TR-91-014

INTERACTIVE DCT IMAGE COMPRESSION FOR PHOTO STORAGE

INTERACTIVE DCT IMAGE COMPRESSION FOR PHOTO STORAGE

Mark C. K. Yang
Institute of Information Science
Academia Sinica, and
Department of Statistics, University of Florida
Gainesville, Florida 32611, USA
and Jun S. Huang
Institute of Information Science
Academia Sinica
Taipei, Taiwan 11529
Republic of China

INTERACTIVE DCT IMAGE COMPRESSION FOR PHOTO STORAGE

Mark C. K. Yang and Jun S. Huang
Institute of Information Science, Academia Sinica

ABSTRACT – Due to the lack of a satisfactory description for the human visual system, high compression is very difficult to reach using the traditional optimization procedure. A DCT data compression system with human interaction is proposed. Human quality judgement is embedded into an optimization problem on weighted least mean squares. Codebook construction connected to this method is also discussed. This system should be useful in photograph, fingerprint, and road map storage and retrieval when the processing time in the compression stage is not restricted too much .

Keywords: Image compression, Discrete cosine transform, Picture enhancement.

1. INTRODUCTION

Image compression by the discrete cosine transform (DCT) has recently received considerable attention. The main idea of DCT image compression is to divide an image into many equal sized square blocks, take a DCT on each block, and compress the block by reducing or eliminating some of the DCT coefficients. More specifically, in any given block, let the grey level of the (i, j)th pixel be x(i, j), i, j=0, 1, ..., n-1. Then the DCT of this block is defined by

$$F(u, v) = \frac{4c(u)c(v)}{n^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}x(i, j)\cos\frac{(2i+1)u\pi}{2n}\cos\frac{(2j+1)v\pi}{2n}, \qquad (1)$$

for u, v=0,1,..., n-1, where

$$c(y)=\begin{cases} 1/\sqrt{2} & \text{if } y=0 \\ 1 & \text{elsewhere.} \end{cases}$$

The F(u, v)'s are referred to as the DCT coefficients or the AC components except for F(0, 0), which is called the DC component. The squares $F^2(u, v)$ are called the DC or AC energy depending on whether (u, v)=(0, 0) or (u, v) $\neq$ (0, 0). The inverse transform exists and it can be shown to be

$$x(i, j) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} c(u)c(v)F(u, v)\cos\frac{(2i+1)u\pi}{2n}\cos\frac{(2j+1)v\pi}{2n}, \qquad (2)$$

for i, j= 0, 1, ..., n-1.

From practical experience, it has been found that most of the AC energy is concentrated on the upper right corner in the u, v=0, 1, ..., n-1 square. Thus, storing (or transmitting) a few upper right corner F(u, v)'s is enough to recover the original image with little distortion. There are two basic methods in determining the way of compressing F(u, v), namely, the codebook method[1] and the zigzag scanning method[2-3]. Since the codebook method is related to vector quantization (VQ), it should have a better chance to reach high compression when the codebook is well constructed and used. Hence the codebook method is the main concern of this paper, though the zigzag scanning method is also used for the purpose of comparison.

A codebook consists of many bit assignment maps. A suitable bit-map is chosen for each block. Table 1 provides the first 9 maps of a codebook with 32 maps. For example, if map 1 is selected, only 8 bits are used to store the DC component of that block. All the AC components are ignored and they are assumed to be 0 when inverse DCT is used to recover the block. On the other hand, if map 2 is selected, 8 bits are used for F(0, 0), 2 bits for F(0,1) and F(1, 0), 1 bit for F(0, 2) and F(1, 1), and all the rest components are ignored. Similar maps can be found in Jain[4], Chen and Smith[1], Wu and Burge[5], and Mitchell and Tabatabai[6]. The methods of map construction include energy distribution[1] and pattern recognition[5]. The numbers of maps in the cited papers are small. If high compression is the main purpose, then the map number has to be increased. This is one of the main problems in the codebook method. Heuristic methods for VQ such as those suggested by Linde, et al.[7], Equitz[8], and Hang and Haskel[9] are available for codebook building. A new method of codebook construction and updating is discussed in this paper.

Once a bit-map codebook has been constructed, one map has to be assigned for each block to compress or store it in an optimal way. This again is not an easy matter when the number of maps is large. Usually, only the nearly optimal solution can be found (see Shoham and Gersho[10]). Let S represent the set of all the maps in the codebook and the maps be ordered as 1, 2, ..., T, with b(t) denoting the number of bits required to use bit-map t. Without loss of generality, we let $b(1) \leq b(2) \leq ... \leq b(T)$. If the blocks are numerically ordered as 1, 2, ..., M, then an optimal allocation is to find a bit-map t_k for block k such that the total discrepancy

$$D = \sum_{k=1}^{M} d_k(t_k) \qquad (3)$$

is minimized subject the channel capacity constraint

$$\sum_{k=1}^{M} b(t_k) = B, \tag{4}$$

where $d_k(t_k)$ denotes the discrepancy of using map t_k to compress block k and B is the total number bits available for image storage or compression. Due to mathematical simplicity, the discrepancy function $d(\cdot)$ in (3) is usually the mean square error(MSE) defined as

$$\sum_{u=0}^{n-1} \sum_{v=0}^{n-1} (F(u,v) - \hat{F}(u, v; t))^2, \tag{5}$$

where $\hat{F}(u, v; t)$ is the stored value for F(u, v) when given map t is assigned for compression. However, some empirical studies have shown that the MSE criterion does not always agree with the human visual quality assessment. Nill[11] has summarized some previous results and used the following weight function to imitate human vision, i.e., (5) is modified as

$$\sum_{u=0}^{n-1} \sum_{v=0}^{n-1} H(u, v)(F(u,v) - \hat{F}(u, v; t))^2, \tag{6}$$

where

$$H(u, v) = (0.2+0.45r)e^{-0.18r}, \quad r=\sqrt{u^2 + v^2}\,.$$

We will call H(u, v) the local human vision compensation function for the obvious reason that (6) works only within a block while sometimes the discrepancy between (5) and human vision happens at a global level. For example, in a human photograph, the blocks that contain important features such as eyes, wrinkles, or scars, are more important than those that contain hair, ears, or the background. The central right position of Figure 6 in Wang and Goldberg[12] provides an illustration. Even at the low compression rate of 13.5:1, the right eye of the subject is distorted. But all the rest of the face including the left eye is of perfect quality. A global weight function is apparently necessary. Thus, an ideal form of (3) would be

$$D \equiv \sum_{k=1}^{M} W_k \cdot E_k(t_k), \text{ with} \tag{7}$$

$$E_k(t_k) \equiv \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} H(u, v)(F_k(u,v) - \hat{F}_k(u, v; t_k))^2,$$

where W_k is the weight function, $F_k(u, v)$ is the DCT component, and $\hat{F}_k(u, v; t_k))$ is the compressed component by bit-map t_k for block k. To determine this weight function by computer alone seems impossible at the present state of the art. Thus, in the proposed system human interaction is used to determine the weights. Though the human interaction part is simple, it is too slow for real-time data compression such as television or a teleconference. However, for non-real-time image compression such as land and sea scapes, maps, paintings, and photos, this method should be useful. In particular, when large number of photos are to be stored for security identification, processing time can be more relaxed at the agency that collects, inspects, and stores them. Only at the moment when a police or a customs station needs a quick retrieval of those pictures, time for transmission and de-compression becomes a major constraint. Thus, we ask for high compression ratio, but allow more time in the compression process. Similar situations occur in fingerprint and road map storage.

II. METHODS AND RESULTS

A) Map Allocation

Let a codebook be given. Then one of the most efficient methods in solving (7) and (4) is the generalized Lagrange multiplier method proposed by Everett[13] and adapted to bit allocation in [10]. The idea is that (7) and (4) can be solved in an individual block by trying to find a real number λ and a map, say $t_k^* \epsilon S$, for block k such that

$$W_k \cdot E_k(t_k^*) + \lambda\, b(t_k^*) = \min_{t \epsilon S} \{ W_k \cdot E_k(t) + \lambda\, b(t)\}, \qquad (8)$$

with

$$B \simeq B(\lambda) \equiv \sum_{k=1}^{M} b(t_k^*).$$

Then the following theorem can be shown.

Theorem 1. (Generalized Lagrange multiplier method, Theorem 1 in [10]) The $\{t_k^*\}$ defined in (8) is the optimal solution for (7) and (4) if B happens to be equal to $B(\lambda)$ for some λ.

Unfortunately, in most cases it is not possible to find such a λ. Let the optimal MSE for this λ be denoted by

$$D(\lambda) = \sum_{k=1}^{M} W_k \cdot E_k(t_k^*). \tag{9}$$

If for any two λ's, say, $\lambda_1<\lambda_2$, such that $B(\lambda_1)<B<B(\lambda_2)$, then the optimal value of D for the total bit constraint B is bounded by $D(\lambda_1)$ and $D(\lambda_2)$. Thus, if the gap between $D(\lambda_1)$ and $D(\lambda_2)$ is not too large, then using the solution for λ_1 is not too far from the optimal solution, i.e., to replace B by the less bit contraint $B(\lambda_1)$ and accept the MSE $D(\lambda_1)$. Though some improvements are still possible, the optimality cannot be guaranteed. A procedure of finding the smallest gap between $B(\lambda_1)$ and $B(\lambda_2)$, or equivalently the smallest gap between $D(\lambda_1)$ and $D(\lambda_2)$, is described in [10]. However, due to the necessity of repeated computation of (7) and (4) during the weight modification process to be described in this paper, it is easier if we first identify a subset set of S for the solutions t_k^* independent of W_k in each block.

Define the backward slope

$$\beta_k(t) = \max_{0 \le t' < t} \frac{E_k(t) - E_k(t')}{b(t) - b(t')} \tag{10}$$

and the forward slope

$$\phi_k(t) = \min_{t < t' \le T} \frac{E_k(t') - E_k(t)}{b(t') - b(t)}\ , \tag{11}$$

with the convention $E_k(0) \equiv \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} F^2(u, v)$, $\beta_k(0) = -\infty$ and $\phi_k(T)=0$. Then the feasible set for block k is defined as:

$\Psi_k = \{t \epsilon S \mid \beta_k(t) \le \phi_k(t)\} \equiv \{t_k^1, t_k^2, \cdots, t_k^{n_k}\}$, with the order

$b(t_k^1) \le b(t_k^2) \le \cdots \le b(t_k^{n_k})$.

The following lemma combines β and ϕ.

Lemma 1. In the feasible set Ψ_k,

$$\phi(t_k^i) = \beta(t_k^{i+1}) = \frac{E(t_k^{i+1}) - E(t_k^i)}{b(t_k^{i+1}) - b(t_k^i)}\ . \tag{12}$$

A proof of this lemma as well as all the other proofs in this section are given in Appendix A. Thus, we may choose one set of slopes to represent β and ϕ. Let us choose the β's and define

$$\mathfrak{B}_k = \{\beta(t_k^1), \beta(t_k^2), \cdots, \beta(t_k^{n_k})\}.$$

The generalized Lagrange multiplier method is now equivalent to the following theorem.

Theorem 2. Any solution $\{t_k^*\}$ of the generalized Lagrange multiplier method (in S) must satisfy $t_k^* \in \Psi_k$ for all k. Moreover, the solution for λ ($\lambda<0$) can be identified as the map $t_k^j \in \Psi_k$ satisfying

$$\beta_k(t_k^j) \le \lambda \le \beta_k(t_k^{j+1}). \tag{13}$$

Also, from the proof, it is obvious that W_k can be embedded into λ, or:

Corollary. The feasible sets are independent of $\{W_k\}$ in (7).

Because the β's in $\mathfrak{B}_k$ are monotonically increasing, the solution is very easy to find for any given λ. Actually, the " $\le$ " signs in (13) are for the convenience of theorem proving. If in any time the equal sign holds, we should take the equal sign, i.e., (13) becomes $\lambda=\beta_k(t_k^*)$. However, for most λ, (13) is with two strict inequalities, $\beta_k(t_k^j) < \lambda < \beta_k(t_k^{j+1})$. Also, finding the bounds λ_1 and λ_2 which minimize the gap

$$\Delta_B = B(\lambda_2) - B(\lambda_1), \quad B(\lambda_2) < B < B(\lambda_1), \tag{14}$$

is also easy using the feasible sets. Let

$$\mathfrak{B} = \bigcup_{k=1}^{M} \mathfrak{B}_k \equiv \{ \beta_1 \le \beta_2 \le \cdots \le \beta_m \}. \tag{15}$$

Since there are no maps that can produce a slope in a feasible set to fill into the gap between any two adjacent β's in $\mathfrak{B}$, we have:

Lemma 2. For any $\lambda \epsilon [\beta_j, \beta_{j+1})$, $(\beta_j, \beta_{j+1} \epsilon \mathfrak{B})$, the generalized Lagrange multiplier solutions are the same.

The consequence of this lemma is that there is no need to check all the λ's. Only the values in $\mathfrak{B}$

need to be evaluated. Hence, the minimum gap in (14) can be easily found by the method of bisection with at most $\log_2 m$ steps. Since only simple comparisons are necessary in finding the solution in Theorem 1, the bisection procedure is extremely fast. It can also be seen from this lemma that the chance of finding a λ such that $B(\lambda)$ equals a given B is very small. Thus, a good codebook should always have small gaps Δ_B in a target range of B. This concept will be used later in codebook construction.

Some improvement is possible by utilizing the leftover bits Δ_B in (14). The improvement is to replace some maps in the feasible set by the maps in S. A simple way to do this is to find a particular block map combination (k^*, t^*) which minimizes

$$\alpha(k,t) \equiv \frac{W_k[\, E_k(t) - E_k(t_k^*)]}{b(t) - b(t_k^*)}, \qquad (15)$$

for all $k=1,...,M$, and all $t\epsilon S$ with $0< b_k(t) - b_k(t_k^*) \le \delta_B \equiv B - B(\lambda_1)$. Since $\{t\epsilon S \mid 0<b_k(t) - b_k(t_k^*) \le \delta_B\}$ is usually a small set, finding (k^*, t^*) is again easy. If $\alpha(\, k^*, t^*)<0$, then t_k^* should be replaced by t^*. It is easy to see that the new map allocation has total number bits $B_1 \equiv B + b_{k^*}(t^*) - b(t_{k^*}^*) \le B$ with MSE

$$D(\lambda_1) + W_{k^*}[\, E_{k^*}(t^*) - E_{k^*}(t_{k^*}^*)] < D(\lambda_1).$$

The procedure can be repeated with the new $\delta_B = B - B_1$ until $\alpha(\, k^*, t^*) \ge 0$ or $\{t\epsilon S \mid 0<b_k(t) - b_k(t_k^*) \le \delta_B\}$ becomes empty for all k.

B) Global Weight Determination

To determine W_k in (7), first let all the weights be 1. Using various λ's and the optimal map assignment scheme (13) and (15) in (A), many compressed images can be constructed. The range of λ should be large enough that some high quality images are generated. Choose the λ, say λ_o, such that the total number of bits $B(\lambda_o)$ is close to B as a base for block quality comparisons. If any block in this image is not satisfactory, search for the smallest λ ($\lambda<\lambda_o$), say λ_1, that gives a satisfactory compressed image for this block. Assign λ_o/λ_1 as the weight to this block. Similarly, for an unimportant block, search for the smallest $\lambda>\lambda_o$ that can still produce a satisfactory compressed block. Let the weight of this block again be λ_o/λ_1. A new compressed image based on the newly assigned weights can then be constructed by (13) and (15) with the β's adjusted by the new weights. It can be easily seen that this method should produce an acceptable image if the number of changes is not large.

If the number of changes is large, more iterative steps need to be taken. The new weight in the next round is $W=W'\cdot(\lambda_0'/\lambda_1)$, where the primes denote the weight and λ in the previous iteration. The process stops when the image is satisfactory. Of course, there is a possibility that the image can never be satisfactory due to the lack of enough bits. Then a compromise has to be reached based on the relative importance of the blocks.

C) Codebook Construction

As mentioned in the introduction, there are cluster algorithms for codebook construction using large training samples. Also, a codebook can be constructed by intuitive study of the image [5], or by the DCT patterns from simulated figures such as Figure 1 which emphasizes the boundaries. In this paper, only map growth and elimination based on the gap criterion described in (A) is discussed. Suppose a initial codebook has been constructed. Then in the next training set, a map is eliminated if it is not selected often enough. Thus, a map can be retained in the updated codebook only when it is selected in feasible sets and is used often. The rate of elimination is somewhat subjective, depending on the number of maps. We used 1/4 of the total maps.

Eliminated maps are replenished by maps interpolated between large gaps in (13). More precisely, suppose $\Delta_B=b(t)-b(t')$ for two maps t and t' in S. If t and t' happen to be two adjacent maps ($t<t'$) in a feasible set of a block, say block k, then the slope $\beta=\beta_k(t)$ is in $\mathfrak{B}$ and for any $\lambda\epsilon[\beta, \beta')$ will produce a gap Δ_B, where β' is the element next to β in $\mathfrak{B}$. For two maps i and j ($i<j$) in S, define

$$\omega(i,j)=[\sum_{k=1}^{M} U_k(i,j)][b(j)-b(i)],$$

$$U_k(i,j)=\begin{cases} 1 & \text{if i ad j are adjacent in } \Psi_k \\ 0 & \text{elsewhere.} \end{cases}$$

A new map is interpolated between maps i and j according the order of $\omega(i, j)$. The priority are assigned from the highest values of $\omega(i,j)$ to the lower. The method of interpolation is given in Appendix B.

An important point, that seems to have not been mentioned in the literature, is that a good codebook depends not only on the DCT patterns of the training sample but also on the compression rate. Thus, the target range of the compression rate B (or equivalently λ) needs to be given in

codebook construction.

3. EXPERIMENTAL RESULTS

Eight human face pictures are used to test the interactive system. Each picture occupies 240x256 pixels with 8 bpp gray levels. Table 2 gives some examples of existing human face compression. Though the quality assessment is highly subjective and the compression ratio depends on the background, it is fair to say that a compression ratio (CR) of 40 : 1 is difficult to reach by the existing methods. Thus, our compression goal is set around 40 : 1.

The block size for DCT is chosen as 16x16 and the Laplacian distribution is used for quantization, except the DC coefficient which is transmitted by 8 bits to reduce blocking effect. The codebook library is built by examining the all the DCT transforms of the 960 blocks from 4 pictures and the boundary samples in Figure 1 (a), where 15 boundaries are shown in each row with different gray levels of contrasts. The contrasts become more and more eminent from the left to the right. Figure 1 (b) gives the DCT for each of the 16x16 blocks. Note that some of the boundaries such as the one in the last several rows produce DCT with a long tail in the diagonal. We choose 32 maps for the codebook library. Due to the large volume, only the first 9 are shown in Table 1. Note that most of bits are allocated in the upper left corner similar to those found in many previous experiments [1, 5, 6]. Some of the elongated ones are observed from the boundary samples. Fig. 1 (c) and (d) are, respectively, the compressed Fig. 1 (a) by the codebook method and the zigzag method (see Appendix C for detail), both with CR=26. It can be seen that the quality of (c) is better, especially for rows 2, 3, 5, 8, 9, and 14, where the zigzag method becomes unacceptable at a much higher boundary contrast than the codebook method. For example, in row 2, column 6 is acceptable in (c), but not acceptable until column 9 in (d). The zigzag scanning cannot handle some boundaries efficiently because their DCT's do not follow well with the zigzag route.

After the initial maps have been constructed, maps are dropped and added according to the scheme discussed in §2.C by using 3 other pictures as training samples. The compression rate was set between $30-50$: 1 which corresponds to λ from -1 to -4. There was a considerable shift from the maps with a large number of bits to those with smaller numbers in the first training picture. The map patterns were basically stabilized when the next two pictures were used for codebook modification. The last of the eight pictures is used for compression which will be described later and shown in Figures 3 and 4.

In our interactive system, a typical compression output for a single given λ is given in Figure 2. It includes the original picture (a), the compressed picture (b), the blockwise DCT (c), and its block divisions (d). Also, with the output are the compression ratio, MSE, and the bit classification maps for all the blocks. In this particular picture, λ=6.0 and CR = 72.9, which is computed by the ratio of $8x15x16^3$ and the B in the classification map plus the bits to transmit the map number, i.e., $\log_2 32=5$ per block. It is apparent that this picture is compressed too much. Blocking effects and blurring are visible everywhere.

Fig. 3 is another codebook compressed picture, where (a) is the original, (b) and (c) are compressed images with λ=4.0, CR=48.0 and λ=3.0, CR=41.8, respectively, while (d) is a modified picture with CR=48.0 by our human interaction system. These pictures, like those compressed pictures with a similar CR in the literature [eg., 5, 14, 15, 16], require careful scrutiny to tell the quality differences. Here, picture (d) is better than (b) in the clarity at the eyes, lips, and teeth. Fig. 4 gives some comparisons with the zigzag compressed pictures. In this figure, the diagonal pictures (a) and (d) are the same as those in Fig. 3, while (b) has T=D=2 with CR=31.3, and (c) T=D=3 with CR=45.0. The blocking effects in both (b) and (c) are apparent.

4. CONCLUSION

In this paper, we presented an image compression system where human visual preference is mingled into the compression process by human intervention. Though it suffers from slowness, it can be useful in many application areas such as photo and map storage and retrieval.

APPENDICES

A) Proofs for §2.A

A.1. Proof of Lemma 1.

We prove this lemma by geometry. Since all the E's are within one block, the subscripts k are dropped. Also, t_i is replaced by t to simplified the notation.

Suppose $t \epsilon \Psi_k$. Then there exist a $t' > t$ and a $t'' < t$ (see Figure 5), such that

$$\phi(t) = \frac{E(t') - E(t)}{b(t') - b(t)}, \qquad (A.1)$$

and

$$\beta(t) = \frac{E(t) - E(t'')}{b(t) - b(t'')}. \qquad (A.2)$$

We will show that $t' \epsilon \Psi_k$ and adjacent to t. We first show that $\beta(t') = \phi(t)$.

Suppose $\beta(t') \neq \phi(t)$. Then by the definition of β, $\beta(t') > \phi(t)$. Then there must be a point $(b(t'''), E(t'''))$ on line the L2 below line BC(L1) such that

$$\beta(t') = \frac{E(t') - E(t''')}{b(t') - b(t''')}.$$

This violates the definition of (A.1) if $b(t''') \epsilon (b(t), b(t'))$, and violates (A.2) if $b(t''') \in (-\infty, b(t))$. Thus $\beta(t') = \phi(t)$. To show $t' \epsilon \Psi_k$, suppose not. Then there must exist a $(b(t'''), E(t'''))$ below L_1 with $t''' \epsilon (t', \infty)$. This violates $\phi(t)$ defined in (A.1). Thus $t' \epsilon \Psi_k$.

Suppose t' is not adjacent to t. Then there exists a point D or D′ in Figure 5 with its map $t''' \epsilon \Psi_k$. This is impossible because if it is at D, (A.1) is violated, or if at D′, then t''' cannot be in Ψ_k because the β based on D is larger than the ϕ based on D. Thus, (12) is proved.

A.2. Proof of Theorem 2.

Since W_k and $E_k(t)$ are always together, we drop W_k in the proof. Let t_k denote the solution for the generalized Lagrange multiplier method by (8). Then

$$E_k(t_k) + \lambda t_k \leq E_k(t) + \lambda t,$$

for all $t \neq t_k$, i.e.,

$$\frac{E_k(t) - E_k(t_k)}{t - t_k} \geq -\lambda, \text{ for all } t > t_k,$$

and

$$\frac{E_k(t) - E_k(t_k)}{t - t_k} \leq -\lambda, \text{ for all } t < t_k.$$

This is equivalent to

$$\beta_k(t_k) \leq -\lambda \leq \phi_k(t_k).$$

By the definition of Ψ_k, $t_k \epsilon \Psi_k$. Then by Lemma 1, (13) is proven.

B) Map Interpolation

The method of adding a map between two maps again follows the idea of Lagrange multipliers. If the map index t *were* a continuous variable, then the Lagrange multipliers solution for (3) and (4) would be

$$f_k(t) \equiv \frac{\partial d_k(t)/\partial t}{\partial b(t)/\partial t} = \lambda, \qquad \text{(A.3)}$$

for all k=1, ..., M, while λ is determined by (4). Note that the solution for (A.3) may not be unique if the derivatives $\{ f_k' \}$ are not monotonic functions. Many local minimums may be reached from (A.3). In order for $\{ f_k' \}$ to be monotonic, the second derivatives of the f_k need to always have the same sign. We will show that a convex combination of two maps can satisfy this requirement under an ideal condition. It is well known from quantization error curves(e.g., Max[17]), that the relation between MSE and the number of bits used to represent it is approximately MSE $\propto e^{-\text{bits}}$. Let the two maps be interpolated to be t_1 and t_2 and the number of bits used by them for compression for the coefficient at (u, v) be $t_1(u, v)$ and $t_2(u, v)$, respectively. A convex combination t_α of t_1 and t_2 uses

$$t_\alpha(u, v) = \alpha t_1(u, v) + (1-\alpha) t_2(u, v)$$

bits to compress F(u, v). Then the MSE for table t_α is approximately

$$E_k(t_\alpha) = K \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} F(u, v) e^{-t_\alpha(u, v)}, \qquad \text{(A.4)}$$

where K is the proportional constant. It can be easily shown that $\partial^2 E_k(t_\alpha)/\partial \alpha^2 > 0$. Actually, if $\beta(t_1) = \partial E_k(0)/\partial \alpha$ and $\phi(t_2) = \partial E_k(1)/\partial \alpha$, then all the newly created maps are in the feasible set Ψ_k if (A.4) is exact. Unfortunately, (A.4) is only an approximation, especially when $t_\alpha(u, v)$ has to be made as an integer.

C) ZIGZAG SCANNING

The zigzag scanning method follows the notation of [3], except we select the threshold value T equal to the feedback normalization factor D. The reason is that if we let x=0 if $|x/D| < 1$, then it is equivalent to letting x=0 if $|x| - T < 0$ when T=D. The Huffman code table for amplitudes and consecutive zeros are adapted from [3]. By changing T (or equivalently D), pictures under different compression rates can be obtained.

REFERENCES

[1] W. H. Chen and H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Commun.*, vol COM-25, pp. 1285-1292, 1977.

[2] J. Parsons and A. Tescher, "An investigation of m.s.e. contribution measures in transform image coding scheme," *SPIE Semin. Proc.* 66, 196-206, 1975.

[3] W. H. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, vol. COM-32, pp. 225-232, 1984.

[4] A. K. Jain, "Image data compression: a review," *Proc. IEEE*, vol 69, pp. 349-389, 1981.

[5] J. K. Wu and R. E. Burge, "Adaptive bit allocation for image compression," *Comput. Graphics Image Processing*, vol 19, pp. 392-440, 1982.

[6] O.R. Mitchell and A. Tabatabai, "Channel error recovery for transform image coding," *IEEE Trans. Commun.*, vol COM-29, pp. 1754-1762, Dec. 1981.

[7] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[8] W. H. Equitz, "A new vector quantization clustering algorithm," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-37, pp. 1568-1575, Oct. 1989.

[9] H. M. Hang and B. G. Haskell, "Interpolative vector quantization of color images," *IEEE Trans. Commun.*, vol. COM-36, pp. 465-470, Apr. 1988.

[10] Y. Shoham and A. Gersho, "Efficient bit allocation for arbitrary set of quantizers," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-36, pp. 1445-1453, Sept. 1988.

[11] N. B. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Trans. Commun.*, vol. COM-33, pp.551-557, 1985.

[12] L. Wang and M. Goldberg, "Progressive image transmission by transform coefficient residual error quantization," *IEEE Trans. Commun.*, vol. Com-36, pp. 75-86, Jan. 1988.

[13] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399-417, 1963.

[14] K. N. Ngan, K. S. Leong, and H. Singh, "Adaptive cosine transform coding of images in perceptual domain," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-37, pp. 1743-1750, 1989.

[15] Y. S. Ro and R. H. Young, "High compression ratio of DCT coding with blocking effects reduction," *Proceedings of 1989 Workshop on CVGIP at Rep. Of China*, pp. 377-388, 1989.

[16] J. W. Modestino, N. Farvardin, and M. A. Ogrinc, "Performance of block cosine image coding with adaptive quantization," *IEEE Trans. Commun.*, vol. COM-33, pp. 210-217, Mar. 1985.

[17] J. Max, "Quantizing for minimum distortion," *IRE Trans. Information Theory*, vol IT-6, pp. 7-12, March, 1960.

[18] J. D. Gibson, "DCT image compression over noisy channel," in *Statistical Signal Processing*, Ed. by E. J. Wegman and J. G. smith, pp. 411-419, 1984.

ACKNOWLEDGMENT

This paper was supported in part by a research grant to the first author from the National Science Council of the Republic of China in 1990.

TABLE 1. First 9 tables for a codebook of 32 maps

The number in the parentheses are the total number of bits for the map

Map 1 (8)	Map 2 (14)	Map 3 (20)
8	8 2 1	8 3 2
	2 1	3 2
		2

Map 4 (34)	Map 5 (38)	Map 6 (39)
8 3 2 2 1	8 4 4 3 2	8 5 4 3 2 1
3 3 2 1	4 4 2	4 3 2 1
2 2 1	3 2	3 2 1
2 1	2	
1		

Map 7 (41)	Map 8 (51)	Map 9 (57)
8 4	8 5 4 4 3	8 5 4 4 2
5 3	5 4 4	5 4 3 3
4 2	4 4	4 3 3
4 1	3	4 3
3	3	2
3		
2		
2		

TABLE 2. Examples of compression ratios by DCT in monochrome images

Reference	Contents	CR	Quality
[18]	Face	8 : 1	good
[12]	Face(Fig. 6-2)	40 : 1	unacceptable
[12]	Face(Fig. 6-3)	27 : 1	Barely acceptable
[12]	Face(Fig. 6-4)	16 : 1	Acceptable
[15]	Face	15 - 24 : 1	good

Legends

Fig. 1. Experiments on boundary compression by DCT. (a) upper left: original , (b) lower left: DCT for all the blocks, (c) upper right: compression by codebook method, CR=26, (d) lower right: Compression by zigzag method, CR=26.

Fig. 2. Outputs for the interactive system. The alphabetic order is the same as Fig. 1. (a) the original, (b) the compressed picture, (c) the DCT of the picture, (d) the block boundaries.

Fig. 3. Comparisons of three codebook method compressed pictures. The (a)-(d) order same as Fig. 1. (a) the original, (b) λ=4.0, CR=48.0, (c) λ=3.0, CR=41.8, (d) by interactive system, CR=48.0.

Fig. 4. (a), (d) same as those in Fig. 3. (b) Zigzag method with T=D=2, CR=31.3, and (c) T=D=3, CR=45.0.

Fig. 5. For the proofs of Lemma 1. Horizontal axis: bits for map t; Vertical axis: MSE by using map t.

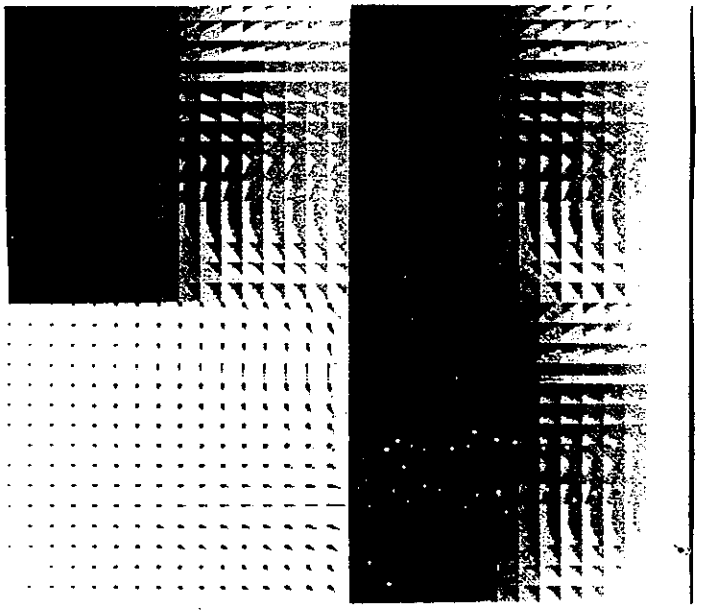

Figure 1

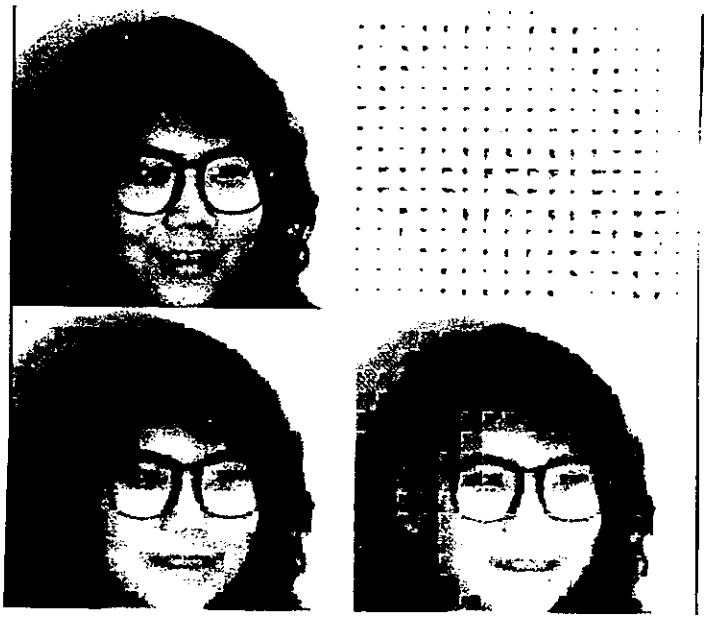

Figure 2

Figure 3

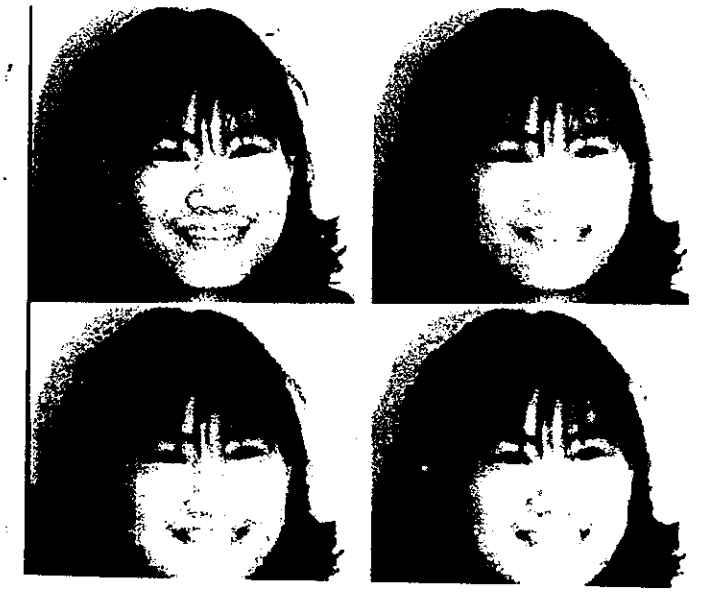

Figure 4

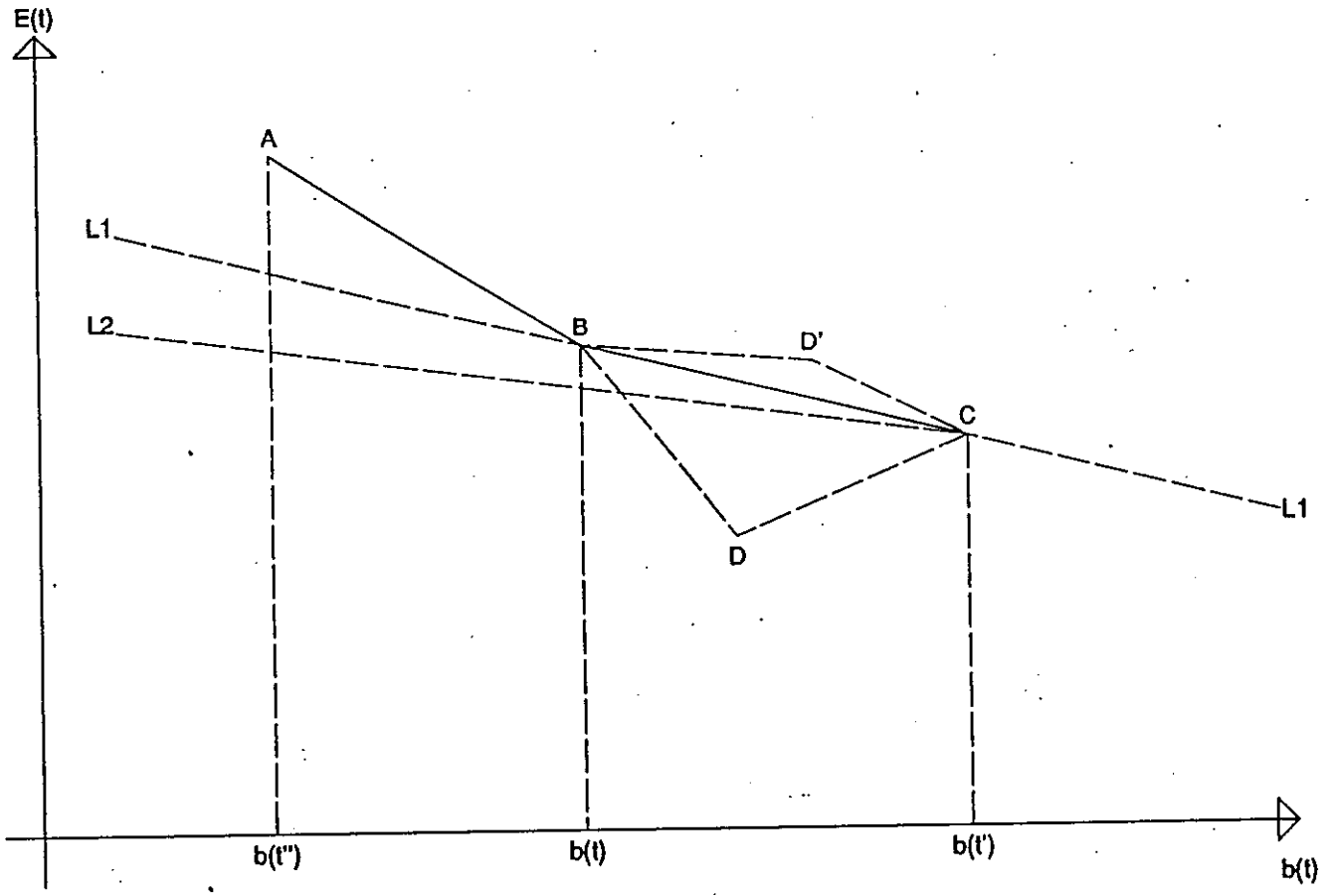

Figure 5