International Journal of Innovative Computing, Information and Control
Volume **2**, Number **6**, December **2006**

ICIC International ©2006 ISSN 1349-4198
pp. 1317–1326

COLOR IMAGE RETRIEVAL SCHEMES USING INDEX HISTOGRAMS BASED ON VARIOUS SPATIAL-DOMAIN VECTOR QUANTIZERS

Wei-Min Zheng[1], Zhe-Ming Lu[2,3] and Hans Burkhardt[2]

[1]Institute of Computing Technology
Chinese Academy of Sciences
Beijing 100080, P. R. China
zhengwm@ict.ac.cn

[2]Institute for Computer Science
University of Freiburg
Georges-Koehler-Allee 052, room 01-046, 79110 Freiburg, Germany
{zheminglu; Hans.Burkhardt}@informatik.uni-freiburg.de

[3]Visual Information Analysis and Processing Research Center
Harbin Institute of Technology Shenzhen Graduate School
Room 202K, Building No.4, HIT Campus Shenzhen University Town, Xili
Shenzhen 518055, P. R. China
zhemingl@yahoo.com

Received October 2005; revised January 2006

Abstract. *This paper proposes two new compressed-domain features for color image retrieval based on the YCbCr color space. They are named Multi-Stage Vector Quantization Index Histograms (MSVQIH) and Mean-Removed Vector Quantization Index Histograms (MRVQIH). For each color component, to obtain the MSVQIH features, we extract two MSVQ Index histograms from the two stage VQ index sequences respectively. Similarly, to obtain the MRVQIH features, we extract two MRVQ index histograms from the mean index and residual VQ index sequences respectively. The retrieval simulation results show that, compared with the traditional Spatial-domain Color-Histogram-based (SCH) features and Vector Quantization Index Histograms-based (VQIH) features, the proposed two kinds of features can largely improve the recall and precision performance.*
Keywords: Image retrieval, Vector quantization, Index histogram

1. **Introduction.** With the rapid development of computer, multimedia and network technologies, a large amount of image and video data are created and broadly distributed over the Internet or via CD-ROM. However, without effective image retrieval it is impossible to make use of the huge image databases. There are basically three ways of retrieving previously stored multimedia data, i.e., free browsing, text-based retrieval and content-based retrieval. Free browsing is only acceptable for the occasional user and cannot be extended to users who frequently need to retrieve specific multimedia information for professional applications. It is a tedious, inefficient, and time-consuming process and it becomes completely impractical for large databases. Text-based retrieval has two big problems associated with the cataloging phase: 1) the considerable amount of time and

effort needed to manually annotate each individual image or clip; and 2) the imprecision associated with the subjective human perception of the contents being annotated. In a content-based image retrieval (CBIR) system [1], instead of being manually annotated by keywords, images are indexed by their own visual content, such as color [2], texture [3] and shape [4], which are more essential and closer to the human perceptual system than the keywords used in a text-based image retrieval system.

Recently, many researchers have shown great interests in image retrieval based on compressed-domains such as vector quantization (VQ) [5-7], DCT [8], DWT [9] and Fractal coding [10]. Reference [5] extracts features directly from the codeword indices of each spatial-domain VQ compressed image. Reference [6] extracts features from the individual codebook generated from each image. Reference [7] uses the index histogram of VQ-compressed index sequence to describe the features, which are denoted as VQ Index Histograms (VQIH) in this paper. Reference [8] obtains the texture features directly from the middle and low-frequency DCT coefficients. Reference [9] retrieves the images in 3 progressive steps using four lowest resolution subbands based on DWT. Reference [10] shows that, compared with DWT, retrieval based on fractal coding can obtain better results for the databases with various kinds of images, but obtains worse results for texture databases. Unlike the reference [7], based on another two kinds of spatial-domain vector quantization, multi-stage vector quantization (MSVQ) and mean-removed vector quantization (MRVQ), this paper presents two new kinds of features denoted by MSVQ Index Histograms (MSVQIH) and MRVQ Index Histograms (MRVQIH). The remainder of this paper is organized as follows. In Section 2, three sub-sections describe the VQ/VQIH, MSVQ/MSVQIH and MRVQ/MRVQIH respectively. The simulation results and conclusions are given in Section 3 and Section 4, respectively.

2. **Proposed MSVQIH and MRVQIH.** In this section, we first give the basic idea of the VQIH features, and then present the proposed MSVQIH and MRVQIH features respectively.

2.1. **Vector quantization and VQIH.** Vector quantization (VQ) is an efficient block-based glossy image compression technique with a high compression ratio and a simple table lookup decoder. VQ can be defined as a mapping from k-dimensional Euclidean space R^k into a finite codebook $C=\{\boldsymbol{c}_i| \quad i=0, 1,\ldots,N\text{-}1\}$, where $\boldsymbol{c}_i$ is called codeword and N is the codebook size. Before online encoding, VQ first generates a representative codebook offline from a number of training vectors using the well-known GLA algorithm [11]. In image vector quantization, the image to be encoded is first segmented into vectors and then sequentially encoded vector by vector. In the encoding stage, for each k-dimensional input vector $\boldsymbol{x}=(x_1, x_2,\ldots, x_k)$, we find its nearest neighbor codeword $\boldsymbol{c}_i=(c_{i1},c_{i2}, \ldots, c_{ik})$ in the codebook $C=\{\boldsymbol{c}_0,\boldsymbol{c}_1, \ldots \boldsymbol{c}_{N-1}\}$, which satisfies the following condition:

$$d(\boldsymbol{x}, \boldsymbol{c}_i) = \min_{0\leq j\leq N-1} d(\boldsymbol{x}, \boldsymbol{c}_j) \tag{1}$$

where $d(\boldsymbol{x}, \boldsymbol{c}_j)$ is the distortion between the input vector $\boldsymbol{x}$ and the codeword $\boldsymbol{c}_j$, which can be defined as follows

$$d(\boldsymbol{x}, \boldsymbol{c}_j) = \sum_{l=1}^{k} (x_l - c_{jl})^2 \tag{2}$$

And then the index i of the nearest neighbor codeword assigned to the input vector $\boldsymbol{x}$ is transmitted over the channel to the decoder. The decoder has the same codebook as the encoder. In the decoding phase, for each index i, the decoder merely performs a simple table look-up operation to obtain $\boldsymbol{c}_i$ and then uses $\boldsymbol{c}_i$ to reconstruct the input vector $\boldsymbol{x}$. Compression is achieved by transmitting or storing the index of a codeword rather than the codeword itself.

For each image, the VQIH feature extraction process for each color component can be illustrated as follows: (1) Divide the image into blocks with the same size of each codeword, typically 4×4. (2) For each block, find its nearest neighbor codeword in the codebook, and assign the codeword's index to this block. (3) Calculate the index histogram from the obtained codeword index sequence, with the number of bins being the number of codewords in the codebook. Here the number of bins is selected to be 64. Because there are three color components, we can in total get a 192-dimensional feature vector for each image.

2.2. **Multi-stage vector quantization and MSVQIH.** The basic idea of multistage VQ is to divide the encoding task into successive stages, where the first stage performs a relatively crude quantization of the input vector using a small codebook. Then, a second stage quantizer operates on the error vector between the original and quantized first stage output. The quantized error vector then provides a second approximation to the original input vector thereby leading to a refined or more accurate representation of the input. A third stage quantizer may then be used to quantize the second stage error to provide a further refinement and so on.

In this paper, we adopt a two-stage vector quantizer as illustrated in Figure 1. It is the simplest case and can be used to generate the general multistage vector quantizer. The input vector $\boldsymbol{x}$ is quantized by the initial or first stage vector quantizer denoted by VQ_1 whose codebook is $C_1=\{\boldsymbol{c}_{10}, \boldsymbol{c}_{11}, \ldots, \boldsymbol{c}_{1(N1-1)}\}$ with size N_1. The quantized approximation $\hat{\boldsymbol{x}}_1$ is then subtracted from $\boldsymbol{x}$ producing the error vector $\boldsymbol{e}_2$. This error vector is then applied to a second vector quantizer VQ_2 whose codebook is $C_2=\{\boldsymbol{c}_{20}, \boldsymbol{c}_{21}, \ldots, \boldsymbol{c}_{2(N2-1)}\}$ with size N_2 yielding the quantized output $\hat{\boldsymbol{e}}_2$. The overall approximation $\hat{\boldsymbol{x}}$ to the input $\boldsymbol{x}$ is formed by summing the first and second approximations, $\hat{\boldsymbol{x}}_1$ and $\hat{\boldsymbol{e}}_2$. The encoder for this VQ simply transmits a pair of indices specifying the selected codewords for each stage and the task of the decoder is to perform two table lookups to generate and then sum the two codewords. In fact, the overall codeword or index is the concatenation of codewords or indices chosen from each of two codebooks. That is to say, this is a product code where the composition function g of the decoder is simply a summation of the reproductions from the different two VQ decoders. Thus the equivalent product codebook C can be generated from the Cartesian product $C_1 \times C_2$. Compared to the full search VQ with the product codebook C, the two-stage VQ can reduce the complexity from $N = N_1 \times N_2$ to $N_1 + N_2$. In other words, with the same complexity, MSVQ codebook can represent more codewords than that of VQ.

For each image, the MSVQIH feature extraction process for each color component can be illustrated as follows: (1) Divide the image into blocks with the same size of each codeword, typically 4×4. (2) For each block $\boldsymbol{x}$, find its nearest neighbor codeword in the first codebook C_1, and then assign the codeword's index to this block. (3) For the error vector $\boldsymbol{e}_2$, find its nearest neighbor codeword in the second codebook C_2, and

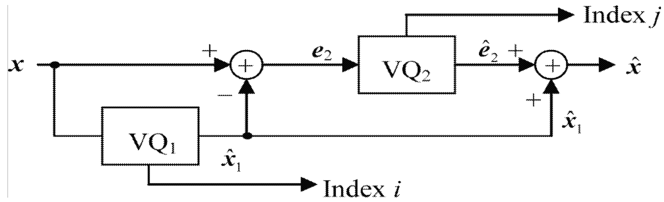

FIGURE 1. Two-stage VQ

also assign this index to the block. Thus each input block corresponds to two indices. (3) Calculate two index histograms from the two obtained codeword index sequences respectively. Here, the number of bins for each index histogram is the same as the number of codewords in the corresponding codebook. We test three cases with N_1+N_2=64 in the experiments. Because there are three color components, for any case, we can in total get a 192-dimensional feature vector for each image.

2.3. **Mean-removed vector quantization and MRVQIH.** Usually we deal with vectors that have zero statistical mean in the sense that the expected value of each component is zero. Nevertheless, many vectors such as sampled image intensity rasters have only non-negative components and hence have nonzero means. The local means over small blocks can vary quite widely over an image. Furthermore, this mean of an image vector can often be regarded as statistically independent of the variation of the vector, that is, of the way the components vary about this average. The term *mean* of a vector is used in this section specifically to refer to the *sample mean*, i.e., the average of all the components in a vector. Thus the mean, m, of the vector $\boldsymbol{x}$ is a scalar random variable given by

$$m = \frac{1}{k}\sum_{i=1}^{k} x_i = \frac{\mathbf{1}}{k}1^t\boldsymbol{x} \tag{3}$$

where $\mathbf{1}$=(1,1,...,1)^t, the k-dimensional vector with all components equal to unity. The mean-removed residual, $\boldsymbol{r}$, of the random variable $\boldsymbol{x}$ is defined as

$$\boldsymbol{r} = \boldsymbol{x} - \frac{1}{k}(\mathbf{1}^t\boldsymbol{x})\mathbf{1} = \boldsymbol{x} - m\mathbf{1} \tag{4}$$

Hence, $\boldsymbol{x}$ can be represented as the sum of a mean vector $m\mathbf{1}$ and the residual vector $\boldsymbol{r}$ according to:

$$\boldsymbol{x} = \boldsymbol{r} + m\mathbf{1} \tag{5}$$

The residual $\boldsymbol{r}$ is the "mean-removed" version of the vector $\boldsymbol{x}$ and has zero mean. Thus we have a natural decomposition of the original vector into separate features, a mean (representing a general background level) and a residual (representing the shape of the vector about its mean). Quantizing these features using separate codebooks is referred to as MRVQ for "mean-removed VQ" or "mean-residual VQ."

In this paper, we use the encoding structure depicted in Figure 2. The mean of $\boldsymbol{x}$ is first computed and quantized with a mean codebook C_m, and the quantized mean $\hat{m}$ is then subtracted from each component of $\boldsymbol{x}$ to obtain the residual vector $\boldsymbol{r}$. Note that here the residual is computed with respect to the decoder's reproduction of the mean rather than with respect to the true mean. The residual vector $\boldsymbol{r}$ is then quantized with a residual codebook C_r. The output of the encoder includes two indices for the mean and residual, respectively. The representation of $\boldsymbol{x}$ offers a simple, and very valuable product code. The reconstructed vector after quantization of the mean and the residual is given by

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{r}} + \hat{m}\mathbf{1} \tag{6}$$

where $\hat{m}$ is a quantization level from a scalar codebook C_m of size N_m for the mean code levels, and $\hat{r}$ is a codeword chosen from a codebook of size N_r for the residual vectors. In fact, the overall codeword or index is the concatenation of codewords or indices chosen from each of two codebooks. That is to say, this is a product code where the composition function g of the decoder is simply a summation of the reproductions from the different two quantizers. Thus, the equivalent codebook for $\boldsymbol{x}$ is the product codebook C that can be generated from the Cartesian product $C_m \times C_r$. Compared to the full search VQ with the product codebook C, the mean-removed VQ can reduce the complexity from $N = N_m \times N_r$ to $N_m/k + N_r$. In other words, with the same complexity, MRVQ codebook can represent more codewords than that of VQ.

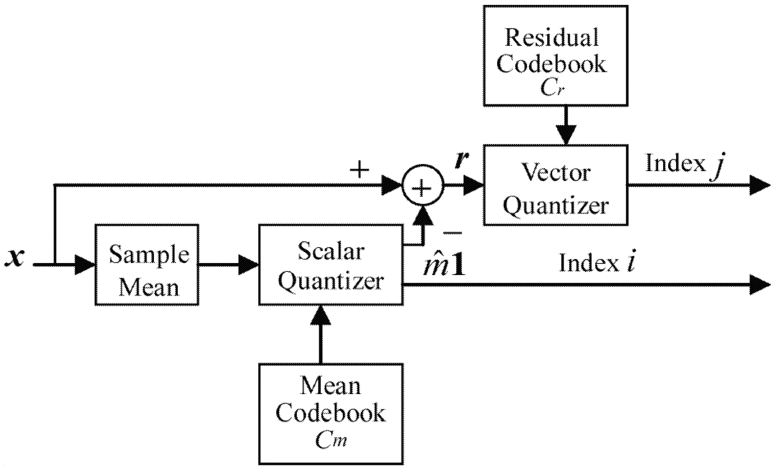

FIGURE 2. Mean-removed VQ

For each image, the MRVQIH feature extraction process for each color component can be illustrated as follows: (1) Divide the image into blocks with the same size of each codeword, typically 4×4. (2) For each block, calculate its mean value first, and then find its nearest neighbor codeword in the mean codebook C_m, and assign the codeword's index to this block. (3) Remove the mean value from each dimension of the block to get the residual vector, and then find its nearest neighbor codeword in the residual codebook C_r, and also assign this index to the block. Thus each input block corresponds to two indices. (3) Calculate two index histograms from the two obtained codeword index sequences respectively. Here, the number of bins for each index histogram is the same as the number of codewords in the corresponding codebook. We test three cases with $N_m + N_r$=64 in the experiments. Because there are three color components, for any case, we can in total get a 192-dimensional feature vector for each image.

3. **Experimental Results.** To demonstrate the effectiveness of the proposed features, we compare our MSVQIH and MRVQIH features with traditional spatial-domain color-histogram-based (SCH) features and vector quantization index histograms-based (VQIH)

features based on the same YCbCr color space and the same feature dimension of 192. We use a standard database [12] in the experiment that is carried out on a Pentium IV computer with the 2.80GHz CPU. This database includes 1000 images of size 384×256 or 256×384, which are classified into ten classes, each class including 100 images. We first randomly select 2 images from each class to be the training images, and then generate the required codebook for VQ (N=64), 2-stage codebooks ($N_1 + N_2$=64) for MSVQ and the required mean and residual VQ codebooks ($N_m + N_r$=64) for MRVQ as described in Sections 2.2 and 2.3, with the size of each block being 4×4. 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 VQ, MSVQ and MRVQ in Table 1. For MSVQ, we test three cases, i.e., (1) N_1=32, N_2=32; (2)N_1=16, N_2=48; (3) N_1=8, N_2=56. For MRVQ, we test the following three cases: (1) N_m=32, N_r=32; (2)N_m=16, N_r=48; (3) N_m=8, N_r=56. Then we calculate the proposed 192-dimensional MSVQIH feature vector and 192-dimensional MRVQIH feature vector for each color image respectively. 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. Here, the precision and recall are defined as follows:

$$\begin{aligned} \text{precision} &= \frac{\text{No. relevant images}}{\text{No. images returned}} \\ \text{recall} &= \frac{\text{No. relevant images}}{100} \end{aligned} \tag{7}$$

The comparisons of the average P-R curves of SCH, VQIH, MSVQIH ($N_1 = N_2$=32) and MRVQIH ($N_m = N_r$=32) are shown in Figure 3. The comparisons of the average P-R curves of MSVQIH with different N_1 and N_2 are shown in Figure 4. The comparisons of the average P-R curves of MRVQIH with different N_m and N_r are shown in Figure 5. From these results, we know that we can obtain much better recall and precision performance with the proposed two kinds of features than that with the traditional SCH features. In addition, the proposed MSVQIH and MRVQIH features are also better than the VQIH features over the whole P-R Curve. We can also note that the performances of MSVQIH (or MRVQIH) with different parameters are nearly the same.

From Table 1, we can see that, for the Y component, we should use more codewords to get better encoding quality for the dynamic range of the Y component is much larger than that of the Cb or Cr component. Thus we also do another experiment to show the performance with different number of codewords for different component. In this experiment, we use N=128, $N_1 = N_2$=64, $N_m = N_r$=64 for the Y component and use N=32, $N_1 = N_2$=16, $N_m = N_r$=16 for the Cb or Cr component. The comparison of the average encoding quality is given in Table 2, and the comparison of the P-R curve is given in Figure 6. Comparing Table 2 with Table 1, we can see that the improvement in the encoding quality for the Y component is obvious, while the decrease in the encoding quality for the Cb or Cr component is acceptable. From Figure 6, we can see that, in the case of allocating more codewords to the Y component, the proposed MSVQIH and MRVQIH methods also have better performance than the VQIH and SCH methods.

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

Method	Parameters	Average encoding quality (dB)		
		Y	Cb	Cr
VQ	N=64	25.97	43.74	43.57
	N_1=32,N_2=32	27.70	47.08	47.72
MSVQ	N_1=16,N_2=48	27.53	46.87	46.99
	N_1=8,N_2=56	27.33	46.27	46,20
	N_m=32,N_r=32	27.55	47.21	48,22
MRVQ	N_m=16,N_r=48	27.58	46.96	46.98
	N_m=8,N_r=56	27.45	45.81	46.10

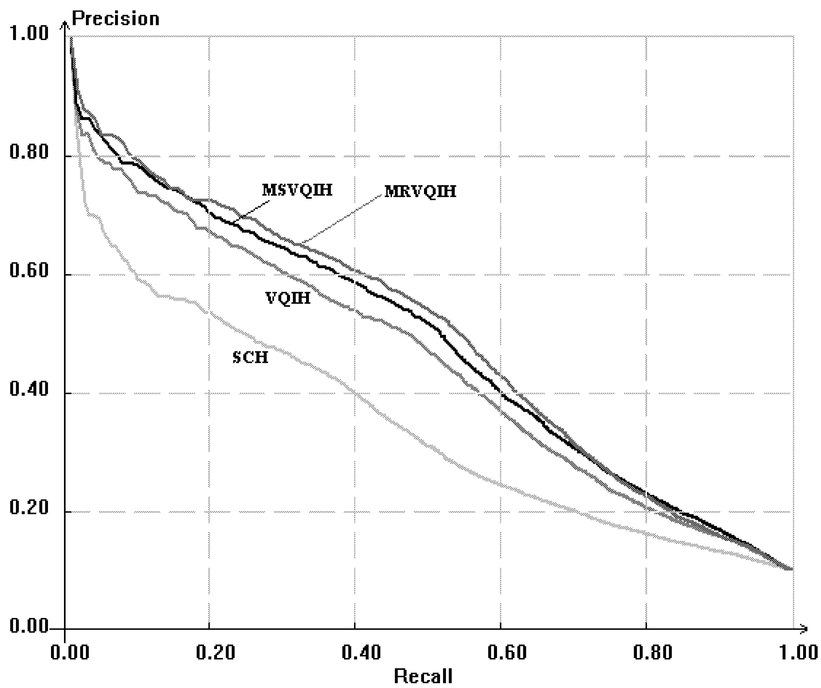

Figure 3. The comparisons of the average P-R curves of SCH, VQIH, MSVQIH (N_1 = N_2=32) and MRVQIH (N_m = N_r=32)

However, comparing Figure 6 with Figure 3, we can see that the retrieval performance of each method decreases slightly. The reason is that the number of codewords for Cb or Cr component are too few for feature description in this case. From this fact, we also know that the better encoding quality cannot guarantee the better retrieval performance.

4. **Conclusions.** This paper presents new features based on two kinds of VQs for color image retrieval. The first advantage is that the proposed features can be quickly extracted from the compressed data. Secondly, because the two stages of MSVQ reflect the rough and detailed information distribution of each image respectively, and the mean

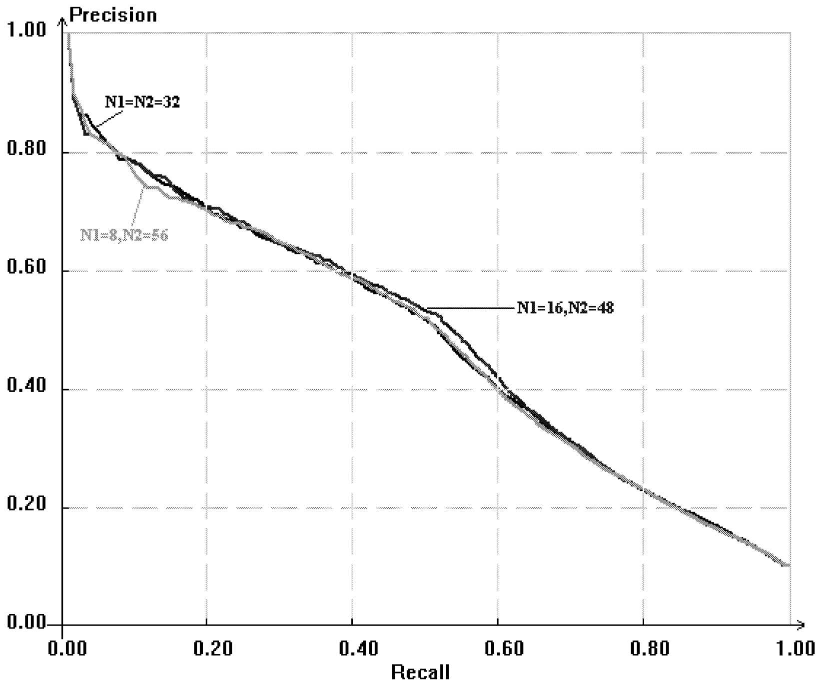

FIGURE 4. Comparisons of the average P-R curves of MSVQIH with different N_1 and N_2

TABLE 2. Comparisons of the average encoding quality using different compression methods and with different parameters for Y, Cb and Cr components

Method	Average encoding quality (dB) Y ($N = 128$, $N_1 = N_2 = 64$, $N_m = N_r = 64$)	Cb ($N = 32$, $N_1 = N_2 = 16$, $N_m = N_r = 16$)	Cr ($N = 32$, $N_1 = N_2 = 16$, $N_m = N_r = 16$)
VQ	26.79	41.77	42.16
MSVQ	28.67	44.81	44.46
MRVQ	28.35	44.64	45.23

and residual parts of MRVQ can reflect the average luminance and detailed information distribution of each image respectively, the proposed MSVQIH and MRVQIH features are much better than the traditional SCH features and also better than the VQIH features. In fact, MSVQ and MRVQ can have more equivalent codewords than VQ (for example, in Table 1 and Figures 3-5, 32×32>16×48>8×56>>64), thus their retrieval performances are better than VQIH features. Future work will concentrate on how to adaptively extract different kinds of features from different (e. g., background and foreground) parts of the image.

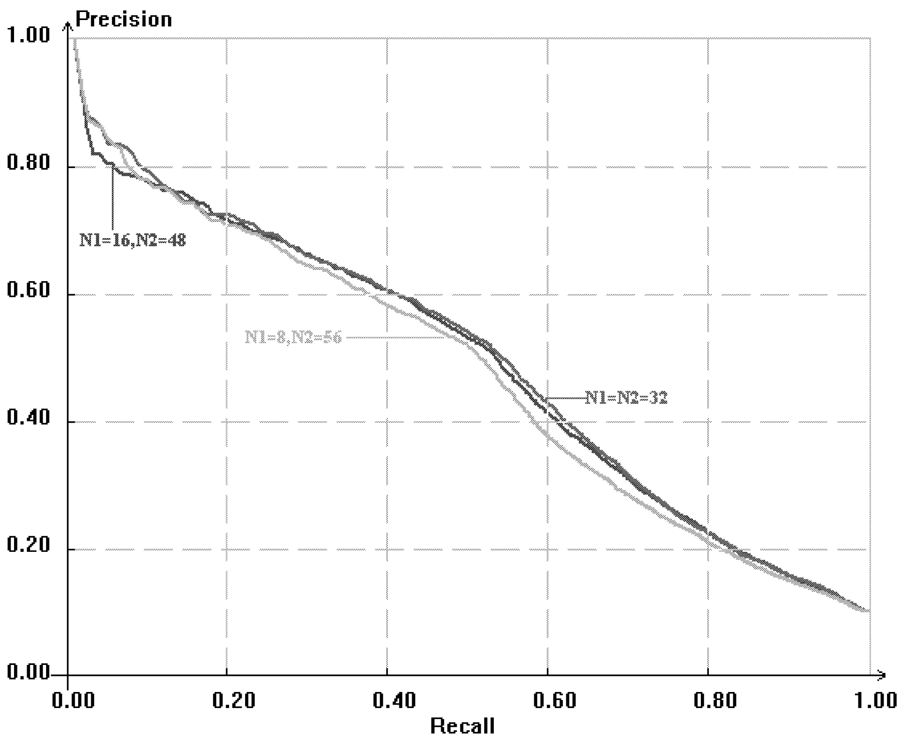

FIGURE 5. Comparisons of the average P-R curves of MRVQIH with different N_m and N_r

Acknowledgment. This work was supported by the Alexander von Humboldt Foundation Fellowship (Germany) grant, ID: CHN 1115969 STP. The authors also gratefully thank Prof. Jia Li for providing the image database for research.

REFERENCES

[1] Long, F. H., H. J. Zhang and D. D. Feng, Fundamentals of content-based image retrieval, in *Multimedia Information Retrieval and Management - Technological Fundamentals and Applications*, D. D. Feng, W. C. Siu and H. J. Zhang (eds.), Springer-Verlag, New York, pp.1-26, 2003.

[2] Yamada, A., E. Kasutani, M. Ohta, H. Ochiai and H. Matoba, Visual program navigation system based on spatial distribution of color, *Proc. of the IEEE International Conference on Consumer Electronics*, pp.280-281, 2000.

[3] Kuan, J. P. K., D. W. Joyce and P. H. Lewis, Texture content based retrieval using text descriptions, *SPIE Proceedings of Storage and Retrieval for Image and Video Databases VII*, vol.3656, pp.75-85, 1999.

[4] Lee, K. and W. N. Street, Incremental feature weight learning and its application to a shape based query system, *Pattern Recognition Letters*, vol.23, no.7, pp.865-874, 2002.

[5] Panchanathan, S. and C. Huang, Indexing and retrieval of color images using vector quantization, *SPIE Proceedings of Applications of Digital Image Processing XXII*, vol.3808, pp.558-568, 1999.

[6] Uchiyama, T., N. Takekawa, H. Kaneko, M. Yamaguchi and N. Ohyama, Multispectral image retrieval using vector quantization, *Proc. of the IEEE International Conference on Image Processing*, vol.1, pp.30-33, 2001.

[7] Idris, F. and S. Panchanathan, Storage and retrieval of compressed images, *IEEE Consumer Electronics*, vol.41, no.3, pp.937-941, 1995.
[8] Sim, D. G., H. K. Kim and R. H. Park, Fast texture description and retrieval of DCT based compressed images, *Electronics Letters*, vol.37, no.1, pp.18-19, 2001.
[9] Wang, J. Z., G. Wiederhold and O. Firschain, Wavelet-based image indexing techniques with partial sketch retrieval capability, *Proc. of the Forum on Research and Technology Advances in Digital Libraries*, pp.13-24, 1997.
[10] Zhang, A. D., 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.
[11] Linde, Y., A. Buzo and R. M. Gray, An algorithm for vector quantizer design, *IEEE Trans. Commun.*, vol.28, no.1, pp.84-95, 1980.
[12] Li, J., Photography image database. http://www.stat.psu.edu /~jiali/index.download. html

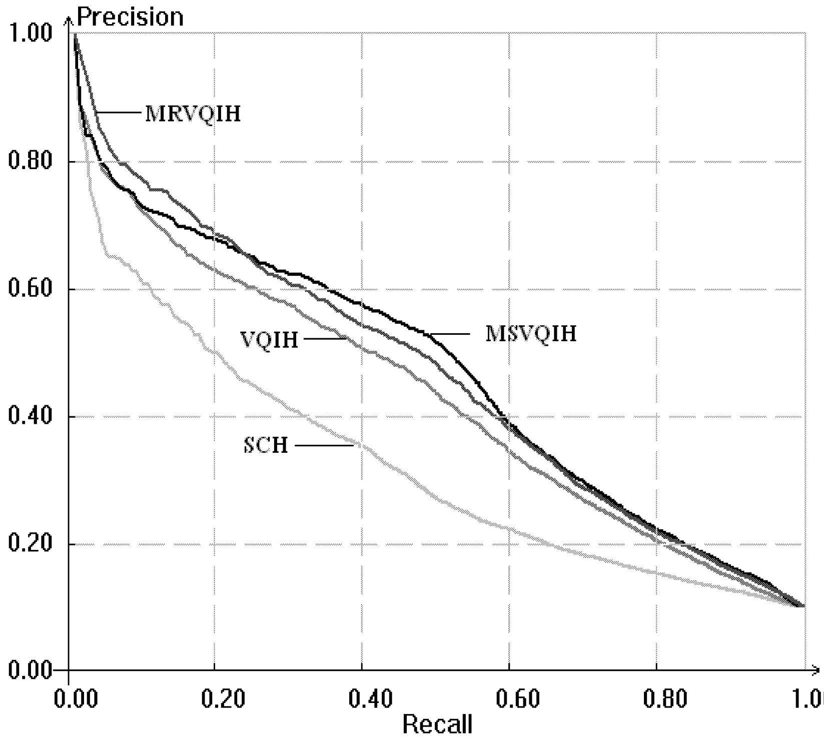

FIGURE 6. The comparisons of the average P-R curves of SCH(the length of histogram, Y:128, Cb:32, Cr:32), VQIH(Y: N=128; Cb, Cr: N=32), MSVQIH (Y: $N_1 = N_2$=64; Cb, Cr: $N_1 = N_2$=16) and MRVQIH (Y: $N_m = N_r$=64; Cb, Cr:$N_m = N_r$=16)