Absolute moment block truncation coding and its application to color images

M. Lema

Robert Mitchell
University of Missouri–Rolla

Follow this and additional works at: http://scholarsmine.mst.edu/faculty_work

Part of the Electrical and Computer Engineering Commons

Recommended Citation
http://scholarsmine.mst.edu/faculty_work/1645

This Article is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. For more information, please contact weaverjr@mst.edu.
tion using the year-to-year standard deviation \( S_R \) at various exceedance percentile levels. These are labeled "0.5%," "5%," "95%," and "99.5%," correspondingly.

Also included on Figs. 4 and 5 are five years (1966-1970) of annual rain rate obtained from observed data for Denver, CO, provided by the Weather-Environment Group of the U.S. Air Force Communications Command at Scott Air Force Base, IL. Fig. 4 shows results using the earlier \( S_R \) values given in [1], whereas Fig. 5 shows results using \( S_R \) values corresponding to those given by (4). Clearly, the bounds of the predictions in Fig. 5 encompass the data, whereas those of Fig. 4 do not. While this may possibly be an exceptional instance, it is, however, apparent that at least in some cases the \( S_R \)'s of Figs. 1-3 will be more realistic than those given in [1].

**Fig. 4.** Comparison of five years of annual rain rate distributions taken at Denver, CO, with distribution predictions made using the year-to-year variability derived from the original Rice-Holmberg model.

**Fig. 5.** Comparison of five years of annual rain rate distributions taken at Denver, CO, with distribution predictions made using the year-to-year variability derived from the modified Rice-Holmberg model.

### References


### Absolute Moment Block Truncation Coding and Its Application to Color Images

**MAXIMO D. LEMA AND O. ROBERT MITCHELL**

**Abstract**—A new quantization method that uses the criterion of preserving sample absolute moments is presented. This is based on the same basic idea for block truncation coding of Delp and Mitchell but it is simpler in any practical implementation. Moreover, output equations are those for a two-level nonparametric minimum mean square error quantizer when the threshold is fixed to the sample mean. The application of this method to single frame color images is developed. A color image coding system that uses absolute moment block truncation coding of luminance and chroma information is presented. Resulting color images show reasonable performance with bit rates as low as 0.013 bits/pixel.

### I. INTRODUCTION

This paper presents a new quantization method that uses the idea of preserving moments as a design criterion of the quantizer. An extensive study of moment-preserving quantization was done by Delp in [2] and was applied to still images by Delp and Mitchell [1], [3] and to moving imagery by Healy and Mitchell [4]. A major application of moment-preserving quantization leads to what is known as block truncation coding or BTC. This coding method takes a finite number of samples and tries to preserve their first two sample moments, assigning a two-level quantizer designed for that goal. The equations of the quantizer are relatively simple, and block truncation coding has shown robust behavior in the presence of channel noise [1]. It also gives good reconstructed images, since the method preserves local characteristics of spatial blocks of the image important to the human observer. The method computes for each of these blocks the sample mean and the sample standard deviation, that is,

\[
\bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i
\]

\[
\sigma = \sqrt{\frac{1}{m} \sum_{i=1}^{m} x_i^2 - \bar{x}^2}
\]

where \( m \) is the total number of pixels in the block, and \( x_i \) represents the grey value of each pixel.

Both values are transmitted along with a bit plane which contains ones in those places where \( x_i \geq \bar{x} \) and zeros otherwise. At the receiver, the block is reconstructed with a two-level quantizer that preserves the sample mean and the sample variance. The output values that achieve this goal are (see [1])

\[
a = \bar{x} - \sigma \sqrt{\frac{q}{m-q}} \quad \text{for } x_i < \bar{x}
\]

\[
b = \bar{x} + \sigma \sqrt{\frac{m-q}{q}} \quad \text{for } x_i \geq \bar{x}
\]

where \( q \) is the number of pixels greater than or equal to \( \bar{x} \).

Paper approved by the Editor for Signal Processing and Communication Electronics of the IEEE Communications Society for publication without oral presentation. Manuscript received November 22, 1982; revised August 4, 1983. This work was supported in part by the U.S. Army Research Office.

M. D. Lema is with the Department of Electrical Engineering, Purdue University, West Lafayette, IN 47907, and the Empresa Nacional de Telecomunicaciones (ENTEL), Argentina.

O. R. Mitchell is with the Department of Electrical Engineering, Purdue University, West Lafayette, IN 47907.
The method presented in this paper preserves absolute moments rather than standard moments (see [5]). It will be shown that this method gives similar pictorial results to block truncation coding but it gives simpler equations which lead to faster computation and smaller mean square error than BTC. The relation between AMBTC and the nonparametric minimum mean square error (MMSE) two-level quantizer is given in Section III.

An application of this method to color images is discussed. The problem of coding color images starts with the trichromatic nature of human color vision and the correlation that generally exists between color planes of actual pictures. It is therefore important to consider some decorrelating operation. A system that uses absolute moment preservation is presented. Finally, results from coding actual color images with the proposed system are given.

II. ABSOLUTE MOMENT BLOCK TRUNCATION CODING (AMBTC)

Since the principal idea used in block truncation coding is to achieve compression while preserving some sample moments, there exist other variants that lead to simpler results. Here, a new method of coding still images that preserves absolute sample moments is presented. Let us call it absolute moment block truncation coding or AMBTC.

Let a digitized image be divided into blocks of $n \times n$ pixels. Each block is quantized in such a way that each resulting block has the same sample mean and the same sample first absolute central moment of each original block. Let $x_i$ be the grey level of a pixel in the block where $1 \leq i \leq m$ and $m = n^2$. Consequently, it is necessary to calculate the sample mean

$$\bar{\eta} = \frac{1}{m} \sum_{i=1}^{m} x_i$$

and the sample first absolute central moment

$$\bar{\alpha} = \frac{1}{m} \sum_{i=1}^{m} |x_i - \eta|.$$  

The mean value contains information about central tendency; this is the same central tendency information used by Delp and Mitchell in the original BTC [1]. On the other hand, the sample first absolute central moment contains information about dispersion about the mean. The corresponding value used by Delp and Mitchell is the sample standard deviation. Therefore, two of the most important local characteristics of the spatial block grey levels are preserved: central tendency and deviation from the center.

The sample first absolute central moment can be calculated in a simple way as follows: from (6)

$$m\bar{\alpha} = \sum_{x_i \geq \bar{\eta}} x_i - \bar{\eta} \sum_{x_i \geq \bar{\eta}} 1 - \sum_{x_i < \bar{\eta}} x_i + \bar{\eta} \sum_{x_i \leq \bar{\eta}} 1,$$

where $\bar{\eta}$ is the decision threshold of the quantizer (see Fig. 1). Since $q$ is the number of pixels above the threshold, then

$$q = \sum_{x_i \geq \bar{\eta}} 1 \quad \text{and} \quad m - q = \sum_{x_i < \bar{\eta}} 1,$$

then, from (5)

$$m\bar{\eta} = \sum_{x_i \geq \bar{\eta}} x_i + \sum_{x_i < \bar{\eta}} x_i.$$  

Inserting into (7)

$$m\bar{\alpha} = \sum_{x_i \geq \bar{\eta}} x_i - \bar{\eta}q - m\bar{\eta} + \sum_{x_i \geq \bar{\eta}} x_i + \bar{\eta}(m - q).$$

$$\bar{\alpha} = \frac{1}{m} \left[ \sum_{x_i \geq \bar{\eta}} x_i - \eta q \right].$$

Equation (8) is a more efficient way of computing $\bar{\alpha}$. Let us define a quantity that will be useful:

$$\bar{\gamma} = \frac{m\bar{\alpha}}{2} = \sum_{x_i \geq \bar{\eta}} x_i - \eta q.$$  

In order to preserve the moments given in (5) and (6), it is necessary to assign two values, $a$ and $b$, at the output of the two-level quantizer (see Fig. 1) such that

$$m\bar{\eta} = \sum_{i=1}^{m} y_i = qb + (m - q)a$$

$$m\bar{\alpha} = \sum_{i=1}^{m} |y_i - \bar{\eta}| = q(b - \bar{\eta}) - (m - q)(a - \bar{\eta}).$$

From (10) the value of $a$ is

$$a = \frac{m\bar{\eta} - qb}{m - q}. $$

Inserting into (11) the value of $b$ is

$$b = \bar{\eta} + \frac{\bar{\gamma}}{q}.$$  

Replacing it into (12) we obtain the value of $a$:

$$a = \bar{\eta} - \frac{\bar{\gamma}}{m - q}.$$  

The behavior of this quantizer can be analyzed by examining (13) and (14). For the case in which all the pixels have equal grey value, the deviation information is equal to zero and both $a$ and $b$ are set to the mean value, which is also the correct value for each pixel. On the other hand, if the pixels are dispersed about the mean, the value assigned to $b$ is the mean plus a bias that depends directly on the dispersion and inversely on the number of pixels above the mean. Similar reasoning can be made for the value $a$. A numerical quantization example of a typical block is presented in the Appendix.
Fig. 2. Original 512 x 480 image with 8 bits/pixel.

Fig. 3. Standard BTC reconstructed image using 4 x 4 pixel blocks, 6 bits for each sample mean, and 4 bits for each sample standard deviation. The bit rate is 1.625 bits/pixel. The mean square error is 32.35.

Fig. 4. AMBTC reconstructed image using 4 x 4 pixel blocks, 6 bits for each sample mean, and 4 bits for each sample first absolute central moment. The bit rate is 1.625 bits/pixel. The mean square error is 30.00.

Fig. 5. Original 256 x 240 image with 8 bits/pixel.

Fig. 7. Quantized AMBTC image. All the images were reconstructed using the same rate, 1.625 bits/pixel, but the mean square error for those using AMBTC is smaller. It has been found that the major artifacts are the same as those noted by Delp in [2], which are edge raggedness, and misrepresentation of some midrange values due to their assignment to either high or low value.

III. ERRORS IN AMBTC

In order to compare this quantizer to the two-level MMSE quantizer, let us substitute (9) into (13) and (14). The result is

\[ a = \frac{1}{m-q} \sum_{x_i \leq \bar{x}} x_i \]
\[ b = \frac{1}{q} \sum_{x_i > \bar{x}} x_i \]
In order to analyze the error introduced by this non-parametric quantizer, let us define a deterministic mean square error as

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} (y_i - x_i)^2.$$  \hfill (17)

Let us define

$$g(\bar{\eta}) = \sum_{x_i < \bar{\eta}} x_i.$$ \hfill (18)

Then

$$a = g(\bar{\eta})(m - q)$$

$$b = (m\bar{\eta} - g(\bar{\eta}))/q$$ \hfill (19)

so

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} [x_i^2 + y_i^2 - 2x_i y_i].$$ \hfill (20)

After replacing values for $y_i$ we get

$$\text{MSE} = \bar{x}^2 - \frac{g(\bar{\eta})^2}{m(m - q)} - \frac{(m\bar{\eta} - g(\bar{\eta}))^2}{mq}$$ \hfill (21)

where

$$\bar{x}^2 = \frac{1}{m} \sum_{i=1}^{m} x_i^2.$$ \hfill (22)

If the histogram of the block is symmetric about the sample mean and $m$ is even, we have $q = m - q = m/2$. From (9)

$$g(\bar{\eta}) = \frac{m}{2} (\bar{\eta} - \bar{\alpha})$$

$$m\bar{\eta} - g(\bar{\eta}) = \frac{m}{2} (\bar{\eta} + \bar{\alpha}).$$ \hfill (23)

Replacing these values into the expression for the mean square error, we get

$$\text{MSE} = \bar{\sigma}_1^2 - \bar{\sigma}_2^2.$$ \hfill (24)

It is important to note here, that in those cases where the standard deviation equals the first absolute central moment, the error becomes zero. This situation occurs, for example, when all the pixels have the same grey level or when half of them take a value $\eta + A$ and the other half take a value of $\eta - A$ for any fixed $A$.

Other values for the threshold could be used. Goeddel and Bass [6] report that using $t = (x_{\min} + x_{\max})/2$ as a threshold yields lower mean square errors in standard BTC, where $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the pixels in a block. This fact was verified by the authors for some images and current research is being done aimed to find other measures of central tendency which could yield even lower mean square errors.

IV. ADVANTAGES OF AMBTC

In the case that the quantizer is used to transmit an image from a transmitter to a receiver, it is necessary to compute at the transmitter two quantities using either BTC or AMBTC. These two quantities are the sample mean and the information about deviation from the center, which is the sample standard deviation for BTC and the sample first absolute central moment for AMBTC. Since computation of central information is the
same for both methods, let us compare the necessary computations for deviation information. For standard BTC is it necessary to compute a sum of \( m \) values, see (2), each of them squared, while in the case of AMBTC it is only necessary to compute the sum of values given in (6). Since the multiplication time in most digital processors is several times greater than the addition time, by using AMBTC the total calculation time at the receiver is also substantially reduced.

A similar situation occurs at the receiver. For standard BTC it is necessary to evaluate two quotients and two square roots, see (3) and (4), while using AMBTC those calculations simplify to only two quotients, see (13) and (14). Consequently, the processing time at the receiver is also substantially reduced. The savings in time achieved by AMBTC might suggest the name of fast BTC for this method.

The rest of the characteristics of AMBTC remain the same as those of BTC since the philosophy of both methods is substantially the same.

An additional advantage of AMBTC is the minimum mean square error achieved under symmetry conditions stated before. As expected, for all the images tested in the Laboratory for Image Processing at Purdue University, AMBTC achieved smaller mean square error than standard BTC. Results for some of these images (see Fig. 8) are shown in Table 1.

### V. APPLICATION OF AMBTC TO COLOR IMAGES

#### A. The Color Signal
Because of the trichromatic nature of the human vision, we can assume that the color signal that we want to code lies in a three-dimensional space. Election of the coordinate system to represent this signal is often limited by practical situations. Because of the monitoring system available, we have used the NTSC receiver phosphor primary system [8] in the implementation of color AMBTC. This system defines three primaries \( R, G, \) and \( B \) such that

\[
0 \leq R \leq 1; \quad 0 \leq G \leq 1; \quad 0 \leq B \leq 1.
\]  

(24)

The relationship between this group of primaries and other coordinate systems is given in the literature [8]. From this point we assume that our gamut of color is limited to the reproducible colors in the NTSC primary system. Taking into account the boundaries for \( R, G, \) and \( B, \) the solid of all reproducible colors is a cube such as in Fig. 9.

#### B. Correlation Between Color Planes
In typical color image data, there exists a high degree of correlation between the planes \( R, G, \) and \( B. \) It is well known that more efficient coding is achieved when the different output signals are decorrelated. It is therefore convenient to perform some decorrelating operation on the planes \( R, G, \) and \( B \) before trying to apply AMBTC to the three signals. It is also desired to have an invertible transformation. The ideal decorrelation operation is a Karhunen-Loeve transform that would theoretically achieve total decorrelation between planes. However, such an approach leads to lengthy computations of eigenvalues and eigenvectors and depends on the image statistics. Thus, it is impractical.

There exist other transformations that, although they do not achieve optimum results, are close to the optimum case and are much simpler in practical implementation. We will make use of one particular transformation, the NTSC \( R-G-B \) to \( Y-I-Q \) transformation. This is the method used in the NTSC color TV system. Also, as is shown in [9], [10], it has \( RGB \) component space energy compaction properties comparable to Karhunen-Loeve for most images.

The direct \( R-G-B \) to \( Y-I-Q \) transformation is given by

\[
\begin{bmatrix}
Y \\
I \\
Q
\end{bmatrix} = L \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}; \quad \text{where } L = \begin{bmatrix}
0.299 & 0.587 & 0.114 \\
0.596 & -0.274 & -0.322 \\
0.211 & -0.523 & 0.312
\end{bmatrix}
\]

(25)

and the inverse \( Y-I-Q \) to \( R-G-B \) transformation is given by

\[
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix} = L^{-1} \begin{bmatrix}
Y \\
I \\
Q
\end{bmatrix}; \quad \text{where } L^{-1} = \begin{bmatrix}
1.000 & 0.956 & 0.621 \\
1.000 & -0.272 & -0.647 \\
1.000 & -1.106 & 1.703
\end{bmatrix}
\]

(26)
is more sensitive to spatial variations of sampled by taking averaging windows of 
\begin{equation}
\text{getting a high compression and remembering that the human eye given in (25) to get the NTSC}
\end{equation}

The range of the \(Y, I\), and \(Q\) signals pass through a linear transformation \(R, G, B\) signals are outside the \(Y-I-Q\) space are reproducible NTSC colors. On the other hand, only those points that lie inside the solid of Fig. 10 in the \(Y-I-Q\) space are reproducible NTSC colors in the mapped \(R-G-B\) cube. Our problem then is to not allow the reconstructed \(Y, I,\) and \(Q\) signals to fall outside the solid of Fig. 10 in order to have reproducible colors when they are converted again into \(R-G-B\) signals. This problem arises in the proposed system since each of the signals suffers different distortions in their paths. Those distortions sometimes give place to groups of \(Y', I',\) \(Q'\) values that are outside the reproducible color solid.

Consequently, it is necessary to modify this situation using some specified criterion. We will propose two methods of correcting the errors. One, which we call “correction by truncation,” is relatively simple to implement. The other, which we call “correction by preserving luminance,” tries to preserve an important quantity for the human observer, the luminance of the image.

**Correction by Truncation:** This method consists in keeping the received \(R-G-B\) value within the bounds for these signals. If the received group is inside the cube of Fig. 9 we take this group as our reconstructed point \(R, G, B\). In the case that one or more signals \(R, G,\) or \(B\) are outside the permitted range, we take the corresponding quantity to its nearest reproducible bound. For example:

\[
R' = 0.5, \quad R = 0.5 \\
G' = 1.2, \quad G = 1.0 \\
B' = -0.3, \quad B = 0.0.
\]

**Correction by Preserving Luminance:** This approach is based on the fact that the luminance signal \(Y\) is the signal that suffers the least distortion in our system (see Fig. 11). Therefore, we would like to preserve the received value of \(Y\).

It turns out that it is more convenient to analyze this problem in the \(Y-I-Q\) space. In this space, the planes with constant luminance intersect the solid of Fig. 10, forming regions \(I-Q\) that show all reproducible values for a given \(Y\). These regions are shown for several luminance values in Fig. 12. As we can see, there are six possible lines which define the borders of those regions and they move on the \(I-Q\) plane according to the value of \(Y\). The equations of those lines come from intersecting each one of the six faces of the solid of Fig. 10 with a plane of constant luminance. From (26) they are

\[
	ext{for } G = 1: \quad I = (Y - 1)/0.272 - 2.378Q \\
	ext{for } R = 1: \quad I = (1 - Y)/0.956 - 0.649Q \\
	ext{for } B = 1: \quad I = (Y - 1)/1.106 + 1.539Q \\
	ext{for } G = 0: \quad I = Y/0.272 - 2.378Q \\
	ext{for } R = 0: \quad I = -Y/0.956 - 0.649Q \\
	ext{for } B = 0: \quad I = Y/1.106 + 1.539Q.
\]
In general, the regions are defined by less than six border lines; however, all of them were used in the correction algorithm. We will study the correcting method, making use of Fig. 13. The received value is given by $Y_R, I_R, Q_R$ and the corrected value is $\hat{Y}, \hat{I}, \hat{Q}$. Therefore, $\hat{Y} = Y_R$ is the value that determines the particular shape of the $I-Q$ region. It must be noted that, given a particular line defined by the previous equations, the reproducible point is always on the semiplane divided by this line that contains the origin. The correction algorithm simply takes a received point, and for each of the lines defined by (30) asks the following question: is the received pair $(I_R, Q_R)$ in the semiplane divided by the current line that contains the origin? If the answer is yes, the question is made for the next line. If the answer is no, it corrects $I$ and $Q$. The corrected values of $\hat{I}$ and $\hat{Q}$ are obtained taking the intersection of the current border line and a line that passes through the received point and the origin. Suppose that the current line equation is

$$I = fQ + g$$

(31)

where $f$ and $g$ depend on the received luminance value and on the line under analysis [see (30)]. The corrected value of $\hat{Q}$ is then

$$\hat{Q} = \frac{g}{\left(\frac{I_R}{Q_R} - f\right)}$$

for $Q_R \neq 0$

(32)
E. Results of the System for Actual Images

The system proposed in Fig. 11 has been tested, having as input a $256 \times 256$ pixel color image with 8 bits per pixel for each color plane, that is, 24 bits per pixel. The original image is in the upper left corner of Fig. 14. The image on the upper right corner of Fig. 14 was obtained with the proposed system using correction by preserving luminance. The image on the lower left corner of Fig. 14 was obtained using correction by truncation. The differences between these two corrected images are minor. The bit rate of both is 2.13 bits/pixel at the input of the receiver. This bit rate is achieved in the following manner: bit rate for $Y$, 26/16 bits/pixel; bit rate for $I$, 26/64 bits/pixel; bit rate for $Q$, 26/256 bits/pixel.

Since the original has 24 bits/pixel, a good deal of compression has been achieved, more than $11:1$. Note also that the color information does not need many additional bits. Indeed, the bit rate of $I$ plus the bit rate of $Q$ is about 0.5 bits/pixel or a third part of the luminance's bit rate.

In order to evaluate errors introduced by the system, let us define the relative mean square error for one color plane (as in [10]) as the average of square differences divided by the estimated variance of the plane. We then can define an average total error as

$$\epsilon_T^2 = (\epsilon_R^2 + \epsilon_G^2 + \epsilon_B^2)/3.$$  \hspace{1cm} (34)

All these errors were evaluated for the images in Fig. 14. The values of the original variances are

$\sigma_R^2 = 587$  \hspace{1cm} $\sigma_G^2 = 189$  \hspace{1cm} $\sigma_B^2 = 252$

For the image obtained using correction by truncation, the errors are

$$\epsilon_R^2 = 12\% \hspace{1cm} \epsilon_G^2 = 24.2\% \hspace{1cm} \epsilon_B^2 = 34.3\% \hspace{1cm} \epsilon_T^2 = 23.5\%.$$  

For the image obtained using correction by preserving luminance, the corresponding values are

$$\epsilon_R^2 = 13.3\% \hspace{1cm} \epsilon_G^2 = 24.4\% \hspace{1cm} \epsilon_B^2 = 34.2\% \hspace{1cm} \epsilon_T^2 = 24\%.$$  

Although the errors of the second set of values are slightly bigger than the first one, the advantage of the method that corrects by preserving luminance is that it keeps a low error on the reconstructed luminance which can be useful in some cases. Furthermore, results of a subjective test using five untrained persons showed that four of them chose the image obtained using correction by preserving luminance.

The principal artifact of the reconstructed upper right and lower left images in Fig. 14 is that they present some false contours in regions of slow changes in luminance and in chrominance. This is more noticeable in the background of the upper right and lower left images of Fig. 14. In order to improve these images, an AMBTC quantizer that uses two-dimensional quantization for $\hat{R}$ and $\hat{Q}$ has been implemented. The major idea in using this type of quantizer is given by Healy and Mitchell in [4]. They explain that in low variance zones, the value of the mean is very important for the observer, and therefore, it is convenient to assign more bits to the mean. On the other hand, in high variance zones, the mean is not critical for the human and fewer bits can be assigned to the mean.

Reference [4, Table I] shows the characteristics of the joint quantizer used. The resulting image, using three AMBTC 10 bit two-dimensional quantizers in the color system of Fig. 11, is shown in the lower right corner of Fig. 14. As expected, the false contours have disappeared. Some chroma errors are still present in this image, especially under the lips and other parts of the face. These chroma errors could be due to the high subsampling in one of the signals that carries color, the $Q$ signal. However, the image on the lower left of Fig. 14 has a great similarity with the original. The errors for this image are

$$\epsilon_R^2 = 12\% \hspace{1cm} \epsilon_G^2 = 22\% \hspace{1cm} \epsilon_B^2 = 33\% \hspace{1cm} \epsilon_T^2 = 22.3\%.$$  

The errors obtained for this two-dimensional quantized image are smaller than those of both previous methods.

VI. CONCLUSION

An improvement in BTC can be obtained by preserving absolute moments. This method is called absolute moment block truncation coding or AMBTC. Both computing speed and reconstructed image quality are improved by preserving absolute moments instead of standard moments. The new method has the same general characteristics as BTC which include low storage requirements and an extremely simple coding and decoding technique. Color images can also be coded using this method. The block size for the chroma data can be increased by less frequent spatial sampling, resulting in good quality reconstructed color imagery.

APPENDIX

To illustrate absolute moment block truncation coding, let us present the following example. The image to be coded has been divided into nonoverlapping blocks of $4 \times 4$ pixels.
Suppose that one of the blocks is

\[
X = \begin{bmatrix}
121 & 114 & 56 & 47 \\
37 & 200 & 247 & 255 \\
16 & 0 & 12 & 169 \\
43 & 5 & 7 & 251
\end{bmatrix}
\]

Using (5) and (6), the corresponding sample moment values are

\[
\bar{\eta} = 98.75 \quad \text{and} \quad \bar{\alpha} = 83.22 \quad \text{or} \quad \bar{\gamma} = \frac{m\bar{\alpha}}{2} = 665.76.
\]

Taking the mean value as threshold, the bit plane is

\[
B = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\(\bar{\eta}, \bar{\gamma}, \) and \(B\) are sent to the receiver. At that point, \(a\) and \(b\) are computed using (13) and (14):

\[
a = 24.8 \quad b = 193.8
\]

so, the reconstructed block becomes

\[
X_R = \begin{bmatrix}
193.8 & 193.8 & 24.8 & 24.8 \\
24.8 & 193.8 & 193.8 & 193.8 \\
24.8 & 24.8 & 24.8 & 24.8 \\
24.8 & 24.8 & 24.8 & 193.8
\end{bmatrix}
\]

One can check that \(X_R\) has the same sample mean and the same sample first absolute central moment as the original block \(X\). The mean square error for this example is \(MSE = 1605\).

This example was taken to be the same as that presented in [1] by Delp and Mitchell, which gives the following reconstructed plane using standard BTC:

\[
X_{R\text{BTC}} = \begin{bmatrix}
204.1 & 204.1 & 16.8 & 16.8 \\
16.8 & 204.1 & 204.1 & 204.1 \\
16.8 & 16.8 & 16.8 & 204.1 \\
16.8 & 16.8 & 16.8 & 204.1
\end{bmatrix}
\]

and the mean square error for this example, using standard BTC, is \(MSE = 1687\). Note that the necessary quantity of information to be transmitted is given by

\[
\frac{N_\eta + N_\alpha + m}{m} \quad \text{[bits/pixel]}
\]

where \(N_\eta\) is the number of bits of the \(\bar{\eta}\) value; \(N_\alpha\) is the number of bits of the \(\bar{\alpha}\) value; and \(m\) the number of pixels in the block. Typical values for \(N_\eta\) are 6–8 bits and for \(N_\alpha\) are 4–6 bits, which gives about 1.675–1.875 bits/pixel.

**REFERENCES**


Algorithm for Construction of Variable Length Code with Limited Maximum Word Length

HITOMI MURAKAMI, SHUICHI MATSUMOTO, AND HIDE0 YAMAMOTO

Abstract—As a high efficiency coding method for TV signals, variable length coding, such as Huffman coding, is extremely effective. However, when this variable length coding is applied to an actual TV codec which requires high-speed real-time processing, maximum word length will be limited for the hardware configuration. This correspondence describes an algorithm for the construction of a modified Huffman code with limited maximum word length by means of a “top-down” procedure meeting this requirement instead of the conventional “bottom-up” algorithm.

I. INTRODUCTION

As a high efficiency coding method for TV signals, variable length coding, such as Huffman coding, is extremely effective. The use of this variable length coding makes it possible to reduce the bit rate by nearly 1 bit/pel [1]. However, when this coding is applied to an actual TV codec, the maximum word length is severely limited for the hardware configuration. For instance, when variable length coding is carried out in a codec in which TV signals are encoded into 8 bit or 9 bit PCM signals by an A/D converter, the desirable maximum word length is 12-14 bits [3], [4]. If a quantizer with 30-40 quantizing levels is used for reducing the bit rate in the codec under such a condition, conventional Huffman code and its “bottom-up” algorithm for construction [5] are not suitable in that such a conventional algorithm may generate lengthy words of more than desirable maximum word length. Thus, an algorithm for a construction obtaining an optimum code with the minimum average word length in limited maximum word length under the above condition becomes necessary. To attain this purpose, the “top-down” algorithm [6], instead of the conventional bottom-up algorithm, is effective and is required.

This correspondence describes a practical algorithm for the construction of a Huffman code by means of a top-down procedure meeting the above-mentioned purpose. An algorithm for construction of variable length codes with limited maximum word length (modified Huffman coding) by applying the top-down algorithm is also described.

II. CONSTRUCTION OF HUFFMAN CODE BY TOP-DOWN ALGORITHM

Let the number of messages (quantization levels) to be coded in the ensemble be $N$ and let the occurrence probability of the $i$th message be $p(i)$. Then

$$
\sum_{i=1}^{N} p(i) = 1. 
$$

The length of a code word, $l(i)$, is the number of coding bits assigned to it. Therefore, the average word length is

$$
l_{av} = \sum_{i=1}^{N} p(i) \cdot l(i). 
$$

For an optimum code, the length of a given message code word can never be less than the length of a more probable message code word. Therefore, it may be assigned that the code word in the ensemble has been ordered in a fashion such that

$$
p(1) \leq p(2) \leq \cdots \leq p(N) 
$$

and that, in addition, for an optimum code, the condition

$$
l(1) \leq l(2) \leq \cdots \leq l(N) 
$$

holds. This requirement is assigned to be satisfied throughout the following discussion.

The top-down procedure for constructing a variable length code that minimizes this average word length $l_{av}$ is as follows.

In the trees constructing optimum codes, it is assumed that message $k$ is assigned to an optional tree at the $n$th node, and each following tree at the $(n+1)$th node is assigned to each of the messages $k+1, k+2, \cdots, k+m-1, k+m$, respectively, as shown in Fig. 1. In this case, the relations of $p(k) \geq p(k+1) \geq \cdots \geq p(k+m)$, and $k+m \leq N$ are held.

In this assignment of messages to the trees, the number of message $m$ to be assigned for the $(n+1)$th node is given by the following equation, where the $l(j)$ means the word length of message $j$.

$$
m = 2^{l(k)+1} \left[ 1 - \sum_{j=1}^{N} 2^{-l(j)} \right] 
$$

Whether the code of the tree of message $k$ at the $n$th node further increases or not is determined by the following algorithm, using (5), which shows the relation between trees $k$ and $k+m$.

1) When

$$
p(k) \geq \sum_{i=m+k}^{N} p(i),
$$

the word length $l(k)$ of the $k$th message does not increase any more. The $k$th code terminates here.

2) When

$$
p(k) < \sum_{i=m+k}^{N} p(i)
$$

Paper approved by the Editor for Signal Processing and Communication Electronics of the IEEE Communications Society for publication without oral presentation. Manuscript received April 1, 1982; revised April 9, 1984.

The authors are with KDD Research and Development Laboratories, Tokyo, 153, Japan.

Fig. 14. Upper left: original $256 \times 256$ color image (8 bits/color = 24 bits/pixel). Upper right: reconstruction using AMBTC and correction by preserving luminance (2.13 bits/pixel). Lower left: reconstruction using AMBTC and correction by truncation (2.13 bits/pixel). Lower right: reconstruction using AMBTC, correction by truncation, and a 10 bit two-dimensional quantizer for each mean and deviation (also 2.13 bits/pixel).