JPEG Standard Uniform Quantization Error Modeling with Applications to Sequential and Progressive Operation Modes

Julià Minguillón
Jaume Pujol
Combinatorics and Digital Communications Group
Computer Science Department
Autonomous University of Barcelona
08193 Bellaterra, Spain
E-mail: jminguillon@ccd.uab.es

ABSTRACT

In this paper we propose a method for computing JPEG quantization matrices for a given mean square error or PSNR. Then, we employ our method to compute JPEG standard progressive operation mode definition scripts using a quantization approach. Therefore, it is no longer necessary to use a trial and error procedure to obtain a desired PSNR and/or definition script, reducing cost.

Firstly, we establish a relationship between a Laplacian source and its uniform quantization error. We apply this model to the coefficients obtained in the discrete cosine transform stage of the JPEG standard. Then, an image may be compressed using the JPEG standard under a global MSE (or PSNR) constraint and a set of local constraints determined by the JPEG standard and visual criteria.

Secondly, we study the JPEG standard progressive operation mode from a quantization based approach. A relationship between the measured image quality at a given stage of the coding process and a quantization matrix is found. Thus, the definition script construction problem can be reduced to a quantization problem.

Simulations show that our method generates better quantization matrices than the classical method based on scaling the JPEG default quantization matrix. The estimation of PSNR has usually an error smaller than 1 dB. This figure decreases for high PSNR values. Definition scripts may be generated avoiding an excessive number of stages and removing small stages that do not contribute during the decoding process with a noticeable image quality improvement.

Keywords: Image Compression, Progressive Image Transmission, JPEG Standard, Quantization Matrices, Definition Scripts

1. INTRODUCTION

Image compression is a fast-growing area in communications. The main goal of image compression is to reduce storage and transmission time requirements of on-demand applications such as Internet browsers or image manipulation programs. Image compression systems can be classified as *lossless* or *lossy*, depending on whether or not the original image can be exactly recovered from the compressed image. Unlike lossless systems, lossy systems exploit the properties of the human visual system, such as different frequency sensitivity, to obtain higher compression ratios. Moreover, lossy compression systems allow the user to establish a criterion between compression ratio and image quality. Here arises an important question: how can the user specify image quality or compression ratio before compressing such image? In this paper we solve the first problem, allowing the user to specify a desired MSE (or PSNR) for a given image.

The JPEG standard 1,2 has become the *de facto* standard for lossy image compression systems, owing to its properties: efficiency, symmetry and completeness. Both image quality and compression ratio are determined by the quantization matrices used in the compression process, which are stored within the JPEG file. The JPEG standard provides a default quantization matrix which produces a medium-to-high image quality and a sufficiently high compression ratio for most images. The quantization matrix is usually multiplied by a scalar (the quality factor) to obtain different image qualities and compression ratios. This method is known as the *classical scaling method*.

Since there is not a clear relationship between the JPEG quality factor and the decompressed image quality, the user is obliged to try several quality factors until one of them fits his requirements of image quality and/or compression ratio. This is computationally expensive because the image must be compressed, then decompressed and, finally, MSE has to be measured in order to know image quality, Instead of this trial and error procedure, we present a method based on a Laplacian model which allows the user to obtain a quantization matrix for a given MSE (or PSNR) at a reduced cost when compared to the trial and error procedure.

One of the most powerful capabilities of the JPEG standard is provided through the progressive operation mode, which is capable of coding an image using several passes, with increasing image quality produced at each pass. An initial coarse image is decoded in a short time, and then it is refined in each pass until a final image is achieved. Common applications such as Internet browsers incorporate progressive encoding to reduce the download time of web pages. The most widespread JPEG software library supporting progressive operation mode is provided by the Independent JPEG Group3.

Progressive operation mode is controlled through the use of a definition script. A definition script is a list of bands or scans, where the i-th band contains information about the coefficients and bit planes that will be sent in the i-th pass. This may be too complicated for non-experienced users, so a default definition script is usually provided by the applications requiring progressive operation mode. Although this default script may achieve good results for basic usage of the progressive operation mode, it will probably not satisfy special user requirement. The method presented in this paper can be employed to construct and evaluate several definition scripts for a given image and a final image quality or sequence of image qualities.

This paper is organized as follows. Section 2 describes uniform quantization of Laplacian sources. Section 3 describes the JPEG standard progressive operation mode, considering it as a quantization problem. Section 4 establishes a relationship between the quantization error of a Laplacian source and the error measure used to compute the quality of the compressed image. The matrix construction algorithm is described in Section 5, and several experiments using the JPEG standard monochrome corpus set are also presented here. Algorithms for definition script construction and evaluation are described in Section 6 and several examples of definition scripts are evaluated. Finally, conclusions and suggestions for future research are given in Section 7.

2. UNIFORM QUANTIZATION OF LAPLACIAN SOURCES

Let X be a zero mean Laplacian random variable, and $\tilde{X}$ be the uniformly quantized version of X using a step size Q. Let X' denote the reconstructed version of X using uniform quantization. The *mean square error* (MSE) due to uniform quantization is defined as

$$MSE = E[(X - X')^2] = \int_{-\infty}^{\infty} (x - x')^2 f_X(x)\, dx \tag{1}$$

where

$$f_X(x) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}}{\sigma}|x|}, \quad \tilde{X} = \left\lfloor \frac{X}{Q} + \frac{1}{2} \right\rfloor, \quad X' = Q\tilde{X}. \tag{2}$$

Notice that in this case we are not trying to find an optimal quantizer 4,5 but to study the effects of uniform quantization on a Laplacian source. An analytical expression for uniform quantization error**?** may be found by decomposing Eq. (1) in subintervals of size Q specified using the different integer values q that $\tilde{X}$ can take. Then,

$$MSE(Q, \sigma) = \sigma^2 - \frac{Q\sqrt{2}\sigma e^{-\frac{Q}{\sigma\sqrt{2}}}}{1 - e^{-\frac{Q\sqrt{2}}{\sigma}}}. \tag{3}$$

2.1. Computing Q for a given MSE

Our principal goal is to obtain a method for computing the quantization factor Q needed to achieve a given MSE for a zero mean Laplacian source with variance σ^2. Solving Eq. (3) for Q yields

$$Q(MSE, \sigma) = \sigma\sqrt{2}F^{-1}\left(1 - \frac{MSE}{\sigma^2}\right) \tag{4}$$

where $F(t) = t/\sinh(t)$ is a smooth decreasing function in the interval $[0, \infty)$. A single zero for $F(t) = C$ can be easily found using a numerical method such as Newton-Raphson. Usually, no more than a few iterations are needed to obtain a good solution, except when t is almost zero or t is rather large. In order to avoid this problem we compute t from $x = F(t)$ as follows

$$t = \begin{cases} 0 & \text{if } x > 0.999 \\ 17.363 & \text{if } x < 10^{-6} \\ F^{-1}(x) & \text{otherwise.} \end{cases} \tag{5}$$

It is very important to note in Eq. (4) that MSE cannot be greater than σ^2 (the maximum variance of the quantized variable is limited by the variance of the Laplacian source). Thus, $F^{-1}(x)$ has only to be computed for positive values of x. The fact that $0 \leq MSE \leq \sigma^2$ is a constraint that has to be taken into account when computing Q for a desired MSE. Additionally, there are more constraints due to the JPEG standard, such as those imposed by quantization values, as shown in Section 4.3.

3. JPEG STANDARD OPERATION MODES

The JPEG standard1,2 defines four operation modes: lossless, sequential, progressive and hierarchical. The sequential mode encodes an image in a single pass, while the progressive mode uses several passes, where each successive pass increases image quality.

In the JPEG standard, the input image is segmented into 8×8 blocks, and then each block is transformed using the discrete cosine transform6 (DCT). The DCT coefficients are independently quantized using integer scalar quantization. Then, the 64 coefficients are sorted using a zig-zag sequence, which exploits the energy compaction and decorrelation properties of the DCT. The first coefficient $I(0,0)$ is called the DC coefficient, and the other 63 coefficients $I(u,v)$, $u,v \neq 0,0$, are called the AC coefficients. Finally, Huffman or arithmetic coding is applied to pairs of zero run-lengths and magnitudes. Compression is achieved in the last two stages: first, DCT coefficients are quantized using Eq. (6). This stage causes an irreversible loss, because each quantization interval is reduced to a single point. Each coefficient is reconstructed by a simple multiplication, as shown in Eq. (7). Second, the quantized coefficients are efficiently coded using an entropy coder, which handles the runs of zeros produced in the quantization stage.

$$I_Q(u,v) = \left\lfloor \frac{I(u,v)}{Q_{u,v}} + \frac{1}{2} \right\rfloor \tag{6}$$

$$I'(u,v) = I_Q(u,v) Q_{u,v}. \tag{7}$$

The order in which these blocks of coefficients are encoded or sent is determined by the operation mode. In sequential operation mode, all 64 coefficients of a transformed block are operated upon, with blocks following a raster scan order, into a single band. In progressive operation mode, coefficients are segmented in one or more bit planes and sent in two or more bands. Progressive operation mode combines two powerful techniques: spectral selection and successive approximation. Spectral selection involves a sub-sampling in the DCT domain, taking advantage of DCT properties. Successive approximation consists of reducing coefficient magnitudes using a point transform which splits a coefficient into several bit planes. When both techniques are combined, one obtains full progression. Successive approximation cannot be used alone because the DC coefficient must be always sent first in a separate band. Thus, this paper considers spectral selection and full progression only.

Four fields are used for band description: start of spectral selection band (Ss), end of spectral selection band (Se), successive approximation bit position high (Ah), and successive approximation bit position low (Al).

3.1. Spectral selection

Spectral selection consists in grouping contiguous coefficients in bands and then transmitting these bands separately. The indexes of the first and the last coefficients sent in a band are described by Ss and Se, respectively. There are some restrictions: coefficients within a band must follow the zig-zag sequence ($Ss \leq Se$), bands must be non-overlapping, that is, the same coefficient cannot be sent in two or more bands, and the DC coefficient has to be sent in a unique first band ($Ss = 0 \Leftrightarrow Se = 0$). In contrast, bands of AC coefficients can be sent in any order, or they can not be sent. We will exploit this possibility in our method, trying to reduce image file size while maintaining image quality, by not sending coefficients that do not increase image quality. Because no bit plane segmentation is done, Ah and Al are always set to zero.

3.2. Full progression

In this case, each coefficient can be segmented in a sequence of bit planes, using a point transform1. Each bit plane is defined using Ah and Al. Ah is the point transform used in the preceding scan for the band of coefficients specified by Ss and Se, and it shall be set to zero for the first scan of each band of coefficients. Al is the current point transform, and because only one bit may be refined at the current stage, it is always Ah minus one but in the first stage, where $Ah = 0$ and Al defines the point transform.

4. UNIFORM QUANTIZATION ERROR

Once the image has been compressed and decompressed, quantization error can be measured using the definition of MSE given in Eq. (8). Given two $M \times N$-pixel images i and i',

$$MSE = \frac{1}{MN} \sum_{m,n=0,0}^{M,N} \left[i(m,n) - i'(m,n)\right]^2 . \tag{8}$$

Since DCT is a unitary transform, MSE can be rewritten as

$$\begin{aligned} MSE &= \frac{1}{64} \sum_{u,v=0,0}^{7,7} \frac{1}{B} \sum_{b=0}^{B-1} \left[I_b(u,v) - I'_b(u,v)\right]^2 \\ &= \frac{1}{64} \sum_{u,v=0,0}^{7,7} MSE_{u,v} \end{aligned} \tag{9}$$

where $B = MN/64$ is the number of 8×8 blocks in which the image is segmented, and I_b is the b-th block in raster scan. In Eq. (9) we suppose that each coefficient contributes in the same way to total MSE, that is $MSE_{u,v} = MSE$. A *weighted* MSE can be defined using a weighting factor for each $MSE_{u,v}$, as shown in Eq. (10). Usually, a weighted MSE is defined in order to include subjective criteria 7,8, such as the human visual system9,10 (HVS) response function, which we will denote by Φ. We will further discuss this subject in Section 4.4.

$$MSE = \frac{1}{64} \sum_{u,v=0,0}^{7,7} \Phi(u,v) MSE_{u,v}. \tag{10}$$

4.1. MSE for DC coefficient

It is easy to see from DCT definition that the DC coefficient is the block average (multiplied by 8). The DC coefficient can be modeled as a Gaussian mixture, and then the EM algorithm 11 can be used to extract the mixture components. This approach is computationally very expensive, so instead of looking for a good statistical model for the DC coefficient, we propose a simple model for MSE as a function of $Q_{0,0}$. Figure 1 shows the relation between $Q_{0,0}$ and the MSE obtained when quantizing the DC coefficient with $Q_{0,0}$. We propose to use a quadratic model to fit these data, due to goodness-of-fit and the simplification of the computation of the $Q_{0,0}$ factor for a given MSE. Therefore,

$$MSE_{0,0}(Q_{0,0}) = a_0 + a_1 Q_{0,0} + a_2 Q_{0,0}^2. \tag{11}$$

Using least squares fitting, the values $a_0 = 4.302$, $a_1 = 0.065$ and $a_2 = 0.082$ are found. Inverting Eq. (11) is straightforward: we compute $Q_{0,0}$ as the positive root, and then it is rounded to the nearest integer satisfying $Q_{0,0} \geq 1$. The minimum achievable MSE yielded by this model is 4.45 ($a_0 + a_1 + a_2$), which is not zero. This is consistent with the rounding error given by the JPEG quantization stage, because DCT coefficients are always quantized to integer values. Therefore,

$$Q_{0,0}(MSE_{0,0}) = \begin{cases} 1 & \text{if } MSE_{0,0} \leq 4.45 \\ \frac{-a_1+\sqrt{a_1^2-4a_2(a_0-MSE_{0,0})}}{2a_2} & \text{otherwise.} \end{cases} \quad (12)$$

The quadratic model is better for low values of Q than for large ones. This is interesting because the DC coefficient is usually quantized by a low value in order to avoid block artifacts caused by a coarse quantization.

4.2. MSE for AC coefficients

Usually, AC coefficients are modeled using a generalized Gaussian model 12, which includes a *shape factor* r that allows to generate a complete family of densities. For example, the Laplacian density is obtained with $r = 1$, and the Gaussian density, with $r = 2$. This model is more accurate, but is difficult for studying quantization error and obtaining a compact equation such as Eq. (3). Therefore, rewriting Eq. (3) we obtain the following model for AC coefficients,

$$MSE_{u,v}(Q_{u,v}, \sigma) = \sigma^2 - \frac{Q_{u,v}\sqrt{2}\sigma e^{-\frac{Q_{u,v}}{\sigma\sqrt{2}}}}{1 - e^{-\frac{Q_{u,v}\sqrt{2}}{\sigma}}}. \quad (13)$$

This model assumes that an AC coefficient may be modeled as a zero mean Laplacian source with variance $\sigma^2_{u,v}$. In this case, computing the quantization factor $Q_{u,v}$ from the desired error can be expressed as

$$Q_{u,v} = clip\left(\left\lfloor \sigma_{u,v}\sqrt{2}F^{-1}\left[1 - \frac{MSE_{u,v}}{\sigma^2_{u,v}}\right] + \frac{1}{2}\right\rfloor, 1, M\right), \quad (14)$$

where F is an auxiliary function (to simplify notation), $clip(x, a, b)$ is defined as a threshold clipping to ensure $a \leq x \leq b$, and M is the maximum quantization value ($M = 255$ for 8 bit quantization tables).

One problem this model shares with the model for the DC coefficient is that only integer values are valid for quantization factors. Although this may seem a secondary problem, for small quantization values this constraint may introduce important errors.

4.3. MSE distribution

To compute Q from a given MSE we have to use Eq. (4), while Eq. (9) is used to distribute total MSE among each $MSE_{u,v}$. This cannot be done independently for each coefficient, because there are some restrictions such as the one imposed by $\sigma^2_{u,v}$, that is,

$$0 \leq MSE_{u,v} \leq \sigma^2_{u,v} \quad (15)$$

and the global condition forced by Eq. (9),

$$MSE = \frac{1}{64}\sum_{u,v=0,0}^{7,7} MSE_{u,v}. \quad (16)$$

Moreover, if we are trying to compute a valid JPEG quantization table, each quantization factor has to be an integer value in the interval $[1, M]$, so

$$MSE(1, \sigma_{u,v}) \leq MSE_{u,v} \leq MSE(M, \sigma_{u,v}) \tag{17}$$

where $MSE(1, \sigma_{u,v})$ and $MSE(M, \sigma_{u,v})$ are the minimum and maximum achievable mean square errors due to quantization constraints and can be easily computed by using Eq. (3). These constraints determine the minimum and maximum achievable MSE for a given image, and they can be written as

$$MSE(1, \sigma_{u,v}) \leq MSE_{u,v} \leq \min\{S_{u,v}, MSE(M, \sigma_{u,v})\} \tag{18}$$

where $S_{u,v}$ is a quantization strategy, and it is function of $\sigma_{u,v}$. It will be discussed in Section 5.

One can think about using linear programming for solving the problem of distributing total MSE among each $MSE_{u,v}$, but this method generates matrices which are not good from the observer's point of view because it does not include subjective criteria and therefore it does not take advantage of image properties. In Section 5 we will provide a simple algorithm to distribute total MSE among the $MSE_{u,v}$ satisfying Eq. (18).

4.4. HVS modeling

Equation (10) is the usual way to include information related to the human visual system7,8,9,10 in MSE. The HVS model is based upon the sensitivity to background illumination level and to spatial frequencies. The aim of including Φ in Eq. (10) is to produce smooth quantization matrices, where total MSE has been well distributed among the 64 coefficients depending on its perceptual relative importance.

There are two problems related to the HVS model: first, Φ is a one-dimensional function, whereas each DCT coefficient has two coordinates, so we need a mapping between coordinates and frequencies. This is accomplished by using the zig-zag sequence defined by the JPEG standard1,2 as $ZZ : [0, 63] \to [0, 7] \times [0, 7]$. Second, Φ is not directly related to quantization error, but to eye sensitivity. Nevertheless, it is fair to think that quantization error will be more visible for those frequencies where eye sensitivity is higher. Therefore, each $MSE_{u,v}$ is inversely proportional to $\Phi(u, v)$, that is,

$$MSE_{u,v} = \frac{MSE}{\Phi(u, v)}. \tag{19}$$

We propose a model based on the DePalma and Lowry model 9, but we use a different constants set, as shown in Eq. (20). The function f maps each coefficient index u, v to a single frequency, as shown in Eq. (21). We use the zig-zag sequence and a linear transformation in order to adapt it to a desired range, where f_{max} is the maximum desired value for $f(u, v)$ (in our experiments, we have used $f_{max} = 20$). Our HVS model is

$$\Phi[f(u, v)] = [0.9 + 0.18 f(u, v)] e^{-0.12 f(u, v)} \tag{20}$$

where

$$f(u, v) = f_{max} \frac{ZZ^{-1}(u, v)}{N^2 - 1}. \tag{21}$$

This model generates a smooth curve, whose variation is smaller than the models previously mentioned. In practice, we are not interested in modeling the real HVS response function, but in distributing a fixed amount smoothly among 64 variables, without great differences among them. We also constrain Φ as follows

$$\sum_{u,v} \frac{1}{\Phi[f(u,v)]} = 64. \tag{22}$$

We will compare the results of our model with two other models. The first one is based on the JPEG default quantization matrix, which is considered to be a good one for medium-high image quality. Basically, this model defines Φ as inversely proportional to $Q_J^{-1}(u,v)$, the elements of the JPEG default quantization matrix, where $f(u,v)$ is again the zig-zag sequence, but no linear transformation is used. The second one is the model defined by Nill 10, which is adapted for DCT. This model assumes radial isotropy, so $f(u,v) = \sqrt{u^2+v^2}$, and a linear transformation is used to adapt the frequency range, in this case $f_{max} = 40$.

4.5. Quantization approach

A progressive transmission scheme may be seen as a sequence of quantization matrices computed from a quantization matrix that would produce the final desired quality using the JPEG sequential operation mode. Therefore, the construction of a definition script consists of computing the quantization matrices for each stage and converting them into band definitions. On the other hand, the evaluation of a definition script consists of computing the quantization matrix equivalent to each stage, then using the method mentioned above to compute the quality of the reconstructed image in such stage. This obviates expensive coding, decoding and error measuring processes. Our goal is to compute the quantization matrix Q^i used in each stage i of the desired progressive definition script.

A coefficient is called *unnecessary* if does not increase image quality when it is sent in a progressive scan. A reason for a coefficient being unnecessary is to be quantized with a large value so the quantized coefficient is always zero. In that case, there is no reason to spend any bits in coding such a coefficient, so it should be detected and removed from any band when possible.

4.5.1. Spectral selection

Spectral selection can be easily studied using a quantization approach. For a given quantization matrix Q, each matrix Q^i is computed as follows. If after band i has been sent, coefficient $I(u,v)$ has not been sent yet, then set $Q^i_{u,v}$ to infinity. Quantizing a coefficient $I(u,v)$ by infinity means $I'_Q(u,v)$ will be always zero, and will exhibit the same behavior as if the coefficient had been not sent. Otherwise, we set $Q^i_{u,v} = Q_{u,v}$.

4.5.2. Full progression

Successive approximation can be seen as a second quantization using a factor that is a power of two, called the point transform. In order to avoid image artifacts that could introduce a large visible distortion, DC and AC coefficients use a different point transform. Since the point transform is defined to be an integer division by 2^{Al}, the inverse point transform is a multiplication by 2^{Al}. This generates a large zero for quantized coefficients, introducing an unacceptable degradation for the DC coefficient. Therefore, the point transform for the DC coefficient is an arithmetic right shift of Al bits, which is the same as using a quantization factor of $Q_{0,0}\,2^{Al}$.

To simulate this large zero for the AC coefficients, we should use a quantization factor of $Q_{u,v}(2^{Al+1}-1)$, but also $Q_{u,v}2^{A}l$. Eq. (13) assumes a constant quantization step Q for a given source, so it is no longer valid for computing its quantization error. Repeating the uniform quantization error analysis for a Laplacian source followed by a point transform of Al bits, we obtain

$$MSE_{AC}(Q_{u,v}, \sigma_{u,v}, Al) = \sigma_{u,v}^2 - \frac{2^{Al}[(2^{Al}-1)Q_{u,v} + \sqrt{2}\sigma_{u,v}]e^{\frac{Q_{u,v}}{\sigma_{u,v}\sqrt{2}}}}{e^{\frac{2^{Al}Q_{u,v}\sqrt{2}}{\sigma_{u,v}}} - 1}, \tag{23}$$

and its corresponding inversion,

$$Q_{u,v} = clip\left(\left\lfloor \sigma_{u,v}\sqrt{2}\tilde{F}^{-1}\left[1 - \frac{MSE_{u,v}}{\sigma_{u,v}^2}, Al\right] + \frac{1}{2}\right\rfloor, 1, M\right), \tag{24}$$

where $\tilde{F}$ is an auxiliary function similar to F in Eq. (14), but including information about Al.

5. QUANTIZATION MATRIX CONSTRUCTION ALGORITHM

Usually, it is better for the user to specify a value for desired PSNR instead of MSE, because it is the most common measure in literature. Computing MSE from a given PSNR is straightforward, that is,

$$MSE = \frac{L^2}{10^{\frac{PSNR}{10}}}, \tag{25}$$

where L is the maximum (or peak) pixel value. For 8-bpp images, $L = 255$.

Basically, the algorithm for computing a quantization matrix Q for a given image and a desired MSE (or PSNR) consists of three steps: first, compute the statistical descriptors for the transformed image to be compressed. Second, distribute the desired total MSE among each $MSE_{u,v}$. Third, compute each $Q_{u,v}$ using the appropriate model.

The first step can be done while image is being transformed during the first stage of the JPEG standard using the DCT. Although the DCT is not really part of the algorithm, it is shown in order to clarify that we are using the transformed coefficients statistical descriptors.

The second step is the most important: MSE has to be distributed among each $MSE_{u,v}$ depending on $\Phi[f(u,v)]$, but the constraints imposed by Eq. (18) have to be satisfied. We call a coefficient *saturated* when it does not satisfy the right-hand inequality in Eq. (18). In our experiments, $S_{u,v} = \sigma_{u,v}^2$, which implies that a saturated coefficient will be always quantized to zero, but a more complex strategy quantization could be used in order to avoid this effect. First, this step solves all saturated coefficients, assigning the maximum possible $MSE_{u,v}$, and then distributing the remaining $MSE_{u,v}$ among all the other coefficients that have not been assigned yet. This is done coefficient by coefficient, starting by the most likely coefficient to be saturated, that is, the last coefficient in the zig-zag sequence. When no more saturated coefficients are found, they are assigned to its computed $MSE_{u,v}$ as well.

Finally, the third step uses Eqs. (12) and (4) to compute the $Q_{u,v}$ quantization factors, which must be rounded to the nearest integer. The complete description of the algorithm is shown below.

Algorithm PSNR2Q

```
compute the 8 × 8 DCT for each block
compute σ_{u,v} and maxMSE_{u,v} using Eq. (18)
compute MSE_obj using Eq. (25)
pns ← 64;  MSE_dist ← 64 MSE_obj;  s ← 64;
saturated(u, v) ← false ∀ u, v
do
   i ← pns;  flag ← false
   do
      u, v ← ZZ(i)   // ZZ is the zig-zag sequence
      if ¬saturated(u, v)
        MSE_{u,v} ← (MSE_dist/s)/Φ[f(u, v)]
        if MSE_{u,v} > maxMSE_{u,v}
          MSE_{u,v} ← maxMSE_{u,v};
          saturated(u, v) ← true;  flag ← true;
          MSE_dist ← MSE_dist − MSE_{u,v};
          s ← s − 1/Φ[f(u,v)]
          if i = pns
            pns ← pns − 1;
          fi
        fi
      fi
      if ¬flag
        i ← i − 1;
      fi
   od until flag ∨ (i < 1)
od until (i < 1)
compute Q_{u,v} using Eqs. (12) and (4)
```

5.1. Algorithm cost

The algorithm presented above is image dependent, that is, it uses the transformed coefficients variances to compute their quantization values. These variances can be computed at the same time as the DCT, increasing the number of operations in only one addition and one multiplication for every AC coefficient in each block. For normal images, the zig-zag sequence yields a good ordering for coefficient variances2, so MSE distribution can be considered almost a linear operation, hence only 64 iterations are needed. Finally, the quantization values are computed. This involves a numerical inversion, but due to the $F(t)$ properties mentioned is Section 2.1, a few iterations (usually less than five) are needed. The method could be speeded-up using interpolation or tabulated values.

This is clearly superior to the scaling method, where the user has to compress the image, then decompress it, and measure the PSNR. The process has to be repeated until the desired PSNR is achieved, which can be difficult due to the lack of relationship between the quality factor and PSNR.

5.2. Simulations

Although we can use Eq. (18) to compute the range of valid PSNR for a given image, this range usually exceeds typical user requirements. Instead of this, we will use the usual range employed in the JPEG standard, which is determined by the quality factor.

5.2.1. JPEG quality factor

Images are compressed with a quantization matrix computed as a scaled version of the JPEG standard default quantization matrix, depending on a quality factor, which can be in range 0–100 (where 0 means a complete black image and 100 means a nearly-lossless image). Nevertheless, the range 25–95 is more appropriate, because a quality factor below 25 generates very low quality images, while a quality factor above 95 generates very large images with no appreciable quality gain. This is the method used by the most widespread JPEG software library, available from the Independent JPEG Group 3 (IJG).

In our experiments we have used the JPEG standard monochrome corpus set. All the images are 720×576 pixels, 8-bpp. This corpus set can be found at `http://ipl.rpi.edu`. When the same quantization table is used for all the images, both PSNR and image size differ so much, because there are fundamental statistical differences between the images. For example, the image called *balloons* is a very smooth image, with almost all energy concentrated in the DCT coefficients occupying the lowest index positions, whereas the image called *barbara2* has the opposite behavior. For example, for a quality factor of 25, image *balloons* yields a compressed image size of 13493 bytes and a PSNR of 38.47 dB, while image *barbara2* yields 30449 bytes and 29.94 dB, respectively. For a quality factor of 95, figures are 85625 bytes and 48.12 dB for *balloons* and 174245 bytes and 43.52 dB for *barbara2*. This confirms the unpredictability of both image size and PSNR for a given quality factor, and it justifies the research for a relationship between PSNR and the JPEG quantization matrix used.

5.2.2. Model accuracy

The first experiment tests the Laplacian model accuracy, as shown in Table 1. It can be seen that the Laplacian model yields accurate results, with an error smaller than 1 dB for low PSNR values and even better for higher PSNR values. Furthermore, the Laplacian model yields better quantization matrices than the classical scaling method due to a better MSE distribution. This first experiment has been done using image *barbara2*, but similar results are obtained when other images from the standard corpus set are used.

5.2.3. Subjective criteria

Quantization matrices computed in the preceding experiment yield a good approximation for the desired PSNR, and a higher PSNR for the same image size than the classical scaling method. However, they cannot be considered good matrices from the observer's perception point of view. This is due to the regular distribution of MSE among the different $MSE_{u,v}$ when $\Phi[f(u,v)] = 1$.

The second experiment shows the algorithm results when different HVS response functions are used for computing the quantization matrix for a desired PSNR. The image is again *barbara2*, and the results are shown in Figure 2. It can be seen that there is no noticeable efficiency loss when the proposed HVS model is used (except maybe for higher PSNR), while the other HVS models yield worse results. Similar results are also obtained for the other images.

As in the previous experiment, quantization matrices computed with our method yield a better PSNR for the same image size than the classical scaling method. These matrices are also good from the observer's point of view, because the total error has been smoothly distributed among the 64 coefficients using an HVS model, and the most important coefficients (in a distortion perception sense) are quantized using smaller values than those coefficients which correspond to frequencies where eye is less sensitive.

6. SCRIPT CONSTRUCTION AND EVALUATION

Our method exploits the quantization approach of the JPEG standard progressive operation mode and some well known facts about subjective perception. First, we will describe a definition script evaluation algorithm, which allows the user to test several definition scripts. Then, we will describe a definition script construction algorithm, giving a simple method for spectral selection technique, including the typical problems that arise in practice.

6.1. Script evaluation

Script evaluation consists of computing the equivalent quantization matrix for each stage of a progressive transmission defined by a given definition script, then using the models previously defined to compute the predicted error. Therefore, the user can try several definition scripts before coding an image without having to encode and decode the image using the JPEG standard to test validity.

The following algorithm describes the definition script evaluation process:

```
Evaluation Algorithm
compute the 8 × 8 DCT for each block
compute σ_{u,v} ∀ u, v ≠ 0, 0
compute Q for the desired final image quality
Q̃ := {∞}
for i := 0 to k // for each band in the def. script
    for j := Ss to Se // for each coefficient in band
        Q̃_{ZZ(j)} := PT(Q_{ZZ(j)}) // update Q̃
    end
    print PSNR(Q̃) // compute PSNR
end
```

The auxiliary function PT applies the corresponding point transform to a coefficient depending on its position in the zig-zag sequence ZZ and the value of Al, as defined in Section 4.5. MSE (and therefore PSNR) are computed using Eqs. (11), (13), (23) and (9).

6.1.1. Example

The evaluation algorithm may be used to compare several definition scripts, in order that the best one (in a practical sense) for a given image can be chosen, without having to encode and decode the image using the JPEG standard progressive operation mode, reducing definition script evaluation cost.

For example, when full progression is used, choices are sending all bit planes consecutively for a given coefficient or band of coefficients (successive approximation alone), or sending all coefficients for a given bit plane (full progression). It is known1 that full progression yields the best results, as shown in Figure 3, where image *barbara2* is segmented using two different definition scripts. Both definition scripts send the same amount of image energy, (quantified as the DC coefficient and the first 9 AC coefficients), which are sent separately, one coefficient per band. Coefficients are split using two bit planes ($Al = 2$), so there is a total of 30 bands. Both definition scripts are not real in a practical sense, but they are useful for showing progressive operation mode properties. It can be seen that full progression is clearly superior to successive approximation, giving the idea that it is better to send more point transformed coefficients than fewer full resolution coefficients.

Notice that evaluation algorithm accuracy is not always very good. This is caused in part by Laplacian assumption for AC coefficients (ignoring the shape factor) and by JPEG constraints on quantization values, which must be integers. Nevertheless, predicted PSNR still follows the same behavior than measured PSNR, so the algorithm may be used as a useful tool for definition script evaluation and comparison.

6.2. Script construction

Suppose the user wants to encode a given image using the progressive operation mode to a desired final quality E_k using $k+1$ bands (we use $k+1$ for notation purposes). We require $E_i < E_{i+1}$ in order that null bands may be avoided. Bands can be described using the desired quality for the reconstructed image after each stage is decoded, for example. Another variable could be the number of bits spent in each band, but the JPEG progressive operation mode is too complicated to achieve this goal.

Basically, a given image is segmented in 8×8 blocks and then each block is transformed using the DCT. Both steps are part of all JPEG standard lossy operation modes, so we are not increasing JPEG computational cost here. While the DCT is being computed for each block, we also compute the AC coefficients variances needed by our method. Then, using these variances and the sequence of image qualities $\{E_i\}$, we compute the quantization matrices needed to simulate the progressive scheme and the corresponding descriptions in the definition script. We also compute the maximum and minimum achievable qualities using $Q = 1$ and $Q = \infty$ (actually $Q = 255$ for 8 bit quantization tables) respectively. The reason is to ensure user requirements are feasible for the given image and qualities sequence, that is, $PSNR_{MIN} \leq E_i \leq PSNR_{MAX}\ \forall\, i = \{0, \ldots, k\}$.

6.3. Spectral selection

In this case the initial quality E_0 is only determined by the DC coefficient computed for E_k, the final quality, because the DC coefficient must be sent separately in a first band, using the same quantization value computed for the last quality E_k. Basically, the idea is to compute Q^k given E_k, and then start with Q^0, constructed as

$$Q^0_{u,v} = \begin{cases} Q^k_{0,0} & u = 0, v = 0 \text{ (first scan)} \\ \infty & \text{otherwise.} \end{cases} \tag{26}$$

Each stage i consists in copying the previous quantization matrix, that is, $Q^i = Q^{i-1}$, then adding as many quantization factors from Q^k as needed to achieve the desired quality E_i using the zig-zag sequence. Therefore, the quantization matrix Q^i at each stage may be computed as

$$Q^i_{u,v} = \begin{cases} Q^k_{u,v} & u, v \text{ are sent in scan } i \\ Q^{i-1}_{u,v} & \text{otherwise.} \end{cases} \tag{27}$$

In order to improve algorithm efficiency, we detect and avoid unnecessary coefficients at the beginning of each band of coefficients. We could also detect unnecessary coefficients within a band and then split such band in two bands, but this will increase the number of bands, which might be not desirable because we want to maintain the sequence of image qualities defined by the user.

The following algorithm describes the definition script construction process:

Construction Algorithm for Spectral Selection

```
compute the 8 × 8 DCT for each block
compute σ_{u,v} ∀ u, v ≠ 0, 0
compute PSNR_min and PSNR_max
compute Q^k and Q^0
generate (0,0,0,0) // (Ss,Se,Ah,Al) for DC coeff.
last := zz := 1
for i := 1 to k
    if PSNR_min ≤ E_i ≤ PSNR_max // valid stages
      Q^i := Q^{i-1}
      while zz is unnecessary do
            last := zz := zz + 1
      od
      do
         Q^i_{ZZ(zz)} := Q^k_{ZZ(zz)}
         compute E for Q^i
         zz := zz + 1
      od until E ≥ (E_i − ϵ)
      generate (last,zz − 1,0,0) // (Ss,Se,Ah,Al)
      last := zz
    fi
end
```

This algorithm generates a valid definition script using spectral selection for a given image and a valid sequence of image qualities. Method precision can be adjusted using ϵ, which may be set to $\epsilon = 0.25$, for example, in order to assume a small error produced by matrix computation.

This algorithm takes advantage of spectral selection technique properties previously mentioned: first, coefficients are sorted using the zig-zag sequence, where coefficients retaining more block energy are sent in first position, so it is useless to change band order. Second, unnecessary coefficients at the beginning of each band are not sent, saving some bits in the coding process without reducing image quality. Finally, not all coefficients are sent, but only those required to achieve the desired final image quality. The main idea is that the zig-zag sequence sorts coefficients almost optimally (in a practical sense) for a progressive transmission.

6.3.1. Example

We will use the algorithm described above to generate a simple definition script. Our goal is to illustrate the typical problems that appear when nothing is known about the image before the coding process starts.

In this example, we would like to compress a given image using five stages, with a sequence of image qualities of 25, 30, 35, 40 and 45 dB respectively. This sequence of image qualities may seem arbitrary, but it will be useful to show progressive operation mode problems. We will use two different images of the JPEG standard monochrome corpus set to show the differences between the generated definition scripts.

For image *balloons*, it is impossible to obtain an initial quality of 25 dB, as shown in Table 2, because sending only the DC coefficient in the first band (as the JPEG standard requires) already produces an image with higher quality. We will try to solve this problem using full progression and segmenting the DC coefficient in several bit planes later in this paper. On the other hand, the other image qualities are correctly achieved. Nevertheless, model precision is limited because the final image quality (45 dB) is high for this image. In contrast, most quantization factors are small, so rounding errors caused by integer quantization values and other JPEG constraints on quantization factors are more noticeable. For example, the DC coefficient should be quantized using a quantization factor smaller than one, which is impossible.

Notice that in this case all 64 coefficients are sent, so the final image needs only 44636 bytes, instead of 47877 bytes needed for the equivalent image encoded using the sequential operation mode with Hufmann table optimization (progressive operation mode also forces optimization). This is one of the most remarkable properties of the JPEG standard progressive operation mode, as it does not increase file size when compared to sequential operation mode when the number of bands is relatively small.

Repeating the same experiment using image *barbara2* yields very different results, as shown in Table 3. The reason is that this image has a lot of detail and most of image energy is concentrated in high frequencies, unlike image *balloons* which is very smooth, where image energy is concentrated in coefficients occupying low frequencies.

The first stage has a PSNR lower than the first specified quality, E_0. Therefore, there is an extra stage needed to achieve 25 dB so a total of six bands instead of five are generated. Small PSNR differences are also caused by the impossibility of using non-integer quantization values. Regarding final image size, notice that in this definition script all 64 coefficients are sent, but there is still a small decrease in file size, as sequential operation mode would need 156838 bytes. Both examples show that it is very interesting to compute $PSNR_{min}$, $PSNR_{max}$ and the real

E_0, in order that the user might redefine the sequence of image qualities according to image statistics and thus to obtain better results.

7. CONCLUSIONS

In this paper we have presented a method for computing JPEG quantization matrices for a given PSNR. Computed matrices are not optimal in a rate distortion theory sense, but they are better than the matrices computed using the classical scaling method, at a reduced cost. Moreover, computed matrices generate compressed images that are visually comparable to those generated by the classical scaling method, with no visual artifacts. The method is useful because it solves a common problem of JPEG users, that is, the trial and error procedure as the unique tool for computing quantization matrices.

Computed matrices can also be a good initial solution for an iterative algorithm such as the one proposed by Fung and Parker 13. Moreover, we also propose to substitute the MSE measure for Eqs. (3) and (11), so the whole process could be speeded-up with almost no loss of effectiveness. Although the method described in this paper is for monochrome images, it can also be used for color images when they use a image color model which can be split into luminance and chrominance channels 14, such as the YC_bC_r color model used in the JPEG standard. The method has to be applied to each channel separately.

We have also described a method for JPEG progressive operation mode script construction and evaluation, using a quantization based approach. The JPEG standard progressive operation mode may be seen as a second quantization applied to DCT coefficients that have been already quantized. This allows us to establish a relationship between each scan of a definition script and its equivalent quantization matrix.

Our method allows the user to compute a valid definition script for a given image and a sequence of image qualities. Generated scripts take advantage of DCT and zig-zag sequence properties, avoiding unnecessary coefficients that do not increase image quality. Due to the JPEG standard progressive operation mode properties, compressed images are usually smaller than if they were compressed using the sequential operation mode. Although the accuracy of evaluation and construction algorithms is limited, they are perfectly valid in a practical sense, because predicted PSNR follows the real PSNR behavior. Our method reduces the cost of constructing and evaluating definitions scripts because it allows the user to avoid the classical trial and error procedure.

Further research in this topic should include a study of the JPEG coding method based on quantized coefficient entropy, in order to have an optimal method in the sense of rate distortion theory. More complex quantization strategies are also needed to avoid complete quantization of high frequency band coefficients. Extension to color images is also an interesting subject, and it should include the search of error measures in DCT transformed color spaces and visual criteria for band construction and evaluation.

Acknowledgements

This work is partially supported by Spanish Government grants 1999 XT 00016 and ITT CTP 99/12. The authors would like to thank the reviewers for their suggestions and insights.

REFERENCES

1. G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM* **ACM-34**, pp. 30–44, Apr. 1991.
2. W. B. Pennebaker and J. L. Mitchell, *JPEG still image data compression standard*, Van Nostrand Reinhold, 1993.
3. T. G. Lane *et al.*, "Independent JPEG Group library v6.1a," 1997. `http://www.ijg.org`.
4. J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory* **6**, pp. 7–12, Mar. 1960.
5. R. C. Wood, "On optimum quantization," *IEEE Transactions on Information Theory* **15**, pp. 248–252, Mar. 1969.
6. K. Rao and P. Yip, *Discrete Cosine Transform. Algorithms, Advantages, Applications*, Academic Press, 1990.
7. H. Marmolin, "Subjective MSE measures," *IEEE Transaction on Systems, Man, and Cybernetics* **SMC-16**, pp. 486–489, June 1986.
8. A. B. Watson, ed., *Digital Images and Human Vision*, The MIT Press, 1993.
9. J. DePalma and E. Lowry, "Sine wave response of the visual system. ii. sine wave and square wave contrast sensitivity," *Journal of the Optical Society of America* **56**, pp. 328–335, Mar. 1962.
10. N. B. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Transactions on Communications* **COM-33**, pp. 551–557, June 1985.
11. D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Representations*, John Wiley & Sons, 1985.
12. R. C. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Transactions on Communications* **COM-31**, pp. 835–839, June 1983.
13. H. T. Fung and K. J. Parker, "Design of image-adaptive quantization tables for JPEG," *Journal of Electronic Imaging* **JEI-4**, pp. 144–150, Apr. 1995.
14. N. M. Moroney and M. D. Fairchild, "Color space selection for JPEG image compression," *Journal of Electronic Imaging* **JEI-4**, pp. 373–381, Oct. 1995.

Julià Minguillón received the BS degree in Computer Science in 1995 and the MS degree in 1997 from the Autonomous University of Barcelona, Spain. Since 1994 he has been with the Combinatorics and Digital Communication Group at the Autonomous University of Barcelona. His research interests are in information theory, pattern recognition and image compression. He is also an Independent JPEG contributor.


Jaume Pujol received the BS degree in Mathematics in 1979 and the BS degree in Computer Science in 1989 from the Autonomous University of Barcelona, Spain. He received the MS and PhD degrees in 1991 and 1995 respectively. Since 1988 he has been with the Combinatorics and Digital Communication Group at the Autonomous University of Barcelona. His research interests are in information theory, error correcting codes, pattern recognition and image compression. Dr. Pujol is a senior member of the IEEE.

TABLES

Table 1. Predicted and real model accuracies (PSNR in dB.)

pred.	real	pred.	real	pred.	real
29	30.06	36	36.84	43	43.41
30	31.08	37	37.78	44	43.74
31	32.04	38	38.75	45	45.13
32	33.10	39	39.61	46	46.40
33	34.05	40	40.45	47	47.06
34	35.02	41	41.37	48	47.06
35	35.91	42	42.32	49	49.51

Table 2. Construction algorithm results for image *balloons*.

k	*Ss*	*Se*	*Ah*	*Al*	computed PSNR	real PSNR	image size
0	0	0	0	0	27.59	29.67	7234
1	1	1	0	0	30.51	31.64	11020
2	2	5	0	0	35.88	35.88	21622
3	6	12	0	0	40.24	40.26	31385
4	13	38	0	0	45.00	45.32	44636

Table 3. Construction algorithm results for image *barbara2*.

k	*Ss*	*Se*	*Ah*	*Al*	computed PSNR	real PSNR	file size
0	0	0	0	0	21.66	22.08	7611
1	1	6	0	0	25.06	25.06	31146
2	7	22	0	0	29.92	29.92	73603
3	23	39	0	0	35.02	35.01	109443
4	40	52	0	0	40.31	40.71	134371
5	53	63	0	0	44.00	44.94	152031

FIGURES

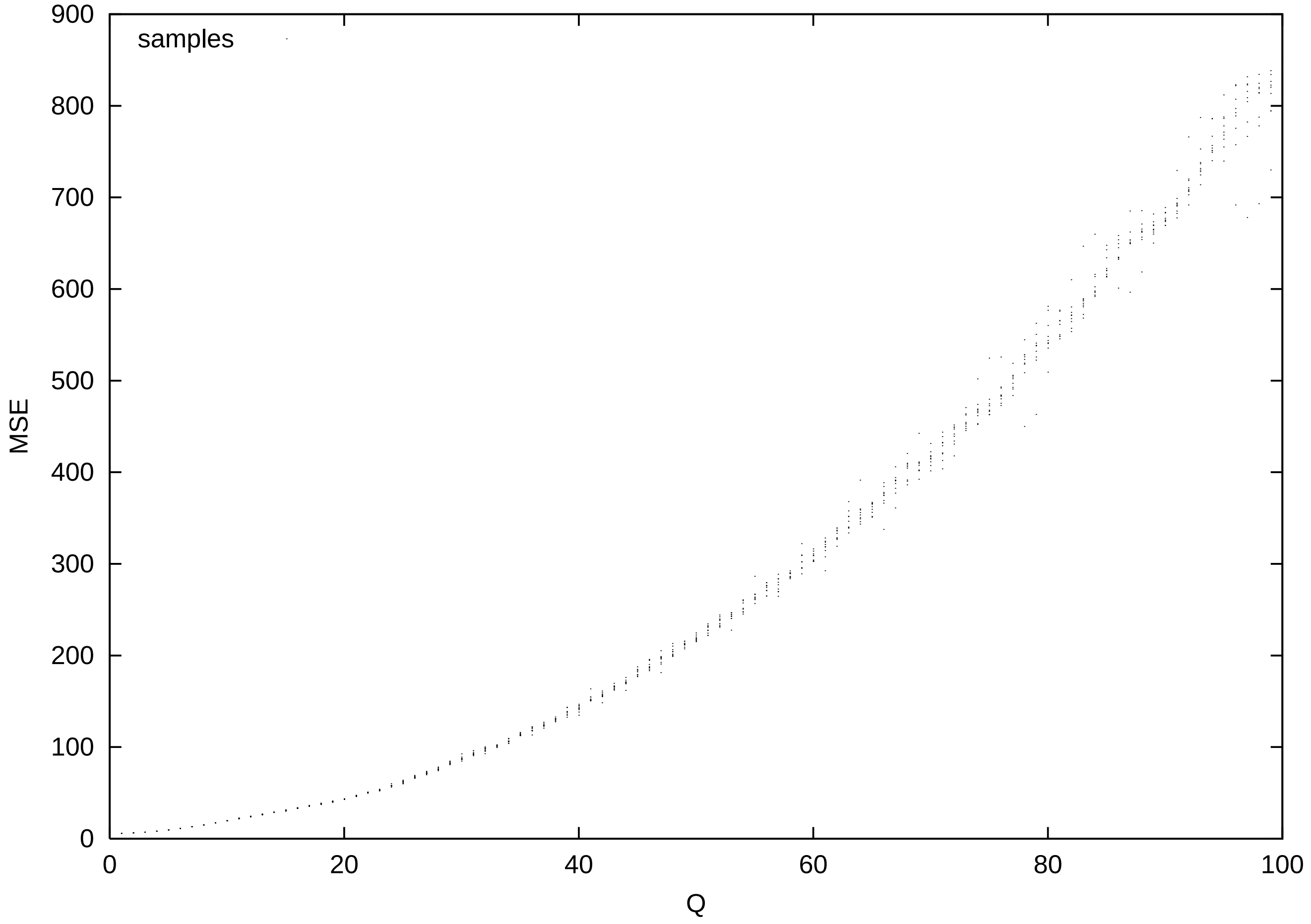

Figure 1. DC quantization error.

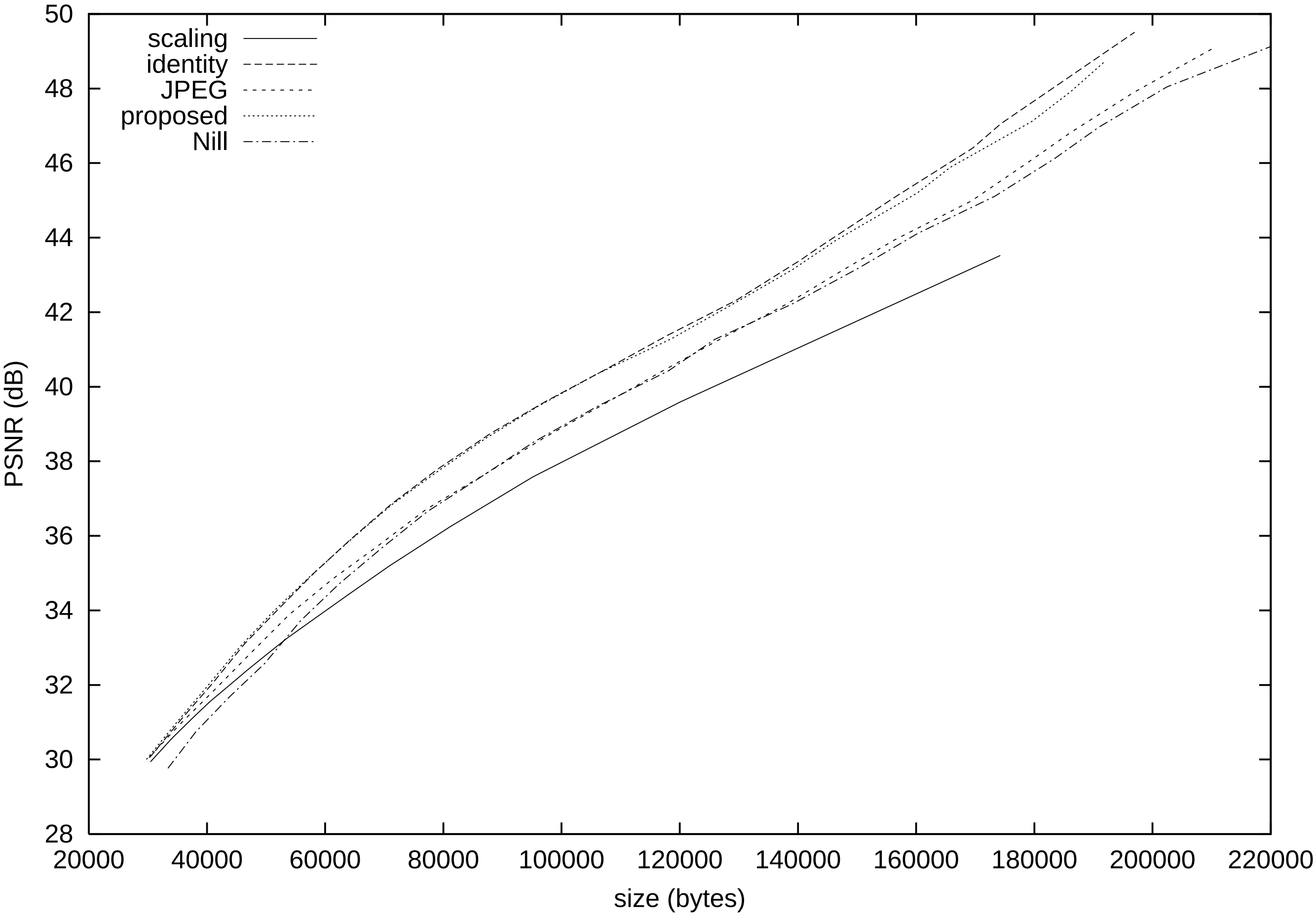

Figure 2. PSNR / image size for different Φ.

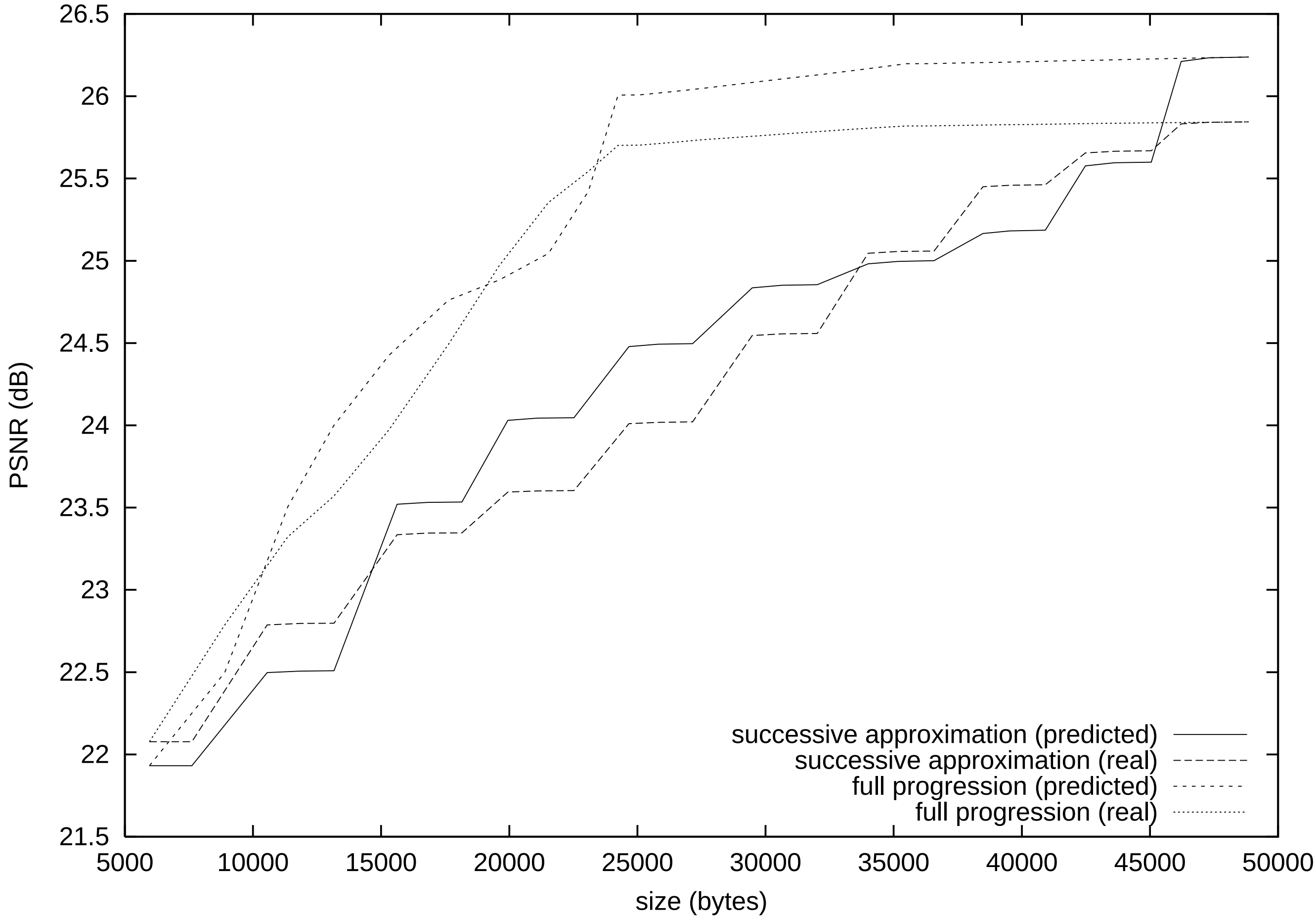

Figure 3. Example of evaluation algorithm results for image *barbara2* using two definition scripts which send the same amount of image energy, showing importance of bit plane ordering.