# MAXIMUM LIKELIHOOD ESTIMATION OF JPEG QUANTIZATION TABLE IN THE IDENTIFICATION OF BITMAP COMPRESSION HISTORY

*Zhigang Fan and Ricardo de Queiroz*

Xerox Corporation, 800 Phillips Road, Webster, NY 14580
`zfan@crt.xerox.com` `queiroz@wrc.xerox.com`

## ABSTRACT

To process previously JPEG coded images the knowledge of the quantization table used in compression is sometimes required. This happens for example in JPEG artifact removal and in JPEG re-compression. However, the quantization table might not be known due to various reasons. In this paper, a method is presented for the maximum likelihood estimation (MLE) of the JPEG quantization tables. An efficient method is also provided to identify if an image has been previously JPEG compressed.

## 1. INTRODUCTION

To process previously JPEG [1] coded images the knowledge of the quantization table used in the compression step is sometimes required. This may happen for example in JPEG artifact removal. In eliminating blocking and/or ringing artifacts [2,3], the information about quantization steps may help to determine the filter parameters and to set the DCT constraints. Another example is to re-compress the JPEG images. Using the same quantization table will typically minimize additional quantization error. The information of quantization table may not always available for a number of reasons. Mostly, this is because the image is obtained only in its reconstructed bitmap representation. No record is kept for its compression history. A typical example is a display or printing driver, which is handed a bitmap of given dimensions as it is told to render the image at a given size with given properties. No information about the bitmap's history is commonly provided.

In this paper, a method for the maximum likelihood estimation (MLE) [4] 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 another approach used to identify if an image has been previously JPEG compressed.

## 2. DETECTION OF JPEG COMPRESSION

The algorithm may be started by first identifying whether the bitmap has been previously JPEG compressed. Our detection algorithm works well even for very high quality, low-ratio, compression. It can detect images compressed with IJG JPEG [5] quality factors as high as 95. In interest of the allotted space, the details of the detection algorithm will be given in this paper's extended version [6], as we will only briefly describe its main features.

The detection algorithm computes the difference between neighbor pixels either across block boundaries or within a block. The differences are compiled in a form of a histogram and the differences can be one or two-dimensional [6]. The concept is that once you compress the images the differences across block boundaries should be larger than those within the block. Thus, by comparing the histograms of the differences one can make a decision whether the image has been compressed before.

We compare the sum of the absolute difference between normalized histograms of the absolute differences between pixels across or not across the block boundaries. The resulting number is then compared to a threshold or given as a confidence parameter. As a matter of fact, the method works as a "blocking" estimator and can also be used to some extent to estimate how much compression distortion was applied to the image. A similar method can also be used to align the block grid in case the image has been cropped. See [6] for further details.

## 3. THE LIKELIHOOD FUNCTION

We assume that the image has been detected as being previously compressed using JPEG. We would like to estimate the quantizer table using the MLE method. In this section, we analyze the probability distribution functions of the DCT coefficients and formulate the likelihood function.

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 [1,7]. 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 1 shows a typical histogram of $Y(m,n)$ for all the blocks in an image. The histogram appears to be discrete. The non-zero entries occur only at the multiples of $q(m,n)$. The envelop of the histogram, as given in Fig. 1, is roughly Gaussian for the DC component and Laplacian for the AC components.

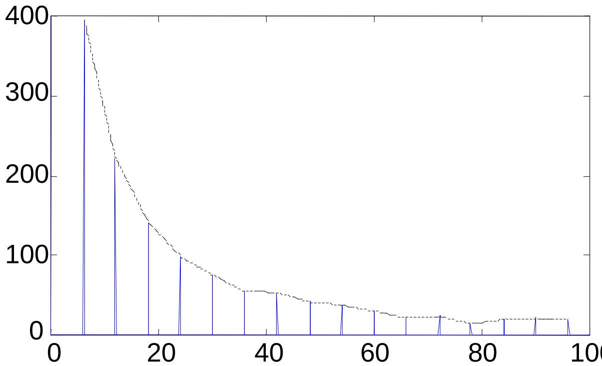


Fig. 1. Histogram of | $Y(0,1)$ | for image Lena ($q(m,n)$ =6)

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 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-rate. 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 independently identically distributed with a uniform distribution in the range of [-0.5, 0.5), a 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 showed that the Gaussian model is fairly reasonable. Although the data have shorter tails than the Gaussian distribution, they fit 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 by:

$$|Y^*(m,n) - Y(m,n)| \le D(m)\,D(n), \qquad (1)$$

where

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

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

$$p\left[Y^*(m,n)\middle|Y(m,n)\right] = \begin{cases} 0 & \left|Y - Y^*\right| > D(m)D(n) \\ \dfrac{e^{-6(Y-Y^*)^2}}{Z} & else \end{cases} \qquad (3)$$

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.

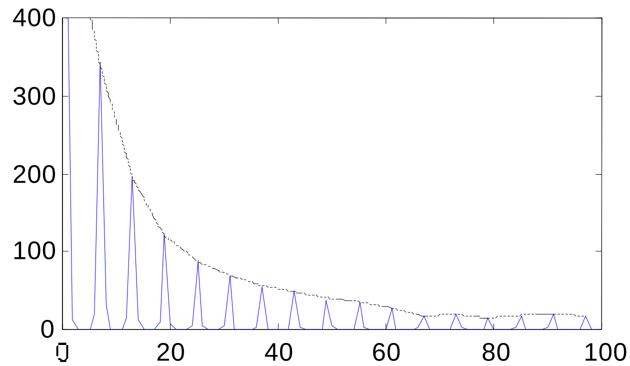


Fig. 2. Histogram of $|Y^*(0,1)|$ for image Lena ($q(m,n)$ =6)

Excluding the uniform blocks and the blocks with truncation, a typical histogram for $Y^*(m,n)$ is shown in Fig. 2. It is a blurred version of Fig. 1. The discrete lines in Figure 1 become *bumps* of Gaussian shapes in Figure 2. These *bumps* remain separated if the quantization interval is large, i.e., $q(m,n) > 2D(m)D(n)$. They may touch each other if $q(m,n) \le 2D(m)\,D(n)$. If the *bumps* are well separated, $Y(m,n)$ can be uniquely determined from $Y^*(m,n)$ as

$Y(m, n) = q(m,n)\, r(m,n),$ (4)

where

$r(m,n) = \text{round}\,[Y^*(m, n) / q(m, n)].$ (5)

For example, if $q(m,n)$ is 16, $D(m)D(n)$ is 4, and $Y^*(m,n)$ is 34. $Y(m,n)$ must be 32. The probability function of $Y^*(m,n)$ for given $q(m, n)$ can be determined as:

$p[Y^*(m,n);q(m,n)]=p[Y^*(m,n)\,|Y(m,n)]\; p_Y\,[Y(m,n)],$ (6)

where $p[Y^*(m,n) \mid Y(m,n)]$ is given in (3). Since the DCT coefficients can be modeled by a Gaussian (for DC) or a Laplacian distribution (for AC), we assume $p_Y\,[Y(m,n)]$ in equation (6) roughly follows the same distribution.

When $q(m,n) < 2D(m)D(n)$, $Y(m,n)$ can not be determined from $Y^*(m,n)$ with certainty. For example, if $q(m,n)$ is 3 and $Y^*(m,n)$ is 34, $Y(m,n)$ could be either 33 or 36. The probability given in (6) has to be revised as

$$p[Y^*(m,n);q(m,n)] = \sum_k p[Y^*(m,n) \mid kq(m,n)]\,p_Y\left[kq(m,n)\right] \quad (7)$$

where the summation is over all integers $k$ such that

$|Y^*(m,n) - k\, q(m,n)\,| \le D(m)D(n).$ (8)

As $p_Y[k\;\; q(m,n)]$ usually changes much slower than $p[Y^*(m,n) \mid k\, q(m,n)]$ does, (8) can be approximated as

$$p[Y^*(m,n);q(m,n)] = \sum_k p\left[Y^*(m,n) \mid kq(m,n)\right] p_Y\left[r(m,n)q(m,n)\right] \quad (9)$$

where $r(m,n)$ is defined in (5). Then, (6) can be considered as a special case of (9), in which only one term exists in the summation.

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

$$L(q) = \sum_S \log\left\{p\left[Y_S^*(m,n);q\right]\right\}, \quad (10)$$

where the index $s$ refers to $s$-th block and the distribution of $Y_s^*(m,n)$ is given in (9). The MLE of $q(m,n)$ can then be formulated as

$$q(m,n) = \arg\max_q L(q) = \arg\max_q \left\{\sum_s \log\left\{\sum_k p\left[Y_S(m,n) \middle| kq\right]\right\} + \sum_s \log p_Y\left[r_S(m,n)q\right]\right\}, \quad (11)$$

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

$$\sum_s \log p_Y\left[r_s(m,n)q\right] = -N\log\left[\sigma^*(m,n)\right] \quad (12)$$

where $N$ is the total number of blocks used in estimation, $\sigma^*(m,n)$ is the estimated parameter for Gaussian (for DC) or Laplacian (for AC) distribution. Specifically,

$$\sigma^*(m,n) = \begin{cases} \sqrt{\dfrac{1}{N}\sum_s r_S^2(m,n)} & DC \\ \dfrac{1}{N}\sum_s \left|r_S(m,n)\right| & AC \end{cases} \quad (13)$$

From another point of view, the second term in equation (11) provides a penalty for a smaller $q$. Suppose $q_2 = mq_1$, where $m$ is an integer. It is always true that the first term in (11) is no smaller for $q = q_1$ than for $q = q_2$. In other words, it is biased towards a smaller $q$. This bias is compensated by the second term, as $\sigma^*(m,n)$ will become smaller when $q$ increases, as indicated in (13).

## 4. THE MLE ALGORITHM

In the estimation of the quantization matrix, we first 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 to evaluate equation (11). The maximization defined in equation (11) does not have an analytical solution. Numerical complexity can be significantly reduced by the following modifications.

First, $Y^*(m,n)$ is rounded to an integer, which we denote as $Y'(m,n)$. Actually, the outputs of many DCT routines are integers. Hence, (11) can be revised as

$$q^*(m,n) = \arg\max_q \left\{\sum_{-\frac{q}{2}<i\le\frac{q}{2}} N(i)w(i,q) - N\log\left[\sigma^*(m,n)\right]\right\}, \quad (14)$$

where $N(i)$ is the number of blocks that satisfy

$Y'_s(m,n) - q\;\; r_s(m,n) = i.$ (15)

The coefficient $w(i,q)$ can be pre-calculated as

$$w(i,q) = \log \sum_{|j| \le \frac{D(m)D(n)-0.5-i}{q}} \left( \int_{i+jq-0.5}^{i+jq+0.5} g(x)dx \right) \qquad (16)$$

where the function g(x) is defined as

$$g(x) = \begin{cases} e^{-6x^2} & x \le D(m)D(n) \\ 0 & else \end{cases} \qquad (17)$$

It is apparent that $w(i,q) = w(-i,q)$.

To further minimize the computation, not all the possible values are tested in maximization of (15). If a histogram is built for $Y'(m,n)$, peaks can be found at the multiples of $q(m,n)$. Normally, the highest peak outside the main lobe (0 and its vicinity) corresponds to $q(m,n)$ 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 ($Y'(m,n) > D(m)D(n)$). For example, if $Q$ is found to be 10. Optimization in (15) will be calculated for $q$ = 1, 2, 3, 5, 9, 10, and 11.

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.

## 5. EXPERIMENT RESULTS

The estimation algorithm was tested on images compressed with different quantization matrices. Fig. 3 gives the experiment results for 7 pictures compressed with various IJG [5] quality factors. When quality factor 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 $Y_s(m,n)$ is quantized to zero and $Y_s'(m,n)$ falls into the main lobe. 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. The "undetermined" cases in general decrease as the quality factor improves. Nevertheless, the estimation performance deteriorates at very high bit rate (quality factor > 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, an estimation error or the "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.

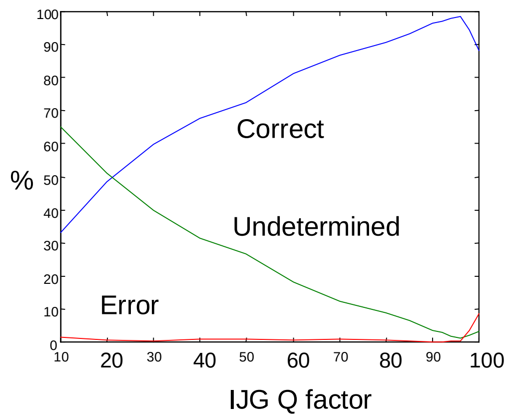


Fig. 3. Experiment Results

## 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 Technol.* Vol2, pp. 91-94, Mar. 1992.

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

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

[6] Z. Fan and R. de Queiroz, "Identification of bitmap compression history," in preparation for submission to *IEEE Trans. on Image Processing*.

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