MULTIPLICATION-FREE VECTOR QUANTIZATION USING L_1 DISTORTION MEASURE AND ITS VARIANTS

V. JOHN MATHEWS AND MEHRDJI KHORCHIDIAN

Department of Electrical Engineering, University of Utah
Salt Lake City, Utah 84112

ABSTRACT

Vector quantization is a very powerful technique for data compression and consequently, it has attracted a lot of attention lately. One major drawback associated with this approach is its extreme computational complexity. This paper first considers vector quantization that uses the L_1 distortion measure for its implementation. The L_1 distortion measure is very attractive from an implementational point of view, since no multiplication is required for computing the distortion measure. Unfortunately, the traditional Linde-Buzo-Gray (LBG) method for designing the codebook for the L_1 distortion measure can become extremely time-consuming, since it involves several computations of medians of very large arrays. We propose a gradient-based approach for codebook design that does not require any multiplications or median computations. The codebook design algorithm is then extended to a distortion measure that has piecewise-linear characteristics. Once again, by appropriate selection of the parameters of the distortion measure, the encoding as well as the codebook design can be implemented with zero multiplications. Finally, we apply our techniques in predictive vector quantization of images and demonstrate the viability of multiplication-free predictive vector quantization of image data.

I. INTRODUCTION

The advantages associated with representing, transmitting, and storing information in digital form are well-known. Perhaps the most serious disadvantage associated with the conversion of information from analog to digital form is the substantial increase in required bandwidth. This disadvantage can be mitigated by operating on the data with an appropriate data compression algorithm prior to transmission or storage. Vector quantization is a particularly effective, but computationally intensive, class of algorithms that has attracted a lot of interest in recent years. These algorithms take their name from the fact that they operate on entire k-dimensional vectors of waveform samples at once, rather than one sample at a time. They can be applied to a variety of waveforms such as speech [4], images [7], and modem signals [8], and they promise to be of great value in a variety of applications involving such waveforms.

The simplest form of a vector quantizer operates as follows. First, a codebook of k-dimensional vectors is created using a training set representative of the waveforms. Once the codebook is designed and the representative vectors are stored, the process of encoding the incoming waveform can begin. k consecutive samples of the waveform are grouped together to form a k-dimensional vector at the input of the vector quantizer. The input vector is successively compared with each of the stored vectors and a metric or distance is computed in each case. The representative vector closest to the input vector is identified, and the index of the closest vector is available at the output for transmission or storage.

Clearly, the full-search vector quantizer described above can become computationally very intensive, depending on the distortion measure employed. One very easy way of reducing the complexity of vector quantizers is to use the L_1 distortion measure where the distance between two vectors X and Y is computed as

$$d_1(X,Y) = \sum_{i=1}^{k} |x_i - y_i| , \qquad (1)$$

where x_i and y_i are the i-th elements of k-dimensional vectors X and Y. It is evident from the above equation that vector quantization using the L_1 distortion measure requires zero multiplication for its implementation. However, the L_1 distortion measure may not be appropriate in all applications. For such cases, we will consider a piecewise linear distortion measure given by

$$d_{\ell}(X,Y) = \sum_{i=1}^{k} f(x_i - y_i), \qquad (2)$$

where

$$f(e) = \sum_{i=1}^{k} \alpha_j |e| + \gamma_j \qquad (3)$$

is a piece-wise linear function and α_j and γ_j are functions of |e| and can take values from a finite set depending on which of the preselected intervals contain |e|. Figure 1 shows a typical f versus e curve. Note that if the slope values α_j's are selected to be integer powers of 2, the vector quantizers that make use of the distortion measure in Eq. 3 can be implemented without any multipliers. Furthermore, by appropriate choice of the thresholds τ_j's in Fig. 1, we may be able to approximate several other distortion measures using the piece-wise linear distortion measure, and therefore make the vector quantizer perform similar to those that use other distortion measures and at the same time implement it with highly reduced computational complexity.

This paper deals with vector quantization using the L_1 and piece-wise linear distortion measures. We first consider the design of a codebook when the L_1 distortion measure is used. The traditional Linde-Buzo Gray (LBG) algorithm [3] when used with the L_1 distortion measure requires computation of the median values of several large sequences and can become computationally very complex. We will present a gradient-based approach that does not require any multiplications or median computations for its implementation. We then extend this method to the piece-wise linear distortion measure. Finally, we will apply our method to predictive vector quantization of images. The implementation of the predictive vector quantizer is made multiplication-free by selecting the coefficients of the predictor to be (possibly negative) integer powers of two.

II. CODEBOOK DESIGN FOR THE L_1 DISTORTION MEASURE

The traditional approach to codebook design for vector quantizers is a clustering method known as the Linde-Buzo-Gray algorithm. Given a representative training sequence and an initial codebook, the LBG method consists of encoding the training sequence and then replacing each code vector by the centroid of all the training vectors that were mapped into the code vector. This is repeated until convergence is achieved. For the L_1 distortion measure, the centroid is the vector whose elements are the medians of the corresponding elements of all the training vectors that were mapped into it. Calculation of the median of a large set of numbers can become very complicated and time-consuming. We now propose a gradient-based algorithm that works very well in spite of the fact that it does not require any multiplications or median calculations.

Gradient Algorithm for Codebook Design

Let $\{Y_1, Y_2, ..., Y_p\}$ denote the training sequence and $C(0) = \{C_1^0, C_2^0, ..., C_M^0\}$ denote the initial codebook. At the m-th iteration, encode Y_m using codebook C(m-1). Let $\beta_m = Y_m - C_L^{m-1}$ denote the quantization error vector where C_L^{m-1} is the code vector closest to Y_m in C(m-1). The new codebook C(m) is obtained by replacing C_L^{m-1} in C(m-1) with

CH2673-2/89/0000-1747 $1.00 © 1989 IEEE

$$C_L^m = C_L^{m-1} - \mu \frac{\partial}{\partial C_L^{m-1}} d_1\left(Y_m, C_L^{m-1}\right) \tag{4}$$

$$= C_L^{m-1} + \mu \operatorname{sign}\left\{\beta_m\right\} , \tag{5}$$

where sign {•} is a vector consisting of the signum function of the corresponding elements of {•}, and μ is a small convergence constant. Note that the implementation does not require multiplications or median calculations.

Proof of Convergence

We will first show that the algorithm converges to the optimal code vector if the codebook contains only one vector. When the codebook contains M vectors we will show that the sequence of codebook vectors converge to a set that consists of the centroids of M subsets of the training sequence, each of which contains all the training vectors that got mapped into the corresponding code vector index. Finally, we will show that there exists a one-to-one mapping from the above set to a set of optimal code vectors, and this mapping is such that the long-term average of the distortion of the training vectors produced during codebook design will be at most $\frac{\mu k}{2}$ larger than the average distortion produced by encoding the training vectors with elements from the optimal codebook chosen using the mapping.

Note that the algorithm in Eq. 5 is very similar to the sign algorithm [2,5] employed in adaptive filtering. In fact, the convergence analysis here is an adaptation of the one in [2] to our particular situation. The only assumption that we will make is that as the number of training vectors becomes very large, the number of training vectors that are mapped into code vectors corresponding to each index (i = 1, 2, ..., M) will also become very large. This will eliminate the possibility that only part of the codebook is used during the algorithm and the optimization is effectively done for a smaller-sized codebook than what is desired.

To start the analysis, assume that the codebook consists of only one vector. Let C_1^m denote the code vector after the m-th iteration. Let $C_{opt,1}$ denote the centroid of the training sequence. Also, define the misalignment vector at time m as

$$V_1^m = C_1^m - C_{opt,1} . \tag{6}$$

Now,

$$C_1^m = C_1^{m-1} + \mu \operatorname{sign}\{\beta_m\} . \tag{7}$$

Subtracting $C_{opt,1}$ from both sides of Eq. 7, we get

$$V_1^m = V_1^{m-1} + \mu \operatorname{sign}\{\beta_m\} . \tag{8}$$

Taking the squared norm of both sides gives

$$\left\|V_1^m\right\|^2 = \left\|V_1^{m-1}\right\|^2 + \mu^2 k + 2\mu \left(V_1^{m-1}\right)^T \operatorname{sign}\left\{\beta_m\right\}, \tag{9}$$

where $(\bullet)^T$ denotes the transpose of (•). Note that

$$V_1^{m-1} = C_1^{m-1} - C_{opt,1} = \beta_{opt,m} - \beta_m , \tag{10}$$

where $\beta_{opt,m}$ is defined as the optimum quantization error in quantizing Y_m. Then

$$\left\|V_1^m\right\|^2 = \left\|V_1^{m-1}\right\|^2 + \mu^2 k + 2\mu \beta_{opt,m}^T \operatorname{sign}\left\{\beta_m\right\} \\ -2\mu \beta_m^T \operatorname{sign}\left\{\beta_m\right\} . \tag{11}$$

Recognizing that

$$\beta_m \operatorname{sign} \beta_m = d_1\left(Y_m , C_1^{m-1}\right) \tag{12}$$

is the distortion when Y_m is encoded using C_1^{m-1} and

$$\beta_{opt,m}^T \operatorname{sign} \beta_m \le \beta_{opt,m}^T \operatorname{sign} \beta_{opt,m} = d_1\left(Y_m , C_{opt,1}\right) , \tag{13}$$

and denoting the distortion in Eq. 12 by e(m) and that in Eq. 13 by $\varepsilon_{opt}(m)$, we get the following result:

$$\left\|V_1^m\right\|^2 \le \left\|V_1^{m-1}\right\|^2 + \mu^2 k + 2\mu\, \varepsilon_{opt}(m) - 2\mu\, e(m). \tag{14}$$

Iterating further, we obtain

$$\left\|V_1^m\right\|^2 \le \left\|V_1^0\right\|^2 + \mu^2 mk - 2\mu \sum_{i=1}^{m} e(i) + 2\mu \sum_{i=1}^{m} \varepsilon_{opt}^{(i)} \tag{15}$$

Since $\left\|V_1^m\right\|^2 \ge 0$, the above implies that

$$\frac{1}{m} \sum_{i=1}^{m} e(i) \le \frac{1}{m} \left\|V_1^0\right\|^2 + \frac{\mu k}{2} + \frac{1}{m} \sum_{i=1}^{m} \varepsilon_{opt}(i) . \tag{16}$$

Taking the limit as m goes to infinity,

$$\limsup_{m \to \infty} \frac{1}{m} \sum_{i=1}^{m} e(i) \le \zeta_{min} + \frac{\mu k}{2} , \tag{17}$$

where ζ_{min} is the average distortion produced by the optimal codebook. The above implies that if the codebook contains only one vector, the long-term average of the distortion produced by the gradient method exceeds the optimum average distortion by at most $\frac{\mu}{2}k$. If μ is small, the performance of this single code vector will be very close to that of the optimal code vector.

Now, consider a situation where we have to design a codebook with M code vectors. Let TS(i) denote the subset of the training sequence that gets mapped to the i-th code vector during codebook design. Obviously, the union of all TS(i)'s is the whole training sequence. Also note that the i-th code vector is trained only on TS(i), and therefore by the previous result, the i-th code vector converges to the centroid of TS(i), say $C_{c,i}$ in the same sense as in Eq. 17. This implies that the codebook sequence converges to a codebook that consists of the vectors $\{C_{c,i} ; i=1,2,...,M\}$.

Let CB(m) denote a vector of Mk elements formed by concatenating the elements of C(m), i.e.,

$$CB(m) = \left\{ \left(C_1^m\right)^T , \left(C_2^m\right)^T ,..., \left(C_M^m\right)^T \right\}^T . \tag{18}$$

Also, let C_{opt} denote another vector of Mk elements formed by the elements of an "optimal" codebook that minimizes the total distortion in encoding the training sequence. Note that any codebook that minimizes the total distortion will suffice. Furthermore, the ordering of the M optimal code vectors in C_{opt} can be arbitrary. Finally, let

$$V(m) = CB(m) - C_{opt} \tag{19}$$

As before, let C_L^{m-1} denote the closest vector in C(m-1) to Y_m. Define an Mk vector G(m) as

$$G(m) = \left\{0, 0, ..., \beta_m^T, 0 ... 0\right\}^T . \tag{20}$$

That is, G(m) is a vector of zero elements, except for the positions corresponding to the codebook element in C(m-1) that is closest to Y_m.

From Eq. 5 and the definitions above,

$$CB(m) = CB(m-1) + \mu \operatorname{sign} \{G(m)\} \tag{21}$$

where sign {0} is taken to be zero. Substracting C_{opt} from both sides and taking the squared norm of both sides, we can show that

$$\left\|V(m)\right\|^2 = \left\|V(m-1)\right\|^2 + \mu^2 k \\ + 2\mu \left\{C_L^{m-1} - C_{opt,L}\right\} \operatorname{sign}\left\{\beta_m\right\} . \tag{22}$$

The last-term comes from the fact that every other element of G(m) is zero. Also, $C_{opt,L}$ denotes the L-th k-tuple in $C_{opt}(m-1)$.

Note that

$$C_L^{m-1} - C_{opt,L} = \left(Y_m - C_{opt,L}\right) - \left(Y_m - C_L^{m-1}\right) = \varphi_m - \beta_m , \tag{23}$$

where φ_m is the quantization error vector when Y_m is encoded using $C_{opt,L}$.

Proceeding as before, substitution of Eq. 23 in Eq. 22 yields

$$\left\|V(m)\right\|^2 \leq \left\|V(m-1)\right\|^2 + \mu^2 k - 2\,\mu\, e(m) + 2\,\mu\,\varepsilon(m) , \tag{24}$$

where $\varepsilon(m)$ corresponds to the encoding distortion $d_1(Y_m, C_{opt,L})$. This inequality when iterated m times gives the following result:

$$\frac{1}{m}\sum_{i=1}^{m} e(i) \leq \frac{\left\|V(0)\right\|^2}{2\,\mu m} + \frac{\mu}{2}\,k + \frac{1}{m}\sum_{j=1}^{m} \varepsilon(i) \ . \tag{25}$$

Taking the limit as m goes to infinity,

$$\limsup_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} e(i) \leq \zeta + \frac{\mu}{2}k , \tag{26}$$

where ζ is the average distortion produced by mapping the training sequence into the optimal codebook. The mapping is such that whenever a training vector Y_m is closest to the i-th code vector of the codebook C(m−1), it is mapped into the i-th k-tuple of the Mk vector C_{opt}. Even though this may not in general be the optimal mapping, the above result shows that the algorithm converges and that there is a one-to-one mapping from the codebook being designed and the optimal codebook such that the long-term average of the distortion during codebook design exceeds that due to the above mapping into the optimal codebook by at most $\frac{\mu}{2}$k. Note that the ordering of the optimal code vectors in C_{opt} was arbitrary. In particular, if we choose the above one-to-one mapping to be the "best" possible one that minimizes the long-term average of the distortion produced by the mapping on the *training sequence*, the long-term average of the distortion produced by the codebook during the design process will still not exceed the performance of the optimal codebook and the one-to-one mapping by more than μk/2. Even though there is a possibility that this mapping may be very different from the optimal one, all the experiments that we have done have produced results that are comparable to that of the LBG algorithm.

III. CODEBOOK DESIGN FOR PIECE-WISE LINEAR DISTORTION MEASURES

Since in many data compression systems the ultimate objective is to produce quantized signals with a minimum amount of subjective distortion rather than produce quantized signals that minimize certain quantitative distortion measure, we must keep in mind that the L_1 distortion measures may not be the most suitable one for many applications. In order to overcome this limitation, we propose the use of piece-wise linear approximations for distortion measures that give rise to more complex encoding procedures. The functional form of the distortion measure is as given in Eqs. 2 and 3. As stated earlier, if we choose the slopes α_i's of the piece-wise linear functions to be integer powers of two, the encoding *can be done with only bit shifting*, additions, and comparisons.

The codebook design algorithm of Eq. 5 can be easily extended to this case. We will use the same notations as in the previous section. Let β_m be the quantization error vector that is due to *the code vector that minimizes the piece-wise linear distortion measure* in Eq. 2. Let $\alpha_i(m)$ be the slope of the piece-wise linear function in Eq. 3 that corresponds to the i-th element of β_m. Define a diagonal matrix Λ(m) as

$$\Lambda(m) = \operatorname{diag}\left\{\alpha_1(m), \alpha_2(m), ..., \alpha_k(m)\right\} \ . \tag{27}$$

Then the update equation for the gradient-descent codebook design algorithm becomes

$$C_L^m = C_L^{m-1} + \mu\,\Lambda(m)\,\operatorname{sign}\left\{\beta_m\right\}. \tag{28}$$

Again, note that the codebook update algorithm can also be *implemented using zero multiplications.*

IV. APPLICATION TO PREDICTIVE VECTOR QUANTIZATION OF IMAGES

Researchers have found that predictive vector quantization algorithms outperform direct vector quantization much as linear predictive coding algorithms work better than PCM in scalar quantization schemes [1,6,9]. There are two main reasons why predictive vector quantization algorithms are attractive: First, the prediction error sequence will in general have a smaller dynamic range than the *original input waveform, and the prediction error sequence can*, in most cases, be done more effectively than the input waveform itself. The second reason addresses a problem that is typical of all vector quantization algorithms. Since the codebook is designed for a training sequence that is representative of the input waveforms to the quantizer, any variations in the structure of the input waveforms from that of the training sequence will degrade the performance of the vector quantizer. The prediction error sequences will have far less structure in them than the waveforms themselves, and therefore a vector quantizer working on prediction error sequences will be more robust to variations in the structure of the input waveforms.

In this section we apply the results of the previous section to predictive vector quantization of digital images. The block diagram of the vector predictive vector quantizer is shown in Fig. 2. In the figure, x(n,m) corresponds to the input image and B is an estimate of the mean value of the image. The rest of the notations are self-explanatory. The predictor coefficients are all chosen to be (possibly negative) integer powers of two and therefore the *structure is truly multiplication-free.*

Note that the prediction filter predicts the current input vector using previous input vectors and the resulting error sequence is vector quantized. The codebook is designed from a set of training images. The predictor coefficients for the images are first estimated. Starting with an initial codebook, the mean-removed training image is quantized, and the codebook is updated each time after an input vector is quantized as described in Section III.

We now present the result of an experiment that demonstrates the good subjective quality of the predictive vector quantizer employing the piece-wise linear distortion measure. The original image that is *quantized is entitled "woman" and is shown in Fig. 3.* It consists of 512 x 512 pixels with eight-bit resolution. The predictor equations are given in Table I. The notations employed in Table I are shown in Fig. 4. Note that these equations are the same as those used in [9]. The image presented in Fig. 5 was obtained using 4 x 4 vectors and 512 code vectors resulting in an effective data rate of 0.5625 bits/pixel. The codebook was trained on the "woman" image itself. The choice of various parameters for the experiments, including the piece-wise linear function that was employed is given in Table II. We can see that the quantizer performs very well even though its implementation requires no multiplications.

V. CONCLUDING REMARKS

The usefulness and effectiveness of vector quantizers in waveform compression is beyond doubt. However, their computational complexity has made their implementation a costly proposition. The purpose of this paper was to propose *an approach to multiplication-free* vector quantization, and then demonstrate its usefulness in predictive vector quantization of digital images. Toward this end, we first presented a gradient-based codebook design algorithm for vector quantizers that required no multiplications or median computations. With very mild assumptions, we showed that the algorithm converges, and the long-term average of the encoding distortion can be made arbitrarily close to that due to a certain mapping of the training sequence into the optimal codebook. We extended the approach to the case when a piece-wise linear distortion measure is employed. This method was applied to predictive vector quantization of images and *experiments using this technique have produced images with good* subjective quality. The ease of implementation and the quality of encoding associated with the methods suggest that they are very viable candidates in waveform coding applications involving vector quantization.

REFERENCES

1. V. Cuperman and A. Gersho, "Vector Predictive Coding of Speech at 16 kbits/s," *IEEE Transactions on Communications,* Vol. COM-33, No. 7, pp. 685-696, July 1985.

2. A. Gersho, "Adaptive Filtering with Binary Reinforcement," *IEEE Transactions on Information Theory,* Vol. IT-30, No. 2, pp. 191-199, March 1984.

3. Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications,* Vol. COM-28, No. 1, pp. 84-95, January 1980.

4. J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding," *Proceedings of the IEEE*, Vol. 73, No. 11, pp. 1551-1588, November 1985.

5. V. J. Mathews and S. H. Cho, "Improved Convergence Analysis of Stochastic Gradient Adaptive Filters Using the Sign Algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 4, pp. 450-454, April 1987.

6. V. J. Mathews, R. W. Waite, and T. D. Tran, "Image Compression Using Vector Quantization of Linear (one-step) Prediction Errors," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 733-736, Dallas, Texas, April 6-7, 1987.

7. N. M. Nasrabadi and R. A. King, "Image Coding Using Vector Quantization: A Review," *IEEE Transactions on Communications*, Vol. COM-36, No. 8, pp. 957-971, August 1988.

8. T. D. Tran, V. J. Mathews, and C. K. Rushforth, "A Baseband Residual Vector Quantization Algorithm for Voiceband Data Signal Compression," to appear in *IEEE Transactions on Communications*, 1989.

9. B. Vasudev, "Predictive VQ Schemes for Gray Scale Image Compression," *Proceedings of GLOBECOM '87*, pp. 436-441, Tokyo, Japan, November 1987.

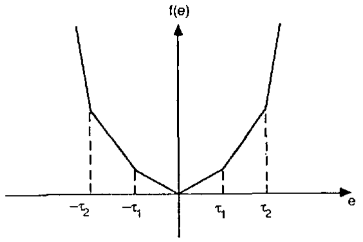


Figure 1. An example of a function that describes the piece-wise linear distortion measure.

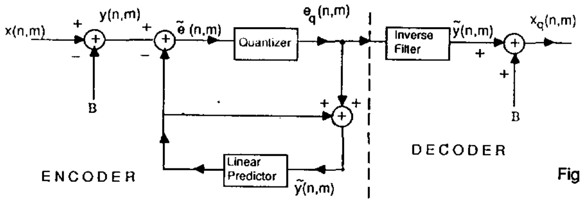


Figure 2. Block diagram for predictive vector quantization of images.

Table 1. Predictor equations for the predictive vector quantizer. ~ denotes predicted quantities.

$$\tilde{h} = \tilde{f} + (\tilde{p} + A5)/4 - A7/2 \qquad \tilde{f} = (A3 + A7)/2$$

$$\tilde{d} = (\tilde{h} + A5)/2 \qquad \tilde{b} = (\tilde{f} + A3)/2$$

$$\tilde{m} = (\tilde{n} + A9)/2 \qquad \tilde{e} = (\tilde{f} + A7)/2$$

$$\tilde{g} = (\tilde{f} + \tilde{h})/2 \qquad \tilde{a} = (\tilde{f} + A1 + A3 + A7)/4$$

$$\tilde{j} = (\tilde{f} + \tilde{n})/2 \qquad \tilde{c} = (\tilde{f} + A3 + A5)/3$$

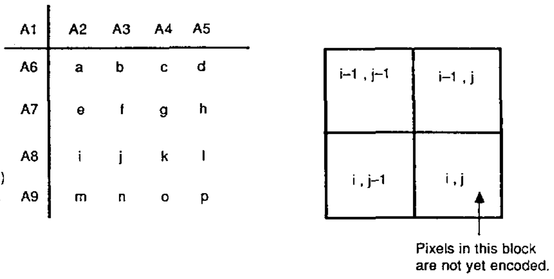
$$\tilde{l} = (\tilde{h} + \tilde{p})/2 \qquad \tilde{i} = (\tilde{f} + A7 + A9)/3$$

$$\tilde{o} = (\tilde{n} + \tilde{p})/2 \qquad \tilde{p} = \tilde{f} + (A5 + A9)/4 - A1/2$$

$$\tilde{k} = (\tilde{f} + \tilde{h} + \tilde{n} + \tilde{p})/4 \qquad \tilde{n} = \tilde{f} + (\tilde{p} + A9)/4 - A3/2$$

Table 2. Parameters of the predictive vector quantizer

Range of |e|	Slope	Intercept
0.0 - 1.0	1.0	0.0
1.0 - 7.0	8.0	-7.0
7.0 - infinity	32.0	-175.0



A1	A2	A3	A4	A5
A6	a	b	c	d
A7	e	f	g	h
A8	i	j	k	l
A9	m	n	o	p

(a) Pixel notation

i-1 , j-1	i-1 , j
i , j-1	i , j

Pixels in this block are not yet encoded.

(b) Block notation

Figure 4. Block and pixel notations for the predictive vector quantizer. A1-A9 are decoded pixels in blocks already encoded. a - p are predicted using A1-A9.

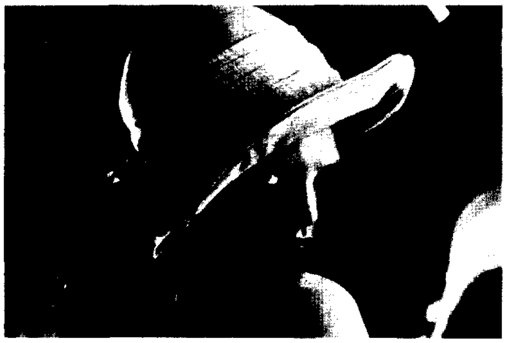


Figure 3. Original "woman" image

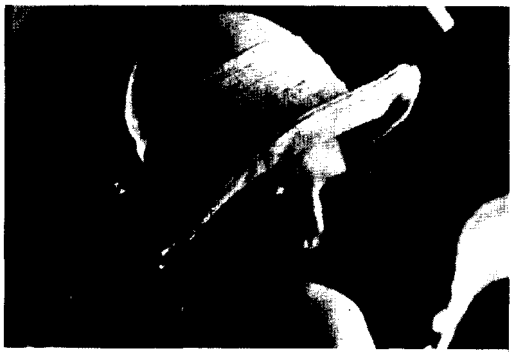


Figure 5. Encoded "woman" image. Average distortion = 48.59