A METHOD FOR RATE CONTROL AND COMPRESSION ESTIMATION IN JPEG

Ricardo de Queiroz and Reiner Eschbach
Xerox Corporation
800 Philips Rd. M/S 128-27E
Webster NY 14580
E-mail: queiroz@ieee.org, reschbach@crt.xerox.com

Abstract. *A method for approximate rate control in JPEG is presented. We derive a quantizer table that would attain a given compression ratio for a given image. For that, we devise a method to quickly estimate the compression ratio that a given "quality" factor (a control knob for generating quantizer tables) would yield. This "quality" factor controls the design of the quantizer tables, so that we developed a new quantizer design strategy, which is a key enabler for the compression estimation procedure. It is based on the human visual system transfer function and provides more flexibility and better compression than typical scaling-based design methods. Overall, we calculate a simple parameter derived from the image data. This parameter in combination with the quality factor is used to estimate the compression ratio. If the ratio does not meet the target, the quality factor is iterated until yielding the desired compression. Extensions to color images are presented and results are shown to evaluate the method's accuracy.*

1) BACKGROUND

The Joint Photographic Experts Group (JPEG) standard [1] for image compression is commonplace in most imaging applications. In JPEG, the image is broken into 8x8-pixel blocks, which are transformed using the discrete cosine transform (DCT) [2]. A quantizer table in JPEG is a set of 64 integers by which the 8x8 DCT coefficients are divided in order to decrease data entropy and enable compression. The entries in a quantizer table are also called quantizer steps. JPEG recommends using different quantizer tables for luminance and chrominance channels of color images. The choice of quantizer tables determines the bit-rate (R) achieved by compression and the distortion (D) obtained after decompressing the image. Hence, each choice of quantizer table set defines a point in the RD plane for compressing a given image. To accommodate different quantizer tables, the most popular programs utilize a single quantizer table (often the example quantizer table provided in the JPEG specifications) as a reference point, and multiply the table entries by a fixed scaling number [3]. The single scaling number can control (parameterize) the compression, hence yielding a curve in the RD plane. In general, the multiplying number is mapped from a scale of *quality factors Q* ranging from 1 to 100 to a scaling number typically ranging from 0.1 to 10. Some companies utilize their own base quantizer table and scale the numbers in a linear fashion from "more" to "less" compression. There are also solutions based on optimizing the quantizer table for a given image in a sense of minimizing bit-rate for a given distortion measure (typically MSE) [4]. Some quantizer tables are designed based on the human visual system (HVS) by means of a modulation transfer function (MTF). For these, table offset and multiplying factor are often chosen ad-hoc in order to conform the table entries to their valid range and to control the compression amount.

A common problem in JPEG compression is the lack of bit rate control, i.e. the final file size is unknown until the image is actually compressed. This may cause problems in several applications, for example if a fixed buffer size is to be used to store the compressed image. Basically, the problem is to know how much compression a given setting yields or, conversely, which settings would yield a given compression ratio. In [4], the direct design of the quantizer table via RD-based search will not determine the table but also attain a specific rate. Even though the method in [4] may yield the best results, it is very complex for most real time applications. A more practical method to attempt rate control in JPEG was proposed in [5] in which the is only one control parameter *Q* and a few blocks are sampled from the image. Q is then iterated until getting the desired rate (or distortion) on the desired set.

Figure 1 depicts a typical correspondence between quality factors and actual compression ratios for a given image. The different compression ratios were obtained by scaling the example JPEG quantizer table. Throughout this paper, we will compare our results with the results for the example quantizer table, acknowledging that better quantizer tables with respect to image quality have been derived elsewhere. The emphasis of this paper is not on image quality, but on predicting compression ratios from simple metrics. Any quantizer table that has properties similar to the properties we derive for the quantizer table can be substituted. Note how non-linear the curve in Fig. 1 is. For different images, different curves would be produced. Hence, it is impractical to attempt to predict the compression ratio as a function of the quality factor. In order to predict compression, we first devise an efficient method to design JPEG quantizer tables based on a single *Q* factor, where the main attribute of the quantizer table is a more linear relationship between compression ratio and quality factor used in scaling. Using that quantizer table, we develop a method to quickly predict the compression ratio for a given image and *Q* factor using only simple computations. With these tools in hand we are able to quickly generate image-dependent JPEG quantizer tables, i.e.: the base table and quality factor combination, aimed at automatically yielding the desired compression ratio.

2) A PARAMETRIC QUANTIZER TABLE DESIGN

Fig. 1 shows a very non-linear correspondence between the quality factor and the resulting compression ratio for the JPEG example quantization table. In order to facilitate the prediction of the output compression it would be helpful to design a quantizer table and scaling factor combination that yields a near-linear relationship. In that case, the relation between quality and compression could characterized by perhaps one single parameter, e.g.: the curve slope. In this case, given the curve slope and quality factor, one could easily predict compression, or, conversely, determine the scaling factor needed for a desired compression ratio. The design method described here is just one of the possible design methods that can be used and was used for simplicity of implementation.

The design method is based on the assumption that a near-linear relationship between the quantizer table and the compression ratio can be obtained if the quantizer table simulates the observation of an image from different viewing distances. This design can be enhanced to take into account other properties such as the HVS. Quantizer design based on visual properties has been explored before by several researchers, e.g. [6]-[12]. In most cases the quantizer table is "shaped" according to the HVS MTF or it incorporates visibility thresholds. We utilize a method to design quantizer tables which basically combines the scaling method with the HVS-based fixed approach. This is so because the MTF may determine the "shape" of the quantizer, i.e. the ratio among quantizer steps, but it does not control the "gain", i.e. the absolute scaling to multiply the quantizer steps and to control the compression amount.

We assume resolution and image width are fixed. What is varied is the scaling factor and a parameter given by:

$$\alpha = \frac{d}{w} = \frac{\text{viewer distance}}{\text{image width}}$$

A single control number is mapped simultaneously to a multiplying factor and to the desired α, so that the user just needs to adjust one "knob". The concept is that if one wants more compression we emulate someone looking at an image from further away and use a more aggressive scaling. The emulation of very close range viewing conditions along with small scaling are used to design a table for low compression ratios.

A simple HVS MTF was assumed. For luminance, the formula we used is given by [2]

$$H(f) = 2.46 \cdot \left(0.1 + 0.25 f\right) \cdot e^{-0.25 f}, \tag{1}$$

while, for chrominance, the formula we used is

$$H(f) = e^{-0.2213 f}, \tag{2}$$

which are plotted in Fig. 2. We also need to specify the maximum visible frequency (f_{max}) in order to facilitate sampling. The viewer is at a distance d from the image (see Fig. 3) which has width w. The image has resolution of r pixels per length unit. Hence, the maximum viewing frequency is

$$f_{\max} = \frac{r\, w\pi}{4\ \ 180\ \ \tan^{-1}\left(\dfrac{w}{2d}\right)} = \frac{NK}{\tan^{-1}\left(\dfrac{1}{2\alpha}\right)}, \tag{3}$$

where N is the number of pixels across the image and $K = \pi/720$.

The MTF functions in eqs. (1) and (2) can now be used to calculate the relative weight of the individual DCT frequency bands. This relative weight will then be used to determine the quantizer step size, quantizing less important bands more coarsely than more important ones.

Let $g_k(n)$ be the entries of the basis functions of the DCT (k=0,...,7; n=0,...,7), whose Fourier transform is given by $G_k\left(e^{j\omega}\right)$. We can compute the variance of a white noise filtered by the HVS MTF in the DCT domain as a measure of the importance (variance) of each DCT subband for the HVS. Hence,

$$\sigma^2{}_{mn} = \frac{1}{\pi^2}\int_0^{\pi} \left|G_m\left(e^{j\omega_1}\right)\right|^2 \left|G_n\left(e^{j\omega_2}\right)\right|^2 H^2\left(e^{j2 f_{\max}\sqrt{\omega_1{}^2+\omega_2{}^2}}\right) d\omega_1 d\omega_2 .$$

Tests have shown that the above integration can be performed using 64-point FFT's (32 point positive frequencies) with sufficiently accurate results. We then force the step values to be inversely proportional to the standard deviation of this so-called shaped noise in the DCT domain. The more visually important the smaller the step sizes, hence, the lower the distortion. The HVS weights are chosen as:

$$\mu_{mn} = \frac{A}{\sigma_{mn}} \tag{4}$$

where A is a constant to scale to normalize μ_{mn} in the range of 1 to 255.

The user is requested to input a number Q from a minimum to a maximum value, which, without loss of generality, can be chosen to be in the range of 0 to 100. We fix N and derive monotonic functions $\alpha(Q)$ for the MTF shape and $A(Q)$ for the quantizer steps, such that the necessary parameters (A,α) to design the table are directly derived from one input quality factor Q. The scaling is done as

$$\mu_{mn} = \frac{A(Q)\max\left(\sigma_{mn}\right)}{\sigma_{mn}} . \tag{5}$$

Note that the smallest entry in the table is μ_{mn}=A(Q) , i.e. A(Q) determines the minimum step value.

The parameters were fine tuned with two goals in mind. First, as we mentioned in the introduction, we would like to have the relation between compression and Q to be as linear as possible. Second, to facilitate comparisons, we would like to closely align our quality factor with the existing one in the JPEG implementation of the Independent JPEG Group (IJG) [3] (which uses the scaled default tables). This alignment has no impact on performance. If the user requests quality factor Q, we would like to achieve comparable compression ratios for some end-points using both methods.

The curve $\alpha(Q)$ was selected as a straight line connecting extreme points $\alpha_{max}=\alpha(0)$ and $\alpha_{min}=\alpha(100)$. The curve $A(Q)$ was selected as a 3-piece combination of straight lines connecting the following points in the (A,Q) corresponding to the values of $Q=\{0,b_1,b_2,100\}$, where two breakpoints were inserted in order to better approximate the RD curves in IJG's implementation. Examples of parameters are b_1 = 10 and b_2 = 90 and:

Luminance	**Chrominance**
α_{max} = 2.2	α_{max} = 3.4
α_{min} = 0.125	α_{min} = 0.25
A(0) = 50	A(0) = 65
A(10) = 24	A(10) = 30
A(90) = 6	A(90) = 7.5
A(100) = 0.7	A(100) = 0.4

Comparisons between the proposed method and IJG's method are shown in Figure 4. RD performance is compared in the bottom-right plot. Note that the proposed method of jointly modifying quantizer table shape and gain easily outperforms the scaling method. Figure 5 shows enlarged portions monochrome image Lena compressed at about 25:1 by adjusting the quality factor using both the scaling method and the proposed one. It should be kept in mind, however, that the non-linear relationship between compression ratio and quality factor was the main reason for the design of the new quantizer scheme.

Figure 6 shows the relationship between compression ratio and the standard quality factor as well as the new ”visual” quality factor for 34 monochrome images. From Fig. 6 it is easy to note that the new design method factor enjoys a nearly linear relation between the quality parameter Q and the resulting compression ratio. This linearity is a key enabler for predicting the compression performance. The only obvious deviation in the 34 images tested was obtained for a uniform gray wedge smoothly varying from 0 to 255 gray value, which is clearly a pathological case. The curve for the gray wedge can be easily identified as being almost horizontal with a sharp drop at a Q factor of roughly 90.

3) PREDICTING COMPRESSION

Since JPEG has no rate control, the output buffer size is undetermined, which might cause problems in some applications. It is important to know what is the compression ratio for a given quality factor, i.e. how big the compressed file will actually be. Conversely, we also want to know what quality factor should we use to yield a given compression ratio.

If we answer the first question we can answer the second. If the only information we possess is the quality factor, inspection of Fig. 6 reveals that the spread of compression ratios for a given quality factor is very large. So, uncertainty is high, but the estimate comes at no cost. On the other hand, if we are allowed to actually compress the image and count the output bits, the uncertainty is null, but there is a sizeable computational cost involved [4],[5]. We propose a compromise approach in which we allow very simple measurements of image characteristics. Based on these simple metrics and the quality factor we are trying to predict the compression ratio. It is obvious that a "perfect" prediction can be made if the metric is actually achieved by compressing the image, however, we will intentionally limit the image metrics to be simple with respect to the complexity of DCT transformations. Since our quantizer design (Sec. 2) produces near-linear quality vs. compression relations, we need to only estimate the slope of the curve for prediction. The slope of the curve reflects the compressibility of the image and it is not unreasonable to assume that this is correlated to the image activity. However, image activity is not a well defined concept and several image attributes might be useful in determining the activity. In order to get the best advantages we would rather keep the computation as small as possible. This leads to the desire to use a very simple metric, one of which might be local dynamic range.

3.1) Estimator model

Since JPG uses an underlying 8x8 block structure in compression, we will consider the dynamic range $\boldsymbol{r_n}$ of the *n*-th 8x8 block in the image as local dynamic range, i.e. if X_n is the set of pixels values in a given block, then $\boldsymbol{r_n} = \max(X_n) - \min(X_n)$. Let $h(k)$ $(0 \le k \le 255)$ be the histogram of the $\boldsymbol{r_n}$, and let the cumulative distribution of the $\boldsymbol{r_n}$ be $d(m) = \sum_{k=0,\ldots,m} h(k)$. The top row of Fig. 7 shows the plots of the cumulative distribution *d(m)* for the 34 images, along with the highlighted cumulative distribution for three representative images.The bottom row of Fig. 7 shows the compression ratio (CR) vs. quality (Q) factor curves (repeated from Fig. 6 with the three images highlighted). The highlighted images in Fig. 7 are illustrative of one prevailing feature of the *d(m)* sequences: the CR curves correlate with the *d(m)* ones in the following way: whenever *d(m)* rises up quickly, higher compression ratios are achieved for the identical quantizer tables. Since the CR plots are approximately linear for the visual quality factor it should be sufficient to derive one measure from the *d(m)* plots to describe “how fast” its plot rises. Several attributes of the cumulative distribution can be used to determine the quickness or steepness of the rise. In order to be less sensitive to noise, we chose the integral (summation) of the *d(m)* of the image as metric. Hence, the measurement we retrieve from an image is described as: compute dynamic ranges $\boldsymbol{r_n}$ and their histogram *h(k)*; then compute

$$S = \sum_{k=0}^{255} \sum_{m=0}^{k} h(m) . \tag{6}$$

In a next step, the mapping between the output compression ratio and the input quality factor Q and metric S is derived. First, for each quality factor, the function CR = $W(Q, S)$ was modeled. The model chosen based on heuristics was the average of the best least-squares fit of quadratic and linear polynomials. The average is also a quadratic curve, even though not the best least-squares fit. A total of 34 images of different characteristics were used for the fit. Figure 8 shows an example of data fitting for Q=30, where the linear, quadratic and average models are shown along with the real data.

The final mapping can be accomplish via a 2D table $W(Q, S)$. Note that S typically ranges from 140 to 240 and we are mainly interested in quality factors between 10 and 90, so that we can build a 90x100 LUT to implement $W(Q, S)$. If we plot said LUT as a surface it will look like in Fig. 9.

Using the simple 2D lookup-table approach, we verified the predicted compression ratio with the actual compression ratio. The average prediction error was less than 12% for the original 34 input images which is a marked improvement over the blind (a priori) form of prediction which yielded an average error greater than 37%. We will discuss the blind estimator and the experimental data in Sec. 3.3.

3.2) From monochrome to color

The parameter S and the mapping $W(f,S)$ are reasonably good estimators for the compression of monochrome images. The data works as well for estimating the compression of the luminance component of color images. New data needs to be collected for chrominance components, but, apart from the actual $W(Q,S)$ mapping, the same assumptions should hold and the base model should be identical. However, since we are looking for a simple method to predict the final compression ratio, we might be able to use the correlation between chrominance components and luminance component for common images. It is clear that this restricts the application to images that viewers might consider "natural" and that it excludes less "natural" images such as color sweeps.

If we are allowed the extra error margin of using the chrominance/luminance correlation we might be able to further simplify the method for a small decrease in precision.. In order to estimate the compression correlation for chrominance and luminance, we computed the typical file size proportion between monochrome and color versions of the same image, i.e. if the monochrome version (luminance channel) of an image compresses n:1, what is the typical compression achieved by compressing the color image? The question obviously has no precise answer, but it is interesting to see what the error margins are. Figure 10 shows the plots of the relative compression ratios among color and monochrome versions of the same image. From those plots, we have two main observations: (i) the relative ratio of compression ratios has some limited variance; (ii) the relative ratio increases with quality factor. From these plots we devised a preliminary estimator for the color compression based on the compression of the luminance channel only. By computing the average and the standard deviation of the plots in Fig. 10, the preliminary estimator is given by:

CR (color) = 2.5 CR(mono) ± 6% (with chrominance subsampling)
CR(color) = 2 CR(mono) ± 12% (without chrominance subsampling)

We computed the percentage variation (6% and 12%) for the range of Q between 10 to 90. But note that the adjustment also varies with Q. Hence, a better a priori estimator for CR(color) can be made as a correction to the monochrome case, i.e.

$$\text{CR(color)} = C \ \text{CR(mono)} \quad ,$$

where C is a correction factor. C is a function of the quality factor Q. Fine tuning C using the average values and standard deviations of the curves for various values of Q in Fig. 10, we devise the correction factor as:

$$C = 0.00375\, Q + 2.3125 \qquad \text{(if there is chrominance subsampling)} \ ,$$
$$C = 0.00625\, Q + 1.6875 \qquad \text{(without chrominance subsampling)} \ .$$

We are only using the luminance to predict the compression of color images. Hence, the method is naturally prone to errors and it is only meant for the average case.

3.3) Experiments: estimation errors

As a reference we can use an unbiased “a posteriori” estimator to see how well the proposed one performs. For a set of images, for a given Q, the unbiased estimator would average all compression ratios in the set. For a set of size 1, there is no estimation error, and there is less error the more uniform the set is. Note that this estimation in impractical since it uses the real CR to make its own estimate. Nevertheless it is a useful reference. In this paper, the CR estimation error is computed as:

$$100 \times \left| \frac{Estimated\_CR - Actual\_CR}{Actual\_CR} \right|$$

In Fig. 11 it is shown the histogram of the proposed (left) and unbiased (right) estimation methods for a number of images and quality factors. We used 38 images and 5 different *Q* factors in Fig. 11. The average estimation error exceeds 37% for the unbiased estimator, while the proposed one brings this number below 12%, which is a marked improvement.

The results in Fig. 11 are for monochrome images. In Fig. 12, we show results for color images, extracted from an exclusively pictorial set. It consists of 3 Corel CDs yielding a total of 302 images. Since 5 quality factors were tested for each image, the total sample count for the pictorial set is 1510. Average errors for the unbiased estimators are in the order of 27% and 24% (with and without subsampling, respectively), while the proposed estimator brings the error down to 17%. This is also a marked improvement, mainly if we take into consideration that the extension to color is done completely blindly.

It is evident from the data in Fig. 11 the large improvement in accuracy. The estimation of the compression of monochrome images is much more precise. Much of the imprecision in Fig. 12 comes from the image-independent linear formula given in Sec. 3.2 to estimate the compression of color images using only the luminance channel.

3.4) Controlling the compression: setting the right quality factor

While the preceding Sections discussed how to estimate the CR for a given *Q*, the inverse question, i.e.: which Q will yield a specific compression ratio, is more interesting in most application. For example, knowing the required Q for a given compression ratio would allow rate control for buffer size or transmission. The problem is to derive the setting that would make JPEG compress the image to a given file size. In other words, users would set the CR and leave up to the program to figure out what the settings should be. With the proposed method we can have a quick approximative solution for the problem . Since *W* is a smooth and well-behaved function of *Q* and *S*, there is another method to directly estimate *Q*. For that we just need to invert the plot in Fig. 9 and actually use $Q=V(\text{CR},S)$, where *V* is a surface derived from *W*. *V* can be computed once and implemented as a 2D LUT. For color, the factor *C* has to be already built into *V*. We first derive *W*, then include *C* into *W*, and last we derive *V* from *W*.

As an alternative we can simply iterate the CR estimator. Note that the computational cost in this method is the computation of *S*. Once that is found, every CR estimate takes only one table look-up (and perhaps 4 extra operations for color). This is negligible compared to the efforts of deriving *f* or of compressing the image. Thus, one can easily iterate *Q* until finding the right CR. One possible method is bisection or the brute force to find the factor *Q* that yields CR closest to the desired target. .

4) Conclusions

We proposed a method to map quality factors to predicted compression ratios. The method includes the on-line design of efficient quantizer tables and the estimation of how much compression would the tables yield. The compression estimator depends only on a simple feature derived from the image, which is related to the histogram of the dynamic range of the image blocks. Once this number is generated, it can be used with any quality factor to estimate the compression. Hence, one or multiple estimations make very little difference in computation. It was shown how to iterate the quality factor to achieve the desired compression so that the method can be used for rate control in JPEG.

It is evident from the plots in Figs 11 and 11 the improvement in estimation precision. However, it needs to be emphasized that deriving *S* alone is more than 10 times faster than compressing the image. Furthermore, using only the luminance, the method is even faster. There is no need to buffer the image (although requires one full pass to determine *S*). Once *f* is computed the CR estimation method is easily extended and repeated to calculate the best *Q* factor for a given compression ratio using negligible computation overhead.

5) References

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

[2] Rao, K. R. and Yip P., Discrete Cosine Transform, Algorithms, Advantages and Applications, Academic Press, San Diego, CA, 1990.

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

[4] S. Wu and A. Gersho, "Rate-constrained picture-adaptive quantization for JPEG baseline coders," *Proc. of Intl. Conf. on Acoust., Speech, Signal Processing*, Minneapolis, MN, vol. V, pp. 389--392, 1993.

[5] C. Honsiger, M. Rabbani, and P. Jones, "Method for efficient rate control," US Patent 6,356,668, Mar 12th 2002.

[6] A. B. Watson, "Visually optimal DCT quantization matrices for individual images," *Proc. SPIE Conf. Human Vision, Visual Processing and Digital Display IV*, pp. 1-10, 1992.

[7] B. Watson, "Perceptual optimization of DCT color quantization matrices," *Proc. of IEEE Intl. Conf. on Image Processing (ICIP)*, Vol. 1, pp. 100-104, Austin, TX, USA, 1994.

[8] B. Chitprasert and K. R. Rao, "Human visual weighted progressive image transmission," *IEEE Trans. On Communications*, COM-38, 1040-1044, July 1990.

[9] S. Daly, C.-T. Chen, and M. Rabbani, "Adaptive block transform image coding method and apparatus," US Patent 4,774,574, Sep. 27th 1988.

[10] H. A. Peterson, A. J. Ahumada, and A. B. Watson, "An improved detection model for DCT coefficient quantization," *Proc. SPIE Conf. Human Vision, Visual Processing and Digital Display IV*, pp. 1-10, 1993.

[11] T. D. Tran and R. Safranek, "A locally adaptive perceptual masking threshold model for image coding," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 1882-1885, Atlanta, May 1996.

[12] S. Westen, R. Lajendijk, and J. Biemond, "Optimization of JPEG color image coding using a human visual system model," *Proc. of SPIE*, Vol. 2657, 370-381, 1996.

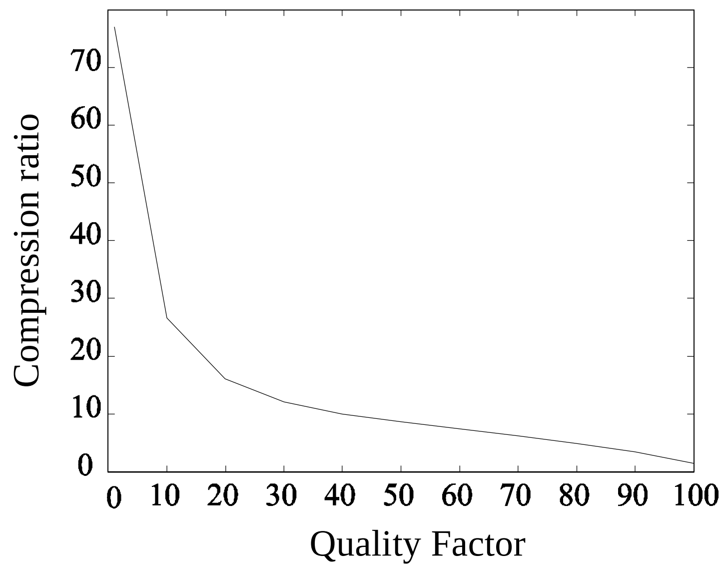


Figure 1 – "Quality" factor controls compression ration in JPEG. Typically, they are mapped to scaling numbers which are used to multiply the entries in the quantizer tables in JPEG, thus modifying monotonically the compression ratio.

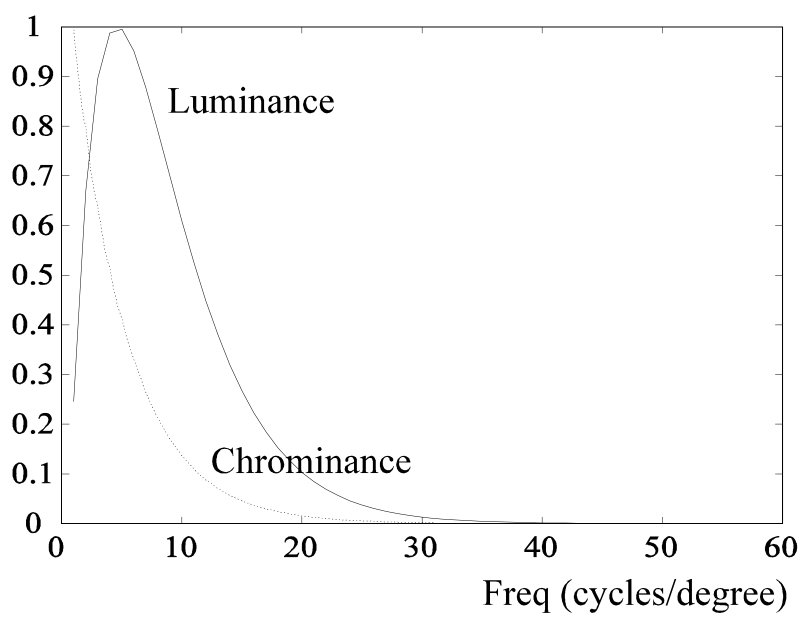


Figure 2 – HVS linear transfer function sensitivity model for luminance and chrominance images.

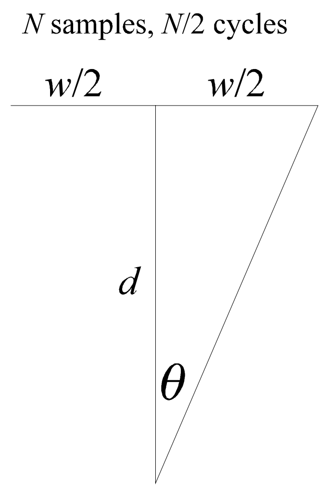


Figure 3 – Geometry of the viewing setting. Viewer is at a distance d watching a screen whose width is *w* comprising *N* pixels. Since There are *N* pixels, there are at most *N*/2 cycles to be observed.

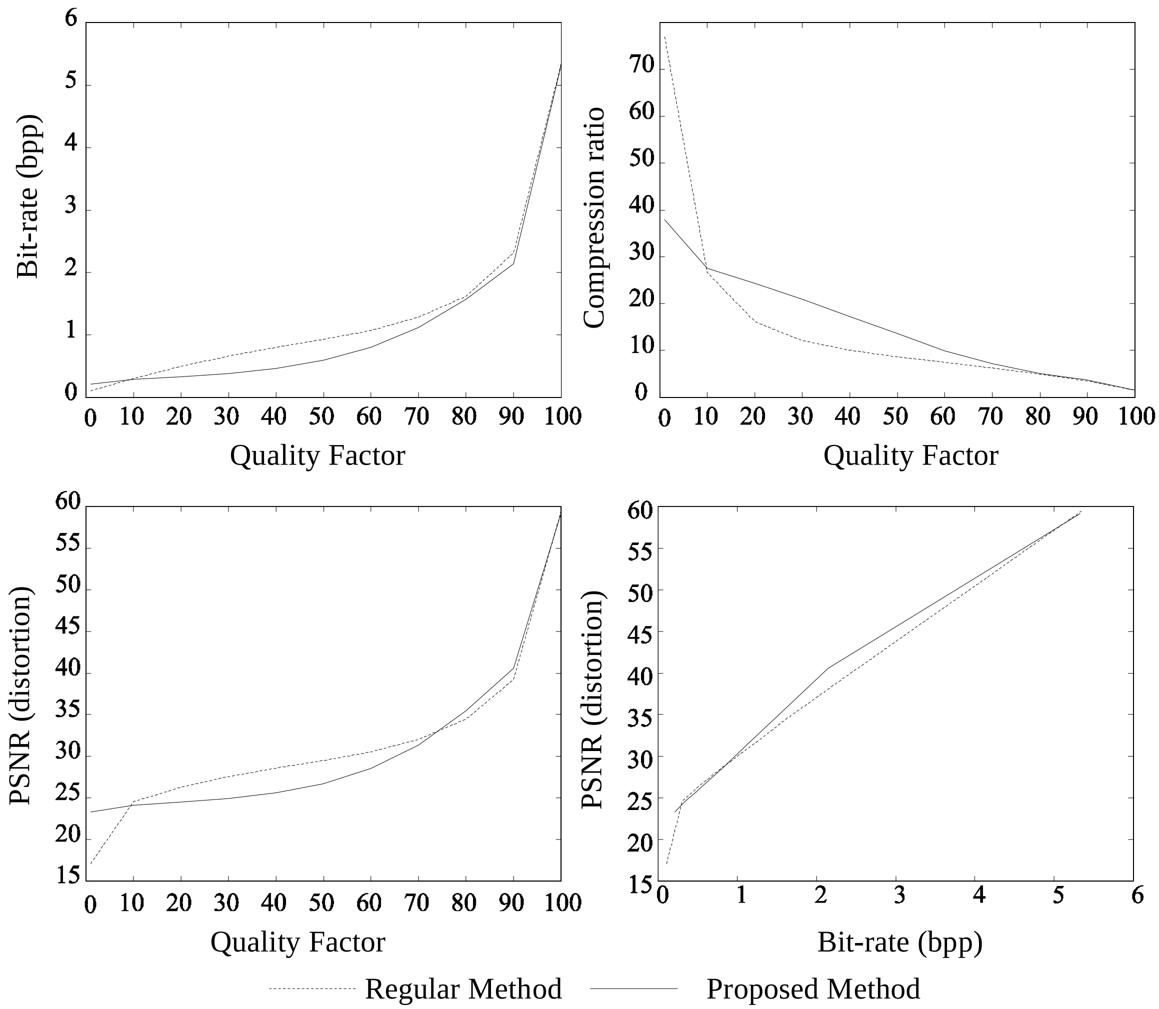


Figure 4 – Comparison of design methods. “Regular” refers to the scaling method.

***Figure 5* –**Zoom of reconstructed image after compression using the regular quantizer design method at 25:1(left). Same result for using the proposed design method is shown on the right.

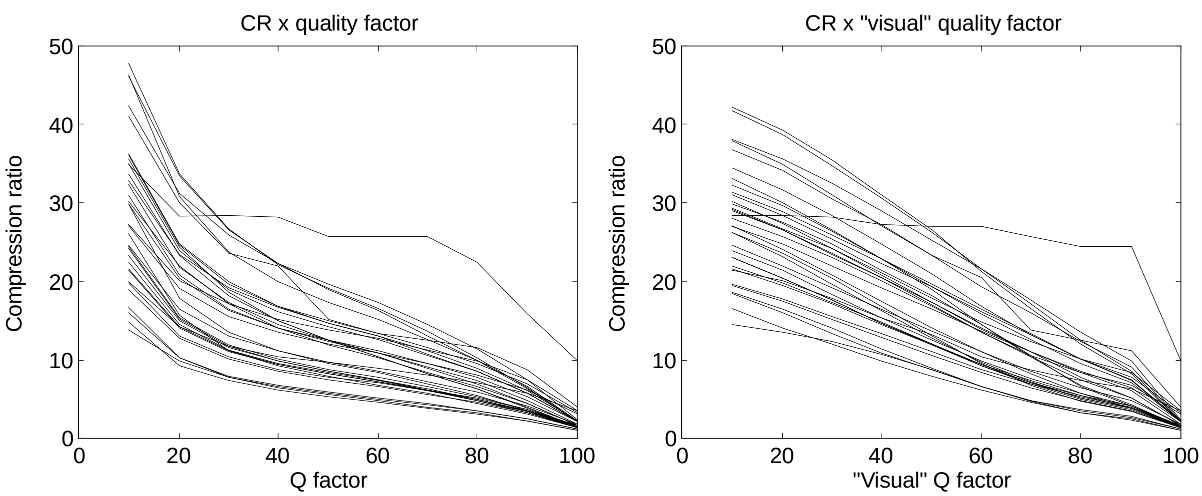


Figure 6 – Relation between quality factors and compression ratios. Visual Q factor refers to the proposed method (right). The plots on the left were computed for several images using the traditional scaling method. The plots on the right used the proposed method. Note the linearity of relation CR vs. Q for most images.

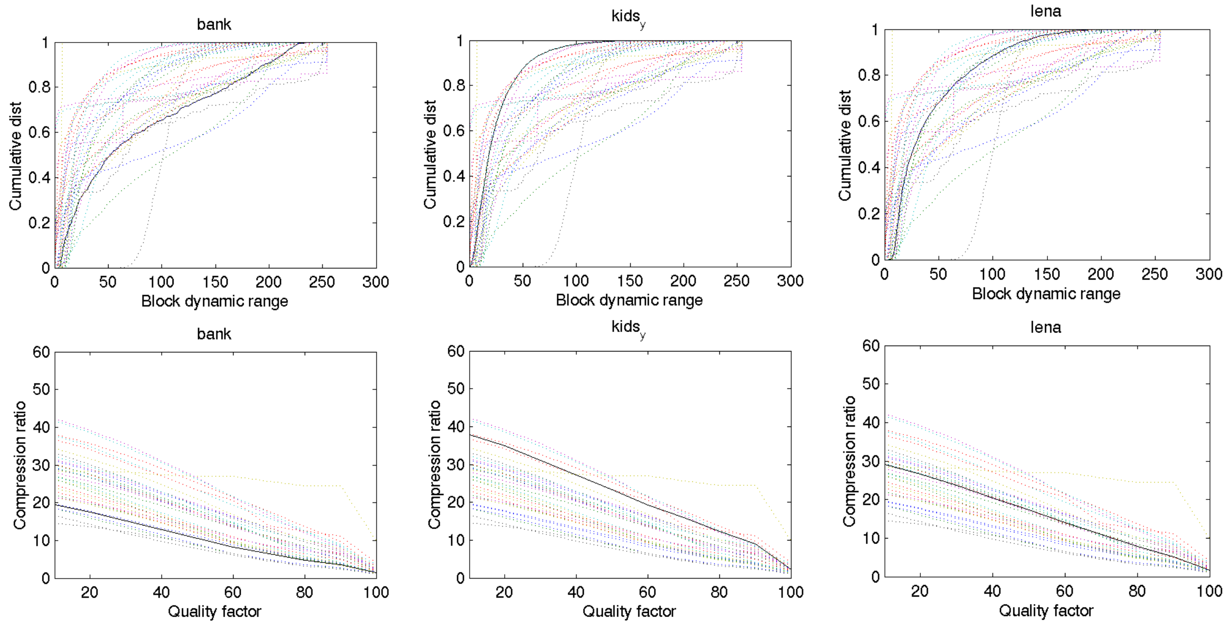


Figure 7 – Correlation between cumulative histograms vs. dynamic range and the relation CR vs. *Q*. Typically when the curves above rise too quickly the CR vs. *Q* line has the highest steep. Hence one can try to estimate the CR/*Q* relation from the cumulative distribution plots.

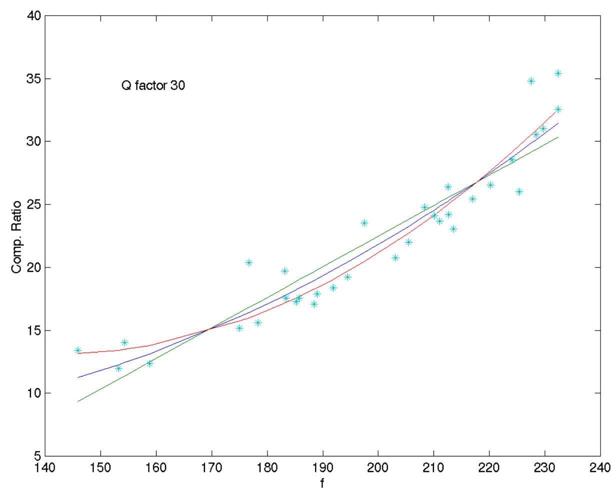


Figure 8 – LS-fitting the relation between the derived parameter f and compression ratios. Fitting can be done linearly or quadratically. We used the average of both approaches as a fitting curve.

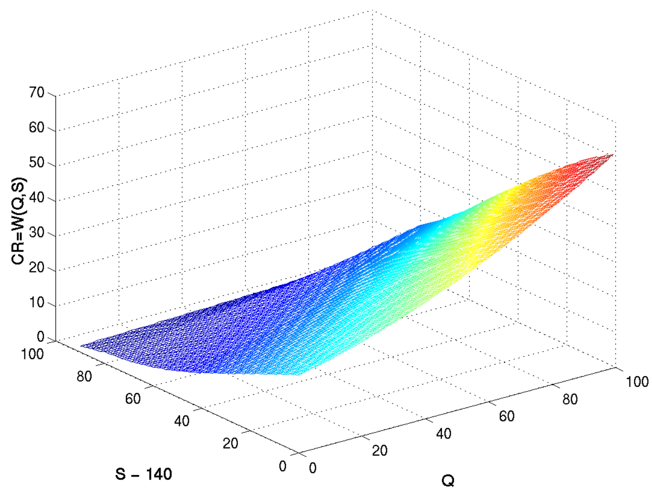


Figure 9 – Overall appearance of the *W*(*Q*,*S*) look-up table.

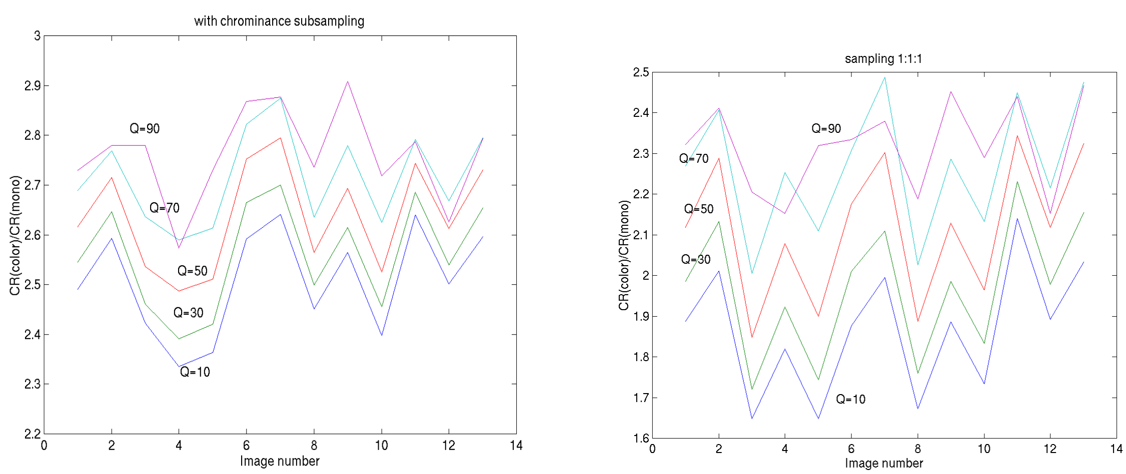


Figure 10 – Relation between the compression achieved by color and monochrome versions of the image. Results shown for different quality factors and different subsampling strategies.

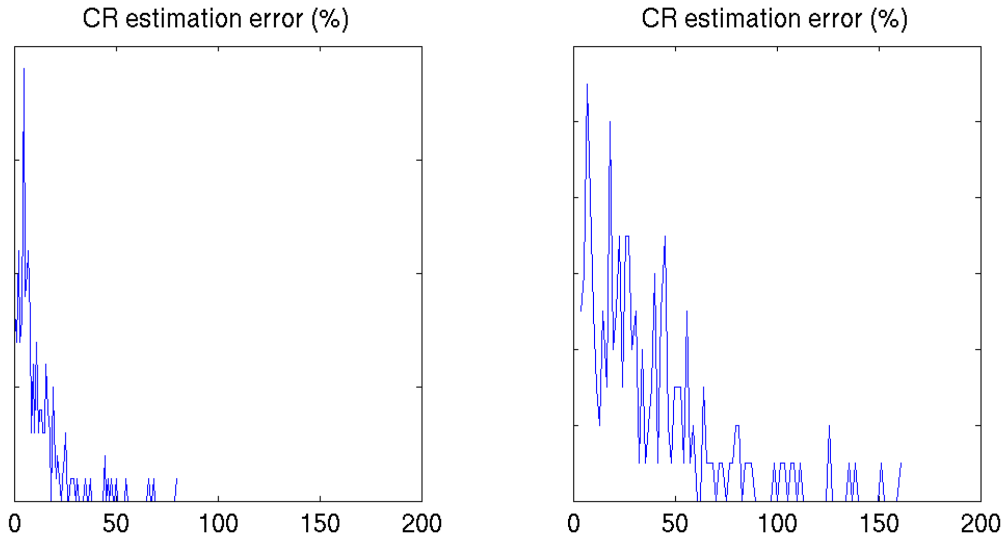
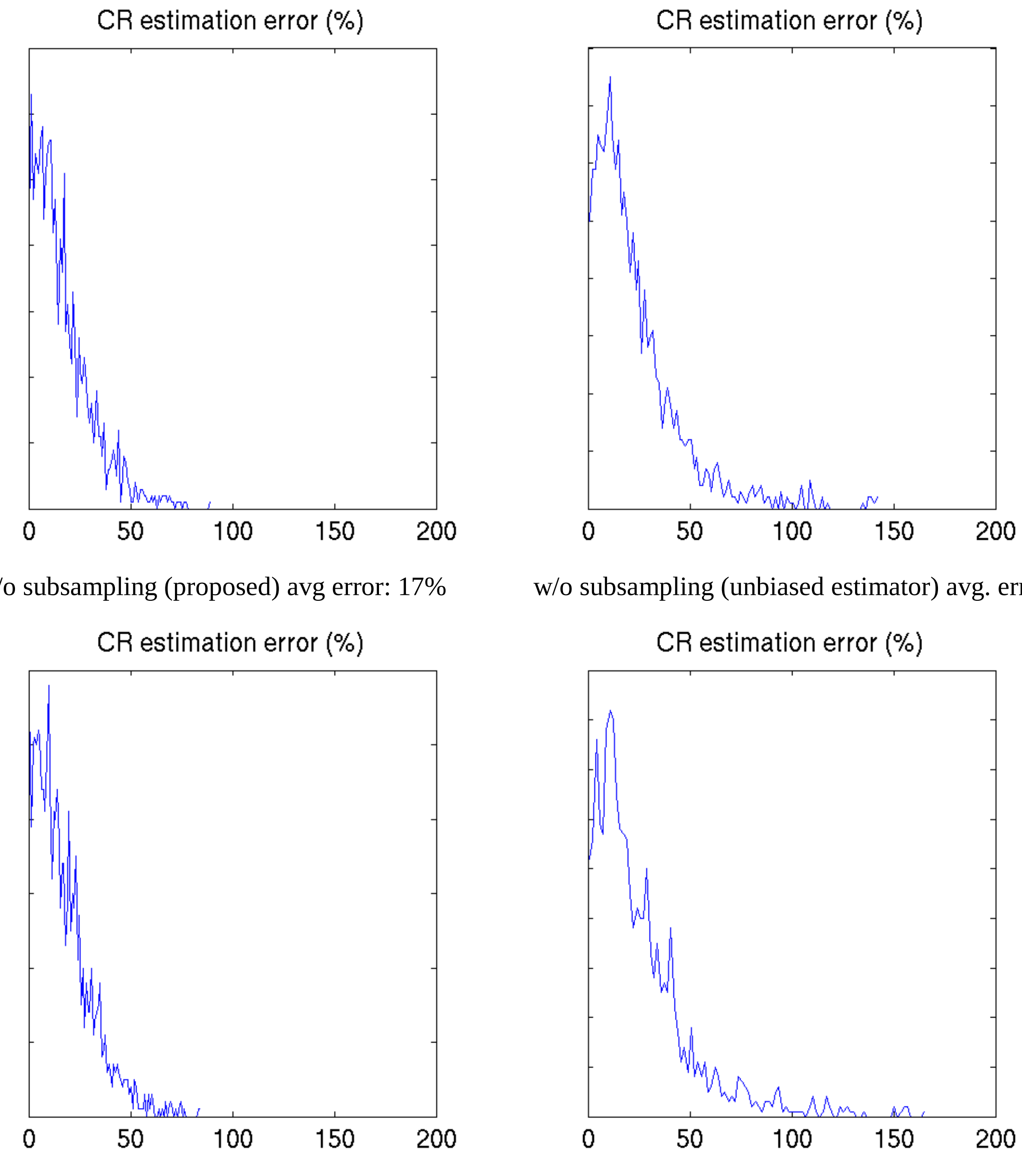


Figure 11- Estimation error comparing the unbiased estimator (right) and the proposed one (left). Measurements were computed from 34 monochrome images. Average estimation error for the proposed method is 12%. If we use the unbiased estimator the average error would be 37%.



w/o subsampling (proposed) avg error: 17%

w/o subsampling (unbiased estimator) avg. error: 24%

w/ subsampling (proposed) avg error: 17%

w/ subsampling (unbiased estimator) avg. error: 27%

Figure 12 – Histogram of estimation error measurements for color images using the proposed approach with “blind” estimation of the compression of the chrominance channels. It was used a pictorial set – Corel CD with 302 images. Using 5 Q factors per image brings the total number of sampels to 1510. The method’s average error is about 17%, against 27% (without subsampling) and 24% (with subsampling). Top: with chrominance subsampling, bottom: without. Left: proposed method, right: unbiased estimator.