Interpolative Vector Quantization of Color Images

HSUEH-MING HANG, MEMBER, IEEE, AND BARRY G. HASKELL, FELLOW, IEEE

Abstract—**Interpolative vector quantization has been devised to alleviate the visible block structure of coded images plus the sensitive codebook problems produced by a simple vector quantizer. In addition, the problem of selecting color components for color picture vector quantization is discussed. Computer simulations demonstrate the success of this new coding technique for color image compression at approximately 0.3 bits/pel. Some background information on vector quantization is also provided.**

I. Introduction

In recent years, vector quantization (VQ) has been found to be an efficient data compression technique for speech and images [1], [2]. The simplicity of a VQ codec, especially the receiver, makes it a particularly valuable coding scheme. Most of the previous research on VQ has been conducted for speech and monochrome images; little has been done for color images.

A direct VQ applied to monochrome images has a very simple hardware structure; however, it suffers from the following problems: visible block structure in the reconstructed pictures and a codebook that is sensitive to pictures other than the training ones used to generate it. These two problems are even worse for color image coding. Because a color image has three color components, finding a codeword that matches three components simultaneously is more difficult.

Many proposals have been raised to solved the above problems. One simple method is to use a very large codebook and a very long training sequence. An immediate shortcoming of this method is its hardware complexity. The increase of codebook size both increases the searching time to allocate the best codeword in encoding and requires expanded memories for both encoder and decoder.

For practical purposes, we would normally prefer the use of a single universal codebook of reasonable size for all the pictures to be coded. We can also trade memory with computation; that is, devise a coding scheme that requires more computations than a direct VQ, but achieves a universal codebook of a much smaller size. This direction has been pursued by many researchers. For example, Baker and Gray [3], [4] suggested a differential VQ (DVQ) technique that removes the mean value of each coding block before it is vector-quantized. The block mean values are transmitted to the receiver via a PCM or a DPCM coder. Baker and Gray compared DVQ against the direct VQ and an adaptive DCT (discrete cosine transform) for monochrome images at the rates from 0.5 to 2 bits/pel. In all cases, DVQ worked better than the direct VQ and as well as the adaptive DCT. However, the hardware structure of a DVQ codec is much simpler than that of an adaptive DCT. Nevertheless, they also pointed out that the coded pictures of VQ and DVQ acquired the "blockiness" artifacts at rates below 1 bit/pel.

Murakami *et al.* [5], [6] proposed a VQ coding scheme that not only removes the block means, but also normalizes the block variances. We can call this scheme *normalized DVQ*. They also used a different distortion criterion—minimizing the block peak error, rather than the mean-squared error used by Baker and Gray. They showed the feasibility of this technique, but did not compare it to the other coding schemes.

Another proposal is the segmented codebook approach introduced by Gersho and Ramamurthi [7]. They first classified the blocks of an image into several constituent groups and then designed codebooks separately for each group. Direct VQ was then performed on each group using the corresponding codebook. This scheme has been refined [8], [9] and needs a complex segmentation process which may be more difficult for real-time image coding.

All the above VQ coding schemes were originally proposed for monochrome pictures. Recently, Boucher and Goldberg [10] suggested a direct VQ for color pictures. Their scheme transmitted the codebook of every individual subpicture (a 32 × 32 window of a picture). Each codebook was designed and used exclusively for only one subpicture; therefore, they could eliminate the codebook sensitivity problem and thus name it *adaptive VQ*. However, because of the repeated transmission of codebooks, the bit rate of this scheme could not be very low. Moreover, the computations required for codebook design for every picture window would pose some difficulty for real-time coding.

We will present an interpolative VQ (IVQ) scheme that can reduce the coded picture blockiness and provide a robust universal codebook for most real-world images. The basic idea is to send separately, via PCM or DPCM, one picture element (pel) of each block to be coded. From these, an interpolation is constructed that is subtracted from the original image, producing a differential picture that is then coded by VQ. Furthermore, in order to encode color pictures, we also look into such related problems as color space selection and component coding. Using this new scheme, we have obtained very successful experimental results at very low rates, around 0.3 bits/pel (for all three colors together).

The organization of this paper is as follows. Section II briefly reviews the fundamentals of VQ including the codebook design techniques. The new IVQ coding scheme is presented in Section III. Section IV discusses a few topics associated with color picture coding. For comparison, computer simulations in Section V not only demonstrate the feasibility of IVQ, but also examine the DVQ and the normalized DVQ schemes for color image coding.[1]

II. Vector Quantization

A VQ is a mapping from a k-dimensional Euclidean space R^k to a finite subset of R^k. This finite set Y is called a *VQ codebook* or *VQ table*. For coding purposes, a complete VQ codec has two parts: an *encoder* which assigns each data vector $x \in R^k$ to a *channel symbol* (or *index*), and a *decoder* which maps a channel symbol to a *codeword* (or *reproducing vector*) $y \in Y$. By choosing the size of codebook Y, we can control the transmission rate of a VQ coding process. Our goal is to select an optimal codebook Y of size N that results in the

Paper approved by the Editor for Image Processing of the IEEE Communications Society. Manuscript received October 1, 1986; revised April 1, 1987. This work was presented at the 1986 Picture Coding Symposium, Tokyo, Japan, April 1986.

The authors are with AT&T Laboratories, Holmdel, NJ 07733.

IEEE Log Number 8819674.

[1] After this work being done, we found that a modified version of DVQ has been applied to color images [26], and a normalized DVQ and its extensions on color image coding are reported in [27].

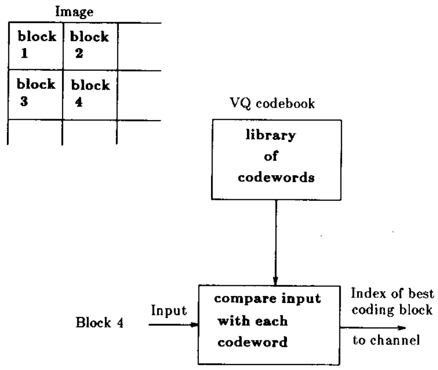


Fig. 1. Vector quantization of images.

lowest possible distortion among all possible codebooks of the same size. The schematic diagram of a vector quantizer (encoder) for image coding is shown in Fig. 1. The VQ decoder structure is not included here, since it is simply a table lookup process.

In order to analyze the performance of VQ, both the source model and the distortion measure need to be simple and mathematically tractable. Indeed, it has been proved that VQ can asymptotically achieve the rate-distortion bound under proper assumptions of signal sources and distortion measures [11]–[13]. In other words, VQ is one of the known coding techniques that could approach the rate-distortion limit.

Although the idea of VQ has been around for a long time, it is the recent discovery of codebook design methods that makes this coding scheme practical. Two groups of VQ design methods have been invented. The first group of methods constructs codebooks based on mathematical formulas. This approach can achieve uniform quantization [11], [14] and can pose a regular structure (e.g., lattice) on the codebook for fast encoding and decoding [15]–[17]. However, it usually needs a probabilistic model of the signal source and requires a huge codebook for good performance.

The second group designs the (locally) optimum codebook based on empirical data (so-called *training set*). This approach is sensitive to the *outside data* (which are not inside the training set), but is considered to be more practical for small size codebooks (less than 1000) and also eliminates the requirement of a probabilistic model. Two empirical VQ design algorithms, at least, are available as outlined below.

The first empirical design scheme is suggested by Linde, Buzo, and Gray [18] and thus named the *LBG algorithm*. This scheme is a generalization of Lloyd's PCM design technique. It contains two basic steps at each iteration: 1) partitioning the training data and 2) updating the codebook. Starting with an initial guess for the codebook, this design process partitions the training vectors into groups by finding the closest codeword for each training vector. The new codewords (vectors) are updated by calculating the minimum distortion representative of each partition group. For mean-squared error distortion, the new codeword of a partition group is simply the average of all vectors in that group. The above two steps proceed alternately until a preselected stopping criterion is met, which is usually a small distortion decrease between two successive iterations.

The LBG algorithm has the advantages that its basic framework does not depend on the distortion measure, and that it does not need explicit models of signal sources. However, it has the following disadvantages: 1) a poorly selected initial codebook may lead to an undesirable final codebook, and 2) a complete design requires a very large number of computations. The computational problem is a more critical issue for images because image coding usually involves a lot of data.

The computation associated with the LBG algorithm is proportional to both the training data size and the codebook size. For example, a regular 512 × 512 picture partitioned into 4 × 4 blocks has 16 384 training blocks (vectors). A straightforward LBG algorithm compares each training block against all the codewords in the codebook, which is normally 200–1000, and then decides the closest match. This procedure is repeated for every training vector. Therefore, numerous computations are required for a totality of 10–20 iterations.

A second empirical VQ design method is proposed by Equitz [19], which eliminates the need of an initial codebook and reduces the computations dramatically. Equitz calls it the *nearest neighbor* (NN) algorithm, although it is somewhat different from the original nearest neighbor algorithm suggested in cluster analysis [20]. If fact, the design of a VQ codebook is basically a clustering problem and, therefore, the methods invented for cluster analysis are applicable to VQ. The LBG algorithm is equivalent to the so-called "k means algorithm" in cluster analysis [21], the Equitz NN algorithm is a version of the *Ward method* [22].

At the beginning of the Equitz NN algorithm, all the training vectors are viewed as initial clusters. Then, the two nearest clusters are merged and replaced by a new cluster which is a weighted average of the merged two clusters. Hence, the number of clusters after merging is reduced by one. This process is repeated until the desired number of clusters is reached. Again, if all the possible pairs are to be compared at each merging step, the total computations are numerous. To ease the computing burden, Equitz poses a *k-d* tree [23] structure on the training data. This tree is constructed in such a way that the "similar" data vectors are usually grouped under the same tree branch. Therefore, we only search the local areas of the tree to allocate the nearest pairs. This suboptimum scheme can dramatically reduce computations, and will be referred as the *Equitz algorithm*. Because of the approximation steps introduced by Equitz to the NN algorithm, the convergence property of his algorithm can only be shown by experiments.

As reported by Equitz [19], the performance of his algorithm is better than that of the LBG algorithm with a randomly selected initial codebook. More importantly, the computer time of the Equitz algorithm only takes 1/20 of that of the LBG algorithm. All the above experiments were conducted on real images. As suggested by Equitz, an even better codebook can be constructed by first using the Equitz algorithm to obtain the initial codebook and then using the LBG algorithm to refine the codebook. This refinement would be important for the cases that the Equitz algorithm generates many similar code vectors due to its suboptimality and/or the computer numerical errors. This modified Equitz and LBG algorithm will be used in the rest of this paper.

III. Interpolative Vector Quantization

Various techniques were invented to improve the coded picture visibility and to increase the codebook robustness of a VQ coding process. As being described in Section I, DVQ and the normalized DVQ are examples along this direction. Both DVQ and the normalized DVQ have quite insensitive universal codebooks, but they could not eliminate the block-like contours along the edges of coding blocks. This is because the mean values of adjacent blocks are often unequal, and the quantized differential signals are not usually good enough to smooth the gaps between the means of neighboring blocks.

The codebook robustness property of the normalized DVQ is even better than that of DVQ. However, it introduces another undesirable defect on the coded images. The differential signals in the picture edge regions have larger variance than those in the smooth regions. Thus, the normalization operation reduces the amount of signal changes at edges, while it enlarges the signal changes in the smooth regions. Therefore, the sharp edges in the original picture are relatively poorly reproduced; i.e., it blurs the picture edges. This may be improved with better codebook construction. Another inherent drawback of the normalized DVQ is the additional block variance information to be transmitted to the receiver.

A new coding scheme named interpolative VQ (IVQ) is hence proposed to alleviate the block boundary visibility problem. Similar to DVQ, it sends the low-frequency components and the high-frequency components of a picture separately. The low-frequency components, which represent the fundamental patterns of an image, are coded in a more precise manner, while the high-frequency components, which enhance the details of an image, are loosely coded. Unlike DVQ, the low-frequency components in IVQ are reconstructed by interpolative surfaces which do not create gaps between neighboring blocks and hence produce perceptually better outcomes.

The basic operation of an IVQ is shown in Fig. 2. A test image is first partitioned into small rectangular blocks and then one representative value is selected for each block. This representative could be the block mean value or the lower right corner pel of a block. Using the neighboring four block representatives A, B, C, and D, we can construct an interpolative surface at block 4. For example, a bilinear interpolative surface using four lower right corner pels would suffice for most coding purposes. The block representatives are then coded either by a PCM or a simple DPCM circuit.

The differential signal between the original image and the interpolative surface is then vector quantized. The complexity increase in this new system is due only to the interpolator which can be easily implemented when a bilinear interpolator size of power of 2 is in use. The decoder of IVQ is drawn in Fig. 3. The interpolative surface is first reconstructed from the block representatives. The vector quantized differential signals are then added to the interpolative surface to form the final outputs. We will show later in Section V that IVQ has about the same strong robustness as DVQ, but has perceptually better coded images.

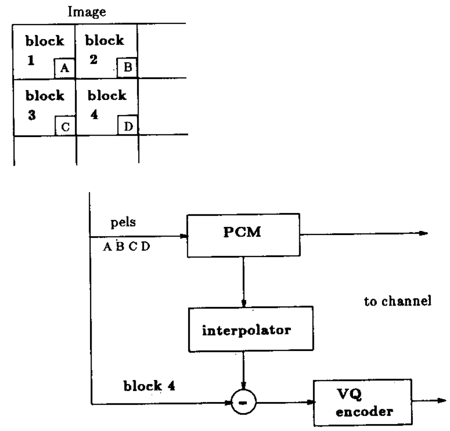


Fig. 2. Interpolative vector quantization (IVQ) encoder.

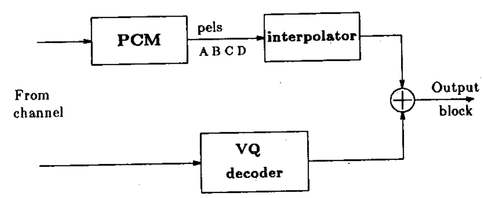


Fig. 3. IVQ decoder.

IV. Color Picture Coding

IVQ can be directly applied to monochrome pictures; however, a few related problems need to be investigated for color picture coding. Three primary components are required to present a color picture. The popular choices of color primaries are: 1) RGB (red, green, and blue) primaries, 2) YIQ components defined by NTSC (National Television System Committee), and 3) YUV components defined by PAL (phase alternative line) [24]. The last two systems are closely related. The Y component in both systems represents the brightness of a color picture, and the other two components are chrominance signals (IQ or UV) that are usually sent with lower resolution than the luminance. A simple linear transformation could convert UV components to IQ components and vice versa.

Two problems are involved in color picture coding: 1) Should separate component coding or combined component coding be used? and 2) Which color primary system is more suitable for data compression? For vector quantization, combined component coding means that a single codeword (vector) in the VQ codebook represents all three color components; they are tied together and are designated by a single channel symbol. In separate component coding, each color primary of a picture is regarded as an independent monochrome picture. Hence, coding a color picture is equivalent to coding three monochrome pictures independently. Our preliminary experiments using the parameters described below have shown that separate component coding is more robust when a universal VQ codebook is to be used. This is especially true for low bit rates and small training sets. If a training set either lacks or has a very small percentage of a certain color, the resulting codebook will not contain that color, and the combined component codes can never produce a good match to that color. On the other hand, the three codebooks in the separate component coding case are more likely to produce a better approximation to a color not in the training pictures. This experimental result can also be explained as the three individual component codebooks are equivalent to a single, extremely large, combined codebook. But unlike such a codebook, each of the individual component codebooks is small and manageable. Of course, if a VQ codebook is used only for its training pictures, the combined component coding would be more efficient since it incorporates all the correlations among its three color primaries.

The next problem is to select the color primaries. We will consider both coding efficiency and coded picture visibility of the aforementioned color primaries. If we have already decided to use the separate component coding for the sake of codebook robustness, the YUV or YIQ primaries are more efficient than RGB. For most color pictures, the spatial bandwidth of the U or V component can be reduced to less than 1/3 that of the Y component [24]. Therefore, we can spatially subsample the U and V components without hurting the overall coding performance. Simulations in Section V will verify this point. This is not the case for RGB components because they all require about the same spatial bandwidth as the Y component.

The second consideration is the perceptual properties of color primaries. A convenient distortion measure is a function of distance in the color space spanned by color primaries. Hence, an ideal color space is perceptually uniform, i.e., the similarity measure of two colors is proportional to their geometric closeness in the color space, and this measure is spatially invariant. Unfortunately, all the primaries in our considerations are perceptually nonuniform [24]. Approximately uniform color spaces suggested so far all require complex and nonlinear transformations of the camera's RGB outputs [24], [25]. The effectiveness of uniform color space for VQ coding remains a research topic.

TABLE I
MEAN SQUARED ERRORS OF SOME DIRECT VQ SCHEMES

Coding schemes	Lady (training)	Lake (outside)	Peppers (outside)
RGB combined 0.5 bit/pel 4×4 block	81.9	1002.3	1445.1
YUV combined 0.5 bit/pel 4×4 block	83.9	1814.6	1097.5
YUV separate 0.33 bit/pel 6×6 block	62.3	513.7	551.8

V. EXPERIMENTAL RESULTS

This section has two parts. The first part compares the combined component coding to the separate component coding. The second part demonstrates the coding performance of IVQ, DVQ, and the normalized DVQ on color images. The picture data base from USC (University of Southern California) Image Processing Institute are used as test images. All pictures in this section are 512×512 with 8 bits per color component (in R, G, B).

As mentioned earlier in Section II, the Equitz VQ algorithm is more efficient than the LBG algorithm and thus was chosen to design the initial codebooks. The final codebooks were obtained by polishing the initial codebooks using the LBG algorithm. Since the initial codebooks are close to the final "optimum" codebooks, the LBG design procedure usually requires only a few iterations to converge. The *k-d* tree structure was used for both codebook design and encoding process (searching the best codeword). The *k-d tree search algorithm* proposed by Friedman *et al.* [23] has the same performance as the exhaustive search (optimal search), but takes much less arithmetic operations. On average, its searching time for one input vector is proportional to log N where N is the codebook size [23].

The popular *mean-squared error* distortion measure is adopted for its simplicity in calculation and in deriving the Equitz clustering algorithm. Coding errors are the differences between the test image and the coded image in the RGB space, viewing R, G, and B as three separate components. Because the RGB space is not perceptually uniform and because the squared error does not match well with the human vision, subjective evaluation of the coded pictures is considered to be more important.

In the first simulation, three coding techniques were performed to test the color spaces and the component coding schemes. They are, respectively, RGB combined, YUV combined, and YUV separate. They all employ direct VQ, i.e., codewords are selected to match the original picture in a certain color space. These coding schemes together with their coding errors on the test images are listed in Table I. The lady's face (Lady) is selected as the training picture, and the rest are *outside* pictures (pictures outside the training set). The coding block size and codebook size are chosen to yield an average bit rate lower than 0.5 bits/pel. In order to achieve this, the separate component coding, which transmits the codeword indexes of three components separately, uses a block size of 6×6, which is larger than the 4×4 blocks used for both the RGB and the YUV combined coding schemes. The two combined coding schemes in Table I use 256-vector codebooks, which gives an average 0.5 bits/pel.

To reduce the transmission rate of the separate component coding schemes, the chrominance signals are subsampled before quantization. The separate coding scheme in Table I has three codebooks: a 1024-vector codebook for Y, and two 256-vector codebooks for U and V, separately. The U and V components are subsampled by choosing every third pel of every third line. Hence, the total pels of both U and V are reduced by a factor of 9. The coding block size of the subsampled U and V is still 6×6. In other words, an 18×18 block of the original U or V is condensed to one 6×6 coding block by subsampling. Therefore, the total average bit rate of this separate coding scheme is $[10 + (2 \times 8)/9]/(6 \times 6) = 0.33$ bits/pel where the factor 9 comes from subsampling.

Even though the squared error is not a good distortion measure for pictures, it is evident from Table I that the separate component coding scheme is much more robust than the other two schemes. This is especially true when one visually inspects the coded pictures produced by these schemes. A few selected coded pictures are shown in Fig. 4. This result verifies the analysis in Section IV that the combined coding scheme would be inferior if the training pictures do not have the colors contained in the outside pictures. As for the color space, the YUV space is marginally better than the RGB space subjectively.

As the second part of our experiments, three robust VQ schemes are examined. They are IVQ, DVQ, and *normalized* DVQ. The separate coding technique, which has been shown to be more robust in the fist part of the experiments, is included in all these schemes. Due to the additional transmission of block means (or block representatives), the coding block size is chosen as large as 8×8 so that the overall bit rate is below 0.5 bits/pel. Chrominance signals are subsampled by choosing every fourth pel of every fourth line, i.e., a 16:1 pel reduction. The codebook size is 1024 for Y and 256 for U and V, individually. These parameters are the same for all the coding schemes in Table II. Consequently, the average bit rate of IVQ and DVQ is $[8 + 10 + 2 \times (8 + 8)/16]/(8 \times 8) = 0.3$ bits/pel where 8 bits/block is used to PCM-encode the block means or block representatives. If a 5-bit DPCM is used to send block means, the overall coding efficiency can be further increased by 10 percent. The normalized DVQ has a higher bit rate, that is $[8 + 8 + 10 + 2 \times (8 + 8 + 8)/16]/(8 \times 8) = 0.42$ bits/pel. For simplicity, the block representative of IVQ is the lower right corner pel of each block, and the bilinear interpolar is used to generate interpolative surfaces. Moreover, if the indexes of VQ codes are further compressed by a variable length entropy code, the bit rate for all cases could drop another 15–25 percent.

As shown in Table II, the coding errors of these three coding schemes are comparable; however, the pictures produced by IVQ are perceptually smoother than the others. Some coded pictures are displayed in Fig. 5. These pictures also demonstrate the feasibility of a universal IVQ for general purpose picture coding. In other words, a nonadaptive IVQ with a fixed codebook is acceptable to encode a large number of pictures. Of course, as one would expect, the outside picture quality is somewhat lower.

VI. CONCLUSIONS

In this paper, we have investigated several VQ coding schemes for color image compression. An interpolative vector quantizer has been invented to improve the visibility of vector-quantized pictures. In our computer simulations, IVQ has been shown to reduce the blockiness of the coded pictures. Moreover, the simulations also demonstrate the feasibility of

Fig. 4. Coded pictures. (a) and (b) are coded by RGB combined scheme, rate = 0.5 bits/pel. (c) and (d) are coded by YUV separate scheme, rate = 0.33 bits/pel.

TABLE II
MEAN SQUARED ERRORS OF IVQ, DVQ, AND NORMALIZED DVQ

Coding schemes	Lady (training)	Lake (outside)	Peppers (outside)
IVQ 0.3 bit/pel 8×8 block	40.2	288.5	365.6
DVQ 0.3 bit/pel 8×8 block	41.9	281.9	367.1
Normalized DVQ 0.42 bit/pel 8×8 block	57.6	230.8	335.7

Fig. 5. Coded pictures. (a) and (b) are coded by DVQ, rate = 0.3 bits/pel. (c) and (d) are coded by IVQ, rate = 0.3 bits/pel.

the universal codebook proposal. Owing to its simple hardware structure, IVQ is believed to be a good candidate for practical applications. IVQ, as well as direct VQ, is expected to be further improved by using a perceptually uniform color space and by using a distortion measure that models the human vision more accurately. In addition, both LBG and Equitz design schemes tend to generate many similar codewords for the flat areas in the training images. A mix of the empirical design and the theoretical design may thus provide a more robust codebook for a larger class of pictures. These studies are still in progress.

REFERENCES

[1] A. Gersho and V. Cuperman, "Vector quantization: A pattern-matching technique for speech coding," *IEEE Commun. Mag.*, Dec. 1983.

[2] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, Apr. 1984.

[3] R. L. Baker and R. M. Gray, "Image compression using non-adaptive spatial vector quantization," presented at Proc. 16th Asilomar Conf. Circuits Syst. Comput., Oct. 1982.

[4] ——, "Differential vector quantization of achromatic imagery," presented at Proc. Int. Picture Coding Symp., Mar. 1983.

[5] T. Murakami, K. Asai, and E. Yamazaki, "Vector quantizer of video signals," *Electron. Lett.*, pp. 1005-1006, Nov. 1982.

[6] ——, "A design of vector quantizer for video signals," presented at Proc. Int. Picture Coding Symp., Mar. 1983.

[7] A. Gersho and B. Ramamurthi, "Image coding using vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing,* Paris, France, Apr. 1982, pp. 428-431.

[8] B. Ramamurthi, A. Gersho, and A. Sekey, "Low-rate image coding using vector quantization," in *Proc. IEEE Global Commun. Conf. Rec.*, Nov. 1983, pp. 184-187.

[9] B. Ramamurthi and A. Gersho, "Image vector quantization with a perceptually-based cell classifier," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing,* Mar. 1984, pp. 32.10.1-10.4.

[10] P. Boucher and M. Goldberg, "Color image compression by adaptive VQ," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing,* Mar. 1984, pp. 29.6.1-6.4.

[11] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory,* vol. IT-25, pp. 373-380, July 1979.

[12] Y. Yamada, S. Tazaki, and R. M. Gray, "Asymptotic performance of a block quantizer with difference distortion measures," *IEEE Trans. Inform. Theory,* vol. IT-26, pp. 6-14, Jan. 1980.

[13] J. A. Bucklew and G. L. Wise, "Multidimensional asymptotic quantization theory with *r*th power distortion measures," *IEEE Trans. Inform. Theory,* vol. IT-28, pp. 239-247, Mar. 1982.

[14] J. A. Bucklew, "Companding and random quantization in several dimensions," *IEEE Trans. Inform. Theory,* vol. IT-27, pp. 207-211, Mar. 1981.

[15] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory,* vol. IT-28, pp. 211-226, Mar. 1982.

[16] ——, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inform. Theory,* vol. IT-28, pp. 227-232, Mar. 1982.

[17] ——, "A fast encoding methods for lattice codes and quantizers," *IEEE Trans. Inform. Theory,* vol. IT-29, pp. 820-824, Nov. 1983.

[18] 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.

[19] W. H. Equitz, "Fast algorithms for vector quantization picture coding," M.Sc. thesis, Dep. Elec. Eng. and Comput. Sci., M.I.T., MA, June 1984; also, in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing,* Dallas, TX, Apr. 1987, pp. 725-728.

[20] K. S. Fu, Ed., *Digital Pattern Recognition.* New York: Springer-Verlag, 1976.

[21] J. A. Hartigan, *Clustering Algorithms, Chap. 4.* New York: Wiley, 1975.

[22] M. R. Anderberg, *Cluster Analysis for Applications.* New York: Academic, 1973.

[23] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Software,* vol. 3, no. 3, pp. 209-226, Sept. 1977.

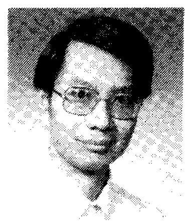
[24] W. N. Sproson, *Colour Science in Television and Display Systems.* Philadelphia, PA: Heyden and Son, 1983.

[25] J. O. Limb, C. B. Rubinstein, and J. E. Thompson, "Digital coding of color video signals—A review," *IEEE Trans. Commun.,* vol. COM-25, pp. 1349-1385, Nov. 1977.

[26] S. E. Budge and R. L. Baker, "Compression of color digital images using vector quantization in product codes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing,* Tampa, Florida, Mar. 1985, pp. 129-132.

[27] T. Murakami, K. Asai, and A. Itoh, "Vector quantization of color images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing,* Tokyo, Japan, Apr. 1986, pp. 133-136.

★

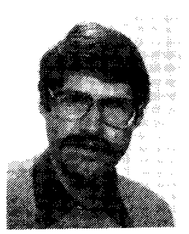


Hsueh-Ming Hang (S'79-M'85) was born in Taipei, Taiwan, on May 22, 1956. He received the B.S. degree in control engineering and the M.S. degree in electronics from National Chiao-Tung University, Hsinchu, Taiwan, in 1978 and 1980, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1984.

From 1980 to 1983 he was a Graduate Assistant in the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute. During 1983-1984 he was an Instructor in the same department, teaching courses in linear and communication systems. Since 1984 he has been with the Visual Communications Research Department, AT&T Bell Laboratories, Holmdel, NJ. His research interests include digital video compression, multidimensional signal processing, and information theory.

Dr. Hang is a member of Sigma Xi.

★



Barry G. Haskell (S'65-F'87) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1964, 1965, and 1968, respectively.

From 1964 to 1968 he was a Research Assistant in the University of California Electronics Research Laboratory, with one summer being spent at the Lawrence Livermore Laboratory. Since 1968 he has been at AT&T Bell Laboratories, Holmdel, NJ and is presently Head of the Visual Communications Research Department. He has also taught graduate courses at Rutgers University, City College of New York, and Columbia University, New York, NY. His research interests include digital transmission and coding of images, videotelephone, satellite television transmission, medical imaging as well as most other applications of digital image processing. He has published over 30 papers on these subjects and has 15 patents either granted or pending.

Dr. Haskell is a member of Phi Beta Kappa and Sigma Xi.