LETTER Color Image Retrieval Based on Distance-Weighted Boundary Predictive Vector Quantization Index Histograms

Zhen SUN[†], Zhe-Ming LU^{††a)}, Nonmembers, and Hao LUO^{††}, Member

SUMMARY This Letter proposes a new kind of features for color image retrieval based on Distance-weighted Boundary Predictive Vector Quantization (DWBPVQ) Index Histograms. For each color image in the database, 6 histograms (2 for each color component) are calculated from the six corresponding DWBPVQ index sequences. The retrieval simulation results show that, compared with the traditional Spatial-domain Color-Histogram-based (SCH) features and the DCTVQ index histogram-based (DCTVQIH) features, the proposed DWBPVQIH features can greatly improve the recall and precision performance.

key words: content-based image retrieval (CBIR), predictive vector quantization (PVQ), distance-weighted boundary PVQ (DWBPVQ), index histograms

1. Introduction

In content-based image retrieval (CBIR) systems [1], instead of being manually annotated by keywords, images are indexed by their own visual content, such as color [2], texture [3], shape [4] and spatial relationship, which are more essential and closer to the human perceptual system than the keywords used in text-based image retrieval systems. In the past fifteen years, many researchers have shown great interests in image retrieval based on compressed-domains such as DCT [5], DWT [6], fractal coding [7] and Vector Quantization (VQ) [8]–[11]. Reference [5] obtains the texture features directly from the middle and low-frequency DCT coefficients. Reference [6] retrieves the images in 3 progressive steps using four lowest resolution subbands based on DWT.

This Letter focuses on VQ based CBIR schemes. Vector quantization [12] is not only an efficient data compression technique but also effective pattern recognition or clustering technique. Although VQ is not so efficient in image coding compared with JPEG and JPEG2000 based methods, however, it has been proved to be the effective feature extraction or feature space clustering technique in image retrieval [8]–[11]. Reference [8] uses tree-structured vector quantization (TSVQ) to organize the feature space as a tree. Reference [9] extracts features directly from the codeword indices of the spatial-domain VQ compressed image. Reference [10] extracts features from the individual codebook generated from the image. Reference [11] presents a new kind of features based on DCT-VQ index histograms. Above features are all based on constrained VQ or Basic VQ, while in this Letter, we will propose a novel kind of features based on a type of Feedback VQ, i.e., Distance-weighted Boundary Predictive VQ (DWBPVQ), which is first presented in Reference [13] for image compression.

2. Related Work

2.1 Vector Quantization

As we know, VQ is an efficient clustering and classification technique for high-dimensional spaces. In the spatialdomain VQ, a representative codebook should be generated offline based on a large training vector set using the wellknown LBG algorithm [12] before online encoding. During the encoding process, the image is first divided into blocks, each block being an input vector. For each input vector, we search in the codebook the nearest codeword for it. Then we use the codeword index to represent the input vector. During the decoding process, we only need a simple table-look-up procedure to obtain the corresponding reproduction vector from the codebook based on the index.

2.2 DCTVQ Index Histograms

Performing the VQ compression in transform domains such as DCT and DWT can reduce the complexity of basic VQ. DCT has the excellent energy compact property, thus we can throw the high-frequency information and only perform VQ on the low-frequency coefficients. On the other hand, we can utilize the normalized histograms to describe the features invariant to translation and nearly invariant to rotation and scaling. Considering the above two aspects, the features based on DCT-VQ index histograms presented in [11] can be described in the following paragraphs:

Obviously, we should first generate a representative codebook for the image database. We randomly select a certain number of images from the database to be the training images. Each image is divided into blocks of size 8×8 . All of these blocks compose the set $O = \{o_1, o_2, ..., o_N\}$, where N is the number of blocks. Then DCT is performed on each block in O to obtain the transformed set $T = \{t_1, t_2, ..., t_N\}$. We rearrange the transformed DCT block from the twodimensional array to the one-dimensional array in the zigzag sequence. We divide each transformed block t into

Manuscript received November 11, 2008.

Manuscript revised March 31, 2009.

[†]The author is with the Department of Automatic Test and Control, Harbin Institute of Technology, Harbin 150001, P. R. China.

^{††}The authors are with the School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, China.

a) E-mail: luzhem@mail.sysu.edu.cn

DOI: 10.1587/transinf.E92.D.1803

four parts, the DC coefficient *d*, 16 low-frequency coefficients denoted by the vector *l*, 9 middle-frequency coefficients denoted by the vector *m* and the remained high-frequency coefficients denoted by the vector *h*. Because high-frequency coefficients are relatively with smaller values, we deal with it in another way by computing the energy *e* of all high-frequency coefficients. Thus, we can compose four training sets, $D = \{d_1, d_2, ..., d_N\}, L = \{l_1, l_2, ..., l_N\}, M = \{m_1, m_2, ..., m_N\}$ and $E = \{e_1, e_2, ..., e_N\}$. Based on these four training sets, four codebooks, i.e., C_D with N_D codewords, C_L with N_L codewords, C_M with N_M codewords and C_E with N_E codewords, can be generated using the LBG algorithm [12] respectively. Note that there are three color components, thus we have 12 codebooks.

After obtaining the above 12 codebooks, we can then encode each image in the database with them. For any input image, we divide it into blocks of size 8×8 , and then perform DCT on each block. Rearrange and divide each DCT block into four parts, and then encode each part with the corresponding codebook to obtain the corresponding codeword index. For each part, we collect codeword indices of all DCT blocks to get an index sequence. Thus, we can obtain 12 index sequences in total for each image, 4 for each color component. Then we calculate the histogram for each index sequence to compose the final DCTVQIH features for each image.

3. Proposed Scheme

3.1 Boundary PVQ (BPVQ)

Boundary predictive VQ (BPVQ) is the simplest predictive VQ for image coding. The main idea of BPVQ is to predict the current block based on its boundary pixels in the up, left and up-left reconstructed blocks as shown in Fig. 1, and then the prediction error is vector quantized. The pixel *Y* in the current block can be predicted by the pixel X_1 in the left block, the pixel X_3 in the up block and the pixel X_2 in the up-left block as follows

$$\tilde{y}(i,j) = P(x_l(i,4), x_u(4,j), x_{ul}(4,4))$$
(1)

Where $\tilde{y}(i, j)$ denotes the predicted pixel at the location (i, j) in the current block, $1 \le i$, $j \le 4$ (here we assume that the block size is $4 \times 4 = 16$ for obtaining better quality together with low complexity), and $x_l(i, 4)$, $x_u(4, j)$ and $x_{ul}(4, 4)$ are the neighboring pixels in the left, up and up-left blocks respectively, and *P* is the prediction function. For simplicity, we generally define *P* as the averaging function over three neighboring pixels.

3.2 Distance-Weighted BPVQ (DWBPVQ)

In fact, in Fig. 1, the up-left block does not affect the prediction performance too much, thus we often ignore it. That is, we only need to consider pixels $x_l(i, 4)$ and $x_u(4, j)$, but ignore the relatively unimportant pixel $x_{ul}(4, 4)$. On the other



Fig. 1 The BPVQ prediction scheme.

hand, the spatial distance between Y and X_1 is often different from the spatial distance between Y and X_2 . The shorter distance is, the more correlation the correspoding neighboring pixel has. Thus, we can use the spatial distance as the weight during the prediction process, i.e.,

$$\tilde{y}(i,j) = \frac{i \cdot x_l(i,4) + j \cdot x_u(4,j)}{i+j}$$
(2)

Above method is called as Distance-Weighted BPVQ (DWBPVQ) in Reference [13].

3.3 Proposed DWBPVQ Index Histograms

Now, we introduce our feature extraction method based on DWBPVQ. Obviously, the first step is to generate the DWBPVQ codebooks for all color components. For each color component, we divide each image in the training set into blocks of size 4×4 . All of these blocks compose the set $X = \{x_1, x_2, ..., x_N\}$. Based on Eq. (2) and Fig. 1, we can get the corresponding predicted vector set $P = \{y_1, y_2, ..., y_N\}$. Subtracted each predicted block from its original block, we can obtain the residual set $Q = \{r_1, r_2, ..., r_N\}$, where $r_i = x_i - y_i, 1 \le i \le N$. Based on these two training sets, we generate corresponding two codebooks based on the LBG algorithm [12] respectively. For the set P, we generate the codebook C_P with N_P codewords. For the set Q, we generate the codebook C_Q with N_Q codewords. Note that there are three color components, thus we totally have 6 codebooks.

After obtaining the above 6 codebooks, we can then extract the features for each image in the database with them. For any input image based on certain color component, we divide it into blocks of size 4×4 . For the blocks in the first row or first column, we remain them unchanged. From the block in the second row and second column, we encode the image block by block, from left to right and from up to down. For each block x to be encoded, we first get the predicted blocks. Subtracted y from x, we get its residual vector r. Search the nearest codeword in Codebook C_P for y and use the corresponding index to encode r. Considering that we cannot get the original x again in the decoder, x should be replaced by

its reconstructed version $\hat{x} = y + \hat{r}$, where \hat{r} is the quantized version of r. Thus for all y, we collect their codeword indices to get a "predicted" index sequence. While for all r, we collect their codeword indices to get a "residual" index sequence. Thus, we can obtain 6 index sequences in total for each image. Then we calculate the histogram for each index sequence to compose the features for each image. Note that, here the "predicted" index histogram can reflect the rough information of the image, and the "residual" index histogram can reflect the detail information of the image.

4. Experimental Results and Discussions

To demonstrate the effectiveness of the proposed features, we compare our DWBPVQIH features with traditional spatial-domain color-histogram-based (SCH) features and DCT-domain vector quantization index histograms-based (DCTVOIH) features based on the same YCbCr color space and the same feature dimension of 192. We use a standard database [14] in the experiment that is carried out on a Pentium IV computer with the 2.80 GHz CPU. This database includes 1000 images of size 384×256 or 256×384 , which are classified into ten classes (people,), each class including 100 images. We first randomly select 2 images from each class to be the training images, and then generate the required codebooks for DCTVQ and DWBPVQ as described in Sects. 2.2 and 3.3 respectively. For DCTVQ, we perform the DCT on all 8×8 blocks and compose four training sets, and then generate four codebooks for each color component, $N_D = 64, N_L = 512, N_M = 256, N_E = 128$. Based on these codebooks, we encode each image to get 12 index sequences, and then we use a 16-bin histogram to represent each index sequence, thus we can get a 192-dimensional feature vector for each color image in the database. For DWBPVQ, we divide all training images into blocks of size 4×4 , and then generate the two codebooks, the predictive codebook C_P with $N_P = 256$ codewords, and the quantization codebook C_O with N_O codewords. Based on these codebooks, we encode each image to get 6 index sequences, and then we use 32-bin histogram to represent each index sequence, thus we can get a 192-dimensional feature vector for each color image in the database. To show the performance of the codebooks for all color components (Note that the encoding quality of the codebook is not so important in the retrieval applications), we list the average encoding qualities based on DCTVQ (bit rate = (6 + 9 + 8 + 7)/64 = 0.47 bpp) and DWBPVQ in Table 1. For DWBPVQ, we test three cases, i.e., (1) $N_Q = 64$ (bit rate = 6/16 = 0.375 bpp); (2) $N_Q = 128$ (bit rate = 7/16 = 0.44 bpp); (3) $N_Q = 256$ (bit rate = 0.5 bpp). For comparisons, based on the YCbCr color space, we also extract three 64-bin color histograms from each image to compose a 192-dimensional SCH feature vector. To compare the performance more reasonably, we randomly select 5 images from each class, and thus in total 50 images, as the test query images. For each test query image, we perform the retrieval process based on each kind of features. For each number of returned images (from 1 to 1000), we average the recall and precision value over 50 test query images.

The comparisons of the average P-R curves of SCH, DCTVQIH and DWBPVQIH ($N_Q = 256$) are shown in Fig. 2. From these results, we know that we can obtain much better recall and precision performance with the proposed features than that with the traditional SCH features. In addition, the proposed DWBPVQIH features are also better than the DCTVQIH features over the whole P-R Curve. The

 Table 1
 Comparisons of the average encoding quality using different compression methods and with different parameters.

	Parameters	Average encoding quality		
Method		(dB)		
		Y	Cb	Cr
DCTVQ	$N_D = 64, N_L = 512$	27.12	43.17	43.47
	$N_M = 256, N_E = 128$			
DWBPVQ	$N_Q=64$	25.87	44.02	44.85
	$N_{O} = 128$	26.91	45.61	46.08
	$N_{O} = 256$	28.14	47.06	46.87



Fig. 2 The performance comparisons among the proposed DWBPVQIH features, DCTVQIH features and SCH features.



Fig. 3 The performance comparisons of the proposed DWBPVQIH features under different N_Q values.

comparisons of the average P-R curves of DWBPVQIH with different N_Q values are shown in Fig. 3. We can see that, the more bit rate we use the much better retrieval results we can obtain for DWBPVQIH features.

5. Conclusions

This paper presents a new kind of features based on DWBPVQ for color image retrieval. The first advantage is that the proposed features can be quickly extracted from the compressed data. Secondly, during the encoding or decoding of DWBPVQ, there are two parts information, one is for prediction, the other is for quantization. The first part can be viewed as the rough information, while the second part can be viewed as detail information. Experimental results show that the proposed DWBPVQIH features are much better than the traditional SCH features and also better than the DCTVQIH features.

References

- F.H. Long, H.J. Zhang, and D.D. Feng, "Fundamentals of contentbased image retrieval," in Multimedia Information Retrieval and Management-Technological Fundamentals and Applications, ed. D.D. Feng, W.C. Siu, and H.J. Zhang, pp.1–26, Springer-Verlag, New York, 2003.
- [2] A. Yamada, E. Kasutani, M. Ohta, H. Ochiai, and H. Matoba, "Visual program navigation system based on spatial distribution of color," Proc. IEEE International Conference on Consumer Electronics, pp.280–281, 2000.
- [3] J.P.K. Kuan, D.W. Joyce, and P.H. Lewis, "Texture content based retrieval using text descriptions," SPIE Proc. Storage and Retrieval

for Image and Video Databases VII, vol.3656, pp.75-85, 1999.

- [4] K. Lee and W.N. Street, "Incremental feature weight learning and its application to a shape based query system," Pattern Recognit. Lett., vol.23, no.7, pp.865–874, May 2002.
- [5] D.G. Sim, H.K. Kim, and R.H. Park, "Fast texture description and retrieval of DCT based compressed images," Electron. Lett., vol.37, no.1, pp.18–19, 2001.
- [6] J.Z. Wang, G. Wiederhold, and O. Firschain, "Wavelet-based image indexing techniques with partial sketch retrieval capability," Proc. Forum on Research and Technology Advances in Digital Libraries, pp.13–24, 1997.
- [7] A.D. Zhang, B. Cheng, R.S. Acharya, and R.P. Menon, "Comparison of wavelet transforms and fractal coding in texture-based image retrieval," Visual Data Exploration and Analysis III, vol.2656, pp.116–125, 1996.
- [8] S. Panchanathan and C. Huang, "Indexing and retrieval of color images using vector quantization," SPIE Proc. Applications of Digital Image Processing XXII, vol.3808, pp.558–568, 1999.
- [9] T. Uchiyama, N. Takekawa, H. Kaneko, M. Yamaguchi, and N. Ohyama, "Multispectral image retrieval using vector quantization," Proc. IEEE International Conference on Image Processing, vol.1, pp.30–33, 2001.
- [10] F. Idris and S. Panchanathan, "Storage and retrieval of compressed images," IEEE Consum. Electron., vol.41, no.3, pp.937–941, 1995.
- [11] Z.-M. Lu and H. Burkhardt, "Colour image retrieval based on DCT-domain vector quantisation index histograms," Electron. Lett., vol.41, no.17, pp.956–957, 2005.
- [12] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun., vol.28, no.1, pp.84–95, 1980.
- [13] B. Yang, Z.M. Lu, D.G. Guo, and S.H. Sun, "Neighboring pixels based low complexity predictive vector quantization algorithms for image coding," Acta Electronica Sinica, vol.31, no.5, pp.707–710, 2003.
- [14] J. Li, Photography image database. http://www.stat.psu.edu /~jiali/ index.download. html