Automatic Edge Class Formulation for Classified Vector Quantization

Jae-young Jung*

요 약 영상 압축 분야에서 분류벡터양자화 방법은 에지와 같이 시각적 인지에 중요한 특징을 잘 복원해주는 특성을 가지고 있다. 그러나, 기존의 분류벡터양자화에서는 수직, 수평, 대각 에지 클래스와 같은 몇 개의 선형 에지 클래스를 사전에 정의하고 분류함으로써, 영상 내 존재하는 다양한 유형의 에지 패턴을 효과적으로 재구성할 수 있도록 일반화되어 있지 못하다. 본 논문에서는 에지 패턴들간의 유사도 측정자를 정의하고 이를 바탕으로 에지 블록을 분류하는 새로운 방법을 제안한다. 영상내의 각 에지블록은 그 블록이 가지는 에지 패턴의 형태에 따라 하나의 특징벡터로 변환된다. 훈련 영상들로부터 다양한 형태의 에지 패턴들을 유사도가 높은 것들끼리 군집화하여 일반화된 에지 클래스를 자동으로 생성한다. 실험에서는 생성된 선형/비선형 에지 클래스의 유형을 보이고, 이를 이용하여 0.6875bpp로 압축된 결과 영상에서 에지가 잘 보존되고 있음을 보인다.

Abstract In the field of image compression, Classified Vector Quantization(CVQ) reveals attractive characteristics for preserving perceptual features, such as edges. However, the classification scheme is not generalized to effectively reconstruct different kinds of edge patterns in the original CVQ that predefines several linear-type edge classes: vertical edge, horizontal edge, diagonal edge classes. In this paper, we propose a new classification scheme, especially for edge blocks, based on the similarity measure for edge patterns. An edge block is transformed to a feature vector that describes the detailed shape of the edge pattern. The classes for edges are formulated automatically from the training images to result in the generalization of various shapes of edge patterns. The experimental results show the generated linear/nonlinear types of edge classes. The integrity of all the edges is faithfully preserved in the reconstructed image based on the various type of edge codebooks generated at 0.6875bpp.

1. Introduction

Vector Quantization(VQ) techniques have been used for coding digital images with many attractive features by utilizing the correlation among the neighboring pixels[1]. However, the direct application of ordinary VQ to digital images results in the edge degradation as edges take small proportion within an image. To alleviate the edge degradation, Classified Vector Quantization(CVQ) is introduced[2]. It firstly classifies each image block into several categories according to the visual characteristics, such as the existence of edges within the block. Then, the separate codebook for each class are designed from the training image blocks which belongs to the same class. Each input image block is encoded by searching only the subcodebook of the class to which the block belongs not by searching the entire codebook. By means of this classification, the perceptual features such as edges are well preserved in CVQ compared to the ordinary VQ.

Since edges contain a large part of perceptual information of the image, various classification strategies of edge patterns are studied either in the spatial or in the transform domain[2~7]. In general, the edge patterns are approximated to linear shapes and classified according to their positions and orientations. In order to reconstruct faithfully the detailed shape of the edge patterns, Block Pattern-Vector Quantization (BPVQ) attempts to formulate edge classes each of which is identified by a

* School of Computer Engineering at Dongyang University

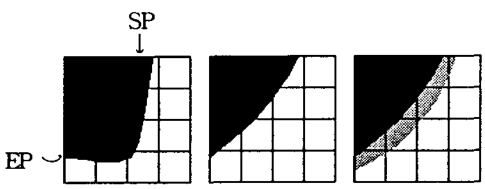
block pattern[4].

In this paper, we propose a new approach to formulate edge classes each of which is a generalized form of the edge patterns with similar shapes. In section 2, the representation of an edge pattern by a feature vector is described. The features are selected to characterize the global direction, the thickness, and the curvedness of the edge patterns. The classification scheme for the image blocks and edge class formulation process are discussed in section 3. Finally in section 4, the experimental results are given.

2. Edge Representation with Similarity Measure

The edges we are considering in this work are assumed to cross the entire subimage block(4x4 block). Thus the edge always meets the boundary at two points named as boundary points. We denote one of them as a starting point(SP) and the other as an ending point(EP). The starting point is chosen to have the darker region in the right side. The locations of SP and EP determine the location and global orientation of an edge pattern. As the edge usually passes through several pixels, the exact locations of these two points need to be estimated.

The detailed shape of an edge pattern is characterized by their curvedness and the width of the edge slope which is the Euclidean distance from the lighter region to the darker region along the orthogonal direction with respect to the edge. <Figure 1>Hows the different kinds of edge patterns despite the same locations of boundary points. In order to represent above characteristics of the edge shape, we formulate a membership function which estimates the proportion of the edge component for each interior pixel. Here, an interior pixel denotes a pixel which is not at the block boundary.



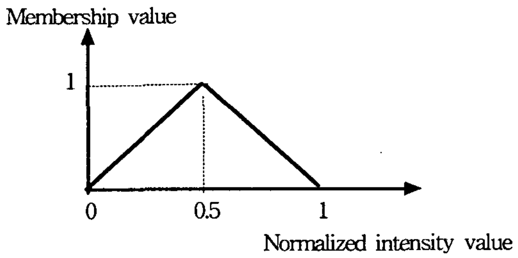
<Figure 1> Several types of edges

An image block is normalized so that each pixel has a gray level value in the range of 0~1. To do this, we firstly partitioned an image block into a lighter region and a darker region. The lighter(darker) region consists of pixels with the gray level values above(below) the mean value of gray level values of the block. Then, the mean of gray levels of the pixels in the lighter region, m_l, and the mean of gray levels of the pixels in the darker region, m_d, are calculated by adopting the procedure suggested by S. A. Mohamed and M. M. Fahmy[4]. For each pixel, the value "1"("0") is assigned if the gray level is above m_l (below m_d). For a pixel which has the gray level, p_i, in between m_l and m_d, a normalized value, p_n, is assigned as

$$p_n = \frac{p_i - m_d}{m_l - m_d} \tag{1}$$

The exact location of the boundary point from the top left corner of the block is estimated by adding the normalized gray levels of the row pixels to which the point belongs in case that the point meets at the horizontal boundary. Otherwise, the location is estimated by adding the normalized gray levels of the column pixels to which the point belongs.

In order to determine the extent to which the edge pattern is contained in a pixel, we formulate a triangular shaped membership function, as shown in <Figure 2> which has a maximum value of "1" at the normalized intensity value of "0.5", and a value of "0" at the normalized intensity values "1" or "0". It is a reasonable assumption that a completely dark or light pixel does not include an edge, while a pixel with a gray level in the middle of dark and light values has a maximum possibility to contain an edge.



<Figure 2> Membership function of edge pixel

In order to describe the shape of an edge pattern, E_i, we formulate a feature vector F_i. The first feature, F_i^1, and the second feature, F_i^2, represent the position of a starting point and an ending point of the edge, respectively. Each feature is computed as an Euclidean distance from top-left corner of the image block to the point following the boundary counter-clockwise. The width of a pixel is regarded as 1. The third feature, F_i^3, denotes the orientation of the edge which is calculated as a tilted degree of the line connecting starting and ending points with respect to a horizontal line. $F_i^{(4\sim7)}$ represent the value of the membership function of edge for four interior pixels, one for each pixel.

The similarity measure, S_{ij}, between edge patterns E_i and E_j is defined as

$$S_{ij} = \alpha\ S_{ij}^{(3)} S_{ij}^{(1,2)} + \beta\ S_{ij}^{(4\sim7)} \quad (2)$$

Where, $S_{ij}^{(f)}$ is the similarity measure between block E_i and E_j based on $f\ th$ feature and α, β are weighting values for the corresponding terms.

The similarity measures are given by

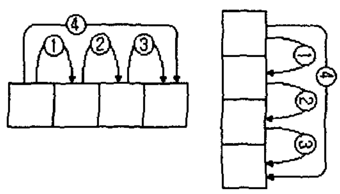
$$S_{ij}^{(1,2)} = \max\left\{ \frac{1}{2}\left(\frac{1+\cos(|F_i^1 - F_j^1| * \frac{\pi}{12})}{2} + \frac{1+\cos(|F_i^2 - F_j^2| * \frac{\pi}{12})}{2} \right), \frac{1}{2}\left(\frac{1+\cos(|F_i^1 - F_j^2| * \frac{\pi}{12})}{2} + \frac{1+\cos(|F_i^2 - F_j^1| * \frac{\pi}{12})}{2} \right) \right\} \quad (3)$$

$$S_{ij}^{(3)} = \frac{1+\cos\left(|F_i^3 - F_j^3| * \frac{\pi}{4}\right)}{2} \quad (4)$$

$$S_{ij}^{(4\sim7)} = \sqrt{\frac{1}{4}\sum_{k=4}^{7}|F_i^k - F_j^k|^2} \quad (5)$$

3. Classification of Image Blocks and Formulation of Edge Classes

The input image is decomposed into non-overlapping 4x4 blocks. For each block, four 4-dimensional subblocks are formulated from the boundary pixels. That is, 1*st*-row-vector, 4*th*-row-vector, 1*st*-column-vector and 4*th*-column-vector of the 4×4 image array are extracted. The reason for this choice is that we regarded an edge block as the block containing an edge crossing the entire block. For each subblock, the number of gray level changes between two pixels greater than a certain threshold is counted. The way of comparison between two pixels is shown in Fig. 3. Here, the comparison ④ is necessary to determine whether the gray level across the block is rapidly changing or not even if it has no edge. According to the number of changes above the threshold, the subblock is categorized as the PLANE, the TEXTURE or the EDGE class. If the number of changes over threshold is zero, the subblock is classified as the PLANE class. If it is one, the subblock belongs to the TEXTURE class as the change is detected in only ④ of <Figure 3>. Otherwise, it is classified as an EDGE subblock.



<Figure 3> Gray level change detection in a subblock

The class distribution of subblocks determines the category of the image block as PLANE, TEXTURE or EDGE class. The classification rule is as follows; If all subblocks are PLANE classes then the block belongs to the PLANE class. If all subblocks belong to either PLANE or TEXTURE, excluding the case that all subblocks are denoted as PLANE classes, then the block is classified as a TEXTURE class. Otherwise, the block is classified as an EDGE block.

The classes for the edge blocks are formulated automatically from the edge blocks of the training images. Each edge block is transformed to a feature vector defined previously. The vectors are clustered using PNN(Pairwise Nearest Neighbor) algorithm based on the similarity measure introduced. In order to clarify the general shape of edge patterns for generated classes, we computed the average of normalized image of codewords for class i, $P^i(x,y)$, as

$$P^i(x,y) = \frac{1}{m}\sum_{k=1}^{m} P^k_{n,i}(x,\ y) \qquad (6)$$

$$x,\ y \in \{1,2,3,4\}$$

where m is the codebook size of the class i, $P^k_{n,i}(x,\ y)$ is the normalized image of k *th* codeword.

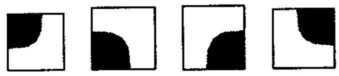
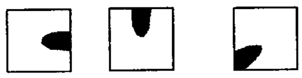
4. Results and Conclusions

The proposed technique has been evaluated by performing several experiments with images. Firstly, the edge classes are formulated from 3 training images. The shape of edge patterns of generated edge classes can be categorized as linear type, convex type, concave type, hole type, and bump type. The shape of edge patterns and the number of classes for each edge type are shown in Table 1. Notice that about 42.3% of edge classes are generated for nonlinear type of edge patterns.

The reproduced LENNA image from outside the training sequences is shown in <Figure 4>. The codebook sizes of the PLANE class and the TEXTURE class were chosen to be 72 and 128, respectively. The obtained PSNR is 31.7 dB at 0.6875 bpp. There is no detectable degradation in the reconstructed image. The integrity of all the edges is preserved faithfully.

It is expected that this technique works favorably on the high-detailed images since the edge classes include various kinds of edge patterns.

<Table 1> Generated edge classes

Edge type		Number of classes
Linear type		30
Nonlinear type	Convex type	6  (1) (2) (2) (1)
	Concave type	8  (2) (1) (2) (3)
	Hole type	4  (1) (1) (2)
	Bump type	4  (2) (1) (1)

(a) Original image (b) Reconstructed image

<Figure 4> Original and Reconstructed images

References

[1] R. M. Gray, "Vector Quantization", IEEE ASSP Mag., vol. 1, pp. 4~29, Apr. 1984.

[2] B. Ramamurthi and A. Gersho, "Classified Vector Quantization of Images", IEEE Trans. on Comm., vol. COM-34, pp. 1105~1115, Nov. 1986.

[3] F. Marques and C. C. J. Kuo, "Classified Vector Quantization Using Fuzzy Theory", IEEE International Conference on Fuzzy Systems, pp. 237~244, Mar. 1992.

[4] S. A. Mohamed and M. M. Fahmy, "Image Compression Using Block Pattern-Vector Quantization with Variable Codevector Dimensions", ICASSP-92, vol. 3, pp. 357~360, Mar., 1992.

[5] J. W. Kim and S. U. Lee, "A Transform Domain Classified Vector Quantizer for Image Coding", IEEE Trans. on Circuits and Systems for Video Technology, vol. 2, no. 1, pp. 3~14, Mar. 1992.

[6] J. T. Virtamo and S. T. Valli, "Vector quantization with hierarchical classification of sub-blocks", Signal Processing: Image Communication, vol. 8, no. 6, pp. 475~479, 1996.

[7] C. N. S. Ganesh Murthy and Y. V. V. Venkatesh, "Encoded pattern classification using constructive learning algorithms based on learning vector quantization", Neural Networks, vol. 11, no. 2, pp. 315~322, Mar. 1998.