IDENTIFICATION OF BITMAP COMPRESSION HISTORY: JPEG DETECTION AND QUANTIZER ESTIMATION

Zhigang Fan and Ricardo de Queiroz

Xerox Corporation, 800 Phillips Road, Webster, NY 14580

zfan@crt.xerox.com queiroz@wrc.xerox.com

ABSTRACT

Sometimes image processing units inherit images in raster bitmap format only, so that processing is to be carried without knowledge of past operations that may compromise image quality (e.g. compression). To carry further processing, it is useful to not only know whether the image has been previously JPEG compressed, but to learn what quantization table was used. This is the case, for example, if one wants to remove JPEG artifacts or for JPEG re-compression. In this paper, a fast and efficient method is provided to determine whether an image has been previously JPEG compressed. After detecting a compression signature, we estimate compression parameters. Specifically, we developed a method for the maximum likelihood estimation of JPEG quantization steps. The quantizer estimation method is very robust so that only sporadically an estimated quantizer step size is off, and when so, it is by one value.

1. INTRODUCTION

In several situations, images are processed as bitmaps without any knowledge of prior processing. It typically happens when the image processing unit inherits the image data as a bitmap without any side information. For example, imaging related drivers in operational systems may receive a bitmap image with instructions for rendering it at a particular size and position, but without further information. The image may have been processed and perhaps compressed. It might even contain severe compression artifacts. All these facts are unknown to the image processing unit. If one wants to ensure that image is rendered in good conditions, it is desirable to understand the artifacts the image might have, i.e. it is desirable to know a bit of the image's "history". Particularly, we are interested in detecting whether the image has ever been compressed using the JPEG standard [1]. Furthermore, we would like to know what quantizer tables were used. This is quite important for a few applications. For example, in removing artifacts, most of the artifact removal algorithms

[2-9] require the knowledge of the quantization table. In another application, if one knows the quantizer table, it is possible to avoid further distortion when recompressing the image. The situation is illustrated in Fig. 1.

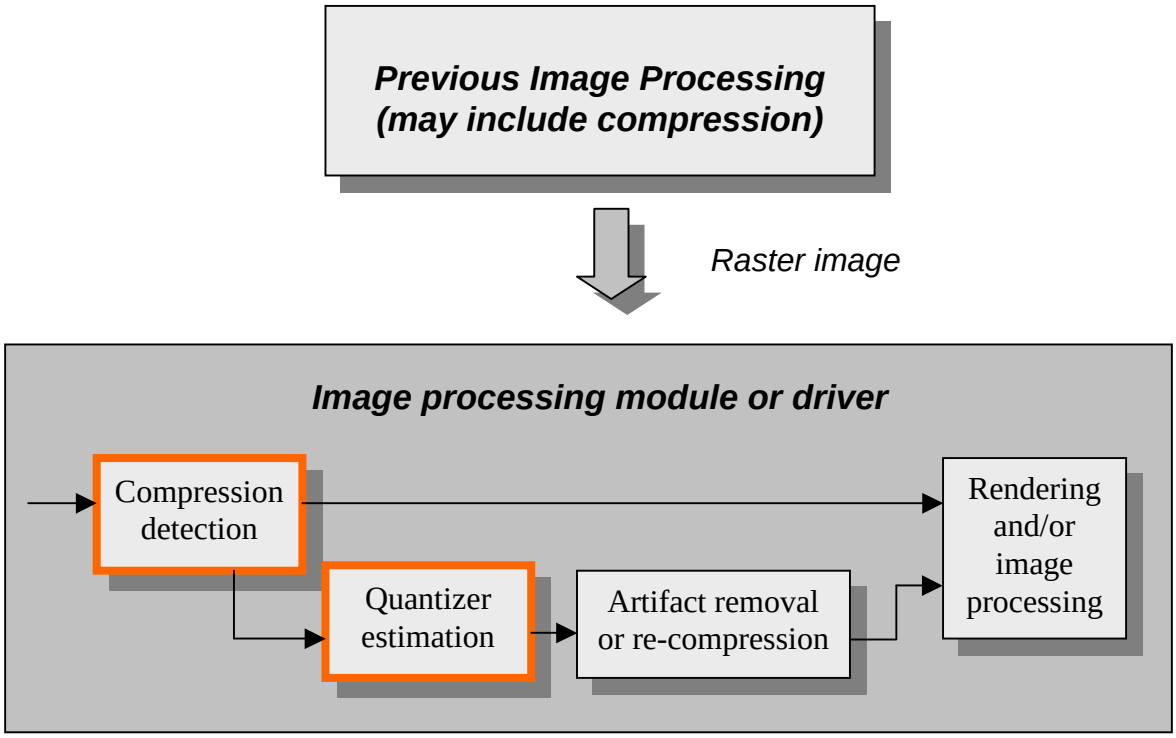

Figure 1 – Image processing module receives the bitmap image and processes it. In order to remove possible artifacts, it first has to determine whether the image has been compressed in the past and estimate its quantizer table. The information is, then, used to remove possible compression artifacts. We are only concerned with the "compression detection" and "quantizer estimation" operations.

In Fig. 1, the image received by the processing module is a bitmap. We first detect if the image has been compressed. In this paper we are only dealing with JPEG [1] compression of monochrome images, hence the only useful compression parameter that affects the image quality is the quantizer table, i.e. the quantizer step sizes applied to the discrete cosine transform (DCT) coefficients. The quantizer table is estimated from the decompressed raster data and this information can be made available to an artifact removal routine. Frequently, algorithms to remove JPEG artifacts use the quantizer information to estimate the amount of distortion caused by quantization so that over-blurring can be avoided. There are several alternatives for JPEG artifact removal. The method recommended in [1] as well as those in [2]-[9], serve as good examples.

Particularly, we favor the implementation of [3],[4], but we are not concerned with artifact removal techniques here. We are only concerned with the steps of detecting a previous JPEG compression and estimating the quantizers used.

In this paper, a method for the maximum likelihood estimation (MLE) [10] of JPEG quantization tables is presented, where the only information available is a bitmap of the decoded image. The method is used in conjunction with a very simple and efficient method to identify if an image has been previously JPEG compressed. In our experiments, we used the popular IJG JPEG implementation [11], so that, throughout the paper, we identify quantizer tables with so called "quality factors" (QF) with which the reader might be already familiar. The QF are just reference numbers which range from 1 to 100 [11], where QF=100 means that all quantizer steps [1] are unity, thus yielding the best quality JPEG can possibly achieve.

A simple method to detect previous JPEG compression is presented in Sec. 2. A suitable likelihood function for the estimation of the quantizer steps is discussed in Sec. 3, while the MLE method is presented in Sec. 4. Reports of experimental results are presented in Sec. 5 and the conclusions of this paper are presented in Sec. 6.

2. DETECTING JPEG COMPRESSION

The first step in the identification of the image's compression history is to identify if the image has been compressed. We are just concerned with JPEG compression and artifacts derived from block based image coding. We look for blocking signatures in the image as an evidence of JPEG compression. Similar ideas have been used before. For example, the method in [5] examines harmonics of the Fourier transform over blocks of 32x32 pixels in order to estimate the "blockiness" present in the image. Another method [6] estimates blocking artifacts by simply comparing the gradient at the border pixels with interpolated (projected) gradient values. In both cases, detecting blocking artifacts is necessary since they are concerned with removing the artifacts. In our case, we do not carry this step. We are only interested in a binary decision on whether or not the image has been compressed. Hence, algorithms that are both simpler and more robust can be applied.

Even "light" compression may leave small but consistent discontinuities across block boundaries. The proposed method detects images compressed with QF as high as 95. The idea is very simple: it assumes that if there is no compression the pixel differences across blocks should be similar to those within blocks. Assume the block grid is known. We then compute a sample of differences within a block and spanning across a block boundary, as illustrated in Fig. 2. For each block *(i,j)* we compute

$$Z'(i,j) = | A - B - C + D | \qquad Z''(i,j) = |E - F - G + H|, \tag{1}$$

where *A* through *H* are the values of the pixels in the positions depicted in Fig. 2. Next, one computes the normalized histograms $H_I(n)$ and $H_{II}(n)$ of the $Z'(i,j)$ and $Z''(i,j)$, respectively. The blocking signature measure that we use is the difference between histograms, i.e.

$$K = \sum_n \left| H_I(n) - H_{II}(n) \right| . \tag{2}$$

K can be compared to a threshold or given as a confidence parameter. Figure 3 serves to illustrate the method. Figure 3(a) shows histograms $H_I(n)$ and $H_{II}(n)$ for a typical image without compression, while Fig. 3(b) shows the same after the image underwent compression with QF 75 (recommended for excellent image quality [1]). The absolute histogram differences $|H_I(n)-H_{II}(n)|$ are shown in Fig. 3(c). In fact, *K* is related to the area under the curves in Fig. 3(c).

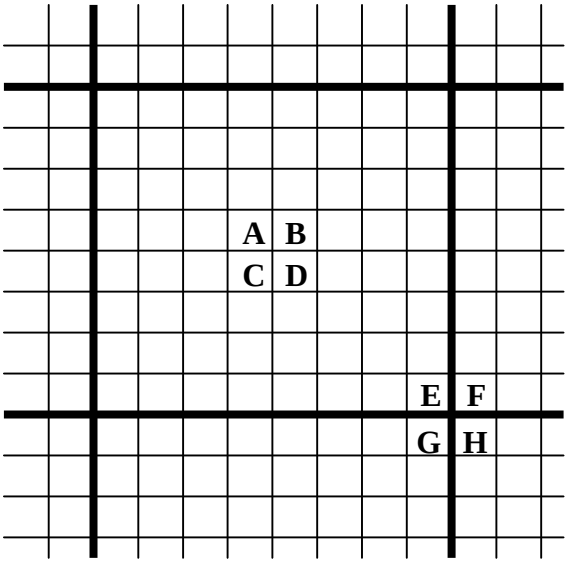

Figure 2 - For each block the numbers $Z'=|A-B-C+D|$ and $Z''=|E-F-G+H|$ are computed, i.e. involving same pixel pattern but spanning or not multiple blocks.

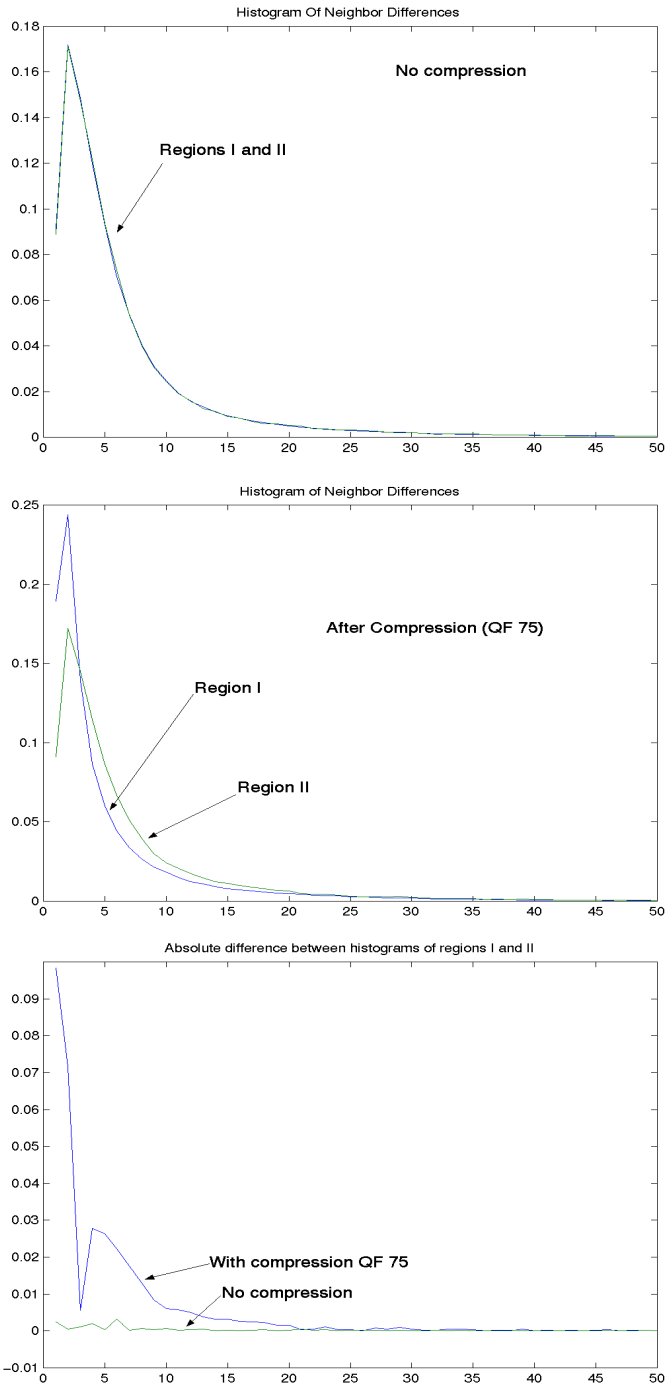

Figure 3: Histogram of neighbor differences comparing region I (within block) and II (across blocks). (a) (Top) No compression is applied to image. (b) (center) Moderate compression is applied to image (IJG code with quality factor 75). (c) (bottom) Difference in histograms (I and II) for both cases

Commonly, the grid origin is the same as the image origin point. This is generally the case if there is no image cropping or pasting. In case one wants to find the grid, let $\{y(m,n)\}$ be the image pixels, then the grid origin can be chosen as the pair $\{(p,q) \mid 0 \le p \le 7,\ 0 \le q \le 7\}$ that maximizes E_{pq}, where

$$E_{pq} = \sum_i \sum_j \left| y(8i+p, 8j+q) - y(8i+p, 8j+q+1) - y(8i+p+1, 8j+q) + y(8i+p+1, 8j+q+1) \right|. \quad (3)$$

In other words, the grid should be aligned with the positions where the horizontal and vertical neighbor differences, in a periodic displacement, are at their maximum. If there is no compression and, therefore, no blocking, all E_{pq} in (3) should be similar and the grid origin will be picked randomly.

There are alternatives to simplify the grid location procedure, for example by selecting a few image rows and columns in order to perform the summation. There are many ways to detect blocking; the proposed computation of $Z'(i,j)$ and $Z''(i,j)$ is just one instantiation.

3. THE LIKELIHOOD FUNCTION FOR QUANTIZER STEP ESTIMATION

Given that the image has been detected as being previously compressed using JPEG, we would like to estimate the quantizer table used. In this section, we analyze the probability distribution function of the DCT coefficients and formulate a likelihood function. We assume that the elements of the quantization matrix are integers, as this is almost always true for transform coders and is certainly true for JPEG.

JPEG compression is typically performed in three steps: discrete cosine transform (DCT), quantization, and entropy coding. At the decoding side, the processes are reversed. The data are entropy decoded, dequantized, and inverse transformed (IDCT).

It was reported that the DCT coefficients typically have a Gaussian distribution for DC component and Laplacian distributions for AC components [1,12]. In the quantization step, the DCT coefficients are discretized. They are recovered in the dequantization step as the multiples of quantization intervals. Specifically, if $Y(m,n)$ denotes the (m,n)-th component of a dequantized JPEG block in the DCT domain, it can be expressed as $k\ q(m,n)$, where $q(m,n)$ is the (m,n)-th entry of the quantization table, and k is an integer. Figure 4 shows a typical discrete histogram of $Y(m,n)$ for all the blocks in an image. Non-zero entries occur only at the multiples of $q(m,n)$. The envelop of the histogram is roughly Gaussian for the DC component and Laplacian for the AC components.

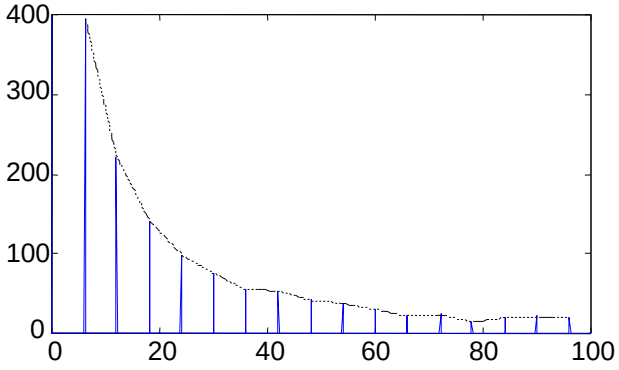

Fig. 4. Histogram of | $Y(0,1)$ | for image Lena ($q(m,n)$ =6), which is formed by spaced peaks. The envelope, which has an exponential decay is only included as a reference.

Once the histogram of $Y(m,n)$ is established, the estimation of $q(m,n)$ is fairly straightforward. However, $Y(m,n)$ only exists as an intermediate result. It is typically discarded after decompression. Theoretically, $Y(m,n)$ can be re-calculated from the decoded image block, since IDCT is reversible. Nevertheless in reality, the DCT of the image block usually generates $Y^*(m,n)$, which is not exactly $Y(m,n)$, but an approximated version of it.

There are mainly two sources of errors. Both of them were introduced during the IDCT calculation. First, the decompressed pixel values, typically integers, are rounded from real numbers. Second, any number greater than 255 or smaller than 0 for a pixel value, which is normally limited to 8 bits, is truncated to 255 or 0, respectively. The truncation errors can be very significant, particularly at low bit-rates. Furthermore, they are difficult to model. Fortunately, they occur only in a small percentage of blocks and these blocks can be detected. We will discuss the treatment of these blocks in Section 4. In this section, we assume all the blocks are not truncated, and we will focus on rounding errors. If we assume the rounding error for each pixel is an independent, identically distributed, random variable with a uniform distribution in the range of [-0.5, 0.5), then the Gaussian distribution will be a natural candidate for modeling $Y^*(m,n)$ according to the Central Limit Theorem. The mean and the variance of the distribution can be calculated as $Y(m,n)$ and 1/12, respectively. With the exception of uniform blocks, which will be discussed later, our simulation have shown that the Gaussian model is fairly reasonable. Although the data distribution has shorter tails than the Gaussian distribution, it fits the model very well when the deviation is small. At the tail part, we can show that $|Y^*(m,n) - Y(m,n)|$ is limited. The reason is fairly simple. As the rounding error for each pixel does not exceed 0.5, the total rounding error $|Y^*(m,n) - Y(m,n)|$ is bounded by:

$$|Y^*(m,n) - Y(m,n)| \le B(m,n) = \sum_{jk} 0.5\, c(j)\, c(k) \,|\, \cos((2j+1)m\pi/16)\, \cos((2k+1)n\pi/16) \,|$$

(4)

where $c(j) = 1/\sqrt{2}$ for $j = 0$, and $c(j) = 1$ otherwise. The right hand side of (4) can be simplified as

$$B(m,n) = D(m)\, D(n),$$

(5)

where

$$D(m) = \begin{cases} 2 & \text{for} \quad m = 0,4 \\ 2\cos(\pi/4) & \text{for} \quad m = 2,6 \\ 2\cos(\pi/4)\cos(\pi/8) & \text{for} \quad m \text{ odd} \end{cases}.$$

(6)

Consequently, we assume $Y^*(m,n)|Y(m,n)$ has a modified Gaussian distribution. It has a Gaussian shape in the range of $\pm B(m, n)$, and is zero outside the range. Specifically

$$p\left[Y^*(m,n)\middle|Y(m,n)\right] = G[Y^*(m,n) - Y(m,n)] = \begin{cases} 0 & \left|Y - Y^*\right| > B(m,n) \\ \dfrac{e^{-6(Y-Y^*)^2}}{Z} & else \end{cases}$$

(7)

where Z is a normalizing constant.

For a block with uniform intensity, where $Y(m,n)$ is non-zero only for the DC term, the rounding errors for all the pixels in the block have the same value, and are highly correlated. As a result, the Central Limit Theorem and the Gaussian model are no longer valid. In fact, in these blocks, $Y^*(m,n)$ has a zero value for all AC coefficients and a uniform distribution for the DC. As it will be explained in Section 4, the uniform bocks are not considered in the estimation process. We assume that in the following discussion all the blocks are non-uniform.

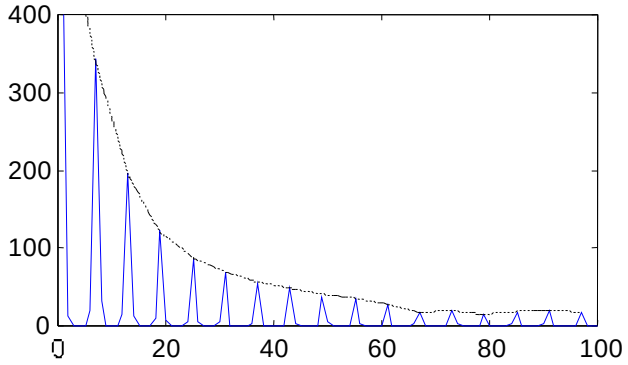

Fig. 5. Histogram of $|Y^*(0,1)|$ for image Lena ($q(m,n)$ =6). As in the case of Fig. 4, the decaying envelope is included only as a reference.

Excluding the uniform blocks and the blocks with truncation, the distribution of $Y^*(m,n)$ is shown in Fig. 5. It is a blurred version of Fig. 4. The discrete lines in Figure 4 become *bumps* of Gaussian shapes in Figure 5. These *bumps* remain separated if the quantization interval is not too small, i.e., $q(m,n) > 2B(m,n)$. They may touch each other if $q(m,n) \le 2B(m,n)$. The probability function of $Y^*(m,n)$ for given $q(m, n)$ can be determined as (For simplicity, we will drop the index (m,n) for rest of this section and for the first part of the next section):

$$p[Y^*; q] = \sum_k G[Y^* - (r+k)q] p_Y[rq] \tag{8}$$

where

$$r = \text{round}\,[Y^* / q], \tag{9}$$

and the function $G(.)$ is given in (7). As $G(.)$ is zero beyond the range of $[-B, B]$, there is only one non-zero term in the summation in (8) for $q > 2B$. For a small q ($q(m,n) \le 2B(m,n)$), there will be a few terms. The summation is then over all integers k such that $-B \le kq \le B$. The first factor in (8) characterizes the *bumps*. The second factor, which represents the envelop, has roughly a Gaussian (for DC) or a Laplacian (for AC) distribution.

Based on the above analysis, if we assume the statistics of Y^* are independent for each image block, the likelihood function can be established as

$$L(q) = \sum_{s} \log\left\{p\left[Y_s^*; q\right]\right\},$$

(10)

where the index s refers to the s-th block and the distribution of Y_s^* is given in (8). The MLE of q, q^* can then be formulated as

$$q^* = \arg\max_q L(q) = \arg\max_q \left\{\sum_s \log\left\{\sum_k [G(Y_S^* - rq - kq)]\right\} + \sum_s \log p_Y\left[r_s q\right]\right\},$$

(11)

There are two terms in the optimization. The first term fits the data Y_s^* to the multiples of q. The second one, which matches the overall DCT distribution can be further calculated as

$$\sum_s \log p_Y\left[r_s(m,n)q\right] = N \log q + Cons\tan t$$

(12)

where N is the total number of blocks used in estimation. Consequently, (11) becomes:

$$q^* = \arg\max_q \left\{\sum_s \log\left\{\sum_k [G(Y_S^* - rq - kq)]\right\} + N \log q\right\} .$$

(13)

From another point of view, the second term in equation (13) provides a penalty for a smaller q. Suppose $q_2 = mq_1$, where m is a positive integer. It is always true that the first term in (13) is no smaller for $q = q_1$ than it is for $q = q_2$. In other words, it is biased towards a smaller q. This bias is compensated by the second term.

We have so far assumed that Y^* is a real number. Practically, the outputs of many DCT routines are integers. Also, as it will been shown in the next section, approximating Y^* with an integer would result in simpler calculations. If we denote Y' as the integer rounded from Y^*, the probability function for Y' can be obtained as:

$$P[Y';q] = \int_{Y'-0.5}^{Y'+0.5} \mathrm{p[Y, q]\,dY} = \int_{Y'-0.5}^{Y'+0.5} \sum_k G[Y - rq - kq] p_Y\left[rq\right] dY \ .$$

(14)

The likelihood maximization becomes:

$$q^* = \arg\max_q \left\{\sum_s \log\left\{\int_{Ys'-0.5}^{Ys'+0.5} \sum_k [G(Y - rq - kq)] dY\right\} + N \log q\right\}.$$

(15)

4. THE MLE ALGORITHM

In this section, we describe the algorithm that calculates the MLE estimation. First, we detect two kinds of image blocks: uniform blocks and those with truncation. For each image block, we find the maximum and the minimum values of the block. If the maximum value is 255 or the minimum value is 0, the block may contain truncated pixels. If the maximum value is equal to the minimum value, the block is uniform. Both kinds of blocks are excluded from further processing.

The data in the remaining blocks are used for MLE estimation. Neither the likelihood defined in (13) nor its integer approximation (15) can be optimized analytically. However, numerical complexity can be significantly reduced in maximization of the integer version (15).

It can be easily shown that the probability function given in (14) depends only on Y'-rq, which is Y' modulated by q. If we denote $d = Y'$-rq, d can assume only q distinguished values. As a result, the probability function for each d value can be pre-calculated, and we only have to count the number of occurrences when d assumes each of the q different values.

To further minimize the computation, not all possible values are tested in maximization of (15). If a histogram is built for Y', peaks can be found at the multiples of q. Normally, the highest peak outside the main lobe (0 and its vicinity) corresponds to q or one of its multiples. Based on this observation, we restrict the search to be Q, Q+1, Q-1 and their integer fractions, where Q is the highest peak outside the main lobe ($Y' > B$). For example, if Q is found to be 10. Optimization in (18) will be calculated for q = 1, 2, 3, 5, 9, 10, and 11.

For a particular coefficient (m,n), it is possible that no peak is detected outside the main lobe. This occurs when $|Y_s'(m,n)|$ is small for all the blocks in the image. Typically, the histogram decays rapidly to zero without showing any periodic structure. The data contain little or no information about $q(m,n)$. The estimation fails in that case and $q(m,n)$ is declared to be "undetermined".

This estimation algorithm can also serve as another method to determine if the image has been previously JPEG compressed. If all the quantization levels are estimated to be 1, it is a good indication that the image has not been JPEG compressed.

It is fairly common that the quantization table used is a scaled version of a "standard" table, for example the JPEG baseline table. In other words,

$$q(m,n) = Q\, t(m,n) \tag{16}$$

where $t(m,n)$ is the (m,n)th entry of the standard quantization table and Q is the scaling factor. In this case, we only need to estimate Q. This can be performed by maximizing the joint likelihood, which is the product of formula (15) over all the entries (m,n). However, a simpler, but equally powerful method exists. We can first estimate, using (15), a few "reliable" entries that $|Y_s'(m,n)|$'s are not too small. Q^*, the estimation of $Q$is then simply obtained from these entries as

$$Q^* = q^*(m,n) / t(m,n) \tag{17}$$

If a conflict occurs, i.e. different entries give different estimation, a joint likelihood optimization may follow. Otherwise, (17) serves as the final estimation, and the quantization matrix can be retrieved using (16).

5. EXPERIMENT RESULTS

The detection algorithm was tested using various images and QFs and results are shown in Table I. For a given threshold T_K found empirically at T_K =0.25, we can detect most compressed images with QF as high as 90.

Table I - Values of K for several images and quality factors using the proposed algorithm. By thresholding the values to 0.25 one can detect JPEG compressed images with some confidence. Values of K>0.25 are highlighted. Note that with this threshold, detection is possible with quality factors as high as 90.

Image	Orig	Q100	Q90	Q70	Q50	Q30	Q10
Baby	0.0779	0.0889	***0.3998***	***0.7072***	***0.7982***	***0.9058***	***1.1212***
Barbara	0.1106	0.1131	***0.4732***	***0.6795***	***0.7581***	***0.8597***	***1.1073***
Chapel	0.0484	0.0468	***0.5243***	***0.6330***	***0.7351***	***0.7863***	***0.8758***
Cameraman	0.2279	0.2258	***0.6171***	***0.6878***	***0.8085***	***0.7555***	***0.7971***
Kids	0.0266	0.0315	***0.5943***	***0.9024***	***1.0789***	***1.2194***	***1.2409***
Lake	0.0506	0.0496	0.1855	***0.4167***	***0.5106***	***0.5868***	***0.8100***
Lena	0.0844	0.0678	***0.6359***	***0.9514***	***1.0343***	***1.1013***	***1.1232***
Mixed	0.0967	0.1071	***0.3341***	***0.5213***	***0.7133***	***0.7921***	***0.8987***
Shaver	0.0355	0.0897	***0.6339***	***0.6844***	***0.5876***	***0.5434***	***0.5790***
Wine	0.1169	0.0978	0.1794	***0.3527***	***0.4689***	***0.5820***	***0.8743***

The estimation algorithm was tested on images compressed with different quantization matrices. When assumption (16) is used, i.e. the quantization table is a scaled version of a standard table, then the estimation provided correct results in all experiments without a single error. However, when the scaling assumption was

dropped, occasional errors occurred at very low or very high bit rates. Fig. 6 summarizes the results for 7 pictures compressed with various quality factors (without the scaling assumption), i.e. for each QF, 7x64=448 quantization steps were estimated. When the QF is low (low bit rate), many cases were declared "undetermined". This is because the more aggressively an image is compressed, the more likely that the $Y_s(m,n)$ are quantized to zero, hence more $Y_s'(m,n)$ fall into the main lobe. The number of "undetermined" cases in general decrease as the quality factor improves. Estimation errors may also be found occasionally at low bit rate. In our experiment it occurred typically when there was only one $Y_s'(m,n)$ that was located beyond the main lobe and its value deviated from $Y_s(m,n)$ by 1 or 2. In this case, the only data available is itself wrong, so that it becomes impractical to "guess" the right quantizer step. However, the errors are commonly very small, typically one value and since there is just one or very few values over the image, any small estimation error in such sporadic data is not likely to cause any significant impact in the image processing algorithm driven by the quantizer estimation method. Nevertheless, the estimation performance indeed deteriorates at very high bit rate (QF > 95). The "bumps" in the histogram start to be squeezed together and the periodic structure becomes less prominent when $q(m,n)$ becomes very small. If at the same time the coefficients are small and many of the samples fall into the main lobe, then an estimation error (or an "undetermined" case) occurs. In summary, the estimation may fail at low bit rate due to an insufficient number of effective data samples. At very high bit rate, its performance is hindered by less information carried by each sample.

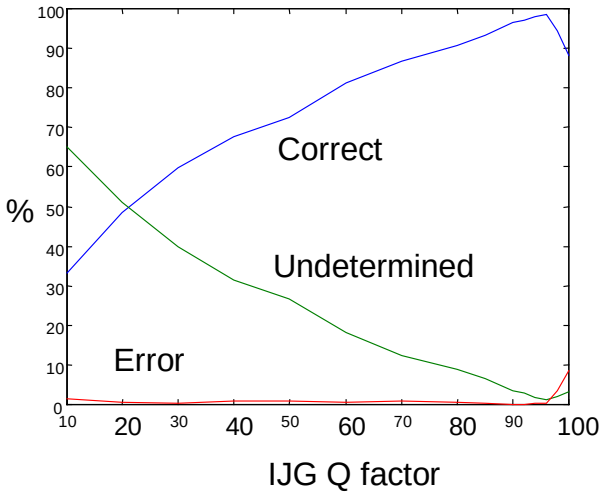

Figure 6. Performance evaluation. Percentage of correct estimation is high and of error is very low. Undetermined coefficients are those can not be estimated due to lack of information. Seven images were tested so that for each QF, 448 step sizes are estimated.

6. CONCLUSIONS

A method was developed for the reliable estimation of the JPEG compression history of a bitmapped image. Not only an efficient method was presented to detect previous JPEG compression but also a very reliable MLE method was devised to estimate the quantizer table used. The detection method can trace JPEG images which are visually undistinguishable from the original and is extremely reliable for higher compression ratios, which is the range of interest. Detection can be made with QF as high as 95. It is likely that there will be no need for further processing the image for high QF, so that it is more important to accurately identify the high-compression cases. Our method has not failed yet in those circumstances.

The quantizer table estimation method is very reliable. Errors are infrequent and when they occur they usually throw the quantizer step estimation off by one value. These errors are often inevitable. For example, if there is just one non-zero DCT coefficient in a particular frequency and it happens to be itself off by one value, it becomes impractical to "guess" the right step. For cases of very high quality compression, the quantizer steps are very small and are overpowered by the imprecision of the coefficients. If the quantization table is known to be a scaled version of a standard table, these infrequent errors can be virtually eliminated by a joint estimation. The computation complexity of the estimation is mainly DCT transformation. For a large image, we may use only a portion of the blocks to save computation.

We have just presented results for monochrome images. This subject is an ongoing research topic and color images are the natural next step.

REFERENCES

[1] W. Pennebaker and J. Mitchell, *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, 1993.

[2] R. Rosenholtz and A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding, *IEEE Trans. Circuits Sys. Video Technology*. Vol2, pp. 91-94, Mar. 1992.

[3] Z. Fan and R. Eschbach, "JPEG decompression with reduced artifacts", *Proc. IS&T/SPIE Symposium on Electronic Imaging: Image and Video Compression*, San Jose, CA, Feb. 1994.

[4] Z. Fan and F. Li, "Reducing artifacts in JPEG decompression by segmentation and smoothing", *Proc. IEEE International Conference on Image Processing*, Vol. II, pp.17-20, 1996.

[5] K. T. Tan and M. Ghanbari, "Blockiness detection for MPEG-2-coded video", *IEEE Signal Processing Letters*, Vol. 7, pp. 213-215, Aug. 2000.

[6] S. Minami and A. Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 5, pp. 74-82, Apr. 1995.

[7] Y. Yang, N.P. Galatsanos and A.K. Katsaggelos, "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, Dec. 1993.

[8] J. Luo, C. W. Chen, K. J. Parker, and T. S. Huang, "Artifact reduction in low bit rate DCT-based image compression," *IEEE Trans. Image Processing*, vol. 5, pp. 1363-1368, 1996.

[9] J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Processing Letters*, vol. 5, pp. 33--35, February 1998

[10] Sir M. Kendall and A. Stuart, *The Advanced Theory of Statistics*, Vol. 2, Macmillan, New York, 1977.

[11] Independent JPEG Group library, http://www.ijg.org.

[12] R. J. Clarke, *Transform Coding of Images*, Academic Press, 1985.