and machine-learning algorithms [49,50] may help in significantly extending the suggested method.

6. References

[1] Dalton, R. P., Cawley, P. and Lowe, M. J. S., "The potential of guided waves for monitoring large areas of metallic aircraft fuselage structure", Journal of Nondestructive Evaluation, Vol. 20, No. 1, pp. 29-46, (2001).

[2] Wiseman Y., "Fuselage Damage Locator System", Proceedinds 2nd International Workshop on Signal Processing, (2013), December 11-13, Jeju Island, Korea.

[3] Khanna, N., Chiu, G. T. C., Allebach, J. P. and Delp, E.J., "Forensic techniques for classifying scanner, computer generated and digital camera images", Proceedings of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008), pp. 1653-1656, (2008).

[4] P. Hongyan, H. Hong, J. Hengtian, "Drive design for ship GPS navigation equipment based on Linux operating system", International Conference on Educational and Network Technology (ICENT), pp. 384-388, (2010), June, Qinhuangdao, China.

[5] Klein S. T. and Wiseman Y., "Parallel Huffman Decoding with Applications to JPEG Files", The Computer Journal, Oxford University Press, Swindon, UK, Vol. 46 no. 5, pp. 487-497, (2003).

[6] Klein S. T. and Wiseman Y., "Parallel Lempel Ziv Coding", Journal of Discrete Applied Mathematics, Vol. 146, No. 2, pp. 180-191, (2005).

[7] Wiseman Y., "A Pipeline Chip for Quasi Arithmetic Coding", IEICE Journal - Transactions Fundamentals, Tokyo, Japan, Vol. E84-A No.4, pp. 1034-1041, (2001).

[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 System 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] Wiseman Y., "The Relative Efficiency of LZW and LZSS", Data Science Journal, Vol. 6, pp. 1-6, 2007.

[15] Wiseman Y., "Burrows-Wheeler Based JPEG", Data Science Journal, Vol. 6, pp. 19-27, (2007).

[16] ISO/IEC 10918-1 Information technology—digital compression and coding of continuous–tone still images requirements and guidelines. International Standard ISO/IEC, Geneva, Switzerland, (1993).

[17] McMillan Jr, Leonard and Lee A. Westover, "Method and apparatus for fast implementation of inverse discrete cosine transform in a digital image processing system using low cost accumulators." U.S. Patent No. 5, 301, 136. 5, Apr. (1994).

[18] Reuven M., and Wiseman Y., Medium-term scheduler as a solution for the thrashing effect. The Computer Journal, 49, No. 3,, 297-309, (2006).

[19] Wallace G. K., "The JPEG Still Picture Compression Standard", Communication of the ACM Vol. 34, pp. 3-44, (1991).

[20] Y. Wiseman, "The still image lossy compression standard – JPEG", Encyclopedia of Information and Science Technology, Third Edition, (2014).

[21] Britanak, V., Yip P. C. and Rao K. R., "Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations", Elsevier, (2010).

[22] Yongyi Y., Galatsanos N. P. and Katsaggelos A. K., "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images", IEEE Transactions on Circuits and Systems for Video Technology, Vol. (3) No. 6, pp. 421-432, (1993).

[23] Krylov, Vladimir, Gabriele Moser, Sebastiano B. Serpico, and Josiane Zerubia, "On the method of logarithmic cumulants for parametric probability density function estimation" IEEE Transactions on Image Processing, Vol. 22, no. 10, pp. 3791-3806, October, (2013).

[24] Y. Wiseman, "Take a Picture of Your Tire!", Proceedings IEEE Conference on Vehicular Electronics and Safety (IEEE ICVES-2010) pp. 151-156, (2010), July 15-17, Qingdao, ShanDong, China.

[25] 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, No. 1, pp. 23-36, (2013).

[26] Y. Wiseman, "Camera That Takes Pictures of Aircraft and Ground Vehicle Tires Can Save Lives", Journal of Electronic Imaging, Vol. 22, No. 4, 041104, (2013).

[27] Wiseman Y. and Fredj E., "Contour Extraction of Compressed JPEG Images", ACM - Journal of Graphic Tools, Vol. 6, No. 3, pp. 37-43, (2001).

[28] Fredj E. and Wiseman Y., "An O(n) Algorithm for Edge Detection in Photos Compressed by JPEG Format", Proceedings of IASTED International Conference on Signal and Image Processing SIP-2001, pp. 304-308, (2001), August 13-16, Honolulu, Hawaii.

[29] Blackwell B. F., DeVault T. L., Fernández-Juricic E. and Dolbeer R. A., "Wildlife collisions with aircraft: a missing component of land-use planning for airports", Landscape and Urban Planning, Vol. 93, No. 1, pp. 1-9, (2009).

[30] DeVault, T.L., M.J. Begier, J.L. Belant, B.F. Blackwell, R.A. Dolbeer, J.A. Martin, T.W. Seamans, Â and B.E. Washburn., "Rethinking airport land-cover paradigms: agriculture, grass, and wildlife hazards", Journal of Human-Wildlife Interactions Vol. 7, No. 1, pp. 10-15, (2013).

[31] Biondi, K.M., J.L. Belant, T.L. DeVault, J.A. Martin, and G. Wang, "Bat incidents with U.S. civil aircraft", Acta Chiropterologica, Museum and Institute of Zoology, Vol. 15, No. 1, pp. 185-192, (2013).

[32] Marra, Peter P., Carla J. Dove, Richard Dolbeer, Nor Faridah Dahlan, Marcy Heacker, James F. Whatton, Nora E. Diggs, Christine France, and Gregory A. Henkes. "Migratory Canada geese cause crash of US Airways Flight 1549." Frontiers in Ecology and the Environment, Vol. 7, no. 6 pp. 297-301, (2009).

[33] Martin J. A., Belant J. L., Devault, T. L., Blackwell B. F., Burger Jr. L. W., Riffell S. K. and Wang G., "Wildlife risk to aviation: a multi-scale issue requires a multi-scale solution", Human-Wildlife Interactions, Vol. 5(2), pp. 198-203, (2011).

[34] Wiseman Y. and Giat Y., "Multi-modal passenger security in Israel", Multimodal Security in Passenger and Freight Transportation: Frameworks and Policy Applications, Luca Zamparini, Dawna Rhoades, Genserik Reniers and Joseph Szyliowicz Editors, Edward Elgar Publishing Limited, (2014).

[35] Souter R. K., and Emerson J. B., "Summary of available hail literature and the effect of hail on aircraft in flight", (No. NACA-TN-2734). National Aeronautics and Space Administration Washington DC, (1952).

[36] Dolbeer R. A., "Height distribution of birds recorded by collisions with civil aircraft", Journal of Wildlife

Management, The Wildlife Society, Vol. 70, No. 5, pp. 1345-1350, (2006).

[37] Alan J. Stolzer, Carl D. Halford and John J. Goglia, "Safety Management Systems in Aviation", Chapter 3 – Principles of Quality Management, MPG Books Group, UK, (2010).

[38] Swift T. "The application of fracture mechanics in the development of the DC-10 fuselage(analysis of the degree of damage tolerance of fuselage pressure shell)." AGARD Fracture Mechanics of Aircraft Structures pp. 226-287(SEE N 74-23413 14-32), (1974).

[39] Wang J. T., Poe Jr C. C., Ambur D. R. and Sleight, D. W., "Residual strength prediction of damaged composite fuselage panel with R-curve method",.Composites science and technology journal, Vol. 66 no. 14, pp. 2557-2565, (2006).

[40] Rouse, M. and Ambur D. R., "Evaluation of Damaged and Undamaged Fuselage Panel Responses Subjected to Internal Pressure and Axial Load." Sixth NASA/DoD/ARPA ACT Conference, (1995), Anaheim, CA.

[41] Singh Ripudaman, Jai H. Park and Satya N. Atluri. "Residual life and strength estimates of aircraft structural components with MSD/MED." In NASA. Langley Research Center, FAA/NASA International Symposium on Advanced Structural Integrity Methods for Airframe Durability and Damage Tolerance, Part 2, pp. 771-783(SEE N 95-19468 05-03). Vol. 1994. (1994).

[42] Jiang X. G., Fu X. R., and Ju J. S., "Global-local Hierarchical Analysis Techniques for Damaged Aircraft Fuselage Considering Bulging Deformation of Crack." Key Engineering Materials Journal, Vol. 324, pp. 871-874, (2006).

[43] Ju J. S., Jiang X. G., and Fu X. R., "Fracture analysis for damaged aircraft fuselage subjected to blast", Key Engineering Materials Journal, Vol. 348, pp. 705-708, (2007).

[44] Haverdings, H. and Chan P. W., "Quick access recorder data analysis software for windshear and turbulence studies" Journal of Aircraft, American Institute of Aeronautics and Astronautics, Vol. 47, no. 4, pp. 1443-1447, (2010).

[45] Wiseman Y. and Barkai A., "Smaller Flight Data Recorders", Journal of Aviation Technology and Engineering, Vol. 2, no. 2, pp. 45-55, (2013).

[46] E. Dickmanns, "The Development of Machine Vision for Road, Vehicles in the Last Decade," Proceedings IEEE Intelligent Vehicle Symposium, (2002).

[47] J. Wang, G. Bebis, and R. Miller, "Overtaking Vehicle Detection Using Dynamic and Quasi-Static Background Modeling," Proceedings IEEE Workshop Machine Vision for Intelligent Vehicles, (2005).

[48] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection: A Review," IEEE Transaction Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 694-711, May (2006).

[49] W. von Seelen, C. Curio, J. Gayko, U. Handmann, and T. Kalinke, "Scene Analysis and Organization of Behavior in Driver Assistance Systems," Proceedings IEEE International Conference on Image Processing, pp. 524-527, (2000), September 10-13, Vancouver, BC, Canada.

[50] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection Using Evolutionary Gabor Filter Optimization" IEEE Transactions Intelligent Transportation Systems, vol. 6, no. 2, pp. 125-137, (2005).